Practical Reasoning and Rationality

Bell, John; Huang, Zhisheng

724

Practical Reasoning and Rationality

Proceedings of the DRUMS II Workshop

Edited by **John Bell**
and **Zhisheng Huang**

Practical Reasoning and Rationality

Proceedings of the DRUMS II Workshop

Edited by

John Bell and Zhisheng Huang

Applied Logic Group
Computer Science Department
Queen Mary and Westfield College
University of London

9-11 February, 1996
Sir Christopher Wren's House,
WINDSOR,
UNITED KINGDOM

Preface

This workshop formed part of the DRUMS II Project which was funded by the European Union (Esprit III BRA 6156).

The theme was, intentionally, a broad one. A comprehensive logical theory of practical reasoning and rationality will include theoretical reasoning (reasoning about what is the case), practical reasoning (reasoning about motivational attitudes such as desires, goals, intentions and obligations) and commonsense causal reasoning about actions and their effects.

The papers in this volume reflect the presentations at the workshop and have notionally been grouped under the headings theoretical reasoning, practical reasoning and reasoning about action.

The workshop was a great success and it is hoped that it will be the first in a regular series of workshops on this topic.

John Bell and Zhisheng Huang

Contents

Theoretical Reasoning

Practical Reasoning

Reasoning About Actions

Defining normative systems for qualitative argumentation

Simon Parsons[1 2]

[1] Advanced Computation Laboratory, Imperial Cancer Research Fund,
P.O. Box 123, Lincoln's Inn Fields, London WC2A 3PX, United Kingdom.
[2] Department of Electronic Engineering, Queen Mary and Westfield College,
Mile End Road, London E1 4NS.
S.Parsons@qmw.ac.uk

Topic: Argumentation theory

Abstract. Inspired by two different approaches to providing a qualitative method for reasoning under uncertainty—qualitative probabilistic networks and systems of argumentation—this paper attempts to combine the advantages of both by defining systems of argumentation that have a probabilistic semantics.

1 Introduction

In the last few years there have been a number of attempts to build systems for reasoning under uncertainty that are of a qualitative nature—that is they use qualitative rather than numerical values, dealing with concepts such as increases in belief and the relative magnitude of values. In particular, two types of qualitative system have become well established, namely qualitative probabilistic networks (QPNs) [5, 19], and systems of argumentation [9, 13, 14]. While the former are built as an abstraction of probabilistic networks where the links between nodes are only modelled in terms of the qualitative influence of the parents on the children, and therefore have an underlying probabilistic semantics, some of the latter lack such a sound foundation. Instead they offer a greater degree of resolution, allowing more precise deductions to be made.

In this paper we present several normative systems of argumentation that both extend QPNs in the sense of reducing the degree of abstraction of such systems, and extend argumentation in the sense of providing it with a probabilistic semantics whilst using only qualitative or semi-qualitative information[3]. Of course this extension might not always be desired, but may be useful at times to ensure that a given system reasons within probabilistic norms. The systems are built upon that of Fox et al. [9, 13], which is introduced in Section 2. It turns out that it is possible to give this system a probabilistic semantics, thus "creating" a normative system of argumentation, which, while similar in some respects to QPNs, is rather different in others, and this is done in Section 3. Then, in

[3] If we don't have any commitment to qualitative information, we can use ordinary probabilities as suggested by Krause et al. [13].

$$\text{Ax}\frac{}{\Delta \vdash_{ACR} (St, l, Sg)}\ (l : St : Sg) \in \Delta$$

$$\wedge\text{-E1}\frac{\Delta \vdash_{ACR} (St \wedge St', G, Sg)}{\Delta \vdash_{ACR} (St, G, Sg)}$$

$$\wedge\text{-E2}\frac{\Delta \vdash_{ACR} (St \wedge St', G, Sg)}{\Delta \vdash_{ACR} (St', G, Sg)}$$

$$\leftarrow\text{-E}\frac{\Delta \vdash_{ACR} (St' \leftarrow St, G, Sg) \quad \Delta \vdash_{ACR} (St, G', Sg')}{\Delta \vdash_{ACR} (St', G \cup G', \mathsf{comb}(Sg, Sg'))}$$

Fig. 1. Argumentation Consequence Relation

Section 4, we show that by taking the basic inference rules of this system and augmenting them we can produce a second normative system which behaves in just the same way as a QPN. This system can then be refined to distinguish probabilities that change to 1 and 0 as well as those that just increase and decrease (Section 5) allowing dominating hypotheses to be detected, and to use order of magnitude (Section 6) and semi-numerical (Section 7) information.

2 Introducing systems of argumentation

In classical logic, an argument is a sequence of inferences leading to a conclusion. If the argument is correct, then the conclusion is true. Consider the simple deductive database [3] Δ_1 which expresses some very familiar information in a Prolog-like notation in which variables are capitalised and ground terms and predicate names start with small letters.

$$\begin{array}{ll} f1 : human(socrates). & \quad \Delta_1 \\ r1 : mortal(X) \leftarrow human(X). & \end{array}$$

The argument $\Delta_1 \vdash mortal(socrates)$ may be correctly made from this database because $mortal(socrates)$ follows from Δ_1 given the usual logical axioms and rules of inference. Thus a correct argument simply yields a conclusion which in this case could be paraphrased '$mortal(socrates)$ is true in the context of $f1$ and $r1$'. In the system of argumentation proposed by Fox and colleagues [13] this traditional form of reasoning is extended to allow arguments to indicate support and doubt in propositions, as well as proving them, by assigning labels to arguments which denote the confidence that the arguments warrant in their conclusions. This form of argumentation may be summarised by the following schema:

$$\text{database } \vdash_{ACR} (\text{Sentence}, \text{Grounds}, \text{Sign})$$

where $\vdash_{ACR}$ is a suitable consequence relation. Informally, Grounds (G) are the facts and rules used to infer Sentence (St), and Sign (Sg) is a number or a symbol drawn from a dictionary of possible numbers or symbols which indicate the confidence warranted in the conclusion.

To formalise this kind of reasoning we start with a language, and we will take $\mathcal{L}$, a set of propositions, including $\perp$, the contradiction. We also have a set of connectives $\{\leftarrow, \neg\}$[4], and the following set of rules for building the well formed formulæ of the language:

- If $l \in \mathcal{L}$ then l is a well formed formula (*wff*).
- If $l \in \mathcal{L}$ then $\neg l$ is a *wff*.
- If $l, m \in \mathcal{L}$ then $l \leftarrow m$, $l \leftarrow \neg m$, $\neg l \leftarrow m$ and $\neg l \leftarrow \neg m$ are *wff*s.
- Nothing else is a *wff*.

The members of $\mathcal{W}$, the set of all *wff*s that may be defined using $\mathcal{L}$, may then be used to build up a database Δ where every item $d \in \Delta$ is a triple $(i : l : s)$ in which i is a token uniquely identifying the database item (for convenience we will use the letter 'i' as an anonymous identifier), l is a wff, and $s \in \{+, -\}$. With this formal system, we can take a database and use the argument consequence relation given in Figure 1 (which is adapted from that in [10] to fit the deductive database context we use here), along with the identity $\neg St \equiv \perp \leftarrow St$ ($\perp$ is logical contradiction) to build arguments for propositions in $\mathcal{L}$ that we are interested in.

Typically we will be able to build several arguments for a given proposition, and so, to find out something about the overall validity of the proposition, we will *flatten* the different arguments to get a single sign.

Together $\mathcal{L}$, the rules for building the formulæ, the connectives, and $\vdash_{ACR}$ define a formal system of argumentation, which, for want of a name we will call $\mathcal{SA}$. In fact, $\mathcal{SA}$ is really the basis of a family of systems of argumentation, because one can define a number of variants of $\mathcal{SA}$ by using different dictionaries of signs. Each dictionary will have its own combination function comb and its own means of flattening arguments, and the meanings of the signs, the flattening function, and the combination function delineate the semantics of the system of argumentation. Thus $\mathcal{SA}$ gives us a general framework for expressing logical facts which can incorporate different models of uncertainty by varying the signs and their associated combination and flattening functions as well as a means of representing default information and of handling inconsistent information [16].

Having introduced $\mathcal{SA}$, the rest of this paper is concerned with different ways of giving a probabilistic interpretation to the signs and combination and flattening functions in order to provide a series of systems of argumentation which are normative in the sense that they accord to the norms of probability theory.

[4] Note that both the set of connectives and the rules for building *wffs* are more restrictive than for other similar systems of argumentation [13].

⊗	+	0	−	?
+	+	0	−	?
0	0	0	0	0
−	−	0	+	?
?	?	0	?	?

⊕	+	0	−	?
+	+	+	?	?
0	+	0	−	?
−	?	−	−	?
?	?	?	?	?

Table 1. The functions ⊗ and ⊕

3 A first normative system

One commonly used system of argumentation within the framework of $\mathcal{SA}$ is one in which the dictionary consists of three symbols, +, − and 0, which represent the notion of an increase, a decrease and no change in belief respectively. When a proposition is labelled with +, it is taken to represent the fact that there is an increase in belief in the proposition, while labelling the rule:

$$mortal(x) \leftarrow human(x)$$

with a + is taken to represent the fact that showing that there is an increase in the belief of something being human causes an increase in belief that it is mortal. The combination function used to combine these signs is ⊗ of Table 1, while the flattening function is one that implements a form of improper linear model with uniform weights and no constant term [4]. This counts the number of +s and − weighted arguments for a proposition, takes the sign that occurs most often and makes that the sign of the proposition, thus taking the sum of all the arguments while giving each argument equal weight.

We will call the system of argumentation which uses this dictionary and pair of functions along with the argument building capabilities of $\mathcal{SA}$ as $\mathcal{NA}_1$. The question that faces us here is how $\mathcal{NA}_1$ may be given a probabilistic semantics. Now, the use of + and − to represent changes in belief suggests a link between this system of argumentation and QPNs [19] since the latter make use of a similar notion. Indeed, it turns out that we can modify the notion of a probabilistic influence in a QPN to give our database facts and rules a probabilistic interpretation. In particular we take triples $(i : l : +)$, where $l \in \mathcal{W}$ and l does not include the connective $\leftarrow$, to denote the fact that $p(l)$ is known to increase, and similar triples $(i : l : -)$, to denote the fact that $p(l)$ is known to decrease. Triples $(i : l : 0)$, clearly denote the fact that $p(l)$ is known to neither increase nor decrease. With this interpretation facts correspond to the nodes in a QPN, and as in QPNs we deal with changes in their probability.

Database rules can similarly be given a probabilistic interpretation by making the triple $(i : m \leftarrow n : +)$, where m and n are members of $\mathcal{W}$ which do not include the connective $\leftarrow$, denote the fact that:

$$p(m \,|\, n, x) \geq p(m \,|\, \neg n, x)$$

where x is any proposition for which there is a triple $(i : m \leftarrow x : s)$ (where s is any sign), while the triple $(i : m \leftarrow p : -)$ denotes the fact that:

$$p(m\,|\,p, x) \leq p(m\,|\,\neg p, x)$$

again for any proposition x for which there is a triple $(i : m \leftarrow x : s)$. We do not make use of triples such as $(i : m \leftarrow p : 0)$ since such rules have no useful effect. As a result a rule $(i : m \leftarrow n : +)$ means that there is a probability distribution over the propositions m and n such that an increase in the probability of n makes m more likely to be true, and a rule $(i : m \leftarrow p : -)$ means that there is a probability distribution over the propositions m and p such that an increase in the probability of p makes m less likely to be true. With this interpretation, rules correspond to qualitative influences in QPNs. It should be noted that the effect of declaring that there is a rule $(i : m \leftarrow n : +)$ is to create considerable constraints on the probability distribution over m and n to the extent that the effect of other rules relating m and n are determined absolutely. That is, a necessary consequence of $(i : m \leftarrow n : +)$ is that we have other rules $(i : \neg m \leftarrow n : -)$, $(i : m \leftarrow \neg n : -)$ and $(i : \neg m \leftarrow \neg n : +)$, and similar restrictions are imposed by rules like $(i : m \leftarrow n : -)$.

With this interpretation of rules and facts, the combination function $\otimes$ has a natural probabilistic interpretation as the function by which changes in probability are combined with probabilistic influences. Indeed $\otimes$ is the function used to combine the two in QPNs. The flattening function also has an obvious probabilistic interpretation in terms of calculating the overall change in probability of a proposition. However, in order for the improper linear model to make sense probabilistically, it is necessary to apply a restriction to the sizes of changes in probability represented by $(i : l : +)$ and $(i : l : -)$. In particular, it requires that all arguments have the same strength, and the simplest situation in which this occurs is that in which all changes in probability have the same magnitude and all rules have the same strength.

As an example of the kind of reasoning that can be performed in $\mathcal{NA}_1$, consider the following simple database Δ_2 of propositional rules and facts. What these rules say is that there are three events that may influence my losing my job—I embezzle funds, I am ill, I am an illegal alien. All of these events have a positive influence on my losing my job, so that if any single one of them on their own becomes more believable, it is more believable that I will lose my job, and, conversely, if they become less believable, it is less believable that I will lose my job.

$f1 : \mathit{embezzle\_funds} : -.$ $\qquad\qquad \Delta_2$
$f2 : \mathit{ill} : +.$
$f3 : \mathit{illegal\_alien} : -.$
$r1 : \mathit{lose\_job} \leftarrow \mathit{embezzle\_funds} : +.$
$r2 : \mathit{lose\_job} \leftarrow \mathit{ill} : +.$
$r3 : \mathit{lose\_job} \leftarrow \mathit{illegal\_alien} : +.$

The database facts say that there is reason to increase belief in that fact that I am ill, and that there are reasons to decrease belief in that fact that I have embezzled funds, and am an illegal alien. From *Delta*$_2$ we can build the arguments:

$$\Delta_2 \vdash_{ACR} (lose\_job, (f1, r1), (-)).$$
$$\Delta_2 \vdash_{ACR} (lose\_job, (f2, r2), (+)).$$
$$\Delta_2 \vdash_{ACR} (lose\_job, (f3, r3), (-)).$$

And the improper linear model will flatten them to come up with the overall conclusion that there is a decrease in belief that I will lose my job after the facts of my situation are known.

4 A second normative system

As stated above, $\mathcal{NA}_1$ is restrictive because its flattening function requires all changes in probability to be of the same magnitude and all rules to have the same strength. To relax this restriction we clearly need a new flattening function. One suitable function is that used by QPNs for combining the effect of several influences on one variable—the QPN "flattening" function. This function is $\oplus$ as specified in Table 1. The use of this function to define a new system of argumentation $\mathcal{NA}_2$ is straightforward after the dictionary of signs is extended to become $\{+, -, 0, ?\}$ where labelling a fact with ? indicates that the change in probability of that fact is unknown, and a rule $(i : m \leftarrow n : ?)$ denotes:

$$p(m \mid n, x) \geq p(m \mid \neg n, x)$$
$$p(m \mid n, y) \leq p(m \mid \neg n, y)$$

for some x and y for which there are triples $(i : m \leftarrow x : s)$ and $(i : m \leftarrow y : s)$, so that if the probability of n increases it is not possible to say how the probability of m will change.

With this interpretation, there is a direct correspondence between a database of formulæ drawn from $\mathcal{W}$ and a qualitative probabilistic network, and it is quite easy to see that any conclusion drawn by $\mathcal{NA}_2$ from a database would also be drawn by the corresponding QPN. The fact that qualitative multiplication distributes over addition ensures that the fact that argumentation builds separate arguments for the same proposition and then flattens them does not mean that it gives a different answer to the equivalent QPN.

To illustrate the difference between $\mathcal{NA}_1$ and $\mathcal{NA}_2$, consider what $\mathcal{NA}_2$ would conclude from Δ_2. Firstly it would build the same arguments as $\mathcal{NA}_1$:

$$\Delta_2 \vdash_{ACR} (lose\_job, (f1, r1), (-)).$$
$$\Delta_2 \vdash_{ACR} (lose\_job, (f2, r2), (+)).$$
$$\Delta_2 \vdash_{ACR} (lose\_job, (f3, r3), (-)).$$

But this time the flattening function would conclude that the overall change in belief in the proposition *lose_job* was ?, indicating that it cannot be accurately identified. This is, of course, probabilistically correct—without information on the relative effects of the various causes of a loss of job, the way in which its probability will change cannot be predicted.

5 A more subtle normative system

Now, in the kind of applications for which $\mathcal{SA}$ was developed [8, 11], it is necessary to represent information of the form "X is known to be true", and "If X is true then Y is true"—information that we might term categorical. It is therefore interesting to investigate if $\mathcal{NA}_2$ can be extended to cover categorical relationships. To do so we first extend the dictionary of signs to be $\{++, +, -, --\}$ where $++$ and $--$ are labels for categorical information. It then turns out that we can give $++$ and $--$ a probabilistic semantics, giving a system of argumentation $\mathcal{NA}_3$ which is $\mathcal{NA}_2$ extended by allowing triples such as $(i : l : ++)$ and $(i : l : --)$ and rules such as $(i : m \leftarrow n : ++)$ and $(i : m \leftarrow n : --)$.

The meaning of $(i : l : ++)$, where l is a *wff* which does not contain $\leftarrow$, is that the probability of l becomes 1, and $(i : l : --)$ means that the probability of l decreases to 0, and to make this clear, we write $(i : l : \overline{\uparrow})$ for $(i : l : ++)$, and $(i : l : \underline{\downarrow})$ for $(i : l : --)$. The meaning of the rules is slightly more complicated. We want a rule $(i : m \leftarrow n : ++)$, where neither m or n contain $\leftarrow$, to denote a constraint on the probability distribution across m and n such that if $p(n)$ becomes 1, so does $p(m)$. This requires that:

$$p(m \,|\, n, x) = 1$$

for all x such that the database contains $(i : m \leftarrow x : s)$ [15]. Similarly, a probabilistic interpretation of a rule $(i : m \leftarrow n : --)$ requires that:

$$p(m \,|\, n, x) = 0$$

for all x such that the database contains $(i : m \leftarrow x : s)$. Once again, the introduction of such rules imposes restrictions on other rules involving the same propositions so that $(i : m \leftarrow n : ++)$ implies that there must be rules $(i : \neg m \leftarrow n : --)$, $(i : m \leftarrow \neg n : --)$ and $(i : \neg m \leftarrow \neg n : ++)$, and similar restrictions are imposed by rules like $(i : m \leftarrow n : --)$. As before, having introduced new

$\otimes_2$	$++$	$+$	0	$-$	$--$	$?$
$\overline{\uparrow}$	$\overline{\uparrow}$	$+$	0	$-$	$\underline{\downarrow}$	$?$
$+$	$+$	$+$	0	$-$	$-$	$?$
0	0	0	0	0	0	0
$-$	$-$	$-$	0	$+$	$+$	$?$
$\underline{\downarrow}$	$-$	$-$	0	$+$	$+$	$?$
$?$	$?$	$?$	0	$?$	$?$	$?$

Table 2. A variant of $\otimes$

$\oplus_2$	$\overline{\uparrow}$	$+$	0	$-$	$\underline{\downarrow}$	$?$
$\overline{\uparrow}$	U	$\overline{\uparrow}$	$\overline{\uparrow}$	$\overline{\uparrow}$	U	$?$
$+$	$\overline{\uparrow}$	$+$	$+$	$?$	$\underline{\downarrow}$	$?$
0	$\overline{\uparrow}$	$+$	0	$-$	$\underline{\downarrow}$	$?$
$-$	$\overline{\uparrow}$	$?$	$-$	$-$	$\underline{\downarrow}$	$?$
$\underline{\downarrow}$	U	$\underline{\downarrow}$	$\underline{\downarrow}$	$\underline{\downarrow}$	U	$?$
$?$	$?$	$?$	$?$	$?$	$?$	$?$

Table 3. A new flattening function

qualitative values and ensured that they have a probabilistic meaning, we have to give a suitably probabilistic means of combining them if we want the whole system to be normative. It is reasonably clear that a suitable comb will be the variant of $\otimes$ given in Table 2. Note the asymmetry in the table.

The correct way to flatten normative arguments, some of which are categorical, is slightly more complex. The problem is that the very strong constraint that a rule $(i : m \leftarrow n : ++)$ puts on the distribution over m and n greatly restricts the values of other rules whose head is m. In fact, if we have $(i : m \leftarrow n : ++)$ then for any other $(i : m \leftarrow x : s)$, $s \in \{+, -\}$ [15]. This means that we have a revised flattening operator $\oplus_2$ as given in Table 3 where the symbol U indicates that the result is not defined. U may also be taken to indicate that if this is the result of flattening, then the database on which its deduction is based violates the laws of probability. Equipping $\mathcal{NA}_3$ with these extensions ensures that it is normative in the sense that all its conclusions will either be in accordance with probability theory or indicate that there has been a violation of the theory.

To see how the system incorporates categorical knowledge, consider the following variation on our example, where information about illegal aliens has been removed, and most information is categorical:

$$\begin{array}{ll} f1 : embezzle\_funds : \overline{\uparrow}. & \Delta_3 \\ f2 : ill : \underline{\downarrow}. & \\ r1 : lose\_job \leftarrow embezzle\_funds : + + . & \\ r2 : lose\_job \leftarrow ill : +. & \end{array}$$

From this using $\mathcal{NA}_3$ we can build the arguments:

$$\Delta_3 \vdash_{ACR} (lose\_job, (f1, r1), (\overline{\uparrow})).$$
$$\Delta_3 \vdash_{ACR} (lose\_job, (f2, r2), (-)).$$

which will flatten to tell us that I will definitely lose my job since the categorical negative effect of embezzling outweighs the positive effect of not being ill. While this might not be a good model of the real world, it does allow us to draw correct probabilistic conclusions from the model.

(A1)	$A \approx A$	(A9)	$A \sim 1 \rightarrow [A] = [+]$
(A2)	$A \approx B \rightarrow B \approx A$	(A10)	$A \ll B \leftrightarrow B \approx (B + A)$
(A3)	$A \approx B, B \approx C \rightarrow A \approx C$	(A11)	$A \ll B, B \sim C \rightarrow A \ll C$
(A4)	$A \sim B \rightarrow B \sim A$	(A12)	$A \approx B, [C] = [A] \rightarrow (A + C) \approx (B + C)$
(A5)	$A \sim B, B \sim C \rightarrow A \sim C$	(A13)	$A \sim B, [C] = [A] \rightarrow (A + C) \sim (B + C)$
(A6)	$A \approx B \rightarrow A \sim B$	(A14)	$A \sim (A + A)$
(A7)	$A \approx B \rightarrow C.A \approx C.B$	(A15)	$A \not\approx B \leftrightarrow (A - B) \sim A$ or $(B - A) \sim B$
(A8)	$A \sim B \rightarrow C.A \sim C.B$		
(P3)	$A \ll B \rightarrow C.A \ll C.B$	(P26)	$A \sim B \rightarrow B \sim A$
(P35)	$A \not\approx B \rightarrow C.A \not\approx C.B$	(P38)	$A \not\approx B, C \approx A, D \approx B \rightarrow C \not\approx D$

Fig. 2. Some of the axioms and properties of ROM[K]

6 Using order of magnitude information

As the example of Δ_3 demonstrated, $\mathcal{NA}_3$ extends the kind of representation and reasoning provided by QPNs by allowing the explicit handling of categorical information. This is not the only extension that is possible. Another is to use some form of order of magnitude reasoning. This would make it possible to say, for instance, that because $p(a)$ increases much more than $p(b)$, and $p(a)$ influences $p(c)$ much more strongly than $p(b)$ influences $p(d)$, it is clear that $p(c)$ will undergo a much larger change in value than $p(d)$. A particularly appropriate system for performing this kind of reasoning, known as ROM[K], is provided by Dague [1]. ROM[K] works by manipulating expressions about the relative size of two quantities Q_1 and Q_2. There are four possible ways of expressing this relation: Q_1 is *negligible with respect to* Q_2, $Q_1 \ll Q_2$, Q_1 is *distant from* Q_2, $Q_1 \not\approx Q_2$, Q_1 is *comparable to* Q_2, $Q_1 \sim Q_2$, and Q_1 is *close to* Q_2, $Q_1 \approx Q_2$. Once the relation between pairs of quantities is specified, it is possible to deduce new relations by applying the axioms and properties of ROM[K], some of which are reproduced in Figure 2.

We can use ROM[K] to define a system of argumentation $\mathcal{NA}_4$ which extends $\mathcal{NA}_2$ with relative order of magnitude reasoning about the size of the changes in probability with which the system deals. As usual, we need to define combination and flattening functions, though here they differ from those of other systems in that they are comparative and additional to those used by $\mathcal{NA}_2$. Once the argument is established as being + or − this new comnbination function gives the relation between the changes based on the strength of the influences that cause the change. Similarly, the new flattening function identifies the greatest influence on a given hypothesis allowing a ? caused by two conflicting arguments to resolved into a + or a −. For the combination function, if the change in $p(a)$ stands in relation rel_1 to the change in $p(b)$ (where rel_1 is one of the relations of ROM[K]) and the strength of the influence of $p(a)$ on $p(c)$ stands in relation rel_2 to the strength of the influence of $p(b)$ on $p(d)$ (rel_2 also being one of the relations of ROM[K]), the relation rel_3 between the changes in $p(c)$ and $p(d)$ is given by Table 4. Thus this table defines the function comb.

		rel$_2$			
	comb	$\approx$	$\sim$	$\not\simeq$	$\ll$
rel$_1$	$\approx$	$\approx$	$\sim$	$\not\simeq$	$\ll$
	$\sim$	$\sim$	$\sim$	V	$\ll$
	$\not\simeq$	$\not\simeq$	V	V	$\ll$
	$\ll$	$\ll$	$\ll$	$\ll$	$\ll$

Table 4. Combining ROM[K] relations.

Note that Table 4 only covers the cases in which the change in $p(a)$ is less than or equal to that in $p(b)$ and the strength of the influence between $p(a)$ and $p(c)$ is less than or equal to that of $p(b)$ on $p(d)$. Obvious permutations of the table will cover the other cases. Also note that the letter V indicates that rel$_3$ may not be determined from the particular values of rel$_1$ and rel$_2$ because to make any prediction would be to step outside the bounds of probability.

For the flattening function, if the change in $p(a)$ stands in relation rel$_1$ to the change in $p(b)$ (where rel$_1$ is one of the relations of ROM[K]) and the strength of the influence of $p(a)$ on $p(c)$ stands in relation rel$_2$ to the strength of the influence of $p(b)$ on $p(c)$ (rel$_2$ also being one of the relations of ROM[K]), the sign of the change in $p(c)$ is given in Table 5 (where $[\Delta p(b)]$ indicates the sign of the change in $p(b)$). Note that Table 5 only covers the cases in which the change in $p(a)$ is less than or equal to that in $p(b)$ and the strength of the influence between $p(a)$ and $p(c)$ is less than or equal to that of $p(b)$ on $p(c)$. Obvious permutations of the table will cover the other cases.

As an example of the kind of reasoning that may be performed in $\mathcal{NA}_4$, consider the following variant on the illness example.

$$
\begin{array}{ll}
f1 : \mathit{ill} : +. & \qquad\qquad \Delta_4 \\
f2 : \mathit{embezzle\_funds} : -. & \\
r1 : \mathit{lose\_job} \leftarrow \mathit{ill} : +. & \\
r2 : \mathit{hospital} \leftarrow \mathit{ill} : +. & \\
r3 : \mathit{lose\_job} \leftarrow \mathit{embezzle\_funds} : +. &
\end{array}
$$

In addition, consider we know that the relationship between the strengths of

		rel$_2$			
		$\approx$	$\sim$	$\not\simeq$	$\ll$
rel$_1$	$\approx$	[?]	[?]	$[\Delta p(b)]$	$[\Delta p(b)]$
	$\sim$	[?]	[?]	[?]	$[\Delta p(b)]$
	$\not\simeq$	$[\Delta p(b)]$	[?]	[?]	$[\Delta p(b)]$
	$\ll$	$[\Delta p(b)]$	$[\Delta p(b)]$	$[\Delta p(b)]$	$[\Delta p(b)]$

Table 5. How to flatten arguments in ROM[K]

$r1$ and $r2$ is $\ll$, while the changes in probability implied by $f1$ and $f2$ stand in relation $\sim$. From the database we can build the arguments:

$$\begin{aligned}
&\Delta_4 \vdash_{ACR} (\mathit{lose\_job}, (f1, r1), (-)).\\
&\Delta_4 \vdash_{ACR} (\mathit{hospital}, (f1, r2), (+)).\\
&\Delta_4 \vdash_{ACR} (\mathit{lose\_job}, (f2, r2), (+)).
\end{aligned}$$

using the old combination function. Considering the first two arguments, the new combination function may then be used to establish which has stronger support. Since both arguments are based upon the same fact, rel_1 is '$\approx$', so that we can conclude that the relation rel_3 between the changes in probability of 'hospital' and *lose_job* must be $\ll$ so that the increase in belief that I will lose my job is much smaller than the increase in belief that I will go to hospital. Similarly, flattening the arguments for *lose_job* with the old flattening function will give ?, while the new flattening function will establish that that the probability of *lose_job* will increase as can be seen by looking at the intersection of $\not\approx$ and $\approx$ in Table 5.

7 Using numerical information

Further precision may be obtained by incorporating numerical information about the size of changes in probability and the strengths of influences. Inspired by Dubois et al. [7], we build a new system of argumentation $\mathcal{NA}_5$ with the same base language as the other systems, but which has a dictionary which includes a set of "linguistic"[5] labels, each of which is an identifer for an interval probability, and may be used to give the strength of rules. A suitable set is:

Strongly Positive	$\geq$	Weakly Positive	$\geq$	Zero	$\geq$	Weakly Negative	$\geq$	Strongly Negative
(SP)		(WP)		(Z)		(WN)		(SN)
$(1, \alpha]$	$\geq$	$[\alpha, 0)$	$\geq$	0	$\geq$	$(0, -\alpha]$	$\geq$	$[-\alpha, 1)$

though we could take any set of intervals we desire—a larger set will give us a finer degree of resolution but be more tedious to use as an example. Note that the open intervals explicitly do not allow the modelling of categorical influences (if these are required we can simply add additional labels at either end of the scale). The dictionary also includes a second set of labels which quantify changes in probability:

[5] The scare quotes denoting that no claim is being made that the probability intervals with which we deal are in any way related to interpretations of natural language—we are just adopting Dubois et al.'s terminology.

comb$_1$	CP	BP	MP	LP	Z
SP	[BP, MP]	[BP, MP]	[MP, LP]	LP	Z
WP	[MP, LP]	[MP, LP]	[MP, LP]	LP	Z
Z	Z	Z	Z	Z	Z

Table 6. Combining "linguistic" labels

$$\begin{array}{ccccccccc} \text{Complete Positive} & \geq & \text{Big Positive} & \geq & \text{Medium Positive} & \geq & \text{Little Positive} & \geq & \text{Zero} \\ (\text{CP}) & & (\text{BP}) & & (\text{MP}) & & (\text{LP}) & & (\text{Z}) \\ & & & & & & & & \\ 1 & \geq & (1, 1-\beta] & \geq & [1-\beta, \beta] & \geq & [\beta, 0) & \geq & 0 \end{array}$$

The definition of the changes Little Negative (LN), Medium Negative (MN), Big Negative (BN) and Complete Negative (CN) are symmetrical, and again we could use a different set if desired.

Like the other systems of argumentation, $\mathcal{NA}_5$ uses the argument consequence relation $\vdash_{ACR}$ to build arguments for hypotheses, and so in order to be able to determine the strength of arguments we must define a combination function comb which says how to combine the "linguistic" labels. To do so we must first choose suitable values of α and β, and on the grounds that we would like our intervals to be evenly sized, we choose $\beta \approx 0.33$ and $\alpha \approx 0.5$. This then gives us the combination function of Table 6 where [MP, LP] stands for the interval whose upper limit is the upper limit of MP and whose lower limit is the lower limit of LP. Results of combining with negative influences and changes can be obtained by symmetry.

To combine several arguments for one proposition we need a suitable flattening function, and this is provided by interval addition. Furthermore, if we are to use the precision of the system we need a way to compare intervals in order to identify which arguments have the greatest support. This may be done using $\geq_{int}$ where $[a, b] \geq_{int} [c, d]$ iff $a \geq c$ and $b \geq d$ [6]. To illustrate the use of $\mathcal{NA}_5$ consider the database:

$$\begin{array}{lr} f1 : embezzle\_funds : CP. & \Delta_5 \\ f2 : ill : BP. & \\ r1 : lose\_job \leftarrow embezzle\_funds : SP. & \\ r2 : lose\_job \leftarrow ill : WP. & \\ r3 : hospital \leftarrow ill : SP. & \end{array}$$

From this we can build the arguments:

$$\begin{array}{l} \Delta_5 \vdash_{ACR} (lose\_job, (f1, r1), [BP, MP]). \\ \Delta_5 \vdash_{ACR} (lose\_job, (f2, r2), LP). \\ \Delta_5 \vdash_{ACR} (hospital, (f2, r3), [BP, MP]). \end{array}$$

The two arguments for *lose_job* may be flattened to give the overall value of [CP, MP] and using $\geq_{int}$ we learn that the increase in probability of *lose_job* is greater than that of *hospital*.

8 Discussion

This paper begain with the claim that it would present a number of normative systems of argumentation, taking this to mean that they have a probabilistic semantics, and that they would thus be an improvement on non-normative systems of argumentation for those cases in which norms are desirable. Furthermore, the claim was made that these systems would also be an improvement on qualitative systems for reasoning with probability such as QPNs since they would allow more precise predictions to be made. In the event five different systems, $\mathcal{NA}_1$-$\mathcal{NA}_5$, which meet these objectives to varying degrees, have been presented.

$\mathcal{NA}_1$, uses a probabilistic notion of qualitative influences between variables to to give meaning to logical rules. The fact that notions of degrees of belief in propositions, and the rules that link propositions, can be given a strict probabilistic semantics means that $\mathcal{NA}_1$ is an extension of non-normative systems of argumentation. However, the restrictions on the meaning of the rules imposed by the improper linear model mean that $\mathcal{NA}_1$ is is not an extension of QPNs. $\mathcal{NA}_2$ is a system of argumentation which is roughly equivalent to QPNs. That is the new information that can be obtained from a set of rules and facts is the same as that which the corresponding QPN could establish. Thus $\mathcal{NA}_2$ is an extension of non-normative systems of argumentation on which it is based. However, once again, it is not an extension of QPNs.

The problem of extending QPNs was addressed by $\mathcal{NA}_3$. The extension takes the form of allowing the representation of categorical influences between variables making it possible to model relations such as "A is known to be true if B is true". Giving these a qualitative representation and a probabilistic meaning makes $\mathcal{NA}_3$ a system which is both normative and can represent and reason with a wider range of information than is possible in a QPN whilst retaining the latter's qualitative nature. Thus it meets overall objectives of the paper. Two further extensions were introduced in the form of $\mathcal{NA}_4$ and $\mathcal{NA}_5$. The first uses order of magnitude reasoning about the size of changes in probability and of the influences between variables to represent and reason with non-categorical data yet be more precise than $\mathcal{NA}_2$. $\mathcal{NA}_5$ achieves the same result by using a set of interval labels to quantify changes and influences.

Finally, it should be noted that the base language of the system of argumentation $\mathcal{SA}$ is more restrictive that that of other similar systems. This is because we exclude formulæ that include the $\wedge$ connective, and complex formulæ such as $(a \leftarrow b) \leftarrow (c \leftarrow d)$. It is currently not clear to what extent these restrictions on formulæ may be lifted, and to what extent they are the price that one must pay to have a normative sematics since it is not obvious how formulæ such as $(a \leftarrow b) \leftarrow (c \leftarrow d)$ can be given any probabilistic meaning.

9 Relation to other work

As ever, it is useful to make some brief remarks about the relation between this work and similar approaches other than those already cited. The close relation between qualitative approaches to probabilistic reasoning in networks and probabilistic systems based on logic was suggested by Wellman [18] while the idea of a database of influences which is equivalent to a probabilistic network has been discussed by, among others, Poole [17] and Wong [20].

The attempt to give an essentially logical system a probabilistic semantics prompts recollection of Goldszmidt's work on normative systems for defeasible reasoning [12]. This clearly has some similarities with our work, but differs in its intent. Goldszmidt aims to build defeasible systems whose behaviour is justified by their probabilistic semantics while we are intent on a more general system. The use of a probabilistic semantics is not our only goal—we are just interested in being able to provide a normative system when one is required, with the choice of alternative combination and flattening functions allowing a broad range of possible systems to be adopted.

In addition, our work has strong connections with that of Darwiche [2], this time differing in the way it is approached. His aim was "...to relax the commitment to numbers while retaining the desirable features of probability theory", which is rather different to the aim of the work described here. We started from the opposite position, taking a completely abstract model of reasoning and seeing how it could be instantiated to behave in a probabilistic way if so desired[6], and the fact that we did so suggests that the work presented here and that in [2] are to some extent complementary.

Acknowledgement

This work was partially supported by Esprit Basic Research Action 6156 DRUMS II (Defeasible Reasoning and Uncertainty Managment Systems). Many thanks to John Fox, Kathy Laskey and Jack Breese for making me think more clearly about some of the issues discussed in this paper.

References

1. Dague, P. (1993) Symbolic reasoning with relative orders of magnitude, *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, Chambery, France.
2. Darwiche, A. (1993) A symbolic generalization of probability theory, PhD. Thesis, Stanford.
3. Das, S. K. (1992) *Deductive databases and logic programming*, Addison-Wesley, Wokingham.

[6] Which often it won't be since probability theory often imposes overly strict constraints for the kind of reasoning that argumentation was designed to provide.

4. Dawes, R. (1979) The robust beauty of improper linear models, *American Psychologist.*
5. Druzdzel, M. J. and Henrion, M. (1993) Efficient reasoning in qualitative probabilistic networks, *Proceedings of the 11th National Conference on Artificial Intelligence*, Washington.
6. Dubois, D. and Prade, H. (1979) Fuzzy real algebra: some results, *Fuzzy sets and systems*, **2**, 327–348.
7. Dubois, D., Prade, H., Godo, L., and Lopez de Mantaras, R. (1992) A symbolic approach to reasoning with linguistic quantifiers, *Proceedings of the 8th Conference on Uncertainty in Artificial Intelligence*, Stanford.
8. Fox, J. (1990) Automating assistance for safety critical decisions, *Philosophical Transactions of the Royal Society*, B, **327**, 555–567.
9. Fox, J., Krause, P. and Ambler, S. (1992) Arguments, contradictions and practical reasoning, *Proceedings of the 10th European Conference on Artificial Intelligence*, Vienna.
10. Fox, J., Parsons, S., Krause, P., and Elvang-Gøransson, M. (1993) A generic framework for uncertain reasoning, in *Qualitative Reasoning and Decision Technologies*, N. Piera Carrete and M. G. Singh eds., CIMNE Press, Barcelona.
11. Glowinski, A., O'Neil, M., and Fox, J. (1987) Design of a generic information system and its application to primary care, *Proceedings of AIME Conference*, Marseille.
12. Goldszmidt, M. (1992) Qualitative probabilities: a normative framework for commonsense reasoning, PhD Thesis, UCLA.
13. Krause, P., Ambler, S., Elvang-Gøransson, M., and Fox, J. (1995) A logic of argumentation for reasoning under uncertainty, *Computational Intelligence*, **11**, 113–131.
14. Loui, R. P. (1987) Defeat among arguments: a system of defeasible inference, *Computational Intelligence*, **3**, 100–106.
15. Parsons, S. (1995) Refining reasoning in qualitative probabilistic networks, *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, Montreal.
16. Parsons, S. and Fox, J. (1994) A general approach to managing imperfect information in deductive databases, *Proceedings of the Workshop on Uncertainty in Databases and Deductive Systems*, Ithaca, NY.
17. Poole, D. (1991) Representing Bayesian networks within probabilistic horn abduction, in *Proceedings of the 7th Conference on Uncertainty in Artificial Intelligence*, Los Angeles, CA.
18. Wellman, M. P. (1994) Some varieties of qualitative probability, *Proceedings of the 5th International Conference on Information Processing and the Management of Uncertainty*, Paris.
19. Wellman, M. P. (1990) *Formulation of tradeoffs in planning under uncertainty*, Pitman, London.
20. Wong, S. K. M., Xiang, Y., and Nie, X. (1994) Representation of bayesian networks as relational databases, *Proceedings of the 5th International Conference on Information Processing and the Management of Uncertainty*, Paris.

This article was processed using the LaTeX macro package with LLNCS style

A Modal Logic with Context-Dependent Inference for Non-Monotonic Reasoning

Philippe Besnard[1] and Yao-Hua Tan[2]

Abstract. Contextual logic CoL is specified through an inference system that takes into account the *context* of reasoning, i.e. all given facts whatsoever. Conclusions are inferred with respect to the full set of premises. Statements are relative to a context, they are expressed by formulas indexed with sets of formulas. A modal extension ECoL is proposed for which examples are given that illustrate how natural it is to apply such a logic to the formalization of non-monotonic reasoning.

1 Introduction

In non-monotonic logics (default logic [Reiter 80] or circumscription [McCarthy 80] among others), extra-logical means only are responsible for non-monotonicity to arise. For example, default logic is based on monotonic reasoning as given by classical logic. If it were not for the so-called default extensions, default logic would be monotonic.

Circumscription captures non-monotonic reasoning via circumscription axioms but the logic applying to these axioms and the logic by which conclusions are derived is again monotonic classical logic. We present contextual logic that enjoys a genuine non-monotonic inference system with no extra-logical part to take care of non-monotonic reasoning. We provide contextual logic with a natural deduction system extended to handle indexed formulas (i.e. formulas with a context) and to include a rule that introduces default assumptions in proofs.

Indeed, the basic idea of reasoning with contextual logic is the following. Contexts are sets of formulas that are added as indices to other formulas. An example of a contextual formula is $q_{\{p,p\rightarrow q\}}$. Roughly speaking, the context specifies the set of formulas with respect to which a conclusion is to be expressed: in $q_{\{p,p\rightarrow q\}}$, the context $\{p, p \rightarrow q\}$ indicates that the case under consideration is whether q is a conclusion when exactly the formulas composing the context are given.

Importantly, a set of premises can have, as a conclusion, q_C but not $q_{C'}$ (where $C \neq C'$). Now, contexts are most useful in admitting *default assumptions*, that are of the form $D\varphi$. For example, in $q_{\{p\rightarrow q,Dp\}}$, the context $\{p \rightarrow q, Dp\}$ refers to q being a conclusion from the formula $\{p \rightarrow q\}$ and the default assumption p. When derived from a set of premises, either of $q_{\{p,p\rightarrow q\}}$ and $q_{\{p\rightarrow q,Dp\}}$ means that q is concluded but not with the same status (the latter is weaker).

The paper is devoted to a few contextual logics. The simplest one, CoL, is introduced first. It is used to illustrate the basic ideas that underly contextual logics. Next, we discuss a modal extension of CoL, the so-called extended contextual logic ECoL, that allows for a more relaxed use of the default operator D, and is more expressive than CoL. The most important difference between CoL and ECoL is that the latter one has the explicit default operator in the object language. A formula $D\varphi$ may not only occur in the context of a formula, as is the case in CoL, but it can also occur in the formula itself. Having the default operator in the object language enhances considerably the expressiveness of ECoL compared to CoL. For instance, the formula $D(p \rightarrow q)_{\{D(p\rightarrow q\}}$ is possible in ECoL, but not in CoL. A very interesting feature of this extra expressiveness of ECoL is that it can be used to represent embedded defaults as, for example, in the formula $D(p \rightarrow D(s \rightarrow t))$. Embedded defaults play an important role in the analysis of common sense reasoning in natural language. It is well-known that embedded defaults cannot be represented in, for example, default logic. So, ECoL is a more expressive non-monotonic logic than default logic.

2 The Syntax of Contextual Logic

We introduce *contextual logic*, abbreviated as CoL, by first giving its language as follows. The syntax, which is propositional for simplicity, is defined in three steps. We first consider a standard propositional language $\mathcal{L}_0$, and we extend $\mathcal{L}_0$ with special formulas called default assumptions. This gives us the language $\mathcal{L}_D$. Finally, we use $\mathcal{L}_D$ to define the syntax of the language $\mathcal{L}$ which consists of propositional formulas indexed with a set of $\mathcal{L}_D$-formulas — the so-called context.

Let $\mathcal{L}_0$ be a standard propositional language. Lowercase letters p, q, r denote atomic propositions. Complex formulas of $\mathcal{L}_0$ are built up in the usual way with the connectives $\neg, \wedge, \vee, \rightarrow$ and $\perp$. The Greek letters of the end of the alphabet φ, ψ, χ will denote well-formed formulas of $\mathcal{L}_0$.

We proceed to define the language $\mathcal{L}_D$ which extends $\mathcal{L}_0$ with formulas called default assumptions (see below). $\mathcal{L}_D$ contains all formulas of $\mathcal{L}_0$, and in addition $\mathcal{L}_D$ contains all formulas of $\mathcal{L}_0$ with a default operator D in front. The well-formed formulas of the language $\mathcal{L}_D$ are defined as follows.

* If φ is a formula of $\mathcal{L}_0$, then φ is an $\mathcal{L}_D$-formula,

[1] IRISA, Campus de Beaulieu, 35042 Rennes cedex, France, besnard@irisa.fr

[2] EURIDIS, Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands, ytan@euridis.fbk.eur.nl

[3] This research has been supported by the CEC under the project DRUMS-2 number 6156 within the ESPRIT-III BRA program.

© 1996 Ph. Besnard and Y.-H. Tan
ECAI 96. *12th European Conference on Artificial Intelligence*
Edited by W. Wahlster
Published in 1996 by John Wiley & Sons, Ltd.

Table 1. Contextual Natural Deduction System

$$(\mathrm{E}\!\to)\ \frac{\varphi_{C_1} \qquad (\varphi\to\psi)_{C_2}}{\psi_{C_1\cup C_2}} \qquad (\mathrm{E}\wedge)\ \frac{(\varphi\wedge\psi)_C}{\varphi_C} \qquad \frac{(\varphi\wedge\psi)_C}{\psi_C} \qquad (\mathrm{E}\vee)\ \frac{(\varphi\vee\psi)_{C_1} \quad \begin{array}{c}\varphi_{\{\}}\\ \vdots\\ \chi_{C_2}\end{array} \quad \begin{array}{c}\psi_{\{\}}\\ \vdots\\ \chi_{C_3}\end{array}}{\chi_{C_1\cup C_2\cup C_3}} \ \text{assumptions [discharged]}$$

$$(\mathrm{I}\!\to)\ \frac{\begin{array}{c}\varphi_{\{\}}\\ \vdots\\ \psi_C\end{array}}{(\varphi\to\psi)_C}\ \text{assumption [discharged]} \qquad (\mathrm{I}\wedge)\ \frac{\varphi_{C_1} \qquad \psi_{C_2}}{(\varphi\wedge\psi)_{C_1\cup C_2}}$$

$$(\mathrm{I}\neg)\ \frac{\begin{array}{c}\varphi_{\{\}}\\ \vdots\\ \bot_C\end{array}}{\neg\varphi_C}\ \text{assumption [discharged]} \qquad (\mathrm{I}\vee)\ \frac{\varphi_C}{(\varphi\vee\psi)_C} \qquad \frac{\psi_C}{(\varphi\vee\psi)_C}$$

$$(\mathrm{E}\neg)\ \frac{\varphi_{C_1} \qquad \neg\varphi_{C_2}}{\bot_{C_1\cup C_2}} \qquad (\mathrm{DNR})\ \frac{\neg\neg\varphi_C}{\varphi_C} \qquad (\mathrm{E}\bot)\ \frac{\bot_C}{\varphi_C}$$

Default Introduction Rule (ID)

$$\frac{}{\varphi_{\{D\varphi\}}}$$

Context Expansion Rule (CE)

$$\frac{\varphi_{C_1}}{\varphi_{C_1\cup C_2}}$$

$*$ If φ is a formula of $\mathcal{L}_0$, then $D\varphi$ is an $\mathcal{L}_D$-formula.

Note that D never occurs in a subformula. So, there is no iteration of default operators: e.g. DDp is not an $\mathcal{L}_D$-formula.

An $\mathcal{L}_D$-formula of the form $D\varphi$ is called a *default assumption.*

The Greek letters of the beginning of the alphabet $\alpha, \beta, \gamma, \delta$ will be used to denote well-formed formulas of $\mathcal{L}_D$.

The language $\mathcal{L}$ consists of propositional formulas indexed with a set of $\mathcal{L}_D$-formulas, by virtue of the single rule:

- If φ is a formula of $\mathcal{L}_0$ and C is a finite set of $\mathcal{L}_D$-formulas, then φ_C is an $\mathcal{L}$-formula.

The set C in φ_C is called the *context* of φ. If C of φ_C is empty, we usually omit the context C. Formulas with the empty context are used as auxiliary *assumptions* in contextual derivations. If a formula has the form $\varphi_{\{\varphi\}}$, then this formula is a *premise.* The difference between assumptions and premises is at the core of contextual logic.

The symbol Σ will be used to denote a set of $\mathcal{L}$-formulas. Most often, it will stand for a set of premises.

We will sometimes need to consider the subset of all default assumptions contained in a context C, written $A(C)$:

$$A(C) = \{\alpha | \alpha \in C \text{ and } \alpha \in \mathcal{L}_D - \mathcal{L}_0\}.$$

In addition, we will sometimes need to consider the "flattened" version of a context C, written $F(C)$, meaning that all occurrences of the default operator in C are deleted:

$$F(C) = \{\varphi | \varphi \in C \text{ and } \varphi \in \mathcal{L}_0\} \cup \{\varphi | D\varphi \in A(C)\}.$$

So, if $C = \{p, p\wedge r \to q, Dr\}$ then $F(C) = \{p, p\wedge r \to q, r\}$.

3 Contextual Natural Deduction

We characterize CoL by a non-monotonic inference operator $\mathrel{|\!\sim}$ as we specify *contextual natural deduction*, an inference system over $\mathcal{L}$-formulas that extends the classical method of natural deduction [Prawitz 65]. We define $\Sigma \mathrel{|\!\sim}^{\Pi} \varphi$, where Σ is a set of premises (i.e. a subset of $\mathcal{L}$), Π is a set of default assumptions and φ_C is a contextual formula (an element of $\mathcal{L}$).

Table 1 gives the inference rules for contextual natural deduction, that define the inference system $\vdash$ underlying CoL.

The Default Introduction Rule (ID) says that at any stage in a proof we can simply introduce φ indexed by the default assumption $\{D\varphi\}$. About the Context Expansion Rule (CE), an overall constraint on contexts guarantees that they are not over-expanded (hence the set Π introduced in the sequel).

Note that the assumptions $\varphi_{\{\}}$ always have an empty context because assumptions (that will be discharged later!) should never introduce new contexts in the reasoning process.

If Σ is a set of premises (hence $\mathcal{L}$-formulas), then $P(\Sigma)$ denotes the set of propositional formulas (that is, from $\mathcal{L}_0$), which are the formulas from Σ with their context removed.

We define the inference operators $\vdash$ and $\mathrel{|\!\sim}$. The first operator $\Sigma \vdash \varphi_C$ denotes that the formula φ_C is derived from the set of premises Σ with the rules of contextual natural deduction in the usual way [Prawitz 65]. Now, $\vdash$ is monotonic but $\mathrel{|\!\sim}$ is non-monotonic. In fact, $\Sigma \mathrel{|\!\sim}^{\Pi} \varphi_C$ is defined as follows.

CoL requirement of maximal context in default conclusions: Let Σ be a set of premises, Π be a set of default assumptions. A formula φ_C is an *acceptable* Π-*default conclusion* from Σ, written $\Sigma \mathrel{|\!\sim}^{\Pi} \varphi_C$, iff

(i) $\Sigma \vdash \varphi_C$,
(ii) $C = P(\Sigma) \cup \Delta$ for some $\Delta \subseteq \Pi$,
(iii) $F(C)$ is consistent wrt $P(\Sigma)$.

By a formula ω consistent with respect to a set of formulas Λ, we mean that $\Lambda \cup \{\omega\}$ must be consistent if Λ is. Of course, extension from ω (in fact, $\{\omega\}$) to the case of a finite set of formulas Ω is unproblematic.

We write $\Sigma \mathrel{|\!\sim} \varphi_C$ instead of $\Sigma \mathrel{|\!\sim}^{\Pi} \varphi_C$ when it does not matter what exact stock of default assumptions Π is given.

Intuitively, Π provides the list of potential default assumptions from which we can freely draw for derivations. We presumably do not use all elements in Π (especially if Π contains contradictory default assumptions $D\varphi$ and $D\neg\varphi$).

If $\Sigma \mathrel{\mid\!\sim}^{\Pi} \varphi_C$, then φ_C is a *final conclusion* from Σ wrt Π. If just $\Sigma \vdash \varphi_C$ then φ_C is an *intermediate conclusion* from Σ.

Note that if a context C is default-free, then the intermediate and final conclusions are the same. This distinction between intermediate and final conclusions might look strange in classical logic, where deduction works as an any-time algorithm, but it is quite natural in common sense reasoning. In fact, the difference between these two types of conclusion is analogous to final verdict and intermediate arguments in legal reasoning. A judge is of course not supposed to draw any conclusions before both parties have made their full arguments.

4 Some Examples of Reasoning with CoL

Example 1 (Bird-Penguin Triangle)
Given the premise set $\Sigma = \{b_{\{b\}}, (b\wedge\neg p \to f)_{\{b\wedge\neg p\to f\}}\}$, the formula $f_{\{b,b\wedge\neg p\to f,D\neg p\}}$ is an acceptable default conclusion.

$$\dfrac{\dfrac{b_{\{b\}} \qquad \dfrac{}{(\neg p)_{\{D\neg p\}}}\,{\scriptstyle (ID)}}{(b\wedge\neg p)_{\{b,D\neg p\}}}\,{\scriptstyle (I\wedge)} \qquad (b\wedge\neg p \to f)_{\{b\wedge\neg p\to f\}}}{f_{\{b,b\wedge\neg p\to f,D\neg p\}}}\,{\scriptstyle (E\to)}$$

So, $\Sigma \mathrel{\mid\!\sim}^{\Pi} f_C$ whenever $A(C)\subseteq\Pi$ (where C is an abbreviation for $\{b, b\wedge\neg p\to f, D\neg p\}$). Also, $C - A(C) = \{b, b\wedge\neg p\to f\} = P(\Sigma)$. Moreover, note that $(f)_{\{b,b\wedge\neg p\to f,D\neg p\}}$ is no longer an acceptable default conclusion if we add either $p_{\{p\}}$ or $(\neg f)_{\{\neg f\}}$ to Σ. But it is still acceptable if we add the unrelated premise $r_{\{r\}}$ to Σ. Also, given the premise set $\Sigma\cup\{p_{\{p\}}\}$ we have that the context of the formula $f_{\{b,b\wedge\neg p\to f,D\neg p\}}$ does not satisfy condition (iii), hence it is not acceptable. And given the premise set $\Sigma\cup\{(\neg f)_{\{\neg f\}}\}$ we have that the context of the formula $f_{\{b,b\wedge\neg p\to f,D\neg p\}}$ does not satisfy condition (iii), hence it is not acceptable either.

Example 2 (Quaker-Republican Diamond)
Given the premise set $\Sigma = \{(q\to p)_{\{q\to p\}}, (r\to\neg p)_{\{r\to\neg p\}}\}$, contextual logic yields:

(a) $\Sigma \mathrel{\mid\!\sim} p_{\{q\to p,r\to\neg p,Dq\}}$
(b) $\Sigma \mathrel{\mid\!\sim} \neg p_{\{q\to p,r\to\neg p,Dr\}}$
(c) $\Sigma \not\mathrel{\mid\!\sim} p_{\{q\to p,r\to\neg p,Dq,Dr\}}$
(d) $\Sigma \not\mathrel{\mid\!\sim} \neg p_{\{q\to p,r\to\neg p,Dq,Dr\}}$

The contexts in the conclusions of (a) and (b) are reminiscent of default extensions in default logic.

In view of the context $\{q\to p, r\to\neg p, Dq\}$ the formula p is true whereas in view of the other context $\{q\to p, r\to\neg p, Dr\}$ the formula $\neg p$ is true. We could say that within the context induced by the default assumption Dq the conclusion p is true, but in the other context induced by the default assumption Dr the conclusion $\neg p$ is true. This is an example of a general property similar to the so-called *orthogonality* of default extensions, i.e. the phenomenon that the union of two default extensions is always inconsistent in default logic.

In example 2, adding two default assumptions does not yield any acceptable default conclusion. More generally, a set of premises Σ such that $P(\Sigma)$ is consistent yields no conclusion whose context involves contradictory default assumptions.

Example 3
Let $\Sigma = \{(p\vee q)_{\{p\vee q\}}\}$ and $\Pi = \{D(p\to r), D(q\to r)\}$. Then, $\Sigma \mathrel{\mid\!\sim} r_{\{p\vee q,D(p\to r),D(q\to r)\}}$

That is, reasoning by cases is captured by contextual logic.

Example 4
Consider $\Sigma = \{(\neg q)_{\{\neg q\}}\}$ and $\Pi = \{D(p\to q)\}$. Then, $\Sigma \mathrel{\mid\!\sim} (\neg p)_{\{\neg q,D(p\to q)\}}$.

Hence, contextual logic also accounts for contraposition.

Example 5
Consider the premise $p_{\{p\}}$ with the default assumptions $D(p\to q)$ and $D(q\to r)$. Now, $p_{\{p\}} \mathrel{\mid\!\sim} r_{\{p,D(p\to q),D(q\to r)\}}$

That is, implication chaining is unproblematic in CoL.

Up to now, we could view CoL as a mere proof system for a variant of default logic similar to *Theorist* [Poole 88]. In the next section, we extend the expressiveness of contextual logic by allowing a more relaxed use of the modal operator D. We thus get the so-called *Extended Contextual Logic ECoL*.

5 Extended Contextual Logic (ECoL)

In CoL, we are a bit restricted in expressing conditional rules "if p then q" that fail to have the strength of the material implication $p\to q$. We have only one way to do it, and that is to make the corresponding formula $p\to q$ a default assumption $D(p\to q)$. Stated otherwise, if we do not want to include $(p\to q)_{\{p\to q\}}$ in Σ then we have only one possibility left, and that is to include $D(p\to q)$ in Π.

The expression $Dp\to q$, if allowed, would still have a different meaning than $p\to q$ being taken into account as a premise or a default assumption. In particular, $D(p\to q)$ is weak enough to have no inconsistency arising when $\Sigma = \{p_{\{p\}}, (\neg q)_{\{\neg q\}}\}$. On the contrary, $Dp\to q$ would yield an inconsistency when $\Sigma = \{p_{\{p\}}, (\neg q)_{\{\neg q\}}\}$. This would be due to having Dp as a consequence of p, which is one of various schemata that should hold in order to deal with the extended syntax. Another theorem would be distributivity of D over implication, that is $D(\varphi\to\psi)\to(D\varphi\to D\psi)$.

An extended contextual logic (ECoL) that allows for explicit D operators in the premises can be defined as follows. First, we adapt the definition of the language in the obvious way. The languages $\mathcal{L}_0$ and $\mathcal{L}_D$ in CoL are combined into one language $\mathcal{L}_{ED}$, which is defined analogously to $\mathcal{L}_0$ except that it has the following extra formation rule:

* If φ is a formula of $\mathcal{L}_{ED}$, then $D\varphi$ is an $\mathcal{L}_{ED}$-formula.

Here is a list of formulas which are in $\mathcal{L}_{ED}$ but not in $\mathcal{L}_D$:

(1) $\varphi\wedge D\psi\to\chi$ (2) $D\varphi\to D\psi$
(3) $\varphi\to D\psi$ (4) $\varphi\to\neg D\psi$

Formula (1) expresses that only $D\psi$ can be assumed by default, and not for example φ.

Formula (2) can be used to express dependencies between default assumptions. For example, $Df\to Dw$ can be used to express that if one assumes by default that something can fly, one could as well assume by default that it has wings.

Formulas (3)–(4) express dependencies of default assumptions on factual information. For example, the formula $p\to\neg Df$ might be used to express that the fact of being a penguin blocks the use of the default of flying. Hence, this type of formulas is important to express the so-called principle of specificity among default rules.

Similarly to $\mathcal{L}$ in CoL, the language $\mathcal{L}_E$ in ECoL is defined by the single clause:

- If φ is an $\mathcal{L}_{ED}$-formula and C a finite set of $\mathcal{L}_{ED}$-formulas, then φ_C is an $\mathcal{L}_E$-formula.

The following rule governs the behaviour of D in ECoL.

Modal Rule (MR)

$$\dfrac{\begin{array}{cccl}[D\varphi_{\{\}}] & \dots & [D\psi_{\{\}}] & \\ [\varphi_{\{\}}] & \dots & [\psi_{\{\}}] & \text{if any: assumptions [discharged]}\\ & \vdots & & \\ & \chi_C & & \end{array}}{D\chi_C}$$

When χ depends on no assumption, the modal rule applies as well (although no default assumption is introduced).

Application of the modal rule is as follows. Given a proof from $\varphi, \dots, \psi$ to χ, we can discharge any of $\varphi, \dots, \psi$ and replace them by the corresponding assumptions of the form $D\varphi, \dots, D\psi$, also introducing $D\chi$ as the conclusion.

Note that, although this inference rule can be "upward growing", it can be so only with default assumptions.

Presumably, the most illustrative inference and proof is that of φ from $D\varphi$ and $\neg D\bot$ (the contexts have been arbitrarily chosen so that the reader can also see how contexts propagate in the course of a proof where the modal rule is applied):

$$\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{l}(D\varphi)_{\{\}}\\ \dfrac{{}^{(1)}\varphi_{\{\}} \qquad {}^{(2)}(\neg\varphi)_{\{\}}}{\bot_{\{\}}}\ (E\neg)\end{array}}{(D\bot)_{\{\}}}\ (1)\ (MR) \qquad (\neg D\bot)_{\{\neg D\bot\}}}{\bot_{\{\neg D\bot\}}}\ (E\neg)}{(\neg\neg\varphi)_{\{\neg D\bot\}}}\ (2)\ (I\neg)}{\varphi_{\{\neg D\bot\}}}\ (DNR)$$

The proof starts with applying the modal rule to the deduction of $\bot_{\{\}}$ from the two assumptions $\varphi_{\{\}}$ and $(\neg\varphi)_{\{\}}$. The modal rule is applied in such a way that $(\neg\varphi)_{\{\}}$ is left as an assumption (remember that discharging any assumption is *optional*) whereas $\varphi_{\{\}}$ is discharged and replaced by $(D\varphi)_{\{\}}$. Also, the conclusion $\bot_{\{\}}$ gives rise to the new conclusion $(D\bot)_{\{\}}$.

The fact that φ is discharged when applying the modal rule is indicated by (1) next to the (MR) bar and by (1) as a superscript in front of φ. Another discharge of an assumption is indicated by (2) when applying the I¬ rule.

Certainly, the simplest proof involving the modal rule is: (omitting the empty context in order to improve readibility)

$$\dfrac{\dfrac{{}^{(1)}\varphi}{D\varphi}\ (MR)}{\varphi \to D\varphi}\ (1)\ (I\to)$$

Here, we apply the rule for the case where no assumption is taken into account, i.e. the only assumption φ is not discharged. It is later discharged when the I→ rule is applied.

More generally, the modal rule admits all degenerate cases (including discharge with respect to irrelevant assumptions) as happens with other rules in natural deduction. The modal rule behaves similarly to the rules defined by [Prawitz 65] for S4 and S5. In particular, the modal rule induces the same phenomenon that Prawitz described for his modal rules, namely the existence of non-trivial proofs with maximal formulas (a formula is maximal when it results from applying an introduction rule and is subject to the corresponding elimination rule). This topic will not be discussed further here because it is not central to the matter of non-monotonic reasoning.

Some formulas of particular interest that have a proof using the modal rule are:

T1. $D(\varphi \to \psi) \to (D\varphi \to D\psi)$
T2. $\varphi \to D\varphi$
T3. $D(\varphi \wedge \psi) \leftrightarrow (D\varphi \wedge D\psi)$
T4. $(D\varphi \vee D\psi) \to D(\varphi \vee \psi)$
T5. $DD\varphi \leftrightarrow D\varphi$
T6. $\neg D\bot \to (\varphi \leftrightarrow D\varphi)$

The above list of formulas is now significant enough: We can say a few words about semantics. The D operator is interpreted in Kripke models as follows. Let $M = \langle W, R\rangle$ be a Kripke model with W a set of worlds and R a binary accessibility relation on these worlds. A formula $D\varphi$ is true at a world w in M if φ is true in all worlds w' that are accessible from w. The crucial point is that $\neg D\bot \to (\varphi \leftrightarrow D\varphi)$ is a theorem. (Note, however, that $\neg D\bot$ itself is not a theorem!) This clearly indicates that, in a Kripke model, if there exists at all a world which is accessible from the actual world, then it can only be the actual world itself. Accordingly, the model theory for the logic axiomatized by the modal rule is defined by the class of all Kripke models in which the accessibility relation is at most reflexive: $\forall w, w' \in W$, if wRw' then $w = w'$.

Returning to proof-theoretic considerations, we replace the default introduction rule of CoL by the following rule.

Extended Default Introduction Rule (EID)

$$\dfrac{}{D\varphi_{\{D_\varphi\}}}$$

All the other inference rules of contextual natural deduction remain unchanged. In ECoL, we again make a distinction between two inference operators $\vdash_E$ and $\mathrel{|\!\!\sim}_E$. The first operator $\Sigma \vdash_E \varphi_C$ denotes that the formula φ_C is derived from the set of premises Σ with the rules of contextual natural deduction (including the modal rule) in the usual way, and the default introduction rule ID is replaced by the extended default introduction rule EID. The operator $\vdash_E$ is monotonic. The second operator $\Sigma \mathrel{|\!\!\sim}^{\Pi}_{E} \varphi_C$ is non-monotonic, and is again defined in terms of the first operator:

ECoL requirement of maximal context in default conclusions:
Let Σ be a set of $\mathcal{L}_E$-formulas and Π be a set of $\mathcal{L}_{ED}$-formulas. Let $\vdash^*_E$ indicate provability without the EID rule. Then, $\Sigma \mathrel{|\!\!\sim}^{\Pi}_{E} \varphi_C$ iff

(i) $\Sigma \vdash_E \varphi_C$,
(ii) $C = P(\Sigma) \cup \Delta$ for some $\Delta \subseteq \Pi$,
(iii) if $P(\Sigma) \nvdash^*_E D\bot$ then $P(\Sigma) \cup C \nvdash^*_E D\bot$.

That is, we use essentially the same definition of Requirement of Maximal Contexts for $\mathrel{|\!\!\sim}^{\Pi}_{E}$ as we did for $\mathrel{|\!\!\sim}^{\Pi}$ in CoL. Of course, $P(\Sigma)$ is redefined in the obvious way: If Σ is a set of $\mathcal{L}_E$-formulas, then $P(\Sigma)$ denotes the set of $\mathcal{L}_{ED}$-formulas which are the formulas from Σ with their context removed.

A more significant difference is that

$F(C)$ is consistent wrt $P(\Sigma)$

which can be written

if $P(\Sigma) \nvdash \bot$ then $P(\Sigma) \cup F(C) \nvdash \bot$

is actually a special case of the new modal constraint (iii). Indeed, $\mathcal{L}$-formulas (i.e. the ones that are dealt with in CoL), are such that, clearly, $P(\Sigma) \vdash^*_E \bot$ iff $P(\Sigma) \vdash^*_E D\bot$. Moreover, $F(C)$ would be adapted so that we delete not only single occurrences of the D operator in front of formulas in the context C, but we delete also sequences of D operators. In fact, we consider that $\varphi \leftrightarrow D\varphi$ holds and to apply it, we simply need to postulate $\neg D\bot$ (cf above). Accordingly, $P(\Sigma) \cup F(C) \vdash \bot$ would become $P(\Sigma) \cup C \cup \{\neg D\bot\} \vdash^*_E \bot$ and then, $P(\Sigma) \cup C \vdash^*_E D\bot$.

Example 6
For $\Sigma = \{b_{\{b\}}, (b \wedge D\neg p \rightarrow f)_{\{b \wedge D\neg p \rightarrow f\}}\}$, ECoL yields:

(a) $\Sigma \mid\!\sim_E f_{\{b, b\wedge D\neg p \rightarrow f, D\neg p\}}$

(b) $\Sigma, p_{\{p\}} \not\mid\!\sim_E f_{\{b, b\wedge D\neg p \rightarrow f, D\neg p\}}$

(c) $\Sigma, (\neg D\neg p)_{\{\neg D\neg p\}} \not\mid\!\sim_E f_{\{b, b\wedge D\neg p \rightarrow f, D\neg p\}}$

A feature of ECoL is that $\neg(D\varphi \wedge D\neg\varphi)$ is not provable in the underlying logic. Allowing for $D\neg\varphi \wedge D\varphi$ to be consistent is harmless, because even if $D\varphi$ and $D\neg\varphi$ are true at the same time, the requirement of maximal contexts prevents that both default assumptions are used in the same context. Hence, the entailment definition of ECoL filters away, so to say, the unintuitive usages of $D\varphi \wedge D\neg\varphi$.

In addition, not having $\neg(D\varphi \wedge D\neg\varphi)$ as a theorem has the consequence that not only $D\varphi \wedge D\neg\varphi$ but also $\neg\varphi \wedge D\varphi$ is consistent. As just discussed, the consistency of $\neg\varphi \wedge D\varphi$ is harmless.

There is also a technical reason not to have $\neg(D\varphi \wedge D\neg\varphi)$ as a theorem of the underlying logic: If this formula were to be a theorem like T1–T6, then $\varphi \rightarrow \psi$ would be equivalent with $D\varphi \rightarrow D\psi$, i.e. $(\varphi \rightarrow \psi) \leftrightarrow (D\varphi \rightarrow D\psi)$ would hold. This would mean that the D operator collapses in the case of implications. The left to right direction of this equivalence comes from T1 and T2. The right to left direction can be shown as follows. Assuming φ, then T2 gives $D\varphi$. Now, $D\psi$ follows due to $D\varphi \rightarrow D\psi$. If we accept $\neg(D\varphi \wedge D\neg\varphi)$ as a theorem, then $D\psi$ implies $\neg D\neg\psi$, and by T2, we get ψ.

Another formula that seems intuitive as a theorem for the D operator is $\neg D\varphi \leftrightarrow D\neg\varphi$. However, should we accept it, then we get the collapse $\varphi \leftrightarrow D\varphi$. So, $D\varphi$ could always be replaced simply by the formula φ, which means that the D operator would be useless. The left to right direction follows immediately from T2. The right to left direction follows from T2 and $\neg D\varphi \leftrightarrow D\neg\varphi$. Here are the details. By virtue of T2, $\neg\varphi \rightarrow D\neg\varphi$ holds. Then, $\neg D\varphi \leftrightarrow D\neg\varphi$ yields $\neg\varphi \rightarrow \neg D\varphi$, hence $D\varphi \rightarrow \varphi$.

A further advantage of ECoL is that it allows for embedded defaults. The need for embedded defaults is argued for in [Morreau 95]. An example of an embedded default is: "Usually, people tell the truth most of the time if they speak". Hence, if somebody is not telling the truth this person can be exceptional in two ways. Either, because he is a pathological liar, or in this particular case he is misled (but tells the truth otherwise). The first case is an exception of the "usually" default, whereas the second case is an exception of the "most of the time" default. This embedded default can be represented in ECoL as $D(p \rightarrow D(s \rightarrow t))$, where p stands for being a person, s for speaking and t for telling the truth.

Example 7
Given $\Sigma = \{D(p \rightarrow D(s \rightarrow t))\}$ and that p and s are contained in the default set Π, we can derive t as a default conclusion, i.e.

$\Sigma \mid\!\sim_E t_{\{D(p\rightarrow D(s\rightarrow t)), Dp, Ds\}}$.

Note that, due to T1 and T5, $D(p \rightarrow D(s \rightarrow t))$ implies $Dp \rightarrow (Ds \rightarrow Dt)$. Furthermore, Dt implies t, due to T6. This conclusion is no longer derivable if we add as a premise $\neg t$, or $\neg p$ or $\neg s$, i.e.

$\Sigma, (\neg t)_{\{\neg t\}} \not\mid\!\sim_E t_{\{D(p\rightarrow D(s\rightarrow t)), \neg t, Dp, Ds\}}$,

$\Sigma, (\neg p)_{\{\neg p\}} \not\mid\!\sim_E t_{\{D(p\rightarrow D(s\rightarrow t)), \neg p, Dp, Ds\}}$,

$\Sigma, (\neg s)_{\{\neg s\}} \not\mid\!\sim_E t_{\{D(p\rightarrow D(s\rightarrow t)), \neg s, Dp, Ds\}}$.

In Morreau's non-monotonic logic based on conditional logic, this embedded default $D(p \rightarrow D(s \rightarrow t))$ would be $(p > (s > t))$, where $\varphi > \psi$, is read as "usually, if φ then ψ". Morreau also argues that default logic is inappropriate for representing embedded defaults. The way we obtain non-monotonicity in ECoL is quite different from the way Morreau gets it in his logic, but we agree with his observation that having an explicit modal operator in the object-language is probably the most appropriate way to represent embedded defaults.

6 Related Work

Work related to (E)CoL includes non-monotonic logics (most notably default logic), but also work about contexts [McCarthy 93] [Nait Abdallah 95], work about deductive dependencies such as ATMS [de Kleer 86], LDS [Gabbay 96] and proof systems close to contextual natural deduction [Batens 91] [Gabbay & Hunter 93].

A crucial difference between CoL and ATMS [de Kleer 86] is that ATMS has only one kind of hypotheses. Hence, ATMS is not a logic, not even a proof system, as it only does bookkeeping: it just records deductive dependencies. Contextual natural deduction has two kinds of hypotheses: auxiliary assumptions and premises, the latter corresponding to the ATMS hypotheses. In reasoning, handling assumptions is essential (discharging assumptions, ...). Auxiliary assumptions (formulas with the empty context) are the reason why contextual natural deduction does more than bookkeeping, it actually defines a logic.

Plainly, contextual natural deduction is a labelled deduction system. It is a matter of notation and focus (LDS [Gabbay 96] is a conceptual framework of which contextual natural deduction is a real instance).

Contextual logic bears some similarity to dynamic dialectical logics [Batens 91] in the way it requires a global use of premises. Nevertheless, contextual natural deduction seems more economical than the proof systems defined for dynamic dialectical logics that refer to maximal "extensions" of subproofs in order to validate a proof.

Contextual natural deduction looks like proof systems that restrict the access for rules to formulas [Gabbay & Hunter 93]. The mechanism for propagating labels/contexts is the same for CoL and restricted access logics. CoL is less constrained because it does not prevent rules to apply to any formulas. Also, it is more expressive as there is nothing like default assumptions in restricted access logics.

On this, the ionic logic approach [Nait Abdallah 95] suffers from no limitations. Up to the point that it is problematic: Many axioms are introduced to govern the behaviour of the special starred expressions (roughly, default assumptions). As a result, the notion of an inference in the ionic logic approach is much more involved than it is in CoL.

CoL only makes use of contexts. This is a major difference with the work on contexts undertaken by [McCarthy 93] where contexts are central to the enterprise. For instance, we do not express links between contexts. On the positive side, by sticking to a simple account of contexts, we were able to give a rather satisfactory inference system.

We now compare (E)CoL and default logic. First, the language of default logic does not allows us to distinguish between valid conclusions and conclusions drawn using default rules. In contrast, such a distinction is made explicit in contexts by CoL. Also, in default logic, there is no representation for the default assumptions that generate a default extension (the so-called "generating defaults"). A fortiori, there is no representation of the possible dependency of a derived formula on specific generating defaults. All corresponding possibilities are given by contexts, as should be clear by now to the reader.

Finally, no incremental approach to computing default extensions is possible because the space of all deductions from the premises must be explored for a conclusion to be formally derived by default. Such a burden on default logic does not extend to contextual logic which is the first nonmonotonic logic with an incremental inference system: Any derivation has a value on its own, that value being given by its context (in particular, a conclusion with an inconsistent context is worthless).

ECoL is comparable to cumulative default logic [Brewka 91] in the sense that in the latter a record is kept of all the justifications of the defaults that are used to generate an extension and this record is checked for consistency. Similarly, the requirement (iii) of maximal contexts in ECoL is such that if $P(\Sigma) \not\vdash^*_E D\bot$ then $P(\Sigma) \cup C \not\vdash^*_E D\bot$, so that, in view of (ii), the latter part means $C \not\vdash^*_E D\bot$. That is, the context should be consistent. So, ECoL can be viewed as a modal formulation of cumulative default logic, but ECoL yields more default conclusions. For instance, if we have the formula $p \wedge Dq \rightarrow r$ and we apply the default assumption Dq, then q is a conclusion in ECoL. However, if we consider the corresponding default $p : q/r$, then the justification q is not necessarily in the extension generated by this default. Another difference is that cumulative default logic is less expressive than ECoL as it inherits the expressiveness drawbacks of default logic, including the difficulty to deal with embedded defaults and case analysis (from p, infer q by default, and from r, infer s by default, should lead to the default conclusion $q \vee s$ from $p \vee r$).

ACKNOWLEDGEMENTS

The authors are grateful to Fabio Massacci and Leon van der Torre for helpful discussions.

REFERENCES

Batens D. [1991]. Dynamic Dialectical Logics, in *Paraconsistent Logic*, Norman, Priest, Routley (eds), Philosophia Verlag.

Brewka G. [1991]. Cumulative Default Logic, *Artificial Intelligence* 50, pp. 183–205.

Gabbay D. [1996]. *Labelled Deductive Systems*, to appear.

Gabbay D., Hunter A. [1993]. Restricted Access Logics, ECSQARU-93, LNCS 747, Granada, Spain, pp. 137–144.

de Kleer J. [1986]. An Assumption-based Truth Maintenance System, *Artificial Intelligence* 28, pp. 127–162.

McCarthy J. [1980]. Circumscription — A Form of Non-Monotonic Reasoning, *Artificial Intelligence* 13, pp. 27–39.

McCarthy J. [1993]. Notes on Formalizing Contexts, IJCAI-93, Chambéry, France, pp. 555–560.

Morreau M. [1995]. Allowed Arguments, IJCAI-95, Montréal, Canada, pp. 14466–1472.

Nait Abdallah A. [1995]. *The Logic of Partial Information*, Springer.

Poole D. [1988]. A Logical Framework for Default Reasoning, *Artificial Intelligence* 36, pp. 27–47.

Prawitz D. [1965]. *Natural Deduction*, Almqvist and Wiksell.

Reiter R. [1980]. A Logic for Default Reasoning, *Artificial Intelligence* 13, pp. 81–132.

Semantic Based Theory Revision in Nonmonotonic Logic

Cees Witteveen witt@cs.{ruu,tudelft}.nl[*]

Wiebe van der Hoek wiebe@cs.ruu.nl[†]

Abstract

By using a nonmonotonic semantics one tries to extract more information from a theory than would be possible by classical means. Such a nonmonotonic semantics is called informative if it satisfies both supraclassicality (the nonmonotonic models are a subset of the set of classical models) and consistency preservation (nonmonotonic models exist whenever the theory is consistent). Most nonmonotonic semantics, however, satisfy supraclassicality but lack consistency preservation. In such cases we propose to apply theory revision in order to construct an informative (revised) semantics.

We present some postulates for such nonmonotonic theory revision and we will show that, unlike in classical theory revision, nonmonotonic theories have to be *expanded* instead of contracted in order to give them a satisfactory meaning. Finally, we state some conditions on the nonmonotonic semantics to be satisfied in order revise theories successfully.

1 Introduction and Motivation

One of the primary advantages nonmonotonic reasoning should have above classical reasoning is to allow one to draw stronger conclusions than can be obtained by classical means, i.e. the semantics should be *more informative* than the classical semantics. *More informative* here means that

- nonmonotonic reasoning, in general, should be *stronger* than classical reasoning. That is, every conclusion obtained by classical means should also be obtainable

[*] Delft University of Technology, Department of Mathematics and Computer Science, P.O.Box 356, 2600 AJ Delft, The Netherlands and Utrecht University, Department of Computer Science, Padualaan 14, 3584 CH Utrecht, The Netherlands.

[†] Utrecht University, Department of Computer Science, Padualaan 14, 3584 CH Utrecht, The Netherlands.

by nonmonotonic reasoning, i.e., nonmonotonic reasoning should be *supraclassical*;

- on the other hand, nonmonotonic reasoning should not collapse, if it is still possible to draw conclusions by classical means. That is, the (set of) conclusions obtained by nonmonotonic reasoning should be as least as *informative* as the conclusions obtained by classical reasoning.

  So nonmonotonic reasoning should not run into inconsistencies whenever the theory is classically consistent. This principle is also known as *consistency preservation*.

Comparing nonmonotonic reasoning with classical reasoning applied to the same theory T implies that we distinguish a nonmonotonic interpretation of T and a classical interpretation of T.

1.1 Classical and non-classical readings of a theory

It may not be immediately clear what we mean by the classical reading and the nonmonotonic models of a theory T. In a forthcoming paper ([14]) we will give a general treatment of nonmonotonic vss. first-order interpretations of a given theory. Here, we give some examples.

In our view, a nonmonotonic semantics gives rise to the selection of certain acceptable or preferred models of the theory instead of considering the total class of ordinary (classical) models of the theory.

This idea is most prominently present in the preferential model semantics developed by Shoham [12] and further analysed by Makinson [5, 6] and Kraus, Lehmann and Magidor [4]. Given a preference relation between models of a first order theory T, instead of taking into account every model of a first-order theory, only the most preferred models are chosen as the (nonmonotonic) models of T. Circumscription is a special case of such preferential semantics.

There are also approaches in which the model selection is guided by special interpretations of classical connectives, such as, for example, in logic programming. Here, the implication and negation connectives can be given a special meaning in order to select the nonmonotonic models of the theory. Since in these semantics every nonmonotonic model also is a classical model, i.e. respects the standard meaning of these connectives, these semantics again are supraclassical. Thus, we propose to use the ordinary meaning of the connectives to obtain the classical meaning of theories T that assign a non-standard interpretation to some of the connectives.

Finally, there is a class of nonmonotonic approaches, where a classical language is *extended* by the introduction of new symbols that have a special meaning. These symbols constitute the syntactical guides that may help us in finding the acceptable models among the set of all models. Here the problem to distinguish between a classical and non-classical reading is more involved.

In such cases we propose to translate these special symbols or formulas containing these special symbols back into first-order formulas. For example, in a default theory $\Delta = (W, D)$ over a first-order language $\mathcal{L}$, we translate every default rule

$$\delta = \frac{\alpha \, ; \; M\beta_1, \ldots, M\beta_n}{\gamma}$$

into a first-order formula

$$\delta_{f.o} = \alpha \wedge \beta_1 \wedge \ldots \wedge \beta_n \rightarrow \gamma$$

Then the classical reading of Δ is given by the theory

$$\Delta_{f.o} = W \cup \{\delta_{f.o} \mid \delta \in D\}$$

It is not difficult to see that every model of an extension E of Δ also is a classical model of $\Delta_{f.o}$.

Likewise, if T is an auto-epistemic theory, its classical interpretation is given by the models of the theory $T_{f.o}$ derived from T by substituting (recursively) every formula of the form $L\phi$ by the formula ϕ. The nonmonotonic models of T are the models of the objective part of the auto-epistemic extensions of T. Again, by the conditions that every auto-epistemic extension E of T has to satisfy, it can be seen that every model of E is also a classical model of $T_{f.o}$.

1.2 The lack of consistency-preservation

While almost all nonmonotonic logics are supraclassical (cf. [6]), most of them do not satisfy consistency preservation. For example, the following well-known formalisms: *default logic* (with the exception of normal default logic), *auto-epistemic logic*, *nonmonotonic semantics of logic programming* and *preferential entailment semantics* all lack consistency preservation.

We consider this an unfortunate state of affairs. In particular, we consider the principle of consistency preservation as extremely useful in such applications as debugging and diagnosis. Here, we use nonmonotonic reasoning to draw conclusions about the expected behaviour of a system if everything goes well, i.e. the (normality) assumptions are not violated. But we would also like to draw conclusions in exceptional cases, where the normality assumptions do not hold and 'ordinary' nonmonotonic reasoning fails.

The reason for this failure is that in most nonmonotonic semantics, especially those used in logic programming, there is no possibility to recover from a conflict, if some assumptions made in reasoning nonmonotonically turn out to be responsible for violating a constraint.

Let us give a motivating example.

Example 1.1
Suppose you are at the airport knowing that

1. you are treated as a class A passenger iff you receive a special pass-through card,
2. everyone receiving a special pass-through card has direct access to the gates via a special port.
3. but, if there is no reason to receive a special card then you will pass through the normal port and
4. if you have access through the normal port and it can be assumed that you are not a class A passenger then you will be checked.
5. Every VIP is treated as a class A passenger and finally,
6. it happens that you are not checked.

The following program describes this situation:

$$\begin{array}{rrcl}
P: & \mathit{class\_A\_passenger} & \leftarrow & \mathit{receive\_special\_card}. \\
 & \mathit{receive\_special\_card} & \leftarrow & \mathit{class\_A\_passenger}. \\
 & \mathit{direct\_access\_by\_special\_port} & \leftarrow & \mathit{receive\_special\_card}. \\
 & \mathit{access\_by\_normal\_port} & \leftarrow & \neg \mathit{receive\_special\_card}. \\
 & \mathit{checked} & \leftarrow & \mathit{access\_by\_normal\_port}, \neg \mathit{class\_A\_passenger}. \\
 & \mathit{class\_\_A\_passenger} & \leftarrow & \mathit{VIP}. \\
 & \perp & \leftarrow & \mathit{checked}.
\end{array}$$

Note that P is classically consistent: it has four classical models where the first two models are

$$M_1 = \{cAp, rsc, dasp, \neg anp, \neg ch, \neg VIP\}$$

and

$$M_2 = \{cAp, rsc, dasp, \neg anp, \neg ch, VIP\},$$

while M_3 and M_4 are obtained by making anp true in M_1 and M_2, respectively.
If you would use the stable model semantics as your intended semantics, however, P is nonmonotonically inconsistent: $\mathit{Stable}(P) = \emptyset$. The reason why, should be clear: there is no reason to assume that you should receive a special card, but this assumption is directly responsible for the violation of the constraint 6.
So the program is not classically inconsistent, but the problem is that none of the classical models is a stable model of the program. This means that stability as a criterion to select acceptable models from the set of classical models fails and that we have to select other models.

The problem is, which models we choose. Clearly, if you add the fact

$$receive\_special\_card \leftarrow$$

or the fact

$$VIP \leftarrow$$

to P, both M_1 and M_2 will occur as a stable model of an expanded version of P. But it is also possible to add the rule

$$class\_A\_passenger \leftarrow \neg access\_by\_normal\_port$$

to P and obtain M_1 as a stable model of the resulting expansion.
Note that in the first and the last case, we add some information that is classically derivable from the program. ■

This example suggests that consistency preservation may be obtained by changing our program P to a related theory P', and to use the intended semantics of P' to give P a suitable meaning. That is, we may find intended models of the original theory by applying *theory revision.*
Basically, this is the idea applied in the dominant classical AGM theory revision framework (see [2]). Here, a classically inconsistent theory T is transformed into a classical consistent theory T' and the models of T' are used to give a meaning to T.
We will generalize this idea to a *classically consistent, but nonmonotonically inconsistent* theory T. We will apply theory revision to T and transform T to another theory T' that does have intended models. Then we use the intended models of T' as the intended models for T. In this way we will construct a semantics that is consistency-preserving.

The problem then is how to characterize *suitable* theory transformations. We will formulate some fairly simple postulates for nonmonotonic theory revision and then we will show that, unlike classical theory revision, nonmonotonic theory revision has to be performed by *expanding* the original theory instead of contracting it.

2 Restoring consistency preservation by theory revision

As stated in the introduction, we would like to have a nonmonotonic logic to satisfy the principles of supraclassicality and consistency preservation.
To state these principles in a more precise way, we will assume that, given a not necessarily closed theory T specified in some first-order language $\mathcal{L}$, our nonmonotonic logic is characterized by a set $Sem(T)$ of *nonmonotonic* models for T. We will denote the set of *classical* models of T by $Mod(T)$. Then both principles can be formulated as follows:

1. **Supra**(classicality):
   For every theory T, $Sem(T) \subseteq Mod(T)$;

2. **Cons**(istency preservation):
   For every theory T, $Sem(T) \neq \emptyset$, whenever $Mod(T) \neq \emptyset$.

It has been observed that almost every nonmonotonic semantics satisfies **Supra**, while few satisfy **Cons** ([6]). So let us assume that we have a theory T and a semantics Sem such that **Supra** is satisfied, but **Cons** is not. We will look for a semantics Sem^* revising Sem that satisfies both principles.

The problem we are confronted with closely resembles the problem of *theory revision* for classical theories: there we are forced to revise our interpretation of a theory T if T turns out to be *classically inconsistent*, i.e. $Mod(T) = \emptyset$, while here we have to revise our nonmonotonic interpretation of the theory if $Sem(T) = \emptyset$, while still $Mod(T)$ may be nonempty.
In the well-known AGM-approach to theory revision (cf. [2]), revision is accomplished by *theory transformation*: the current inconsistent theory T is replaced by a transformation $R(T)$ of T and the (classical) meaning of $R(T)$ is used to provide a suitable meaning for T.
Since we are aiming at restoring consistency preservation, we will not deal with the problem what to do if $Sem(T) = Mod(T) = \emptyset$, but we will also apply this idea of theory revision by theory transformation. That is, if $Sem(T) = \emptyset$, we propose to derive the proposed meaning $Sem^*(T)$ of T by

(i) transforming T to some theory $T' = R(T)$,

(ii) applying the original semantics Sem to T', and

(iii) requiring that $Sem^*(T) = Sem(R(T))$.

Since we want to deal with nonmonotonic revision in classically consistent theories, we will assume that there is some class $\mathcal{T}$of classically consistent theories, i.e. for every $T \in \mathcal{T}$, $Mod(T) \neq \emptyset$. We call a pair $(\mathcal{T}, Sem)$ a *nonmonotonic semantics*. The nonmonotonic semantics we want to have are *informative* semantics:

Definition 2.1 (Informative semantics)
A nonmonotonic semantics $(\mathcal{T}, Sem)$ *is called* informative *if it satisfies both* **Supra** *and* **Cons.**

A theory transformation R is a computable mapping from $\mathcal{T}$ to $\mathcal{T}$. We would like to know which transformations are suitable and which are not. In order to give such a characterization, we will present some postulates for the triple $(\mathcal{T}, Sem, R)$, called a *revision framework*, where $(\mathcal{T}, Sem)$ is a nonmonotonic semantics and R is a theory-transformation $R : \mathcal{T} \rightarrow \mathcal{T}$. Furthermore, we assume that Sem is supraclassical with respect to $\mathcal{T}$.

We propose to use the following simple postulates:

P1. $Sem(R(T)) \neq \emptyset$ whenever $Mod(T) \neq \emptyset$.
This postulate specifies that revision should be *successful*: we should find a transformation R such that $Sem(T)$ exists if T is consistent.

P2. $R(T) = T$, whenever $Sem(T) \neq \emptyset$.
We should be careful in extending our semantics: only in those cases in which Sem does not provide a meaning for T, it is allowed to change T.

P3. $Mod(R(T)) = Mod(T)$.
This postulate stipulates that theory transformation should be classically neutral: we should not change the class of models from which the subset of models has to be chosen that we (nonmonotonically) prefer.

Note that the revised semantics Sem^* based on Sem is meant to satisfy both **Supra** and **Cons**.

Definition 2.2 (Successful revision frameworks)
A revision framework $(\mathcal{T}, Sem, R)$ *is called* successful *if*

1. *it satisfies the postulates P1-P3 and*
2. *the revised nonmonotonic semantics* $(\mathcal{T}, Sem^*)$*, where* $Sem^*(T) = Sem(R(T))$ *is informative.*

Let us first check that indeed, if $(\mathcal{T}, Sem, R)$ satisfies the postulates **P1-3** and **Supra,** then Sem^* will satisfy both **Supra** and **Cons.**

Observation 2.3 (the postulates guarantee success)
If

1. $(\mathcal{T}, Sem, R)$ *satisfies P1-P3 and*
2. Sem *satisfies* **Supra**

then $(\mathcal{T}, Sem^*)$ *is informative.*

PROOF By P1, it immediately follows that Sem^* satisfies **Cons**.
Since Sem satisfies **Supra**, we have $Sem^*(T) = Sem(R(T)) \subseteq Mod(R(T))$. Hence, by P3, $Sem^*(T) \subseteq Mod(T)$. So Sem^* satisfies **Supra**, too. ∎

Given these postulates P1-P3, we would like to investigate the following problems:

1. What can we say about the nature of revision functions for nonmonotonic theories and how do they compare to revision functions used in classical theory revision?
2. What will be needed for *minimal* revision?
3. Which conditions have to be satisfied by a semantics $(\mathcal{T}, Sem)$ in order to find a successful revision framework $(\mathcal{T}, Sem, R)$?

We will try to formulate some general answers to this question.

3 What revision functions should be used

In the standard (AGM-inspired) theory revision literature ([2]), *retraction* is the only appropriate theory revision operator to give a suitable meaning to an inconsistent theory: From the current inconsistent theory T some parts are retracted and the (consistent) remaining part T' is used to give T its meaning.
It turns out that revision by retraction, at least for a large part of nonmonotonic semantics, is *not* suitable[1].

3.1 Reasonable semantics: Weak Confirmation of Evidence

Given that a nonmonotonic semantics $(\mathcal{T}, Sem)$ obeys the principle of supra-classicality, what should be reasonable to expect from it? Let us define the following consequence operator $\mid\!\sim_{Sem}$:

$$T \mid\!\sim_{Sem} x \quad \text{iff} \quad \exists M \in Sem(T) \text{ s.t. } M \models x$$

So $T \mid\!\sim_{Sem} x$ holds iff according to an acceptable model M of T, x is true. Then the least thing we might expect is that there is still some acceptable model for the theory

[1] In fact, we are not able to come up with a reasonable nonmonotonic semantics for which it can be proven that retraction is an option in nonmonotonic theory revision.

T' obtained from T and x, that is, nonmonotonic reasoning should not collapse just by adding a (brave) nonmonotonic consequence of the theory.
Slightly generalizing, we introduce the following *Weak Confirmation of Evidence* (abbreviated by **WCE**)[2] principle:

Definition 3.1 (WCE)
For every $T \in \mathcal{T}$ and $\Phi \subseteq WFF(\mathcal{L})$, if $T \mathrel{|\!\sim}_{Sem} \Phi$, then there exists some $\Psi \subseteq WFF(\mathcal{L})$, such that $T + \Phi \mathrel{|\!\sim}_{Sem} \Psi$.

As far as we know, this principle holds for every nonmonotonic semantics currently known.

Remark. Note that this principle can be seen as a weakening of a brave form of *cautious monotony*:

$$T \mathrel{|\!\sim}_{Sem} x, T \mathrel{|\!\sim}_{Sem} y \text{ implies } T + x \mathrel{|\!\sim}_{Sem} y$$

■

The following proposition is an almost direct consequence of the definition of $\mathrel{|\!\sim}_{Sem}$ and is useful in proving properties of WCE:

Proposition 3.2 *$T \mathrel{|\!\sim}_{Sem} \Phi$ iff $Sem(T) \cap Mod(\Phi) \neq \emptyset$.*

PROOF By definition $T \mathrel{|\!\sim}_{Sem} \Phi$ iff there is an $M \in Sem(T)$ such that $M \models \Phi$ iff there is an $M \in Sem(T)$ such that $M \in Mod(\Phi)$ iff $M \in Sem(T) \cap Mod(\Phi)$. ■

It is interesting to note that **WCE** is satisfied by every informative semantics:

Proposition 3.3
If Sem is an informative semantics, then Sem satisfies **WCE**.

PROOF Suppose that $T \mathrel{|\!\sim}_{Sem} \Phi$ for some set Φ. Since **Supra** is satisfied, this implies that there is a model $M \in Mod(T)$ satisfying Φ. Therefore, $Mod(T) \cap Mod(\Phi) = Mod(T+\Phi) \neq \emptyset$. Then by **Cons**, it follows that $Sem(T+\Phi) \neq \emptyset$. Hence, there exists a Ψ such that $T + \Phi \mathrel{|\!\sim}_{Sem} \Psi$ and, therefore, **WCE** is satisfied. ■

Since **Cons** is not implied by **Supra+WCE**, so **WCE** is a weaker property than **Cons** in the presence of **Supra**.

[2] This principle is a weaker variant of the Confirmation of Evidence principle, introduced by Reiter. We are grateful to Wiktor Marek for giving the reference.

3.2 Retraction is not suitable for revision

Note that in the AGM approach, whenever a theory is classically inconsistent we have to apply theory contraction to give it a meaning. If the theory is consistent, we can leave the theory unchanged. So let us define the following notion of a *pure* retraction function:

Definition 3.4 (Pure retraction)
Let $\mathcal{T}$ be a class of theories and $R : \mathcal{T} \rightarrow \mathcal{T}$ be a theory-transformation.
Then R is a pure *retraction function iff $\forall T \in \mathcal{T}.\ R(T) \subseteq T$ and there exists some $T \in \mathcal{T}$ such that $R(T) \neq T$.*

It might be that pure retraction functions are not useful for nonmonotonic theory revision, while for some (but not all) theories a mixture of adding some information and retraction is more appropriate. Therefore, we will allow for such cases and define the following notion of *weak* retraction functions:

Definition 3.5 (Weak retraction)
Let $\mathcal{T}$ be a class of theories and $R : \mathcal{T} \rightarrow \mathcal{T}$ be a theory-transformation.
Then R is a weak *retraction function iff $\exists T \in \mathcal{T}.\ R(T) \subset T$ and $T \neq R(T)$.*

Note that the class of pure expansion functions is contained in the class of weak expansion functions.

We will now prove that even weak retraction is not a suitable option for nonmonotonic theory-revision if the postulates stated above are satisfied and the nonmonotonic semantics satisfies **WCE** and **Supra**.

Theorem 3.6
Let $(\mathcal{T}, Sem)$ be a nonmonotonic semantics satisfying **Supra+WCE**, *but not* **Cons**. *Then the revision framework $(\mathcal{T}, Sem, R)$ cannot be successful if R is a weak retraction function.*

PROOF Suppose, on the contrary, that R is a weak retraction function in the successful framework $(\mathcal{T}, Sem, R)$, where $(\mathcal{T}, Sem)$ satisfies **Supra+WCE**.

Since R is a weak contraction function, there is a theory $T \in \mathcal{T}$ such that $R(T) = T' \subset T$ and $T \neq T'$, while $Mod(T) \neq \emptyset$.

Since the revision framework is assumed to be successful, the postulates P1-P3 have to be satisfied. By P2, we have

$$Sem(T) = \emptyset$$

and by P1, it follows that $Sem(R(T)) \neq \emptyset$. Hence,

$$Sem(T') = Sem(R(T)) \neq \emptyset.$$

So, let M be an arbitrary model in $Sem(T')$.

By P3 and **Supra**, we have

$$\emptyset \subset Sem(T') = Sem(R(T)) \subseteq Mod(T).$$

Hence, $Sem(T') \cap Mod(T) \neq \emptyset$, so by Proposition 3.2

$$T' \mathrel{|\!\!\sim}_{Sem} T$$

and therefore, by **WCE**,

$$T + T' \mathrel{|\!\!\sim}_{Sem} \Psi$$

for some Ψ. But then, since $T' \subset T$, it follows that

$$Sem(T) = Sem(T + T') \neq \emptyset,$$

contradicting the fact that $Sem(T) = \emptyset$. Therefore, R cannot be a weak retraction function if the framework is successful. ■

Since pure and weak retraction functions now can be excluded, we can take a look at *mixed transformations* and *pure expansion functions*.

Definition 3.7 (Pure expansion)
A pure expansion *function is a theory transformation R such that for all $T \in \mathcal{T}$, $R(T) \supseteq T$.*

Definition 3.8 (Mixed transformation)
We call a revision function R a mixed *revision function if for some $T \in \mathcal{T}$, $R(T) - T \neq \emptyset$ and $T - R(T) \neq \emptyset$.*

It turns out that the class of mixed transformations is obsolete in the following sense: we can show that the class of mixed functions can be represented by the class of pure expansion functions, i.e. for every mixed revision function R satisfying the postulates, there exists a pure expansion function R' such that $R'(T) = R(T) + T$ and R' also satisfies the postulates.

Lemma 3.9
Let $(\mathcal{T}, Sem, R)$ be a successful revision framework, where Sem satisfies **Supra** *and* **WCE** *(but not* **Cons***) and R is a mixed transformation.*
Then the revision framework $(\mathcal{T}, Sem, R')$, where R' is a pure expansion function defined as[3] $R'(T) = R(T) + T$, is also successful.

PROOF Let the pure expansion function R' be defined as $R'(T) = R(T) + T$. We have to prove that $(\mathcal{T}, Sem, R')$ satisfies the postulates P1-P3.

[3] Note that $R'(T)$ and $R(T)$ are classically equivalent, i.e. $Mod(R'(T)) = Mod(R(T))$

Since $R(T) = T$ whenever $Sem(T) \neq \emptyset$, it follows immediately that R' satisfies Postulate P2.
Since

$$Mod(R'(T)) = Mod(R(T) + T) = Mod(R(T)) \cap Mod(T)$$

and, by P3,

$$Mod(R(T)) = Mod(T),$$

it follows that

$$Mod(R'(T)) = Mod(T),$$

hence, R' satisfies Postulate P3.

Finally, we have to show that R' is successful, i.e. P1 is satisfied.
Assume that $Mod(T) \neq \emptyset$. Since postulate P1 is satisfied for R, $Sem(R(T)) \neq \emptyset$. So take a model $M \in Sem(R(T))$.
By **Supra**, it follows that $M \in Mod(R(T))$ and since R satisfies P3, $M \in Mod(T)$. Hence,

$$M \in Sem(R(T)) \cap Mod(T).$$

But then, by Proposition 3.2 and **WCE** it follows that

$$Sem(R(T) + T) = Sem(R'(T)) \neq \emptyset,$$

so P1 is satisfied for R'. ∎

3.3 Minimal revision and pure expansion

While in the previous section we showed that revision functions can be represented by pure expansion functions, in this section we will show that in order to perform minimal revision, only pure expansion functions should be applied.

Definition 3.10 (Minimal revision)
We say that $(\mathcal{T}, Sem, R)$ *is a* minimal *revision system if R satisfies the postulates P1-P3 and the following minimality postulate:*

> *P4 For every $R' \neq R$ satisfying the postulates P1-P3, if $(R(T) \ominus T) \subseteq (R'(T) \ominus T)$ then $R(T) = R'(T)$.*
>
> *Here, $\ominus$ is the symmetrical set-difference operator.*

This postulate expresses that successful revisions should minimize the additions to and retractions from the original theory.
As an almost direct consequence of the preceding lemma we have:

Theorem 3.11
If $(\mathcal{T}, Sem, R)$ is a minimal revision system satisfying the postulates P1-P4, then R is a pure expansion function.

Example 3.12
Let $T = \{Lp\}$ be an auto-epistemic theory. This theory is classically consistent, having a classical model $M = \{p\}$. T, however, does not have an auto-epistemic extension. Consider the revision $T' = R(T) = T + \{p\}$. It satisfies the postulates and now M is the model of the objective part of T''s only auto-epistemic extension.

4 Successful revision frameworks

In the preceding sections we assumed that the nonmonotonic semantics $(\mathcal{T}, Sem)$ satisfies **Supra** and **WCE** and then we proved some properties of the theory transformation R and the resulting revised semantics.
Let us now turn to the other side and let us assume that we have a successful revision framework $(\mathcal{T}, Sem, R)$. Then we would like to know which properties we could derive for Sem and R to hold.
Our first result states that indeed supraclassicality is a necessary condition for a nonmonotonic semantics in order for the framework to be applicable:

Proposition 4.1
If $(\mathcal{T}, Sem, R)$ is a successful revision framework, $(\mathcal{T}, Sem)$ must satisfy **Supra**.

Proof Let $T \in \mathcal{T}$. We prove that $Sem(T) \subseteq Mod(T)$. If $Sem(T) = \emptyset$ we are done, so assume $Sem(T) \neq \emptyset$. Then, by P2, $R(T) = T$, hence $Sem(T) = Sem(R(T))$. By P3, it follows that $Sem(T) = Sem(R(T) \subseteq Mod(T)$. ■

Without **WCE**, we have a simple necessary and sufficient condition for a successful revision framework. Essentially, it states that in every subclass of classically-equivalent theories, there exists at least one theory T such that $Sem(T) \neq \emptyset$.

Lemma 4.2
Let $(\mathcal{T}, Sem)$ satisfy **Supra**. *Then there exists a successful framework $(\mathcal{T}, Sem, R)$ iff for every $T \in \mathcal{T}$ there is a $T' \in \mathcal{T}$ such that $Mod(T) = Mod(T')$ and $Sem(T') \neq \emptyset$.*

Proof Trivial, by the definition of successful revision frameworks. ■

If, however, we add **WCE**, we have a stronger result:

Lemma 4.3
Let $(\mathcal{T}, Sem)$ *satisfy* **Supra + WCE.**
Then there exists a successful framework $(\mathcal{T}, Sem, R)$ *iff for every* $T \in \mathcal{T}$, $Sem(Cn(T)) \neq \emptyset$.

PROOF Assume that Sem satisfies **Supra + WCE** and that there exists a successful framework $(\mathcal{T}, Sem, R)$ for some R. So, take an arbitrary $T \in \mathcal{T}$. Then $Sem(R(T)) \neq \emptyset$. We have to prove that $Sem(Cn(T)) \neq \emptyset$. Note that, by P3, $Mod(R(T)) = Mod(T)$. Since $Mod(T) = Mod(Cn(T)$, it follows that

$$Sem(R(T)) \cap Mod(Cn(T)) \neq \emptyset,$$

so by Proposition 3.2 and **WCE** it follows that $R(T) + Cn(T) \mid\!\sim_{Sem} \Phi$, for some Φ. Since $Mod(R(T)) = Mod(Cn(T))$, it follows that $R(T) \subseteq Cn(T)$. Hence, by **WCE**, $Sem(R(T) \cup Cn(T)) = Sem(Cn(T)) \neq \emptyset$.
Conversely, let Sem satisfy **Supra + WCE** and assume that for every $T \in \mathcal{T}$, $Sem(Cn(T)) \neq \emptyset$. Then define $R(T)$ as $R(T) = T$ if $Sem(T) \neq \emptyset$ and $R(T) = Cn(T)$ otherwise. It is not difficult to see that $(\mathcal{T}, Sem, R)$ is a successful revision framework. ∎

Sometimes we have a semantics Sem that is not consistency-preserving, but we can use another semantics Sem' that is consistency-preserving, if, by syntactical manipulation of (parts of) a theory, Sem can be reduced to Sem':

Definition 4.4 (Reducibility)
Let Sem, Sem' *be two semantics for a class* $\mathcal{T}$ *of theories. We say that* Sem *is* classically reducible *to* Sem' *iff for all* $T \in \mathcal{T}$*, there is a theory* $T' \in T$ *such that* $T \models T'$ *and* $Sem(T') = Sem'(T')$.

Theorem 4.5
If $\mathcal{T}$ *is class of theories,* Sem *is classically reducible to* Sem' *in* $\mathcal{T}$ *and* Sem' *satisfies* **Supra** *and* **Cons**, *then there is a successful revision framework* $(\mathcal{T}, Sem, R)$ *for* $(\mathcal{T}, Sem)$.

PROOF For every $T \in \mathcal{T}$, define $R(T)$ as follows: $R(T) = T$ if $Sem(T) \neq \emptyset$ and $R(T) = T + T'$ else.
We show that in $(\mathcal{T}, Sem, R)$, the postulates P1-P3 are satisfied.
Let $Mod(T) \neq \emptyset$. Let T' be such that $Sem(T') = Sem'(T')$. Since T is consistent, T' is also consistent. Since Sem' is consistency-preserving, it follows that $Sem'(T') = Sem(T') \neq \emptyset$. Let $M \in Sem(T') \subseteq Mod(T') \subseteq Mod(T)$. Then, by **WCE**, it follows that $Sem(T + T') = Sem(R(T)) \neq \emptyset$. So P1 is satisfied.
Postulate P2 is satisfied by construction of R. Finally, P3 is satisfied, since

$$Mod(R(T)) = Mod(T + T') = Mod(T) \cap Mod(T') = Mod(T).$$

■

An example of such a class of theories is the class of normal logic programs with constraints, where for each program P always an equivalent program P' can be found such that $MinMod(P') = Stable(P')$, where $MinMod$ is the minimal model semantics and *Stable* the stable model semantics.

5 Discussion

We have presented some postulates for revision of nonmonotonic theories and we have shown that given some fairly weak conditions on the nonmonotonic semantics, revision of such theories should be accomplished by expansion instead of contraction.
At first sight, the idea of revision by retraction might be strange. It can be explained as follows. In nonmonotonic reasoning we reason by making assumptions concerning the truth or falsehood of certain statements and derive conclusions from them. However, as soon as we detect some violation of constraints or are not able to derive any conclusion from a theory, we realize that we must have assumed to much: some of these assumptions may not be compatible. Now the only way to get out, is to state explicitly that one or more assumptions should not be made, i.e. to expand the original theory.
The idea of revision by expansion in nonmonotonic theory revision has been discussed before. In Auto-Epistemic Logic (AEL) for example, Morris ([8]) has suggested something like theory expansion for auto-epistemic theories that do not have an AE-extension. The simple idea is: if there is no AE-extension for a set of premises S, then a set-inclusion minimal set of ordinary (i.e. modal-operator-free) premises is added to S such that an AE-extension exists.
In logic programming, the work of Pereira et al. ([9]) on Contradiction Removal Semantics can be seen as a special expansion method, allowing for revision of assumptions.

In truth maintenance, belief revision has been performed by a pure expansion technique, called *dependency-directed backtracking (ddb)* ([1, 10, 11]). As these methods mainly have been stated informally and in a procedural way, there were little or no formal results. Recently, in [13], we have shown that ddb is not suitable for the stable model semantics and only can be complete if the semantics is as weak as a positivistic or supported model semantics.

Recently, Inoue and Sakama in [3] proposed a very general approach to revision of nonmonotonic theories by proposing to revise a theory T by a minimal set of additions I and removals O such that $T + I - O$ has an acceptable model. Our results show that in most cases, when T is classically consistent, removal of formulas in the form of retraction is not necessary.

References

[1] J. Doyle, A Truth Maintenance System, *Artificial Intelligence*, vol. 12, pp. 231–272, 1979.

[2] P. Gärdenfors, *Knowledge in Flux*, MIT Press, Cambridge, MA, 1988.

[3] K. Inoue, C. Sakama, Abductive Framework for Nonmonotonic Theory Change. *Proceedings IJCAI'95*, 1995.

[4] S. Kraus, D. Lehmann and M. Magidor, Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence*, vol 44, pp. 167–207, 1990.

[5] D. Makinson, General Theory of Cumulative Inference, in: M. Reinfrank et al.(eds) *Non-Monotonic Reasoning*, vol. 346, Lecture Notes on Artificial Intelligence, pp. 1–18, Springer Verlag, Berlin, 1989.

[6] D. Makinson, General Patterns in Nonmonotonic Reasoning, in: D.M. Gabbay, C.J. Hogger, J.A. Robinson (eds) *Handbook of Logic in Artificial Intelligence and Logic Programming*, Vol. 3, Nonmonotonic Reasoning, Oxford Science Publications, Oxford, 1994.

[7] V. Marek and M. Truszczyński, *Nonmonotonic Logic*, Springer Verlag, Heidelberg, 1993.

[8] Morris, P., Stable Closures, Defeasible Logic and Contradiction Tolerant Reasoning, *Proceedings of the 7th National Conference on Artificial Intelligence*, 1988.

[9] L. M. Pereira, J. J. Alferes and J. N. Aparicio. Contradiction Removal within well-founded semantics. In: A. Nerode, W. Marek and V. S. Subrahmanian, (eds.), *First International Workshop on Logic Programming and Non-monotonic Reasoning*, MIT Press, 1991.

[10] C. J. Petrie, Revised Dependency-Directed Backtracking for Default Reasoning, *Proc. AAAI*, 1987.

[11] M. Reinfrank, Fundamentals and Logical Foundations of Truth Maintenance, Linköping Studies in Science and Technology. Dissertations no. 221, Linköping University, 1989.

[12] Y. Shoham. A Semantical approach to non-monotonic logics. in: *Proceedings of the Tenth Joint International Conference on Artificial Intelligence* (IJCAI), 1987.

[13] C. Witteveen and W. van der Hoek, Revision by Communication In V. Marek, A. Nerode and M. Truszczyński, editors, *Logic Programming and Non-Monotonic Reasoning, LNAI 928*, pp. 189–202, Springer Verlag, 1995.

[14] C. Witteveen and W. van der Hoek, Classical interpretations of nonmonotonic theories (to appear), 1996

In: *Theoretical Aspects of Rationality and Knowledge (TARK)*, 1996

Counterfactuals and updates as inverse modalities

(Preliminary version)

Mark Ryan
School of Computer Science
University of Birmingham
Birmingham B15 2TT, UK.
mdr@cs.bham.ac.uk
http://www.cs.bham.ac.uk/~mdr

Pierre-Yves Schobbens
Institut d'Informatique
Facultés Universitaires de Namur
Rue Grandgagnage 21
5000 Namur, Belgium
pys@info.fundp.ac.be
http://www.info.fundp.ac.be/~pys

Odinaldo Rodrigues
Department of Computing
Imperial College
London, SW7 2BZ, UK.
otr@doc.ic.ac.uk
http://theory.doc.ic.ac.uk/~otr

Abstract

We point out a simple but hitherto ignored link between the theory of updates and counterfactuals and classical modal logic: update is a classical existential modality, counterfactual is a classical universal modality, and the link between the two (called the Ramsey rule) is simply the link between two inverse accessibility relations of a classical Kripke model.

1 Introduction

Background. An intuitive connection between theory change and counterfactuals was observed by F. P. Ramsey [17], who proposed what has become known as the Ramsey Rule:

> To find out whether the counterfactual 'if A were true, then B would be true' is satisfied in a state S, change the state S minimally to include A, and test whether B is satisfied in the resulting state.[1]

It was initially hoped that the AGM theory of belief revision [3, 13] would provide the right notion of minimal change. However, the intuitively acceptable AGM postulates for belief revision are known to be incompatible with the Ramsey Rule [2, 3].

It turns out that the theory of updates proposed by Katsuno and Mendelzon [10] is compatible with the Ramsey Rule [6]. Updates, like revisions, are a formalisation of theory change; but whereas revisions are intended to model changing knowledge about a fixed world, updates are intended to model a changing world. The difference between the formalisations of updates and revisions can be seen in terms of postulates; for example, the AGM postulate

$$A * B = A \wedge B \quad \text{if } A \wedge B \text{ is consistent}$$

[1] Actually, Ramsey proposed the rule only for non-counterfactual conditionals, but the term 'Ramsey Rule' is now taken to refer to counterfactuals too.

is accepted for revisions, but rejected for updates. The difference can also be seen in terms of operations on models; in revision, we measure the distance to the models of the old theory as a whole, while in update we measure the distance to them pointwise. Justifications of these differences (and further details and examples) can be found in [11, 6].

Our contribution. In this paper, we show that the standard treatments of updates (eg. [10]) and conditionals [18, 12, 14] are systems of multi-modal logic, whose Kripke accessibility relations are inverses of each other. This is the semantic equivalent of the Ramsey rule. For many of the standard postulates for updates and counterfactuals, we work out the correspondence property of the accessibility relation. This enables us to translate between postulates for counterfactuals and postulates for updates. In this way, we use Ramsey's Rule to translate between theories of update and theories of counterfactuals.

Structure. The paper is arranged as follows. Section 2 contains some preliminaries. In section 3, we show that updates and conditionals are systems of multi-modal logic, and that they have inverse accessibility relations. In section 4, we show that this is equivalent to the Ramsey rule, and in section 5 we translate the standard axioms for update into conditional axioms, and vice versa. Conclusions are in section 6.

2 Preliminaries

2.1 Multi-modal logic

We assume a propositional language L with finitely[2] many atomic propositions $p, q, r, \ldots$ and connectives $\wedge, \vee, \neg, \rightarrow, \leftrightarrow, \Box, \Diamond$. The connectives $\Box$ and $\Diamond$ take two arguments; if A, B are formulas then so are $\Box_A B$ and $\Diamond_A B$. The set L is the set of atomic formulas $p, q, r, \ldots$; the set $\mathbf{L}$ is the set of all formulas over L.

The semantics of multi-modal logic is given as follows (cf. [5, 15, 16, 8, 7]). A *model* $M = \langle W, R, V \rangle$ of the multi-modal language L is a set W of *worlds*, an *accessibility relation* $R \subseteq \mathcal{P}(W) \times W \times W$ and a *valuation* $V : L \to \mathcal{P}(W)$. The ternary relation R may also be thought of as an $\mathcal{P}(W)$-indexed family $\{R_S \mid S \subseteq W\}$ of binary relations in $W \times W$.

The relation $\Vdash$ of *satisfaction* between a model $M = \langle W, R, V \rangle$, a world $x \in W$ and a formula A is defined inductively on A as follows.

$$\begin{array}{rll} x \Vdash_M p & \text{iff} & x \in V(p) \\ x \Vdash_M \neg A & \text{iff} & x \nVdash_M A \\ x \Vdash_M A \wedge B & \text{iff} & x \Vdash_M A \text{ and } x \Vdash_M B \\ x \Vdash_M \Box_A B & \text{iff} & \text{for each } y \in W,\ R_{|A|}(x, y) \text{ implies } y \Vdash_M B \\ x \Vdash_M \Diamond_A B & \text{iff} & \text{there is a } y \in W \text{ such that } R_{|A|}(x, y) \text{ and } y \Vdash_M B \end{array}$$

The missing connectives $\vee$, $\rightarrow$, $\leftrightarrow$ are defined by similar (standard) clauses.

In the context of a model M, $|A|$ is defined to be $\{x \in W \mid x \Vdash_M A\}$. Note that $|A \wedge C| = |A| \cap |C|$. We will use this fact in some proofs. The subscript on $\Vdash_M$ will usually be dropped in order to make the notation lighter.

The model M satisfies the formula A, written $M \Vdash A$, if $x \Vdash_M A$ for each $x \in W$. A *frame* $F = \langle W, R \rangle$ consists of a set of worlds and an accessibility relation. Such a frame

[2] The restriction that the number of propositional atoms be finite is imposed by Katsuno and Mendelzon, whose results we use.

F satisfies A, written $F \Vdash A$, if for each valuation V, we have $\langle W, R, V \rangle \Vdash A$. A formula A is *valid*, written $\models A$, if it is satisfied by every frame. A formula A is *satisfiable* in a model M if $|A| \neq \emptyset$. If $A_1, A_2, \dots, A_n, B$ are formulas, the rule

$$\frac{A_1 \quad A_2 \quad \dots \quad A_n}{B}$$

means: if each of the A_i is valid, then B is valid. Notice that this is rather weaker than asserting the axiom $A_1 \wedge \dots \wedge A_n \to B$. The double-barred rule

$$\begin{array}{c} A \\ \hline\hline B \end{array}$$

means: A is valid iff B is valid.

2.2 Inverse modalities

Classical modal logic is based on Kripke accessibility relations; it is thus natural to examine the logical counterparts of operations on relations. For instance, dynamic logic [16, 8, 7] (a logic of programs) uses relation composition (to express sequencing), transitive closure (to express iteration), union (to express non-deterministic choice). In this paper, we will be interested in the *inverse* operation on relations:

$$R^{-1}(x, y) = R(y, x)$$

Given a unary modality $\Box$ associated with accessibility relation R, we will use $\bar{\Box}$ to denote the modality associated with R^{-1}.

Inverse modalities have already been used in modal logics: in linear temporal logic, they are called *past* modalities. The table below summarises their intuitive meaning. These inverse modalities should not be confused with the dual modalities, nor with the inverse of the dual (which is of course dual of the inverse).

modality	inverse	dual	inverse dual
$\Box B$	$\bar{\Box} B$	$\Diamond B$	$\bar{\Diamond} B$
henceforth B	up to now, B	eventually B	once upon a time, B
tomorrow B	yesterday B	tomorrow B	yesterday B
I believe that B	in all situations where my beliefs admit the current situation, B is true	B is consistent with my beliefs	the current situation is consistent with my beliefs, and B
necessarily B	if reality is possible, B	possibly B	reality is possible, and B
Any result of program P satisfies B	Any input of program P satisfies B	Some result of program P satisfies B	Some input of program P satisfies B

In order to understand the reading of $\bar{\Box}$ for a particular reading of $\Box$, one should think about the meaning of the accessibility relation R, and then about its inverse. For example, if $\Box B$ is 'I believe that B', then $R(x, y)$ means: if the actual world is x, then y is a possible world according to my beliefs (in x). Thus, $\Box B$ holds at x if B holds in all worlds which could be the actual world according to my beliefs.

Now look at the inverses. $R^{-1}(x, y)$ means: if the actual world is y, then x is a possible world according to my beliefs (in y). So, $\bar{\Box} B$ holds at x if B holds in all worlds which, if it is the actual world then x is a possible world according to my beliefs in it. Thus, $\bar{\Box} B$ says: if the actual world is consistent with my beliefs, then B.

2.2.1 Axiomatising inverse modalities

How can we axiomatize the link between a modality and its inverse? It turns out that there are two ways, depending on the language we wish to use: We may either wish to keep the positive and negative modalities in separate languages, or to have a single language including both.

We first look at the latter:

Theorem 2.1 The following two axioms (each of which is also given in its dual form), when added to the classical rules of distribution and necessitation, axiomatize a pair of inverse modalities.

$$\begin{array}{lll} (1) & B \to \Box\bar{\Diamond} B & \Diamond\bar{\Box} B \to B \\ (2) & B \to \bar{\Box}\Diamond B & \bar{\Diamond}\Box B \to B \end{array}$$

Proof Let $\langle W, (S, R)\rangle$ be a frame, where R is the accessibility relation for $\Box, \Diamond$, and S is the relation for $\bar{\Box}, \bar{\Diamond}$. We show that the axioms hold in the frame iff $S = R^{-1}$.

($\Leftarrow$) is straightforward. For ($\Rightarrow$), suppose $S(x, y)$; choose the valuation V s.t. $V(p) = \{x\}$ then $x \Vdash p$, so by (2) $x \Vdash \bar{\Box}\Diamond p$ so $y \Vdash \Diamond p$ so $\exists z, R(y, z) \wedge z \Vdash p$. But by $V, z = x$, so $R(y, x)$. So $S \subseteq R^{-1}$. The converse inclusion is similar, but uses (1). □

We might prefer a rule that allows us to work with two separate sublanguages, one containing only the modalities $\Box, \Diamond$ and the other the modalities $\bar{\Box}, \bar{\Diamond}$. Theorems in one logic could be translated in the other one, provided we find inference rules that do not mix languages.

Theorem 2.2 Axiom (1) is equivalent to the following rule (which is also given in its dual form):

$$\frac{B \to \bar{\Box} C}{\Diamond B \to C} \qquad\qquad \frac{\bar{\Diamond} C \to B}{C \to \Box B}$$

Axiom (2) is equivalent to the rule (also given in dual form):

$$\frac{\Diamond B \to C}{B \to \bar{\Box} C} \qquad\qquad \frac{C \to \Box B}{\bar{\Diamond} C \to B}$$

Proof First half:

$$\begin{array}{lll} (\Rightarrow) & B \to \bar{\Box} C & \text{by hyp} \\ & \Diamond B \to \Diamond\bar{\Box} C & \text{by K,MP} \\ & \Diamond B \to C & \text{by (1) dual form} \end{array}$$

$$\begin{array}{lll} (\Leftarrow) & \bar{\Diamond} B \to \bar{\Diamond} B & \\ & B \to \Box\bar{\Diamond} B & \text{by rule (dual form)} \end{array}$$

The other half is symmetrical. □

Thus, the rule

$$\frac{B \to \bar{\Box}C}{\Diamond B \to C}$$

completely axiomatises the relationship of inverse between $\Box$ and $\bar{\Box}$. We will see in section 3.2 that the multi-modal version of this rule,

$$\frac{B \to \bar{\Box}_A C}{\Diamond_A B \to C}$$

exactly expresses the Ramsey rule for counterfactuals. $\bar{\Box}_A C$ will be interpreted as 'if A were true, then C would be true; $\Diamond_A B$ will be interpreted as the update of B by A, so the rule states that the counterfactual 'if A, C' is supported in a state B iff the state obtained by updating B with A supports C.

2.3 Other preliminaries

If the formula A has no modalities, we write $\mathrm{mod}(A)$ to mean the set of valuations which make A true, in the usual propositional way. Notice the difference between mod and $|\ |$.

The formula A over L is *complete* if for all formulas B over L, $A \models B$ or $A \models \neg B$.

If $(X, \leq)$ is a pre-ordered set and $Y \subseteq X$, then $\mathrm{Min}_{\leq}(Y)$ is the set of $\leq$-minimals in Y, i.e. $\mathrm{Min}_{\leq}(Y) = \{y \in Y \mid \forall x \in Y. x \not< y\}$. An order $\leq$ on a set of worlds W in some model M is *stoppered* if for every non-empty $|A| \subseteq W$ and $y \in |A|$ there is $x \in \mathrm{Min}_{\leq}(|A|)$ with $x \leq y$.

3 Updates and counterfactuals

We show that updates and conditionals are systems of modal logic, in sections 3.1 and 3.2 respectively. In section 3.3, we observe that they have inverse accessibility relations.

3.1 Updates

In [11], the difference between updates and revisions was pointed out and in [10] new postulates for updates were proposed. These postulates are similar to those for revisions [4] and are presented in Table 1. In the last column, we have indicated the name used in belief revision. As can be seen, properties of updates and revisions have much in common, explaining the historical confusion; indeed, in [9] updates are referred to as *pointwise revisions*.

Katsuno/Mendelzon's update axioms U8 and U4.1 suggest that update behaves like an existential modality on its second argument. Moreover, they observe [9, theorems 6.1, 6.3], [10, theorem 3.4] that

$$\mathrm{mod}(q \diamond p) = \bigcup_{y \in \mathrm{mod}(q)} \mathrm{Min}_{\leq_y}(\mathrm{mod}(p))$$

where $\leq_y$ is a preorder of closeness to y ($x \leq_y z$ means that x is at least as close to the world y as z is). If we write $\Diamond_p q$ in place of $q \diamond p$ and define the multi-modal model $M = \langle W, R, V \rangle$ where W is the set of valuations of the language, the relation R is given by

$$R_S(x, y) \Leftrightarrow x \in \mathrm{Min}_{\leq_y}(S),$$

name [10]	axiom	name [4]
U1	$q \diamond p \to p$	K*2
U2.1	$q \to p$ implies $q \to q \diamond p$	
U2.2	$q \to p$ implies $q \diamond p \to q$	K*4w
U3	$q \diamond p$ satisfiable, if p, q satisfiable	$\sim$ K*5
U4.1	$q \leftrightarrow r$ implies $q \diamond p \leftrightarrow r \diamond p$	
U4.2	$q \leftrightarrow r$ implies $p \diamond q \leftrightarrow p \diamond r$	K*6
U5	$(q \diamond r) \wedge p \to q \diamond (r \wedge p)$	K*7
U6	$q \diamond p \to r, q \diamond r \to p$ imply $q \diamond p \leftrightarrow q \diamond r$	
U7	q complete implies $(q \diamond p) \wedge (q \diamond r) \to q \diamond (p \vee r)$	
U8	$(q \vee r) \diamond p \leftrightarrow (q \diamond p) \vee (r \diamond p)$	

Table 1: Update postulates according to Katsuno/Mendelzon [10], using their notation. They use $\diamond$ as an infix operator; $q \diamond p$ means q updated by p.

and V is the identity, then Katsuno/Mendelzon's observation is equivalent to the standard satisfaction condition for an existential modality

$$x \Vdash \diamond_A B \quad \text{iff} \quad \text{there exists } y \text{ s.t. } R_{|A|}(x,y) \text{ and } y \Vdash B.$$

Adopting this suggestion, we recast U1-U8 as multi-modal axioms in Table 2.

To obtain the classical properties of a modality, we should also have necessitation:

$$\frac{B}{\Box_A B}$$

Theorem 3.1 Necessitation follows from the axioms and rules in Table 2.

Proof Assume $\models B$. Then $\models \neg B \leftrightarrow \bot$, and by U4.1: $\models \diamond_A \neg B \leftrightarrow \diamond_A \bot$. By U2.2: $\models \diamond_A \bot \to \bot$, so $\models \diamond_A \bot \leftrightarrow \bot$. Therefore, $\models \diamond_A \neg B \leftrightarrow \bot$, hence $\models \neg \diamond_A \neg B$. □

As usual within the framework of modal logic, we can study the 'correspondence properties' on R imposed by each of the axioms U1-U8. Moreover, since R is expressed in terms of $\leq$, we can look at what conditions of R and $\leq$ each of the conditions corresponds to. This occupies us for the remainder of this section.

Theorem 3.2 An axiom scheme or rule in Table 2 holds in a frame $F = \langle W, R \rangle$ iff R has the corresponding property stated in Table 3.

Compare correspondence theorems for standard modal logic, eg. [15, §5.2], [5, theorems 1.12, 1.13].

Proof The proofs follow the usual pattern in correspondence theory. In the $\Leftarrow$ direction, we add to the frame $\langle W, R \rangle$ an arbitrary valuation V to form the model $M = \langle W, R, V \rangle$, and show that the constraint on R is enough to guarantee that the axiom scheme or rule is satisfied at any point $x \in W$. In the $\Rightarrow$ direction, we make a judicious choice of the valuation and an instance of the scheme, to show that the constraint on R must hold.

name [10]	rewritten as
U1	$\Diamond_A B \to A$
U2.1	$\dfrac{B \to A}{B \to \Diamond_A B}$
U2.2	$\dfrac{B \to A}{\Diamond_A B \to B}$
U3	$\Diamond_A B$ satisfiable if A, B satisfiable
U4.1	$\dfrac{B \leftrightarrow C}{\Diamond_A B \leftrightarrow \Diamond_A C}$
U4.2	$\dfrac{B \leftrightarrow C}{\Diamond_B A \leftrightarrow \Diamond_C A}$
U5	$\Diamond_A B \wedge C \to \Diamond_{A \wedge C} B$
U6	$\dfrac{\Diamond_A B \to C \quad \Diamond_C B \to A}{\Diamond_A B \leftrightarrow \Diamond_C B}$
U7	B complete implies $\Diamond_A B \wedge \Diamond_C B \to \Diamond_{A \vee C} B$
U8	$\Diamond_A (B \vee C) \leftrightarrow \Diamond_A B \vee \Diamond_A C$

Table 2: Update postulates of Table 1 rewritten as modal logic axioms and rules

name	corresponding property of R	corresp. property of $\leq_y$
R1	$R_S(x, y)$ implies $x \in S$	–
R2.1	$y \in S$ implies $R_S(y, y)$	$y \leq_y x$ (weak centering)
R2.2	$y \in S$ and $R_S(x, y)$ imply $x = y$	$x \leq_y y$ implies $x = y$
R3	$S \neq \emptyset$ implies $\forall y \exists x . R_S(x, y)$	$\leq_y$ stoppered[a]
R5	$x \in S$ and $R_T(x, y)$ imply $R_{S \cap T}(x, y)$	–
R6	if $x \in S$, $\neg R_S(x, y)$ then $\exists z . R_S(z, y) \wedge$ $\forall T . (z \in T \to \neg R_T(x, y))$[a]	$\leq_y$ stoppered[a].
R7	$R_S \cap R_T \subseteq R_{S \cup T}$	

[a] sufficient condition only.

Table 3: Properties of R corresponding to the axioms/rules in Table 2

U1 $\Leftarrow$ R1. Let V be any valuation, and let $M = \langle W, R, V\rangle$. Suppose $x \Vdash_M \Diamond_A B$. Let $S = |A|$, and take a y such that $R_S(x,y)$ and $y \Vdash B$. By R1, $x \in S$, so $x \Vdash_M A$.

U1 $\Rightarrow$ R1. Suppose $R_S(x,y)$ in the frame $\langle W, R\rangle$; pick V such that $V(p) = S$ and $V(q) = \{y\}$. Since $y \Vdash q$, we have $x \Vdash \Diamond_p q$; so $x \Vdash p$ by U1, and $x \in S$ by def. of V.

U2.1 $\Leftarrow$ R2.1. Suppose $\models B \to A$ and $x \Vdash B$. Then $x \Vdash A$. We have $x \Vdash \Diamond_A B$ iff $\exists y$ s.t. $y \Vdash B$, and $R_{|A|}(x,y)$, which is true if we set $y = x$.

U2.1 $\Rightarrow$ R2.1. Suppose $y \in S$; we prove that $R_S(y,y)$. Let V be such that $V(p) = S$ and $V(q) = \{y\}$. It follows that $\models q \to p \vee q$, and therefore, by U2.1, $y \Vdash \Diamond_{p\vee q} q$. But $|p \vee q| = S$, so $\exists z \Vdash q$ s.t. $R_S(y,z)$. But $V(q) = \{y\}$. Thus, z must be y, and therefore $R_S(y,y)$.

U2.2 $\Leftarrow$ R2.2. Suppose $\models B \to A$ and $x \Vdash \Diamond_A B$. Take y such that $R_{|A|}(x,y)$ and $y \Vdash B$. Since $\models B \to A$, $y \in |A|$; so by R2.2, $x = y$; so $x \Vdash B$.

U2.2 $\Rightarrow$ R2.2. Suppose $y \in S$ and $R_S(x,y)$. Let V be such that $V(p) = S$ and $V(q) = \{y\}$. It follows that $\models q \to p \vee q$, and therefore, by U2.2, $\Diamond_{p\vee q} q \to q$. Since $y \Vdash q$, $x \Vdash \Diamond_{p\vee q} q$, so $x \Vdash q$, so $x = y$ (by def. of V).

U3 $\Leftarrow$ R3. Take any valuation. Suppose A and B are satisfiable in M, so take $a \Vdash A$ and $b \Vdash B$. Then $|A| \neq \emptyset$, so by R3 there is an x with $R_{|A|}(x,b)$. Hence, $x \Vdash \Diamond_A B$.

U3 $\Rightarrow$ R3. Suppose $S \neq \emptyset$ and y is given. Take V such that $V(p) = S$ and $V(q) = \{y\}$. Then by U3, $\Diamond_p q$ is satisfiable, i.e. there is an x with $x \Vdash \Diamond_p q$. Then there is a z with $R_S(x,z)$ and $z \Vdash q$. But by choice of V, $z = y$; so $R_S(x,y)$.

U5 $\Leftarrow$ R5. Suppose $x \Vdash \Diamond_A B \wedge C$. Since $x \Vdash \Diamond_A B$, $\exists y \Vdash B$, $R_{|A|}(x,y)$. By R5, $R_{|A|\cap|C|}(x,y)$ and thus $x \Vdash \Diamond_{A\wedge C} B$.

U5 $\Rightarrow$ R5. Suppose $x \in S$ and $R_T(x,y)$. Pick V such that $V(p) = S$, $V(q) = \{y\}$, and $V(r) = T$. Then $y \Vdash q$ and $R_T(x,y)$ imply that $x \Vdash \Diamond_r q$. Since $x \Vdash p$, by U5 $x \Vdash \Diamond_{r\wedge p} q$. Therefore, $\exists y'$ s.t $y' \Vdash q$ and $R_{S\cap T}(x,y')$. But y' must be y, since $V(q) = \{y\}$, so $R_{S\cap T}(x,y)$.

U6 $\Leftarrow$ R6. The condition R6 deserves an explanation. Intuitively, it says: if x is eliminated from our preferred set, it is justified by some z that consistently gets preferred.

Suppose $\models \Diamond_A B \to C$, $\models \Diamond_C B \to A$, and $x \Vdash \Diamond_A B$. Then $\exists y_0\, R_{|A|}(x,y_0)$ and $y_0 \Vdash B$; and from $\models \Diamond_A B \to C$ we have $x \Vdash C$. Suppose for a contradiction that $x \nVdash \Diamond_C B$; then $\neg R_{|C|}(x,y_0)$. Now apply R6 with $S = |C|$, and take $T = |A|$; there exists z with $R_{|C|}(z,y_0) \wedge z \nVdash A$. Use $\models \Diamond_C B \to A$ to show that $z \nVdash \Diamond_C B$, which contradicts $R_{|C|}(z,y_0) \wedge y_0 \Vdash B$.

U7 $\Leftarrow$ R7 Suppose $x \Vdash \Diamond_A B \wedge \Diamond_C B$. Then exist y_1, y_2 with $R_{|A|}(x,y_1)$, $R_{|C|}(x,y_2)$ and $y_1, y_2 \Vdash B$. Since B complete, $y_1 = y_2$, so $(R_{|A|} \cap R_{|C|})(x,y_1)$, hence by R7 $R_{|A|\cup|C|}(x,y_1)$

U7 $\Rightarrow$ R7 Supposing $R_S(x,y)$ and $R_T(x,y)$, pick V such that $V(p) = S$, $V(q) = \{y\}$ and $V(r) = T$. Then $y \Vdash q$, so $x \Vdash \Diamond_p q \wedge \Diamond_r q$. By U7, $x \Vdash \Diamond_{p\vee r} q$, so there exists z (which by definition of V must be y), with $R_{|p\vee r|}(x,z)$. But $|p \vee r| = S \cup T$, so $R_{S\cup T}(x,y)$. □

The axioms U4.1, U4.2 and U8 are simply the usual properties of an existential modality; thus, they do not constrain the accessibility relation.

Theorem 3.3 The axiom schemes U4.1, U4.2 and U8 hold in any frame.

Proof U4.1. Suppose $\models B \leftrightarrow C$ and $x \Vdash \Diamond_A B$. Then there exists y with $R_{|A|}(x,y)$ and $y \Vdash B$. But also, $y \Vdash C$, so $x \Vdash \Diamond_A C$. The other half is similar.

U4.2. Suppose $\models B \leftrightarrow C$ and $x \Vdash \Diamond_B A$. Then there exists y with $R_{|B|}(x,y)$ and $y \Vdash A$. But also, $R_{|C|}(x,y)$, so $x \Vdash \Diamond_C A$. The other half is similar.

U8. $x \Vdash \Diamond_A(B \vee C)$ iff $\exists y\, R_{|A|}(x,y), y \Vdash B \vee C$ iff $\exists y_1\, R_{|A|}(x,y_1), y_1 \Vdash B$ or $\exists y_2\, R_{|A|}(x,y_2), y_2 \Vdash C$ iff $x \Vdash \Diamond_A B \vee \Diamond_A C$.

□

3.2 Counterfactuals

According to [18, 12, 14], the counterfactual 'if A was the case, then B would be the case' may be interpreted by: "In all closest worlds satisfying A, we find that B holds." It is well-known that counterfactuals have the properties of classical universal modalities [1, 12]. The counterfactual 'if A was the case, then B would be the case' holds at a world x if B holds in all y in $\mathrm{Min}_{\leq_x}|A|$. But this relation between x and y is simply the inverse of the relation R given in section 3.1. Thus, we see that counterfactuals are not just a universal modality; they are the inverse dual modality to updates. The counterfactual sentence 'if A was the case, then B would be the case' can be written $\bar{\Box}_A B$, where

$$x \Vdash \bar{\Box}_A B \quad \text{iff} \quad \text{for all } y,\ R^{-1}_{|A|}(x,y) \text{ implies } y \Vdash B.$$

One can perform the same analysis as we did for updates, namely, the correspondence theory for standard postulates for counterfactuals. This is work in progress.

4 Inter-translating systems for counterfactuals and updates

We have observed that postulates for updates correspond to particular properties of the accessibility relation R, and similarly for counterfactuals, whose postulates correspond to properties of the inverse relation R^{-1}. This means that a particular postulate for updates can be translated into a postulate for counterfactuals. The proof of the equivalence between the postulates can be performed

- either by going via the accessibility relation R;
- or by working directly with the axiomatisations of theorem 2.1;
- or by working with the Ramsey Rule (theorem 2.2).

We have worked out the counterfactual counterpart of the postulates U1-U8. Our preliminary findings at the time of going to press are summarised in the following table.

name	counterfactual axiom
C1	$\Box_A A$
C2.1	$B \rightarrow A$ implies $B \rightarrow \bar{\Diamond}_A B$
C2.2	$B \rightarrow A$ implies $B \rightarrow \bar{\Box}_A B$
C3	$B \rightarrow \bar{\Box}_A \bot$ implies $\neg B$ or $\neg A$
C4.2	$B \leftrightarrow C$ implies $\bar{\Box}_B A \leftrightarrow \bar{\Box}_C A$
C5	$(\bar{\Box}_A C \wedge \bar{\Box}_B C) \rightarrow \bar{\Box}_{A \vee B} C$[a]
C6	$B \rightarrow \bar{\Box}_A C, B \rightarrow \bar{\Box}_C A$ imply $\bar{\Box}_A B \leftrightarrow \bar{\Box}_C$
C7	B complete implies $\bar{\Diamond}_A B \wedge \bar{\Diamond}_C B \rightarrow \bar{\Diamond}_{A \vee C} B$

[a] using U1,U4,U5

For example, the translation of U1 is obtained as follows.

$$\begin{array}{rcll} \models \Diamond_A B \rightarrow A & \Leftrightarrow & \models B \rightarrow \bar{\Box}_A A & \text{Ramsey Rule} \\ & \Leftrightarrow & \models \bar{\Box}_A A & \end{array}$$

5 Conclusions

The link between counterfactual and updates, often considered as esoteric, is only the usual link between a relation and its inverse; counterfactuals, as is known, can be considered as a universal modality, and update as its inverse existential modality.

Some work remains to be done:

- obtain similar results for the inverse modalities used in other logics (e.g. temporal logic)
- study also known axioms proposed for counterfactuals and derive the corresponding update axiom.

5.1 Acknowledgements

Thanks to the ModelAge'96 Workshop referees for their comments on an early draft of this paper, and to Esprit for partial funding under the WG ModelAge (8319).

References

[1] B. F. Chellas. Basic conditional logic. *Journal of Philosophical Logic*, 4:133–153, 1980.

[2] P. Gärdenfors. Belief revision and the Ramsey test for conditionals. *Philosophical Review*, 91:81–93, 1986.

[3] P. Gärdenfors. *Knowledge in Flux: Modelling the Dynamics of Epistemic States.* MIT Press, 1988.

[4] Peter Gärdenfors. *Knowledge in Flux: Modeling the Dynamics of Epistemic States.* MIT Press, Bradford Books, Cambridge, MA, 1988.

[5] R. Goldblatt. *Logics of Time and Computation.* CSLI Lecture Notes, 1987.

[6] G. Grahne. Updates and counterfactuals. In *Proc. Second International Conference on Principles of Knowledge Representation and Reasoning (KR '91)*, pages 269–276. Morgan Kaufmann, San Francisco, CA, 1991.

[7] D. Harel. Dynamic logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Phylosophical Logic, Vol II*, page 715 ff. Reidel Publ. Co., Dordrecht, Holland, 1987.

[8] David Harel. *First-Order Dynamic Logic*, volume 68 of *Lecture Notes in Computer Science*. Springer-Verlag, New York, 1979.

[9] Hirofumi Katsuno and Alberto O. Mendelzon. Propositional knowledge base revision and minimal change. *Artificial Intelligence*, 52:263–294, 1991.

[10] Hirofumi Katsuno and Alberto O. Mendelzon. On the difference between updating a knowledge base and revising it. In Peter Gärdenfors, editor, *Belief Revision*, number 29 in Cambridge Tracts in Theoretical Computer Science, pages 183–203. Cambridge University Press, 1992.

[11] Arthur M. Keller and Marianne Winslett Wilkins. On the use of an extended relational model to handle changing incomplete information. *IEEE Transactions on Software Engineering*, 11(7):620–633, July 1985.

[12] David K. Lewis. *Counterfactuals*. Harvard University Press, 1973.

[13] D. Makinson. How to give it up: A survey of some formal aspects of the logic of theory change. *Synthèse*, 62:347–363, 1985.

[14] John L. Pollock. A refined theory of counterfactuals. *Journal of Philosophical Logic*, 10:239–266, 1981.

[15] S. Popkorn. *First Steps in Modal Logic*. Cambridge University Press, 1994.

[16] V. R. Pratt. Models of program logics. In *Proc. 20th IEEE Symp. on Foundations of Computer Science*, pages 115–122, 1979.

[17] F. P. Ramsey. Truth and probability. In R. B. Braithwaite, editor, *The Foundations of Probability and other Logical Essays*. Harcourt Brace, New York, 1931.

[18] Robert C. Stalnaker. *A Theory of Conditionals*, volume 2 of *American Philosophical Quarterly Monograph Series (ed. Nicholas Rescher)*, pages 98–112. Blackwell, Oxford, 1968.

An Outline of ALX3, a Multi-Agent Action Logic

Zhisheng Huang
Computer Science Department
Queen Mary and Westfield College
University of London

Michael Masuch
Center for Computer Science in Organization and Management (CCSOM)
University of Amsterdam
{huang,michael}@ccsom.uva.nl

Abstract

ALX3 is a multi-agent version of ALX with a first-order description language. ALX, a modal action logic, combines ideas from H.A. Simon's *bounded rationality*, S. Kripke's *possible world semantics*, G. H. von Wright's *preference logic*, Pratt's *dynamic logic*, and Stalnaker's *conditional logic*. ALX avoids important drawbacks of other action logics, especially the counterintuitive necessitation rule for goals (every theorem must be a goal) and the equally counterintuitive closure of goals under logical implication. ALX3 is sound and complete, and is already proving its practical use in the formal representation of modern organization theory.

1 Introduction

Action logics are usually developed for the (hypothetical) use by intelligent robots [CL90, GS87, RG91] or as a description language of program behavior [Har84]. Our effort is motivated by a different concern. We want to develop a formal language for social science theories, especially for theories of organizations. The difference in motivation leads to a new approach to action logic. It combines ideas from various strands of thought, notably H.A. Simon's notion of *bounded rationality*, Kripke's *possible world semantics*, V.R. Pratt's *dynamic logic*, Stalnaker's notion of *minimal change*, G. H. von Wright's approach to *preferences*, and J. Hintikka's approach to *knowledge* and *belief*. ALX3 has a first order description language with multiple agents, and four modal operator types.[1] We outline the language, and discuss some of its important properties. A fuller presentation is given in [HM94]. In a companion paper, ALX3's potential for knowledge representation is extensively demonstrated in the formalization of an important organization theory, J.D. Thompson's *Organizations in Action* [MH94].

2 ALX's Background

Most social science theories are expressed in natural language, but natural language does not provide a formal scaffold for checking a theory's logical properties. As a consequence, the social sciences have acquired a reputation for "softness" — a soft way of saying that

[1] ALX stands for the x's *A*ction *L*ogic. ALX1, the first version, had a propositional description language and a (backward-looking) update operator instead of the conditional; it was a single-agent language [HMP95]. ALX2, the intermediate version, is not multi-agent [HMP93].

the logical properties of their theories are often dubious. Reformulating a social theory in a formal language with known logical properties would facilitate the tasks of consistency checking, disambiguation, or the examination of other important logical properties, such as contingency (whether or not the theory is falsifiable).

We focus on action logic as a formal language, because actions are key to the understanding of social phenomena. In fact, most social scientist agree that action theory provides the underlying framework for the social sciences in general [Blu69, Gid79, Par37]. Yet actions involve attitudes and engender change, and both phenomena are notoriously hard to grasp in the extensional context of first order languages [Gam90]. This explains our attempt to develop a new logic, rather than taking First Order Logic off the shelf.

Herbert A. Simon's conceptualization of *bounded rationality* [Sim55] serves as a point of departure. His approach is intuitively appealing, and had great impact on the postwar social sciences. Simon wanted to overcome the omniscience claims of the traditional conceptualizations of rational action. He assumed (1) an agent with (2) a set of behavior alternatives, (3) a set of future states of affairs (each such state being the outcome of a choice among the behavior alternatives), and (4) a preference order over future states of affairs. The omniscient agent, endowed with "perfect rationality", would know all behavior alternatives and the exact outcome of each alternative; the agent would also have a complete preference ordering for those outcomes. An agent with bounded rationality, in contrast, might not know all alternatives, nor need it know the exact outcome of each; also, the agent might lack a complete preference ordering for those outcomes.

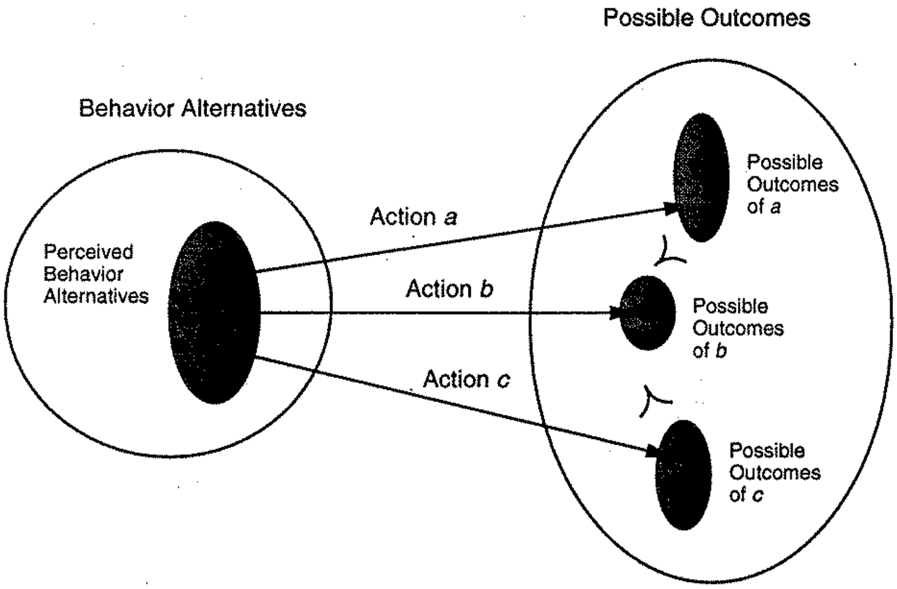


Figure 1: Simon's Bounded Rationality

Kripke's *possible world semantics* provides a natural setting for Simon's conceptualization. We assume a set of possible worlds with various relations defined over this set (we may also call those possible worlds *states*). One can see a behavior alternative as a mapping from states to states, so each behavior alternative constitutes an accessibility relation. An accessibility relation, in turn, can be interpreted as an opportunity for action, that is, as an opportunity for changing the world by moving from a given state to another state. Accessibility relations are expressed by indexed one-place modal operators, as in dynamic logic [Har84]. For example, the formula $\langle a_i \rangle \phi$ expresses the fact that the agent has an action a

at its disposal such that effecting a in the present situation would result in the situation denoted by ϕ.

Preferences – not goals – provide the basic rationale for rational action in ALX3. Following von Wright [vWr63], a preference statement is understood as a statement about situations. For example, the statements that "I prefer oranges to apples" is interpreted as the fact that "I prefer the states in which I have an orange to the states in which I have an apple." Following von Wright again, we assume that an agent who says that she prefers oranges to apples should prefer a situation where she has an orange but *no* apple to a situation where she has an apple but *no* orange. We call this principle *conjunction expansion principle* and restrict attention to preference statements that obey it. Preferences are expressed via two-place modal operators; if the agent prefers the proposition ϕ to the proposition ψ, we write $\phi\mathbf{P}_i\psi$.

Normally, the meaning of a preference statement is context dependent, even if this is not made explicit. An agent may say to prefer an apple to an orange – and actually mean it – but she may prefer an orange to an apple later – perhaps because then she already had an apple. To capture this context dependency, we borrow the notion of minimal change from Stalnaker's approach to conditionals [Sta68]. The idea is to apply the conjunction expansion principle only to situations that are minimally different from the agent's present situation – just as different as they really need to be in order to make the propositions true about which preferences are expressed. We introduce a binary function, cw, to the semantics that determines the set of "closest" states relative to a given state, such that the new states fulfill some specified conditions, but resembles the old state as much as possible in all other respects.

The syntactic equivalent of the closest world function is the wiggled "causal arrow". It appears in expressions such as $\phi \leadsto \psi$ where it denotes: in all closest worlds where ϕ holds, ψ also holds. The causal arrow expresses the conditional notion of a causal relation between ϕ and ψ: if ϕ *were* the case, ψ *would* also be the case.

The last primitive operator of ALX3 is the indexed belief operator. In a world of bounded rationality, an agent's beliefs do not necessarily coincide with reality, and in order to make this distinction, we must be able to distinguish between belief and reality; $\mathbf{B}_i(\phi)$ will denote the fact that agent i believes ϕ. As the logical axioms characterizing the belief operator show, $\mathbf{B}$ represents a sense of "subjective knowledge", not metaphysical attachment, or epistemic uncertainty.

3 Syntax and Semantics

3.1 Formal Syntax

ALX3 has the following primitive symbols:

(1) For each natural number $n(\geq 1)$, a countable set of n-place predicate letters, $p_i, p_j, ...$

(2) A countable set of regular variables, written $x, x_1, y, z, ...$; a countable set of action variables, written $a, a_1, b, ...$; a countable set of agent variables, written $i, i_1, j, ...$

(3) A countable set of regular constants, written $c, c_1, c_2, ...$; a countable set of actions constants, written $ac, ac_1, ac_2, ...$; a countable set of agent constants, written $ag, ag_1, ag_2, ...$

(4) The symbols $\neg$(negation), $\wedge$(conjunction), $\boldsymbol{B}$(belief), $\exists$(existential quantifier), $\boldsymbol{P}$(preference), $\leadsto$(conditional), ;(sequence), $\cup$(choice), $\langle, \rangle$, (, and).

Furthermore, ALX3 has the following syntax rules:

$$\begin{array}{ll}
\langle\text{Variable}\rangle & ::= \langle\text{Regular variable}\rangle|\langle\text{Action variable}\rangle|\langle\text{Agent variable}\rangle \\
\langle\text{Constant}\rangle & ::= \langle\text{Regular constant}\rangle|\langle\text{Action constant}\rangle|\langle\text{Agent constant}\rangle \\
\langle\text{Term}\rangle & ::= \langle\text{Variable}\rangle|\langle\text{Constant}\rangle \\
\langle\text{Action term}\rangle & ::= \langle\text{Action variable}\rangle|\langle\text{Action constant}\rangle \\
\langle\text{Agent term}\rangle & ::= \langle\text{Agent variable}\rangle|\langle\text{Agent constant}\rangle \\
\langle\text{Atom}\rangle & ::= \langle\text{Predicate}\rangle(\langle\text{Term}\rangle,\cdots,\langle\text{Term}\rangle) \\
\langle\text{Action}\rangle & ::= \langle\text{Action term}\rangle_{\langle\text{Agent term}\rangle}|\langle\text{Action}\rangle;\langle\text{Action}\rangle|\langle\text{Action}\rangle \cup \langle\text{Action}\rangle \\
\langle\text{Formula}\rangle & ::= \langle\text{Atom}\rangle|\neg\langle\text{Formula}\rangle|\langle\text{Formula}\rangle \wedge \langle\text{Formula}\rangle|\exists\langle\text{Variable}\rangle\langle\text{Formula}\rangle| \\
 & \langle\langle\text{Action}\rangle\rangle\langle\text{Formula}\rangle|\langle\text{Formula}\rangle \leadsto \langle\text{Formula}\rangle| \\
 & \langle\text{Formula}\rangle\mathbf{P}_{\langle\text{Agent term}\rangle}\langle\text{Formula}\rangle|\mathbf{B}_{\langle\text{Agent term}\rangle}\langle\text{Formula}\rangle
\end{array}$$

3.2 Semantics

Definition 1 (ALX3 Model)

$$\textit{Call}\;\; M = \langle O, PA, AGENT, W, cw, \succ, \mathcal{R}, \mathcal{B}, I\rangle$$

an ALX3 model, if O is a set of objects, PA is a set of primitive actions, $AGENT$ is a set of agents, W is a set of possible worlds, $cw : W \times \mathcal{P}(W) \rightarrow \mathcal{P}(W)$ is a closest world function, $\succ: AGENT \rightarrow \mathcal{P}(\mathcal{P}(W) \times \mathcal{P}(W))$ is a function that assigns a comparison relation for preferences to each agent, $\mathcal{R} : AGENT \times PA \rightarrow \mathcal{P}(W \times W)$ is a function that assigns an accessibility relation to each agent and each primitive action, $\mathcal{B} : AGENT \rightarrow \mathcal{P}(W \times W)$ is a function that assigns a serial and transitive accessibility relation for the belief operation to each agent, I is a interpretation function, and if $cw, \succ$, satisfy the following conditions respectively:

$$\begin{array}{ll}
(\boldsymbol{CS_1}): & cw(w,X) \subseteq X. \\
(\boldsymbol{CS_2}): & w \in X \Rightarrow cw(w,X) = \{w\}. \\
(\boldsymbol{CS_3}): & cw(w,X) = \emptyset \Rightarrow cw(w,Y) \cap X = \emptyset. \\
(\boldsymbol{CS_4}): & cw(w,X) \subseteq Y \text{ and } cw(w,Y) \subseteq X \Rightarrow cw(w,X) = cw(w,Y). \\
(\boldsymbol{CS_5}): & cw(w,X) \cap Y \neq \emptyset \Rightarrow cw(w, X \cap Y) \subseteq cw(w,X). \\
(\boldsymbol{NORM}): & (\emptyset \not\succ_i X), (X \not\succ_i \emptyset), \text{where } \;\succ_i = \succ(i) \text{ for each agent } i \in AGENT. \\
(\boldsymbol{TRAN}): & cw(w, X \cap \overline{Y}) \succ_i cw(w, Y \cap \overline{X}) \text{ and } cw(w, y \cap \overline{Z}) \succ_i cw(w, Z \cap \overline{Y}) \;. \\
 & \Rightarrow cw(w, X \cap \overline{Z}) \succ_i cw(w, Z \cap \overline{X}), \text{where } \overline{X} = W - X.
\end{array}$$

The (CS#) constrain the closest world function. They constitute the standard requirements for a closest world function as established by Stalnaker[Sta68]. (NORM) and (TRAN) constrain the semantic preference relation. (NORM) is required in support of the logical axiom (N) (normality), which, in turn, protects the preference logic against counterintuitive consequences. (TRAN) guarantees the soundness of the logical axiom (TR) which, in turn, assures transitivity for preferences. The seriality and transitivity of the relation $\mathcal{B}$ are standard requirements for the semantics of beliefs.

Definition 2 (Meaning function)

$$\begin{array}{l}
[\![p(t_1,...,t_n)]\!]^v_M = \{w : \langle v_I(t_1),...,v_I(t_n)\rangle \in I(p,w)\} \\
[\![\neg\phi]\!]^v_M = W \setminus [\![\phi]\!]^v_M \\
[\![\phi \wedge \psi]\!]^v_M = [\![\phi]\!]^v_M \cap [\![\psi]\!]^v_M \\
[\![\exists x\phi]\!]^v_M = \{w : (\exists d \in D)(w \in [\![\phi]\!]^{v(d/x)}_M)\} \\
[\![\langle a\rangle\phi]\!]^v_M = \{w : (\exists w')(R^a ww' \text{ and } w' \in [\![\phi]\!]^v_M)\} \\
[\![\phi \leadsto \psi]\!]^v_M = \{w : cw(w, [\![\phi]\!]^v_M) \subseteq [\![\psi]\!]^v_M\} \\
[\![\phi \boldsymbol{P}_i \psi]\!]^v_M = \{w : cw(w, [\![\phi \wedge \neg\psi]\!]^v_M) \succ_{v_I(i)} cw(w, [\![\psi \wedge \neg\phi]\!]^v_M)\} \\
[\![\boldsymbol{B}_i\phi]\!]^v_M = \{w : (\forall w')(\langle w,w'\rangle \in \mathcal{B}_{v_I(i)} \Rightarrow w' \in [\![\phi]\!]^v_M)\}
\end{array}$$

The interpretation of the atomic formulas, the boolean connectives and the existential quantifier is straightforward. The interpretation of $\langle a\rangle\phi$ yields the set of worlds from where the agent can access at least one ϕ-world via action a. The interpretation of $\phi \rightsquigarrow \psi$ yields the set of worlds in which the closest ϕ-worlds are also ψ-worlds. So, $\phi \rightsquigarrow \psi$ is true at a world if ϕ would *ceteris paribus* entail ψ. This is the standard counter-factual conditional used to express a causal relation between ϕ and ψ. Note that our wiggled arrow does not require actual counter-factuality, so ϕ may be true in the actual world. The interpretation of $\phi\boldsymbol{P}_i\psi$ assures the conjunction expansion principle.

Definition 3 (ALX3 inference system)

$$
\begin{array}{ll}
(\boldsymbol{BA}): & \textit{all tautologies of the first order logic} \\
(\boldsymbol{A1}): & \langle a\rangle\bot \leftrightarrow \bot \\
(\boldsymbol{A2}): & \langle a\rangle(\phi \vee \psi) \leftrightarrow \langle a\rangle\phi \;\vee \langle a\rangle\psi \\
(\boldsymbol{A3}): & \langle a;b\rangle\phi \leftrightarrow \langle a\rangle\langle b\rangle\phi \\
(\boldsymbol{A4}): & \langle a \cup b\rangle\phi \leftrightarrow \langle a\rangle\phi \vee \langle b\rangle\phi \\
(\boldsymbol{AU}): & [a]\forall x\phi \leftrightarrow \forall x[a]\phi \\
(\boldsymbol{ID}): & \psi \rightsquigarrow \psi \\
(\boldsymbol{MPC}): & (\psi \rightsquigarrow \phi) \rightarrow (\psi \rightarrow \phi) \\
(\boldsymbol{CC}): & (\psi \rightsquigarrow \phi) \wedge (\psi \rightsquigarrow \phi') \rightarrow (\psi \rightsquigarrow \phi \wedge \phi') \\
(\boldsymbol{MOD}): & (\neg\psi \rightsquigarrow \psi) \rightarrow (\phi \rightsquigarrow \psi) \\
(\boldsymbol{CSO}): & [(\psi \rightsquigarrow \phi) \wedge (\phi \rightsquigarrow \psi)] \rightarrow [(\psi \rightsquigarrow \chi) \leftrightarrow (\phi \rightsquigarrow \chi)] \\
(\boldsymbol{CV}): & [(\psi \rightsquigarrow \phi) \wedge \neg(\psi \rightsquigarrow \neg\chi)] \rightarrow [(\psi \wedge \chi) \rightsquigarrow \phi] \\
(\boldsymbol{CS}): & (\psi \wedge \phi) \rightarrow (\psi \rightsquigarrow \phi) \\
(\boldsymbol{CEP}): & \phi\boldsymbol{P}_i\psi \leftrightarrow (\phi \wedge \neg\psi)\boldsymbol{P}_i(\neg\phi \wedge \psi) \\
(\boldsymbol{N}): & \neg(\bot\boldsymbol{P}_i\phi), \neg(\phi\boldsymbol{P}_i\bot) \\
(\boldsymbol{TR}): & (\phi\boldsymbol{P}_i\psi) \wedge (\psi\boldsymbol{P}_i\chi) \rightarrow (\phi\boldsymbol{P}_i\chi) \\
(\boldsymbol{PC}): & (\phi\boldsymbol{P}_i\psi) \rightarrow \neg((\phi \wedge \neg\psi) \rightsquigarrow \neg(\phi \wedge \neg\psi)) \wedge \neg((\psi \wedge \neg\phi) \rightsquigarrow \neg(\psi \wedge \neg\phi)) \\
(\boldsymbol{KB}): & \boldsymbol{B}_i\phi \wedge \boldsymbol{B}_i(\phi \rightarrow \psi) \rightarrow \boldsymbol{B}_i\psi \\
(\boldsymbol{DB}): & \neg\boldsymbol{B}_i\bot \\
(\boldsymbol{4B}): & \boldsymbol{B}_i\phi \rightarrow \boldsymbol{B}_i\boldsymbol{B}_i\phi \\
(\boldsymbol{BFB}): & \forall x\boldsymbol{B}_i\phi \leftrightarrow \boldsymbol{B}_i\forall x\phi
\end{array}
$$

$$
\begin{array}{ll}
(\boldsymbol{MP}): & \vdash \phi \,\&\, \vdash \phi \rightarrow \psi \Rightarrow\vdash \psi \\
(\boldsymbol{G}): & \vdash \phi \Rightarrow\vdash \forall x\phi \\
(\boldsymbol{NECA}): & \vdash \phi \Rightarrow\vdash [a]\phi \\
(\boldsymbol{NECB}): & \vdash \phi \Rightarrow\vdash \boldsymbol{B}_i\phi \\
(\boldsymbol{MONA}): & \vdash \langle a\rangle\phi \,\&\, \vdash \phi \rightarrow \psi \Rightarrow\vdash \langle a\rangle\psi \\
(\boldsymbol{MONC}): & \vdash \phi \rightsquigarrow \psi \,\&\, \vdash \psi \rightarrow \psi' \Rightarrow\vdash \phi \rightsquigarrow \psi' \\
(\boldsymbol{SUBA}): & \vdash (\phi \leftrightarrow \phi') \Rightarrow\vdash ((a)\phi) \leftrightarrow ((a)\phi') \\
(\boldsymbol{SUBC}): & \vdash (\phi \leftrightarrow \phi') \,\&\, \vdash (\psi \leftrightarrow \psi') \Rightarrow\vdash (\phi \rightsquigarrow \psi) \leftrightarrow (\phi' \rightsquigarrow \psi') \\
(\boldsymbol{SUBP}): & \vdash (\phi \leftrightarrow \phi') \,\&\, \vdash (\psi \leftrightarrow \psi') \Rightarrow\vdash (\phi\boldsymbol{P}_i\psi) \leftrightarrow (\phi'\boldsymbol{P}_i\psi')
\end{array}
$$

Most axioms are straightforward. As usual, we have the tautologies (BA). Since ALX3 is a normal modal logic, the absurdum is not true anywhere, so it is not accessible (A1). The action modalities behave as usual, so they distribute over disjunction both ways (A2) (they also distribute over conjunction in one direction, but the corresponding axiom is redundant). (A3) characterizes the sequencing operator ';' and (A4) does the same for the indeterminate choice of actions. (AU) establishes the Barcan formula for universal action modalities. We have the Barcan formula because the underlying domain D is the same in all possible worlds.

The next seven axioms characterize the intensional conditional. Informally speaking, they syntactically specify the meaning of "ceteris paribus" in ALX3. They are fairly standard, and, with the exception of (CC), they already provide a characterization of Lewis' system **VC**, which, in turn, is an adaptation of Stalnaker's conditional logic to a system for non-unique closest worlds. (ID) establishes the triviality that ψ is true in all closest ψ-worlds; (MPC) relates the intensional and the material conditional in the obvious way: so if ϕ would hold given ψ, then, if ψ actually does hold, ϕ must also hold. Conjunction distributes over the "wiggled arrow" in one way (CC). (MOD) rules out the eventuality of closest absurd worlds; (CSO) gives an identity condition for closest worlds, (CV) establishes a cautious monotony for the intensional conditional, and (CS) relates the conjunction to the intensional conditional. Replacing (CS) by $(\phi \rightsquigarrow \psi) \vee (\phi \rightsquigarrow \neg\psi)$ yields Stalnaker's original system, as the new axiom would require the uniqueness of the closest possible world).

The next four axioms characterize the preference relation. (CEP) states the conjunction expansion principle. (N) establishes "normality" and (TR) transitivity. As noted before, (TR) would go if its semantic equivalent, (TRAN), goes, so we could have non-transitive preferences. (CEP) and (N) together imply the irreflexivity of the **P** operator[Hua94]. The axiom (PC) says that if an agent i prefers ϕ to ψ, then both $\phi \wedge \neg\psi$ and $\psi \wedge \neg\phi$ are possible.

The last four axioms give a characterization of the belief operator. As pointed out above, our belief operator is designed to represent subjective knowledge. (KB) is standard in epistemic logic, but it is often criticized, since it requires logical omniscience with respect to the material conditional. On the other hand, one would expect to draw correct logical inferences when necessary, so not having (KB) may be worse. (DB) rules out the belief in absurdities, (4B) establishes positive self-introspection for beliefs, and (BFB) is the Barcan formula for beliefs. These four axioms give a standard characterization of subjective knowledge. Together with the inference rules (MP) and (NECB), they turn the belief operation into a weak S4 system. As shown in [Hua94], we could weaken the belief operator considerably, but these weaker alternatives have their own problems that would overload ALX. (One radical alternative would be an empty belief operator.)

The remaining expressions characterize ALX3's inference rules. We have the modus ponens and generalization for obvious reasons. By the same token, we have the necessitation rule for the universal action modality: if indeed, ϕ is true in all worlds, then all activities will lead to ϕ-worlds; by the same token, we have the necessitation rule for beliefs. (MONA) connects the meaning of the action modality with the meaning of the material conditional. We have right monotonicity for the intensional conditional but *not* left monotonicity. Furthermore, logically equivalent propositions are substitutable in action-, conditional-, and preference formulae (SUBA), (SUBC), (SUBP). Note that we do *not* have monotonicity for preferences. Because of this, we are able to avoid the counterintuitive deductive closure of goals that mars other action logics. In [HM94], we prove that ALX3 is sound and complete.

4 Defined Operators

ALX provides considerable flexibility via the definition of additional modal operators. The standard alethic operators "Necessity" and "Possibility" are (inter)definable via the causal conditional [Sta68]:

$$\Box\phi \overset{\text{def}}{\Longleftrightarrow} \neg\phi \rightsquigarrow \phi \qquad \Diamond\phi \overset{\text{def}}{\Longleftrightarrow} \neg(\Box\neg\phi)$$

These alethic operators have the conventional properties; in particular, they support the traditional (K), (T), and (D) axioms. We define the knowledge operator along classical lines as true belief:

$$\boldsymbol{K}_i\phi \overset{\text{def}}{\Longleftrightarrow} \phi \wedge \boldsymbol{B}_i\phi$$

The knowledge-operator has the familiar properties. See [Hua94, HM94]. It it sometimes relevant whether agent i can directly access a particular state via an action. Define direct accessibility as follows:

$$\mathbf{DA}_i(\phi) \overset{\text{def}}{\Longleftrightarrow} \exists a\langle a_i\rangle\phi$$

so a state ϕ is directly accessible if the agent has an action that can bring about ϕ. A state may not be directly accessible, even though it may be accessible via another, directly accessible state. Define accessibility:

$$\mathbf{A}_i(\phi) \overset{\text{def}}{\Longleftrightarrow} \mathbf{DA}_i(\phi) \vee (\mathbf{DA}_i(\psi) \wedge (\psi \rightsquigarrow \phi))$$

so a state ϕ is accessible if it is either directly accessible or if another state is directly accessible that leads to ϕ. Following von Wright, we define a "good" state ϕ as a state that agent i prefers to its negation, and conversely for a bad state:

$$\mathbf{GO}_i(\phi) \overset{\text{def}}{\Longleftrightarrow} (\phi\mathbf{P}_i\neg\phi) \qquad \mathbf{BA}_i(\phi) \overset{\text{def}}{\Longleftrightarrow} (\neg\phi\mathbf{P}_i\phi)$$

5 Goals

ALX allows for the definition of various goal-operators. Based on the notion of preferences, our operators eschews the counter-intuitive behavior that primitive goal-operators tend to exhibit (e.g., necessitation and closure under implication). We present four definitions in this section.

Agents might opt for a state simply because it is better than its negation, particularly if only few alternatives are considered. A "good" goal can be defined by using the "good" operator **GO**. Let $\mathbf{G}_i^g\phi$ denote the fact that ϕ is a good goal of i. Define: $\mathbf{G}_i^g\phi \overset{\text{def}}{\Longleftrightarrow} \mathbf{GO}_i\phi$. Thus a good goal is a situation that is preferred to its negation.

The second definition involves a satisficing goal. "Satisficing" has been an important procedural procedural addendum to the conceptualization of bounded rationality. Whereas the declarative part of bounded rationality concerns incomplete knowledge, the procedural part concerns the question what to do when the knowledge is not complete enough. So, the definition should involve the action of satisficing. Let $\mathbf{S}_i\phi$ stand for an arbitrary situation ϕ satisficing agent i and relax the definition of action terms by allowing for mnemonic expressions: $\mathbf{S}_i\phi \Leftrightarrow \langle \textit{satisficing-search}\rangle\phi$.

Note that this definition does not exclude the possibility that the search action is void in cases that the satisficing solution is already at hand. Define a satisficing goal in terms of a satisficing state. Let $\mathbf{G}_i^s\phi$ denote the fact that ϕ is a satisficing goal for i. Define: $\mathbf{G}_i^s\phi \overset{\text{def}}{\Longleftrightarrow} \mathbf{S}_i\phi$.

Agents may try to maximize, or even optimize, if the context supports the search for extremal values. Both attitudes involve some total view of available alternative and thus require quantification over situations. In the general case, this quantification would push us beyond the syntactic boundaries of ALX3, and undermine ALX's completeness, since the set of situations need not be countable. However, we can avoid this complication via some technical means, at least as far as preference formulas are involved (See [Hua94]). Define an element of i's preference order as follows:

$$\mathbf{PO}_i\phi \Leftrightarrow \phi\boldsymbol{P}_i\psi \vee \psi\boldsymbol{P}_i\phi$$

Whether a solution is maximal or optimal depends, of course, on the structure of the preference order of an agent. If it is partial, but not total, the order may contain several maximal, incomparable elements. If, furthermore, more than one maximal element is accessible, then an optimal goal in the sense of a best overall solution cannot be defined. Conversely, if the order is total, then an optimal goal can be identified (under the assumption that at least one situation occurring in the order is accessible). Alternatively, a partial order may give rise to an optimal goal if only one maximal situation is accessible. In the light of this reasoning, we define a "best choice" as a maximal goal and specify the conditions under which such a best choice may, in fact, be optimal. Let $\mathbf{G}_i^{ma}\phi$ denote the fact that ϕ is a maximal goal. Define: $\mathbf{G}_i^{ma}\phi \overset{\text{def}}{\Longleftrightarrow} \mathbf{PO}_i\phi \wedge \forall\chi(\chi\mathbf{P}_i\phi \rightarrow \neg\mathbf{A}_i\chi)$.

A best choice is an accessible situation to which no other accessible situation is preferred. A best choice ϕ is optimal, if ϕ is unique. Let $\mathbf{G}_i^{op}\phi$ denote the fact that ϕ is optimal. Define: $\mathbf{G}_i^{o}\phi \overset{\text{def}}{\Longleftrightarrow} \mathbf{G}_i^{bc} \wedge \forall\psi((\mathbf{G}_i^{bc}\psi) \rightarrow (\phi \leftrightarrow \psi))$

Ironically, best choices need not be good nor satisficing. In a tight spot, an agent's best alternative might simply be the best among dubious alternatives.

6 Conclusions and Summary

ALX is the first action logic modeled on the decision cycle of potentially rational agents. Perhaps its most important feature is its preference operator [HMP92]. The preference operator has a closest-world semantics in combination with the conjunction expansion principle, so agents prefer ϕ to ψ if they prefer the closest ϕ-and-not-ψ worlds to the closest ψ- and-not-ϕ worlds. This facilitates the representation of situation-dependent preferences, but does not exclude the representation of stable preferences. The preference operator is "normal" in the sense that agents cannot have preferences with respect to absurd worlds. This normality protects the conjunction expansion principle against counterintuitive utilizations [CS66]. On the other hand, the preference operator is not "tarskian", i.e., it does not distribute over disjunction [Mar95] and this protects all intensional operators built on the preference operator against the necessitation rule and against undesired closure properties. For example, goal operators can be defined as preferred states subject to additional qualifications (e.g., the best accessible state, the best state not believed to be inaccessible). The preference operator is transitive, but a non-transitive version is easily generated by removing the constraint (TRAN) on the semantic preference relation. The conditional operator is adapted from Stalnaker's system but allows for non-unique closest worlds. It allows for an easy representation of the notion of "ceteris paribus", and hence for the standard notion of causality. This, in turn, greatly facilitates the representation of causal effects, side effects, and similar non-monotonic relations that would have otherwise to be represented by the (monotonic) material conditional.

An important property of ALX is the virtual absence of interaction between modal operators. This property may raise eyebrows in philosophical circles, but we designed ALX as a flexible knowledge-representation tool, and such a tool should, in our view, not preempt the structure of domains to be represented. As a flexible tool, ALX3 allows for the definition of various additional intensional operators, such as the alethic modalities, goal operators, intention-operators, and for the characterization of other action-related notions, such as ability, effect, side effect, intended effect, etc [HM94].

As demonstrated in a companion paper [MH94], ALX3 serves already as a versatile tool of knowledge representation. However, there are some desiderata left to be satisfied. ALX3 has no explicit notion of time in its semantics, and this complicates a direct representation of events. For example, we cannot define a "do"-operator, and hence no actions that are not deliberate. Second, we think that the notion of closest worlds needs closer inspection. The constraints on the closest world function are relatively weak; stronger constraints may be required, in particular if one wants to combine the notion of action with the notion of the closest world. Third, and related, one may want to strengthen ALX so that it allows for a *calculation* of causal outcomes. Such a calculation would be an answer to the frame problem, but it requires a more specific notion of possible worlds.

References

[Blu69] Blumer, H., *Symbolic Interactionism: Perspective and Methods*, (Englewood Cliffs, NJ, Prentice-Hall, 1969).

[CS66] Chisholm, R., and Sosa, E., Intrinsic preferability and the problem of supererogation, *Synthese* **16** (1966), 321-331.

[CL90] Cohen, P. R. and Levesque, H. J., Intention is choice with commitment. *Artificial Intelligence* **42** (1990) 213-261.

[Gam90] Gamut, L.T.F., *Logic, Language, and Meaning*, (The University of Chicago Press, 1991).

[Gid79] Giddens, A., *Central Problems in Social Theory: Action, Structures, and Contradiction in Social Analysis*, (Berkeley, CA, University of California Press, 1979).

[GS87] Ginsberg, M., and Smith, D., Reasoning about action I: a possible worlds approach, in: *Readings in Non-monotonic Reasoning*, (Morgan Kaufman, 1987).

[Har84] Harel, D., Dynamic logic, in: *Handbook of Philosophical Logic*, Vol.II, (D. Reidel Publishing Company, 1984) 497-604.

[HMP92] Huang, Z., Masuch, M., and Pólos, L., Een preference-logica voor rationele handelingen, in: H. de Swaan Arons, et al. (eds.), *NAIC'92 Conferentie Proceedings*, 17-28.

[HMP95] Huang, Z., Masuch, M., and Pólos, L., ALX, an action logic for agents with bounded rationality, *Artificial Intelligence* (forthcoming).

[HMP93] Huang, Z., Masuch, M., and Pólos, L., ALX2, a quantifier ALX logic, CCSOM Working Paper 93-99.

[Hua94] Huang, Z., *Logics for Agents with Bounded Rationality*, ILLC Dissertation series 1994-10, University of Amsterdam, (1994).

[HM94] Huang, Z., and Masuch, M., ALX3, a Multi-agent Action Logic, CCSOM Working Paper 94-102.

[MH94] Masuch, M., and Huang, Z., A Logical Deconstruction: Formalizing J.D. Thompson's *Organizations in Action* in a Multi-agent Action Logic, CCSOM Working Paper 94-120.

[Mar95] Marx, M., *Algebraic Relativization and Arrow Logic*, ILLC Dissertation series 1995-3, University of Amsterdam, 1995.

[Par37] Parsons, T., *The Structure of Social Action*, (Glencoe, IL, Free Press, 1937).

[RG91] Rao, A. and Georgeff, M., Modeling rational agents within a BDI- architecture, in: *Proceedings of the 1991 KR Conference*, (Morgan Kaufmann Publishers, 1991) 473-484.

[Sta68] Stalnaker, R., A theory of conditionals, in: *Studies in Logical Theory, American Philosophical Quarterly* **2** (1968) 98-122

[Sim55] Simon, H. A., A behavioral model of rational choice, *Quarterly Journal of Economics* **69** (1955) 99-118.

[vWr63] von Wright, G. H., *The Logic of Preference*, (Edinburgh, 1963).

What Must, Ought, Is Worth and May Not Be Done

Hamish Taylor
Department of Computing and Electrical Engineering
Heriot-Watt University

Abstract

The Praxis model of reasons for action takes choice among alternative courses of action to be a matter of what must, ought, is worth and may not be done. It envisages practical decision making systems being given their reasons for action as requirements, duties, rules of valuation, proscriptions and contingent circumstances in a knowledge base. The model's principles of selection and automated deduction then elicit from them what is to be done. Starting from local circumstances and rules specifying what must or ought to be done in various eventualities, initial options for consideration are deduced. Means end reasoning refines them into practicable alternatives. Alternatives to actions which must be done or actions entailing proscribed activity are excluded. Remaining alternatives are classified along incommensurable dimensions of value specified by relevance principles to bear on the decision context. Choice among alternatives is made using selection rules like the dominance principle. It states that alternatives that are more valuable in one way and no less valuable in any other are better ones. If one is better than any other, it is selected. If not then recourse is made to default selection rules. They provide reasonable ways for resolving conflicting rankings of alternatives along incommensurable value dimensions, for discriminating among indifferent alternatives, or for weakening conflicting requirements so that forbidden options become choosable.

Keywords: reason, practice, value, duty, proscription, axiology, deontology

1. Introduction

A rich variety of concepts and principles are open for use in reasoning about what to do. They provide more than one possible basis for developing a useful and valid model of practical decision making. Whereas Bayesian decison theory [6,8] begins with the relative desirability of consequences of actions open to an individual and his partial belief in such consequences happening upon performing these actions, this paper starts with the requirements, duties and proscriptions governing acting and the values yielded in performing alternatives, and develops a different kind of model of practical reasoning.

The Praxis model is intended for use in knowledge based approaches to automating practical reasoning. Reasons for decison making about what to do will be specified in the knowledge base in a propositional form, and depending upon their category serve as requirements (musts), duties (oughts), proscriptions (may nots), value relevance principles, canons of valuation assessment for kinds of action, or default rules for resolving what is to be done when no answer is obtained using the basic option selection process. Updating the knowledge base with contingent circumstances and deducing consequences from them and the stored reasons for acting, according to the principles of the Praxis model, will answer the question of what is to be done.

The paper begins by outlining the Praxis model. It then contrasts the model with Bayesian decision theory. After that it develops the notions of requirement, duty, proscription and value in more detail to show how they can serve in an alternative model of practical decision making. It ends with some brief remarks on resolving blocked choices.

2. The Praxis Model

The Praxis model takes the basic question of practical reasoning to be

What is to be done?

Tactical, strategic and methodological questions about how it is to be done involving

agency which agent(s) is to do it and how is answering the basic question to cause that

control how is doing it to be steered and managed

methods how are techniques and procedures to be employed in doing it

instrumentality how are tools, aids and settings to be employed in doing it

scheduling when and in what order are means and opportunity to be exploited in doing it

are beyond the scope of this paper, as are interaction questions concerning how to act in contexts where hostile or cooperative agencies are present and may intervene. Such questions are particularly complex where communication in the form of threatening, bargaining or reasoning with such agencies is possible. Also beyond this paper's scope are questions of how exigencies of practical decision making like limited time, resources, and knowledge to deliberate with, and limited agency to enact decisions with, affect deciding what is to be done. This paper leaves these questions to one side. It addresses the issue of what is to be done in terms of reasons for and against acting as four separate questions:

Question	*Type of Reason*	*Study*
What must be done?	requirement	Deontology
What ought to be done?	duty	Praxeology
What may not be done?	proscription	Deontology
What is worth being done?	value	Axiology

The decision circumstances together with general requirements specifying what must be done in various eventualities may imply that some actions must be done now. More probably the same circumstances and general praxeological principles expressing duties, needs and interests of the decision maker only imply what ought to be done now. Both requirements and duties can arise out of a role (e.g. professional expert or assistant) or be the raison d'etre of an automaton (a Mars exploration robot) or be duties attached to the purpose of a decision support system (advising about financial affairs). Needs and interests arise out of what it means for the decision maker to have itself as an agent to enact some of its decisions. Needs and interests specify what ought to be done to maintain that agency's well-being, power, and potential to do things.

Whereas requirements exclude their alternatives, these duties, needs and interests are likely to generate conflicting options for the decision maker to consider in deciding what is to be done. These options are then confronted with contingent circumstances in means end reasoning, in ways not discussed in this paper, to elicit practicable courses of action, and then regimented to form a set of alternative things to be done. The identification of alternatives enables courses of action which must be done to exclude their alternatives.

The third question of what may not be done is now addressed to ensure that alternatives involving forbidden actions are also excluded from consideration. The role of these considerations is to set limits on certain possibilities of practical decision making. While some practical decision making might not be subject to any requirements or proscriptions, critical matters such as security and safety invite decision making in terms of what must be done and what may not be done in those circumstances.

The Praxis model offers two variations to cope with conflicts among requirements and/or proscriptions. The hard Praxis model imposes an integrity constraint on acceptable sets of requirements and proscriptions which precludes the deducibility for any circumstance of conflicting musts or may nots. This may be achieved by global consistency checking or by prioritising the bearing of conflicting requirements. The soft Praxis model imposes no such integrity constraint but handles conflicts between rival mandatory alternatives or between a mandatory course of action and a proscription of an action entailed by it by equitably downgrading the force of relevant requirements to lesser force ought to be dones, and the force of proscriptions to attaching an apt strong negative value to performing the action.

In the Praxis model if only one thing must be done, then that thing answers the question of what is to be done. If more than one thing may be done, then the fourth question of what is worth being done is addressed. Principles of relevant values specifying which kinds of values are pertinent to the decision context are applied and the remaining alternatives are evaluated using canons of valuation along each such dimension of value. Alternatives are then compared using evaluative principles. They specify sufficient conditions over the values of alternatives A and B under which

A is a better alternative than B

These evaluative principles order alternatives partially or totally. If they order them totally - i.e. there is one alternative which is better than any other, it becomes the alternative which is to be done. If there is no single course of action which is better than every other alternative, then the Praxis model allows default selection rules to be applied in turn over the elicited alternatives to try to produce a totally ordered ranking of alternatives. Various such principles are proposed in a later section. None of them can be defended as a basis for making a best choice, but each has some merits as a way of making a reasonable choice among alternatives. They are applied independently or iteratively until the alternatives are totally ordered in merit, or value indifference is sufficiently established among a set of alternatives, all of whose members are better than all non-members. In the first case a single answer is supplied to the question of what is to be done, and in the latter case any member of the indifference set is an answer to the question.

3. Bayesian Decision Theory

The Bayesian approach to normative decision theory [6,7,8,12] analyses practical decision-making as follows. A decision is a choice among a set of pairwise exclusive and exhaustive alternative things to do.

$$A_1, \ldots, A_n$$

Each alternative A_j has an exhaustive and exclusive set of consequences

$$C_1, \ldots, C_m$$

Each consequence C_i has a probability Pi between 0 and 1 of coming about if A_j is done, and that consequence has a finitely valued utility U_i to an agent *I* of happening. The merit of choosing an alternative A_j to the agent *I* is the sum of the expected utility E_j of all its consequences:

$$E_j = \sum_{i=1}^{m} P_i \times U_i$$

The Bayesian choice rule is to choose a course of action Ak which has a maximal expected utility

$$E_k = \max\{E_1, \ldots, E_n\}$$

The possibility of ties means this choice may not be unique.

The Bayesian model embodies four striking characteristics

- consequentialism
- relativism to alternatives
- value monism
- synthetic subjectivism

Alternatives are only evaluated in terms of their consequences. They are not valued in terms of the merit of the action itself. Furthermore choice is always relative to alternatives. Whether an alternative gets chosen or rejected depends only on whether its expected utility is more than any other or less than another alternative. It never depends only on the reasons relating to the alternative itself. A third characteristic is that only one value utility is used to assess the merits of an alternative's consequences. Multiple dimensions of worth for assessing whether to choose an alternative are not recognised. Furthermore this value is not one belonging to states of affairs or things but is synthesised from individual preferences using betting measures [7] or other means [6]. Thus the approach is subjectivist and delivers conclusions which are only valid for an individual with those subjective probability and utility assignments. By contrast the Praxis model embraces

- non-consequentialism
- singularism about alternatives
- value pluralism
- naturalistic intersubjectivism

The Praxis model allows alternatives to have reasons for doing them which are not stated in terms of their consequences. Doing that alternative may be a requirement or duty of a role, or be valued like courage or integrity for the sake of its own performance. The Praxis model also allows alternatives to be chosen or rejected solely by their attached reasons irrespective of alternatives. An alternative which must be done or one which may not be done can be chosen or rejected on its own merits irrespective of what else should or must be done. The Praxis model is also based on accepting that there are many different values, which cannot be reduced or measured by a single one. This ineradicable value pluralism creates the problem of resolving conflicting valuations among incommensurable values for reasoned choice. The Praxis model encompasses methods for its reasonable management and views doing so as central to most interesting practical decision-making. Lastly the Praxis model endorses a naturalistic intersubjectivism. It treats value ascriptions as claims about the nature of things and states of affairs. It is neutral between supposing possession of a value is an objective property of a thing and supposing that it is intersubjectively constituted within a normative framework whose scope of validity is conditioned by socio-cultural, historical and anthropological facts. This stance empowers the Praxis model to deliver conclusions with intersubjective validity about what is to be done.

4. Reasons for Acting

Natural language expresses possibilities using a family of modal terms. They include "must", "ought", "may", "might", "can", "could", "should", "has to", "is necessary" and "is possible". These terms express different kinds and strengths of possibility [4] appendix B. Three strengths of these modalities can be distinguished.

strong terms:	*must, has to* and *is necessary*
intermediate terms:	*ought* and *should*
weak terms	*may, might, can* and *could*

Thus the propositions

Kate must eat her pudding

Kate ought to eat her pudding

Kate may eat her pudding

express a progressively weaker set of propositions regarding what Kate has reason to do. The different strengths provide the Praxis model with different categories of reasons which constrain practical decision making in quite different ways.

5. Deontology

The strong form "must" and the negated weak form "may not" provide an intuitive idiom for expressing necessities for acting. It is natural to seek to formulate binding instructions which practical decision makers must obey and to specify courses of action which they may not perform. Thus codes of conduct are formulated for organisations like the British civil service, which specify what civil servants must or may not do. They cover matters like forbidding receiving gifts from persons with whom civil servants conduct their official business. In a similar way it is natural to seek to specify binding instructions for machines. The science fiction writer Asimov makes a good case for doing this with his three Laws of Robotics [2]

First Law A robot *may not* injure a human being, or, through inaction, *allow* a human being to come to harm.

Second Law A robot *must* obey orders given it by human beings except where such orders would conflict with the First Law.

Third Law A robot *must* protect its own existence as long as such protection does not conflict with the First or Second Laws.

While it is possible to cavil at Asimov's formulations of his Laws, Asimov is persuasive in arguing that robots (and decision support systems) need to be subject to such binding restrictions which admit of no exceptions. If the decision making of machines is to be trusted, absolute guarantees are needed that certain matters such as allowing injury to human beings or letting itself be destroyed, won't get permitted in some complex trade off of reasons for and against certain courses of action. Although it is somewhat ironic that many of Asimov's pieces of robot fiction turn on just those exceptional cases where violations of these Laws would make a certain sense, this argument still seems persuasive. By giving such considerations overriding priority in a sound deductive framework, it should be possible to frame safety constraints for practical decision making by machines and prove that such machines will never knowingly violate them. This would be useful in automated reasoning applications like control of vital monitoring equipment or dangerous engineering plant where maintenance of safety is a critical issue.

Right away Bayesian Decision Theory becomes an inept vehicle for articulating the application of such practical precepts to guide action. At heart it is a mechanism for trading off any course of action against any alternative depending on their relative costs and benefits. No course of action is deemed forbidden or mandatory irrespective of alternatives. So long as all utilities of consequences are finitely valued, it remains possible that the expected utility of any action may be higher than any of its alternatives.

A more apt vehicle for expressing considerations like Asimov's three laws is Deontology. Deontology is concerned with what must, may or may not be done. Deontology is often misleadingly described as the study of duty. This conflation of duty with what one must do has been obscured by the idea that duty is somehow compelling. However, it is not plausible to suppose that duties are always compelling, in the sense of overriding other considerations for acting. Intermediate strength terms in the deontic modality like "ought" are more apt for expressing the claims of duty than terms like "must".

5.1 Deontic Logic

Deontic logic [13] attempts to capture the properties of what must or may not be done. It uses a modal operator **P** which qualifies propositions.

P p *means* it is permissible that p

Using Davidson's method for rendering action sentences [3] which gives them an extensional representation by introducing an extra argument for the action's event, the first part of Asimov's First Law of robotics can be rendered as follows

$\neg$**P** (all event e) (all thing x) (all person y) (robot(x) $\supset$ injure(x, y, e))

The relation between what is mandatory and what is permissible is intuitively captured by the idea that something must be done if and only if it is not permissible that it may not be done.

Thus Asimov's Third Law of robotics without its exclusion clause is

$$\neg\mathbf{P}\,\neg(\text{all thing } x)\ (\text{all event } e)\ (\ \text{robot}(x) \supset \text{protect}(x, x, e)\ )$$

Standard formulations of deontic logic regard

$$\neg\mathbf{P}\,\neg\, p \,\&\, \neg\mathbf{P}\, p$$

as a contradiction. The significance of this for the pragmatics of using this logic to articulate reasoning with permissions, is that the set of what is mandatory and what is forbidden must be formulated to be free of contradictions. This approach is adopted by the hard variation of the Praxis model. However, it is a tough requirement to meet without introducing ad hoc measures.

5.2 Conflicting Demands

Suppose that a Mars explorer robot has the following mandatory requirement in its knowledge base.

> A report must be made to the command base on Earth every day.

and the following proscription in its knowledge base.

> The backup generator may not be used to power the main radio transmitter.

The latter we can suppose has been put in to avoid the chance of damage being caused to the backup generator by the heavy power drain of using the main transmitter. Now it may happen that means end reasoning determines that, given the current state of the primary generator, the only way to power the main radio transmitter to make the daily report to the command base on Earth is by using the backup generator. Furthermore means end reasoning may determine that the daily report can only be made to the command base on Earth by using the main transmitter. The Mars explorer robot is now in the impossible situation of being subject to contradictory demands to use and not to use the backup generator.

Asimov addresses this kind of problem by prioritising the application of his Laws of Robotics. Undoubtedly this method could be made to work for the case above by attaching priorities to requirements and proscriptions. However, the worry is that such a method is blindly powerful. It is too easy to prioritise all musts and may nots, but it is much harder to be sure that such prioritisation will handle conflicts in a sensible way.

A different approach tries to dissolve the incompatibility by equitably downgrading inconsistent demands that generate the conflict for this local part of the decision making process. This is the approach of the soft variation of the Praxis model. The requirement that a report must be made to the command base on earth can be downgraded to a duty that a report ought to be made. At the same time the proscription that the backup generator may not be used to power the main transmitter can be waived and the force of its injunction transmuted into a relevant valuation which would give strong relevant disvalue to performing that action.

It can be argued that allowing requirements and proscriptions to be downgraded in this way undermines the point of having them at all. If a requirement or a proscription can be waived then it fails to fulfill its function of necessitating performance or non-performance of a course of action. In defence of the soft Praxis model it can be replied that waiving a requirement or proscription does not preclude the nature of the consideration from having considerable force in determining what is to be done. Furthermore such considerations are only waived when there are contradictory necessities governing what is to be done. Necessities are not discarded lightly on the soft Praxis model.

Neither the hard nor the soft Praxis model adopts an ideal solution. Each has its merits and demerits. It would be invidious to prefer one solution to the other. So both are countenanced as acceptable variations of one model.

6. Praxeology

Whereas the strong modality of requirements and proscriptions expresses some useful necessities of acting, the degree of its own modality renders it unable to articulate genuinely conflicting reasons for acting. Intermediate modal terms such as "ought" offer a better idiom for articulating and dealing with conflicting reasons for acting. Because "ought" is a term of intermediate modal strength, there is no contradiction in saying of someone that he ought to do *F* and ought not to do *F*. The apparent conflict can be explained away by noting that each use of "ought" will be relative to a different reason. By contrast there would be a contradiction in using the strong modal term "must" and saying of someone that he must do *F* and must not do *F*. The point of using the strong modal term "must" is to exclude the possibility of his not doing *F*. No relativisation to reasons saves the statement from contradiction.

Two different forms of relativisation to reasons by "ought" can be distinguished:

prima facie	relative to a reason;
all things	considered relative to all relevant reasons.

Although a person may have several alternative things he ought prima facie to do, he can have only one thing he ought to do all things considered. Ought propositions can also be qualified with regard to a set of considerations. Thus the forms ought morally, ought legally and ought as a matter of etiquette can be distinguished. Sometimes the qualification morally to a use of ought is intended but is not made explicit. This can be detected by asking whether in the decision context what ought to be done is what ought to be done morally or what ought to be done all things considered.

Another relevant distinction is that between what ought to happen or be the case and what ought to be done. Although something ought to happen or be the case, it does not follow that it ought to be done. What ought to happen may not be open to being achieved by action, or it may only be open to being achieved by collective forms of action which are not open to any single agency (collective or otherwise) to effect. This point is really underpinned by the principle

P1 What ought to be done must be within the power of some relevant agency.

This idea is often expressed by the slogan "ought implies can". If what ought to happen is not within any agency's power to effect, then it is not the case that it ought to be done.

It seems plausible to suppose that practical reasoning conclusions about what ought to be done, can only be validly derived from premisses containing a praxeological principle about what ought to be done. Such conclusions cannot be validly derived from premisses containing no ought premisses at all. This principle is often summed up in the slogan "*an ought cannot be derived from an is*". Hume is usually credited with authorship of this useful principle in his *Treatise of Human Nature* Book III, part I, section I [5] p.469.

This principle is useful as a heuristic in searching for suppressed premisses in practical reasoning. In a rule such as

```
IF      patient P has an inflammatory joint disease &
        oral drugs are suitable for patient P &
        patient P is not suspected of having a peptic ulcer
THEN    patient P ought to take aspirin orally
```

it suggests that a reason why the patient ought to do something has been suppressed, and should be brought out. After all the patient's taking aspirin is only a means to something else, his regaining his health. So that should be brought out explicitly. In this way the principle motivates the development of a model of practical reasoning where all of an agent's ultimate "ought" reasons for acting are made manifest as a set of praxeological principles for acting. In a natural sense some of these might be called the duties of an agent.

With human beings these can arise from an agent's roles, such as father, husband, friend, official, citizen etc., or they can be duties of codes of conduct which govern the agent's behaviour, such as ethics, local custom, honour code, religious convention etc.. With an automated practical reasoning system duties will arise from its raison d'etre, what its job is. Needs and interests of an agent will also serve as a source of possible actions which an agent ought to perform.

A corollary of the is-ought principle is that means end reasoning propagates oughts from ends to means. Clearly a doctor ought to treat patients of his who consult him. Furthermore if such a patient upon consulting his doctor is found to need to be prescribed a drug, then that doctor ought prima facie to prescribe his patient that drug. Here the "ought" attaching to the doctor's duty has by means end reasoning been propagated to the means for accomplishing the end.

Agents can have both positive and negative duties. Two such duties of a doctor might be:

> A doctor ought to inform the authorities upon discovery that a patient of his has a notifiable disease.
>
> A doctor ought not to disclose details of a patient's medical history to unauthorised persons.

In the Praxis model of practical reasoning positive duties provide broad kinds of things for an agent to consider whether they are to be done. However, negative duties cannot do the same thing. They count against performing certain kinds of action. In order to be brought into play, an action of the type they count against has to be licensed by a positive duty. Hence negative duties are really either constraints on duty or part of the evaluation of duties and not part of the generation of alternatives for consideration. Thus a negative duty really has to be represented either as something forbidden in the proscriptions part of the Praxis model or as having negative value in the valuation part of the model, and not in the duty part we are considering here. A complication, which we shall not discuss, is that it is not always easy to distinguish positive from negative duties. A positive duty to keep a promise might also be represented to be a negative duty not to break a promise.

The properties of the relevant form of ought can now be summarised as follows. Its logical form is

ought(Agent, Deed, Reason)

An ought of reasons is relative to an agent, a deed, and a reason. Agents are usually individuals but can also sometimes be collective entities like organisations, teams etc.. Deeds are actions of the agent. Reason is a single or a set of reasons for performing the deed. Some relate the Deed as means to a further end in acting. Others give a description of the type of end that the Deed is. Such oughts are usually applied by conditions, which determine the circumstances, context and time, under which the reason applies to the agent. The following conjunction relating an agent A and a deed D is not a contradiction.

$$(\exists\, R1)(\exists\, R2)\ \text{ought}(A, D, R1)\ \&\ \neg\text{ought}(A, D, R2)$$

Two properties observed by "ought" are:

$$\text{ought(agent, deed, reason)} \supset \text{can\_do(agent, deed)}$$

$$\text{ought(He,Deed,Reason)}\ \&\ \text{means(He,Act,Deed)} \supset (\exists\, R)\ \text{ought(He,Act,R)}\ \&\ R = \text{in\_order\_to(Deed,Reason)}$$

The first implies the Deed is within the agent's general power to perform. The second states that that if an Act is a means for an agent to perform a Deed and He should do that Deed for a Reason, then He should perform that Act in order to do that Deed for that Reason.

7. Value Theory

Practical decision making is not only involved with doing what must be done, avoiding what may not be done and doing as much of what ought to be done as can be done. It also involves selecting among alternative things which ought to be done. This involves choosing better alternatives over worse alternatives on the basis of their relative merits. In an impersonal model of practical reasons, this means in terms of their relative worths. Thus the Praxis model requires the means to assess the value of alternatives and judge their respective worths. To get this we must turn to value theory [9 chap 5, 10].

A value can be characterised as a beneficial quality of an action or a state of affairs, which would give a reason inter alia for performing it or for bringing such a state of affairs about. Values are values because their attainment and maintenance contributes to well-being and the good life. However, they are not subjectively constituted by individual attitudes to such matters. They have an interpersonal standing, and in this sense are impersonal [10] p. 11.

Trying to incorporate into the Praxis model a mechanism to adjust the worth of a value by whether and how much it is valued by a particular individual would be to abandon the impersonal standpoint. It would render all adjusted valuations idiosyncratic. The Praxis model is better off remaining as impartial as it may, so that its valuations remain grounded in a contemporary framework of evaluation which has as much intersubjectively constituted authority as it can get. Careful choice of apt value concepts should ensure that the right standards of valuation get applied to appropriate decision contexts.

Values can be either instrumental or intrinsic. Instrumental values are valuable because they contribute to the realisation of other values, whereas intrinsic values are valuable in their own right. Intelligence would be an instrumental value in making a good employee whereas truth would be an intrinsic value in theoretical inquiry. Values can be classified in various ways [10]

Classification Dimension	*Examples*
subscribers	personal, professional, or national values
domain of application	bravery in humans, purity in things, justice in societies
nature of benefit	economic, aesthetic, or moral values
purposes at issue	exchange, food, or persuasive value
subscriber-beneficiary relation	egocentric, disinterested values

Values can have either an ordinal value scale

unhappy equable happy

or a cardinal value scale like cost (£1000s). These value scales can be bounded or unbounded. Most values are bounded in having limiting degrees on their scale at both ends. Thus there is no greater degree of integrity than full integrity and no lesser degree of lack of integrity than complete lack of integrity. However, some values like cost are unbounded in having no limit to their scale at their disvaluable end. Some people have believed that the dimension of good and evil is unbounded at both ends. Namely, that however evil or good a state of affairs is, more evil or more good states of affairs might exist than it.

Values can be monopolar in the sense of having only a single dimension of worth, or be bipolar and have both a positive dimension of value and a countervailing negative dimension of disvalue. Bipolar values have a median or neutral point on their scale which is the possession of no degree of disvalue nor of value. Grace is best construed as monopolar in having the following scale

graceless partly graceful graceful

whereas beauty is a bipolar value

ugly plain beautiful

Ordinal scales of value can usually be partitioned in coarser and finer fashions. Thus the scale of beauty above might be refined into a finer scale of gradations as follows

very ugly ugly plain beautiful very beautiful

Important concepts in generating bipolar dimensions of value are good and bad. The attributive form "a good X" expresses the idea that the X is valuable for use as or in the role of an X. The related form "a bad X" expresses the polar opposite. These concepts are useful for creating concepts of (dis)value which sum up various components aspects of value. The scale

bad knife indifferent knife good knife

sums up various instrumental values like robustness, sharpness, easiness to wield etc..

The dimensional nature of value is underpinned by the following principle

P2 Something is a value only if other things being equal it is more valuable to exhibit or realise more of it and less valuable to exhibit or realise less of it.

This dimensionality of value principle is important for screening putative values to see whether they are really values. The principle shows that a parameter like a house's size is not an instrumental value in making a good house. While greater size in a house often makes it more valuable other things being equal, there is also such a thing as a house being too big to be a good house. A good house should be of an apt size. Aptness of size not size is the pertinent instrumental value. Other things being equal a house of a more apt size is always a better house and a house of a less apt size is always a worse house.

Flexibility in value gradations shows its usefulness, because choice can and should be more or less sensitive to matters of degree along dimensions of value. An employer choosing a front office receptionist should recognise that only whether a prospective employee is of good appearance or not is germane to the choice and that any more discriminating interest in exactly how attractive he or she is is inept. So long as the prospective employee is at least of good appearance that is all that matters with respect to that scale of worth. However when considering diligence, the employer should be more discriminating and be interested in making reasonably fine distinctions between degrees of diligence which would distinguish between being moderately, quite, rather, very and extremely diligent.

Values are related in various ways to action. Actions can manifest, create, destroy and maintain the existence of values [9] chapter 5. An important distinction here is that between performance values, state values and production values.

performance value	gracefulness, truthfulness, courage, reasonableness
state value	happiness, welfare, justice, truth

Performance values are transiently manifested during action. State values belong to objects, events or states of affairs. Some values like beauty can be both performance and state values. They can be transiently manifested during performances such as dancing and belong to objects in the world such as a statue. A third kind of value is the ability of an action to produce a state value. This kind of value of the action will be called its production value. The Praxis model is concerned with both the performance and the production values of alternative courses of action. By contrast Bayesian decision theory focuses on production values and ignores performance values.

A key feature of value is that there is an essential plurality of values. There is no plausible way of accounting for the worth of each value using some single category of value. There may be ways in which values can be related to each other. Thus instrumental values can be related to intrinsic values and intrinsic values may complement each other in organic unities [9] chapter 5. Also some partial rankings of values look plausible e.g. harmfulness to humans rates as more important than gracefulness. But even these value rankings don't determine how much gracefulness is outweighed by how much harmfulness. The existence of a plurality of values has far reaching significance. It means that the question of which of two alternatives is the better one often has no right answer. A course of action A may exhibit or realise more of the value V and less of the value W. An alternative B may exhibit or realise more of the value W and less of the value V. Thus the following relationships may hold

$$V(a) >_V V(b)$$

$$W(b) >_W W(a)$$

asserting that A's V value is greater than B's V value, but that B's W value is greater than A's W value. However, the different subscripts of > show that different ordering relations are in question. If V and W are distinct values, there is usually no way in which the differences between these values (if they are definable)

$$U(A\text{-}B) = V(A) - V(B)$$
$$U'(A\text{-}B) = W(B) - W(A)$$

can be rendered in some common value U and related by an ordering relation "$>_u$" in a fashion such as

$$U(A\text{-}B) >_u U'(A\text{-}B)$$

Hence, there is no way that the overall relative value of A to B can be determined. Frames of consideration of practical decision making which exhibit this kind of structure will be said to exhibit a conflict of values. They are encountered very often. However, they do not render reasoned choice intractable as will be seen.

However, not all practical decision making encounters conflicts of values. In many decision contexts the existence of a plurality of values can be admitted and yet a better choice be identified. The key to doing this is the dominance principle that better alternatives exhibit or realise more value then worse ones.

P3 If a course of action A is worth more in at least one relevant respect of value and is not worth less in any other relevant respect of value than an alternative course of action B then A is a better alternative than B.

This principle does not say that either alternative should be chosen. It only identifies the better alternative. The existence of a plurality of values requires it to be framed in this form, because there is no general way for measuring the relative worth of one grade of value in terms of another (c.f. Richards' teleological dominance principle in [11]).

7.1 Example of Evaluation

All significant relative appraisal involves a significant degree of interpretation. In the Praxis model this will proceed along several different dimensions of value. Relevance principles will clarify which values are relevant and at what level of discrimination, and a process of appraisal will be needed to transform the manifest parameters in which the choices present themselves into the relevant categories and grades of value and disvalue. It is not to be expected that there will be a precise theory of value measurement to govern how this should be done.

When choosing to buy a house to live in, options can be considered in terms of how big the house is, how near various main roads and railway stations it is, what the house and its surrounds look like, what its views are, how many bedrooms it has, what its cellars, kitchen and living rooms are like, whether there is anything wrong with it structurally, how much it costs etc.. These factors can be assessed by inspection or from a surveyor's report. Evaluation now requires that these parameters be translated into grades on relevant scales of value.

Praxeological and means end reasoning might deliver three alternative options in a house buying choice

Ashgrove ought to be bought.

Bellevue ought to be bought.

Cornhall ought to be bought.

Features of these three options might be

House Name	Bedrooms	Transport Links	Situation	Modern Fittings	Survey Report	Price
Ashgrove	5	1km to station	countryside	all mod cons	OK	£88K
Bellevue	4	5km to M25	leafy suburbs	old but OK	excellent	£78K
Cornhall	5	3km to station	terrace house	all mod cons	minor damp	£82K

In order to render these into the Praxis model, it is necessary to determine what the relevant values are, and what their grading scales are. The following appraisal values are plausible scales for assessing the merits of these three choices

size aptness	how well the house size fits the need
accessibility	how well the house has good transport links
amenity	how well the house's situation conduces to living there
utility	how well the rooms and layout serve the requirements
soundness	how structurally sound the building is
cost	how much the purchase of the building and grounds is

Thus the model will need to contain a relevant value principle such as the following:

> Relevant values for choosing a house to purchase and live in are size aptness, accessibility, amenity, utility, soundness, and cost.

Perhaps this principle does not capture all relevant values but it suffices for illustration purposes. The model also needs to contain grading scale information such as the following:

Value	**Grading Scale**			
size aptness	poor	moderate	good	very good
accessibility	poor	moderate	good	
amenity	low	middling	high	very high
utility	poor	moderate	good	very good
soundness	unsound	sound		
cost	cardinal scale in units of £15 000			

This information can also be rendered in propositional form in the model's knowledge base as

> The grades on the scale of accessibility in choosing a house to purchase and live in are in ascending order - poor, moderate and good.

When appraisal is finished, the assessment of the values of the house choices might be as follows.

Name	Size Aptness	Access	Amenity	Utility	Soundnes s	Cost
Ashgrove	good	good	very high	very good	sound	£75-90K
Bellevue	moderate	good	high	good	sound	£75-90K
Cornhall	good	good	middling	very good	sound	£75-90K

The appraisal places each alternative on each relevant dimension of value using a relevant grade of discrimination. When the process is complete, the formal principle of choice can be applied. In our example the most expensive house "Ashgrove" comes out as the dominant choice. If there is no other alternative then all things considered Ashgrove ought to be bought.

7.2 Formal Theory of Value

This example can be formalised as follows:

Relevant Vocabulary

1) 3-place relational expressions buys(x, y, e), $x >_v y$ and $x =_v y$
2) variables for events e, f, g, for values v, w, for houses x, y, z, for persons i, j
3) 2-place *better alternative* connective $\Rightarrow$
4) house constants A, B, C
5) value constants Size, Access, Amenity, Utility, Sound, Cost
6) event constants X, Y, Z

The rules of well-formedness are the same as for the predicate calculus where event, house, person, and value variables are taken to be individual variables, the better alternative connective is taken to be a two place connective between relational expressions. The 3-place relations $x >_v y$ and $x =_v y$ stand for whichever 2-place value ordering relation the value variable v is interpreted as.

The interpretations of the terms are all straight forward. The constants A, B and C represent the three houses beginning with those letters. The constants X, Y and Z represent the three possible events of buying the houses A, B and C. The values constants are abbreviations of the relevant value names. The value ordering and equality relations are indexed by abbreviations of the relevant value dimensions.

The value ordering relations $>_v$ are irreflexive, asymmetrical, and transitive, and the value equality relations are transitive, reflexive and symmetrical. These assumptions are stated as A1-A6.

Assumptions of Theory

A1 $(\forall e) \neg e >_v e$

A2 $(\forall e)(e >_v f \supset \neg f >_v e)$

A3 $(\forall e)\,(e >_v f \,\&\, f >_v g \supset e >_v g)$

A4 $(\forall e)\,(e =_v f \,\&\, f =_v g \supset e =_v g)$

A5 $(\forall e)\; e =_v e$

A6 $(\forall e)(e =_v f \supset f =_v e)$

A7 expresses the relationship between value equality and value ordering.

A7 $(\forall e)(e =_v f \supset \neg\, e >_v f)$

The detailed valuations of the theory are stated in A8-A19.

A8 $X >_{Size} Y$

A9 $Z >_{Size} Y$

A10 $X =_{Access} Y$

A11 $Y =_{Access} Z$

A12 $X >_{Amenity} Y$

A13 $Y >_{Amenity} Z$

A14 $X >_{Utility} Y$

A15 $Z >_{Utility} Y$

A16 $X =_{Sound} Y$

A17 $Y =_{Sound} Z$

A18 $X =_{Cost} Y$

A19 $Y =_{Cost} Z$

These assumptions state the relationships along each dimension of value. These events X, Y and Z are the three (potential) acts of purchase. They are related to buying by the auxiliary assumptions

E1 $(\exists i)(\exists e)(\text{buys}(i, A, e) \,\&\, e = X)$

E2 $(\exists i)(\exists e)(\text{buys}(i, B, e) \,\&\, e = Y)$

E3 $(\exists i)(\exists e)(\text{buys}(i, C, e) \,\&\, e = Z)$

In order to bring overall merits into the picture a version of the dominance principle is needed. It is stated as two propositions A20 and A21.

A20 $(\forall e)(\forall f)((\exists v)(e >_v f) \,\&\, (\forall w)\,(\neg\, f >_w e) \supset e \Rightarrow f)$

A21 $(\forall w)(w = \text{Size} \vee w = \text{Access} \vee w = \text{Amenity} \vee w = \text{Utility} \vee w = \text{Sound} \vee w = \text{Cost})$

These assumptions A1-A21 suffice to prove the conclusions C1 and C2.

C1 $X \Rightarrow Y$

C2 $X \Rightarrow Z$

which establishes that buying Ashgrove is a better choice than either alternative. Since these are all the alternatives there are, buying Ashgrove is the thing to be done all things considered.

7.3 Risk and Valuations

The acknowledgement of a fundamental plurality of values and the admission that most values have only an ordinal and not a cardinal scale, deprives the Praxis model of the ability claimed by Bayesians to measure the relative merits of any set of alternative courses of action. However, this kind of model can still handle risk. Risk affects the worth of production values possessed by a course of action. The basic impact of risk on value is embodied as follows:

P4 If two alternatives A and B have chances P_a and P_b of producing a consequence of grade G of value V and $P_a > P_b$ then A has greater production value of V in that respect than B does.

This risk principle can be used directly as a basis of choice where risk is involved. Given the two alternatives of accepting a 50% chance of getting a £1 or a 30% chance of getting a £1, the 50% chance of getting a £1 has a greater production value of the same monetary value. Hence if monetary production value for the chooser is the sole value relevant to the choice, the dominance principle implies that the 50% alternative is the better one. However, given a choice between the alternatives T1 and T2

T1 50% chance of getting £1

T2 30% chance of getting £2

the risk principle is of no help in valuing the alternatives. It would be nice if some valid general principle for discounting production value in terms of risks could be established. Such a principle would specify the degree for discounting a production value of an action in terms of the chances of its valued consequences being realised. A consequence of such a discounting principle for a particular value might be the rule R:

R The monetary production value of a course of action is equal to the numerical product of the probability of the course of action producing monetary value with the numerical value of the money.

This rule R presupposes numerical measures of probability and money. Under some obvious assumptions it would allow one to value the alternatives A1 and A2 as having £0.5 and £0.6 monetary production values. If monetary production value for the chooser was the only relevant value in the choice situation, then it would enable one to identify A2 as the better alternative. However, it is highly doubtful that there can be any valid general discount principle which would entail a rule such as R. There is nothing intrinsic to the idea of valuation and risk that would justify such a principle. Rules such as R are highly dubious anyway because there is little reason to agree that there is no difference in monetary production value between the alternative choices D1 and D2.

D1 100% chance of getting £10

D2 50% chance of getting £20

The Praxis model operates with no general principle for discounting production values in terms of risk. It does this because under its assumptions it is not plausible to suppose that such a principle exists. However, rules such as R can still figure in the model as auxiliary postulates of evaluation or canons of assessment in order to determine how to grade the production value of a course of action. It is up to the user of the Praxis model to ensure that such use of auxiliary postulates will result in the right grades of production value being accorded to alternatives with only a chance of realising valued consequences.

8. Blocked Decision Making

The Praxis model makes a sharp separation between straight forward applications of the model of practical reasoning in accordance with praxeological, deontological and axiological considerations, and use of the default choice procedures. Straight-forward applications of the Praxis model are intended to yield uncontroversially valid results. So long as either one thing must be done or there are a set of permissible alternatives such that

a) they ought to be done

b) they are correctly valued by relevant values at the right degree of discrimination

c) one is a better alternative than any other

then it is right that that thing be done. Where these assumptions do not all hold, the model cannot resolve the issue of choice uncontroversially. Nonetheless good scope remains for resolving the issue of choice in a reasonable fashion.

Where there is more than one permissible alternative that ought to be done and no alternative is better than every other, then no alternative is the best. However, some alternatives may still be better than others. Without loss of generality the set of alternatives can be reduced to those alternatives that no alternative is better than. This set of alternatives will be called the undominated set. By hypothesis there will be two or more members of this set. Now members of this set will either all have the same grades of the relevant values or not.

same valued alternatives	*value indifference*
differently valued alternatives	*value conflict*

The set of alternatives will either be in a state of value indifference or in a state of value conflict. A state of value conflict does not prevent a strict subset of the alternatives from being in a state of value indifference. The Praxis model requires that choice be made in both these circumstances, because each alternative expresses something that ought to be done. Not choosing would mean that less of what ought to be done would be done than otherwise. A state of value indifference can be handled either by random choice or by reassessing the way the alternatives are valued.

Random choice is apposite for the logical donkey equidistant between two identical piles of hay, because the alternatives by definition are exactly alike. This situation is not typical. Usually some quite plausible changes in ways of valuing alternatives will alter valuations among the set of alternatives. Each of them gives an opportunity for a dominant alternative to emerge. For this reason random choice looks unappealing as a way of resolving a state of value indifference. Using such a method evades looking for some reason for showing that one alternative is more worthy than another.

Reassessing the way alternatives are valued also provides a good way of resolving value conflicts. There are many possible methods for reassessing the set of valuations. Two of them are:

More Discrimination	making the valuation of alternatives more precise
Less Discrimination	making the valuation of alternatives less precise

In terms of the house buying example, it would be possible to increase or decrease the number of grades of relevant values. Thus the utility scale could be given five or three gradations rather than four. The cardinal scale of cost could also have its granularity increased or diminished. Thus the units of significance for cost could be changed to £10,000 steps or to £20,000 steps. Clearly the less grades of values used, the more likely it is that a dominant alternative or a state of value indifference will emerge. Conversely the more discriminating valuation is, the more likely that the alternatives will be put into a state of value conflict. Nonetheless it is still possible to eliminate value conflicts by introducing more grades of values.

Let it be supposed that two alternatives A and B are graded in one order by the value V and in the other order by the value W. Furthermore let it be supposed that they are graded equal by all other relevant values. Lastly let it be supposed that A and B were graded as *slightly* and *moderately* valuable in terms of V. Introducing a new intermediate grade of V between *slightly* and *moderately* called "*mildly*" might result in both A and B being reclassified as *mildly*. This would dissipate the value conflict, and result in one alternative emerging as the dominant one.

Altering the scope of relevant values used in a valuation would also be a way of trying to resolve a blocked decision.

Widening Scope	increasing number of values considered relevant
Narrowing Scope	decreasing number of values considered relevant

In a state of value indifference in choosing a house to buy a deadlock breaking value like architectural quality could be introduced to try to obtain a dominant choice. Or in a state of value conflict a total ranking among values such as

Soundness > Cost > Amenity > Size Aptness > Accessibility > Utility

could be exploited to select the least important value which could be removed from the valuation. By eliminating the value *Utility* from the set of relevant values, a dominant alternative might emerge from the valuation. This process might be iterated by eliminating successively more important values until a dominant choice emerges. Plainly if such a process continues until only one value, the most important one is left, then either one alternative will emerge as the dominant one, or two or more alternatives will be left in a state of value indifference.

Each of these ways of resolving blocked choices has a certain appeal, but none seems to require to be tried first. For this reason using any one method for resolving blocked choices is potentially controversial. Use of another method might resolve a blocked choice a different way. Furthermore independent use of any of these methods might not be enough. They might be used in a number of possible combinations with different results. Plainly there are many ways of combining these methods, and it would seem likely that there are many other reasonable methods for resolving blocked choices which have not been considered. So there would seem to be wide scope for judgement. The Praxis model acknowledges this by imposing no general requirements on default rules for resolving blocked choices.

9. Conclusion

The Praxis model conceives of reasons for acting in terms of requirements, duties, proscriptions and values. It provides a framework for representing different kinds of reasons for and against alternatives. The first set specify what must be done. They are deduced from general requirements and the circumstances of the decision context. A second set specify what ought to be done. They are deduced from general praxeological principles and often imply that conflicting courses of action ought to be done. These oughts arise out of duties, needs and interests of the reasoner. The two sets of what must and ought to be done generate the options which means end reasoning refines into alternative things to be done. A third set of considerations are proscriptions specifying what may not be done. They set constraints on practical decision-making by excluding alternatives which entail forbidden acts. A fourth set of considerations in the Praxis model are values. They provide the dimensions of assessment for alternatives which ought to be done. They succeed if they show that one alternative has more value than any others. If none of these considerations determine what is to be done all things considered, the Praxis model allows default rules for resolving blocked choices to be applied.

10. Acknowledgements

I would like to thank Mike Clarke, John Fox, Paul Krause and Saki Hajnal for discussions on these topics. Work on this paper was supported by a SERC project concerned with formalising reasoning for decision support systems. The grant holders were Mike Clarke of QMW and John Fox of ICRF.

References

1. L. Aqvist, *Introduction to Deontic Logic and the Theory of Normative Systems*, Bibliopolis, Naples, 1987.
2. I. Asimov, *Robot City*, Orbit, 1988.
3. D. Davidson, *The Logical Form of Action Sentences, in Essays on Actions and Events*, Clarendon Press, Oxford, 1980.
4. G. Harman, *Change in View*, pp, 23-35, MIT Press 1986.
5. D. Hume, *Treatise on Human Nature*, Clarendon Press, Oxford, 1978.
6. R.C. Jeffrey, *The Logic of Decision*, McGraw-Hill Book Company, New York, 1965.
7. D.V. Lindley, *Making Decisions*, John Wiley, London, 1985.
8. R.D. Luce and H Raiffa, *Games and Decisions*, New York, 1957.
9. R. Nozick, *Philosophical Explanations*, Oxford University Press, 1984.
10. N. Rescher, *Introduction to Value Theory*, Prentice Hall, 1969.
11. D.A.J. Richards, *A Theory of Reasons for Action*, 1972.
12. L.J.Savage, *The Foundations of Statistics*, Dover, New York, 1972.
13. G.H. von Wright, *Essay in Deontic Logic and General Theory of Action*, North Holland, 1968.

Formalising Motivational Attitudes of Agents

On Preferences, Goals and Commitments

B. van Linder, W. van der Hoek, J.-J.Ch. Meyer

Utrecht University, Department of Computer Science
P.O. Box 80.089, 3508 TB Utrecht, The Netherlands
Email: bernd@cs.ruu.nl

Abstract. In this paper we propose a formalisation of motivational attitudes of rational agents. We deal with three such attitudes, situated at two levels. At the level of assertions we define preferences and goals of agents, at the practition level we define commitments. By providing an awareness-based semantics for preferences we avoid both the side-effect problem and the transference problem, as well as other problems related to logical omniscience. Goals are defined to be those preferences that the agent knows not to hold but to be possible. With respect to commitments we consider both a static and a dynamic aspect. The static aspect formalises the commitments that agents have made, the dynamic aspect formalises the act of making commitments. The resulting theory is a highly expressive one which satisfies many of the desiderata for motivational attitudes.

1 Introduction

The formalisation of rational agents is a topic of continuing interest in Artificial Intelligence. Research on this subject has held the limelight ever since the pioneering work of Moore [18] in which knowledge and actions are considered. Over the years important contributions have been made on *informational* attitudes like knowledge and belief, on *motivational* attitudes like commitments and obligations, on aspects of actions, and on combinations of these [3, 12, 21, 22, 24, 26].

Our research [8, 9, 13, 14, 15, 16, 17] deals with defining an all-embracing framework in which all relevant aspects of agency may be formalised. The area of application of this formal framework is the analysis, specification and verification of rational agents, i.e. the framework is meant to be used by *theorists* to *reason about* rational agents. For this framework to be all-embracing it needs to deal with informational attitudes, various aspects of action, and motivational attitudes. Whereas until now we have been concerned with formalising informational attitudes, actions and the interaction between these notions, in this paper we present a formalisation of the motivational attitudes of agents.

In the last decade various formalisations of different kinds of motivational attitudes have been proposed [3, 19, 20, 23]. The approach presented in this paper makes three main contributions to the theory of formalising motivational attitudes. Firstly, we consider a fairly wide scope of motivational attitudes, situated at two different levels. At the *assertion* level, this is the level where operators deal with assertions, we consider *preferences* and *goals*. At the *practition*[1] level, where operators range over actions, we

[1] The term *practition* is due to Castañeda [2].

define *commitments*. With respect to these commitments we introduce both an operator modelling the commitments that an agent has made, and an action which models the act of committing. The notions that we formalise avoid (most of) the well-known problems that plague formalisations of motivational attitudes. Secondly, our formalisation of the various notions is strictly *bottom up*. That is, after defining the primitive notion of preferences, goals are defined in terms of preferences, and commitments are introduced using the notion of goals. In this way, we provide a formalisation of motivational attitudes that does not have to resort to tricks like (circularly) defining the intention to do an action in terms of the goal to have done it. Lastly, in our formalisation we aim at being faithful — to a certain degree — to the insights on motivational attitudes as they have been gained in the philosophical research on practical reasoning.

The rest of the paper is organised as follows. We start in Sect. 2 with the formalisation of informational attitudes and aspects of action. In Sect. 3 we formalise a notion of preference. The formalisation of goals is the subject of Sect. 4. In Sect. 5 we discuss the importance of the notion of accordance of actions. In Sect. 6 we formalise commitments and the act of committing. In Sect. 7 we round off.

2 Knowledge, abilities, opportunities, and results

The main informational attitude that we consider is that of *knowledge*. In representing knowledge we follow the approach common in epistemic logic [5, 7]: the formula $\mathbf{K}_i\varphi$ denotes the fact that agent i knows φ, and is interpreted in a Kripke-style possible worlds semantics.

At the action level we consider *results*, *abilities* and *opportunities*. Slightly simplifying ideas of Von Wright [27], we consider any aspect of the state of affairs brought about by the execution of an action by an agent in some state to be among the results of the event consisting of the execution of that particular action by the particular agent, in the particular state. An important aspect of any investigation of action is the relation that exists between ability and opportunity. In order to successfully complete an action, both the opportunity and the ability to perform the action are necessary. Although these notions are interconnected, they are surely not identical: the abilities of agents comprise mental and physical powers, moral capacities, and human and physical possibility, whereas the opportunity to perform actions is best described by the notion of circumstantial possibility (cf. [11]). The abilities of agents are formalised via the $\mathbf{A}_i$ operator; the formula $\mathbf{A}_i\alpha$ denotes the fact that agent i has the ability to do α. When using the descriptions of opportunities and results as given above, the framework of (propositional) dynamic logic provides an excellent means to formalise these notions. Using events $\mathrm{do}_i(\alpha)$ to refer to the performance of the action α by the agent i, we consider the formulae $\langle\mathrm{do}_i(\alpha)\rangle\varphi$ and $[\mathrm{do}_i(\alpha)]\varphi$. In our deterministic framework, $\langle\mathrm{do}_i(\alpha)\rangle\varphi$ is the stronger of these formulae; it represents the fact that agent i has the opportunity to do α and that doing α leads to φ. The formula $[\mathrm{do}_i(\alpha)]\varphi$ is noncommittal about the opportunity of the agent to do α but states that if the opportunity to do α is indeed present, doing α results in φ.

Definition 1. Let a finite set $\mathrm{A} = \{1, \ldots, n\}$ of agents, and some denumerable sets Π of propositional symbols and At of atomic actions be given. The language L is the smallest superset of Π such that:

- if $\varphi, \psi \in \mathrm{L}, i \in \mathrm{A}, \alpha \in \mathrm{Ac}$ then $\neg\varphi, \varphi \vee \psi, \mathbf{K}_i\varphi, \langle \mathrm{do}_i(\alpha)\rangle\varphi, \mathbf{A}_i\alpha \in \mathrm{L}$

where Ac is the smallest superset of At such that if $\varphi \in \mathrm{L}, \alpha, \alpha_1, \alpha_2 \in \mathrm{Ac}$ then

- $\mathtt{confirm}\,\varphi \in \mathrm{Ac}$ *confirmations*
- $\alpha_1 ; \alpha_2 \in \mathrm{Ac}$ *sequential composition*
- $\mathtt{if}\,\varphi\,\mathtt{then}\,\alpha_1\,\mathtt{else}\,\alpha_2\,\mathtt{fi} \in \mathrm{Ac}$ *conditional composition*
- $\mathtt{while}\,\varphi\,\mathtt{do}\,\alpha\,\mathtt{od} \in \mathrm{Ac}$ *repetitive composition*

The purely propositional fragment of L is denoted by L_0. Constructs $\wedge, \rightarrow, \leftrightarrow$, $\top, \bot$ and $[\mathrm{do}_i(\alpha)]\varphi$ are defined in the usual way.

Definition 2. The class $\mathbf{M}$ of models contains all $\mathrm{M} = \langle \mathrm{S}, \pi, \mathrm{R}, \mathrm{r}_0, \mathrm{c}_0 \rangle$ where

- S is a set of possible worlds, or states.
- $\pi : \Pi \times \mathrm{S} \rightarrow \{0, 1\}$ assigns a truth value to propositional symbols in states.
- $\mathrm{R} : \mathrm{A} \rightarrow \wp(\mathrm{S} \times \mathrm{S})$ is a function that yields the epistemic accessibility relations for a given agent. It is demanded that $\mathrm{R}(i)$ is an equivalence relation for all i. We define $[s]_{\mathrm{R}(i)}$ to be $\{s' \in \mathrm{S} \mid (s, s') \in \mathrm{R}(i)\}$.
- $\mathrm{r}_0 : \mathrm{A} \times \mathrm{At} \rightarrow (\mathrm{S} \cup \{\emptyset\}) \rightarrow (\mathrm{S} \cup \{\emptyset\})$ is such that $\mathrm{r}_0(i, a)(s)$ yields the (possibly empty) state transition in s caused by the event $\mathrm{do}_i(a)$.
- $\mathrm{c}_0 : \mathrm{A} \times \mathrm{At} \rightarrow (\mathrm{S} \cup \{\emptyset\}) \rightarrow \{0, 1\}$ is the capability function such that $\mathrm{c}_0(i, a)(s)$ indicates whether the agent i is capable of performing the action a in s.

In our interpretation of actions as given in Def. 3 below, we generalise the standard paradigm of actions as state-transitions [6] by interpreting actions as transitions between pairs (Model, State) rather than transitions between states *per se*. Using this more general interpretation we can both account for regular actions that cause a transition between states upon execution, and special actions that transform models. Among the special actions that we previously considered were those modelling observations and communication; in Sect. 6 we formalise another special, non-regular action, viz. one modelling the act of committing.

Definition 3. The binary relation $\models$ between a formula from L and a pair M, s consisting of a model $\mathrm{M} \in \mathbf{M}$ and a state s in M, is for φ a propositional symbol, a negation, or a disjunction inductively defined as usual. For the other cases $\mathrm{M}, s \models \varphi$ is defined by:

$$\begin{array}{ll}
\mathrm{M}, s \models \mathbf{K}_i\varphi & \Leftrightarrow \forall s' \in \mathrm{S}((s, s') \in \mathrm{R}(i) \Rightarrow \mathrm{M}, s' \models \varphi) \\
\mathrm{M}, s \models \langle \mathrm{do}_i(\alpha)\rangle\varphi & \Leftrightarrow \exists \mathrm{M}', s'(\mathrm{M}', s' \in \mathrm{r}(i, \alpha)(\mathrm{M}, s) \,\&\, \mathrm{M}', s' \models \varphi) \\
\mathrm{M}, s \models \mathbf{A}_i\alpha & \Leftrightarrow \mathrm{c}(i, \alpha)(\mathrm{M}, s) = \mathbf{1}
\end{array}$$

where r and c are defined as follows:

$$\begin{array}{ll}
\mathrm{r}(i, a)(\mathrm{M}, s) & = \mathrm{M}, \mathrm{r}_0(i, a)(s) \\
\mathrm{r}(i, \mathtt{confirm}\,\varphi)(\mathrm{M}, s) & = \mathrm{M}, s \text{ if } \mathrm{M}, s \models \varphi \text{ and } \emptyset \text{ otherwise} \\
\mathrm{r}(i, \alpha_1 ; \alpha_2)(\mathrm{M}, s) & = \mathrm{r}(i, \alpha_2)(\mathrm{r}(i, \alpha_1)(\mathrm{M}, s)) \\
\mathrm{r}(i, \mathtt{if}\ \varphi\ \mathtt{then}\ \alpha_1 & = \mathrm{r}(i, \alpha_1)(\mathrm{M}, s) \text{ if } \mathrm{M}, s \models \varphi \text{ and} \\
\quad \mathtt{else}\ \alpha_2\ \mathtt{fi})(\mathrm{M}, s) & \quad \mathrm{r}(i, \alpha_2)(\mathrm{M}, s) \text{ otherwise} \\
\mathrm{r}(i, \mathtt{while}\,\varphi\,\mathtt{do}\,\alpha\,\mathtt{od})(\mathrm{M}, s) & = \mathrm{M}', s' \text{ such that } \exists k \in \mathbb{N} \exists \mathrm{M}_0, s_0 \ldots \exists \mathrm{M}_k, s_k \\
 & \quad (\mathrm{M}_0, s_0 = \mathrm{M}, s \,\&\, \mathrm{M}_k, s_k = \mathrm{M}', s' \,\&\, \forall j < k \\
 & \quad (\mathrm{M}_{j+1}, s_{j+1} = \mathrm{r}(i, \mathtt{confirm}\,\varphi; \alpha)(\mathrm{M}_j, s_j)) \,\& \\
 & \qquad\qquad \mathrm{M}', s' \models \neg\varphi)
\end{array}$$

$$
\begin{array}{ll}
\text{where } \mathtt{r}(i,\alpha)(\emptyset) & = \emptyset \\
\text{and} & \\
\quad \mathtt{c}(i,a)(\mathrm{M},s) & = \mathtt{c}_0(i,a)(s) \\
\quad \mathtt{c}(i,\mathtt{confirm}\,\varphi)(\mathrm{M},s) & = \mathbf{1} \text{ if } \mathrm{M},s \models \varphi \text{ and } \mathbf{0} \text{ otherwise} \\
\quad \mathtt{c}(i,\alpha_1;\alpha_2)(\mathrm{M},s) & = \mathtt{c}(i,\alpha_1)(\mathrm{M},s)\ \&\ \mathtt{c}(i,\alpha_2)(\mathtt{r}(i,\alpha_1)(\mathrm{M},s)) \\
\quad \mathtt{c}(i,\mathtt{if}\ \varphi\ \mathtt{then}\ \alpha_1 & = \mathtt{c}(i,\mathtt{confirm}\,\varphi;\alpha_1)(\mathrm{M},s) \text{ or} \\
\qquad \mathtt{else}\ \alpha_2\ \mathtt{fi})(\mathrm{M},s) & \ \ \mathtt{c}(i,\mathtt{confirm}\,\neg\varphi;\alpha_2)(\mathrm{M},s) \\
\quad \mathtt{c}(i,\mathtt{while}\,\varphi\,\mathtt{do}\,\alpha\,\mathtt{od})(\mathrm{M},s) & = \mathbf{1} \text{ if } \mathtt{c}(i,(\mathtt{confirm}\,\varphi;\alpha)^k;\mathtt{confirm}\,\neg\varphi)(\mathrm{M},s) \\
 & \qquad\qquad = \mathbf{1} \text{ for some } k \in \mathbb{N} \text{ and } \mathbf{0} \text{ otherwise} \\
\text{where } \mathtt{c}(i,\alpha)(\emptyset) & = \mathbf{1}
\end{array}
$$

Validity in a model and in a class of models is defined as usual.

With regard to the abilities of agents, the motivation for the choices made in Def. 3 is the following. The definition of $\mathtt{c}(i, \mathtt{confirm}\,\varphi)$ expresses that an agent is able to get confirmation for a formula φ iff φ holds. An agent is capable of performing a sequential composition $\alpha_1; \alpha_2$ iff it is capable of performing α_1 (now), and it is capable of executing α_2 after it has performed α_1. An agent is capable of performing a conditional composition, if either it is able to get confirmation for the condition and thereafter perform the then-part, or it is able to confirm the negation of the condition and perform the else-part afterwards. An agent is capable of performing a repetitive composition $\mathtt{while}\,\varphi\,\mathtt{do}\,\alpha\,\mathtt{od}$ iff it is able to perform the action $(\mathtt{confirm}\,\varphi; \alpha_1)^k; \mathtt{confirm}\,\neg\varphi$ for some natural number k, i.e. it is able to perform the kth unwinding of the while-loop.

2.1 The Can-predicate and the Cannot-predicate

To formalise the knowledge of agents on their practical (im)possibilities, we introduced the so-called Can-predicate and Cannot-predicate. These are binary predicates, pertaining to a pair consisting of an action and a proposition, and denoting that an agent knows that performing the action constitutes a practical (im)possibility to bring about the proposition. We consider practical possibility to consist of two parts, viz. correctness and feasibility: action α is *correct* with respect to φ iff $\langle \mathrm{do}_i(\alpha)\rangle\varphi$ holds and α is *feasible* iff $\mathbf{A}_i\alpha$ holds.

Definition 4. The Can-predicate and the Cannot-predicate are, for all agents i, actions α and formulae φ, defined as follows.

- $\mathbf{PracPoss}_i(\alpha,\varphi) =^{\mathrm{def}} \langle \mathrm{do}_i(\alpha)\rangle\varphi \wedge \mathbf{A}_i\alpha$
- $\mathbf{Can}_i(\alpha,\varphi) =^{\mathrm{def}} \mathbf{K}_i\mathbf{PracPoss}_i(\alpha,\varphi)$
- $\mathbf{Cannot}_i(\alpha,\varphi) =^{\mathrm{def}} \mathbf{K}_i\neg\mathbf{PracPoss}_i(\alpha,\varphi)$

3 Formalising preferences

The primitive notion that forms the foundations of our treatment of motivational attitudes is that of *preferences*[2]. Preferences of an agent describe aspects of states of affairs

[2] One should not feel to strongly about our use of the term 'preference'. The reason for using the term 'preference' will be made clear below. Should one nevertheless consider this term inap-

that it prefers to be the case. Agents may for instance prefer states of affairs in which towers have been build out of blocks, teeth are restored, and the enemy's weapons factories have been bombed. The most important feature of preferences as we formalise them is that — among others — the *side-effect problem* [1], which causes agents to prefer all (logical) consequences of one of their preferences, and the transference problem [20], which forces agents to prefer all things that are (logically) inevitable, are avoided.

Syntactically, both the side-effect problem and the transference problem can be seen as special cases of the well-known problems of *logical omniscience* that play an important part in the research on formalising informational attitudes [4]. We explore this relation by broadening our scope to include other problems of logical omniscience, rather than focusing solely on the side-effect and transference problem.

Definition 5. For φ and ψ formulae, and $\mathbf{X}$ some operator, we consider the following properties of logical omniscience.
(LO1) $\models \mathbf{X}\varphi \wedge \mathbf{X}(\varphi \rightarrow \psi) \rightarrow \mathbf{X}\psi$
(LO2) $\models \varphi \Rightarrow \models \mathbf{X}\varphi$
(LO3) $\models \varphi \rightarrow \psi \Rightarrow \models \mathbf{X}\varphi \rightarrow \mathbf{X}\psi$
(LO4) $\models \varphi \leftrightarrow \psi \Rightarrow \models \mathbf{X}\varphi \leftrightarrow \mathbf{X}\psi$
(LO5) $\models (\mathbf{X}\varphi \wedge \mathbf{X}\psi) \rightarrow \mathbf{X}(\varphi \wedge \psi)$
(LO6) $\models \mathbf{X}\varphi \rightarrow \mathbf{X}(\varphi \vee \psi)$
(LO7) $\models \neg(\mathbf{X}\varphi \wedge \mathbf{X}\neg\varphi)$

Properties LO1 and LO3 as given in Def. 5 capture the side-effect problem, and property LO2 captures the transference problem. Of the other properties given not all are considered equally harmful when formalising preferences. In our opinion, property LO4 does not do much harm, and LO7 could even be considered desirable, dependent on the demands for rationality that one is willing to make: consistency of the preferences of an agent can be seen as constituting part of its rationality (this is actually the approach we follow in formalising preferences). Property LO5, which we like to think of as representing the problem of *unrestricted combining*, is in general undesirable. This is for instance shown by the example of an agent that prefers watching TV and prefers reading a book, while not preferring to watch TV and read a book at the same time. Property LO6, representing the problem of *unrestricted weakening*, is a special instantiation of the side-effect problem. That this property is undesirable is shown by the example of an agent preferring itself to be painted green, without preferring being green or being crushed under a steam roller.

One of the most general and flexible approaches towards a solution for the problem of logical omniscience is the one using *awareness* as proposed by Fagin & Halpern [4]. In this approach agents explicitly know those formulae that they implicitly know and are aware of. The semantics for preferences that we present here is based on the same ideas: an agent prefers some formula iff it is true at a set of preferred alternatives, i.e. the agent *implicitly* prefers the formula, and it is furthermore considered to be an *explicit preference*. This combination of implicit and explicit preferences has some very

propriate, then there is nothing to prevent one from thinking of preference in terms of 'wish', 'desire' or 'concern', as long as the intuition underlying this term is that of a primitive, fundamental, motivational attitude.

nice properties. First of all, using the explicit preferences of the agent to act as a kind of filter on its implicit preferences allows one to avoid the side-effect and the transference problem: these problems do arise for the agent's implicit preferences, but not for the combination of implicit and explicit preferences. Secondly, by employing a modal semantics — and in particular a semantics that validates the D-axiom — for implicit preferences, it can easily be ensured that the preferences of an agent are consistent, a property which we consider to be a sign for the rationality of our rational agents. Also from an intuitive point of view it seems reasonable to define preferences in this way. For humans may explicitly prefer states of the world in which it does not rain, without preferring the sun to rise in the east, even though this will be the case in all preferred states of the world, i.e. there is some sort of filter that makes that this latter fact is not preferred.

The following definitions formalise the informal ideas given above.

Definition 6. The language L is extended as follows:

- if $i \in \mathrm{A}$ and $\varphi \in \mathrm{L}$ then $\mathbf{P}_i\varphi \in \mathrm{L}$

The intuitive interpretation of $\mathbf{P}_i\varphi$ is that agent i prefers φ to be true.

Definition 7. The Kripke models of Def. 2 are extended with

- a function $\mathrm{P} : \mathrm{A} \to \wp(\mathrm{S} \times \mathrm{S})$ yielding the *preferential accessibility* relation for agents and such that for all $i \in \mathrm{A}$, $\mathrm{P}(i)$ is serial.
- a function $\mathrm{Ep} : \mathrm{A} \times \mathrm{S} \to \wp(\mathrm{L})$ denoting *explicit preferences*.

Definition 8. For all models $\mathrm{M} = \langle \mathrm{S}, \pi, \mathrm{R}, \mathrm{r}_0, \mathrm{c}_0, \mathrm{P}, \mathrm{Ep} \rangle$ with $s \in \mathrm{S}$, we define for all $i \in \mathrm{A}$ and $\varphi \in \mathrm{L}$: $\mathrm{M}, s \models \mathbf{P}_i\varphi \Leftrightarrow (\forall s' \in \mathrm{S}((s, s') \in \mathrm{P}(i) \Rightarrow \mathrm{M}, s' \models \varphi) \,\&\, \varphi \in \mathrm{Ep}(i, s))$.

Def. 8 indeed provides for a semantics in which all problems of logical omniscience are avoided that we consider undesirable when formalising preferences.

Proposition 9. *Of the logical omniscience properties mentioned in Def. 5 only LO7 is valid for the* $\mathbf{P}_i$ *operator when defining the semantics as in Def. 8.*

Although the definition given above is perfectly acceptable from the point of view that all problems of logical omniscience that are to be avoided are indeed avoided, one may feel that the use of a totally unstructured set of explicit preferences allows for too liberal a semantics. Just as it is done for the epistemic awareness approach of Fagin & Halpern, one may be tempted to demand certain *closure* properties of the set of explicit preferences. However, whereas these closure properties are quite usable in the epistemic approach, they are much less so for preferences. Properties like closure under conjunction (if $\varphi \in \mathrm{Ep}(i, s)$ and $\psi \in \mathrm{Ep}(i, s)$ then $\varphi \wedge \psi \in \mathrm{Ep}(i, s)$) and decomposition under conjunction (if $\varphi \wedge \psi \in \mathrm{Ep}(i, s)$ then $\varphi \in \mathrm{Ep}(i, s)$ and $\psi \in \mathrm{Ep}(i, s)$), which are among the least controversial closure properties proposed for epistemic awareness, are clearly unacceptable for explicit preferences, since they reintroduce all kinds of problems of logical omniscience that are to be avoided. We feel that, in the light of the special nature of preferences, it is best not to impose any closure properties upon the set of explicit preferences.

3.1 Preferences persist and are known

We think of preferences as being both *persistent* and *known*, i.e. agents know their preferences, and preferences do not tend to change, neither easily nor often[3]. A counterexample [10] against persistence of preferences is that of an agent that would prefer a banana right here and now, but may not prefer a banana after having eaten one (and instead preferring an orange). In our opinion this example is not completely convincing since it somehow makes an improper use of the concept of preferences. If one rephrases the example such that the agent prefers having eaten a banana, then it would still prefer having eaten a banana after eating one, but this preference would no longer influence the course of action that it is going to take, simply because it was already brought about. That is, the *preference in itself is not changed*, but due to a *change of circumstances* the preference is dealt with differently by the agent. Both the property that preferences are known and the property of persistence of preferences are implemented by imposing suitable, first-order expressible, constraints on the models that we consider.

Definition 10. A model $\mathrm{M} = \langle \mathrm{S}, \pi, \mathrm{R}, \mathrm{r}_0, \mathrm{c}_0, \mathrm{P}, \mathrm{Ep} \rangle$ as defined in 2 and 7 satisfies the demand for *persistence of preferences* iff

$$\forall s' \in \mathrm{S} \forall i \in \mathrm{A} \forall a \in \mathrm{At} \forall s \in \mathrm{S}(s' = \mathrm{r}_0(i,a)(s) \Rightarrow$$
$$(\forall s'' \in \mathrm{S}((s',s'') \in \mathrm{P}(i) \Rightarrow (s,s'') \in \mathrm{P}(i))\,\&\, \mathrm{Ep}(i,s) \subseteq \mathrm{Ep}(i,s')))$$

The model M satisfies the demand that *preferences are known* iff

$$\forall s, s' \in \mathrm{S} \forall i \in \mathrm{A}((s,s') \in \mathrm{R}(i) \Rightarrow$$
$$(\forall s'' \in \mathrm{S}((s',s'') \in \mathrm{P}(i) \Rightarrow (s,s'') \in \mathrm{P}(i))\,\&\, \mathrm{Ep}(i,s) \subseteq \mathrm{Ep}(i,s')))$$

Imposing the constraints given in Def. 10 upon the models under consideration, indeed suffices to ensure that preferences are known and persist.

Proposition 11. *For all models* M, *we have for all* $i \in \mathrm{A}$, $\varphi \in \mathrm{L}$ *and* $\alpha \in \mathrm{Ac}$:

- *if* M *meets the demand for persistence of preferences then* $\mathrm{M} \models \mathbf{P}_i\varphi \rightarrow [\mathrm{do}_i(\alpha)]\mathbf{P}_i\varphi$.
- *if* M *meets the demand that preferences are known then* $\mathrm{M} \models \mathbf{P}_i\varphi \rightarrow \mathbf{K}_i\mathbf{P}_i\varphi$.

Although we assume preferences to persist and to be known, these properties are brought about by imposing additional constraints on the models under consideration, and are by no means forced by the framework. Assuming these properties has some technical advantages, in particular for the formalisation of commitments as presented in Sect. 6. Nevertheless, relaxation of these properties and the consequences of such a relaxation are subject of further research.

[3] This idea of persistence was actually one of the reasons to refer to our primitive motivational attitude as preference, rather than wish or desire, for preferences seem intuitively to be much less liable to change than wishes or desires.

4 Goals are unfulfilled, realistic preferences

Informally speaking, the goals of an agent are determined by its *unfulfilled preferences*. However, not all of the unfulfilled preferences of an agent qualify as goals. Besides being unfulfilled, goals also need to be *realistic*, i.e. an unfilled preference is a goal for an agent only if the agent knows that it is somehow possible to fulfil the preference.

To formalise realism of goals, we introduce the notion of *implementability*. Intuitively, a formula is implementable for an agent if there is some way open to the agent to bring about the formula.

Definition 12. For agents i and formulae φ, we extend the language L with the formula $\Diamond_i\varphi$, with intuitive interpretation that φ is implementable for agent i. The semantics for the $\Diamond_i$ operator is defined by:

$$\mathrm{M}, s \models \Diamond_i\varphi \Leftrightarrow \exists k \in \mathbb{N} \exists a_1, \ldots, a_k \in \mathrm{At}(\mathrm{M}, s \models \mathbf{PracPoss}_i(a_1; \ldots; a_k, \varphi))$$

A formulae φ is realistic for agent i in M, s iff *i knows* φ to be *implementable*.

A goal is now defined to be an unfulfilled, realistic, known preference.

Definition 13. The operator $\mathbf{Goal}_i$ is for agents i and formulae φ introduced by definitional abbreviation: $\mathbf{Goal}_i\varphi =^{\mathrm{def}} \mathbf{K}_i\mathbf{P}_i\varphi \wedge \mathbf{K}_i\neg\varphi \wedge \mathbf{K}_i\Diamond_i\varphi$.

In the definition of goals as presented above, the second conjunct is intended to provide the *halting condition* for action, i.e. the condition that determines whether a preference of an agent (still) determines the course of action that the agent is supposed to take. For the agents that we formalise it is indeed the case that after performing an action that is correct with regard to a given goal, the goal ceases to be such.

Proposition 14. *For $i \in$ A, for all actions α and for all formulae φ we have:*

- $\models \langle \mathrm{do}_i(\alpha)\rangle\varphi \rightarrow \langle \mathrm{do}_i(\alpha)\rangle\neg\mathbf{Goal}_i\varphi$

Prop. 14 nicely indicates that preferences, though persistent, may cease to influence the course of action to be taken by an agent. For if some action α is performed which is correct with respect to a given formula φ which is preferred by the agent, then φ will definitely not be a goal of the agent after execution of α. Consider the example of the agent that prefers having eaten a banana, and finds itself in a situation where it knows that although it has not yet eaten a banana, it has the opportunity to do so (since one is lying right in front of its face). This agent has 'having eaten a banana' as one of its goals. However, as soon as it has eaten one, it no longer considers having eaten a banana as its goal, even though it still considers states in which it has eaten a banana to be pleasant (preferable) ones.

The following proposition summarises some of the properties of the goal operator defined above.

Proposition 15. *For all agents i and formulae φ, ψ we have:*

1. $\models \mathbf{Goal}_i\varphi \rightarrow \mathbf{K}_i\neg\varphi$
2. $\models \mathbf{Goal}_i\varphi \rightarrow \mathbf{K}_i\Diamond_i\varphi$
3. $\models \mathbf{Goal}_i\varphi \rightarrow \neg\mathbf{Goal}_i\neg\varphi$

4. $\models \mathbf{Goal}_i\varphi \to \mathbf{K}_i\mathbf{Goal}_i\varphi$
5. $\models \varphi \Rightarrow \models \neg\mathbf{Goal}_i\varphi$
6. *not for all* φ, ψ *does* $\models \varphi \to \psi$ *imply* $\models \mathbf{Goal}_i\varphi \to \mathbf{Goal}_i\psi$
7. $(\varphi \to \psi) \to (\mathbf{Goal}_i\varphi \to \mathbf{Goal}_i\psi)$ *is not for all* φ, ψ *valid*
8. $\mathbf{K}_i(\varphi \to \psi) \to (\mathbf{Goal}_i\varphi \to \mathbf{Goal}_i\psi)$ *is not for all* φ, ψ *valid*

The first two items of Prop. 15 state that goals are both unfulfilled and realistic. Item 3 formalises the idea that goals are mutually consistent. Item 4 states that agents know their goals, which seems to be highly desirable a property for rational agents: for how are agents to act upon their goals if they do not know them? Item 5 states that the transference problem is avoided for goals: validities are never goals. The last three items indicate that the side-effect problem is avoided in our formalisation of goals: goals are neither closed under logical consequence nor under known consequence.

5 The importance of accordance

In this section we discuss a property of actions, which is important both from an intuitive and from a technical point of view, namely that of *accordance*. As we see it, accordance captures the idea that actions behave *according to plan*, i.e. if some agent knows that φ holds as a result of executing α, then it knows that φ holds after α is executed.

Definition 16. Let $\mathrm{M} = \langle \mathrm{S}, \pi, \mathrm{R}, \mathbf{r}_0, \mathrm{c}_0, \mathrm{P}, \mathrm{Ep}\rangle$ be some Kripke model.

- $\mathrm{do}_i(\alpha)$ is *accordant* in $s \in \mathrm{S}$ iff $\mathrm{M}, s \models \mathbf{K}_i[\mathrm{do}_i(\alpha)]\varphi \to [\mathrm{do}_i(\alpha)]\mathbf{K}_i\varphi$ for all formulae φ.
- $\mathrm{do}_i(\alpha)$ is accordant for M iff $\mathrm{do}_i(\alpha)$ is accordant in all s from S.

Accordance of all actions can be brought about by imposing a constraint on the functions $\mathbf{r}_0$ and R in the models.

Definition 17. A model $\mathrm{M} = \langle \mathrm{S}, \pi, \mathrm{R}, \mathbf{r}_0, \mathrm{c}_0, \mathrm{P}, \mathrm{Ep}\rangle$ satisfies the demand for accordance iff for all $a \in \mathrm{At}$

$$\begin{aligned} &\forall s_0 \in \mathrm{S}\forall s_1 \in \mathrm{S}(\exists s_2 \in \mathrm{S}(s_2 = \mathbf{r}_0(i,a)(s_0)\,\&\,(s_2,s_1) \in \mathrm{R}(i)) \Rightarrow \\ &\qquad \exists s_3 \in \mathrm{S}((s_0,s_3) \in \mathrm{R}(i)\,\&\,s_1 = \mathbf{r}(i,a)(s_3))) \end{aligned}$$

It turns out that imposing the constraint given in Def. 17, which deals with *atomic* actions only, suffices to achieve the property of accordance for *all* actions.

Proposition 18. *For all models* M *satisfying the demand for accordance we have for all agents i, actions* α *and formulae* φ*:* $\mathrm{M} \models \mathbf{K}_i[\mathrm{do}_i(\alpha)]\varphi \to [\mathrm{do}_i(\alpha)]\mathbf{K}_i\varphi$.

The notion of accordance is rather important when considering the planning of agents. For whenever an agent *knows* some action to be correct with respect to a given goal, the agent will *know* its goal to be fulfilled after performing the action. Accordance is furthermore an important and useful notion with regard to the agents' commitments. Informally speaking, the fact that actions behave according to plan ensures that agents can honour their future commitments in due time.

6 Formalising commitments

At the practition level we formalise the notion of commitment. In this formalisation we distinguish a *static* and a *dynamic* component. The static aspect represents the commitments that agents have made, the dynamic aspect formalises the act of making commitments. As such, the static part corresponds to the *past participle 'committed'*, whereas the dynamic part corresponds to the *infinitive 'to commit'*. The act of committing is of a different nature than the regular actions that are at the agents' disposal. After an agent has found out which action constitutes a correct and feasible plan for some goal, it may decide to commit itself to this action; this committing constitutes a kind of '*meta* action', which is fundamentally different from other actions. To account for the distinction between the act of committing and regular actions, the language that we use in this section is defined in two stages. In the first stage we define the language L_1^C as an extension of L in which also preference and implementability formulae are included. The class of actions used in defining L_1^C is the class of regular, mundane actions. In the second stage the language L^C is defined. In this language operators are included that model the commitments that agents have made. The class Ac^C of actions contains special actions modelling the act of committing to one of the regular, mundane actions defined in the first stage.

Definition 19. The language L_1^C and the class Ac_1^C of regular actions are defined as L and Ac in Def. 1, with the addition that L_1^C is furthermore such that:

- if $i \in \mathrm{A}$ and $\varphi \in \mathrm{L}_1^C$ then $\mathbf{P}_i\varphi, \Diamond_i\varphi \in \mathrm{L}_1^C$

The language L^C and the class Ac^C of general actions are defined by:

- L^C is the smallest superset of $\mathit{\Pi}$ such that
  - if $\varphi, \psi \in \mathrm{L}^C$ then $\neg\varphi, \varphi \vee \psi \in \mathrm{L}^C$
  - if $i \in \mathrm{A}$ and $\varphi \in \mathrm{L}^C$ then $\mathbf{K}_i\varphi, \mathbf{B}_i\varphi, \mathbf{P}_i\varphi, \Diamond_i\varphi \in \mathrm{L}^C$
  - if $i \in \mathrm{A}, \alpha \in \mathrm{Ac}^C$ and $\varphi \in \mathrm{L}^C$ then $\langle \mathrm{do}_i(\alpha) \rangle \varphi, \mathbf{A}_i\alpha \in \mathrm{L}^C$
  - if $i \in \mathrm{A}, \alpha \in \mathrm{Ac}_1^C$ then $\mathbf{Committed}_i\alpha \in \mathrm{L}^C$
- Ac^C is the smallest superset of $\mathrm{Ac}_1^C \cup \{\mathtt{commit\_to}\, \alpha \mid \alpha \in \mathrm{Ac}_1^C\}$ such that
  - if $\varphi \in \mathrm{L}^C$ then $\mathtt{confirm}\, \varphi \in \mathrm{Ac}^C$
  - if $\alpha_1, \alpha_2 \in \mathrm{Ac}^C$ then $\alpha_1 ; \alpha_2 \in \mathrm{Ac}^C$
  - if $\varphi \in \mathrm{L}^C, \alpha_1, \alpha_2 \in \mathrm{Ac}^C$ then $\mathtt{if}\, \varphi\, \mathtt{then}\, \alpha_1\, \mathtt{else}\, \alpha_2\, \mathtt{fi} \in \mathrm{Ac}^C$
  - if $\varphi \in \mathrm{L}^C, \alpha \in \mathrm{Ac}^C$ then $\mathtt{while}\, \varphi\, \mathtt{do}\, \alpha\, \mathtt{od} \in \mathrm{Ac}^C$

The shift from motivational attitudes at the assertion level to motivational attitudes at the practition level has been subject of research in analytical philosophy ever since Aristotle. From a philosophical point of view it is the *practical reasoning* of agents that transforms motivational attitudes at the assertion level into motivational attitudes at the practition level. The essence of practical reasoning is captured by the so-called *syllogism of practical reasoning*, here given in a version by Von Wright [28]:

> i intends to make it true that φ
> i thinks that, unless it does α, it will not achieve this
> Therefore i intends to do α.

In our formal system the gap between the assertion level and (the static part of) the practition level is bridged by the `commit_to` action. Ideally, this `commit_to` action should capture some of the characteristic ideas of practical reasoning, and this is indeed what we are aiming at with our formalisation. From a philosophical point of view the formalisation of the `commit_to` action as we present it, corresponds to the following syllogism:

> i knows that φ is one of its *goals*
> i knows that α is a *correct* and *feasible* action to achieve φ
> Therefore i has the *opportunity to commit* itself to α.

Obviously — given the relation between the past participle 'committed' and the infinitive 'to commit to' — performing `commit_to` α will result in the agent being committed to α, formalised through the $\mathbf{Committed}_i$ operator. The semantics for this operator should meet certain desiderata. It is for instance desirable that agents know of their commitments, and that agents have the practical possibility to perform the actions they are committed to. Furthermore some desiderata that concern composite actions should be met, the most obvious of which is one that concerns sequential compositions. For whenever an agent is committed to a sequential composition $\alpha_1; \alpha_2$ it should be committed to α_1 now, and to α_2 after α_1 has been performed. Another quite intuitive property concerns conditional compositions. It seems desirable that an agent that is committed to a conditional composition `if` φ `then` α_1 `else` α_2 `fi` is also committed to α_1 whenever it knows that φ holds. Likewise it should be committed to α_2 whenever it knows that φ does not hold[4].

The global idea behind the semantics that we define is that the commitments of an agent are recorded in its *agenda*. This agenda is defined to be a function that yields for a given agent in a state of a model the actions that the agent is committed to. Whenever an agent commits itself to an action its agenda is updated accordingly.

To ensure that the desiderata for commitments that concern composite actions are met, we make use of so-called *computation runs* in defining the semantics. A computation run of an action α for an agent i is a finite length string of atomic actions and confirmations, representing the sequence of atomic steps that constitutes the halting execution — whenever existing — of the event $\mathrm{do}_i(\alpha)$. As such, computation runs capture the *essence* of an event in a given state, i.e. whenever for two actions α and α' holds that the finite computation runs of $\mathrm{do}_i(\alpha)$ and $\mathrm{do}_i(\alpha')$ are equal in a state s, the actions are from i's viewpoint *practically identical*. By updating the agenda of agents with computation runs of actions rather than the actions themselves it is ensured that the $\mathbf{Committed}_i$ operator displays a desirable behaviour with regard to composite actions.

Definition 20. The class of basic actions Ac_b^C is the smallest superset of At such that
- if $\varphi \in \mathrm{L}_{\mathrm{I}}^C$ then `confirm` $\varphi \in \mathrm{Ac}_b^C$
- if $\alpha_1, \alpha_2 \in \mathrm{Ac}_b^C$ then $\alpha_1; \alpha_2 \in \mathrm{Ac}_b^C$

Actions that are either atomic or a confirmation, are called *semi-atomic.*

[4] The definitions of preferences and goals as given in the previous sections are in some aspects tailor-made for the desiderata imposed on the notion of commitment as formalised here. For instance, the combination of persistence of preferences with the accordance of actions ensures that agents can always fulfil their commitments.

Definition 21. The class $\mathbf{M}^C$ of models for L^C contains all tuples $\langle \mathrm{S}, \pi, \mathrm{R}, \mathtt{r}_0, \mathtt{c}_0, \mathrm{P}, \mathrm{Ep}, \mathrm{Agenda}\rangle$, where all elements but Agenda have their usual connotation, and this latter function is such that $\mathrm{Agenda}(i, s)$ records the commitments of agent i in state s. It is furthermore demanded for a model to be in $\mathbf{M}^C$ that the demands given in Def. 10 and Def. 17 are satisfied, i.e. preferences are ensured to persist and be known, and all actions are accordant.

Definition 22. The function CS, yielding the *finite computation sequences* of a given action, is inductively defined as follows.

$$\begin{array}{ll}
\mathrm{CS} & : \ \mathrm{Ac}_1^C \to \wp(\mathrm{Ac}_b^C) \\
\mathrm{CS}(\alpha) & = \{\alpha\} \text{ if } \alpha \text{ is semi-atomic} \\
\mathrm{CS}(\alpha_1; \alpha_2) & = \{\alpha_1'; \alpha_2' \mid \alpha_1' \in \mathrm{CS}(\alpha_1), \alpha_2' \in \mathrm{CS}(\alpha_2)\} \\
\mathrm{CS}(\mathtt{if}\ \varphi\ \mathtt{then}\ \alpha_1 & \\
\qquad\quad \mathtt{else}\ \alpha_2\ \mathtt{fi}) & = \mathrm{CS}(\mathtt{confirm}\,\varphi; \alpha_1) \cup \mathrm{CS}(\mathtt{confirm}\,\neg\varphi; \alpha_2) \\
\mathrm{CS}(\mathtt{while}\,\varphi\,\mathtt{do}\,\alpha\,\mathtt{od}) & = \cup_{k=1}^{\infty} \mathit{Seq}_k(\mathtt{while}\,\varphi\,\mathtt{do}\,\alpha\,\mathtt{od}) \cup \{\mathtt{confirm}\,\neg\varphi\}
\end{array}$$

where for $k \geq 1$

$$\begin{array}{ll}
\mathit{Seq}_k(\mathtt{while}\,\varphi\,\mathtt{do}\,\alpha\,\mathtt{od}) & = \{(\mathtt{confirm}\,\varphi; \alpha_1'); \ldots; (\mathtt{confirm}\,\varphi; \alpha_k'); \\
& \qquad\qquad \mathtt{confirm}\,\neg\varphi \mid \alpha_j' \in \mathrm{CS}(\alpha_1) \text{ for } j = 1, \ldots, k\}
\end{array}$$

Convention. From now on we assume that a projection function π_2, yielding the second element of a pair, is given. Furthermore, for reasons of convenience we introduce the abbreviation $\mathbf{CanG}_i(\alpha, \varphi)$ for $(\mathbf{Can}_i(\alpha, \varphi) \wedge \mathbf{K}_i\mathbf{Goal}_i\varphi)$.

Definition 23. For a model $\mathrm{M} \in \mathbf{M}^C$, a state s in M, an agent i, and an action $\alpha \in \mathrm{Ac}_1^C$, the *finite computation runs* of α for i in M, s are defined by:

$$\begin{array}{ll}
\mathrm{CR} & : \ \mathrm{A} \times \mathrm{Ac}_1^C \times \mathbf{M} \times \mathrm{S} \to \wp(\mathrm{Ac}_b^C) \\
\mathrm{CR}(i, \alpha, \mathrm{M}, s) & = \{\alpha' \in \mathrm{CS}(\alpha) \mid \mathtt{r}^{\mathrm{c}}(i, \alpha')(\mathrm{M}, s) \neq \emptyset\}
\end{array}$$

The binary relation $\models^C$ between a formula $\varphi \in \mathrm{L}^C$ and a pair M, s consisting of a model $\mathrm{M} \in \mathbf{M}^C$ and a state s in M is for φ a propositional symbol, a negation or a conjunction, an epistemic, doxastic, preference or implementability formula defined as $\models$ in Def. 3, Def. 8 and Def. 12. For φ a dynamic or an ability formula $\models^C$ is defined as $\models$ in Def. 3 where $\mathtt{r}$ and $\mathtt{c}$ are replaced by $\mathtt{r}^{\mathrm{c}}$ and $\mathtt{c}^{\mathrm{c}}$ respectively. The functions $\mathtt{r}^{\mathrm{c}}$ and $\mathtt{c}^{\mathrm{c}}$ are for atomic actions, confirmations, sequential compositions, conditional compositions and repetitive compositions defined as in Def. 3, and $\mathtt{r}^{\mathrm{c}}$ is for the $\mathtt{commit\_to}\,\alpha$ action defined by:

$$\begin{array}{l}
\mathtt{r}^{\mathrm{c}}(i, \mathtt{commit\_to}\,\alpha)(\mathrm{M}, s) = \emptyset \text{ if } \mathrm{M}, s \models^C \neg\mathbf{CanG}_i(\alpha, \varphi) \text{ for all } \varphi \in \mathrm{Ep}(i, s) \\
\mathtt{r}^{\mathrm{c}}(i, \mathtt{commit\_to}\,\alpha)(\mathrm{M}, s) = \mathrm{M}', s \text{ with } \mathrm{M}' = \langle \mathrm{S}, \pi, \mathrm{R}, \mathtt{r}_0, \mathtt{c}_0, \mathrm{P}, \mathrm{Ep}, \mathrm{Agenda}'\rangle \\
\text{where for all } s' \in [s]_{\mathrm{R}(i)}, \mathrm{Agenda}'(i, s') = \mathrm{Agenda}(i, s') \cup \mathrm{CR}(i, \alpha, \mathrm{M}, s') \\
\text{and for } \beta_1; \ldots; \beta_m = \mathrm{CR}(i, \alpha, \mathrm{M}, s') \text{ and } 1 \leq k \leq m - 1, s' \in [s]_{\mathrm{R}(i)} \\
\quad \mathrm{Agenda}'(i, \pi_2(\mathtt{r}^{\mathrm{c}}(i, \beta_1; \ldots; \beta_k)(\mathrm{M}, s'))) = \\
\quad \mathrm{Agenda}(i, \pi_2(\mathtt{r}^{\mathrm{c}}(i, \beta_1; \ldots; \beta_k)(\mathrm{M}, s'))) \cup \{\beta_{k+1}; \ldots; \beta_m\}
\end{array}$$

Def. 23 formalises the idea that an agent has the opportunity to commit itself to those actions that it knows to be correct and feasible plans for goals; attempted commitments to other actions are doomed to fail. Whenever an agent commits itself to a correct and feasible plan for some goal the agenda of the agent is updated as follows. In all of the agent's epistemic alternatives the appropriate computation run is recorded in the agent's agenda. Furthermore, future commitments — represented by suffixes of computation runs — are recorded in the appropriate agendas. Since all regular actions are accordant, it follows that agents can honour these future commitments in due time (cf. item 4 of Prop. 27).

Definition 24. For $\alpha, \beta, \gamma \in \mathrm{Ac}_b^C$ we define the relation Prefix $\subseteq \mathrm{Ac}_\mathrm{b}^\mathrm{C} \times \mathrm{Ac}_\mathrm{b}^\mathrm{C}$ as the smallest superset of (α, α) such that:

- if $(\alpha, \beta) \in$ Prefix then $(\alpha, \beta; \gamma) \in$ Prefix
- $(\alpha; \beta, \alpha; \gamma) \in$ Prefix iff $(\beta, \gamma) \in$ Prefix

The Prefix relation is in general denoted as a prefix relation.

Definition 25. For $\mathrm{M} \in \mathbf{M}^C$, s in M, $i \in \mathrm{A}$ and $\alpha \in \mathrm{Ac}_1^C$ we define:

$$\mathrm{M}, s \models^C \mathbf{Committed}_i\alpha \Leftrightarrow$$
$$\forall s' \in [s]_{\mathrm{R}(i)} \exists \beta \in \mathrm{CR}(i, \alpha, \mathrm{M}, s') \exists \gamma \in \mathrm{Agenda}(i, s')(\mathrm{Prefix}(\beta, \gamma))$$

Def. 25 formalises the idea that agents should begin at the very beginning (for that's always a good begin): agents are committed to all prefixes of their commitments. Thus an agent i is committed to an action α iff in each of its epistemic alternatives α is practically identical to a prefix of some action that is in the agent's agenda, i.e. the computation run of α equals this prefix.

Definition 26. The notion of *historical correctness* is inductively defined by:

- The model $\mathrm{M} = \langle \mathrm{S}, \pi, \mathrm{R}, \mathrm{r}_0, \mathrm{c}_0, \mathrm{P}, \mathrm{Ep}, \mathrm{Agenda} \rangle$ such that $\mathrm{Agenda}(i, s) = \emptyset$ for all $i \in \mathrm{A}$ and $s \in \mathrm{S}$ is historically correct.
- If M is historically correct then for all $i \in \mathrm{A}$, $\alpha \in \mathrm{Ac}_1^C$ and $s, s' \in \mathrm{S}$, if $\mathrm{M}', s = \mathrm{r}^c(i, \texttt{commit\_to}\,\alpha)(\mathrm{M}, s)$ then M′ is historically correct.
- No other models are historically correct.

The class of historically correct models is denoted by $\mathbf{M}_h^C$. Validity in the class of historically correct Kripke models is denoted by $\models_h^C$.

The definition of historical correctness formalises the intuition that subjects in an agent's agenda should be generated as the result of the agent committing itself; it is not acceptable that an agent's agenda contains subjects out of the blue that are not due to the agent having committed itself.

Proposition 27. *For $i \in \mathrm{A}$, $\alpha, \alpha_1, \alpha_2 \in \mathrm{Ac}_1^C$, and $\varphi \in \mathrm{L}_1^C$, $\psi \in \mathrm{L}^C$ we have:*

1. $\models^C \mathbf{Can}_i(\alpha, \psi) \wedge \mathbf{K}_i\mathbf{Goal}_i\psi \to \langle \mathrm{do}_i(\texttt{commit\_to}\,\alpha)\rangle \mathbf{Committed}_i\alpha$
2. $\models_h^C \mathbf{Committed}_i\alpha \to \mathbf{K}_i\mathbf{Committed}_i\alpha$
3. $\models_h^C \mathbf{Committed}_i\alpha \to \mathbf{Can}_i(\alpha, \top)$
4. $\models_h^C \mathbf{Committed}_i(\alpha_1; \alpha_2) \to \mathbf{Committed}_i\alpha_1$
5. $\models_h^C \mathbf{Committed}_i(\alpha_1; \alpha_2) \to \mathbf{K}_i\langle \mathrm{do}_i(\alpha_1)\rangle \mathbf{Committed}_i\alpha_2$
6. $\models_h^C \mathbf{Committed}_i(\texttt{if}\,\varphi\,\texttt{then}\,\alpha_1\,\texttt{else}\,\alpha_2\,\texttt{fi}) \wedge \mathbf{K}_i\varphi \to \mathbf{Committed}_i\alpha_1$
7. $\models_h^C \mathbf{Committed}_i(\texttt{if}\,\varphi\,\texttt{then}\,\alpha_1\,\texttt{else}\,\alpha_2\,\texttt{fi}) \wedge \mathbf{K}_i\neg\varphi \to \mathbf{Committed}_i\alpha_2$

8. $\models^C_h$ $\mathbf{Committed}_i(\texttt{while}\,\varphi\,\mathrm{do}\,\alpha\,\mathrm{od}) \wedge \mathbf{K}_i\varphi \to \mathbf{Committed}_i(\alpha;\texttt{while}\,\varphi\,\mathrm{do}\,\alpha\,\mathrm{od})$

The first item of Prop. 27 states that the opportunity of an agent to commit itself is determined by the preconditions given in our version of the practical syllogism. In item 2 it is stated that agents know of their own commitments, and item 3 states that agents know that they have the practical possibility to perform the actions they are committed to. Items 4 and 5 state that our notion of commitment meets the desiderata for sequential composition, i.e. if an agent is committed to a sequential composition $\alpha_1;\alpha_2$ it is also committed to α_1 now, and to α_2 in the state that results from performing α_1. Items 6 and 7 state that the desiderata for the conditional composition are also met: an agent that is committed to a conditional composition $\texttt{if}\,\varphi\,\texttt{then}\,\alpha_1\,\texttt{else}\,\alpha_2\,\texttt{fi}$ is also committed to α_1 whenever it knows that φ holds, and to α_2 whenever it knows that φ does not hold. Item 8 deals with the unfolding of while-loops under commitments. Note that the first item holds for general models, whereas all other items concern validity in the class of historically correct models.

6.1 The ability to commit

At least two meaningful interpretations of the ability of an agent to commit itself come into mind. The first of these interpretations corresponds to *one-track minded* agents, which can commit themselves to a single action only. Hence these agents must have empty agendas in order for them to be able to commit themselves. This intuitive idea can be formalised by defining $\mathrm{c}^{\mathrm{c}}(i, \texttt{commit\_to}\,\alpha)(\mathrm{M}, s)$ to be $\mathbf{1}$ iff $\mathrm{Agenda}(i, s) = \emptyset$, which implies that $\mathbf{Committed}_i\alpha_1 \to \neg\mathbf{A}_i\texttt{commit\_to}\,\alpha_2$ is a valid formula for all actions α_1, α_2, i.e. whenever agent i is committed to action α_1 it is not able to commit itself to action α_2. The $\texttt{commit\_to}$ action is furthermore *ability-destructive*, i.e. by making a commitment an agent destroys its ability to commit. Formally this amounts to the formula $[\mathrm{do}_i(\texttt{commit\_to}\,\alpha_1)]\neg\mathbf{A}_i\texttt{commit\_to}\,\alpha_2$ being valid.

The second interpretation of capabilities for commitments would be to emphasise the *moral* component of capabilities [25], i.e. the ability to commit expresses that it is within the *moral capacities* of an agent to perform some action. Given the intuitive interpretation of ability as given in Sect. 2, i.e. ability is the complex of physical, mental and *moral* capacities, this notion of ability to commit to an action is obviously related to the ability to perform the action itself. Even stronger, it seems reasonable to identify the *(moral) ability to perform* an action with the *ability to commit* to the action. For on the one hand, the ability to perform an action implies the moral ability to do so and hence the ability to commit to the action, while on the other hand agents should be able to commit to those actions only that are within their capacities. This leads to a definition of c^{c} such that $\mathrm{c}^{\mathrm{c}}(i, \texttt{commit\_to}\,\alpha)(\mathrm{M}, s)$ equals $\mathrm{c}^{\mathrm{c}}(i, \alpha)(\mathrm{M}, s)$. A consequence of this definition would be that whenever an agent knows that some action is a correct and feasible plan to bring about some goal, it knows that it can update its commitments to account for this plan, i.e. the formula $\mathbf{CanG}_i(\alpha, \varphi) \to \mathbf{Can}_i(\texttt{commit\_to}\,\alpha, \mathbf{Committed}_i\alpha)$ would be valid.

7 Discussion

In this paper we formalised various motivational attitudes of rational agents. At the level of assertions we defined preferences of agents. The awareness-based semantics for preferences avoids both the side-effect and the transference problem, and does furthermore not suffer from the problems of unrestricted combining and weakening. Goals are defined to be unfulfilled and realistic preferences. At the practition level we defined commitments. With respect to commitments we considered both a static and a dynamic aspect. The static aspect, represented through an action-parametrised operator, formalises the commitments that agents have made; the dynamic aspect, represented through an action-parametrised action, formalises the act of making commitments. The relation between the static and dynamic component of commitments is as one would expect: when an agent i (successfully) performs a `commit_to` α action this results in $\mathbf{Committed}_i\alpha$ being true. Agents are defined to have the opportunity to commit themselves to exactly those actions that constitute plans for goals, i.e. actions that are known to be correct and feasible to achieve some (known) goal. We discussed two possible ways to define the ability of agents to commit themselves, one in which agents are assumed to be one-track minded, the other in which the moral component of ability is stressed. Our formalisation of preferences and goals, as well as the static part of commitments, validates many of the desiderata for motivational attitudes. The dynamic part of commitment constitutes one of the novel aspects of this approach, which considerably enhances the expressiveness of our formal framework as compared to other formal approaches.

Future work on the topics discussed in this paper will concentrate on extending and refining the notion of commitment as it is considered here. In particular, we would like to incorporate a notion of *rational commitment*: agents commit themselves only if the benefits of such a commitment exceed its expenses; as soon as this no longer is the case agents should (have the opportunity to) drop the commitment. Incorporation of this notion of rational commitment will probably necessitate a considerable extension and revision of the formal framework.

Acknowledgements Thanks are due to the referees for their comments which helped improve this paper. This research is partially supported by Esprit BRWG project No.8319 'ModelAge' and the Vrije Universiteit Amsterdam; the third author is furthermore partially supported by the Katholieke Universiteit Nijmegen.

References

1. M.E. Bratman. *Intentions, Plans, and Practical Reason.* HUP, 1987.
2. H.-N. Castañeda. The paradoxes of deontic logic. In *New Studies in Deontic Logic*, pp. 37–85. Reidel, 1981.
3. P. Cohen and H. Levesque. Intention is choice with commitment. *AI*, 42:213–261, 1990.
4. R. Fagin and J. Halpern. Belief, awareness and limited reasoning. *AI*, 34:39–76, 1988.
5. J.Y. Halpern and Y. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *AI*, 54:319–379, 1992.
6. D. Harel. Dynamic logic. In D.M. Gabbay and F. Guenthner, eds, *Handbook of Philosophical Logic*, volume 2, chapter 10. Reidel, 1984.

7. J. Hintikka. *Knowledge and Belief.* Cornell University Press, 1962.
8. W. van der Hoek, B. van Linder, and J.-J. Ch. Meyer. A logic of capabilities. In Nerode and Matiyasevich, eds, *Proceedings of LFCS'94*, LNCS 813, pp. 366–378.
9. W. van der Hoek, B. van Linder, and J.-J. Ch. Meyer. Unravelling nondeterminism. In Jorrand and Sgurev, eds, *Procs. of AIMSA'94*, pp. 163–172. World Scientific, 1994.
10. Z. Huang, M. Masuch, and L. Pólos. ALX, an action logic for agents with bounded rationality. TR 92-70, CCSOM, 1992.
11. A. Kenny. *Will, Freedom and Power.* Basil Blackwell, Oxford, 1975.
12. Y. Lesperance, H. Levesque, F. Lin, D. Marcu, R. Reiter, and R. Scherl. Foundations of a logical approach to agent programming. This volume.
13. B. van Linder, W. van der Hoek, and J.-J. Ch. Meyer. Communicating rational agents. In Nebel and Dreschler-Fischer, eds, *Proceedings of KI-94*, LNCS 861, pp. 202–213.
14. B. van Linder, W. van der Hoek, and J.-J. Ch. Meyer. Tests as epistemic updates. In Cohn, ed., *Proceedings of ECAI'94*, pp. 331–335. John Wiley & Sons, 1994.
15. B. van Linder, W. van der Hoek, and J.-J. Ch. Meyer. The dynamics of default reasoning. In Froidevaux and Kohlas, eds, *Proceedings of ECSQARU'95*, LNCS 946, pp. 277–284.
16. B. van Linder, W. van der Hoek, and J.-J. Ch. Meyer. Actions that make you change your mind. In Wachsmuth and Rollinger, eds, *Proceedings of KI-95*, LNCS 981, pp. 185–196.
17. B. van Linder, W. van der Hoek, and J.-J.Ch. Meyer. Seeing is believing – and so are hearing and jumping. In Gori and Soda, eds, *Proceedings of AI*IA'95*, LNCS 992, pp. 402–413.
18. R.C. Moore. Reasoning about knowledge and action. TR 191, SRI, 1980.
19. T.J. Norman and D. Long. Alarms: An implementation of motivated agency. This volume.
20. A.S. Rao and M.P. Georgeff. Asymmetry thesis and side-effect problems in linear time and branching time intention logics. In *Procs of IJCAI91*, pp. 498–504, 1991.
21. A.S. Rao and M.P. Georgeff. Modeling rational agents within a BDI-architecture. In *Procs of KR'91*, pp. 473–484, 1991.
22. Y. Shoham. Agent-oriented programming. *AI*, 60:51–92, 1993.
23. M.P. Singh. *Multiagent Systems: A Theoretical Framework for Intentions, Know-How and Communications.* LNCS 799, Springer-Verlag.
24. M.P. Singh. Semantical Considerations on Some Primitives for Agent Specification. This volume.
25. B. Williams. Moral incapacity. In Procs of the Aristotelian Society, 93: 59–70, 1993.
26. M. Wooldridge. Time, Knowledge, and Choice. This volume.
27. G.H. von Wright. *Norm and Action.* Routledge & Kegan Paul, 1963.
28. G.H. von Wright. On so-called practical inference. In *Practical Reasoning*, OUP, 1978.

Rational Commitment in Resource Bounded Agents

Paul Dongha

Dept. of Computation, UMIST
Manchester, M60 1QD
United Kingdom

EMAIL dongha@sna.co.umist.ac.uk
TEL/FAX (+44 161) 881 6414

Cristiano Castelfranchi[*]

Istituto di Psicologia, CNR
Viale Marx, 15 - 00137
Roma - ITALY

EMAIL cris@pscs2.irmkant.rm.cnr.it
TEL (+39 6) 860 90 518
FAX (+39 6) 82 47 37

Abstract

The concept of *commitment* is widely seen as important in the design of deliberative AI agents that reason about intentions. We present a theory of commitment, based on notions of engagement and persistence, which we argue is rational for resource bounded agents to utilize in acting for their intentions. We claim that this theory avoids the irrationality inherent in human based commitment, whilst retaining some useful properties of micro-economic models of commitment. The new theory of commitment is formalized in an abstract branching time model, and properties of it are examined to demonstrate the dynamic nature of commitment. In addition to defining commitment, we also stipulate conditions for intention adoption, persistence and reconsideration.

1 Introduction

The importance of future-directed intentions in the ongoing practical reasoning of resource bounded agents has been recognized by a number of researchers [Bra87, Har86, CL90, SA91]. A crucial property of an intention is that it involves commitment, that is, an agent will persist in trying to achieve an intention despite partial knowledge of the world, computational resource limitations and the uncertainty in its environment. Although much work has attempted to formalize the semantics of intentions, only recently has it been recognized that utilizing commitments is potentially a rational way for AI agents to manage their limited reasoning capabilities [Sin91, KG91, Cas93, Don95, DC95].

In attempting to formulate a theory of rational commitment, we find economic models of rationality are infeasible because they assume perfect and unbounded computational abilities of the reasoner. In contrast, psychological studies show that humans approach commitment irrationally. AI theories of commitment do not provide a fully rational solution either. We propose a theory of commitment which we argue is rational, and which incorporates the salient properties of economic rationality, whilst avoiding some of the inconsistencies inherent in human based commitment. We take a first step toward a solution by presenting a logical model of rational commitment for resource bounded situated agents.

The approach in this paper is a step toward a theory of *rational* intention revision. That is, ultimately, this work will contribute toward a theory for managing the revision of intentions in practical reasoning. Commitment is a central property of intentions since it controls their persistence. If one accepts that commitments are a useful technique for resource bounded agents to employ — agents who are unable to assess the merits of their intentions at every moment — then an essential part of a theory of rational intention revision reduces to rationality requirements for commitment.

This paper is organized as follows. In §2 we review the uses of the term rationality in different disciplines. In §3 we examine the usefulness of commitment for resource bounded AI agents that utilize intentions as partial plans. In §3.1 we examine how human subjects use commitment in practical reasoning. Surprisingly we find that people generally approach commitment irrationally. In §4 we show

[*]Cristiano Castelfranchi's work is realized in the framework of the ESPRIT III Working Group No.8319: A Common Formal Model of Cooperating Intelligent Agents (ModelAge).

that commitment can be generated from notions of engagement and persistence. Using these notions, we claim that a theory of rational commitment can be constructed which avoids the irrationality which humans exhibit. This theory is based on a micro-economic theory of rationality involving a cost-benefit analysis. In §5 we introduce a logical model based on the branching time logic CTL^*. In §6 we show how the logical model can be used to capture properties of the theory of commitment from §4. We conclude with §7, and outline future areas of work in §8.

2 Use of Rationality

Rationality is used in a number of different fields of study; decision theory, philosophy, psychology and AI. Each of these fields uses the term in different ways. This section will give a brief overview of its different uses in these fields.

In the philosophical literature [Bra83], an agent is judged as rational in choosing an action if that action is believed by the agent to maximize long-term desire satisfaction. Bratman [Bra87, p. 68], judges an agent as rational in holding an intention if it was rational of the agent not to reconsider that intention between the time of its formation until the time of action. According to Bratman, an agent is rational in not reconsidering an intention if the habits and skills displayed in not reconsidering were in-line with the agent's long term interest in getting what it wants.

The economic theory of rationality is based on the subjective expected utility (SEU) model of decision making [Sav72]. In this context an agent is judged to be rational if it makes exactly those choices which would be recommended had an SEU model been applied in the same situation. Thus rationality is equated with the outcome of SEU decision making. Therefore SEU theory is useful as a normative theory in prescribing a standard for rationality. However, the SEU model has many properties that make it inappropriate as a model of decision making for resource bounded agents (see [Doy92, pp. 384–391] for a comprehensive discussion); perhaps the most damning criticism is that SEU theory assumes perfect world knowledge on the part of the reasoner.

Simon [Sim82] made the distinction between systems which engage in a reasonable amount of deliberation in order to select actions (procedural rationality) and systems which always select the most objectively rational action (substantive rationality). He concluded that designing a system which attempted to achieve substantive rationality is extremely unrealistic for anything but a simple class of environments, and thus argues for more realistic satisficing decision procedures.

The most prevalent view of rationality in the AI literature to date has been that of *epistemic* rationality. According to this view, reasoning is seen as a sequence of logical inferences from a set of axioms to a desired goal state, where rationality imposes constraints of either logical consistency, soundness or completeness of the agent's beliefs at every step of reasoning. Gärdenfors [Gär88], formulates an axiomatic theory to describe the relationships between updating beliefs when an agent is confronted with new information. Harman explains how humans approach belief revision [Har86], and argues that not only do they minimize the number of changes to their beliefs, but also they do so in a way which increases their overall coherence. Harman's concludes that this coherence theory is normatively correct. Both Harman's and Gardenfors' work describes rational belief revision. However, these logical approaches fail to say anything about how to control inferencing, or which inferences are useful and which are not, or why certain inferences should be preferred to others. That is, they say little about the *process* of practical reasoning itself, or of the quality of different types of reasoning.

More recently, Russell and Wefald [RW91], argue that reasoning about the process of reasoning is itself critical to intelligent behaviour, hence they coined the term *metareasoning* to mean exactly that. In this approach the object level decision problem is the problem of selecting which physical action the agent should perform. At the meta-level, actions are deliberation steps, and the metalevel decision problem is the problem of selecting which deliberation step to perform. Thus the agent must choose the deliberation step which will result in the greatest increase in value for the agent. Hence rationality is about controlling reasoning so that only the most useful reasoning is performed.

Tversky and Kahneman have examined the process of human decision making under uncertainty in game like situations [TK86]. The results of their experiments show that the process of human decision making in these situations departs significantly from the theory of SEU, and suggests that human decision making often leads to systematic irrational decisions caused by judgemental biases.

Simon has proposed a study of the role of emotions in reasoning. Simon argues that a key component of practical reasoning involves an agent diverting its attention and perceptual focus to those aspects of its environment which affect its current plans [Sim83]. A primary way of diverting attention is by triggering emotions. Thus, Simon argues, emotions play an important role in helping resource bounded agents achieve rational decision making. However, no formal analysis of emotion is provided.

It is apparent that rationality has different meanings in different fields of study. However, little attention has been given to the concept of commitment for agents that reason about intentions. In the next section we examine the benefits of commitment for resource bounded agents and argue for a theory of rational commitment.

3 Commitment in Resource Bounded Agents

Intentions involve commitment, in that an agent continues acting for an intention only as long as it remains committed to it [Bra87, Chap. 5]. This paper is concerned with the study of an agent being committed internally (as it were) to its own intentions.[1]

Singh [Sin91] and [KG91] were among the first to recognize the importance of commitment for resource bounded AI agents who use intentions as partial plans in practical reasoning. They argue that once an agent commits itself to an intention it can proceed to deliberate about how to achieve it, and perform actions for it, without the computational burden of reconsidering whether its still a worthwhile state to achieve. Intuitively, this makes good sense for resource bounded agents. We claim, in addition to this, there are other reasons why too much reconsideration should be avoided; (i) Due to incomplete knowledge, an agent can never be certain at every moment that its current intention is worth pursuing. Consequently, this will always be a reason for suspending execution of an intention, reconsidering it, or searching for better alternatives. However, without executing actions an intention can not be achieved. If an agent allocates all its processing time to reconsideration, never executing actions until its certain that those actions are its best choices, it will never achieve its intention. (ii) Even if reconsideration does recommend changing an intention or dropping an intention, there will be costs involved in changing and re-allocating internal and external resources. The notion of commitment therefore avoids the regress into repeated reconsideration.

However, while commitment brings benefits to the use of intentions in practical reasoning, it also presents a problem. An agent who is committed to an intention may continue performing actions for it to a point beyond which a theory of rational decision making would prescribe. Typically, this would happen when the cumulative cost of those actions involved in achieving an intention outweighs the benefits that the intention brings to the agent. This is a problem which has been empirically observed among human subjects, and is the topic of the next section.

3.1 Psychological Theory of Commitment

It has been empirically observed and reported in the psychological literature [Sta81, AB85] and [Baz90, Chap. 6], that human subjects display an approach to commitment which has been termed the *sunk cost* approach. According to these authors, sunk cost behaviour is manifested by an individual who becomes increasingly committed to a course of action on the basis of the amount of resources they have already invested in the course of action to date, *i.e.*, at any point in time their level of commitment is directly proportional to how much they have invested to date.

However, this model of commitment is fundamentally flawed — in fact its is *irrational* — since it forces an agent to lock itself into a particular course of action merely because it has already heavily invested in it, consequently leading to a situation of escalating commitment to a point beyond which a theory of rational decision making would prescribe. The result is a non-rational escalation of commitment: increased investment results in increased commitment which fuels further investment, which in turn results in increased investment, and so the cycle continues. Psychologists tell us that individuals are prone to such over-commitment for a number of social and psychological factors, primary among them is a need to justify the rationality of engaging in the course of action in the first instance. Clearly, irrationality is caused by individuals not recognizing that future costs are important in determining commitment — an approach which is very much recommended by models of economic rationality.

[1] At least one more notion of commitment in the AI literature has been coined; that of *social* commitment [Cas93, Sho90, Jen93]. Social commitment refers to a commitment made by an agent to another agent (or agents) regarding an intention, and would typically occur in a multi-agent cooperative problem solving situation. In this sense commitment is an obligation made by one agent to one or more agent(s).

In this paper, we wish to combine the advantages that commitment brings for resource bounded agents who utilize intentions, with the advantages that the economic theory of rationality brings to decision making. In doing this we wish to avoid the irrationality inherent in human commitment (sunk cost tendencies), and avoid the assumptions of perfect world-knowledge inherent in the economic theory of rationality.

4 A Theory of Rational Commitment

According to the psychological literature, commitment to an intention comprises two factors: *engagement* and *persistence*. Engagement is the amount of investment that an agent is willing to spend on achieving an intention, and is based on a subjective estimate of its likely resource cost combined with an estimate of the risk involved in achieving it. Persistence is the degree to which an agent continues trying to achieve an intention once it has started acting. These factors are subtly different, and we will examine each in turn.

4.1 Engagement

There are two factors which determine the level of engagement (or allocated resources) that an agent has for an intention:

Perceived Difficulty of the intention. An intention can be categorized as difficult in two different ways: (i) difficult as in costly, laborious or resource intensive; (ii) difficult as in risky, unlikely to succeed or improbable.

Utility assigned to the intention. This is the usefulness of achieving the intention for the agent.

The engagement for an intention is a function of its perceived difficulty, provided that this does not exceed its utility. In other words, the amount of resources an agent is willing to invest in achieving an intention is determined by its perceived difficulty; however, if this exceeds utility then the intention is not worth adopting, and there should be no engagement. Obviously, perceived difficulty is increased with an increase in cost and with an increase in risk. The greater the utility of the intention, the greater the possible engagement by the agent, up to the limit of the utility. Informally, a simple notion of engagement for an intention θ, can be expressed as;

$$Engagement(\theta) = \frac{Total\text{-}cost(\theta)}{1 - Risk(\theta)}$$

Here, $Risk(\theta)$ is similar to a "probability measure" in the unit interval $[0, 1]$, where 0 means low risk and 1 means high risk (see 5.2). However $Total\text{-}cost(\theta)$ will typically be a much higher value, since it encodes "units of resources" . Thus engagement is a combination of difficulty in terms of resource costs and difficulty in terms of risk. Although engagement is expressed as a very straightforward function, its behavior conforms to what we would expect; it is proportionately greater for a costly or risky intention.

For commitment, Bazerman recommends a micro-economic theory of rationality: "we should consider all alternative courses of action by evaluating only the *future* costs and benefits associated with each alternative." [Baz90, p.72]. At any point in time during the life of an intention, its future costs are those that will be incurred from this time point up to the achievement of the intention, and its future benefits constitute the utility of the intention. Following the micro-economic theory, we should compare future costs against future benefits in determining commitment to an intention. This simple notion of commitment for an intention θ is:

$$\begin{aligned} Commit(\theta) &= \frac{Utility(\theta) - Future\text{-}cost(\theta)}{Future\text{-}cost(\theta)} \\ &= \frac{Utility(\theta)}{Future\text{-}cost(\theta)} - 1 \qquad (1) \end{aligned}$$

Where, on adopting θ, (*i.e.*, before the agent begins acting for θ), $Future\text{-}cost(\theta)$ is assigned the value of $Engagement(\theta)$. An intention, viewed as a partial plan, is achieved by executing a series of actions.

where each action is a "means" to the intention. When the agent executes an action for an intention, its future cost is reduced by the cost of performing the action. Thus commitment will increase as actions are performed because *Future-cost*(θ) will reduce from its initial value of *Engagement*(θ). Also, quite correctly, commitment defined in this way makes no reference to "sunk costs". As an intention approaches completion, its commitment value will increase dramatically according to (1), and a high commitment value therefore is an indication of a high utility-to-future-cost ratio.

4.2 Persistence

An agent should remain committed to an intention only for as long as the sum of the cost all the executed actions is less than the intention's engagement value. In other words, an agent persists with an intention as long as it has not spent more than its engagement. However, an agent may reach the limit of its engagement value and still fail to achieve its intention. This is because, after all, engagement is based on subjective *estimates* of resource cost and risk. So an agent may well spend its entire engagement and still fail to achieve its intention. At this point the agent should reconsider whether the intention is still worth persisting with. The outcome of reconsideration will either result in the agent abandoning its intention, and accepting the loss of its invested resources, or result in re-commitment to the intention.

Reconsideration involves the agent deriving a new engagement and commitment value. The new engagement value will typically be less than the original engagement value, since some actions have already been performed (*i.e.*, *Future-cost*(θ) will be less than it was the first time round). Utility, however, remains unchanged. Thus, by (1), resulting in a higher commitment value than when the intention was first adopted. A higher commitment value makes the intention more attractive than when it was first adopted, and thus would be a good reason for increased persistence. So reaching the limit of the engagement value does not necessarily mean the agent should abandon its intention. Rather it is a reason for reconsideration, which may or may not result in increased persistence.

Recall that in the micro-economic theory of rationality, "sunk costs" should be ignored in deciding on a future course of action. In the process of reconsideration presented here, when a new commitment value is derived, the cost of actions already executed (*i.e.*, sunk costs), do not affect the new commitment value, and thus they do not enter into the reconsideration process.

5 The Formal Model

This section presents a formal model in which commitment can be expressed. It is based on the propositional branching-time logic CTL^* in [Eme90], and is a considerable extension of the work presented in [DC95].

5.1 Syntax

In the formal model we have three sorts; I (for intentions), A (for actions) and $\mathbb{Q}$ (the rationals). The set of all constants of sort S is $Const_S$, and the set of all variables of sort S is Var_S. The set of all terms of sort S, $(\langle term\rangle_S)$, and the set of formulae, $(\langle fmla\rangle)$, of our language are given by the following **BNF** grammar;

$$
\begin{array}{lcl}
\langle var\rangle_S & ::= & \text{any member of } Vars_S \\
\langle const\rangle_S & ::= & \text{any member of } Consts_S \\
\langle term\rangle_I & ::= & \langle var\rangle_I \mid \langle const\rangle_I \\
\langle term\rangle_A & ::= & \langle var\rangle_A \mid \langle const\rangle_A \\
\langle term\rangle_{\mathbb{Q}} & ::= & \langle var\rangle_{\mathbb{Q}} \mid \\
 & & Risk(\langle term\rangle_{\mathbb{Q}}) \mid Util(\langle term\rangle_{\mathbb{Q}}) \mid \\
 & & Act\text{-}Cost(\langle term\rangle_A) \mid \\
 & & Engagement(\langle term\rangle_I) \mid \\
 & & \langle term\rangle_{\mathbb{Q}} * \langle term\rangle_{\mathbb{Q}} \mid \\
 & & \langle term\rangle_{\mathbb{Q}} - \langle term\rangle_{\mathbb{Q}} \mid \\
 & & \langle term\rangle_{\mathbb{Q}} / \langle term\rangle_{\mathbb{Q}} \mid \\
 & & \langle term\rangle_{\mathbb{Q}} + \langle term\rangle_{\mathbb{Q}} \mid \\
\langle state\text{-}fmla\rangle & ::= & Commit(\langle term\rangle_I, \langle term\rangle_{\mathbb{Q}}) \mid \\
 & & Acts\text{-}for(\langle term\rangle_A, \langle term\rangle_I) \mid \\
 & & Achieved(\langle term\rangle_I) \mid \\
 & & \langle term\rangle_S = \langle term\rangle_S \mid \\
 & & \neg\langle state\text{-}fmla\rangle \mid \\
 & & \langle state\text{-}fmla\rangle \vee \langle state\text{-}fmla\rangle \mid \\
 & & \forall\langle var\rangle \cdot \langle state\text{-}fmla\rangle \mid \\
 & & \mathsf{A}\langle path\text{-}fmla\rangle \mid \\
\langle path\text{-}fmla\rangle & ::= & \langle state\text{-}fmla\rangle \\
 & & \neg\langle path\text{-}fmla\rangle \mid \\
 & & \langle path\text{-}fmla\rangle \vee \langle path\text{-}fmla\rangle \mid \\
 & & (\,\langle term\rangle_A\,) \mid \\
 & & \bigcirc\langle path\text{-}fmla\rangle \mid \\
 & & \Box\langle path\text{-}fmla\rangle \mid \\
 & & \langle path\text{-}fmla\rangle\, \mathcal{U}\, \langle path\text{-}fmla\rangle \\
\langle fmla\rangle & ::= & \langle state\text{-}fmla\rangle
\end{array}
$$

5.2 Semantics

Intuitively, the world can be in any one of a set of states W. The changing nature of the world is modeled by a transition from one state to another state, caused by the occurrence of an action. The agent can perform an action which causes a state transition. However, since we also want to capture the occurrence of actions which are not caused by the agent, (*e.g.*, changes in the agent's environment), we allow such actions to cause a state transition. Some of these actions may occur without the knowledge of the agent.

The relation R labels the transition from one state to another by an action. If $\sigma_1, \sigma_2 \in W$ and a is an action, then $(\sigma_1, a, \sigma_2) \in R$ means that the world moves from being in the state described by σ_1, to the state described by σ_2 by the occurrence of action a.

The domain of quantification for sort S is D_S, where S is either I, A or $\mathbb{Q}$. A model M is a tuple;

$$\langle W, \sigma_0, R, \mathbb{Q}, D_I, D_A, risk, util, comm, cost, acts\text{-}for, engage, achieved\rangle,$$

where;

- $W \neq \emptyset$ is a set of states.
- $\sigma_0 \in W$ is an initial starting state.
- $R \subseteq W \times D_A \times W$ is a reflexive labelled accessibility relation on states in W, such that at most one action occurs on a state transition, *i.e.*,

$$\forall\sigma, \sigma' \in W, \forall a, a' \in D_A,\ \text{if } (\sigma, a, \sigma') \in R \text{ and } (\sigma, a', \sigma') \in R \text{ then } a = a' \tag{2}$$

$$\forall\sigma, \exists\sigma' \in W, \exists a \in D_A \text{ s.t. } R(\sigma, a, \sigma') \tag{3}$$

Equation (2), says that if there is a transition from state σ to another state σ' then there is only one action which labels this transition. Condition (3) says that there is always somewhere else to go to from any state. This condition corresponds to the totality condition usually imposed on a branching time model. These two restrictions mean we have a model rooted at σ_0, which branches infinitely into the future.

- $\underline{\mathbb{Q}} = (\mathbb{Q}, +, -, \times, /, 0, 1)$, is the usual model of the rationals, and the domain of rationals is exactly the set $\mathbb{Q}$, *i.e.*, $D_{\mathbb{Q}} = \mathbb{Q}$.
- D_I, D_A are the domains of intentions and actions respectively.
- $risk : D_I \rightarrow [0, 1] \cap \mathbb{Q}$. For $\theta \in D_I$, $risk(\theta)$ returns a value in the rational unit interval $[0, 1] \cap \mathbb{Q}$. This function is intended to capture the agent's own subjective assessment of the likelihood of it achieving an intention. So, for an intention θ, $risk(\theta) \approx 1$ indicates that the agent considers it highly unlikely that it will achieve θ. At the other extreme, $risk(\theta) \approx 0$, means the agent considers it practically certain that it can achieve θ.[2] Risk is comprised of essentially two components: (i) the agent's measure of certainty about its ability to achieve an intention; and (ii) the agent's assessment of the inherent unpredictability of its environment, since a highly unpredictable environment will lead to greater uncertainty.
- $util : D_I \rightarrow [0, \infty) \cap \mathbb{Q}$, returns the agent's utility measure for an intention. The set $[0, \infty) \cap \mathbb{Q}$ is the set of positive values in $\mathbb{Q}$ excluding ∞, therefore *util* returns a finite rational value. This ensures that all intentions have a positive utility.
- $comm : W \rightarrow \wp(D_I \times \mathbb{Q})$, where $(\theta, n) \in comm(\sigma)$, means that in state σ, the agent is committed to intention θ with value n.
- $cost : D_A \rightarrow [0, \infty) \cap \mathbb{Q}$ returns the cost of performing a primitive action.
- $acts\text{-}for : \wp(D_A \times D_I)$, where $(a, \theta) \in acts\text{-}for$, means that the execution of action a contributes to the achievement of intention θ. That is a is one of the "means to" θ.
- $engage : D_I \rightarrow [0, \infty) \cap \mathbb{Q}$ returns an engagement value for an intention.
- $achieved : W \rightarrow \wp(D_I)$ where $\theta \in achieved(\sigma)$ means that at state σ the intention θ is achieved.

At any state there are many possible actions which may occur, therefore many resulting states. This gives us a branching structure of states connected by actions, and represents every possible way the world could develop. Each branch (also called a path) in the structure represents one possible way in which the world could develop, although in reality only one branch will be realized.

5.2.1 Term Semantics

The semantics of terms in our language are not state dependent therefore they are given relative only to a variable assignment. For each sort S, the function $V_S : Vars_S \rightarrow D_S$, gives an interpretation for variables of sort S. For ease of presentation we use the abbreviation $\vec{V} = V_I, V_A, V_{\mathbb{Q}}$ in the formal semantics. The function $[\![\tau_S]\!]_{\vec{V}} \in D_S$, gives the denotation of a term τ, of sort S, relative to $\vec{V}$. The formal semantics for terms in our language are given by the following:

$$\begin{aligned}
[\![\langle var\rangle_S]\!]_{\vec{V}} &= V_S(\langle var\rangle_S) \\
[\![\langle const\rangle_S]\!]_{\vec{V}} &= \lceil\langle const\rangle_S\rceil(\text{the model's interpretation of } \langle const\rangle_S) \\
[\![Risk(\langle term\rangle_I)]\!]_{\vec{V}} &= risk([\![\langle term\rangle_I]\!]_{\vec{V}}) \\
[\![Util(\langle term\rangle_I)]\!]_{\vec{V}} &= util([\![\langle term\rangle_I]\!]_{\vec{V}}) \\
[\![Act\text{-}Cost(\langle term\rangle_A)]\!]_{\vec{V}} &= cost([\![\langle term\rangle_A]\!]_{\vec{V}}) \\
[\![Engagement(\langle term\rangle_I)]\!]_{\vec{V}} &= engage([\![\langle term\rangle_I]\!]_{\vec{V}}) \\
[\![\langle term\rangle_{\mathbb{Q}} \times \langle term\rangle_{\mathbb{Q}}]\!]_{\vec{V}} &= [\![\langle term\rangle_{\mathbb{Q}}]\!]_{\vec{V}} \times [\![\langle term\rangle_{\mathbb{Q}}]\!]_{\vec{V}}
\end{aligned}$$

[2] Seen from another perspective, $risk(\theta)$, can be regarded as returning an "inverse probability measure". A value close to 1 means the agent considers it highly improbable that it can achieve θ, and a value close to 0 indicates the agent considers it highly probable that it can achieve θ. To avoid any confusion with probability measures, it is not the case that $\sum_{\theta \in D_I} risk(\theta) = 1$.

$$
\begin{aligned}
[\![\langle term\rangle_{\mathrm{Q}} - \langle term\rangle_{\mathrm{Q}}]\!]_{\vec{V}} &= [\![\langle term\rangle_{\mathrm{Q}}]\!]_{\vec{V}} - [\![\langle term\rangle_{\mathrm{Q}}]\!]_{\vec{V}} \\
[\![\langle term\rangle_{\mathrm{Q}} / \langle term\rangle_{\mathrm{Q}}]\!]_{\vec{V}} &= [\![\langle term\rangle_{\mathrm{Q}}]\!]_{\vec{V}} / [\![\langle term\rangle_{\mathrm{Q}}]\!]_{\vec{V}} \\
[\![\langle term\rangle_{\mathrm{Q}} + \langle term\rangle_{\mathrm{Q}}]\!]_{\vec{V}} &= [\![\langle term\rangle_{\mathrm{Q}}]\!]_{\vec{V}} + [\![\langle term\rangle_{\mathrm{Q}}]\!]_{\vec{V}}
\end{aligned}
$$

5.2.2 Formula Semantics

Before presenting the semantics of the formula in our language we introduce some notation:

- Let $p = \sigma_0 \xrightarrow{a_0} \sigma_1 \xrightarrow{a_1} \cdots$ be an infinite labelled path starting at σ_0 if and only if, $\forall i (\sigma_i, a_i, \sigma_{i+1}) \in R$. That is, a path in the model is defined as a sequence of states, where each state is connected by a transition labeled by an action. From now on, any reference to "path" is intended to mean "infinite path" unless stated otherwise.

- Let p_n, pick out the n^{th} state in a path rooted at a state indexed by 0, *i.e.*, $p_n = \sigma_n$ in a path $p = \sigma_0 \xrightarrow{a_0} \sigma_1 \xrightarrow{a_1} \cdots$.

- As in CTL^*, let $p^i = \sigma_i \xrightarrow{a_i} \sigma_{i+1} \xrightarrow{a_{i+1}} \cdots$, denote the suffix path. That is, p^i, returns the suffix path beginning at the ith state p_i.

- For some $\sigma \in W$, $\mathbf{S}_\sigma$ denotes the set of all labelled paths in $\langle W, R\rangle$ starting at σ. That is, $\mathbf{S}_\sigma$ is the set of all paths rooted at σ.

We say a formula is *satisfied* at a particular state $\sigma \in W$ if its true relative to a model M, variable assignment $\vec{V}$, and the state σ; $M, \vec{V} \models_\sigma \theta$, expresses "$M$, $\vec{V}$ satisfies θ at state σ". The semantics of state formulae in the formal language are given relative to a state σ as described above, and the semantics of path formulae are given relative to a path $p \in \mathbf{S}_\sigma$ rather than a state. We say a formula θ is *valid* if it is satisfied at all states in M, *i.e.*, "$M, \vec{V} \models \theta$". The formal semantics of state and path formulae are as follows;

$$
\begin{array}{ll}
M, \vec{V} \models_\sigma \neg\alpha & \text{iff } M, \vec{V} \not\models_\sigma \alpha \\
M, \vec{V} \models_\sigma \alpha \vee \beta & \text{iff } M, \vec{V} \models_\sigma \alpha \text{ or } M, \vec{V} \models_\sigma \beta \\
M, \vec{V} \models_\sigma \forall x_S \cdot \theta & \text{iff for any } v \in D_S\ M, \vec{V} \dagger \{x_S \mapsto v\} \models_\sigma \theta, \text{ where} \\
 & \vec{V} \dagger \{x_S \mapsto v\} = \begin{cases} V_I \dagger \{x_S \mapsto v\}, V_A, V_{\mathrm{Q}} & \text{if } x_S \in D_I \\ V_I, V_A \dagger \{x_S \mapsto v\}, V_{\mathrm{Q}} & \text{if } x_S \in D_A \\ V_I, V_A, V_{\mathrm{Q}} \dagger \{x_S \mapsto v\} & \text{otherwise} \end{cases} . \\
M, \vec{V} \models_\sigma \tau_1 = \tau_2 & \text{iff } [\![\tau_1]\!]_{\vec{V}} = [\![\tau_2]\!]_{\vec{V}} \\
M, \vec{V} \models_\sigma \mathsf{A}\theta & \text{iff } \forall p' \in \mathbf{S}_\sigma (M, \vec{V} \models_{p'} \theta) \\
M, \vec{V} \models_\sigma Acts\text{-}for(a, \theta) & \text{iff } acts\text{-}for([\![a]\!]_{\vec{V}}, [\![\theta]\!]_{\vec{V}}) \\
M, \vec{V} \models_\sigma Commit(\theta, n) & \text{iff } comm(\sigma)([\![\theta]\!]_{\vec{V}}, [\![n]\!]_{\vec{V}}) \\
M, \vec{V} \models_\sigma Achieved(\theta) & \text{iff } achieved(\sigma)([\![\theta]\!]_{\vec{V}}) \\
M, \vec{V} \models_p \theta & \text{iff } M, \vec{V} \models_{p_0} \theta \\
M, \vec{V} \models_p \neg\alpha & \text{iff } M, \vec{V} \not\models_p \alpha \\
M, \vec{V} \models_p \alpha \vee \beta & \text{iff } M, \vec{V} \models_p \alpha \text{ or } M, \vec{V} \models_p \beta \\
M, \vec{V} \models_p \langle a\rangle & \text{iff } R(p_0, [\![a]\!]_{\vec{V}}, p_1) \\
M, \vec{V} \models_p \bigcirc\theta & \text{iff } M, \vec{V} \models_{p^1} \theta \\
M, \vec{V} \models_p \Box\theta & \text{iff } \forall i (M, \vec{V} \models_{p^i} \theta) \\
M, \vec{V} \models_p \eta\,\mathcal{U}\,\theta & \text{iff } \exists j (M, \vec{V} \models_{p^j} \theta \text{ and } (\forall i < j \Rightarrow M, \vec{V} \models_{p^i} \eta))
\end{array}
$$

6 Modeling Commitment

As it stands the logical model is very general. However, to make it more specific we will introduce some extra logical apparatus which will constrain the class of our models to capture some of our earlier intuitions about commitment. Some relations we define will have a model interpretation that is state dependent (*e.g.*, see *Future-cost* below). So, we fix our attention to models satisfying the following. We introduce the less than, and greater than operators which we will use later on;

$$\begin{array}{ll} n \geq 0 & \Leftrightarrow \exists m \in \mathbb{Q} \cdot m \times m = n \\ n > 0 & \Leftrightarrow n \geq 0 \wedge n \neq 0 \\ n < 0 & \Leftrightarrow 0 \geq m \wedge n \neq 0 \\ n \leq 0 & \Leftrightarrow 0 > n \\ n \geq m & \Leftrightarrow n - m \geq 0 \text{ etc.} \end{array}$$

6.1 Engagement

Recall that the engagement for an intention is a combination of two factors; its total cost (the agent's estimate of the amount of resources required to achieve θ), and its risk (the agent's estimate of the likelihood that it can achieve θ). For $\theta \in D_I$, we set *total-cost*(θ) to the sum of the cost of all actions which contribute to achieving θ. Risk is derived from the function *risk* in our model. We then set *engage*(θ) to be simply *total-cost*(θ) multiplied by $(1 - risk(\theta))^{-1}$, so that engagement increases for a risky or costly intention. The following model restriction serves our purpose:

$$\begin{array}{l} \text{for } \theta \in D_I; \\ \quad \textit{total-cost}(\theta) = \sum_{a \in D_A \text{ s.t. } \textit{acts-for}(a,\theta)} \textit{cost}(a) \\ \text{and} \\ \quad \textit{engage}(\theta) = \textit{total-cost}(\theta) \times (1 - \textit{risk}(\theta))^{-1} \end{array} \tag{4}$$

6.2 Intention Adoption

On deciding to adopt an intention, the agent must ensure that its utility is greater than or equal to its engagement. In addition, an agent should not be committed to something which is achieved in the current state. A condition for adopting an intention can be defined as:

$$\textit{Adopt-cond}(\theta) \overset{\text{def}}{=} (\textit{Utility}(\theta) \geq \textit{Engagement}(\theta)) \wedge \neg \textit{Achieved}(\theta)$$

However, we can be more specific than this. Ideally, an agent would not adopt an intention solely because its utility is greater than its cost. We change the definition of *Adopt-cond*(θ) to be:

$$\textit{Adopt-cond}(\theta) \overset{\text{def}}{=} (\textit{Utility}(\theta) - \textit{Engagement}(\theta) \geq \pi) \wedge \neg \textit{Achieved}(\theta) \tag{5}$$

where $\pi \in \mathbb{Q}$ is some threshold value, internal to the agent, which is a lower limit on how worthwhile an intention has to be in order for it to be adopted. We do not require π to depend in any way on θ[3]; we simply want to stipulate that intuitively there should be some limit which the agent sets itself that indicates how worthy an intention should be for it to be adopted.

6.3 Introducing Commitment on Adoption

At the point an intention is adopted, it should have an associated future cost and commitment value. If θ is adopted, its future cost is set to the value of *Engagement*(θ). The relation *Future-cost*(θ, n), where $n \in D_{\mathbb{Q}}$ is the future cost of achieving $\theta \in D_I$. The commitment value of the intention is simply the ratio of utility to future cost. Also, an intention should only be adopted if its adoption condition is satisfied, thus we introduce the following definition of intention adoption:

$$\begin{array}{l} \textit{Adopt}(\theta) \overset{\text{def}}{=} \ \textit{Adopt-cond}(\theta) \wedge \left[\begin{array}{l} \bigcirc \textit{Future-cost}(\theta, \textit{Engagement}(\theta)) \wedge \\ \bigcirc \textit{Commit}(\theta, n) \end{array} \right] \\ \qquad\qquad \text{where } n = \textit{Utilty}(\theta) / \textit{Engagement}(\theta) \end{array} \tag{6}$$

This reads: an agent adopts θ if and only if its adoption condition is satisfied at the current state, then at the next state θ's future cost is assigned its engagement value, and its commitment value is assigned the ratio of utility to future cost.

[3] We accept that in some scenarios π might be dependent on θ, since an agent might be inherently more reluctant to adopt some intentions than others

For ease of presentation we introduce the following abbreviation, which says a happens for θ with cost l, whenever action a is the one executed on the current transition, and a is an action for intention θ whose cost of execution is l.

$$Happens(a,l,\theta) \overset{\text{def}}{=} (\langle a\rangle \wedge Acts\text{-}for(a,\theta) \wedge Cost(a) = l)$$

6.4 Changing Future Costs

We can now restrict our model, by the following axiom, which states that after performing an action for an intention, the future cost of it decreases by the cost of the action:

$$\begin{aligned} &\mathsf{A}\,\Box\forall\theta,m,n,a,l\cdot((Commit(\theta,m) \wedge Happens(a,l,\theta) \wedge Future\text{-}cost(\theta,n)) \\ &\quad\Rightarrow \bigcirc Future\text{-}cost(\theta,n-l)) \end{aligned} \tag{7}$$

6.5 Changing Commitment

We are now ready to describe how commitment values change for an intention. If an agent is committed to θ with value m, and action a happens, then at the next state, the commitment value for θ changes as described earlier by (1):

$$\begin{aligned} &\mathsf{A}\,\Box\forall\theta,m,n,a,l\cdot((Commit(\theta,m) \wedge Happens(a,l,\theta) \wedge \bigcirc Future\text{-}cost(\theta,n)) \\ &\quad\Rightarrow \bigcirc Commit(\theta,p)) \\ &\text{where} \\ &\ p = (Util(\theta)/n) - 1 \end{aligned} \tag{8}$$

Thus, as an agent performs actions toward an intention, its commitment to the intention increases. This conforms to our theory of commitment in §4; sunk costs are ignored, and commitment is based on a ratio of utility to future cost.

6.6 Properties of Commitment

Let us now examine commitment and future cost in more detail. Recall that the utility of an intention is given by a function, therefore by definition, the utility of an intention is constant for every state in the model. At every state transition where an action is performed for an intention, future cost and commitment change. Future cost is decreased by the cost of performing the action, and commitment is changed to be the ratio of utility to future cost. Consequently, the following hold in the model:

- If utility is greater than future costs, then commitment has a positive value:

$$\mathsf{A}\,\Box\forall\theta,n,l\cdot\left[\begin{array}{l} Future\text{-}cost(\theta,n) \wedge Commit(\theta,l) \wedge \\ Util(\theta) > n \end{array}\right] \Rightarrow l > 0 \tag{9}$$

- If utility equals future costs then commitment is zero:

$$\mathsf{A}\,\Box\forall\theta,n,l\cdot\left[\begin{array}{l} Future\text{-}cost(\theta,n) \wedge Commit(\theta,l) \wedge \\ Util(\theta) = n \end{array}\right] \Rightarrow l = 0 \tag{10}$$

- Let the future cost of some intention θ be n, and let a be the action performed with cost l, where $l \geq n$. Then at the next state, future cost of θ will be less than or equal to zero. In other words, the agent will have performed actions for the intention whose cumulative cost is greater than his initial engagement. The action a happens to be the last action which takes the future cost of θ to a negative value:

$$\mathsf{A}\,\Box\forall\theta,m,n,l\cdot\left[\begin{array}{l} Future\text{-}cost(\theta,n) \wedge Happens(a,l,\theta)) \wedge \\ l > n \wedge \bigcirc Future\text{-}cost(\theta,m) \end{array}\right] \Rightarrow m \leq 0 \tag{11}$$

- Since commitment is determined by future costs, its value will be less than zero, in fact it will be less than -1, precisely when future cost is less than zero;

$$\mathsf{A}\,\Box\forall\theta,m,n\cdot(Future\text{-}cost(\theta,m) \wedge Commit(\theta,n)) \Rightarrow (m < 0 \Leftrightarrow n < -1) \tag{12}$$

- If utility is less than future costs, then commitment has a value in the interval $(-1, 0)$. This interval excludes the value -1 and 0;

$$\mathsf{A}\,\Box\forall\theta, n, l \cdot \begin{bmatrix} \mathit{Commit}(\theta, l)\wedge \\ \mathit{Future\text{-}cost}(\theta, n)\wedge \\ \mathit{Util}(\theta) < n \end{bmatrix} \Rightarrow (-1 < l \wedge l < 0) \tag{13}$$

Assume $\mathit{Future\text{-}cost}(\theta, n)$, *i.e.*, the future cost of θ is n. Our model restriction for intention adoption (5) ensures $(\mathit{Util}(\theta) \geq n)$ at the state we first adopt θ, since $n = \mathit{Engagement}(\theta)$. Furthermore, since an intention always has constant utility, and by (7) future costs always decrease by the cost of an action executed for the intention, the condition $(\mathit{Util}(\theta) < n)$ should never actually hold in the model. Nonetheless, we have included (13) to make the analysis of changing commitment complete. If we incorporate agent perception into the model we can show that in fact $(\mathit{Util}(\theta) < n)$ can hold. In other words, perception can change the agent's estimate of future costs so that they may increase beyond the utility of the intention.

Table 1, summarizes the interaction between utility, future cost and its effect on commitment, and the action recommended by our model. We omit condition (13); for an analysis of it and the perception process see [Don96].

Relation of $\mathit{Utility}(\theta) = m$ to $\mathit{Future\text{-}cost}(\theta, n)$	Effect on $\mathit{Commit}(\theta, l)$	Action	Consequence
$m > n$	$l > 0$	persist with θ	Have not reached limit of engagement
$m = n$	$l = 0$	reconsider θ	Have reached limit of engagement
$n < 0$	$l < -1$	reconsider θ	Have reached limit of engagement

Table 1: Interaction between utility, future cost and commitment

6.7 Persistence

Recall that an agent should persist with an intention only as long as it is committed to it. In our model this holds by the following axiom, except we say that an agent persists with an intention only as long as it has a positive commitment value for it, and of course, it has not been achieved:

$$\mathit{Persist}(\theta) \stackrel{\text{def}}{=} (\mathit{Commit}(\theta, n) \wedge n > 0 \wedge \neg\mathit{Achieved}(\theta)) \tag{14}$$

6.8 Reconsidering

Reconsidering involves the agent determining whether it should re-commit to an intention, or abandon it. An agent should reconsider an intention when its commitment value for it is less than or equal to 0, and it is still not achieved.

$$\mathit{Reconsider}(\theta) \stackrel{\text{def}}{=} \mathit{Commit}(\theta, n) \wedge n \leq 0 \wedge \neg\mathit{Achieved}(\theta) \tag{15}$$

This conforms to our prescriptive theory which says an agent should reconsider an intention when it has spent its engagement value for it. The definition of reconsidering is exactly the inverse of persistence, thus they are both mutually exclusive.

The result of reconsideration is either that the intention is re-adopted at the next state, or its simply not adopted. The micro-economic theory of commitment recommends that an intention is adopted if, based on a cost benefit analysis, it is still worthwhile. So, we formalize this condition by the following axiom, which says that an agent should adopt an intention after reconsidering it only if the adoption condition is satisfied:

$$\mathsf{A}\,\Box\forall\theta \cdot (\mathit{Reconsider}(\theta) \Rightarrow (\mathit{Adopt\text{-}cond}(\theta) \Leftrightarrow \bigcirc\mathit{Adopt}(\theta))) \tag{16}$$

The use of the equivalence relation ($\Leftrightarrow$) means not only that an intention is adopted at the next state if its adoption condition is satisfied, but also that if an intention is adopted then at the previous state its adoption condition was satisfied.

Persistence and commitment can be succinctly related by the following axiom which hold in our model, and states that once an agent adopts an intention it persists with it until the agent reconsiders it or it is achieved:

$$\mathsf{A}\,\Box\forall\theta \cdot Adopt(\theta) \Rightarrow Persist(\theta)\,\mathcal{U}\,(Reconsider(\theta) \vee Achieved(\theta)) \tag{17}$$

This definition however permits an agent to persist with an intention indefinitely if $Reconsider(\theta) \vee Achieved(\theta)$ never hold. We want to disallow infinite persistence, so we add the following axiom to our model:

$$\mathsf{A}\,\Box\forall\theta Adopt(\theta) \Rightarrow \Diamond Achieved(\theta) \vee \Diamond Reconsider(\theta) \tag{18}$$

This states that if an intention is adopted then eventually its reconsidered, or eventually its achieved.

7 Concluding Remarks

In this paper we have argued that a theory of rational commitment should be based on micro-economic notions of risk, cost and utility. We have also presented a formal logical model of commitment for an intention which captures these notions, and their effect on the dynamic nature of commitment.

8 Future Work

Future work includes extending the logical model to address the following issues;

Risk. Refining the model with a more precise formalism of risk. For example, an agent's certainty about its own beliefs would effect its judgment of risk, and therefore its commitment value for an intention. An agent may also want to model the volatility of its environment in making assessments of risk. A highly volatile environment may make any intention more unstable, and therefore more risky.

Perception. A crucial part of any situated or autonomous agent is its capability to periodically perceive the environment, and for the results of perception to effect its beliefs [MK93, Gär88]. The result of perception may undermine the beliefs upon which existing intentions are based, such that their successful achievement becomes threatened. The volatility of the environment will effect how often perception should be performed; a highly volatile environment should lead to more frequent perception. Thus an agent should recognize the degree of (in)stability in its environment and manage the frequency of its perceptions accordingly. Although some very useful experimental results in this area have been reported in [KG91], we wish to concentrate on formalizing the perception process in our logical framework. For a description of how perception can be incorporated into the logical model see [Don96].

Utility The concept of utility has been formalized in a very straight forward manner, however, an investigation into some of the factors which determine utility in human decision making may be a fruitful area of work. In decision-theoretic reasoning, it may be captured more precisely using multi-attribute utility theory [KR76].

Cost We have presented a formalism which allows an agent to determine the cost of achieving an intention as the sum of the primitive actions which are "means" to it. More realistically however, deriving total cost presents a problem for resource bounded agents. Such agents plan in stages, and do not have in advance an exact plan for achieving an intention. The agent may well have an idea of which action(s) to perform initially on adopting an intention, but which primitive actions to perform later on will only become known to the agent nearer the time of their execution. We are investigating the use of fuzzy logic in decision making which might allow an agent to estimate cost based on likely future actions.

Acknowledgments
The authors would like to thank Carl Pulley for help with constructing the formal model, as well as Afsaneh Haddadi, Nick Jennings and Michael Wooldridge for reviewing an earlier draft of this paper. Thanks also to the reviewers for their useful comments.

References

[AB85] H. Arkes and C Blumer. The Psychology of Sunk Cost. *Organizational Behaviour and Human Decision Process*, 35:124–140, 1985.

[Baz90] M. Bazerman. *Judgement in Managerial Decision Making.* John Wiley and Sons, 2^{nd} edition, 1990.

[Bra83] R. Brandt. The Concept of Rational Action. *Social Theory and Practice*, 9(2-3):143–164, 1983.

[Bra87] M. Bratman. *Intention, Plans, and Practical Reason.* Harvard University Press, 1987.

[Cas93] C. Castelfranchi. Commitments: from individual intentions to groups and organizations. In *AAAI '93 Workshop on AI and Theories of Groups and Organizations*, 1993.

[CL90] P. Cohen and H. Levesque. Intention is Choice With Commitment. *Artificial Intelligence*, 42:213–261, 1990.

[DC95] P. Dongha and C. Castelfranchi. Rationality in Commitment. In *AAAI Fall Symposium on Rational Agency: Concepts, Theories, Architectures and Applications*, November 1995.

[Don95] P. Dongha. Toward a Formal Model of Commitment for Resource Bounded Agents. In *Intelligent Agents: Theories, Architectures and Languages — Proceedings of the ECAI '94 Workshop (LNAI Series. Vol 890).* Springer-Verlag, 1995.

[Don96] P Dongha. *In Preparation.* PhD thesis, Dept. of Computation, UMIST, Manchester, UK, February 1996.

[Doy92] J. Doyle. Rationality and its Role in Reasoning. *Computational Intelligence*, 8(2):376–409, 1992.

[Eme90] E. A. Emerson. Temporal and Modal Logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, pages 996–1072. Elsevier, 1990.

[Gär88] P. Gärdenfors. *Knowledge in Flux.* Bradford Books/MIT Press, 1988.

[Har86] G. Harman. *Change in View.* The MIT Press, 1986.

[Jen93] N.R. Jennings. Commitments and conventions: The foundation of coordination in multi-agent systems. *Knowledge Engineering Review*, 8(23):223–250, 1993.

[KG91] D. Kinny and M. Georgeff. Commitment and Effectiveness of Situated Agents. In *Proceedings of the 1991 International Joint Conference on Artificial Intelligence (IJCAI '91)*, pages 82–88. Morgan Kaufmann, 1991.

[KR76] R.L. Keeney and H. Raiffa. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs.* John Wiley & Sons, 1976.

[MK93] D. Musto and K. Konolige. Reasoning about Perception. *AI Communications*, 6(3–4):207–212, 1993.

[RW91] S. J. Russell and E. Wefald. *Do The Right Thing.* MIT Press, 1991.

[SA91] M. P. Singh and N. M. Asher. Towards a Formal Theory of Intentions. In *Logics in AI Proceedings of the European Workshop JELIA '90 (LNAI Volume 478).* Springer-Verlag, 1991.

[Sav72] L.J. Savage. *The Foundations of statistics.* Dover Press, New York, 2^{nd} edition, 1972.

[Sho90] Y. Shoham. Agent Oriented Programming. Technical Report STAN–CS–1335–90, Dept. of Computer Science, Stanford University, Cal.: USA, 1990.

[Sim82] H. A. Simon. *Models of Bounded Rationality*, volume 2, chapter Rational choice and the structure of the environment. MIT Press, 1982.

[Sim83] H.A. Simon. *Reason in Human Affairs*, chapter Alternative Visions of Rationality, pages 3–35 Cambridge University Press, 1983.

[Sin91] M. P. Singh. Intentions, Commitments and Rationality. In 13^{th} *Annual Conference of the Cognitive Science Society*, 1991.

[Sta81] B. M. Staw. The Escalation of Commitment to a Course of Action. *Academy of Management Review*, 6(4):577–587, 1981.

[TK86] A. Tversky and D. Kahneman. Judgement under uncertainty: Heuristics and biases. In H. Arkes and R. Hammond, editors, *Judgemnet and Decision Making*. Cambridge University Press, 1986.

A Planning Theory of Practical Rationality

John Bell
Applied Logic Group
Computer Science Department
Queen Mary and Westfield College
University of London
London E1 4NS
email: jb@dcs.qmw.ac.uk

Abstract

This paper combines ideas from recent philosophical work on rational agency and work in artificial intelligence on hierarchical non-linear planners which combine plan generation, execution, and monitoring, to produce a planning theory of rational agency for resource-bounded agents.

1 Introduction

This paper combines ideas from recent philosophical work on rational agency, notably that of Bratman [3; 4] and Pollock [8], and from work in artificial intelligence on hierarchical non-linear planners which combine plan generation, execution and monitoring (such as IPEM [1] and SIPE [10]) to produce a planning theory of rational agency for resource-bounded agents. We begin by proposing the following definition:

> A resource-bounded agent behaves rationally if it reasons and acts so as to achieve as many of its goals, in their comparative order of importance to the agent, as is possible given the resources available to it and the constraints in force.

We then outline a planning theory of practical rationality which realises it. Thus, in a slogan, practical rationality is reduced to a planning problem.

In order to place the theory in context we begin by sketching an architecture for a rational agent in Section 2. The theory is then outlined in Section 3 and discussed in Section 4. Some related work is discussed briefly in Section 5.

2 Architecture for a Rational Agent

The high-level functional architecture on which the discussion is based is pictured in Figure 1. A rational agent is embedded in the real world, and consists of two connected modules; a high-level (symbolic) reasoning system (or "mind"), and a low-level (procedural) action system (or "body").

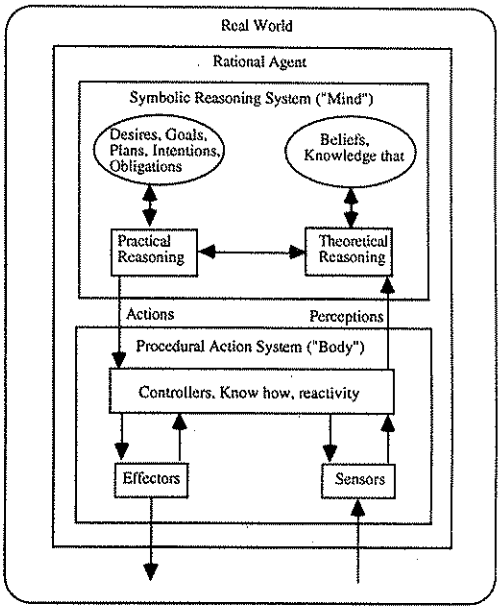

Real World
Rational Agent
Symbolic Reasoning System ("Mind")
Desires, Goals, Plans, Intentions, Obligations
Beliefs, Knowledge that
Practical Reasoning
Theoretical Reasoning
Actions
Perceptions
Procedural Action System ("Body")
Controllers, Know how, reactivity
Effectors
Sensors

Figure 1: Architecture for a Rational Agent

The symbolic reasoning system is composed of a module for theoretical reasoning and a module for practical reasoning, each with an associated database.

The theoretical reasoning module represents the agent's beliefs, knowledge and reasoning about the world. The reasoning done by this module includes standard deductive reasoning (traditionally called "theorem proving") as well as inductive, abductive, and probabilistic reasoning. It also performs database-type operations (lookup, update, addition, revision, deletion). The associated database may simply be a set of sentences, or it may have structure; e.g. the"web" of beliefs described in [9].

The practical reasoning module represents the agént's reasoning about what it should do. It consists of a high-level AI planning system, and is described in greater detail in the next section. It uses the theoretical reasoner to consult and update thebeliefs database, and may make further use of it if required.

The procedural action system (or "body") consists of controllers, sensors and effectors. This system represents the agent's physical capacities and skills ("know how"). The controllers control and monitor the I/O systems (the sensors and effectors) and mediate between the I/O systems and the reasoning systems. They receive high-level action commands from the practical reasoning system, expand the actions to the appropriate level of detail and execute and perform low-level monitoring on them by interacting with the I/O systems. They also pass perceptions (symbolic descriptions of the environment based on the feedback from the I/O systems) to the theoretical reasoning system.

The sensors include vision, audio and tactile systems. For example, the vision system might consist of cameras and many layers of software in order to interpret the camera images and to transform them into symbolic form. It would use low-level default reasoning (following visual cues, etc.). The effectors might include robot arms, tracks, etc.

The "mind" and "body" function as co-routines each of which is more or less active depending on the environment, the processing (reasoning) resources available, and the tasks in hand. This allows the agent to form long-term strategic plans (e.g. planning a shopping trip and the best order in which to visit shops), to execute them (often simultaneously with planning activities), and to react to events (e.g. when executing the shopping trip plan the agent forms and executes simple plans to avoid cars and people as it proceeds along the streets).

3 Rational Agency

The practical reasoning system is an idealised hierarchical non-linear planning system which integrates planning, execution and monitoring. Actual, but more limited, examples are IPEM and SIPE. The planner generates plans to achieve the agent's goals, schedules the resulting actions for execution, and monitors the outcome of their execution by the procedural action system.

Goals are rational desires; that is, desires which the agent believes to be possible and which the agent has chosen to pursue. The agent can have goals of many kinds. These include primitive goals (such as survival) and more advanced goals (such as being a good citizen, and getting a degree). Goals can be divided into maintenance goals (such as surviving, and continuing to watch a TV programme) and achievement goals (such as getting a degree, and opening a door).

Plan generation and execution is restricted by a number of constraints. These include physical constraints (e.g. that an object cannot be in two places at once) and normative constraints (such as the need to act safely, and the need to abide by legal or moral codes).

The plan generator produces plans by means of which the agent expects to achieve its goals. These will typically be hierarchical, non-linear and partial. The non-linearity of plans allows for the concurrent execution of actions (operators). Plans are generated hierarchically, at descending levels of abstraction. Plans are partial because the planner's ground level is an abstract level to the controllers and I/O devices; for example, unstacking block A from block B may be a ground level action to the planner, but is an abstract action to the action system as a much more detailed navigation plan has to be generated in order to guide the agent's arm. Typically plans will also be partial because of the need to integrate planning, execution and monitoring; thus plans may include unexpanded actions, information acquiring actions [1], etc.

The goal evaluation function evaluates goals on the basis of their expected utility to the agent. The definition of this function depends on an underlying value system, and may thus vary from agent to agent; an example is given in the next section.

At any stage the agent has an agenda of goals that it wishes to achieve. The scheduler decides which of these goals to pursue next; where 'pursue' may mean continue to generatea plan which achieves the goal, or it may mean execute (i.e. schedule for execution and pass to theaction system) the next (executable) action in the plan that has been generated to achieve the goal. In the simplest case the scheduler selects the highest priority goal (the most valuable goal according to the evaluation function) from the agenda which can be pursued (on which progress can be made). A more complex scheduling algorithm is discussed at the end of the next section. An action in a plan is considered to be executable if it is the first action that has not already been scheduled for execution, the preconditions of the action are satisfied, and the action violates none of the constraints currently in force.

4 Discussion

> Though this be madness, yet there's method in't. [Hamlet 2.2.205-6]

The theory makes a number of assumptions. First, it is stressed that we are interested in the problem of resource-bounded, rather than omnipotent, agents. The agent acts rationally if it acts so as to achieve as many of its goals, in their comparative order of importance to the agent, as is possible given the resources available to it and the constraints in force. The agent may not, for example, succeed in finding a plan in the time allowed because it runs out of reasoning resources or because it is pursuing more important goals. The definition of practical rationality may be considered to set an ideal standard for rationality for resource-bounded agents, which non-ideal resource-bounded agents like ourselves can only approximate (because of "human factors" such as weakness of will). Second, we are mainly concerned with high-level agents, such as ourselves, which act strategically. We are thus interested in agents of the kind which occur at the top of the hierarchy outlined by Pollock [8]; the most primitive agents are purely reactive, and there is a continuum between the two extremes. We have indicated (by means of shopping trip example and the agent architecture) that an advanced agent needs both strategic and reactive planning, and have suggested how they might be combined. Third, the theory is expressed in terms of the primitives of folk psychology, unlike Pollock's theory; this point is discussed further in the next section. Finally, the theory depends on the definition of the goal evaluation function, which may vary from agent to agent. This reflects the fact that practical rationality is a relative concept. For example, one society might consider the preservation of honour to be more important than the preservation of life, while in another the reverse may be the case. Faced with certain defeat, a soldier from the former society can rationally refuse to surrender and fight to the death, while one from the latter can rationally choose surrender instead.

Given these assumptions, what does the theory tell us about practical rationality? The theory has the right consequences when it comes to the question of rational persistence. According to Cohen and Levesque [5] an agent may give up (cease pursuing) a goal when (and only when) (1) the goal has been achieved, or (2) the agent believes that the goal is no longer achievable, or (3) the agent's reason for adopting the goal no longer holds. The example given on p. 47 of [2] shows that it is necessary to add that a goal may also be given up if (4) achieving it would violate one or more of the normative constraints in force. Finally, a goal may also be given up if (5) achieving it prevents the achievement of a more valuable goal. Clearly the theory satisfies these conditions. (1) When a goal is achieved, it is removed from the agenda. (2) If a goal is believed to be unachievable, the plan generation process will terminate. (3) If the agent's reason(s) for adopting the goal no longer hold, its utility will drop and the goal will no longer be scheduled. (4) The normative constraints are respected in plan generation and in execution. (5) The most valuable goal is pursued. The theory also has the property that the agent persists with its goal(s) by default, in the sense that it does not reconsider them unless it has to, and this is clearly desirable in resource-bounded agents [3; 4].

The theory can be further refined to yield even more purposeful behaviour. If a new goal arises which has a higher value (according to the evaluation function) than an incompatible goal for which planning is already under way, then the scheduler will suspend work on the old goal. This may not be desirable; for example, if the new goal is only marginally more valuable than the old one and considerable resources have been invested in pursuing the old one. To cater for such cases the scheduler has to be more selective. If a goal is already being pursued, the scheduler should also take the cost of re-planning into account [4], and this includes a rough estimation of the cost of "undoing" any commitments entered into in pursuing the old goal, it should also include the cost of the resources already invested in pursuing the old goal [6]. If the value of the new goal exceeds the value of the old one plus the cost of replanning, then the new goal should be pursued instead, otherwise the agent should persist with the old goal.

5 Related Work

The architecture outlined in Section 2 was inspired by those of SIPE and OSCAR [3], and has much in common with other layered architectures; for example, those discussed in [11].

A major contribution of Bratman's work is the view of intentions as parts of larger partial plans. While Bratman distances himself somewhat from the definite notions of plans and planning developed in AI, the theory outlined here stresses the AI perspective, and modulo the goal evaluation function, reduces practical rationality to a planning problem. On this view, the only difference between, say, fulfilling an intention and meeting an obligation, is that the latter will typically be given a higher value by the goal evaluation function, and will thus be given priority by the scheduler.

Our theory has many points of similarity with that proposed by Pollock [8]. It differs significantly in detail; for example in the view of planning. Also, Pollock's theory is intended to give a general account of rational agency, while ours is more anthropomorphic. For example, in our theory, desires are the primary motivators of action. They need not be rational, indeed an agent may have conflicting and even unrealisable desires. In order to lead to rational action the agent's desires have to be regulated by the constraints imposed by practical reasoning. By contrast, Pollock's agents have ersatz desires which they can adopt or drop according to whether it is rational to hold them. Real desires may not be necessary in a rational agent; e.g., Pollock bases his theory on situation-likings. However it may be that desires play an important role in rationality; e.g. in (non-)persistence. The same applies a fortiori to obligations, which also play an important role in constraining the planning process. The closer approximation to human rationality is also useful when it comes to building agents which will interact with humans.

6 Concluding Remarks

We have outlined an AI-planning theory of practical rationality in resource-bounded agents. The theory is intended as a prolegomenon to formalisation and implementation. The beginnings of the formal theory are presented in [2].

This work forms part of the RED project and is supported by the United Kingdom Science and Research Council under grant number GR/H12348.

References

[1] J. Ambros-Ingerson and S. Steel. Integrated Planning Execution and Monitoring. Proceedings AAAI'88, pp. 83-88.

[2] J. Bell. Changing Attitudes. In: [11] pp. 40-55.

[3] M.E. Bratman. Intention, Plans and Practical Reason. Harvard University Press, Cambridge Massachusetts 1988.

[4] M.E. Bratman. Planning and the Stability of Intention. Minds and Machines 2, pp 1-16,1992.

[5] P. Cohen and H. Levesque. Intention is Choice with Commitment. Artificial Intelligence 42 (1990) pp. 213-261.

[6] P. Dongha. Toward a Formal Model of Commitment for Resource Bounded Agents. In: [11], pp. 86-101.

[7] J.L. Pollock. *How to Build a Person.* MIT Press, Cambridge Massachusetts, 1989.

[8] J.L. Pollock. The Phylogeny of Rationality. Cognitive Science 17 (1993) pp. 563-588.

[9] W.V.O. Quine. Two Dogmas of Empiricism. In: From a Logical Point of View. Harvard University Press, Cambridge, Massachusetts, 1953.

[10] D. Wilkins. *Practical Planning.* Morgan Kaufmann, San Mateo, California, 1988.

[11] M.J. Wooldridge and N.R. Jennings (Eds.). *Intelligent Agents.* Proceedings of the ECAI'94 Workshop on Agent Theories, Architectures, and Languages. Springer Lecture Notes in Artificial Intelligence, No. 890. Springer, Berlin, 1995.

A Unified Framework for Hypothetical and Practical Reasoning (1): Theoretical Foundations

S. K. Das[1], J. Fox[2], P. Krause[2]

[1] William Penney Laboratory, Imperial College, London SW7 2AZ
[2] Advanced Computation Laboratory, ICRF, London WC2A 3PX

Abstract. We describe here a general and flexible framework for decision making which embodies the concepts of beliefs, goals, options, arguments and commitments. We have employed these concepts to build a generic decision support system which has been successfully applied in a number of areas in clinical medicine. In this paper, we present the formalisation of the decision making architecture within a framework of modal propositional logics. A possible-world semantics of the logic is developed and the soundness and completeness result is also established.

1 Introduction

A *decision* is a choice between two or more competing hypotheses about some world or possible courses of action in the world. A *decision support system* is a computerised system which helps decision makers by utilizing knowledge about the world to recommend beliefs or actions [1]. Such a system when built on *symbolic theory* [4] offers a general and flexible framework for decision making. The theory embodies the concepts of beliefs and goals of decision makers, represents options to satisfy a goal, argues for and against for every option and commits to a suitable option. We have employed this theory to build a generic decision support system [9, 13]. A paper focussing on applications of the theory to clinical medicine has also been submitted to this conference [12].

The scope of our decision framework is illustrated by the simple medical example in Figure 1. We should emphasise at this stage that the medical content of the examples in this paper has been simplified to avoid the need to explain complex medical terminology. Suppose that a patient presents complaining of serious and unexplained loss of weight. As an abnormal and potentially serious condition a decision has to be taken as to the most likely cause of the complaint. In the graphical notation in the figure circles are used to represent decisions; this diagnosis decision is on the left of the figure. To make the decision we have to identify all potentially relevant causes of weight loss (such as cancer and peptic ulcer) and then generate and evaluate arguments for and against each of these candidates. Arguments will be based on relevant information sources such as the patient's age, history, symptoms and so on. Suppose, after evaluating the arguments, we take the decision that cancer is the most likely cause of the weight loss - i.e. we commit to acting on this conclusion.

Now we must take another decision, about the appropriate therapy. Suppose the possible candidates for this are chemotherapy and surgery. As before, arguments for and against these options need to be considered, taking into account such information as the patient's age (if a patient is elderly this may argue against more aggressive forms of treatment), the likely efficacy, cost and side-effects of each therapy etc. Finally, suppose that after weighing up the arguments we conclude that chemotherapy is most appropriate for the patient. Once again we must be careful about committing to this option since once an action is taken it cannot be reversed.

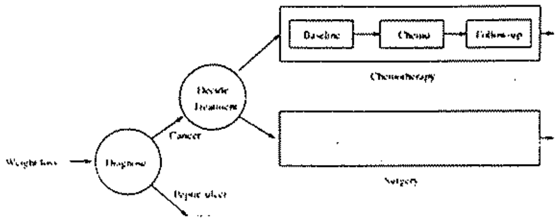


Fig. 1. An example of decision making.

A system which implements this model operates as follows (refer to Figure 2). First it maintains a database of beliefs about a particular situation; in a medical context this may include a set of clinical data about a patient for example. Certain beliefs (e.g. unexplained weight loss) cause the system to raise problem goals (e.g. to explain the weight loss). Such goals lead to problem solving to find candidate solutions (e.g. the weight loss may be caused by cancer or peptic ulcer) and arguments are constructed for and against the candidates, perhaps by instantiating general argument schemas with specific patient data and specific knowledge and beliefs about cancer, ulcers and so forth. As additional patient data are acquired a point may arise where an assessment of the various arguments for and against the various candidates permits the system to commit to a single hypothesis (e.g. cancer). This is adopted as a new belief which, while the belief is held, guides further problem solving and action. Since the belief concerns an undesirable - indeed life threatening - condition a new goal is raised, to decide on the best therapy for the patient. This initiates a further cycle of reasoning summarised in the left half of the figure. As before, candidate decision options are proposed (surgery, chemotherapy etc) and arguments are generated for and against the alternatives. In due course a commitment is made to a single therapy (e.g. chemotherapy).

Many clinical therapies, such as chemotherapy, are in fact complex procedures executed over time. Such therapies can usually be modelled as hierarchical plans that decompose into atomic actions (e.g. administer a drug) and subplans (e.g. take baseline measurements, administer several cycles of therapy, and then follow up the patient for a period after the last cycle of treatment). Our framework acknowledges this by providing formalisms for representing plans and for specifying the control processes required during plan execution. In particular the atomic actions of a plan must be scheduled with respect to any other actions which have been previously scheduled as a result of problem-solving or decision processes required for achieving the goal, or other goals raised by the system.

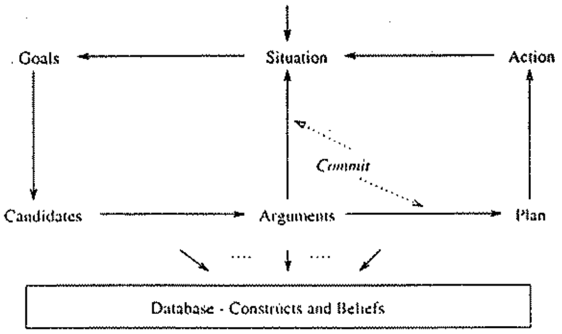


Fig. 2. The decision making architecture - Domino Model

We have proposed a high-level and expressive language called R^2L (RED Representational Language) [8] to be used by knowledge engineers who wish to encode knowledge and beliefs of a particular application domain for decision making based on the above scheme. R^2L explicitly supports the key concepts required in our framework, such as beliefs, goals, arguments and commitments. Our approach to formalisation of the above decision making scheme is by providing a sound translation mechanism of R^2L structures into a lower level but more general language which is the language of the logic LR^2L. The logic LR^2L enriches the propositional calculus by the introduction of a number of appropriately specialized modal operators and its semantics are well defined. This kind of modal formalism has the major advantage that its possible-world semantics [2, 16, 17] reflects the dynamic nature of applications such as medicine. In the implementation, decision making and scheduling processes are performed by an LR^2L theorem prover.

The paper is organized as follows. We present the R^2L language in the following section 2. This incorporates a model of argumentation in which arguments may be annotated with some qualifier indicating the confidence in a line of argument. These qualifiers may be drawn from one of a number of numeric or symbolic dictionaries, and then aggregated to provide an overall indication of the support for a given proposition. Section 3 describes the aggregation of arguments. The LR^2L syntax for representing theories of properties and of actions is presented in section 4 which is followed by the section describing the translation mechanism from R^2L to LR^2L syntax. The axioms and possible-world semantics of LR^2L are presented in sections 6 and 7, respectively. To illustrate and motivate the abstract model LR^2L, we provide a medical example in section 8. The soundness and completeness result is established in section 9. The proofs of the propositions and theorems can be found in the longer version of the paper.

2 R^2L

A decision schema in R^2L corresponding to "Diagnose" in figure 1 is represented as follows:

```
decision:: weight_loss_diagnosis
  situation  weight_loss
  goal       weight_loss_diagnosed
  candidates cancer;
             peptic_ulcer
  arguments  elderly => support(cancer, d1);
             smoker => support(cancer, d2);
             positive_biopsy => support(cancer, d3);
             pain_in_upper_abdomen => support(peptic_ulcer, d4);
             young => support(~cancer & ~peptic_ulcer, d2);
             pain_after_meals=>support(cancer & peptic_ulcer, d5);
  commits    netsupport(X, M) & exceed_threshold(M) => add(X).
```

In this example the decision `weight_loss_diagnosis` is considered for activation when the *situation* `weight_loss` is observed. A *goal* is considered as a property to be brought about. An *argument* schema is like an ordinary rule with `support(F, <degree>)` as its consequent, where `<degree>` is drawn from a dictionary of qualitative or quantitative measures of support, and indicates the support conferred on this candidate by the argument [15].

A commitment rule will often, though not necessarily, make use of the meta-predicate `netsupport` such as above. This meta-predicate has the general form `netsupport(F, <support>)`. It computes the total support for the specified candidate using an aggregation algorithm (discussed in the following section) selected from a library of such algorithms [15]. When a commitment rule involves the `netsupport` as in the above example, the computation of its truth value involves meta-level reasoning whose formalisation is beyond the scope of this paper.

A *dictionary* is a set of symbols which can be used to label a proposition. In general, a dictionary will be a semi-lattice with the partial order relation $\leq$. For simplicity, we shall consider a dictionary as a chain with one distinguished element Δ known as the *top element*. Let $d1$ be an element from some dictionary. Then the argument `elderly => support(cancer, d1)` specifies that if a person is believed to be elderly then this argument confers evidence level $d1$ on the candidate `cancer`. We might consider $d1$ as a member of the quantitative dictionary of probabilities $dict(Prob) =_{def} [0, 1]$. However, there is no requirement that we should restrict dictionaries to $dict(Prob)$. Among the obvious dictionaries we may consider is $dict(Qual) =_{def} \{+, ++\}$. As mentioned, a dictionary has always a top element to represent the highest support for arguments. For example, elements $++$ and 1 are the top elements of the two dictionaries $dict(Qual)$ and $dict(Prob)$ respectively. A number of different dictionaries for reasoning under uncertainty have been discussed in [15], together with their mathematical foundations and their relation to classical probability and other uncertainty formalisms.

A *commitment rule* is like an ordinary rule with one of add(<property>) or schedule(<plan>) as its consequent. The former adds a new belief to the database and the latter causes a plan to be scheduled as in the following construct:

```
decision:: cancer_treatment
  situation  cancer
  goal       cancer_cured
  candidates chemotherapy;
             surgery
  arguments  elderly => support(chemotherapy, +);
             young => support(surgery, +)
  commits    netsupport(X, M) & netsupport(Y, N) & M > N =>
                 schedule(X).
```

When we are committed to scheduling a plan, the plan involves executing constituent subplans and actions in a certain order. For example, if we schedule chemotherapy then a *plan construct* [8] of R^2L guides us to achieve the goal of carrying out the chemotherapy plan. Formalisation of this aspect involving actions and temporal reasoning is presented elsewhere [9] and is out of the scope of this paper.

3 Argumentation and Aggregation

In classical logic an argument is a sequence of inferences leading to a conclusion. The usual interest of the logician is in procedures by which arguments may be used to establish the validity (truth or falsity) of a formula. In LA, a logic of argument [15], arguments do not necessarily prove formulae but may merely indicate support for (or doubt about) them. Also in classical logic, so long as we can construct *one* argument (proof) for F, any further arguments for F are of no interest. In our system all distinct arguments of candidates are of interest (intuitively, the more arguments we have for F the greater is our knowledge about the validity of F). We therefore distinguish distinct arguments by identifying the unique grounds of each (essentially a normalised proof term in LA [15]) and a sign drawn from some dictionary which indicates the support provided to F by the argument. An example of an R^2L argument is elderly => support(cancer, d1), where F is cancer, the *ground* is elderly => cancer and the support is $d1$.

Suppose a decision maker has a set of arguments for and against a set of mutually exclusive decision options ("candidates", that is, alternative beliefs or plans under consideration) whose signs are drawn from a single dictionary. The decision maker can *aggregate* these arguments to yield a sign representing the decision maker's overall confidence in each of the candidates. Every dictionary has a characteristic aggregation function for aggregating arguments. Consider the argument presented above and positive_biopsy => support(cancer, d3).

Considering the dictionary as $dict(Prob)$, the two arguments can be aggregated by using a special case of Dempster's epistemic probability giving the value $d1 + d3 - d1 \times d3$. This formula can be generalised incrementally if there are more than two arguments for the candidate cancer.

In general, suppose a decision maker has a set of arguments for and against a set of mutually exclusive decision options, C, (candidates, that is, alternative beliefs or plans under consideration) whose signs are drawn from a single dictionary D. The decision maker can *aggregate* these arguments to yield a sign drawn from D' which represents the decision maker's overall confidence in each C. The general form of an aggregation function is as $\mathcal{A} : \Pi(C \times G \times D) \to C \times D'$, where Π stands for "power set" and G is the set of all grounds. The simple `netsupport` predicate in R^2L implements the function $\mathcal{A}$. If D is $dict(Qual)$ then D' is the set of non-negative integers whereas D' is D itself when we consider $dict(Prob)$ as D. In the former case, $\mathcal{A}$ assigns an aggregation number to each decision option, giving a total preference ordering over the options. This suggests a simple rule for taking a decision; choose the alternative which maximises this value.

If we allow both F and ~F to occur in the support then by applying our usual aggregation algorithm we compute total evidence for F (say, $d1$) and ~F (say, $d2$) separately. If we have used the dictionary $\{+, ++\}$ then we have the following four cases:
- total evidence for F is $d1 - d2$ if $d1 > d2$;
- the total evidence for ~F is $d2 - d1$ if $d2 > d1$;
- dilemma if $d1 = d2$
- inconsistent if $d1 = d2$.

If we have used the dictionary $dict(Prob)$ then we have the following cases:
- total evidence for F $= d1(1 - d2) \div (1 - d1 \times d2)$;
- total evidence for ~F $= d2(1 - d1) \div (1 - d1 \times d2)$;
- dilemma if $d1 = d2$;
- inconsistent if $d1 \times d2 = 1$.

4 Syntax of LR^2L

The language of LR^2L is the usual propositional language extended with a few modal operators introduced as follows. Suppose $\mathcal{P}$ is the set of all propositions which includes the special symbol $\top$ (true). Suppose D is an arbitrary dictionary with the the top element Δ. The modal operators of LR^2L corresponding to belief and goal are $\langle bel \rangle$ and $\langle goal \rangle$ respectively. In addition, for each dictionary symbol $d \in D$, we have a modal operator $\langle sup_d \rangle$ for support. The *formulae* (or *assertions*) of LR^2L are as follows:
- propositions are formulae.
- $\langle bel \rangle F$ and $\langle goal \rangle F$ are formulae, where F is a formula.
- $\langle sup_d \rangle F$ is a formula, where F is a formula and d is in the dictionary D.
- $\neg F$ and $F \wedge G$ are formulae, where F and G are formulae.

We take $\bot$ (false) to be an abbreviation of $\neg\top$. Other logical connectives are defined using '$\neg$' and'$\wedge$' in the usual manner.

5 Translating R^2L to LR^2L

This section details how R^2L constructs can be translated into sentences of the base language LR^2L. First of all, if the situation in a decision construct is believed then the corresponding goal is raised. Thus the situation and goal portion in decision *weight_loss_diagnosis* is translated to the rule
$\langle bel\rangle weight\_loss \rightarrow \langle goal\rangle weight\_loss\_diagnosed$
For any particular situation a raised goal is considered as achieved if it is true. The raised goal from a decision construct is true if any possible situation for the candidates is believed. In the context of the decision *weight_loss_diagnosis*, this is reflected in the following formulae:
$\langle bel\rangle(cancer \wedge \neg peptic\_ulcer) \rightarrow weight\_loss\_diagnosed$
$\langle bel\rangle(peptic\_ulcer \wedge \neg cancer) \rightarrow weight\_loss\_diagnosed$
$\langle bel\rangle(\neg cancer \wedge \neg peptic\_ulcer) \rightarrow weight\_loss\_diagnosed$
$\langle bel\rangle(cancer \wedge peptic\_ulcer) \rightarrow weight\_loss\_diagnosed$
Note that $\langle bel\rangle weight\_loss\_diagnosed$ can be derived from the first of the above four. where $\langle bel\rangle cancer$ and $\langle bel\rangle\neg peptic\_ulcer$ are true. The equivalent LR^2L representations of all the arguments in decision *weight_loss_diagnosis* are given below:
$\langle bel\rangle(weight\_loss \wedge elderly) \rightarrow \langle sup_{d1}\rangle cancer$
$\langle bel\rangle(weight\_loss \wedge smoker) \rightarrow \langle sup_{d2}\rangle cancer$
$\langle bel\rangle(weight\_loss \wedge positive\_biopsy) \rightarrow \langle sup_{d3}\rangle cancer$
$\langle bel\rangle(weight\_loss \wedge pain\_in\_upper\_abdomen) \rightarrow \langle sup_{d4}\rangle peptic\_ulcer$
$\langle bel\rangle(weight\_loss \wedge young) \rightarrow \langle sup_{d3}\rangle(\neg cancer \wedge \neg peptic\_ulcer)$
$\langle bel\rangle(weight\_loss \wedge pain\_after\_meals) \rightarrow \langle sup_{d5}\rangle(cancer \wedge peptic\_ulcer)$
If `support(cancer & peptic_ulcer, d)` holds then there is support d for each of `cancer` and `peptic_ulcer`. The converse is not necessarily true.

The version of LR^2L presented here excludes temporal reasoning and reasoning with actions; otherwise, the set of propositional symbols would have been divided into properties (e.g. the patient is elderly) and actions (e.g. the patient is given chemotherapy). A plan construct in R^2L would have been transformed to a set of temporal rules of LR^2L [9] involving action symbols. Reasoning with time and actions is out of scope of the current, as we are concentrating primarily on the process of decision making.

6 Axioms of LR^2L

We consider every instance of a propositional tautology to be an axiom. Instances of propositional tautologies may involve any number of modal operators, for example, $\langle bel\rangle p \rightarrow \langle bel\rangle p$. We have the modus ponens inference rule and adopt a set of standard axioms of beliefs which can be found in [5, 11, 14, 18]:

$$\neg\langle bel\rangle \perp \tag{1}$$

$$\langle bel\rangle F \wedge \langle bel\rangle(F \rightarrow G) \rightarrow \langle bel\rangle G \tag{2}$$

$$\langle bel\rangle F \rightarrow \langle bel\rangle\langle bel\rangle F \tag{3}$$

$$\neg\langle bel\rangle F \rightarrow \langle bel\rangle\neg\langle bel\rangle F \tag{4}$$

Axiom (1) expresses that an inconsistency is not believable by a decision maker. The derivation of the symbol $\perp$ from the database implies inconsistency. Axiom (2) states that a decision maker believes all the logical consequences of its beliefs, that is, a decision maker's beliefs are closed under logical deduction. The two facts that a decision maker believes that s/he believes in something and a decision maker believes that s/he does not believe in something are expressed by axioms (3) and (4) respectively. We also have the rule of necessitation for beliefs:

$$if \vdash F \; then \vdash \langle bel \rangle F \qquad (5)$$

Proposition 1. *The following are theorems of LR^2L:*
$\langle bel \rangle (F \wedge G) \leftrightarrow \langle bel \rangle F \wedge \langle bel \rangle G$
$\langle bel \rangle F \vee \langle bel \rangle G \rightarrow \langle bel \rangle (F \vee G)$
$\langle bel \rangle F \rightarrow \neg \langle bel \rangle \neg F$

There is no support for an inconsistency and the following axiom reflect this property:

$$\neg \langle sup_d \rangle \perp, \; for \; every \; d \in D \qquad (6)$$

Support is closed under tautological implications by preserving degrees. In other words, if F has a support d and $F \rightarrow G$ is an LR^2L tautology then G too has a support d:

$$if \vdash F \rightarrow G \; then \vdash \langle sup_d \rangle F \rightarrow \langle sup_d \rangle G, \; for \; every \; d \in D \qquad (7)$$

If an observation in the real world generates support d for F and if $F \rightarrow G$ is a decision maker's belief then it is unreasonable to conclude that d is also a support for G. This prevents us from considering supports closed under believed implications. The following rule of inference states that an LR^2L tautology has always the highest support:

$$if \vdash F \; then \vdash \langle sup_\Delta \rangle F \qquad (8)$$

Proposition 2. *For every d in D, the following is a theorem of LR^2L:*
$\langle sup_\Delta \rangle \top$
$\langle sup_d \rangle (F \wedge G) \rightarrow \langle sup_d \rangle F \wedge \langle sup_d \rangle G$

A ***rational decision maker*** believes in something which has support with the top element of the dictionary. Thus, the axiom $\langle sup_\Delta \rangle F \rightarrow \langle bel \rangle F$. should be considered for a rational decision maker. This axiom, of course, derives that an assertion and its negation are not simultaneously derivable with the top element as support, that is, an integrity constraint [6] of the form $\langle sup_\Delta \rangle F \wedge \langle sup_\Delta \rangle \neg F \rightarrow \perp$. It is difficult to maintain consistency of a database in the presence of this axiom, particularly when the database is constructed from different sources; mutual inconsistency and mistakes sometimes need to be tolerated. In these circumstances, it might be left to the decision maker to arbitrate over which to believe or not believe.

A decision maker might believe in something even if the database derives no support for it. We call a decision maker who does not believe in something unless there is support with the top element a *strict decision maker.* If a decision maker is both rational and strict then the concepts of believability and support with the top element coincide. In other words, $\langle sup_{\triangle}\rangle F \leftrightarrow \langle bel\rangle F$. Note that we do not consider $\langle sup_{d1}\rangle F \rightarrow \langle sup_{d2}\rangle F$ (where $d2 \leq d1$) as an axiom which says that certain evidence for an assertion also implies every evidence for the assertion lower than the evidence. The reason for exclusion will be given in the context of model definition. The exclusion also avoids the unnecessary contributions to the aggregation process for F.

We adopt the following two standard axioms of goals [5, 19]:

$$\neg\langle goal\rangle \perp \tag{9}$$

$$\langle goal\rangle F \wedge \langle goal\rangle(F \rightarrow G) \rightarrow \langle goal\rangle G \tag{10}$$

Axiom (9) says that something that is impossible to achieve cannot be a goal of a decision maker. Axiom (10) states that all the logical consequences of a decision maker's goal are goals themselves.

Proposition 3. *The following are theorems of LR^2L:*
$\langle goal\rangle F \wedge \langle bel\rangle(F \rightarrow G) \rightarrow \langle goal\rangle G$
$\langle goal\rangle(F \wedge G) \leftrightarrow \langle goal\rangle F \wedge \langle goal\rangle G$
$\langle goal\rangle F \vee \langle goal\rangle G \rightarrow \langle goal\rangle(F \vee G)$
$\langle goal\rangle F \rightarrow \neg\langle goal\rangle\neg F$
$\langle goal\rangle F \rightarrow \neg\langle bel\rangle\neg F$

According to [5], worlds compatible with a decision maker's goals must be included in those compatible with the decision maker's beliefs. This is summarised in the following axiom:

$$\langle bel\rangle F \rightarrow \langle goal\rangle F \tag{11}$$

A database is full of decision maker's belief. Consequently, many redundant goals will be generated due to the presence of the above axiom. A goal will be considered *achieved* (resp. *active*) in a state if it is derivable (resp. not derivable) in the state.

7 Semantics of LR^2L

A *model* of LR^2L is a tuple $\langle W, V, R_b, R_s, R_g\rangle$ in which W is a set of all possible worlds and V is a valuation which associates a world to a set of propositions which are true in that world. In other words, $V : W \rightarrow \Pi(P)$, where P is the set of propositions and $\Pi(P)$ is the power set of P. Some additional restrictions have to be placed on the relations of the model corresponding to some of the axioms presented in section 6.

The relation R_b relates a world w to a set of worlds considered possible by the decision maker from w. If there are n candidates in a decision construct which is active in a world w then the size of such set of possible worlds will be 2^n. An assertion is said to be a *belief* of the decision maker at a world w if and only if it

is provable in every possible world accessible from w by the accessibility relation R_b. The presence of axiom $\neg\langle bel\rangle \perp$ (axiom (1)) in our system guarantees the existence of a world in which an assertion is true if it is believed in the current state. The accessibility relation R_b has the following set of properties due to axioms (1), (3) and (4):
(A) R_b is serial, transitive and euclidean.

The relation R_s is a *hyperelation* which is a subset of $W \times D \times \Pi(W)$. Semantically, if $\langle w, d, W' \rangle \in R_s$ then there is an amount of support d for moving to one of the worlds in W' from the world w, where W' is non-empty. In other words, the support d is for the set of assertions uniquely characterised by the set of worlds W'. Given the current world w, a decision maker either continues to stay in the current world or to move to one of the possible worlds accessible from w by R_b. In either case, the changed world always belongs to W. This states that there is always the highest support for moving to one of the worlds in W from any world. The restrictions on R_s are now summarised as the following set of properties:
(B) for every w in W and d in D, if $\langle w, d, W' \rangle \in R_s$ then $W' \neq \emptyset$, and
for every w in W, $\langle w, \Delta, W \rangle \in R_s$.

Suppose, $\langle w, 0.7, \{w_1, w_2\} \rangle$ is in R_s, where the support 0.7 gets distributed as 0.4 and 0.3 to w_1 and w_2 respectively. In addition, $\langle w, 0.6, \{w_1, w_2\} \rangle$ is also in R_s, where the support 0.6 gets distributed as 0.5 and 0.1 to w_1 and w_2 respectively. Although, 0.6 is less than 0.7, having $\langle w, 0.7, \{w_1, w_2\} \rangle$ in R_s does not imply $\langle w, 0.6, \{w_1, w_2\} \rangle$ is in R_s. This demonstrates that, in general, $\langle w, d1, W' \rangle \in R_s$ does not necessarily mean $\langle w, d2, W' \rangle \in R_s$, for $d2 \leq d1$.

Aggregation of arguments introduces a hierarchy of preferences [7] among the set of all possible worlds accessible from w by the relation R_b. The maximal elements and possibly some elements from the top of the hierarchy of this preference structure will be called *goal worlds*. The relation R_g, which is a subset of R_b, relates the current world to the set of goal worlds. An assertion is a *goal* in a world w if and only if it is provable in every goal world accessible from w by the accessibility relation R_g. Axiom (9) introduces the seriality property on the accessibility relation R_g and axiom (11) restricts R_g to a subset of R_b. Thus we have the following set of properties of R_g:
(C) R_g is serial and $R_g \subseteq R_b$.

To keep our development practical and simple we have excluded a number of axioms related to goals. Two such axioms concerned with goals [19] are (a) if a decision maker has a goal of having a goal then s/he has this goal and the converse (b) if a decision maker has a goal of not having a goal then s/he has not got this goal and vice versa. If we had considered these axioms this would have introduced some extra properties on the accessibility relation R_g. Only one of the goal worlds is committed to move from the current world and this world will be called the *committed world*.

Given a model $\mathcal{M} = \langle W, V, R_b, R_s, R_g \rangle$, truth values of formulae with respect to a world w are determined by the rules given below:
$\models^{w}_{\mathcal{M}} \top$

$\models^w_{\mathcal{M}} p$ iff $p \in V(w)$.
$\models^w_{\mathcal{M}} \langle sup_d \rangle F$ iff there exists $\langle w, d, W' \rangle$ in R_s such that $\models^{w'}_{\mathcal{M}} F$, for every $w' \in W'$
$\models^w_{\mathcal{M}} \langle bel \rangle F$ iff for every w' in W such that wR_bw', $\models^{w'}_{\mathcal{M}} F$
$\models^w_{\mathcal{M}} \langle goal \rangle F$ iff for every w' in W such that wR_gw', $\models^{w'}_{\mathcal{M}} F$
$\models^w_{\mathcal{M}} \neg F$ iff $\not\models^w_{\mathcal{M}} F$
$\models^w_{\mathcal{M}} F \wedge G$ iff $\models^w_{\mathcal{M}} F$ and $\models^w_{\mathcal{M}} G$

A formula F is said to be *true* in model $\mathcal{M}$, written as $\models_{\mathcal{M}} F$, if and only if $\models^w_{\mathcal{M}} F$, for every world w in W. A formula F is said to be *valid*, written as $\models F$, if F is true in every model.

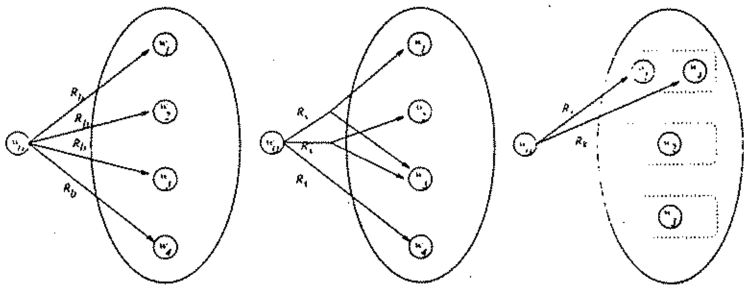


Fig. 3. Relation between the current world and possible worlds.

8 Medical Example

This section provides an example which illustrates the semantics presented in the previous section. First of all, we consider the dictionary D as $dict(Prob)$ and D' is D itself. Suppose the current world w_0 is described by a database consisting of the formulae in section 5 (which are translated from the decision constructs presented in section 2) as hypotheses and the following set as *knowledge* ($\equiv F \wedge \langle bel \rangle F$): {*young*, *smoker*, *pain_in_upper_abdomen*, *weight_loss*}
The valuation V on w_0 is defined as follows:
$V(w_0) = \{$*young*, *smoker*, *pain_in_upper_abdomen*, *weight_loss*$\}$
Since there are 2 candidates in the *weight_loss_diagnosis* decision construct, there will be 2^2, that is, four possible worlds w_1, w_2, w_3 and w_4 whose valuations are as follows (see Figure 3):
$V(w_1) = V(w_0) \cup \{$*cancer*, *weight_loss_diagnosed*$\}$
$V(w_2) = V(w_0) \cup \{$*peptic_ulcer*, *weight_loss_diagnosed*$\}$
$V(w_3) = V(w_0) \cup \{$*cancer*, *peptic_ulcer*, *weight_loss_diagnosed*$\}$
$V(w_4) = V(w_0) \cup \{$*weight_loss_diagnosed*$\}$
The relations R_b and R_s in the model definition are defined as follows:
$R_b = \{\langle w_0, w_1 \rangle, \langle w_0, w_2 \rangle, \langle w_0, w_3 \rangle, \langle w_0, w_4 \rangle\}$
$R_s = \{\langle w_0, d2, \{w_1, w_3\} \rangle, \langle w_0, d4, \{w_2, w_3\} \rangle, \langle w_0, d2, \{w_4\} \rangle\}$

Note that *weight_loss_diagnosed* is true in each of the possible worlds and therefore this is a goal as the set of goal worlds is a subset of the the set of possible worlds. The goal corresponds to the provability of $\langle goal\rangle \mathit{weight\_loss\_diagnosed}$ in the current world using $\langle bel\rangle \mathit{weight\_loss}$ in conjunction with the formula
$\langle bel\rangle \mathit{weight\_loss} \rightarrow \langle goal\rangle \mathit{weight\_loss\_diagnosed}$
The goal is active in w_0. We are, of course, assuming that the theorem prover of LR^2L is able to derive the negation of $\langle bel\rangle \mathit{weight\_loss\_diagnosed}$ from the current world by a mechanism similar to negation by failure. The total supports for the mutually exclusive possibilities are computed by the aggregation process (using the domain knowledge that *cancer* and *peptic_ulcer* are almost mutually exclusive candidates) as follows:
- support for $C_1(\mathit{cancer} \wedge \neg \mathit{peptic\_ulcer}) = \mathcal{A}(\{\langle C_1, G_1, d2\rangle\}) = d2$
- support for $C_2(\neg \mathit{cancer} \wedge \mathit{peptic\_ulcer}) = \mathcal{A}(\{\langle C_2, G_2, d4\rangle\}) = d4$
- support for $C_3(\mathit{cancer} \wedge \mathit{peptic\_ulcer}) = \mathcal{A}(\{\langle C_3, G_1, d2\rangle, \langle C_3, G_2, d4\rangle\}) = 0$
- support for $C_4(\neg \mathit{cancer} \wedge \neg \mathit{peptic\_ulcer}) = \mathcal{A}(\{\langle C_4, G_4, d4\rangle\}) = d2$
where each di is drawn from $dict(Prob)$ and the grounds G_1, G_2 and G_4 are:
$G_1 = \mathit{weight\_loss} \wedge \mathit{smoker} \rightarrow \mathit{cancer}$
$G_2 = \mathit{weight\_loss} \wedge \mathit{pain\_in\_upper\_abdomen} \rightarrow \mathit{peptic\_ulcer}$
$G_4 = \mathit{weight\_loss} \wedge \mathit{young} \rightarrow \neg \mathit{cancer} \wedge \neg \mathit{peptic\_ulcer}$
Assuming that $d4$ is less than $d2$, the preference relation $\prec$ among the set of possible worlds is derived as $w_3 \prec w_2$, $w_2 \prec w_1$ and $w_2 \prec w_4$. The maximally preferred possible worlds are w_1 and w_2. The relation R_g in the model definition is now defined as follows:
$R_g = \{\langle w_0, w_1\rangle, \langle w_0, w_4\rangle\}$
This yields a dilemma. In case the decision maker cannot gather any more evidence, s/he may commit to w_4 by preferring w_4 to w_1. This involves adding $\neg \mathit{cancer}$ and $\neg \mathit{peptic\_ulcer}$ to the current state of the database as either beliefs or knowledge depending on the strength of support and decision maker's confidence. The goal *weight_loss_diagnosis* in the new situation will no longer be active due to the presence of
$\langle bel\rangle(\neg \mathit{cancer} \wedge \neg \mathit{peptic\_ulcer}) \rightarrow \mathit{weight\_loss\_diagnosed}$
Note that we add only beliefs to keep the belief revision option open in case of wrong diagnosis. Alternatively, if we now add $\langle bel\rangle \mathit{positive\_biopsy}$ as an additional evidence into the database that would increase the total support for C_1 as follows:
- total support for $C_1 = \mathcal{A}(\{\langle C_1, G_1, d2\rangle, \langle C_1, G'_1, d3\rangle\}) = d2 + d3 - d2 * d3$
where the additional ground G'_1 for C_1 is the following:
$G'_1 = \mathit{weight\_loss} \wedge \mathit{positive\_biopsy} \rightarrow \mathit{cancer}$
The revised valuation on each w_i will be as before except *positive_biopsy* changes its truth value. The relations R_s and R_g will be redefined as follows:
$R_s = \{\langle w_0, d2, \{w_1, w_3\}\rangle, \langle w_0, d4, \{w_2, w_3\}\rangle, \langle w_0, d2, \{w_4\}\rangle, \langle w_0, d3, \{w_1, w_3\}\rangle\}$
$R_g = \{\langle w_0, w_1\rangle\}$
Since w_1 is the only goal world, the decision maker considers w_1 as the committed world. Changing to the committed world from the current world involves adding *cancer* and $\neg$*peptic_ulcer* to the database as decision maker's beliefs. Adding

$\langle bel\rangle cancer$ to the database will trigger the decision for cancer treatment and the decision making process continues as before.

If $d2 = d3$ then the two members $\langle w_0, d2, \{w_1, w_3\}\rangle$ and $\langle w_0, d3, \{w_1, w_3\}\rangle$ of R_s are indistinguishable although they correspond to two different arguments. The relation in a more accurate model takes the following form:
$R_s = \{\langle w_0, d2, W_1\rangle, \langle w_0, d4, W_2\rangle, \langle w_0, d2, W_4\rangle, \langle w_0, d3, W_5\rangle\}$
where W_1 (resp. W_2, W_4 and W_5) is the set of possible worlds uniquely characterised by the set $\{smoker, cancer\}$ (resp. $\{pain\_in\_upper\_abdomen, peptic\_ulcer\}$, $\{young\}$ and $\{positive\_biopsy, cancer\}$) which includes $\{w_1, w_3\}$ (resp. $\{w_2, w_3\}$, $\{w_4\}$ and $\{w_1, w_3\}$). In fact, the distribution of support $d2$ (resp. $d4$, $d2$ and $d3$) among the members of W_1 (resp. W_2, W_4 and W_5) is confined to the subset $\{w_1, w_3\}$ (resp. $\{w_2, w_3\}$, $\{w_4\}$ and $\{w_1, w_3\}$) as any other member is not considered possible by the decision maker.

9 Soundness and Completeness

The technique adopted in this section for establishing soundness and completeness of LR^2L is similar to the one for the class of normal logics in [3]. First of all, the following two propositions prove that the validity in a class of models is preserved by the use of the rule of inference and the axioms of LR^2L.

Proposition 4. *For every formulae F and G the following hold:*
- *if $\models F$ then $\models \langle bel\rangle F$*
- *if $\models F \rightarrow G$ then $\models \langle sup_d\rangle F \rightarrow \langle sup_d\rangle G$*
- *if $\models F$ then $\models \langle sup_\Delta\rangle F$*

Proposition 5. *For every formulae F and G the axioms (1)-(11) are valid in LR^2L.*

Propositions 4 and 5 establishes the basis of the soundness result. In order to prove the completeness result, the following class of models is relevant.

Definition 6. A model $\mathcal{M} = \langle W, V, R_b, R_s, R_g\rangle$ of LR^2L is called a *canonical model*, written as $\mathcal{M}_c$, if and only if
- $W = \{w$: w is a maximal consistent set in logic $LR^2L\}$.
- For every w, $\langle bel\rangle F \in w$ iff for every w' in W such that wR_bw', $F \in w'$.
- For every w, d and W', $\langle sup_d\rangle F \in w$ iff there exists $\langle w, d, W'\rangle \in R_s$ such that $F \in w'$, for every w' in W'.
- For every w, $\langle goal\rangle F \in w$ iff for every w' in W such that wR_gw', $F \in w'$.
- For each proposition p, $\models^w_{\mathcal{M}_c} p$ iff $p \in w$.

Proposition 7. *Let $\mathcal{M}_c = \langle W, V, R_b, R_s, R_g\rangle$ be a canonical model of LR^2L. Then, for every w in W, $\models^w_{\mathcal{M}_c} F$ if and only if $F \in w$.*

Therefore, the worlds in a canonical model for LR^2L will always verify just those sentences they contain. In other words, the sentences which are true in such a model are precisely the theorems of LR^2L.

Theorem 8. *Let $\mathcal{M}_c = \langle W, V, R_b, R_s, R_g\rangle$ be a canonical model of LR^2L. Then $\vdash F$ if and only if $\models_{\mathcal{M}_c} F$, for every formula F.*

Existence of a canonical model for LR^2L is shown by the existence of a proper canonical model defined as follows.

Definition 9. A model $\mathcal{M} = \langle W, V, R_b, R_s, R_g\rangle$ of LR^2L is called a *proper canonical model*, written as $\mathcal{M}_{pc}$, if and only if
- $W = \{w$: w is a maximal consistent set in logic $LR^2L\}$.
- For every w and w', wR_bw' iff $\{F : \langle bel\rangle F \in w\} \subseteq w'$.
- For every w, d and W', $\langle w, d, W'\rangle \in R_s$ iff $\{\langle sup_d\rangle F : F \in \cap W'\} \subseteq w$.
- For every w and w', wR_gw' iff $\{F : \langle goal\rangle F \in w\} \subseteq w'$.
- For each proposition p, $\models^{w}_{\mathcal{M}_{pc}} p$ iff $p \in w$.

By definition, a proper canonical model exists and the following proposition establishes that a proper canonical model is a canonical model.

Proposition 10. *Suppose $\mathcal{M}_{pc}$ is a proper canonical model of LR^2L as defined above. Then $\mathcal{M}_{pc}$ is also a canonical model.*

Theorem 11. *If $\mathcal{M}_{pc}$ is a proper canonical model of LR^2L then the model satisfies properties (A), (B) and (C).*

Suppose Γ is the class of all models satisfying the properties (A), (B) and (C). Then the following soundness and completeness theorem establishes the fact that LR^2L is determined by Γ.

Theorem 12. *For every formula F in LR^2L, $\vdash F$ if and only if $\models F$.*

10 Discussion

We have presented a modal formalisation of a very general framework for the design of knowledge-based decision support systems. As well as the conventional evaluation of decision options, we are able to handle within our framework the proposal of decision candidates and reasoning about actions once a decision has been made. Consequently, the model presented in this paper provides a more comprehensive account of decision making than classical decision theory. Furthermore, we are not constrained by requirements for comprehensive probability tables to enable a decision to be made; qualitative reasoning can be used if complete probability tables are unavailable. A discussion detailing the advantage of our approach over the traditional statistical decision theory (e.g. expected utility theory) and knowledge-based (expert) systems can be found in [9].

Although we have focussed on the decision making component of our model in this paper, the theory can be expanded into a general formalism incorporating actions and temporal reasoning. This aspect has been described elsewhere [9, 10].

Our formal model of decision making has already been successfully used in decision support applications in the domains of cancer and asthma protocol management [12]. In these applications, a human decision maker has over-riding control of the management process. However, we envisage that this extended model will provide a promising framework for the design of autonomous agents and multi-agent systems.

References

1. R. H. Bonczek, C. W. Holsapple, and A. B. Whinston. Development in decision support systems. *Advances in Computers*, 23:123–154, 1984.
2. R. Bradley and N. Swartz. *Possible Worlds*. Basil Blackwell, 1979.
3. B. Chellas. *Modal Logic*. Cambridge University Press, 1980.
4. D. A. Clark, J. Fox, A. J. Glowinski, and M. J. O'Neil. Symbolic reasoning for decision making. In K. Borcherding, O. I. Larichev, and D. M. Messick, editors, *Contemporary Issues in Decision Making*, pages 57–75. Elsevier Science Publishers B. V. (North-Holland), 1990.
5. P. R. Cohen and H. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42, 1990.
6. S. K. Das. *Deductive Databases and Logic Programming*. Addison-Wesley, 1992.
7. S. K. Das. A logical reasoning with preference. *Decision Support Systems*, 15:19–25, 1995.
8. S. K. Das and D. Elsdon. R^2L. Technical Report RED/ QMW/ WP/ 740/ 1/ 4, QMW, University of London, London, 1994.
9. S. K. Das, J. Fox, P. Hammond, and D. Elsdon. A flexible architecture for autonomous agents. *revised version is being considered by JETAI*, 1995.
10. S. K. Das and P. Hammond. Managing tasks using an interval-based temporal logic. *Journal of Applied Intelligence*, in press.
11. R. Fagin and J. Y. Halpern. Belief, awareness and limited reasoning. *Artificial Intelligence*, 34:39–76, 1988.
12. J. Fox and S. K. Das. A unified framework for hypothetical and practical reasoning (2): lessons from medical applications. In *Proceeding of FAPR*, June 1996.
13. J. Fox, S. K. Das, and D. Elsdon. Decision making and planning in autonomous systems: theory, technology and applications. In *Proceedings of the ECAI Workshop on Decision Theory for DAI Applications*, 1994.
14. J. Y. Halpern and Y. O. Moses. A guide to the modal logics of knowledge and belief. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, pages 480–490, 1985.
15. P. J. Krause, S. J. Ambler, M. Elvang-Goransson, and J. Fox. A logic of argumentation for uncertain reasoning. *Computational Intelligence*, 1995.
16. S. A. Kripke. Semantical analysis of modal logic I: normal modal propositional calculi. *ZMLGM*, 9:67–96, 1963.
17. E. J. Lemmon. *An Introduction to Modal Logic*. Basil Blackwell, 1977.
18. J.-J. Ch. Meyer, W. van der Hoek, and G. A. W. Vreeswijk. Epistemic logic for computer science: a tutorial (part one). *EATCS*, 44:242–270, 1991.
19. J. Wainer. Yet another semantics of goals and goal priorities. In *Proceedings of the 11th European Conference on Artificial Intelligence*, pages 269–273, August 1994.

This article was processed using the LaTeX macro package with LLNCS style

A unified framework for hypothetical and practical reasoning (2): lessons from medical applications[1]

John Fox[*], Subrata Das[**]

[*] Imperial Cancer Research Fund, London
[**] William Penney Laboratory, Imperial College, London

Abstract . A general theory of decision making which unifies hypothetical reasoning (reasoning about beliefs) and practical reasoning (reasoning about actions and plans) has been reported elsewhere (e.g. Fox et al, 1988; 1990). The theory is grounded in research on non-standard logics, and particularly our proposals for LA, a logic of argumentation (Fox et al, 1992; Krause et al, 1995). A model-theoretic semantics for the theory is presented in a companion paper (Das et al, 1995). Here we focus on applications in medicine, to illustrate the utility of the approach in medical decision making (e.g. in diagnosis; selection of investigations and therapies; drug prescribing) and management of clinical procedures (e.g. execution of care plans, scheduling of clinical actions, and acquisition of clinical data). The theory provides foundations for an application development methodology based on a formal language for representing medical knowledge. The paper discusses some lessons learned from these applications and identifies some important problems which theoreticians might valuably address.

1. Introduction

Medical informaticians have been developing computer systems to help doctors make clinical decisions for well over thirty years. However, there has been relatively little impact on routine clinical practice. Among the reasons for this have been a lack of convincing demonstrations that computers can help to improve care; unacceptable demands on busy clinical staff; inflexibility of user interfaces, and perhaps a certain unwillingness by doctors to accept that they might benefit from the use of computers.

This situation is now changing rapidly. Evidence is becoming available that decision support systems can significantly improve medical care (e.g. Johnstone et al, 1994), and doctors are recognising that rapid growth of medical knowledge, combined with increasing economic pressures and the implications of the ageing population, mean that

[1] Many people have been involved in work described in this paper, though the views presented are only those of the authors. We particularly benefited from contributions by: Paul Krause, Paul Taylor, Jean-Louis Renaud-Salis, Paul Ferguson, Saki Hajnal and Peter Hammond. We also wish to thank Jun Huang, David Elsdon, Claude Gierl, Richard Thomson, Andrew Jackson-Smale, Colin Gordon, Philippe Lagouarde, Ian Herbert, Peter Johnson, Robert Walton, Andrzej Glowinski, Mike O'Neil, Ahmed Benlamkaddem, Pierre Roblez and Zeljko Ilic, all of whom have contributed to the applications described in the paper.

it will be increasingly difficult to continue to deliver high quality care unaided. In the coming years the main obstacle to the successful introduction of computerised decision support systems will not be professional doubts, but limitations of the technologies and concerns about safety.

The term "decision support system" (DSS) is used to describe many different kinds of clinical software. Some authors include general information sources, such as drug databases, electronic texts and multimedia documents in the category, since they provide general information that can be relevant to a clinical decision. Others restrict use of the term to systems which provide assistance that is specific to the care of a particular patient (Johnstone et al, op cit). Examples of this kind of assistance include *reminders* (e.g. about topics which are relevant to specific diagnosis or treatment decisions), *alerts* (e.g. drug interaction warnings) and *recommendations* (e.g about the most likely diagnosis or most preferred therapy). Recently the term has also come to include systems that help in planning and scheduling clinical procedures (e.g. complex therapy plans such as cancer chemotherapy).

This paper describes a framework for supporting all the above aspects of decision making in a unified way, and which is intended to be flexible and intuitive for doctors to use without sacrificing theoretical soundness or practical safety.

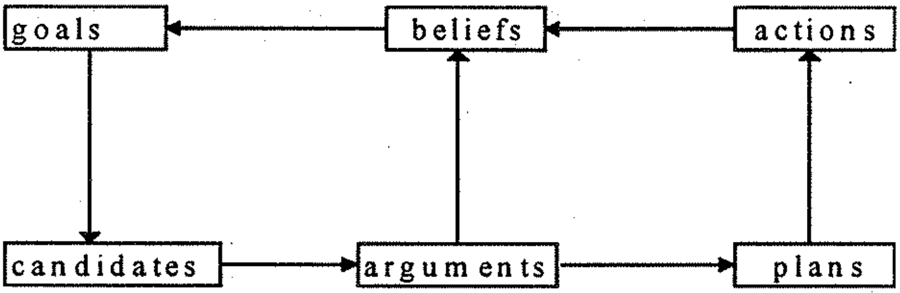


Fig 1: The "DOMINO" model of decision making and plan management

The framework is summarised in figure 1, the "DOMINO model". This consists of a set of reasoning elements which work together in an integrated way. The model is grounded in research on non-standard logics and logic programming.

The DOMINO model has emerged from a decade of work on applying computational logic to the problem of reasoning under uncertainty. This has been motivated by a desire to build practical aids for use in a variety of clinical settings. Our work has addressed general practice (e.g. Fox et al, 1987; Walton and Gierl, 1995), cancer care (e.g. Renaud-Salis and Taylor, 1990; Shortliffe, 1990), radiology (e.g. Taylor, 1995), clinical trials (e.g. Hammond et al, 1994) and multi-disciplinary shared care (Huang et al, 1994).

Many of the systems presented below share functionality with other knowledge-based systems reported in the literature, including diagnostic systems, and systems for therapy management (e.g. Shortliffe, 1990). However, the approach outlined here is believed to differ significantly from other work in a number of respects.

1. Knowledge based DSSs are flexible but have not been based on a well understood theory of decision making. In contrast, systems based on statistical decision theory draw on a "normative", well understood framework, but are inflexible and difficult to use. The DOMINO model is believed to offer a significant advance on these older traditions by providing practical flexibility while preserving theoretical soundness.

2. The model integrates a number of practical reasoning techniques. It can be used to support a range of tasks in many areas of medicine (and perhaps in non-medical fields as well). The main objectives of the present paper are to demonstrate this generality, and to argue that the framework acquires its versatility from a property of "compositionality": different kinds of application can be constructed from different subsets of reasoning elements.

3. The model has strong links with research on autonomous agents (Fox and Krause, 1991) and, in particular, with the class of intelligent agents known as Beliefs-Desires-Intentions or BDI agents (Wooldridge and Jennings, 1995).

4. The DOMINO model is embodied in a well-defined knowledge representation language, R^2L. This is intended to support application designers in clearly specifying the knowledge required for decision making and planning in the application domain.

5. A formal semantics for most of R^2L has been developed; this is presented in our companion paper (Das et al, 1995).

6. Whether used only to support decision making or incorporated in autonomous agents, decision models raise important issues concerning operational safety. We advocate using a number of techniques for maximising safety (Fox, 1994), ranging from formal specification and verification of knowledge bases to more novel methods by which systems reason explicitly from general "safety axioms" (Hammond et al 1994). We believe the work is distinctive in addressing the challenges of safety-critical applications.

Overview

The next section provides a summary of the history and an intuitive presentation of the DOMINO model. Section 3 presents a number of applications. Section 4 summarises the R^2L language and outlines a methodology for developing applications. Finally, section 5 discusses general lessons learned from the medical domain and theoretical challenges arising from it.

2. The DOMINO model: history and informal explanation

The origins of the model are in the *Oxford System of Medicine* (OSM), a decision support system for general practitioners (Fox et al, 1987). When the OSM was conceived, medical DSS research was generally concerned with decision making in specialist medicine, such as statistical methods for diagnosis of abdominal pain or expert systems for diagnosis and treatment of bacterial infections. General practice has requirements which do not arise in specialist applications; in our view these demanded a new approach.

Among these requirements were the following.

(1) Patient care involves many different types of decision for a patient, not just diagnosis (e.g. decisions about investigation, therapy selection, risk assessment, specialist referral etc.).

(2) Doctors will wish to choose the type and level of support required rather than have a technologist's view imposed upon them. An experienced doctor will often only want quick access to up to date information, or reminders about topics relevant to a specific decision, and perhaps only occasionally will require a "second opinion" about diagnosis or treatment. The DSS should permit a doctor to use it as appropriate.

(3) General practice covers the whole of medicine; it should incorporate sufficient medical knowledge to address a wide range of common problems seen by the practitioner.

(4) The system should be able to provide practical help using only qualitative information (because reliable statistical data are only available for a small fraction of medicine).

The OSM design addressed these requirements using the layered architecture shown in figure 2. Here the outer layers correspond to general knowledge which is used to interpret more specific, and particularly patient-specific knowledge. Knowledge was encoded declaratively as facts which instantiated a decision procedure encoded as first-order rules (Fox et al, 1990; Huang et al, 1993). These rules embodied *ad hoc* inference techniques, though these were later developed formally.

The *Oxford System of Medicine* provided support for the general practitioner in making several types of decision including diagnosis, investigation selection, and treatment decisions. The OSM demonstrated a range of queries on dynamic patient data and static medical knowledge, and offered various kinds of help to the user in focussing on specific clinical goals and structuring the decision to be taken (Fox et al, 1987; 1990). For example, the user could identify weight loss as a clinical focus in the patient record, and select "diagnosis", "investigation" or "treatment" as a goal. In response the OSM applied an appropriate strategy for decision making in order to:

(1) generate possible decision candidates (e.g. diagnoses, treatments);

(2) construct arguments for and against the different decision options, and

(3) aggregate the arguments to yield a preference ordering on the various diagnostic or treatment options.

The OSM was implemented as a deductive database (in Prolog). This permitted the declarative encoding of first-order rules and meta-level knowledge, and simplified the provision of a range of user queries on the static knowledge base (for information retrieval) and dynamic patient database (for patient-specific reminders, explanations etc.).

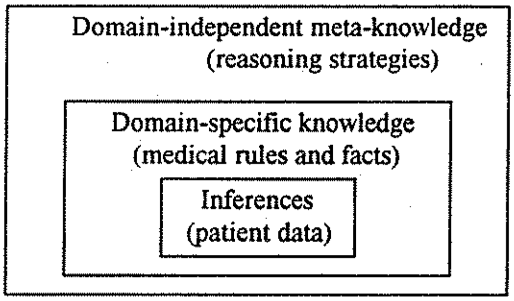

Domain-independent meta-knowledge
(reasoning strategies)
Domain-specific knowledge
(medical rules and facts)
Inferences
(patient data)

Fig 2: layered architecture of the Oxford System of Medicine

A second source of requirements which influenced the development of the DOMINO model was a system designed to support decision making in oncology (cancer care). The *Bordeaux Oncology Support System* (BOSS) was developed to meet many of the same requirements as the OSM, but in addition it was designed to support the execution of complex therapy plans, as pioneered in the ONCOCIN system at Stanford University. BOSS applied decision making functions to the diagnosis of breast cancer, assessment of the stage of the disease, and selection of a therapy plan (often referred to as a "protocol" in medicine), represented as a logic database. The protocol was used to guide the execution of medical tasks: the acquisition of clinical data, and sequencing of decisions and clinical acts etc (Renaud-Salis and Taylor, 1990). BOSS exploited a fairly complete, though still *ad hoc*, version of the DOMINO.

The DOMINO model as a decision procedure

The DOMINO can also be viewed as an architecture for an "agent" which maintains a set of beliefs about a world, and responds to situations which arise in that world by formulating problems, taking decisions about how to solve them, and carrying out plans and actions to implement its decisions.

In the context of clinical medicine the agent is a software system which maintains a set of *beliefs* about a patient (consisting of a set of facts about a patient, recorded over time and stored in a patient record system). Suppose data are added to the patient record which indicate that the patient has lost weight. The agent may infer that this is abnormal and, consequently, adopt a *goal* to diagnose the cause of the abnormaliy. From its general medical knowledge it deduces that possible causes of weight loss include, say, cancer and peptic ulcer and records these as *candidate solutions* for the diagnosis problem. Using meta-knowledge about the kinds of knowlege which are relevant in making diagnosis decisions, and additional facts in the patient record (e.g. the patient is elderly), the agent constructs *arguments* for and against each candidate diagnosis. This is done using meta-knowledge about the facts and/or rules which are relevant in generating candidates and constructing arguments for each type of problem. To illustrate, the OSM knowledge base included first-order meta-rules such as "any condition which could *cause* the presenting problem is a possible diagnosis" or "when looking for arguments in favour of a disease consider whether any symptoms or signs in the patient record are statistically associated with each hypothetical disease".

At some point enough information may be available about the patient to *commit* to one diagnostic candidate or another, say cancer. This decision is added to the patient record as a new belief.

The commitment to a diagnosis causes another goal to be raised: to *treat* the cancer. As before, a set of candidate solutions is inferred from the medical knowledge base, but this time the candidates are *plans* and/or *actions* rather than hypotheses: chemotherapy, radiotherapy or surgery, for example. Once again, the pros and cons of these alternatives are *argued* for the specific patient and, in due course, the agent takes a decision to *commit* to a particular therapy. This decision puts in train a plan, a sequence of actions, recursive subplans and decisions, for treating the patient. In principle, therapy plans may be constructed dynamically, but in the applications described here we only choose among standardised protocols.

Execution of a clinical plan or protocol will typically involve a number of steps or *tasks*. These will need to be scheduled for execution in a particular order and/or at particular times. For example, we may need to establish various baseline measurements on the patient, administer several cycles of treatment, monitor the effects of treatment and follow up the patient's condition at intervals following therapy. Scheduling must be done using general knowledge of temporal and logical constraints, and situation-specific factors like the patient's well-being and availability of resources. Some tasks will be obligatory, some optional and some based on decisions. Tasks can be specified as collections of ground facts in the agent's knowledge base.

3. Demonstrations of the practicality of the model

Figure 3 summarises the operations which the OSM carried out in terms of the DOMINO model. The goal of decision making was established manually, by the user, in this system but reasoning about possible candidate decision options and the arguments pro and con each alternative was carried out automatically. Arriving at a decision (about the preferred diagnosis or treatment) was carried out by *aggregating* the arguments. This yielded a preference ordering over the candidates; the system could in principle make the decision autonomously, though in practice we have adopted the policy that a trained doctor must actually take (commit to) any decision.

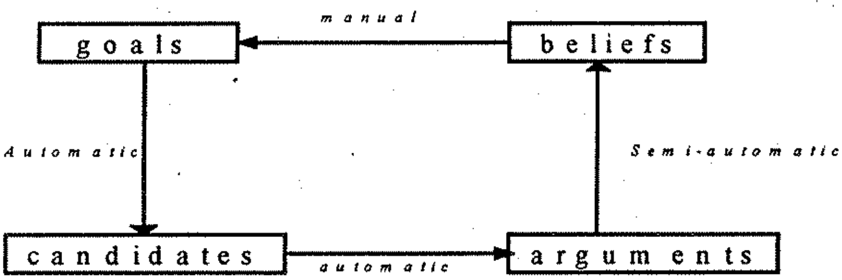
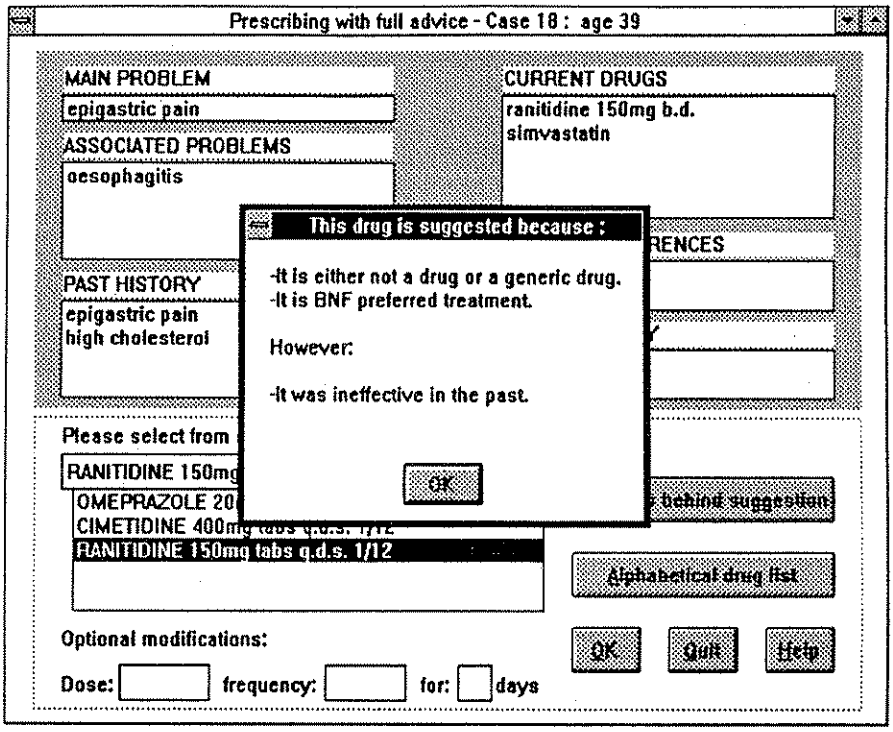


Fig 3: schematic representation of reasoning in the Oxford System of Medicine and CAPSULE prescribing system

An almost identical pattern of decision making was implemented in the CAPSULE drug prescribing system (Walton and Gierl, 1995). CAPSULE was designed to explore the use of logical argumentation in routine prescribing. Figure 4 shows a typical screen from the CAPSULE system. The top half of the display (behind the inset box) contains a view of a simple patient record. It shows the patient's main problem (that which requires medication) and additional information about associated problems, relevant past history, other current drugs etc. These data are all potentially relevant to formulating arguments about the best treatment for the main problem.


Prescribing with full advice - Case 18 : age 39

MAIN PROBLEM
epigastric pain

ASSOCIATED PROBLEMS
oesophagitis

PAST HISTORY
epigastric pain
high cholesterol

CURRENT DRUGS
ranitidine 150mg b.d.
simvastatin

RENCES

This drug is suggested because :

-It is either not a drug or a generic drug.
-It is BNF preferred treatment.

However:

-It was ineffective in the past.

Please select from

RANITIDINE 150mg
OMEPRAZOLE 20
CIMETIDINE 400mg tabs q.d.s. 1/12
RANITIDINE 150mg tabs q.d.s. 1/12

Optional modifications:

Dose: frequency: for: days

Fig 4: CAPSULE, a system for help in prescribing for common conditions. The system examines data in a patient record (top, back) and proposes a set of possible medications in order of preference (on the left near the bottom of the figure). On request CAPSULE provides a summary of the arguments for and against a selected candidate medication (central window). CAPSULE was, developed by Robert Walton and Claude Gierl, who have shown it can significantly improve the speed and quality of GP prescribing.

CAPSULE and the OSM only support individual decisions. This may be sufficient for many situations but often it would be desirable to provide support for more complex patient management processes. For example decisions may be chained together (e.g. a diagnosis decision followed by a treatment decision). In addition, routine patient management typically involves tasks which do not entail significant uncertainty: collecting and recording information; ordering tests; making appointments; scheduling hospital admissions and, increasingly in modern medical practice, paying attention to issues of resource utilisation and the maintenance of administrative records.

Plans and protocols

An important and growing topic in medicine is the use of standardised care plans ("guidelines" and "protocols" in medical terminology). BOSS used a chemotherapy protocol for the treatment of breast cancer, for example. Among the purposes of standardised guidelines are:

(1) ensuring use of state-of-the-art disease management strategies (e.g. in the management of cancer or chronic diseases, where medical research is leading to frequent changes in preferred practice),

(2) minimising errors in diagnosis or therapy selection (which may occur because doctors are tired or overworked, for example), and

(3) ensuring consistency of care (so that the practice of all doctors approaches that of the most skilled).

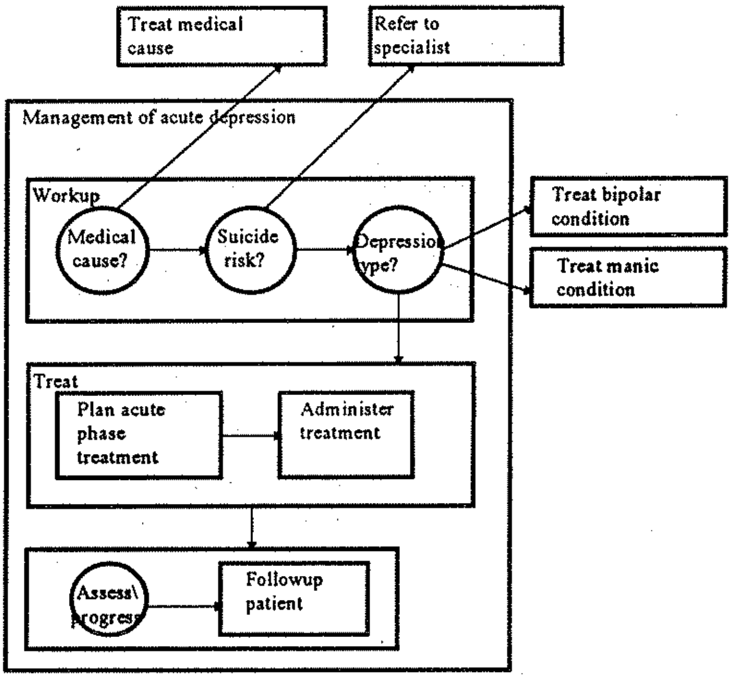
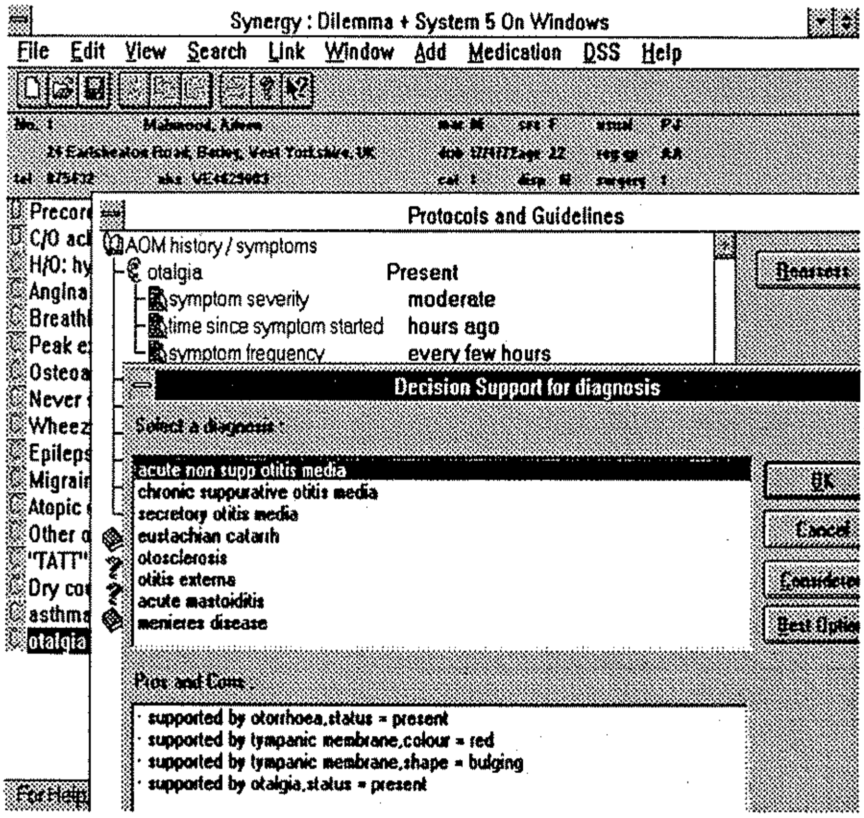


Fig 5: Guideline for the management of acute depression, circles are decisions, boxes are plans or actions, and arrows are temporal or logical scheduling constraints

A typical guideline consists of a number of tasks. These include information collection, decision making, therapy planning and administration tasks. In primary care, for example, the management of chronic conditions like asthma, diabetes and hypertension will typically require a detailed "history" and characterisation of the problem, careful selection of therapy and continued monitoring and reassessment of the patient. In specialist care, such as cancer chemotherapy, guidelines are often complex and include many obligatory rather than optional procedures (which is why they are often referred to as protocols). If protocols are presented in the form of conventional documents they can be hard to follow and use in busy clinical situations. It has been recognised for some time that computers might be used to help doctors comply with guidelines and protocols more successfully.



AOM history / symptoms

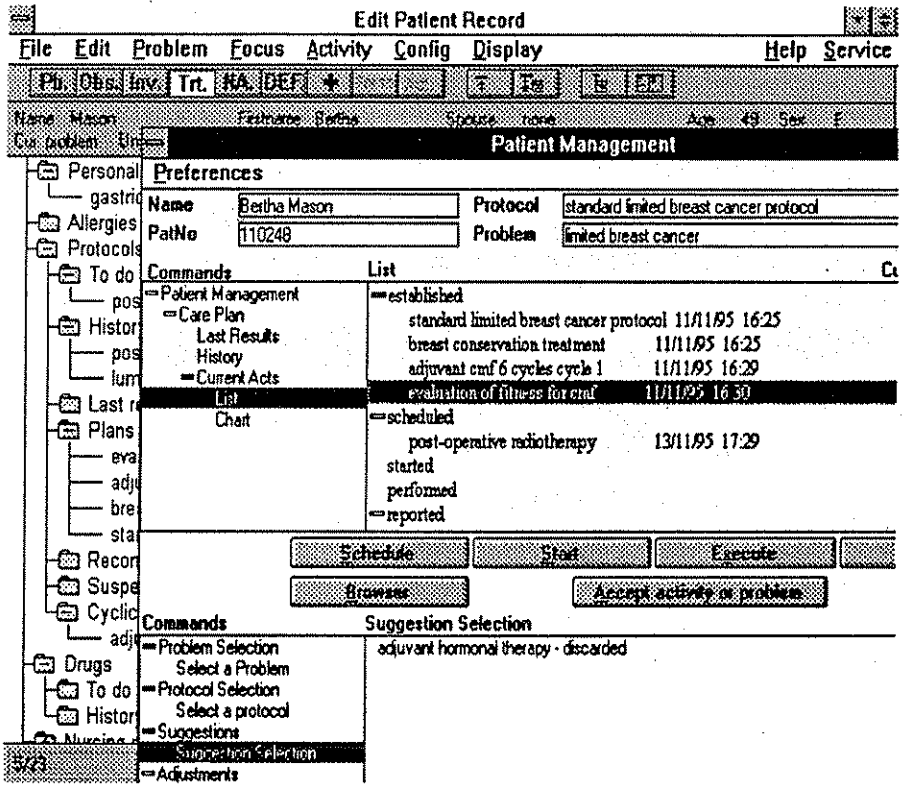
otalgia	Present
symptom severity	moderate
time since symptom started	hours ago
symptom frequency	every few hours

acute non supp otitis media
chronic suppurative otitis media
secretory otitis media
eustachian catarrh
otosclerosis
otitis externa
acute mastoiditis
menieres disease

- supported by otorrhoea.status = present
- supported by tympanic membrane.colour = red
- supported by tympanic membrane.shape = bulging
- supported by otalgia.status = present

Fig 6: The Synergy system demonstrates the integration of "intelligent" guidelines for common conditions with conventional general practice management software. The figure shows a summary of the patient record (back window); an interactive guideline reminds the GP of topics which are relevant to the patient's problems (middle) and an example of decision support, in this case showing a suggested diagnosis with the most likely diagnosis highlighted together with the reasons for the recommendation (front). Synergy was developed by Peter Johnson and Claude Gierl.

Consider the management of acute depression in general practice. Although this is a common problem it is often poorly managed, because preferred practice demands extensive data collection and scheduling of procedures, as well as several significant decisions. Figure 5 gives a schematic representation of part of a guideline for the management of acute depressesion published by the American College of Psychiatrists; although it involves many steps it is not a particularly complex procedure in comparison with many used routinely in some fields of medicine (such as oncology, see Hammond, 1994 for examples).



Name: Mason
Firstname: Bertha
Spouse: none
Age: 49
Sex: F

Name: Bertha Mason
PatNo: 110248
Protocol: standard limited breast cancer protocol
Problem: limited breast cancer

Commands
- Patient Management
  - Care Plan
    - Last Results
    - History
    - Current Acts
      - List
      - Chart

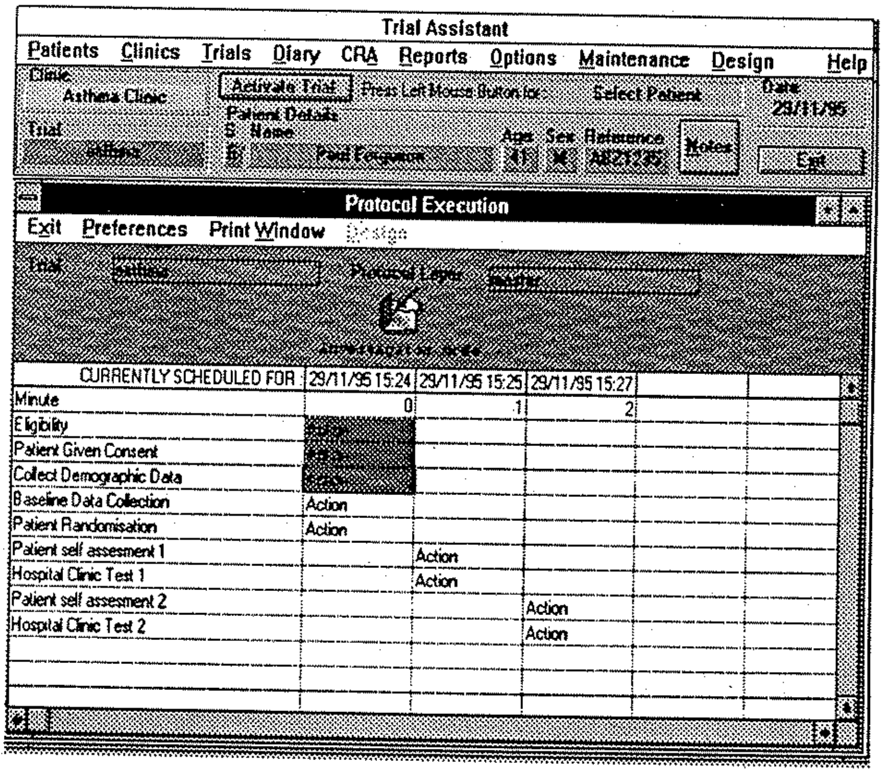
List
- established
  - standard limited breast cancer protocol 11/11/95 16:25
  - breast conservation treatment 11/11/95 16:25
  - adjuvant cmf 6 cycles cycle 1 11/11/95 16:29
  - evaluation of fitness for cmf 11/11/95 16:30
- scheduled
  - post-operative radiotherapy 13/11/95 17:29
- started
- performed
- reported

Commands
- Problem Selection
  - Select a Problem
- Protocol Selection
  - Select a protocol
- Suggestions
  - Suggestion Selection
- Adjustments

Suggestion Selection
adjuvant hormonal therapy - discarded

Fig 7: A screen from a cancer management system ((BOSS2, developed under the EU's Advanced Informatics in Medicine Programme). The application is specifically concerned with managing a combined surgery, radiotherapy and chemotherapy protocol, and provides help with patient data recording, decision making and the scheduling of clinical procedures. The application is accessed via an electronic patient record system at the "back" of the figure (developed by collaborators at Fondation Bergonie) from which members of medical staff call up the patient management assistant, in the front of the figure. The main panel in the top half of the patient management window summarises the current care plan, with facilities for reviewing and updating it. The panel at the bottom provides decision support (e.g. for diagnosis and staging, protocol selection and suggestions for action). BOSS2 was developed in collaboration with the Institut Bergonie, Bordeaux, France.

We have developed several applications which embody the capability to execute guidelines like those in Figure 5. The SYNERGY system is designed to support use of guidelines by general practitioners (Figure 6). BOSS2 provides facilities for supporting protocol-based cancer care (Figure 7). As with BOSS1 outlined earlier it is concerned with chemotherapy for breast cancer, but it is considerably more developed than the earlier version. (SYNERGY and BOSS2 were developed in the EU's Health Telematics project A2005.) CREW is designed to support the management of clinical trials (figure 8), and CADMIUM is a radiology workstation which combines decision support and protocol management functions with both manual and automated image interpretation (figure 9).


Trial Assistant
Patients Clinics Trials Diary CRA Reports Options Maintenance Design Help
Activate Trial
Press Left Mouse Button for
Select Patient
Notes
Exit
Protocol Execution
Exit Preferences Print Window Design

Clinic: Asthma Clinic

Trial: Asthma

Patient Details

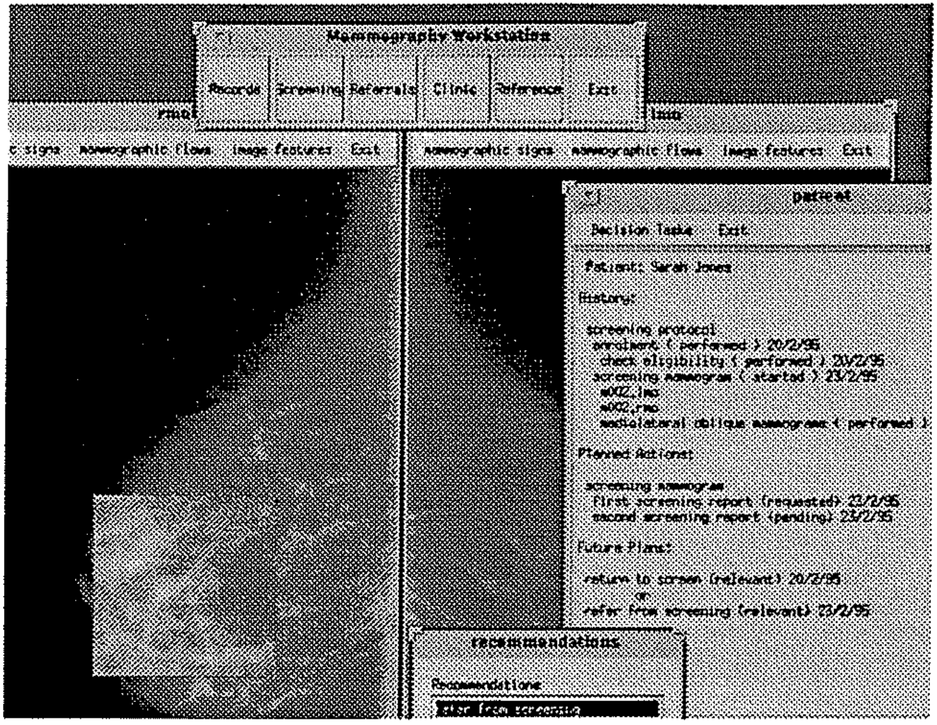
S: M Name: Paul Ferguson Age: 41 Sex: M Reference: AB21235

Date: 29/11/95

Trial: Asthma Protocol Layer: Master

CURRENTLY SCHEDULED FOR	29/11/95 15:24	29/11/95 15:25	29/11/95 15:27		
Minute	0	1	2		
Eligibility	[illegible]				
Patient Given Consent	[illegible]				
Collect Demographic Data	[illegible]				
Baseline Data Collection	Action				
Patient Randomisation	Action				
Patient self assesment 1		Action			
Hospital Clinic Test 1		Action			
Patient self assesment 2			Action		
Hospital Clinic Test 2			Action		

Fig 8: The CREW clinical trials management system. A clinical trial of a new treatment requires precise compliance with a research protocol which sets out the criteria for including patients in the trial (establish medical eligibility, patient consent), defines the data to be collected (e.g. demographic data and baseline clinical data) and the therapeutic procedure to be followed. The primary function of CREW is to schedule and manage these tasks over time, under the control of the clinical research assistant who interacts with the task chart in the bottom window. CREW was developed by Paul Ferguson, now at Chelsea and Westiminster Hospital, and Jonathan Gee, now at the Hammersmith Hospital, London.



```
Patient: Sarah Jones

History:

 screening protocol
  enrolment ( performed ) 20/2/95
   check eligibility ( performed ) 20/2/95
  screening mammogram ( started ) 23/2/95
   m002.lmo
   m002.rmo
   mediolateral oblique mammograms ( performed )

Planned Actions:

 screening mammogram
  first screening report (requested) 23/2/95
  second screening report (pending) 23/2/95

Future Plans:

 return to screen (relevant) 20/2/95
        or
 refer from screening (relevant) 23/2/95
```

Fig 9: CADMIUM, a radiology workstation which combines decision support with image interpretation. On the left are mammograms taken as part of a breast cancer screening protocol. On the right is the patient record, which records background information about the patient, and summarises past activities and schedules those as they become necessary. Future activities are selected on the basis of clinical decisions. In the process of decision making the system may automatically interrogate the images in order to extract information which is relevant to a decision. CADMIUM was developed by Paul Taylor.

The most complete implementation of the DOMINO model is an intelligent agent developed in the RED project (DTI/SERC project ITD 4/1/9053). An application of the agent to the management of acute asthma in the accident and emergency department is illustrated in Figure 10.

The relationships between these systems can be understood by referring to figure 11. This shows the standard DOMINO model but also indicates those reasoning paths which are automated or semi-automated in the different applications. Most of the systems deduce decision candidates, construct arguments and schedule tasks entirely automatically. The clinical trials system CREW, on the other hand, only automates the

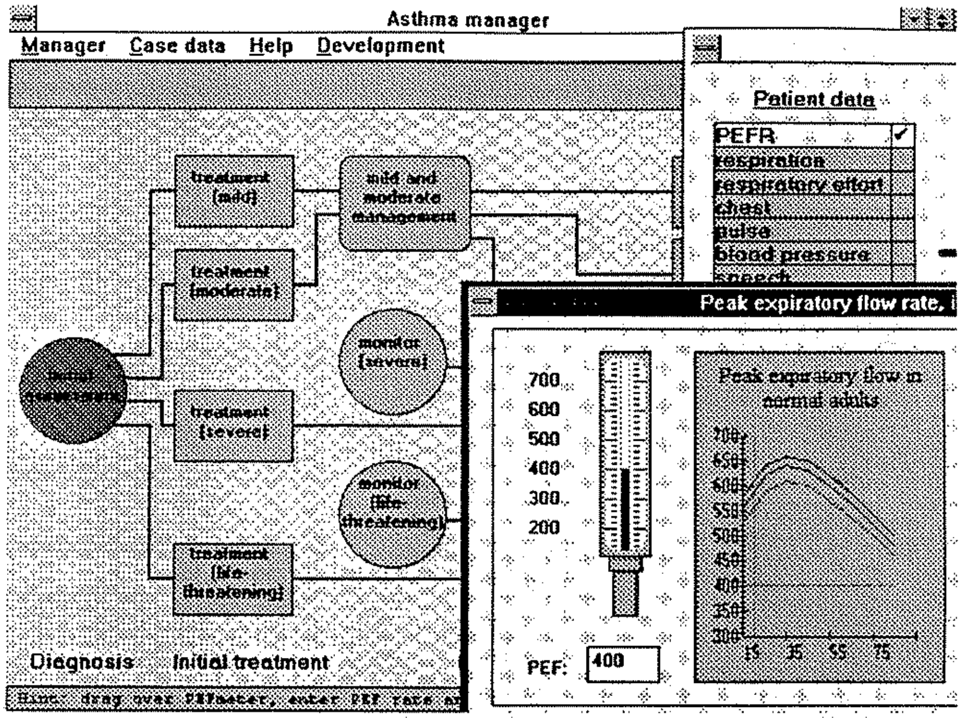


Fig 10: Decision support and guideline management technology used in an accident and emergency application: the management of acute asthma in adults. The decisions and actions recommended by the guideline are shown as a graphical workflow chart which reminds clinical staff of the tasks that need to be carried out. The panels on the right prompt for relevant patient information using easily understood forms. Colour and other kinds of coding show the need for specific decisions or actions or flag hazardous conditions which may arise. The application is based on a guideline published by the British Thoracic Society; it includes a hypertext document providing access to the original text and figures. The applciation was developed in collaboration with Integral Solutions Ltd, Basingstoke, UK and Paul Ferguson, now at the Chelsea and Westminster Hospital, London.

scheduling function. Only the CADMIUM radiology workstation is capable of automatically interrogating medical images and only the RED asthma management system is capable of generating decision-making goals without human intervention. As remarked earlier, the processes of commitment - to new beliefs or to plans or actions, could be carried out automatically in any of these systems, but in practice preferences are only offered as recommendations; the decision should only be taken by a trained doctor.

4. Development of DOMINO applications

The DOMINO model has been embodied in a general purpose reasoning engine which provides the required inference functions. As with an expert system shell, in order to build an application we have to provide an application-specific knowledge base. Most medical knowledge is not available in anything approaching the necessary formalism, however, so the creation of a knowledge base has to be carried out in a number of steps. Figure 12 summarises the current methodology for development of DOMINO applications.

The first step is concerned with acquiring and structuring relevant sources of knowledge (textbooks and other published documents, records of interviews etc) using text processing tools to extract, organise and index relevant material.

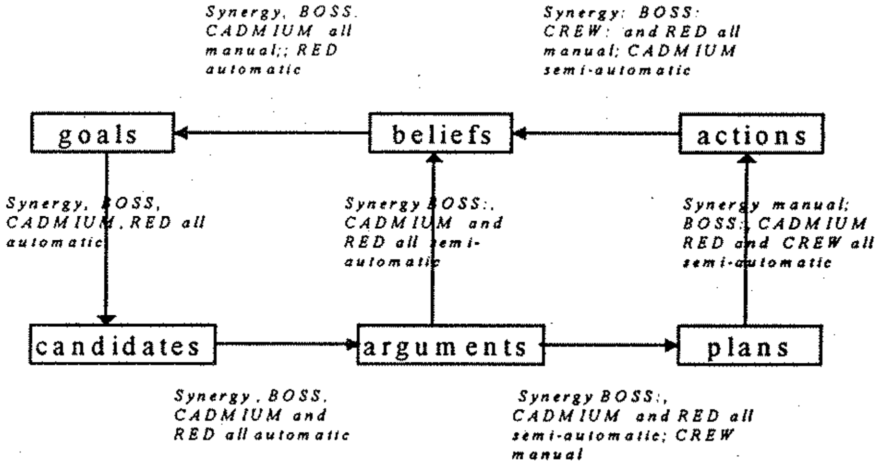

Synergy, BOSS. CADMIUM all manual;; RED automatic
Synergy: BOSS: CREW: and RED all manual; CADMIUM semi-automatic
goals
beliefs
actions
Synergy, BOSS, CADMIUM, RED all automatic
Synergy BOSS:, CADMIUM and RED all semi-automatic
Synergy manual; BOSS:, CADMIUM RED and CREW all semi-automatic
candidates
arguments
plans
Synergy, BOSS, CADMIUM and RED all automatic
Synergy BOSS:, CADMIUM and RED all semi-automatic; CREW manual

Figure 11. Relationship between SYNERGY, BOSS, CREW, RED and the DOMINO

The second step is to define the overall structure of the application, in terms of the procedures, plans, decisions and other components required for the knowledge base. The task network for the management of acute depression shown in figure 5 is an example. In the graphical notation we use decisions are normally indicated by circles and plans by rectangles (the latter can be decomposed into subplans and primitive actions, as shown by the embedding). The arrows linking decisions and plans represent ordering and temporal constraints on the execution of these tasks which are to be taken into account by the task scheduler embodied in the DOMINO software. (Note that we do not use a conventional flow diagram notation because task networks are not flow diagrams; flow diagrams are procedural while the purpose of the notation is to provide a declarative representation of task structure.)

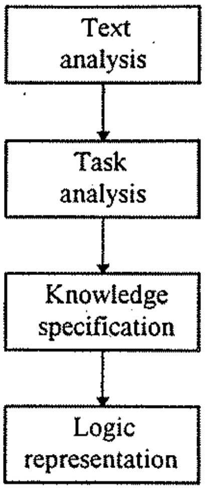


Fig 12: application development methodology

The next step is to implement the detailed knowledge associated with each of the decisions, plans etc in the task network. The graphical network is transformed into a partially instantiated logic database representing the set of tasks, and then the database is populated with the necessary details using an appropriate syntax directed editor. The editor generates the knowledge in a formal language, the Red Representation Language (R^{2}L), which provides a method of specifying the necessary constructs (decisions, plans, information sources, actions etc) as objects with a well-defined *object-attribute-value* structure. Decisions have the following general structure:

decision <decision ID>

situation <set of beliefs> d1

problem <goal> d2

candidates <set of candidates> d3

oracles <set of information sources> d4

arguments <set of first-order argumentation rules> d5

commitments <set of first-order commitment rules> d6

This structure reflects the entities and inference rules on the decision making (left) side of the DOMINO. Decisions need to be taken in particular situations which are considered to be problems in the application domain (d1,d2: e.g. an abnormal medical observation requires diagnosis and, once the diagnosis is known, therapy). Given a decision problem there may be one or more possible solutions (candidates, d3) and one or more relevant information sources (oracles, d4) which are relevant to deciding among the candidates. The decision procedure applies an argumentation theorem prover to the argumentation rules (d5) together with information in the situation data (d1) in order to derive pros and cons of the different decision candidates. Finally, when one or more of the commitment rules associated with a decision is satisfied (d6) the decision is taken. This results of decisions, such as a diagnosis of the abnormality, or an intension to carry out a treatment plan, are added to the database of beliefs. Plans have the following structure:

```
plan <plan ID>

        preconditions <set of beliefs>                                          p1

        components <set of subplans or actions (subtasks)>                      p2

        scheduling <set of ordinal or temporal constraints on subtasks>         p3

        termination conditions <set of beliefs>                                 p4
```

Once plans are committed for execution the scheduler will schedule the plan as and when the preconditions of the plan become satisfied (p1). Once a plan is scheduled all the subplans and actions (p2) are scheduled, taking into account any constraints on the order or timing indicated in the plan definition (p3). This is a recursive process which bottoms out in atomic actions, typically involving messages to the user interface or to other agents. The scheduler is sensitive to situation updates; it can terminate plan execution at any time if termination conditions become satisfied (p4). For example if a hazard of some sort arises, such as a patient having an allergic response to a medication, then the plan should be aborted and any scheduled actions cancelled.

The final step in the development methodology is to translate the R^2L definition into an executable knowledge base. The executable language (LR^2L) supports all the basic types of logical inference shown in figure 1, incorporating a sub-language for specifying temporal conditions, integrity and safety constraints and obligations. LR^2L does not have primitive constructs for the R^2L concepts of decisions, plans etc, but these are supported with more primitive notions. LR^2L is therefore a lower level but more expressive language than R^2L; in principle we believe it could be used to implement different architectures than the one discussed here. LR^2L is interpreted by a specialised theorem prover written in Prolog (Das et al, 1995).

5. Lessons learned

We believe that the body of experience gained in building the applications described in this paper has much to say to researchers in formal reasoning. We need their help in return. Medical decision support systems are safety-critical, they can kill people if poorly designed or implemented. Having committed much effort to providing a formal foundation for our work we have become aware how difficult some of the underlying issues are (including logical, computational and epistemological issues). In contrast to the common expectations of medical informaticians we believe that the development of successful decision support systems is not just a matter of good practice in software implementation and testing. Important as these matters are, we also require a deep understanding of many theoretical questions. Consequently we would strongly encourage the FAPR community to become actively involved in the medical field and support clinical technologists in their desire to build safe and sound applications.

Medicine is one of the richest fields of human practical reasoning so there is much to be gained in return for such an effort, even for the very theoretically minded. In the remainder of this paper, therefore, we shall try to underline some of the theoretical challenges medicine raises by outlining some of the lessons learned that impact on conference themes, together with some important outstanding questions. Among the themes that our work addresses are: theories of argumentation and aggregation; reasoning about beliefs, actions and plans; formal models of reasoning and integrated reasoning mechanisms; temporal reasoning; agent theory and logic programming.

Theories of argumentation and aggregation.

Medical problems frequently involve high levels of uncertainy. This uncertainty may be about what to believe (e.g. what is wrong with somebody) or uncertainty about what to do (e.g. what would be the best treatment). The orthodox approach to the management of uncertainty in decision support systems is to quantify the uncertainty in statistical or other terms and use standard probabilistic, fuzzy, or other methods to compute degrees of belief (e.g. in diagnostic propositions) or expected utilities (e.g. for alternative actions) in order to establish a preference ordering over the candidates. The challenge raised by medicine is that there is frequently no reliable basis for establishing objective quantitative parameters.[2] Nevertheless we must deal with the uncertainty we face in the practical world, how may we do this?

Classical logic helps us very little, of course, and even non-monotonic logics are of limited value because they provide no notion of *degrees* of belief, *relative amounts* of evidence etc.

For these reasons we have developed the intuitive idea of *argumentation* into a formal theory for reasoning about uncertain propositions in the absence of quantitative information (Fox, 1986) and the *aggregation* of collections of arguments in order to establish preferences over competing propositions (Fox et al, 1988). As remarked earlier, however, these ideas were somewhat ad hoc when first introduced so we have since put considerable effort into formalising them. The result is the logic LA, a variant of intuitionistic logic which provides a sound axiomatisation of argumentation as a general method for decision making under uncertainty (Fox et al, 1992; Krause et al, 1995; Ambler, 1996; Das et al, 1996).We believe that LA provides a very general framework for practical reasoning under uncertainty which is more flexible than, but potentially compatible with, orthodox probability theory.

[2] Some advocate the use of subjective opinion as a substitute for objective probability but we know from research in psychology that such estimates are unreliable, due to well-known biases in human judgement. In any case there is no scientific justification for interpreting subjective levels of confidence as mathematical probabilities.

Argumentation, however, is a complex subject which raises many issues. Among the questions which arise in medicine are the following. When is it safe to act on a diagnosis (what makes an argument for or against a proposition sound and persuasive?). How should we handle evidence which may not be reliable or exploit promising medical theories which are not yet proven? (how do we deal with arguments which have some *force* but are not *conclusive*?). How do we decide whether or not a partcular body of medical knowledge is relevant to a particular case? (when is an argument relevant to a proposition and when irrelevant?). A common feature of many such questions is that they involve *meta-level reasoning about arguments*, for which at present we have little theoretical understanding but which formal analysis may shed light on.

Reasoning about beliefs, actions and plans.

The formulation and aggregation of a set of arguments to support some clinical conclusion, such as "the patient is suffering from cancer" or "chemotherapy is an appropriate course of action", may establish a preference over the alternatives (the patient does not have cancer; chemotherapy is not appropriate). In some cases the arguments may be conclusive (as when an imaging technique visualises the presence of a tumour, or chemotherapy is the only available therapy). In others there is residual doubt. Under this condition we require safe and sound rules for taking action. That is to say we must be able to prove that, given what we know at the time, our choice is the *best* choice before we make a commitment that could prove to be wrong, and possibly disastrous.

In practical medicine a wide range of policies may be adopted in. A *risk-averse* clinician, for example, might say "we should obtain all relevant information about a patient before committing to action". A more *resource-aware* doctor may say "there is no point in spending time, effort and money on further investigations if none of the possible remaining tests could result in a different preference ordering". Frequently general practitioners appear to adopt apparently ad hoc policies like "if there is any possibility of my patient suffering from a life-threatening condition, however small, I will refer the patient to an appropriate specialist". Or alternatively, "I know enough about this aspect of medicine; I can take treat the patient without a second opinion".

What theory might sanction or refute such policies? How may we formalise risk in logical terms? How do we provide a semantics for the concepts of subjective values or professional ethics? When is it reasonable for an agent to act upon a possibility, as distinct from a probability? and how may an agent reason soundly about what it knows and doesn't know?

Formal models of reasoning and the integration of reasoning models.

The computer science literature, such as that on medical expert systems and logic programming, tends to make the assumption that medical reasoning is just diagnostic reasoning. Clearly it is much more than that. First of all it involves reasoning about goals, which is to say that an agent has a problem if it finds itself in some situation that challenges its goals, or threatens to challenge them in the future. Practical medical reasoning must therefore embody a theory of goals and an understanding of different ways of achieving and maintaining goal states. Adaptive behaviour in the face of clinical problems will sometimes only require simple actions (e.g. prescribe some medication) but often we shall have to put plans in place for long term therapy, perhaps involving many individuals from different medical disciplines over many years. We have mentioned the example of cancer chemotherapy many times, but this is also true of the management of chronic conditions such as arthritis, diabetes, hypertension and many neurological conditions. All this has to be done in the face of considerable uncertainty about the state of the world and the efficacy of our actions for achieving our goals in the world.

To address problems of this complexity we need to combine many different types of reasoning. To be confident of them they must be formalised in some well understood manner, as a set of logics or in some other way. The DOMINO model is our own attempt to meet this objective; it provides a framework in which we can reason about goals and problems; possible solutions to problems; arguments for beliefs, plans and actions; reasoning about time, scheduling and obligations. We hope that the intuitive model we have presented here, and our formalisation in the accompanying paper, will prove to be of interest to theoreticians.

We are aware, however, that our framework is neither complete nor unique.

It is incomplete in several ways. We have provided no theory of dynamic planning, only a theory for selecting among alternative plans (protocols or guidelines). Furthermore the safe execution of plans involves constantly reasoning about the consequences and side-effects of plans (Fox 1993; Hammond et al, 1994) so we need some theory of knowledge-based forecasting. Also we need to develop some theoretical framework for perception of the environment (e.g. regarding the detection of hazardous events in the clinical environment, or abnormal structures in images).

The framework is not unique. Actions and guidelines/protocols are equivalent to *operators* and *skeletal plans* which are discussed in the AI literature on planning. There is a considerable body of work on goal-based reasoning, logics of belief, knowledge, planning and acting etc. Perhaps there are many "dominos", composed from different logics, that interact in different ways to yield different behaviour in the same conditions. While different, these behaviours may be equally "rational" in that they satisfy normative axioms but represent different tradeoffs for different kinds of environment (recall our risk-averse and resource-aware doctors - who is right?). Our conclusion from this is that we require a meta-theory of practical reasoning which acknowledges the possibility of many different instantiations of rationality, and some well understood rules for composing different logics for different purposes.

Agent theory

The DOMINO can be viewed as a collection of logics, or alternatively as an architecture for an agent which unites a set of practical reasoning methods. As an agent theory the model is similar to the BDI agent concept (Wooldridge and Jennings, 1995): they share mentalistic ideas like beliefs and "desires" (goals), and guidelines and protocols seem similar in spirit to the "intentions" of the BDI agent.

The main contribution of our work to agent theory is the proposal embodied in the DOMINO model for the integration of plan management with functions for reasoning about goals and unpredictable events and making decisions under uncertainty. Generally, goals will require hypothetical and/or practical reasoning but many problems may be solved in more than one way. To deal with this we introduced the idea that agents may construct arguments for and against different candidate solutions. Decision taking may involve the aggregation of arguments to induce a preference ordering over the candidates, committing to one or other of the options based on these preferences or on other decision policies which offer different tradeoffs for the agent.

An important area of future development is in multi-agent planning. As remarked, the care of an ill person often requires severeal different clinical disciplines and hence communication between and coordination of individuals in different places. Huang et al (1995) have explored ways of supporting interactions in healthcare teams and between their decision support systems which will ensure that decision making, planning and acting take place in a coordinated way. To do this Huang et al built a communication and coordination layer on top of an agent implementation; this provided a set of communication primitives for informing other agents of needs or data, requesting information or actions, and establishing and maintaining responsibilities for different tasks be members of the care team. This language is practically promising but, as with our earlier work, their practical solution may be theoretically ad hoc; we look forward to working with formalists to establish a firmer foundation for a technology which is likely to be of growing importance.

6. Conclusions

We have developed a general model for knowledge based decison making and plan management based on an integration of non-standard logics. The model has been validated on a range of medical applications. It shows practical promise, as well as addressing a number of theoretical problems for formal and applied practical reasoning. These include issues concerning argumentation and the management of uncertainty; reasoning about actions and plans; integration of reasoning systems, and agent architectures and theories. Perhaps our main conclusion is to urge computational logicians and other theorists to address problems in complex fields such as medicine, in part because they raise many important problems and challenges to theory, and in part because technologists in the field need more formal tools if they are to successfully develop technologies which can be trusted.

7. References

Ambler, S "A categorical approach to the semantics of argumentation" *Mathematical structures in computer science* (in press).

Das S, Fox J and Krause P "A unified framework for mathematical reasoning (2): theoretical foundations (submitted), 1995a.

Das S, Fox J, Elsdon D and Hammond P "Making safe decisions" (submitted) 1995b

Fox J "Three arguments for extending the framework of probability" *In Proc. 1st Conference on Uncertainty in Artificial Intelligence*, Los Angeles; Machine intelligence and pattern recognition 4, Amsterdam: North Holland, 1986.

Fox J "Decision theory and autonomous systems" in MG Singh and L Trave-Massuyes (ed) *Proc IMACS International Conference on Decision Support Systems and Qualitative Reasoning*, Amsterdam: Elsevier, 1991.

Fox J "On the soundness and safety of expert systems" *Artificial Intelligence in Medicine, 5*, 159-179,1993

Fox J, Glowinski A J, O'Neil M "The Oxford System of Medicine: a prototype information system for primary care" *Lecture notes in Medical Informatics*, Berlin: Springer, 213-226. 1987.

Fox J "Decision making as a logical process" In B Kelly and A Rector (eds*) Research and development in expert systems*, Cambridge: Cambridge University Press, 1988.

Fox J, Clark D A, Glowinski A J and O'Neil M "Using predicate logic to integrate qualitative reasoning and classical decision theory" *IEEE Trans. Systems Man and Cybernetics*, 20(2), 347-357, 1990.

Fox J, Krause P and Ambler S "Arguments, contradictions and practical reasoning" in B Neumann (ed) *ECAI 92: Proc. 10th Eur. Conf. on AI*, 623-627, 1992.

Fox J and Krause P "Symbolic decision theory and autonomous systems" in D'Ambrosio, B Smets P and Bonissone P (eds) *Proc. 7th Conference on Uncertainty in AI*, 103-110, Morgan Kaufman, 1991.

Hammond P, Harris A L, Das S K and Wyatt J "Safety and decision support in oncology" *Methods of Information in Medicine*, 33 (4), 371-381, 1994.

Huang J, Fox J, Gordon C and Smale A "Symbolic decision support in Medical Care" *Artificial Intelligence in Medicine*, 5 (5), 415-430, 1993.

Huang J, Jennings N R and Fox J "An agent based approach to healthcare management" *Int. J Applied Artificial Intelligence*, 9, 4, 1995

Johnstone M, Langton K B, Haynes B and Matheu A "Effects of computer-based clinical decision support systems on clinician performance and patient outcome, a critical appraisal of research" *Annals of Internal Medicine*, 120, 135-142, 1994.

Krause P, Ambler S, Elvang-Goransson M and Fox J, "A logic of argumentation for reasoning under uncertainty" *Computational Intelligence*, 11 (1), 1995.

Musen M, Tu S, Das A K and Shahar, Y "A component based architecture for automation of protocol-directed therapy", Stanford University Medical School, 1995.

Renaud-Salis J L and Taylor P "The Bordeaux Oncology Support System: Knowledge Representation and prototype" Project LEMMA technical report, Imperial Cancer Research Fund, February 1990

Shortliffe E S "An integrated oncologist's workstation" *National Cancer Institute*, 1990.

Taylor P "Decision support for image interpretation: A mammography workstation" in Bizais, Barillot and Di Paola (eds) *Image processing and medical imaging*, Dordrecht: Kluwer, 1995.

Walton R and Gierl C in Jackson-Smale ed "Evaluation of primary health care decision support systems" DILEMMA project deliverable D11, Imperial Cancer Research Fund, 1995

Wooldridge M and Jennings N "Intelligent agents: theory and practice" *The Knowledge Engineering Review*, 10 (2), 115-152, 1995

A SIMPLE DECLARATIVE LANGUAGE FOR DESCRIBING NARRATIVES WITH ACTIONS

To appear in the Journal of Logic Programming Special Issue on Reasoning about Actions (to be published end of 1996)

ANTONIOS KAKAS AND ROB MILLER

▷ We describe a simple declarative language $\mathcal{E}$ for describing the effects of a series of action occurrences within a narrative. $\mathcal{E}$ is analogous to Gelfond and Lifschitz's Language $\mathcal{A}$ and its extensions, but is based on a different ontology. The semantics of $\mathcal{E}$ is based on a simple characterisation of persistence which facilitates a modular approach to extending the expressivity of the language. Domain descriptions in $\mathcal{A}$ can be translated to equivalent theories in $\mathcal{E}$. We show how, in the context of reasoning about actions, $\mathcal{E}$'s narrative-based ontology may be exploited in order to characterise and synthesise two complementary notions of explanation. According to the first notion, explanation may be partly modelled as the process of suitably extending an apparently inconsistent theory written in $\mathcal{E}$ so as to establish consistency, thus providing a natural method, in many cases, to account for conflicting sets of information about the domain. According to the second notion, observations made at later times can sometimes be explained in terms of what is true at earlier times. This enables domains to be given an alternative characterization in which knowledge arising from observations is appropriately separated from other aspects of the domain. We also describe how $\mathcal{E}$ domains may be implemented as Event Calculus style logic programs, which facilitate automated reasoning both backwards and forwards in time, and which behave correctly even when the knowledge entailed by the domain description is incomplete. ◁

Address correspondence to ANTONIOS KAKAS, Deptment of Computer Science, University of Cyprus, 75 Kallipoleos Street, P.O. Box 537, CY-1678 Nicosia, CYPRUS. Email: antonis@turing.cs.ucy.ac.cy *or* ROB MILLER, Department of Computing, Imperial College, 180 Queen's Gate, London SW7 2BZ, U.K. Email: rsm@doc.ic.ac.uk, http://www-lp.doc.ic.ac.uk/UserPages/staff/rsm/rsm.html

An earlier version of this paper (August 1995) appeared as a joint Imperial College / University of Cyprus Research Report (Imperial College Research Report Number DoC 95/12)

1. INTRODUCTION

This paper largely concerns narrative reasoning, that is, reasoning about actions which actually occur at various times, and reasoning about the properties that hold or do not hold at different times as a consequence of such occurrences. The importance of narrative reasoning has been recognised elsewhere (see [2], [32] or [30]). For example, to deal with *observations* a formalism must allow the representation of a narrative, since phenomena can only be 'observed' at actual times. In the context of the Situation Calculus it makes little sense to state that Fred is observed to be alive in the 'hypothetical' or 'projected' situation

$$Result(Shoot, Result(Wait, Result(Load, S0)))$$

unless there is some extra mechanism to relate the sequence of actions *Load - Wait - Shoot* to the time at which the observation took place.

For the purposes of discussion, we will informally define a *narrative-based formalism* as a formalism in which the structure or flow of time is represented independently from the notion of an action, and in which actions are 'embedded' in this independent structure using an explicit notion of an action *occurrence*. Examples of such formalisms are Allen's interval-based approach [1], Sandewall's *Features and Fluents* framework [34] (where a series of action occurrences are captured as a *schedule*) and Kowalski and Sergot's Event Calculus [23]. On the other hand, formalisms such as McCarthy and Hayes' Situation Calculus [28], Dynamic Logics (see for example [17]) and Gelfond and Lifschitz's Language $\mathcal{A}$ [15] are not narrative-based. This is not to say that they cannot be extended to deal with narrative information (see for example [32], [30] or [25]). But the notions of an independent flow of time and of an action occurrence are not central to their underlying ontology.

In [15] and [16], Gelfond and Lifschitz proposed a particular methodology for research into reasoning about action. The authors introduced the Language $\mathcal{A}$ as a "simple declarative language for describing actions", and suggested that, by describing general translation procedures from $\mathcal{A}$ domains into other formalisms, an insight could be gained into the comparative possibilities and limitations of each approach. The success of this idea is exemplified in a paper by Kartha [20], which uses translations from $\mathcal{A}$ to show the equivalence of three well-known characterisations of the Situation Calculus for a whole class of domains. The primary intention of $\mathcal{A}$ was to provide a language and semantics simple enough to be regarded as uncontroversial and intuitive, even if (initially) somewhat limited in expressivity. Of course, the 'neutrality' of any such language, used as a measuring stick for other formalisms, will inevitably be compromised to some extent by the necessity of choosing a particular ontology as a starting point. The ontology underlying the Language $\mathcal{A}$ is inherited from the Situation Calculus.

This paper shows that the methodology described above need not be limited to this particular ontology. Our central aim is to propose and develop a simple declarative language for describing narratives, called $\mathcal{E}$, based not upon the ontology of the Situation Calculus, but instead upon a narrative-based ontology similar to that of the Event Calculus. Furthermore, we aim to use $\mathcal{E}$ as a specification for developing logic programs for automated reasoning about action and change in a principled manner. We believe that the use of, and comparison between, different ontologies is vital in the study of reasoning about action. Central issues such as the *frame problem*, the *ramification problem* and the *qualification problem* all take on different flavours when set in different ontological contexts. Comparisons between approaches can help reveal which aspects of these problems are fundamental, and which are merely the product of a particular method of representation.

In particular, the notion of *persistence* (or 'inertia') is somewhat different in a narrative setting. A simple declarative characterisation of persistence is central to the semantics of $\mathcal{E}$ described below. This semantics, like that of $\mathcal{A}$, is model-theoretic, and the characterisation of persistence is achieved by listing three specific conditions which each model must satisfy. The definition of a model is intended to be modular in the sense that, in future extensions to $\mathcal{E}$, further conditions or constraints not relating to persistence may simply be added. (To illustrate this point, two simple extensions to $\mathcal{E}$ are given in an appendix.) The semantics gives rise to a notion of entailment which is independent from any particular method of derivation or computation. Thus the Language $\mathcal{E}$ helps differentiate between the ontological commitments of the Event Calculus and the computational mechanisms provided by its original logic programming setting. In this respect it serves a purpose similar to that of the formalisms described in [37] and [7].

The paper is organised as follows. In Section 2, we describe the basic syntax and semantics of $\mathcal{E}$, give some examples and discuss some general properties of the formalism. In Section 3 we show a correspondence between the Languages $\mathcal{A}$ and $\mathcal{E}$ by describing a sound and complete translation from theories written in $\mathcal{A}$ into Language $\mathcal{E}$ domain descriptions. We also briefly discuss the relationship between $\mathcal{E}$ and the Language $\mathcal{L}_0$, a narrative extension of $\mathcal{A}$ recently proposed by Baral, Gelfond and Provetti [5]. In the next three sections, we use $\mathcal{E}$ to characterise two complementary notions of *explanation* in temporal domains. In Section 4, we show how explanation may be partly modelled as the process of extending an apparently inconsistent theory written in $\mathcal{E}$ so as to establish consistency. In particular, we show that the syntax and semantics of $\mathcal{E}$ allows a class of 'narrative-based' explanations to be identified in a natural way. We also show how different preference relations between explanations can be combined with the simple object-level definition of entailment described previously in order to define a meta-level semantics with both an abductive and a deductive flavour. In Sections 5 and 6 we show that observations made at later times may also sometimes be explained in terms of what is true at earlier times. To do this, we identify a special class of $\mathcal{E}$ domain descriptions, and separate out observations from these theories. In Section 7 we describe how $\mathcal{E}$ domains may be implemented as Event Calculus style logic programs. We do this in a way which avoids potential problems when the information entailed by the domain is 'incomplete', which could otherwise be caused by logic programming's implicit completion of the *HoldsAt* predicate. These programs also encode a limited form of reasoning backwards in time. Finally, in Section 8 we show how, for a particular class of domains, we may use some of the notions of explanation developed earlier to add a meta-level component to these implementations, in order to facilitate a more 'complete' form of automated temporal reasoning.

2. A CLASS OF LANGUAGES FOR DESCRIBING NARRATIVES OF ACTION OCCURRENCES

First, we will describe the basic syntax of the Language $\mathcal{E}$. Strictly speaking, $\mathcal{E}$ represents a family of languages, all of which use a basic ontology and vocabulary of fluents (properties), actions and time points. The progression of time is represented by an ordering relation over the set of time points. Time may either be continuous or progress via discrete steps, and need not necessarily be linear.

Definition 2.1. [Domain Language] A *domain language* is a tuple $\langle \Pi, \preceq, \Delta, \Phi \rangle$, where $\preceq$ is a partial (possibly total) ordering[1]defined over the non-empty set Π of *time points*, Δ is a non-empty set of *action constants*, and Φ is a non-empty set of *fluent constants*.

Except where the context implies otherwise, for the remainder of the paper we assume a particular domain language $\mathcal{E} = \langle \Pi, \preceq, \Delta, \Phi \rangle$. We will often write $T_1 \prec T_2$ to mean $T_1 \preceq T_2$ and $T_1 \neq T_2$.

Definition 2.2. [Fluent literal] A *fluent literal* of $\mathcal{E}$ is an expression either of the form F or of the form $\neg F$, where $F \in \Phi$.

Three types of statements are possible within $\mathcal{E}$. *C-propositions* ("c" for "causes") express the conditions under which particular actions can potentially initiate or terminate periods (i.e. intervals) in which a property holds. *H-propositions* ("h" for "happens") indicate that a particular action occurs at a particular time, and *t-propositions* ("t" for "time point") express that a particular property holds at a particular time.

Definition 2.3. [c-proposition] A *c-proposition* in $\mathcal{E}$ is an expression either of the form

$$A \textbf{ initiates } F \textbf{ when } C$$

or of the form

$$A \textbf{ terminates } F \textbf{ when } C$$

where $F \in \Phi$, $A \in \Delta$, and C is a set of fluent literals of $\mathcal{E}$.

Notation:
We shall often write c-propositions of the form "A **initiates** F **when** $\emptyset$" and "A **terminates** F **when** $\emptyset$" as "A **initiates** F" and "A **terminates** F" respectively.

Definition 2.4. [h-proposition] An *h-proposition* in $\mathcal{E}$ is an expression of the form

$$A \textbf{ happens-at } T$$

where $A \in \Delta$ and $T \in \Pi$.

Definition 2.5. [t-proposition] A *t-proposition* in $\mathcal{E}$ is an expression of the form

[1] We mean 'partial ordering' in the usual mathematical sense, i.e. $\preceq$ is reflexive, transitive and anti-symmetric. $\preceq$ should not be regarded as representing partial knowledge about a total order – our formalism would have to be modified to cope with this type of incomplete information. Although it might be argued that time is in fact linear, so that $\preceq$ should always be a total ordering, we consider partial orderings here for the sake of mathematical generality, and because in Section 3 a particular partially ordered set is useful in showing a correspondence between the Languages $\mathcal{E}$ and $\mathcal{A}$.

$$L \textbf{ holds-at } T$$

where L is a fluent literal of $\mathcal{E}$ and $T \in \Pi$.

Definition 2.6. [Domain description] A *domain description* in $\mathcal{E}$ is a triple $\langle \gamma, \eta, \tau \rangle$, where γ is a set of c-propositions, η is a set of h-propositions[2], and τ is a set of t-propositions in $\mathcal{E}$.

The semantics for $\mathcal{E}$ is expressed by defining the notion of an *interpretation*, and stating when an interpretation qualifies as a model for a given domain description. In the definitions below, an interpretation is defined simply as a mapping of fluent/time-point pairs to *true* or *false* (i.e. a *holds* relation). A model is an interpretation that respects four properties. The first three of these are intended to characterise a 'commonsense' notion about the persistence of properties as time progresses. In particular, they encapsulate the idea that all points at which a property starts (ceases) to hold are earmarked by an initiating (terminating) action occurrence – in other words, actions are the only mechanisms for change[3]. This is stated explicitly in condition (1) of Definition 2.10. Conditions (2) and (3) confirm that the terms *initiate* and *terminate* have their intended meanings, relative to this 'commonsense' principle.

Definition 2.7. [Interpretation] An *interpretation* of $\mathcal{E}$ is a mapping

$$H : \Phi \times \Pi \mapsto \{true, false\}$$

Definition 2.8. [Point satisfaction] Given a set of fluent literals C of $\mathcal{E}$ and a time point $T \in \Pi$, an interpretation H *satisfies C at T* iff for each fluent constant $F \in C$, $H(F,T) = true$, and for each fluent constant F' such that $\neg F' \in C$, $H(F',T) = false$.

Definition 2.9. [Initiation/termination point] Let H be an interpretation of $\mathcal{E}$, let $D = \langle \gamma, \eta, \tau \rangle$ be a domain description, let $F \in \Phi$ and let $T \in \Pi$. T is an *initiation-point* (respectively *termination-point*) *for F in H relative to D* iff there is an $A \in \Delta$ such that (i) there is both an h-proposition in η of the form "A **happens-at** T" and a c-proposition in γ of the form "A **initiates** F **when** C" (respectively "A **terminates** F **when** C") and (ii) H satisfies C at T.[4]

Definition 2.10. [Model] Given a domain description $D = \langle \gamma, \eta, \tau \rangle$ in $\mathcal{E}$, an interpretation H of $\mathcal{E}$ is a *model* of D iff, for every $F \in \Phi$ and $T, T', T_1, T_3 \in \Pi$ such

[2] Thus, since η can include more than one h-proposition that refers to the same time point, $\mathcal{E}$ allows for a limited form of concurrency (its semantics forces the assumption that concurrently performed actions do not interfere, for example to cancel each other's effects). In contrast $\mathcal{A}$ does not allow any concurrency, whereas its extension $\mathcal{A}_c$ [4] allows for a more general form of concurrency.

[3] This principle might be considered too strong for some domains, e.g. those involving continuous change. In this case, some distinction will be required between those properties which naturally persist (*frame fluents* in Lifschitz's terminology) and those which do not.

[4] According to this definition, not all initiation and termination points will be points of change of the fluent within a particular model. For example, if a fluent already holds before an initiation-point it will remain unchanged. Thus in Sergot's terms [35] the semantics uses "weak" initiation and termination (but see footnote, Appendix A.1).

that $T_1 \prec T_3$, the following properties hold:

1. If there is no initiation-point or termination-point T_2 for F in H relative to D such that $T_1 \preceq T_2 \prec T_3$, then $H(F, T_1) = H(F, T_3)$.
2. If T_1 is an initiation-point for F in H relative to D, and there is no termination-point T_2 for F in H relative to D such that $T_1 \prec T_2 \prec T_3$, then $H(F, T_3) = true$.
3. If T_1 is a termination-point for F in H relative to D, and there is no initiation-point T_2 for F in H relative to D such that $T_1 \prec T_2 \prec T_3$, then $H(F, T_3) = false$.
4. For all t-propositions in τ of the form "F **holds-at** T", $H(F, T) = true$, and for all t-propositions of the form "$\neg F$ **holds-at** T'", $H(F, T') = false$.

Definition 2.11. [Consistency] A domain description is *consistent* iff it has a model.

Definition 2.12. [Entailment] A domain description D *entails* the t-proposition "F **holds-at** T", written "$D \models F$ **holds-at** T", iff for every model H of D, $H(F, T) = true$. D entails the t-proposition "$\neg F$ **holds-at** T" iff for every model H of D, $H(F, T) = false$.

In keeping with our adopted methodology, two important simplifying assumptions are implicitly included in the above semantics. First, the information about the general effects of actions, expressed as c-propositions, is assumed to be complete. An analogous assumption is made about the e-propositions in a Language $\mathcal{A}$ theory. Second, the information about the occurrence of actions, expressed as h-propositions, is also assumed to be complete. (There is no directly analogous assumption in the definition of a Language $\mathcal{A}$ model, since the notion of an action occurrence is not included in its ontology.) Clearly, both these assumptions will be sources of nonmonotonicity in the language. The h-propositions not only give complete information about which actions occur, but (since $\preceq$ is assumed to be well-defined) also give complete information about the order and timing of these occurrences[5].

Condition (4) in Definition 2.10 above expresses pointwise constraints on a model which arise from the inclusion of t-propositions in the domain description. We envisage other such constraints being added in future, more expressive extensions of $\mathcal{E}$. Such extensions might also require refinements to the definitions of an interpretation or of an initiation or termination point. But we expect the characterisation of persistence encapsulated in conditions (1)-(3), which can be regarded as the 'core' of the semantics, to remain unaltered. To remain faithful to the methodology we are using, we wish here to keep the syntax and semantics of $\mathcal{E}$ as simple as possible, even at the loss of some expressivity. However, to illustrate this modular aspect of

[5] Relaxing these assumptions would be an interesting area for future research, but would inevitably lead to a more complex semantics. For example, it would be straightforward to allow for incomplete knowledge about the order and timing of action occurrences by using *temporal variables* in h-propositions, and including a fourth type of proposition in domain descriptions with which to express ordering constraints between these variables. An interpretation would then include a mapping (i.e. variable assignment) from temporal variables to time points as a second component.

the semantics we have included two simple extensions to $\mathcal{E}$ in Appendix A (relating to 'qualifications' and 'ramifications' of action occurrences).

The following two examples illustrate the effects of our model-theoretic characterisation of persistence.

Example 2.1. This example is intended to illustrate the necessity of including the first condition in Definition 2.10 of a Language $\mathcal{E}$ model above. It concerns vaccinations against a particular disease. Vaccine A only provides protection for people with blood type O, and vaccine B only works on people with blood type other than O. Fred's blood type is not known, so he is injected with vaccine A at 2 o'clock and vaccine B at 3 o'clock. For simplicity we will model time as the real number line with the usual ordering relation, so that for this example, $\mathcal{E}_v = \langle \Re, \leq, \{InjectA, InjectB\}, \{Protected, TypeO\}\rangle$. The domain description D_v consists of two c-propositions, two h-propositions and a single t-proposition:

$$InjectA \textbf{ initiates } Protected \textbf{ when } \{TypeO\}$$

$$InjectB \textbf{ initiates } Protected \textbf{ when } \{\neg TypeO\}$$

$$InjectA \textbf{ happens-at } 2$$

$$InjectB \textbf{ happens-at } 3$$

$$\neg Protected \textbf{ holds-at } 1$$

If we now consider some time later than 3 o'clock, say 4 o'clock, we can see intuitively that Fred should be protected. Now by condition (1), in all models of D_v Fred's blood group remains constant, so that in any given model, by condition (2), Fred becomes protected either at 2 o'clock or at 3 o'clock. Consequently,

$$D_v \models Protected \textbf{ holds-at } 4$$

Had condition (1) not been included in the definition of a model, it would have been possible to construct a model, for example, in which Fred's blood type "mysteriously" changed from $\neg TypeO$ to $TypeO$ at 2.30, thus rendering both vaccinations ineffective.

Example 2.2. This example shows that the Language $\mathcal{E}$ can be used to infer information about what conditions hold at the time of an action occurrence, given other information about what held at times before and afterwards. It is similar to Baker's 'murder mystery' scenario [3]. Let $\mathcal{E}_{ys} = \langle \Re^+, \leq, \{Shoot\}, \{Alive, Loaded\}\rangle$, where $\Re^+$ signifies the non-negative real numbers, and let the domain description D_{ys} consist of a single c-proposition, a single h-proposition and two t-propositions:

$$Shoot \textbf{ terminates } Alive \textbf{ when } \{Loaded\}$$

$$Shoot \textbf{ happens-at } 2$$

$$Alive \textbf{ holds-at } 1$$

$$\neg Alive \textbf{ holds-at } 3$$

Since by condition (4) in any model H of D_{ys}, $H(Alive, 1) \neq H(Alive, 3)$, then by condition (1), in all models an action must occur at some time point between 1 and 3 whose initiating or terminating conditions for the property *Alive* are satisfied at that point. The only candidate is the *Shoot* occurrence at 2, whose condition for terminating *Alive* is *Loaded*. Hence

$$D_{ys} \models \textit{Loaded} \ \textbf{holds-at} \ 2$$

Indeed, by applying condition (1) again, it is easy to see that for all n, $n \geq 0$,

$$D_{ys} \models \textit{Loaded} \ \textbf{holds-at} \ n$$

Two properties of $\mathcal{E}$ will prove useful later. The first is that $\mathcal{E}$ is monotonic as regards addition of t-propositions to domain theories (although, as observed earlier, not as regards addition of h-propositions or c-propositions). That is to say, if H is a model of a domain description $\langle \gamma, \eta, \tau \rangle$ and $\tau' \subseteq \tau$, H is also a model of $\langle \gamma, \eta, \tau' \rangle$. This follows directly from Definition 2.10. The second property of interest concerns the deterministic nature of actions' effects within a narrative, and is somewhat analogous to Lin and Shoham's notion of *epistemological completeness* [26]. Provided the domain description under consideration is consistent and contains no finite intervals of time in which an infinite number of actions occur, then the set of fluents which hold at any point T completely determines the set of fluents which hold at any later time point. This claim is made precise in the following definition and proposition.

Definition 2.13. [Occurrence Sparsity] Let $D = \langle \gamma, \eta, \tau \rangle$ be a domain description written in a language $\mathcal{E} = \langle \Pi, \preceq, \Delta, \Phi \rangle$. D and η are *occurrence-sparse* iff for any two points $T_1, T_2 \in \Pi$ there are only a finite number of h-propositions in η of the form"A **happens-at** T" such that $T_1 \preceq T \prec T_2$.

Proposition 2.1. Let D be an occurrence-sparse domain description written in a language $\mathcal{E} = \langle \Pi, \preceq, \Delta, \Phi \rangle$, and let $T_1, T_2 \in \Pi$ be such that $T_1 \preceq T_2$. Let H and H' be models of D such that for all $F \in \Phi$, $H(F, T_1) = H'(F, T_1)$. Then for all $F \in \Phi$, $H(F, T_2) = H'(F, T_2)$.

PROOF. See Appendix B.1

The occurrence of infinite numbers of actions in a finite period of time leads to interesting and/or unexpected results in many formalisms for reasoning about action. For a general discussion of this complex issue, the reader may refer to Davis [9]. However, for the remainder of this paper, we restrict our attention to domain descriptions which are occurrence-sparse, and thus deterministic in the sense of Proposition 2.1.

3. SIMULATING THE LANGUAGE $\mathcal{A}$ AS A CLASS OF LANGUAGE $\mathcal{E}$ DOMAINS

Because they use different ontologies, any correspondence between the Languages $\mathcal{A}$ and $\mathcal{E}$ must inevitably be expressed in rather artificial terms. In this section we show that Language $\mathcal{A}$ domains can be simulated or re-formulated as Language $\mathcal{E}$ domains with a branching structure $\langle \Pi, \preceq \rangle$ of time points, analogous to the branching 'tree' of situations often incorporated in formulations of the Situation Calculus. Since the Language $\mathcal{A}$ is not narrative-based and so does not directly include the notion of action occurrences, in our re-formulation an appropriate action occurrence has to be 'built in' at each point in this tree structure. In situation calculus terms, we must ensure that for each situation S in the tree, the action A 'occurs' between the situations S and $\mathit{Result}(A, S)$. To express this, we need to insert an extra time-point between S and $\mathit{Result}(A, S)$ – in graphical terms, we need to be able to refer to the arcs connecting the situation nodes in the tree. This is achieved in a simple way below by considering 'doubled' sequences of actions. Sequences of even length correspond to nodes of the tree structure (i.e. to situations), and sequences of odd length correspond to the inter-connecting arcs. A temporal ordering relation is then defined on both nodes and arcs.

Definition 3.1. [Δ-sequence] Given a set Δ of action constants, a Δ-*sequence* of Δ is defined inductively as follows:

- The empty sequence $\langle\!\langle\ \rangle\!\rangle$ is a Δ-sequence
- For each $A, A' \in \Delta$, the singleton sequence $\langle\!\langle |A| \rangle\!\rangle$, the sequence $\langle\!\langle |A|, A \rangle\!\rangle$ and the sequence $\langle\!\langle |A|, A, |A'| \rangle\!\rangle$ are all Δ-sequences
- For each $A_1, \ldots, A_n \in \Delta$, $\langle\!\langle |A_1|, A_1, \ldots, |A_n|, A_n \rangle\!\rangle$ is a Δ-sequence
- For each $A' \in \Delta$ and for each Δ-sequence $\langle\!\langle |A_1|, A_1, \ldots, |A_n|, A_n \rangle\!\rangle$, $\langle\!\langle |A_1|, A_1, \ldots, |A_n|, A_n, |A'| \rangle\!\rangle$ is a Δ-sequence

Definition 3.2. [Δ-ordering] Given a set Δ of action constants and the corresponding set Π_Δ of all Δ-sequences, the Δ-*ordering* $\leq_\Delta$ over Π_Δ is defined as follows

- For all $S \in \Pi_\Delta$, $\langle\!\langle\ \rangle\!\rangle \ \leq_\Delta\ S$
- For all $\langle\!\langle \alpha_1, \ldots, \alpha_n \rangle\!\rangle \in \Pi_\Delta$ and for all m such that $1 \leq m \leq n$, $\langle\!\langle \alpha_1, \ldots, \alpha_m \rangle\!\rangle \ \leq_\Delta\ \langle\!\langle \alpha_1, \ldots, \alpha_n \rangle\!\rangle$

Examples:
Suppose $\Delta = \{\mathit{Wait}, \mathit{Load}, \mathit{Shoot}\}$. The Δ-sequence

$$\langle\!\langle |\mathit{Load}|, \mathit{Load}, |\mathit{Wait}|, \mathit{Wait}, |\mathit{Shoot}|, \mathit{Shoot} \rangle\!\rangle$$

corresponds to the Situation Calculus term

$$\mathit{Result}(\mathit{Shoot}, \mathit{Result}(\mathit{Wait}, \mathit{Result}(\mathit{Load}, S0)))$$

and (regarding situations as arranged in a branching tree structure) the Δ-sequence

$$\langle\!\langle |\mathit{Load}|, \mathit{Load}, |\mathit{Wait}|, \mathit{Wait}, |\mathit{Shoot}| \rangle\!\rangle$$

corresponds to the inter-connecting arc between the situations

$$\mathit{Result}(\mathit{Wait}, \mathit{Result}(\mathit{Load}, S0))$$

and

$$\mathit{Result}(\mathit{Shoot}, \mathit{Result}(\mathit{Wait}, \mathit{Result}(\mathit{Load}, S0)))$$

Regarding the ordering $\leq_\Delta$, it is easy to see that, for example

$$\langle\!\langle |Load| \rangle\!\rangle \leq_\Delta \langle\!\langle |Load|, Load \rangle\!\rangle$$

$$\langle\!\langle |Load|, Load \rangle\!\rangle \leq_\Delta \langle\!\langle |Load|, Load, |Wait|, Wait, |Shoot|, Shoot \rangle\!\rangle$$

Notation:
We shall sometimes refer to the Δ-sequence

$$\langle\!\langle |A_1|, A_1, \ldots, |A_n|, A_n \rangle\!\rangle$$

simply as $A_1, \ldots, A_n$ and refer to the Δ-sequence

$$\langle\!\langle |A_1|, A_1, \ldots, |A_n|, A_n, |A'| \rangle\!\rangle$$

as $A_1, \ldots, A_n, |A'|$. Notice that in this notation

$$A_1, \ldots, A_n, |A'| \leq_\Delta A_1, \ldots, A_n, A'$$

The next definition allows us to express that, in general, the action A' occurs at $A_1, \ldots, A_n, |A'|$, so that the effects of A' are apparent at the following time point $A_1, \ldots, A_n, A'$.

Definition 3.3. [Complete occurrence set] Let $\mathcal{E} = \langle \Pi_\Delta, \leq_\Delta, \Delta, \Phi \rangle$, where Π_Δ is the set of Δ-sequences of Δ and $\leq_\Delta$ is the Δ-ordering over Π_Δ. The set η_Δ, called the *complete occurrence set* of Δ, is the set of all h-propositions of $\mathcal{E}$ either of the form

$$A \textbf{ happens-at } \langle\!\langle |A| \rangle\!\rangle$$

or of the form

$$A_n \textbf{ happens-at } \langle\!\langle |A_1|, A_1, \ldots, |A_{n-1}|, A_{n-1}, |A_n| \rangle\!\rangle$$

The following proposition shows a sense in which the Language $\mathcal{A}$ may be regarded as a special case of the Language $\mathcal{E}$.

Proposition 3.1. Let D_A be a consistent theory written in a language $\mathcal{A}$ in the sense of [15], with a set of action constants Δ and a set of fluent constants Φ. Let $\mathcal{E} = \langle \Pi_\Delta, \leq_\Delta, \Delta, \Phi \rangle$, where Π_Δ is the set of Δ-sequences of Δ and $\leq_\Delta$ is the Δ-ordering over Π_Δ. Let D_E be the domain description $\langle \gamma, \eta_\Delta, \tau \rangle$ in $\mathcal{E}$ defined as follows:

- *η_Δ is the complete occurrence set of Δ*
- *For each v-proposition in D_A of the form "L **after** $A_1; \ldots; A_m$" there is a t-proposition in τ of the form "L **holds-at** $A_1, \ldots, A_m$", and for each v-proposition in D_A of the form "**initially** L" there is a t-proposition in τ of the form "L **holds-at** $\langle\!\langle\ \rangle\!\rangle$"*
- *For each $F \in \Phi$, then for each e-proposition in D_A of the form "A **causes** F **if** $L_1, \ldots, L_n$" there is a c-proposition in γ of the form "A **initiates** F **when** $\{L_1, \ldots, L_n\}$"*
- *For each $F \in \Phi$, then for each e-proposition in D_A of the form "A **causes** $\neg F$ **if** $L_1, \ldots, L_n$" there is an c-proposition in γ of the form "A **terminates** F **when** $\{L_1, \ldots, L_n\}$"*

Then for each $F \in \Phi$ and each $A_1, \ldots, A_n \in \Delta$

- $D_E \models F$ **holds-at** $\langle\!\langle\ \rangle\!\rangle$ *if and only if*
  D_A *entails* **initially** F
- $D_E \models \neg F$ **holds-at** $\langle\!\langle\ \rangle\!\rangle$ *if and only if*
  D_A *entails* **initially** $\neg F$
- $D_E \models F$ **holds-at** $A_1, \ldots, A_n$ *if and only if*
  D_A *entails* F **after** $A_1; \ldots; A_n$
- $D_E \models \neg F$ **holds-at** $A_1, \ldots, A_n$ *if and only if*
  D_A *entails* $\neg F$ **after** $A_1; \ldots; A_n$

PROOF. Let R be the unique transition function such that there is a model (σ_0, R) of D_A (for definitions see [15]). Let $\sigma \subseteq \Phi$ be a set of fluent constants. For each $F \in \Phi$ and $A_1, \ldots, A_n, A' \in \Delta$ let the interpretation $H_{[\sigma,R]}$ be defined as follows:

- $H_{[\sigma,R]}(F, \langle\!\langle\ \rangle\!\rangle) = H_{[\sigma,R]}(F, \langle\!\langle |A'| \rangle\!\rangle) = true$ if and only if $F \in \sigma$
- $H_{[\sigma,R]}(F, \langle\!\langle |A_1|, A_1, \ldots, |A_n|, A_n \rangle\!\rangle) =$
  $H_{[\sigma,R]}(F, \langle\!\langle |A_1|, A_1, \ldots, |A_n|, A_n, |A'| \rangle\!\rangle) = true$
  if and only if $F \in R(A_n, R(A_{n-1}, \ldots, R(A_1, \sigma) \ldots))$

Clearly, for each $\sigma \subseteq \Phi$, $H_{[\sigma,R]}$ is a model of $\langle \gamma, \eta, \emptyset \rangle$, and $H_{[\sigma,R]}$ is a model of $D_E = \langle \gamma, \eta, \tau \rangle$ if and only if (σ, R) is a model of D_A in the sense of [15]. Since D_E is occurrence-sparse, by Proposition 2.1 all models of D_E are of this form, so that there is a one-to-one correspondence between models of D_A and such models of D_E, and the proposition follows directly from the definition of $H_{[\sigma,R]}$.

We conclude this section with some brief remarks about the relationship between $\mathcal{E}$ and the Language $\mathcal{L}_0$ recently introduced by Baral, Gelfond and Provetti in [5], and discussed in more detail in [6]. $\mathcal{L}_0$ is a 'narrative' extension to $\mathcal{A}$ which uses $\mathcal{A}$'s underlying Situation Calculus based ontology. Conceptually, it is close to the extension of the Situation Calculus described by Pinto and Reiter in [32]. Both are concerned with describing and reasoning about an 'actual path' through the 'tree of situations'.

A model of an $\mathcal{L}_0$ domain description is a pair (Ψ, Σ). The "Ψ" component roughly corresponds to the notion of a (partial) "transition function" in a Language $\mathcal{A}$ model. For a given domain, Ψ is characterised by a collection of *effect laws* in $\mathcal{L}_0$, which is equivalent to a set of e-propositions in $\mathcal{A}$ and may be translated into a Language $\mathcal{E}$ domain description of the form $\langle \gamma, \eta_\Delta, \tau \rangle$ as described in Proposition 3.1. Given such a translation, the "Σ" component of an $\mathcal{L}_0$ model can be regarded as an assignment of the *situation* symbol s_N (always included in $\mathcal{L}_0$'s vocabulary) to a particular Δ-sequence δ_N in Π_Δ, together with an assignment of each other situation symbol s_i to a Δ-sequence δ_i such that $\delta_i \leq_\Delta \delta_N$. (In Pinto and Reiter's terms, δ_N represents the 'actual path of situations'.) To be acceptable, the assignment δ_N must be of minimal length, subject to certain constraints which are expressed as *fluent facts*, *occurrence facts* and *precedence facts*. For example, the occurrence fact "A_1, A_2 **occurs-at** s_i" constrains δ_N to be of the form $\langle\!\langle \alpha_1, \ldots, \alpha_i, |A_1|, A_1, |A_2|, A_2, \ldots, \alpha_n \rangle\!\rangle$ (where $\langle\!\langle \alpha_1, \ldots, \alpha_i \rangle\!\rangle$ is the Δ-sequence δ_i assigned to s_i), and the precedence fact "s_i **precedes** s_j" expresses the constraints $\delta_i \leq_\Delta \delta_j$ and $\delta_i \neq \delta_j$. The minimal length requirement for δ_N corresponds to the minimisation of the *occurs* predicate in Pinto and Reiter's extended Situation Calculus.

A number of other papers which discuss the relationship between the Situation Calculus and the Event Calculus have recently been published, see for example [24], [29], [33] and [39].

4. DEALING WITH INCONSISTENT DOMAINS BY EXPLANATIONS

Like the Language $\mathcal{A}$, the Language $\mathcal{E}$ provides a very rigid way of specifying the effects of actions. Each separate effect has to be explicitly described by a c-proposition (analogous to an e-proposition in the Language $\mathcal{A}$), and the definition of entailment does not facilitate the inference of any other effects. For example, without an explicit representation of the statement "the action occurrence A terminates the property P" the statement "P is true before A, but false afterwards" gives rise to an inconsistency. This is in contrast to some other approaches to representing persistence, such as representations in circumscribed predicate calculus, which in this respect are more flexible. The Language $\mathcal{E}$ is similarly rigid in its representation of a narrative – all action occurrences must be explicitly represented by an h-proposition.

However, greater flexibility can be achieved, without altering the underlying semantics of $\mathcal{E}$, by introducing the notion of an *explanation.* Clearly, the statement that "property P is true before A, but false afterwards" is easily 'explained' by the statement "A terminates P". In this section we model the task of explanation as the task of restoring consistency, in some principled or selective way, to an inconsistent collection of facts, represented as an inconsistent domain description in $\mathcal{E}$. Under this view, explanation is a form of belief revision. The nonmonotonicity built into our language sometimes allows an inconsistent domain theory to be 'revised' simply by adding sets of propositions. We will call such sets 'explanations'. In the present context, explanations may be either in narrative or in causal terms, or in both. In other words, a given set of facts may be explained away in terms of what has happened and/or in terms of what causes what.

In Sections 5 and 6 we will extend the notion of an explanation, showing how in the context of narrative reasoning it is sometimes appropriate to regard information about what holds at earlier times as an 'explanation' of what holds at later times.

4.1. Explanations in Terms of Action Occurrences

The first class of explanations we will consider are those which can be expressed entirely in terms of action occurrences, i.e. extra h-propositions. We will use a version of Kautz's Stolen Car problem [22] as an illustration.

Example 4.1. Let $\mathcal{E}_{sc} = \langle \mathcal{N}, \leq, \{Park, Steal\}, \{Parked\} \rangle$, where $\mathcal{N}$ signifies the natural numbers, and let D_{sc} be the domain description consisting of the following two c-propositions, single h-proposition and single t-proposition:

$$Park \textbf{ initiates } Parked$$

$$Steal \textbf{ terminates } Parked$$

$$Park \textbf{ happens-at } 2$$

$$\neg \mathit{Parked}\ \textbf{holds-at}\ 6$$

By itself, D_{sc} is inconsistent, since there is no terminating action occurrence for the fluent *Parked* between 2 and 6. However, we may restore consistency by adding one or more h-propositions.

Definition 4.1. [h-explanation] Let $D = \langle \gamma, \eta, \tau \rangle$ be a domain description. An *h-explanation* for D is a (possibly empty) occurrence-sparse set η_ϵ of h-propositions, such that $\langle \gamma, \eta \cup \eta_\epsilon, \tau \rangle$ is consistent.

For example, the following are all h-explanations for D_{sc}:

$$\{\ \mathit{Steal}\ \textbf{happens-at}\ 3,\ \mathit{Steal}\ \textbf{happens-at}\ 4\ \}$$

$$\{\ \mathit{Steal}\ \textbf{happens-at}\ 4,\ \mathit{Park}\ \textbf{happens-at}\ 8\ \}$$

$$\{\ \mathit{Steal}\ \textbf{happens-at}\ 5\ \}$$

Indeed, it is obvious that we may construct an h-explanation for D_{sc} containing as many h-propositions as we like. Clearly, extra mechanisms are needed which enable us to prefer some explanations to others. The following definition reflects a very simple, set-theoretic notion of preference. However, the definition could be modified for specific domains, for example to reflect the fact that we wish to regard some types of occurrence as more likely than others.

Definition 4.2. [Preferable h-explanation][6]
Let η_ϵ and η'_ϵ be h-explanations for D. η_ϵ is *preferable* to η'_ϵ iff $\eta_\epsilon \subset \eta'_\epsilon$.

Having identified a class of explanations, such as the class of h-explanations in Definition 4.1, and a preference criterion such as that of Definition 4.2, it is possible to construct a corresponding meta-level, explanation-based 'semantics' similar to the semantics of Abductive Logic Programming [18], [19], [12]. We consider all possible extensions of a given domain description D with optimal explanations, and accept conclusions if and only if these hold in each such extension. This semantics can then be used to decide what can be safely 'inferred' from a seemingly inconsistent domain description.

Definition 4.3. [Optimal h-explanation] η_ϵ is an *optimal h-explanation* for D iff η_ϵ is an h-explanation for D and there is no other h-explanation η'_ϵ for D such that η'_ϵ is preferable to η_ϵ.

Definition 4.4. [h-model] Let $D = \langle \gamma, \eta, \tau \rangle$ be a domain description. H is an *h-model* of D iff there exists an optimal h-explanation η_ϵ for D such that H is a model of $\langle \gamma, \eta \cup \eta_\epsilon, \tau \rangle$

Definition 4.5. [h-consistency] A domain description is *h-consistent* iff it has at least one h-model.

[6] In this definition and throughout the paper "$\subset$" is intended to mean "is contained in and not equal to".

Definition 4.6. [h-entailment] The domain description $D = \langle \gamma, \eta, \tau \rangle$ *h-entails* the t-proposition "F **holds-at** T", written "$D \models_h F$ **holds-at** T", iff for every h-model H of D, $H(F, T) = true$. D h-entails the t-proposition "$\neg F$ **holds-at** T" iff for every h-model H of D, $H(F, T) = false$.

H-entailment is an abductive notion in the sense that optimal h-explanations are not derived from or entailed by a domain description, but are added to it (according to an external preference criterion). It also has a 'deductive flavour' in the sense that a t-proposition is entailed from a domain description simply if it is true in all h-models. And any procedure for verifying truth in all h-models will, explicitly or implicitly, have to take into account *all* optimal h-explanations.

The important point is that h-entailment is a meta-level concept, whereas the entailment relation defined in Section 2 is object-level. H-entailment is defined both in terms of object-level entailment and a particular preference criterion among explanations. Notice that Definitions 4.3 to 4.6 would still be applicable even if 'h-preferability' were to be defined differently (perhaps according to domain-specific considerations), but would yield different results.

A desirable property of any such meta-level entailment is that it should coincide with object-level entailment whenever the domain description is consistent. The following proposition shows that for h-entailment as defined Definition 4.6 above, this is indeed the case.

Proposition 4.1. Let D be a consistent domain description. Then H is a model of D if and only if H is an h-model of D.

PROOF. The proposition follows directly from the observation that since D is already consistent, it has a unique optimal h-explanation $\emptyset$.

Returning to the Stolen Car problem, D_{sc} has three optimal h-explanations:

$$\{ \, Steal \textbf{ happens-at } 3 \, \}$$

$$\{ \, Steal \textbf{ happens-at } 4 \, \}$$

$$\{ \, Steal \textbf{ happens-at } 5 \, \}$$

D_{sc} is therefore h-consistent, and has a total of six h-models (two corresponding to each optimal h-explanation), since in any h-model H, $H(Parked, 0)$, $H(Parked, 1)$ and $H(Parked, 2)$ may either all be *true* or all be *false*. It is easy to see, therefore, that

$$D_{sc} \models_h Parked \textbf{ holds-at } 3$$

and for all $n \geq 6$,

$$D_{sc} \models_h \neg Parked \textbf{ holds-at } n$$

Hence this example illustrates how the notion of h-entailment exploits the narrative ontology of $\mathcal{E}$ to give a natural and simple way to handle such apparently inconsistent domains.

H-entailment corresponds closely to Shanahan's formulation of explanation-based temporal reasoning in [36]. The main difference is that whereas Shanahan's concern is to abduce a single explanation for a given fact or observation, ours is to 'safely' infer new information by considering all (optimal) explanations. Shanahan's work was partly based on earlier work by Eshghi [14] showing how *planning* could be formulated within an abductive Event Calculus framework. In Language $\mathcal{E}$ terms, an initial state I and goal state G can both be represented as sets τ_I and τ_G of t-propositions, domain information can be modelled as a set γ of c-propositions, and a plan can be regarded as a single h-explanation η_e for $\langle\gamma, \emptyset, \tau_I \cup \tau_G\rangle$ such that $\langle\gamma, \eta_e, \tau_I\rangle \models \omega$ for each $\omega \in \tau_G$.

Our notion of an h-model is also somewhat analogous to the description of a Language $\mathcal{L}_0$ model given by Baral, Gelfond and Provetti in [5] and [6] (see end of Section 3). An important difference here is that, whereas the requirement that action occurrences be minimal is 'hardwired' into $\mathcal{L}_0$'s object-level semantics, our minimality requirement is to be found in the particular definition of h-preference, which could potentially be replaced by or extended with other, domain-specific preference criteria. Indeed, in the following section we give an example of a preference criterion among a class of explanations which is not entirely based around a simple notion of minimality.

4.2. Explanations in Terms of New Causal Rules

It will not always be possible to find an h-explanation for an inconsistent domain description D. In this section we briefly explore the possibilities of including both c-propositions and h-propositions in explanations. The discussion here is intended only to illustrate some of the problems relating to this – we are obviously trespassing into the related A.I. topic of *learning*. We will use the following (deliberately abstract) example to motivate the discussion:

Example 4.2. Let $\mathcal{E}_{ex} = \langle\mathcal{N}, \leq, \{A_1, A_2\}, \{F_1, F_2\}\rangle$, where $\mathcal{N}$ signifies the natural numbers, and let the domain description D_{ex} consist of three t-propositions:

$$F_1 \textbf{ holds-at } 4$$

$$F_2 \textbf{ holds-at } 5$$

$$\neg F_2 \textbf{ holds-at } 10$$

This example is neither consistent nor h-consistent. To establish consistency in this case we need to consider explanations which include both narrative and causal information.

Definition 4.7. [hc-explanation] Let $D = \langle\gamma, \eta, \tau\rangle$ be a domain description. An *hc-explanation* for D is a pair $\langle\gamma_\epsilon, \eta_\epsilon\rangle$, where γ_ϵ is a (possibly empty) set of c-propositions and η_ϵ is a (possibly empty) occurrence-sparse set of h-propositions, such that$\langle\gamma \cup \gamma_\epsilon, \eta \cup \eta_\epsilon, \tau\rangle$ is consistent.

For example, according to Definition 4.7 the following are all hc-explanations for D_{ex}:

1. $\langle\{A_1 \textbf{ terminates } F_2 \textbf{ when } \{F_1\}\}, \{A_1 \textbf{ happens-at } 8\}\rangle$
2. $\langle\{A_1 \textbf{ terminates } F_2\}, \{A_1 \textbf{ happens-at } 8\}\rangle$

3. $\langle\{A_1$ **terminates** F_2 **when** $\{\neg F_1\}$, A_2 **terminates** $F_1\}$, $\{A_2$ **happens-at** 7, A_1 **happens-at** $8\}\rangle$

What qualifies as a 'reasonable' preference criterion among hc-explanations? In practice such a criterion will probably be domain-specific. Nevertheless, it is interesting to speculate at an abstract level, if only to discover the complexity of some of the issues involved. A simple or liberal view (in the sense that it would allow a wide class of optimal hc-explanations) would be to prefer hc-explanation ϵ over hc-explanation ϵ' only if ϵ's set of c-propositions was strictly contained in ϵ''s. However, this policy would not yield any preferences between the three hc-explanations in Example 4.2 above. Regarding D_{ex} as representing three observations at different time points, it seems reasonable that we should prefer explanations which account for the observed change in F_2 but allow F_1 to persist. This would, for example, cut out explanation (3) above. The use of $\mathcal{E}$ to describe the process of learning new c-propositions from such sets of observations is the subject of further investigation.

5. PROJECTION DOMAIN DESCRIPTIONS

In the following two sections, and again in Section 8, we focus attention on a particular sub-class of languages and domain descriptions, which we will call the class of *projection languages* and the class of *projection domain descriptions* respectively[7]. We have three reasons for doing so. In this section, we will show that it is possible to state syntactically verifiable conditions under which projection domain descriptions are consistent. In Section 6 we will show how we can use projection domain descriptions to formulate a notion of explanation complementary to that of Section 4. And in Section 8 we will show how for a particular class of domains we can build on this idea to develop meta-level Prolog implementations which facilitate a 'complete' form of automated reasoning both backwards and forwards in time.

The defining characteristic of a projection language is that the set of time points includes a null or least element, which is given a special status as regards formulation of projection domain descriptions.

Definition 5.1. [Projection Language] A *projection language* is a domain language $\langle \Pi, \preceq, \Delta, \Phi\rangle$, where Π includes an element T_0 (a null or least element) such that for all $T \in \Pi$, $T_0 \preceq T$.

In this section and the next, we assume that $\mathcal{E} = \langle \Pi, \preceq, \Delta, \Phi\rangle$ is a projection language, and that T_0 is the null element of Π. It will be convenient to identify a particular type of t-proposition which we will call an *i-proposition* ("i" for "initial").

Definition 5.2. [i-proposition] An *i-proposition* in $\mathcal{E}$ is a t-proposition of the form

$$L \textbf{ holds-at } T_0$$

where L is a fluent literal of $\mathcal{E}$. We shall sometimes write this expression as

$$\textbf{initially } L$$

[7] Note that the class of Situation-Calculus-style languages of the form $\langle \Pi_\Delta, \leq_\Delta, \Delta, \Phi\rangle$ defined in Proposition 3.1 of Section 3 are all examples of projection languages.

Definition 5.3. [Projection domain description] A *projection domain description* in $\mathcal{E}$ is a triple $\langle \gamma, \eta, \tau_i \rangle$, where γ is a set of c-propositions, η is a set of h-propositions, and τ_i is a set of i-propositions in $\mathcal{E}$.

Thus projection domain descriptions are domain descriptions that only allow t-propositions about the initial time point T_0.

Example 5.1. The following projection domain description uses the projection domain language $\mathcal{E}_{ys} = \langle \Re^+, \leq, \{Shoot\}, \{Alive, Loaded\} \rangle$ of Example 2.2, where $\Re^+$ signifies the non-negative real numbers (so that $T_0 = 0$).

Shoot **terminates** *Alive* **when** $\{Loaded\}$

Shoot **happens-at** 2

initially *Alive*

initially *Loaded*

At first sight it appears that the restriction of the set τ_i in the definition above to contain only i-propositions is a major limitation. However, in Section 6 we will describe a mode of reasoning involving both projection domain descriptions and extra sets of t-propositions which have been identified as 'observations' requiring explanation.

As stated above, one advantage of projection domains is that it is possible to characterise a whole class of such domains whose consistency can be easily verified. To state the appropriate proposition, we first need some extra definitions.

Definition 5.4. [Initial consistency] Let $D = \langle \gamma, \eta, \tau_i \rangle$ be a projection domain description. D is *initially-consistent* iff there is no fluent constant F such that both the i-proposition "**initially** F" and the i-proposition "**initially** $\neg F$" are in τ_i.

The next definition is related to the notion of "e-consistency" defined by Denecker and De Schreye [11] in the context of the Language $\mathcal{A}$.

Definition 5.5. [Conflicting actions] Let D be a domain description. The action constants A_1 and A_2 *conflict in* D iff D contains two c-propositions of the form "A_1 **initiates** F **when** C_1" and "A_2 **terminates** F **when** C_2" and there is no fluent symbol F' in $\mathcal{E}$ such that both $F' \in C_1 \cup C_2$ and $\neg F' \in C_1 \cup C_2$. When $A_1 = A_2 = A$ we say that the action constant A *self-conflicts* in D.

Definition 5.6. [Fluent independence] Let D be a projection domain description. D is *fluent-independent* iff (i) there is no time point t and pair of h-propositions in D of the form "A_1 **happens-at** t" and "A_2 **happens-at** t" such that A_1 and A_2 conflict in D, and (ii) there is no h-proposition in D of the form "A **happens-at** t" such that A self-conflicts in D.

Definition 5.7. [Non-convergence] Let D be a domain description written in a language $\mathcal{E} = \langle \Pi, \preceq, \Delta, \Phi \rangle$. D and $\mathcal{E}$ are *non-converging* iff for every three (not necessarily distinct) time points T_1, T_2 and T_3 in Π such that $T_1 \preceq T_3$ and $T_2 \preceq T_3$, then either $T_1 \preceq T_2$ or $T_2 \preceq T_1$.

Proposition 5.1. Let D be a projection domain description which is occurrence-sparse, non-converging, initially-consistent, and fluent-independent. Then D is consistent.

PROOF. See Appendix B.2.

6. OBSERVATIONS AND EXPLANATIONS

In Section 4 we linked the notion of an explanation to the idea of transforming an inconsistent domain description into a consistent one. Another aspect of explanation that we might want to capture in the context of temporal domains is to explain what holds at a later time in terms of what holds at an earlier time. For example, we might wish to consider the statement "Fred is not alive" as explained by the statement "the gun was loaded when he was shot". This is an explanation of what holds at a later time in terms of what holds at an earlier time.

In this section we consider only a special case of this type of explanation, which arises specifically in the case of projection domain descriptions. We will regard information about what holds at the least time point T_0 of a projection domain description as a potential explanation for observations about what holds at all later times. In doing so, we are implicitly according i-propositions a special status among t-propositions – rather than recording 'observations', i-propositions are here regarded as statements about what was true 'in the beginning' or 'originally' (implicitly, before anyone was around to start recording observations or performing actions). Other t-propositions are re-introduced not simply as additional statements of the domain description, but as observations that need to be explained (in terms of what was originally true and/or in terms of events that have occurred).

Definition 6.1. [o-proposition] An *o-proposition* in $\mathcal{E}$ is a t-proposition of the form

$$L \textbf{ holds-at } T$$

where $T \neq T_0$.

Definition 6.2. [Observation set] An *observation set* is a non-empty set of o-propositions.

Definition 6.3. [i-explanation] Let $D = \langle \gamma, \eta, \tau_i \rangle$ be a projection domain description and let τ_{ob} be an observation set. An *i-explanation* for τ_{ob} in D is a set $\tau_{i\epsilon}$ of i-propositions such that $\langle \gamma, \eta, \tau_i \cup \tau_{i\epsilon} \rangle$ is consistent and such that for each $p \in \tau_{ob}$,

$$\langle \gamma, \eta, \tau_i \cup \tau_{i\epsilon} \rangle \models p$$

For simplicity, and in contrast to the discussion of Section 4, we do not include here any definition of preference between i-explanations, or any definition of optimality of an i-explanation. Our motivation for defining 'i-entailment' below is simply to maintain the distinction between observations and other aspects of the domain theory while allowing conclusions arising from observations to be properly characterised. As with h-entailment, we are concerned with capturing a 'safe' form of inference – in terms of abduction, we want to i-entailment to correspond to 'entailment in all abductive extensions'.

It is important to point out, however, that for specific domains we might well wish to introduce a definition of preference among i-explanations, and build this into the definition of i-entailment in exactly the same way as h-preference is incorporated in the definition of h-entailment. For example, given the observation that "the car didn't start after the ignition was turned", we might prefer the explanations that "the battery was dead" or that "the car had no petrol" to the explanation that "the car had no engine".

Definition 6.4. [i-model] Let $D = \langle \gamma, \eta, \tau_i \rangle$ be a projection domain description and let τ_{ob} be an observation set. H is an *i-model of D with τ_{ob}* iff there exists an i-explanation $\tau_{i\epsilon}$ for τ_{ob} in D such that H is a model of $\langle \gamma, \eta, \tau_i \cup \tau_{i\epsilon} \rangle$.

Definition 6.5. [i-consistency] Let D be a projection domain description and let τ_{ob} be an observation set. D is *i-consistent with τ_{ob}* iff there is at least one i-model of D with τ_{ob}.

Definition 6.6. [i-entailment] Let D be a projection domain description and let τ_{ob} be an observation set. D with τ_{ob} *i-entails* the t-proposition "F **holds-at** T", written "$D, \tau_{ob} \models_i F$ **holds-at** T", iff for every i-model H of D with τ_{ob}, $H(F,T) = true$. D with τ_{ob} i-entails the t-proposition "$\neg F$ **holds-at** T" iff for every i-model H of D with τ_{ob}, $H(F,T) = false$.

The following proposition shows that the particular definition of i-entailment above amounts to a re-characterisation of entailment as defined in Section 2, keeping the distinction between observation sets and projection domain descriptions. Note, however, that the proposition would not necessarily hold if we were to incorporate a (domain-specific) notion of i-preference in the definitions above.

Proposition 6.1. Let $D = \langle \gamma, \eta, \tau_i \rangle$ be an occurrence-sparse projection domain description and let τ_{ob} be an observation set. Then H is a model of $\langle \gamma, \eta, \tau_i \cup \tau_{ob} \rangle$ if and only if H is an i-model of D with τ_{ob}.

PROOF. See Appendix B.3

Example 6.1. Let $\mathcal{E}_{ysp} = \langle \Re^+, \leq, \{Shoot\}, \{Alive, Loaded\} \rangle$, where $\Re^+$ signifies the non-negative real numbers, and let the projection description D_{ysp} consist of a single c-proposition and a single h-proposition:

Shoot **terminates** *Alive* **when** $\{Loaded\}$

Shoot **happens-at** 2

Let τ_{ysp} be the observation set containing the following two o-propositions:

Alive **holds-at** 1

$\neg$*Alive* **holds-at** 3

It is easy to see that there is a unique i-explanation for τ_{ysp} in D_{ysp}:

$$\{\ \textbf{initially}\ \textit{Loaded},\ \ \textbf{initially}\ \textit{Alive}\ \}$$

Hence for all $n \geq 0$

$$D_{ysp}, \tau_{ysp} \models_i \textit{Loaded}\ \textbf{holds-at}\ n$$

The notions of i-explanation and h-explanation described in this section and in Section 4 are complementary and can be combined in an obvious way.

Definition 6.7. [ih-explanation] Let $D = \langle \gamma, \eta, \tau_i \rangle$ be a projection domain description and let τ_{ob} be an observation set. An *ih-explanation* for τ_{ob} in D is a pair $\langle \eta_\epsilon, \tau_{i\epsilon} \rangle$, where η_ϵ is an occurrence-sparse set of h-propositions and $\tau_{i\epsilon}$ is a set of i-propositions, such that $\langle \gamma, \eta \cup \eta_\epsilon, \tau_i \cup \tau_{i\epsilon} \rangle$ is consistent and such that for each $p \in \tau_{ob}$,

$$\langle \gamma, \eta \cup \eta_\epsilon, \tau_i \cup \tau_{i\epsilon} \rangle \models p$$

However, it is less obvious when one ih-explanation should be considered preferable to another. To conclude this section, we compare two preference relations for ih-explanations, both of which might be reasonable for particular classes of domains. We illustrate their different effects with an example. The following discussion lends further weight to the viewpoint that, although the object level semantics of Section 2 is domain-independent, decisions as to which explanations for inconsistent domains or for sets of observations should be preferred will, in general, be based partly on consideration of the specific domain.

The first preference relation we consider is as follows. It compares only the h-propositions in two given ih-explanations.

Definition 6.8. [Preferable ih-explanation]
Let $\epsilon = \langle \eta_\epsilon, \tau_{i\epsilon} \rangle$ and $\epsilon' = \langle \eta'_\epsilon, \tau'_{i\epsilon} \rangle$ be ih-explanations for τ_{ob} in D. ϵ is *ih-preferable* to ϵ' iff $\eta_\epsilon \subset \eta'_\epsilon$.

Definition 6.9. [Optimal ih-explanation] ϵ is an *optimal ih-explanation* for τ_{ob} in D iff ϵ is an ih-explanation for τ_{ob} in D and there is no other ih-explanation ϵ' for τ_{ob} in D such that ϵ' is ih-preferable to ϵ.

Definition 6.10. [ih-model] Let $D = \langle \gamma, \eta, \tau_i \rangle$ be a projection domain description and let τ_{ob} be an observation set. H is an *ih-model of D with τ_{ob}* iff there exists an optimal ih-explanation $\epsilon = \langle \eta_\epsilon, \tau_{i\epsilon} \rangle$ for τ_{ob} in D such that H is a model of $\langle \gamma, \eta \cup \eta_\epsilon, \tau_i \cup \tau_{i\epsilon} \rangle$

Definition 6.11. [ih-entailment] Let D be a projection domain description and let τ_{ob} be an observation set. D with τ_{ob} *ih-entails* the t-proposition "F **holds-at** T", written "$D, \tau_{ob} \models_{ih} F$ **holds-at** T", iff for every ih-model H of D with τ_{ob}, $H(F,T) = \textit{true}$. D with τ_{ob} ih-entails the t-proposition "$\neg F$ **holds-at** T" iff for every ih-model H of D with τ_{ob}, $H(F,T) = \textit{false}$.

Example 6.2. This example involves a video-recorder with an automatic timer. Suppose Fred returns home one evening and wishes to check if his video-recorder has automatically started recording his favourite TV show. Although he cannot look inside the recording machine to check if it is recording directly, he knows that if the machine is working, when the timer triggers it will both turn the 'record' light on and begin recording. Fred also knows that if the machine is not working it can be repaired. When Fred left home, the record light was not on. When he returns, the light is on, and Fred concludes that the machine is recording.

We can represent Fred's domain knowledge with a projection domain description D_{rec} consisting of three c-propositions:

$$TimerTrigger \textbf{ initiates } LightOn \textbf{ when } \{Working\}$$

$$TimerTrigger \textbf{ initiates } Recording \textbf{ when } \{Working\}$$

$$Repair \textbf{ initiates } Working$$

Fred's observations, that the recording light was of when he left home but is on when he returns, are represented by the observation set τ_{rec}:

$$\{\neg LightOn \textbf{ holds-at } T_L,\ LightOn \textbf{ holds-at } T_R\}$$

where $T_L, T_R \in \Pi$ represent the times at which Fred left and returned home respectively. It is easy to see that all optimal ih-explanations for τ_{rec} in D_{rec} are of the form

$$\begin{gathered}\langle\{TimerTrigger \textbf{ happens-at } T\},\\ \{\textbf{initially } Working,\ \textbf{initially } \neg LightOn\}\rangle\end{gathered}$$

for some $T \in \Pi$ such that $T_L \preceq T \prec T_R$, so that

$$D_{rec}, \tau_{rec} \models_{ih} Recording \textbf{ holds-at } T_R$$

and hence the notion of ih-entailment correctly models Fred's reasoning in this respect.

In the example above, Fred can also use ih-entailment to reason that his video-recorder was functioning properly at time T_L (and indeed at any time point whatsoever), i.e.

$$D_{rec}, \tau_{rec} \models_{ih} Working \textbf{ holds-at } T_L$$

In doing so, Fred rejects ih-explanations in which an extra *Repair* action occurs at some time before the *TimerTrigger* action. It is a matter of debate as to whether, for this domain, it is reasonable to do this. But it is not hard to formulate an alternative preference criterion among ih-explanations which does not cause such alternatives to be rejected:

Definition 6.12. [$\overline{\text{Preferable}}$ ih-explanation]
Let $\epsilon = \langle \eta_\epsilon, \tau_{i\epsilon} \rangle$ and $\epsilon' = \langle \eta'_\epsilon, \tau'_{i\epsilon} \rangle$ be ih-explanations for τ_{ob} in D. ϵ is $\overline{ih}$*-preferable* to ϵ' iff $\eta_\epsilon \subset \eta'_\epsilon$ and $\tau_{i\epsilon} = \tau'_{i\epsilon}$.

In other words, explanations are only "$\overline{\text{ih}}$-comparable" if they contain the same i-propositions. Assuming definitions exactly analogous to Definitions 6.9, 6.10 and 6.11 for an optimal $\overline{\text{ih}}$-explanation, an $\overline{\text{ih}}$-model, and $\overline{\text{ih}}$-entailment respectively, it is easy to see that all optimal ih-explanations are also optimal $\overline{\text{ih}}$-explanations, but that (for the example above) optimal $\overline{\text{ih}}$-explanations can also be of the form

$$(\{\mathit{Repair}\ \textbf{happens-at}\ T',\ \mathit{TimerTrigger}\ \textbf{happens-at}\ T\},\\ \{\textbf{initially}\ \neg \mathit{Working},\ \textbf{initially}\ \neg \mathit{LightOn}\})$$

for some $T', T \in \Pi$ such that $T_L \preceq T \prec T_R$ and $T' \prec T$. Hence, although (as for ih-entailment)

$$D_{rec}, \tau_{rec} \models_{\overline{ih}} \mathit{Recording}\ \textbf{holds-at}\ T_R$$

Fred cannot use $\overline{\text{ih}}$-entailment to conclude that his video-recorder was working when he left the house.

It is not hard to think up examples either where ih-entailment or where $\overline{\text{ih}}$-entailment seems more plausible. The choice seems to depend on whether (for the particular domain) one is more concerned with assuming as few action occurrences as possible (ih-entailment), or whether one is more concerned with giving equal consideration to all possible initial conditions, even in the face of observations at later times ($\overline{\text{ih}}$-entailment).

7. LOGIC PROGRAMS FOR $\mathcal{E}$ DOMAINS

In the following two sections we discuss the implementation of Language $\mathcal{E}$ domains. In this section we study how we can construct Event Calculus style logic programs from domain descriptions in general. In Section 8 we show how, for a class of projection domain descriptions, (simplified versions of) these programs can be enhanced using standard Prolog 'second-order' programming techniques.

In the original Event Calculus there was an implicit assumption that all predicate definitions were complete. In other words, it was assumed that for each predicate its negation (negation as failure) was true whenever the positive instance did not hold. Although the semantics of $\mathcal{E}$ incorporates an analogous assumption for h-propositions and c-propositions, this assumption does not extend to t-propositions (equivalent to *Holds* or *HoldsAt* literals in Event Calculus programs) – it is possible for a domain description D to be 'incomplete' in the sense that, for some fluent constant F and time T, neither "F **holds-at** T" nor "$\neg F$ **holds-at** T" is entailed by D.

However, we can partly avoid the implementation difficulties that this creates by representing negative fluent literals *inside* the *HoldsAt* predicate. In the program translations defined below, the t-proposition "$\neg F$ **holds-at** T" is represented by the positive literal $\mathit{HoldsAt}(\mathit{Neg}(F), T)$, whereas the negative literal *not* $\mathit{HoldsAt}(F, T)$ is simply interpreted as 'the t-proposition "F **holds-at** T" is not provable'. In this and other respects, the translation method here is similar to that in [29]. Analogous techniques are used in [15], [13] and [4], although not with Event Calculus style programs.

Given that our aim is to develop programs able to deal correctly with the form of incompleteness described above, it is useful to first consider incomplete or partial interpretations for a domain description and examine what can be computed from these.

Definition 7.1. [Partial interpretation] A *partial interpretation* of $\mathcal{E}$ is a partial mapping

$$I : \Phi \times \Pi \mapsto \{true, false\}$$

In the discussion which follows, we assume that Definitions 2.8 and 2.9 of 'point satisfaction' and an 'initiation' or 'termination point' are extended to cover partial interpretations as well as interpretations. In addition, we need counterparts to these notions which deal with cases where $I(F, T)$ is undefined.

Definition 7.2. [Possible point satisfaction] Given a set of fluent literals C of $\mathcal{E}$ and a time point $T \in \Pi$, a partial interpretation I *possibly satisfies* C *at* T iff for each fluent constant $F \in C$, $I(F, T) \neq false$, and for each fluent constant F' such that $\neg F' \in C$, $I(F', T) \neq true$.

Definition 7.3. [Possible initiation/termination point] Let I be a partial interpretation of $\mathcal{E}$, let D be a domain description, let $F \in \Phi$ and let $T \in \Pi$. T is a *possible initiation-point* (respectively *possible termination-point*) *for* F *in* I *relative to* D iff there is an $A \in \Delta$ such that (i) there is both an h-proposition in η of the form "A **happens-at** T" and a c-proposition in γ of the form "A **initiates** F **when** C" (respectively "A **terminates** F **when** C") and (ii) I possibly satisfies C at T.

Let us denote the set of all partial interpretations by $\mathcal{I}$. Motivated by Definition 2.10 of a model for a domain D, we can define an associated (partial) operator on $\mathcal{I}$ as follows.

Definition 7.4. [Operator $\mathcal{F}$] Given a domain description $D = \langle \gamma, \eta, \tau \rangle$ the partial operator $\mathcal{F} : \mathcal{I} \mapsto \mathcal{I}$ is defined as follows: For any partial interpretation $I \in \mathcal{I}$, and any $F \in \Phi$, $T \in \Pi$,

1. (a) For any $T_1 \in \Pi$ such that $T_1 \prec T$, if there is no possible initiation-point or possible termination-point T_2 for F in I relative to D such that $T_1 \preceq T_2 \prec T$, then $(\mathcal{F})(I)(F, T) = I(F, T_1)$.

   (b) For any $T_2 \in \Pi$ such that $T \prec T_2$, if there is no possible initiation-point or possible termination-point T_1 for F in I relative to D such that $T \preceq T_1 \prec T_2$, then $(\mathcal{F})(I)(F, T) = I(F, T_2)$.

2. If T_1 is an initiation-point for F in I relative to D, $T_1 \prec T$ and there is no possible termination-point T_2 for F in I relative to D such that $T_1 \prec T_2 \prec T$, then $(\mathcal{F})(I)(F, T) = true$.

3. If T_1 is a termination-point for F in I relative to D, $T_1 \prec T$ and there is no possible initiation-point T_2 for F in I relative to D such that $T_1 \prec T_2 \prec T$, then $(\mathcal{F})(I)(F, T) = false$.

4. If there is a t-proposition in τ of the form "F **holds-at** T", then

$$(\mathcal{F})(I)(F, T) = true,$$

and if there is a t-proposition of the form "$\neg F$ **holds-at** T",

$$(\mathcal{F})(I)(F,T) = false.$$

5. Otherwise $(\mathcal{F})(I)(F,T)$ is undefined.

Note that this operator is not always defined, as it is possible for these rules to require the assignment of both true and false to $(\mathcal{F})(I)(F,T)$ for some F and T.

It is easy to see that conditions (1) to (4) in the definition above correspond closely to conditions (1) to (4) of Definition 2.10 of a model. The following definition and three propositions show the relationship between Language $\mathcal{E}$ models and the operator $\mathcal{F}$.

Definition 7.5. [The ordering $\subseteq$] For any two partial interpretations $I_1, I_2 \in \mathcal{I}$, I_1 *is contained in* I_2, written $I_1 \subseteq I_2$, iff (i) for any F and T if $I_1(F,T) = true$ then $I_2(F,T) = true$, and (ii) if $I_1(F,T) = false$ then $I_2(F,T) = false$.

Proposition 7.1. Let D be a domain description. If D is consistent then any model H of D is a greatest (with respect to the ordering $\subseteq$) fixed point of $\mathcal{F}$.

PROOF. The proof follows directly from the construction of $\mathcal{F}$ and the observation that since H is a total mapping the definition for possible initiation-point (respectively possible termination-point) coincides with that of initiation-point (respectively termination-point).

When a domain D is consistent we can apply the operator $\mathcal{F}$ iteratively to compute a partial interpretation that would be a subset of any model of D.

Proposition 7.2. Let D be a consistent domain description, let H be a model of D and let I be a partial interpretation such that $I \subseteq H$. Then $\mathcal{F}(I) \subseteq H$.

PROOF. The proof follows by comparing the different cases under which $\mathcal{F}$ applies with conditions (1)–(4) in Definition 2.10 of a model, and noticing that the following two properties hold for every $T \in \Pi$ and $F \in \Phi$: (i) if T is an initiation-point (resp. termination-point) for F in I then T is an initiation-point (resp. termination-point) for F in H and (ii) if T is not a possible initiation-point (resp. possible termination-point) for F in I then T is not an initiation-point (resp. termination-point) for F in H.

Proposition 7.3. Let D be a consistent domain description and let the sequence $I_0, \ldots, I_n, \ldots$ of partial interpretations be defined as follows:

- $I_0 = \emptyset$
- $I_{n+1} = I_n \cup \mathcal{F}(I_n)$ *for each countable ordinal* $n > 0$.

Then there exists a least fixed point I^+ of this sequence, and D entails any t-proposition of the form "F **holds-at** *T" (resp. "$\neg F$* **holds-at** *T") such that $I^+(F,T) = true$ (resp. $I^+(F,T) = false$).*

PROOF. By Proposition 7.2 when D is consistent this sequence is well defined. Also, by construction the sequence is monotonic (with respect to the ordering $\subseteq$

of Definition 7.5) and hence the least fixed point I^+ exists. Again by Proposition 7.2 $I^+ \subseteq H$ for any model H of D and hence the result follows.

Given Proposition 7.3, we can use Definition 7.4 to 'read off' a logic program that implements $\mathcal{F}$ and computes consequences belonging to I^+. To simplify our definitions we assume that some logic program or external definition of the order relation $\preceq$ is available.

Definition 7.6. [Ordering program] Given the language $\mathcal{E} = \langle \Pi, \preceq, \Delta, \Phi \rangle$, the program$P(\Pi, \preceq)$ is an *ordering program* for $\mathcal{E}$ iff

- for all $T, T' \in \Pi$, $P(\Pi, \preceq)$ succeeds on the query $T \preceq T'$ if and only if $T \preceq T'$, and finitely fails otherwise.
- for all $T, T' \in \Pi$, $P(\Pi, \preceq)$ succeeds on the query $T \prec T'$ if and only if $T \prec T'$, and finitely fails otherwise.
- None of the following predicate symbols appear in $P(\Pi, \preceq)$:
HoldsAt, *Given*, *ClippedBetween*, *HappensAt*, *PossiblyInitiates*, *Initiates*, *PossiblyTerminates*, *Terminates*, *AffectedBetween*.

We will also need the following preliminary definitions.

Definition 7.7. [lp-term and lp-complement] Given a fluent literal L of $\mathcal{E} = \langle \Pi, \preceq, \Delta, \Phi \rangle$, the *lp-term* of L, written $\lambda(L)$, is defined to be

- *Pos*(F) if $L = F$ for some $F \in \Phi$
- *Neg*(F) if $L = \neg F$ for some $F \in \Phi$

and the *lp-complement* of L, written $\overline{\lambda(L)}$, is defined to be

- *Neg*(F) if $L = F$ for some $F \in \Phi$
- *Pos*(F) if $L = \neg F$ for some $F \in \Phi$

Definition 7.8. [Finite domain description] The domain description $\langle \gamma, \eta, \tau \rangle$ is *finite* iff γ, η and τ are all finite, and for each c-proposition in γ either of the form "A **initiates** F **when** C" or of the form "A **terminates** F **when** C", C is also finite.

Our translation from domain descriptions to logic programs can now be given. In the definition below, the five clauses defining *HoldsAt* correspond to rules (1a)–(4) in Definition 7.4 of the operator $\mathcal{F}$ (Clauses (LP1a) and (LP1b) are 'special cases' of rules (1a) and (1b)). The use of negation-as-failure and fluent converses in the domain-specific clauses defining the predicates *PossiblyInitiates* and *PossiblyTerminates* reflects Definition 7.3 of a possible initiation and possible termination point.

Definition 7.9. [$LP[D, P(\Pi, \preceq)]$] Given a finite domain description $D = \langle \gamma, \eta, \tau \rangle$ written in the language $\mathcal{E} = \langle \Pi, \preceq, \Delta, \Phi \rangle$, and an ordering program $P(\Pi, \preceq)$, the logic program $LP[D, P(\Pi, \preceq)]$ is defined as the program $P(\Pi, \preceq)$ augmented with the following general clauses

$$\begin{aligned}&\mathit{HoldsAt}(l,t_3) \leftarrow\\ &\quad \mathit{Given}(l,t_1),\ t_1 \prec t_3,\ \mathit{not}\ \mathit{AffectedBetween}(t_1,l,t_3).\end{aligned} \tag{LP1a}$$

$$\begin{aligned}&\mathit{HoldsAt}(l,t_1) \leftarrow\\ &\quad \mathit{Given}(l,t_3),\ t_1 \prec t_3,\ \mathit{not}\ \mathit{AffectedBetween}(t_1,l,t_3).\end{aligned} \tag{LP1b}$$

$$\begin{aligned}&\mathit{HoldsAt}(\mathit{Pos}(f),t_3) \leftarrow\\ &\quad \mathit{HappensAt}(a,t_1),\ t_1 \prec t_3,\ \mathit{Initiates}(a,f,t_1),\\ &\quad \mathit{not}\ \mathit{ClippedBetween}(t_1,\mathit{Pos}(f),t_3).\end{aligned} \tag{LP2}$$

$$\begin{aligned}&\mathit{HoldsAt}(\mathit{Neg}(f),t_3) \leftarrow\\ &\quad \mathit{HappensAt}(a,t_1),\ t_1 \prec t_3,\ \mathit{Terminates}(a,f,t_1),\\ &\quad \mathit{not}\ \mathit{ClippedBetween}(t_1,\mathit{Neg}(f),t_3).\end{aligned} \tag{LP3}$$

$$\mathit{HoldsAt}(l,t) \leftarrow \mathit{Given}(l,t). \tag{LP4}$$

$$\begin{aligned}&\mathit{ClippedBetween}(t_1,\mathit{Pos}(f),t_3) \leftarrow\\ &\quad \mathit{HappensAt}(a,t_2),\ t_1 \preceq t_2,\ t_2 \prec t_3,\\ &\quad \mathit{PossiblyTerminates}(a,f,t_2).\end{aligned} \tag{LP5}$$

$$\begin{aligned}&\mathit{ClippedBetween}(t_1,\mathit{Neg}(f),t_3) \leftarrow\\ &\quad \mathit{HappensAt}(a,t_2),\ t_1 \preceq t_2,\ t_2 \prec t_3,\\ &\quad \mathit{PossiblyInitiates}(a,f,t_2).\end{aligned} \tag{LP6}$$

$$\mathit{AffectedBetween}(t_1,l,t_3) \ \leftarrow\ \mathit{ClippedBetween}(t_1,l,t_3). \tag{LP7}$$

$$\begin{aligned}&\mathit{AffectedBetween}(t_1,\mathit{Neg}(f),t_3) \ \leftarrow\\ &\quad \mathit{ClippedBetween}(t_1,\mathit{Pos}(f),t_3).\end{aligned} \tag{LP8}$$

$$\begin{aligned}&\mathit{AffectedBetween}(t_1,\mathit{Pos}(f),t_3) \ \leftarrow\\ &\quad \mathit{ClippedBetween}(t_1,\mathit{Neg}(f),t_3).\end{aligned} \tag{LP9}$$

and the following domain-specific clauses

- For each t-proposition "L **holds-at** T" in τ, the clause

  $\mathit{Given}(\lambda(L),T).$

- For each h-proposition "A **happens-at** T" in η, the clause

  $\mathit{HappensAt}(A,T).$

- For each c-proposition "A **initiates** F **when** $\{L_1,\ldots,L_n\}$" in γ, the clause

$$\begin{array}{l} \mathit{Initiates}(A, F, t) \leftarrow \\ \qquad \mathit{HoldsAt}(\lambda(L_1), t), \ldots, \mathit{HoldsAt}(\lambda(L_n), t). \end{array}$$

and the clause

$$\begin{array}{l} \mathit{PossiblyInitiates}(A, F, t) \leftarrow \\ \qquad \mathit{not}\ \mathit{HoldsAt}(\overline{\lambda(L_1)}, t), \ldots, \mathit{not}\ \mathit{HoldsAt}(\overline{\lambda(L_n)}, t). \end{array}$$

- For each c-proposition "A **terminates** F **when** $\{L_1, \ldots, L_n\}$" in γ, the clause

  $$\begin{array}{l} \mathit{Terminates}(A, F, t) \leftarrow \\ \qquad \mathit{HoldsAt}(\lambda(L_1), t), \ldots, \mathit{HoldsAt}(\lambda(L_n), t). \end{array}$$

  and the clause

  $$\begin{array}{l} \mathit{PossiblyTerminates}(A, F, t) \leftarrow \\ \qquad \mathit{not}\ \mathit{HoldsAt}(\overline{\lambda(L_1)}, t), \ldots, \mathit{not}\ \mathit{HoldsAt}(\overline{\lambda(L_n)}, t). \end{array}$$

Intuitively, given Propositions 7.1, 7.2 and 7.3 it is easy to see that the programs described above behave correctly for consistent domain descriptions, since Clauses (LP1a)–(LP4) either correspond exactly to or are just special cases of conditions (1a)–(4) in Definition 7.4. The following proposition[8]confirms this intuition.

Proposition 7.4. Let $P(\Pi, \preceq)$ be an ordering program for $\mathcal{E}$, and let D be a finite domain description. Then for any fluent literal L of $\mathcal{E}$ and any $T \in \Pi$, if

$$LP[D, P(\Pi, \preceq)] \vdash_{SLDNF} \mathit{HoldsAt}(\lambda(L), T)$$

then

$$D \models L\ \textbf{holds-at}\ T$$

PROOF. See Appendix B.4

To a certain extent, the above logic programs overcome the two limitations of formalizations of action in normal logic programming identified by Gelfond and Lifschitz in [15]. That is to say, (i) if the values of some fluents at one or more time points are given, they facilitate automated reasoning about what holds at other time points before, afterwards or in between, and (ii) as shown by Proposition 7.4, the programs behave correctly even when the information entailed by their Language $\mathcal{E}$ specifications is incomplete.

However, although they are sound, the above logic programs do not compute all consequences of every domain under its semantics as given by Definition 2.10. This

[8] In this and subsequent propositions, the symbol $\vdash_{SLDNF}$ signifies the existence of an SLDNF derivation, defined as for example in [27]. However, for clarity of presentation, in proposition proofs we avoid working directly with this type of definition, and instead refer to SLDNF derivations using the usual (Prolog) programming terminology.

potential incompleteness is illustrated by the following example.

Example 7.1. Let $\mathcal{E}_v$ and D_v be the domain language and domain description respectively of Example 2.1, and suppose that $P(\Re, \leq)$ is an ordering program for $\mathcal{E}_v$. Then the logic program $LP[D_v, P(\Re, \leq)]$ consists of the program $P(\Re, \leq)$ together with clauses (LP1a)–(LP9) of Definition 7.9 and the domain-specific clauses

$$\begin{aligned}&Initiates(InjectA, Protected, t) \leftarrow\\&\quad HoldsAt(Pos(TypeO), t).\end{aligned}$$

$$\begin{aligned}&PossiblyInitiates(InjectA, Protected, t) \leftarrow\\&\quad not\ HoldsAt(Neg(TypeO), t).\end{aligned}$$

$$\begin{aligned}&Initiates(InjectB, Protected, t) \leftarrow\\&\quad HoldsAt(Neg(TypeO), t).\end{aligned}$$

$$\begin{aligned}&PossiblyInitiates(InjectB, Protected, t) \leftarrow\\&\quad not\ HoldsAt(Pos(TypeO), t).\end{aligned}$$

$$HappensAt(InjectA, 2).$$

$$HappensAt(InjectB, 3).$$

$$Given(Neg(Protected), 1).$$

As it stands, the query *HoldsAt*(*Pos*(*Protected*), 4) will fail on this program even though the corresponding t-proposition is entailed by its specification. Notice however that the query can be made to succeed by adding either *Given*(*Pos*(*TypeO*), n) or *Given*(*Neg*(*TypeO*), n) to the program for some time point $n < 2$. The example also illustrates the necessity of using the *PossiblyInitiates* and *PossiblyTerminates* predicates. Had the program definition of *ClippedBetween* been given simply in terms of *Initiates* and *Terminates*, the query *HoldsAt*(*Neg*(*Protected*), 4) would succeed, even though the t-proposition "$\neg$*Protected* **holds-at** 4" is not entailed by D_v. Finally, notice that the success of the goal *HoldsAt*(*Neg*(*Protected*), 0) (trivially) demonstrates the utility of this type of program for reasoning backwards in time.

8. META-LEVEL PROGRAMS FOR COMPUTING I-ENTAILMENT

In this section we show how, for a class of domains, we can exploit the meta-level characterisation of i-entailment given in Section 6 to build meta-level programs which facilitate a more 'complete' form of reasoning (both backwards and forwards in time) than the object-level programs of the previous section. The important characteristic of the programs given below is that they maintain the distinction between observations, which are dealt with at the meta-level, and other parts of the domain theory.

The object-level programs upon which our meta-level implementation is built are simplified versions of the programs described in Section 7. The simplification is possible because of the following property of projection domain descriptions. If D is a projection domain description which contains either the i-proposition "**initially** F" or the i-proposition "**initially** $\neg F$" for every fluent constant F, then D is 'complete' in the sense that it has at most one model (this follows from Proposition 2.1). If $\mathcal{E}$ is non-converging, and D is fluent-independent, D has exactly one model (this follows from Proposition 5.1). Hence, in the case where D is also finite (so that there are only a finite number of fluent constants in $\mathcal{E}$), it is not hard to construct a simplified Event Calculus style program for D enabling complete automated reasoning forwards in time from the initial time point T_0. For reasons which will shortly become apparent, we will represent the ('complete' set of) i-propositions of D in list form inside a three-argument version of the *HoldsAt* predicate, rather than with a *Given* predicate as previously. In Definition 8.2 below, $HoldsAt(M, L, T)$ should be read as "$D \models L$ **holds-at** T, where M is (a list representation of) the set of i-propositions in D".

It is easy to see that clauses (EC7)–(EC11) which define *HoldsAt* are a simplification of clauses (LP1a)–(LP9) in Definition 7.9. Here it is not necessary to distinguish between the predicates *Initiates* (resp. *Terminates*) and *PossiblyInitiates* (resp. *PossiblyTerminates*), and clauses (LP1a), (LP1b) and (LP4) can be condensed into the single clause (EC7). This is because, at the (object) level of calls to the *HoldsAt* predicate, complete information is available about what holds at the initial time point T_0, and we are only interested in reasoning forwards in time from this point. Incompleteness and reasoning backwards in time are instead dealt with at the meta-level.

We wish to construct a program able to test whether a particular t-proposition is i-entailed by some projection domain description D' (which, unlike D above, may not have an i-proposition for each fluent) together with some observation set τ_{ob}. All that remains to be done is to define a meta-level program able to use the *HoldsAt* predicate to test the truth of the t-proposition in each extension of D' with a 'maximal' i-explanation for τ_{ob}. (It is sufficient to consider only maximal i-explanations, i.e. those which mention every fluent in the language, because of the monotonicity of $\mathcal{E}$ as regards addition of t-propositions to any domain description – see the remarks at the end of Section 2.) This is achieved in a straightforward way by clauses (EC1)–(EC6) in Definition 8.2 below. In this definition, $IHoldsAt(L, T)$ should be read as "$D', \tau_{ob} \models_i L$ **holds-at** T". $IExplanation(M)$ should be read as "M is a (maximal) i-explanation for τ_{ob} in D'".

In Definition 8.2, a Prolog-like syntax for lists is used. Thus the term [] represents the empty list and the term $[Head|Remainder]$ represents a non-empty list whose first element is $Head$. Suitable definitions of the standard list predicates $Member$ and $Append$ are assumed. The two meta-level (or 'second-order') predicates $Setof$ and $Forall$ are also used. To summarise their functions, $Forall(Condition, Goal)$ succeeds if for all solutions of $Condition$, $Goal$ succeeds. $Setof(X, Goal, Instances)$ succeeds if $Instances$ is the set of instances of X for which $Goal$ succeeds, where sets are represented as (possibly empty) lists without repetitions. For practical details of the use of these predicates in the context of Prolog programming, the reader may consult [38]. (It is also assumed that the predicates $IHoldsAt$, $IExplanation$, $Permutation$, and $ConsistentWithObservations$ do not appear in the ordering program $P(\Pi, \preceq)$.)

Definition 8.1. The domain language $\mathcal{E} = \langle \Pi, \preceq, \Delta, \Phi \rangle$ is *fluent-finite* iff the set Φ is finite.

Definition 8.2. $[EC[D, \tau_{ob}, P(\Pi, \preceq)]]$
Let $D = \langle \gamma, \eta, \tau \rangle$ be a finite projection domain description written in a fluent-finite projection language $\mathcal{E} = \langle \Pi, \preceq, \Delta, \Phi \rangle$. Let T_0 be the null element of Π, and let $P(\Pi, \preceq)$ be an ordering program for $\mathcal{E}$. Let τ_{ob} be a finite observation set. The logic program $EC[D, \tau_{ob}, P(\Pi, \preceq)]$ is defined as the program $P(\Pi, \preceq)$ augmented with the following general clauses

$$\begin{aligned}&IHoldsAt(l,t) \leftarrow \\ &\quad Forall(IExplanation(m), HoldsAt(m,l,t)).\end{aligned} \tag{EC1}$$

$$\begin{aligned}&IExplanation(m) \leftarrow \\ &\quad Setof(l, (Initially(l)), i), \\ &\quad Setof(f, \\ &\qquad\quad (Fluent(f),\ not\ Initially(Pos(f)),\ not\ Initially(Neg(f))), \\ &\qquad\quad p), \\ &\quad Permutation(p,c),\ Append(c,i,m), \\ &\quad ConsistentWithObservations(m).\end{aligned} \tag{EC2}$$

$$Permutation([], []). \tag{EC3}$$

$$Permutation([f|r1], [f|r2]) \leftarrow Permutation(r1, r2). \tag{EC4}$$

$$Permutation([f|r1], [Neg(f)|r2]) \leftarrow Permutation(r1, r2). \tag{EC5}$$

$$\begin{aligned}&ConsistentWithObservations(m) \leftarrow \\ &\quad Forall(Observation(l,t), HoldsAt(m,l,t)).\end{aligned} \tag{EC6}$$

$$\begin{aligned}&HoldsAt(m,l,t_3) \leftarrow \\ &\quad Member(l,m),\ not\ ClippedBetween(m,T_0,l,t_3).\end{aligned} \tag{EC7}$$

$$\begin{aligned}&HoldsAt(m,Pos(f),t_3) \leftarrow \\ &\quad HappensAt(a,t_1),\ t_1 \prec t_3,\ Initiates(m,a,f,t_1), \\ &\quad not\ ClippedBetween(m,t_1,Pos(f),t_3).\end{aligned} \tag{EC8}$$

$$\begin{aligned}&HoldsAt(m,Neg(f),t_3) \leftarrow \\ &\quad HappensAt(a,t_1),\ t_1 \prec t_3,\ Terminates(m,a,f,t_1), \\ &\quad not\ ClippedBetween(m,t_1,Neg(f),t_3).\end{aligned} \tag{EC9}$$

$$\begin{aligned}&ClippedBetween(m,t_1,Pos(f),t_3) \leftarrow \\ &\quad HappensAt(a,t_2),\ t_1 \preceq t_2,\ t_2 \prec t_3, \\ &\quad Terminates(m,a,f,t_2).\end{aligned} \tag{EC10}$$

$$\begin{array}{ll} ClippedBetween(m, t_1, Neg(f), t_3) \leftarrow & \text{(EC11)} \\ \quad Happens(a, t_2),\ t_1 \preceq t_2,\ t_2 \prec t_3, & \\ \quad Initiates(m, a, f, t_2). & \end{array}$$

and the following domain-specific clauses

- For each fluent constant $F \in \phi$, the clause

  $Fluent(F)$.

- For each i-proposition "**initially** L" in τ_i, the clause

  $Initially(\lambda(L))$.

- For each o-proposition "L **holds-at** T" in τ_{ob}, the clause

  $Observation(\lambda(L), T)$.

- For each h-proposition "A **happens-at** T" in η, the clause

  $HappensAt(A, T)$.

- For each c-proposition "A **initiates** F **when** $\{L_1, \ldots, L_n\}$" in γ, the clause

  $$\begin{array}{l} Initiates(m, A, F, t) \leftarrow \\ \qquad HoldsAt(m, \lambda(L_1), t), \ldots, HoldsAt(m, \lambda(L_n), t). \end{array}$$

- For each c-proposition "A **terminates** F **when** $\{L_1, \ldots, L_n\}$" in γ, the clause

  $$\begin{array}{l} Terminates(m, A, F, t) \leftarrow \\ \qquad HoldsAt(m, \lambda(L_1), t), \ldots, HoldsAt(m, \lambda(L_n), t). \end{array}$$

Example 8.1. Let $\mathcal{E}_{ysp}$, D_{ysp} and τ_{ysp} be the domain language, domain description and observation set respectively of Example 6.1. Let $P(\Re^+, \leq)$ be an ordering program for $\mathcal{E}_{ysp}$. Then the logic program $EC[D_{ysp}, \tau_{ysp}, P(\Re^+, \leq)]$ consists of the program $P(\Re^+, \leq)$ together with clauses (EC1)–(EC11) of Definition 8.2 and the domain-specific clauses

$Terminates(m, Shoot, Alive, t) \leftarrow HoldsAt(m, Pos(Loaded), t)$.

$Fluent(Alive)$.

$Fluent(Loaded)$.

$Observation(Pos(Alive), 1).$

$Observation(Neg(Alive), 3).$

$HappensAt(Shoot, 2).$

Although potentially somewhat inefficient, the programs described in Definition 8.2 are of interest because they enable sound derivations[9]of t-propositions which would not be possible with the object-level logic programs given in the previous section. For example, it is easy to verify, either by inspection or using a Prolog interpreter, that

$$EC[D_{ysp}, \tau_{ysp}, P(\Re^+, \leq)] \vdash_{SLDNF} IHoldsAt(Pos(Loaded), 0)$$

Indeed, for a wide class of domains they are both sound and 'complete', in the sense of Proposition 8.1 below. Since any finite, consistent Language $\mathcal{A}$ domain as defined in [15] may be translated directly into a Language $\mathcal{E}$ projection domain description together with an observation set (see Section 3), finite $\mathcal{A}$ domains may also be given a meta-level implementation of this type[10].

Proposition 8.1. Let $\mathcal{E} = \langle \Pi, \preceq, \Delta, \Phi \rangle$ be a fluent-finite, non-converging projection language, let $P(\Pi, \preceq)$ be an ordering program for $\mathcal{E}$, and let $D = \langle \gamma, \eta, \tau_i \rangle$ be a finite, initially-consistent, fluent-independent projection domain description in $\mathcal{E}$. Let τ_{ob} be a finite observation set. Then for any fluent literal L of $\mathcal{E}$ and any $T \in \Pi$,

$$EC[D, \tau_{ob}, P(\Pi, \preceq)] \vdash_{SLDNF} IHoldsAt(\lambda(L), T)$$

if and only if

$$D, \tau_{ob} \models_i L \textbf{ holds-at } T$$

PROOF. See Appendix B.5

As stated in Proposition 5.1, it is possible to verify the consistency of the class of projection domain descriptions described in Proposition 8.1 by a syntactic check. Note that we can now build on this proposition to check for i-consistency with a given observation set, simply by verifying the success of the unground call $IExplanation(m)$. Finally, the style of the programs described in this section offers some clue as to how we might in principle implement a preference criterion added to the definition if i-entailment, by appropriately extending the program definition of $IExplanation$.

[9] In order to continue to refer to 'SLDNF derivations', we assume that the meta-level primitives *Setof* and *Forall* are appropriately re-interpreted (see [38] for details).

[10] Strictly speaking, the translation method described in Definition 8.2 is not applicable to the Language $\mathcal{A}$ type domain descriptions described in Section 3 Proposition 3.1. This is because the "complete occurrence set" of h-propositions is infinite (see Definition 3.3). But the appropriate modification to Definition 8.2 is trivial, and does not necessitate a significant change to the proof of Proposition 8.1. An example ordering program $P(\Pi_\Delta, \leq_\Delta)$ is given in Appendix C together with the necessary changes to Definition 8.2.

9. CONCLUSIONS AND FURTHER WORK

Following the methodology of the Language $\mathcal{A}$ introduced in [15], we have presented a simple declarative language, $\mathcal{E}$, for describing narratives with actions. $\mathcal{E}$ is based on a narrative ontology inherited from the Event Calculus, thus demonstrating that this methodology is not limited to the particular ontology of $\mathcal{A}$. $\mathcal{E}$'s semantics is based around a simple characterisation of persistence which facilitates a modular approach to extending the expressivity of the language. This characterisation relies heavily on the notion of a flow of time which is independent from any actions which may occur. The benefits of this become particularly apparent when representing domains where periods of time elapse in which a change may or may not have taken place. It is not necessary to 'fill in' time with an artificial 'action' such as a '*Wait*'.

The explicit notion of an action occurrence incorporated in $\mathcal{E}$ allows an important class of 'narrative' explanations (h-explanations) to be characterised in a simple way. These enable us to extend an otherwise inconsistent theory written in $\mathcal{E}$ so as to establish consistency, thus providing a natural method, in many cases, to account for conflicting sets of information about the domain. More generally, our formalisation of various notions of explanation within $\mathcal{E}$ illustrates that commonsense reasoning need not always be modelled as deduction at a single object level. Our results are built upon much previous work concerning the role of abduction in Artificial Intelligence and related areas. Once again we have demonstrated that reasoning from cause to effect can be modelled at the object level, whereas reasoning from effect to cause can be regarded as an essentially meta-level (for example abductive) activity. In the context of reasoning about action, causation is temporally directed. Hence in our work this distinction manifests itself in the fact that reasoning forwards in time is modelled as object-level deduction, whereas reasoning backwards in time is captured at the meta-level. The success of this approach lends extra weight to a developing consensus (see [8] for a general discussion) that observations should somehow be treated separately from other aspects of theories of action.

We have also shown how domains in $\mathcal{E}$ can be implemented in normal logic programming with extended versions of Event Calculus programs that behave correctly even when the knowledge entailed by the domain description is incomplete. These programs have the capability of reasoning backwards as well as forwards in time.

We can envisage at least three areas of future research relating to $\mathcal{E}$. Firstly, in line with the methodology described in our introduction, we could use $\mathcal{E}$ as a 'measuring stick' to show correspondences between various narrative-based formalisms for reasoning about action, perhaps in the manner of Katha in [20]. Example candidates for comparison are the formalisms in [30], [32] and [37].

Secondly, it would be interesting to investigate different styles of implementation as regards $\mathcal{E}$ domains. The approaches followed in this paper are based on a relatively simple use of normal logic programming and standard techniques within this. We could develop a more general implementation for computing entailment or i-entailment using abductive logic programming, building on the work in [10], [11], [18] and [19]. The 'abductive flavour' of our definition of h-entailment suggests that abductive logic programming could be a useful implementation tool here as well.

Thirdly, the expressivity of $\mathcal{E}$ could be increased in various ways. We have already briefly indicated how $\mathcal{E}$ might be extended to partially deal with *ramifications* and *qualifications* (see Appendix A). Since $\mathcal{E}$ already allows for actions to occur *concurrently* within a narrative, it seems likely that the language could also be extended to allow for a theory of cancelling and combined effects of actions similar to that in [4]. It has already been pointed out in [31] that a narrative based approach offers alternative ways to model *non-deterministic* effects of actions. Finally, we might extend the syntax and semantics of $\mathcal{E}$ to deal with incomplete information about the order and timing of action occurrences, perhaps introducing temporal variables into the language in a manner similar to [5] and [6].

The utility and appeal of specialised declarative languages such as $\mathcal{A}$ and $\mathcal{E}$ lies in their simplicity. They are of sufficiently 'high level' to allow various issues to be aired without immediately becoming involved in technical details, and are perhaps best regarded as useful stepping stones towards the ultimate goal of developing comprehensive formal theories of action using general purpose representational mechanisms. Hence their capacity for retaining their simplicity when extended to cover more complex domains is a crucial measure of their utility.

Acknowledgements

We would particularly like to thank Marc Denecker, Yannis Dimopoulos, David Evans, Alessandro Provetti, Alessandra Russo, Fariba Sadri, Murray Shanahan and Eugenia Ternovskaia for helpful discussions on the topic of this paper, and for careful reading of earlier drafts. Thanks as well to the anonymous referees for their thoughtful and detailed comments, which helped considerably in improving the paper.

REFERENCES

1. James Allen, *Towards a General Theory of Action and Time*, Artificial Intelligence 23, Elsevier Science Publishers, pages 123-154, 1984.
2. Jonathan Amsterdam, *Temporal Reasoning and Narrative Conventions*, Proceedings of the 2nd International Conference on Knowledge Representation (KR 91), ed.s J. Allen, R. Fikes and E. Sandewall, Morgan Kaufmann, pages 15–21, 1991.
3. Andrew Baker, *Nonmonotonic Reasoning in the Framework of the Situation Calculus*, Artificial Intelligence 49, Elsevier Science Publishers, page 5, 1991.
4. Chitta Baral and Michael Gelfond, *Representing Concurrent Actions in Extended Logic Programming*, Proceedings IJCAI 93, Morgan Kaufmann, page 866, 1993.
5. Chitta Baral, Michael Gelfond and Alessandro Provetti, *Representing Actions - I: Laws, Observations and Hypotheses (Short Version)*, in Working Notes of the AAAI Spring Symposium: Extending Theories of Action - Formal Theory and Practical Applications, Stanford University, California, USA, 1995.
6. Chitta Baral, Michael Gelfond and Alessandro Provetti, *Representing Actions: Laws, Observations and Hypotheses*, Journal of Logic Programming, Special Issue on Reasoning about Actions (this issue), 1996.
7. Iliano Cervesato, Luca Chittaro and Angelo Montanari, *A Modal Calculus of Partially Ordered Events in a Logic Programming Framework*, in Proceedings ICLP'95, pages 299-313, 1995.

8. James Crawford and David Etherington, *Observations on Observations in Action Theories*, in Working Notes of the AAAI Spring Symposium: Extending Theories of Action - Formal Theory and Practical Applications, Stanford University, California, USA, 1995.

9. Ernest Davis, *Infinite Loops in Finite Time: Some Observations*, Proceedings KR 92 (3rd International Conference on Principles of Knowledge Representation and Reasoning), Cambridge, Massachusetts, ed.s B. Nebel, C. Rich and W. Swartout, Morgan Kaufmann, 1992.

10. Marc Denecker, Lode Missiaen and Maurice Bruynooghe, *Temporal Reasoning with Abductive Event Calculus*, in Proceedings ECAI 92, Vienna, 1992.

11. Marc Denecker and Danny De Schreye, *Reperesenting Incomplete Knowledge in Abductive Logic Programming*, in Proceedings of the International Symposium on Logic Programming, 1993.

12. Marc Denecker, *A Terminological Interpretation of (Abductive) Logic Programming*, in Proceedings of the Third International Conference on Logic Programming and Non-monotonic Reasoning, Lexington, KY, USA, Springer Verlag, 1995.

13. Phan Minh Dung, *Representing Actions in Logic Programming and its Applications in Database Updates*, Proceedings of the Tenth International Conference on Logic Programming, ed David S. Warren, MIT Press, pages 222-238, 1993.

14. Kave Eshghi, *Abductive Planning with Event Calculus*, Proceedings of the 5th International Conference and Symposium on Logic Programming, ed.s Robert Kowalski and Kenneth Bowen, MIT Press, pages 562-579, 1988.

15. Michael Gelfond and Vladimir Lifschitz, *Representing Actions in Extended Logic Programming*, Proceedings of the Joint International Conference and Symposium on Logic Programming, ed. Krzysztof Apt, MIT Press, page 560, 1992.

16. Michael Gelfond and Vladimir Lifschitz, *Representing Action and Change by Logic Programs*, Journal of Logic Programming, volume 17 numbers 2, 3 and 4, North-Holland, pages 301-322, 1993.

17. David Harel, *Dynamic Logic*, In Handbook of Philosophical Logic II: Extensions of Classical Logic, ed.s D. Gabbay and F. Guenther, Reidel, Boston, USA, pages 497-604, 1984.

18. Antonios Kakas and Paolo Mancarella, *Generalized Stable Models: a Semantics for Abduction.* Proceedings of the 9th European Conference on Artificial Intelligence, Stockholm, ed. L. Aiello, pages 385-391, 1990.

19. Antonios Kakas, Robert Kowalski and Francesca Toni, *Abductive logic programming*, Journal of Logic and Computation vol. 2 no. 6, pages 719-770, 1993.

20. G. Neelakantan Kartha, *Soundness and Completeness Theorems for Three Formalizations of Action*, Proceedings IJCAI 93, page 724, 1993.

21. G. Neelakantan Kartha, *Two Counterexamples Related to Baker's Approach to the Frame Problem*, Artificial Intelligence 69, Elsevier Science Publishers, pages 379-392, 1994.

22. Henry Kautz, *The Logic of Persistence*, Proceedings AAAI 86, page 401, 1986.

23. Robert A. Kowalski and Marek J. Sergot, *A Logic-Based Calculus of Events*, New Generation Computing, vol 4, page 267, 1986.

24. Robert A. Kowalski and Fariba Sadri, *The Situation Calculus and Event Calculus Compared*, in Proceedings of the International Logic Programming Symposium (ILPS'94), 1994.

25. Vladimir Lifschitz, *A Language for Describing Actions*, in Working Papers of Common Sense '93: The Second Symposium on Logical Formalizations of Commonsense Reasoning, pages 103-113, Austin, Texas, U.S.A., 1992.

26. Fangzhen Lin and Yoav Shoham, *Provably Correct Theories of Action*, in Proceedings AAAI 91, page 349, MIT Press, 1991.

27. John W. Lloyd, *Foundations of Logic Programming*, Springer Verlag, 1984.

28. John McCarthy and Patrick Hayes, *Some Philosophical Problems from the Standpoint of Artificial Intelligence*, in Machine Intelligence 4, ed.s D. Michie and B. Meltzer, Edinburgh University Press, 1969.

29. Rob Miller, *Situation Calculus Specifications for Event Calculus Logic Programs*, in Proceedings of the Third International Conference on Logic Programming and Non-monotonic Reasoning, Lexington, KY, USA, Springer Verlag, 1995.

30. Rob Miller and Murray Shanahan, *Narratives in the Situation Calculus*, in Journal of Logic and Computation, Special Issue on Actions and Processes, vol 4 no 5, Oxford University Press, 1994.

31. Javier Pinto, *Temporal Reasoning in the Situation Calculus*, PhD. Thesis, University of Toronto, 1994.

32. Javier Pinto and Raymond Reiter, *Temporal Reasoning in Logic Programming: A Case for the Situation Calculus*, Proceedings ICLP 93, page 203, 1993.

33. Alessandro Provetti, *Hypothetical Reasoning about Actions: From Situation Calculus to Event Calculus*, Computational Intelligence, volume 12, number 2, 1995.

34. Erik Sandewall, *Feature and Fluents*, Oxford University Press, 1994.

35. Marek Sergot, *An Introduction to the Event Calculus*, in Lecture Notes of the GULP Advanced School on Foundations of Logic Programming, (unpublished), Alghero, Sardinia, 1990.

36. Murray Shanahan, *Prediction Is Deduction but Explanation Is Abduction*, Proceedings IJCAI 89, pages 1055-1060, 1989.

37. Murray Shanahan, *A Circumscriptive Calculus of Events*, Artificial Intelligence, vol 75 no 2, Elsevier Science Publishers, 1995.

38. Leon Sterling and Ehud Shapiro, *The Art of Prolog*, MIT Press, 1986.

39. Kristof Van Belleghem, Marc Denecker and Danny De Schreye, *On the Relation Between Situation Calculus and Event Calculus*, Journal of Logic Programming, Special Issue on Reasoning about Actions (this issue), 1996.

APPENDICES

A. EXTENDING THE EXPRESSIVITY OF $\mathcal{E}$

Two extensions to the syntax and semantics of $\mathcal{E}$ are given in this appendix. This is in order to illustrate that the basic notion of a model, encapsulated in Definitions 2.7 to 2.10, may be modified to accommodate extra types of propositions, without altering the basic principle of persistence captured in conditions (1)-(3) of Definition 2.10. Both extensions are very simple, and although they are obviously related to aspects of the *qualification problem* and *ramification problem* respectively, it is not our intention to suggest that, in this short space, we have developed a comprehensive approach to these subtle and complex issues.

A.1. Describing Conditions under which an Action Cannot Occur

In some circumstances it may be possible to infer knowledge about the conditions at the time of an action occurrence from the fact that the action did occur. For example, given that we know that "at 2 o'clock the caretaker unlocked the door", we might typically infer that (at 2 o'clock) "she had the key". This is because it is impossible to unlock a door without a key, which might be expressed by a proposition such as

$$Unlock\ \textbf{impossible-if}\ \{\neg HasKey\}$$

This motivates a general definition for a new type of proposition:

Definition A.1. [q-proposition] A *q-proposition* in $\mathcal{E}$ is an expression of the form

$$A\ \textbf{impossible-if}\ C$$

where $A \in \Delta$, and C is a set of fluent literals of $\mathcal{E}$.

Such propositions may be accommodated in the semantics of $\mathcal{E}$ by strengthening condition (4) of Definition 2.10 (which expresses simple pointwise constraints on a model), without changing in the basic notion of persistence encapsulated in conditions (1)-(3). Assuming that domain descriptions are now defined as a quadruple $\langle\gamma, \eta, \tau, \kappa\rangle$, where γ, η and τ are as before, and κ is a set of q-propositions[11], the condition now becomes:

(a) For all t-propositions in τ of the form "F **holds-at** T", $H(F,T) = true$, and for all t-propositions of the form "$\neg F$ **holds-at** T", $H(F,T) = false$.

(b) For all pairs of h-propositions and q-propositions in $\eta \times \kappa$ of the form "A **happens-at** T" and "A **impossible-if** C", H does not satisfy C at T.

A.2. Describing Indirect Effects of Actions

The following extension to $\mathcal{E}$ is included to illustrate how Definition 2.9 of an initiation or termination point might be refined, without necessitating a change in the basic notion of persistence encapsulated in conditions (1)-(3) of Definition 2.10. Suppose that we wish to express simple constraints between fluents. To take a canonical example, suppose that we want to express that a room is stuffy when the window is closed and the ventilator blocked. This might be expressed by a proposition such as

$$Stuffy\ \textbf{whenever}\ \{Closed, Blocked\}$$

Hence we define a new type of proposition as follows:

Definition A.2. [r-proposition] An *r-proposition* in $\mathcal{E}$ is an expression of the form

[11] Note that, if for every c-proposition "A **initiates** F **when** C" there is a q-proposition "A **impossible-if** $C \cup \{F\}$", in any model all initiation points for F are actual points of change for F. An analogous observation holds for termination points. Thus Sergot's notions of *strong initiation* and *strong termination* [35] can be incorporated into $\mathcal{E}$ by addition of q-propositions where appropriate.

L **whenever** C

where L is a fluent literal and C is a set of fluent literals of $\mathcal{E}$.

We now need a recursive definition of an initiation point and of a termination point, since, for example, if the ventilator is blocked, the action of closing the window will (indirectly) initiate the property of the room being stuffy. The modified definitions below assume that domain descriptions are defined as a tuple $\langle\gamma, \eta, \tau, \kappa, \rho\rangle$, where γ, η and τ are as before, κ is a set of q-propositions and ρ is a set of r-propositions.

Definition A.3. [Initiation/termination point for domains with r-propositions] Let H be an interpretation of $\mathcal{E}$, let $D = \langle\gamma, \eta, \tau, \kappa, \rho\rangle$ be a domain description, let $F \in \Phi$ and let $T \in \Pi$. T is an *initiation-point* (respectively *termination-point*) *for F in H relative to D* iff one of the following two conditions holds.

1. There is an $A \in \Delta$ such that (i) there is both an h-proposition in η of the form "A **happens-at** T" and a c-proposition in γ of the form "A **initiates** F **when** C" (respectively "A **terminates** F **when** C") and (ii) H satisfies C at T.
2. There is an r-proposition in ρ of the form "F **whenever** C" (respectively "$\neg F$ **whenever** C") and a partition $\{C_1, C_2\}$ of C such that (i) C_1 is non-empty, for each fluent constant $F' \in C_1$, T is an initiation point for F', and for each fluent literal $\neg F' \in C_1$, T is a termination point for F', and (ii) there is some $T_2 \in \Pi$, $T \prec T_2$, such that for all T_1, $T \preceq T_1 \preceq T_2$, H satisfies C_2 at T_1.

Intuitively, condition (2) above states that in order find time points at which the fluent F becomes indirectly initiated via the r-proposition F **whenever** C, we need to look for points at which one or more of the conditions in C become satisfied, and at which the remaining conditions were already satisfied (and continue to be satisfied up to some point T_2 beyond the point in question).

Condition (4) of Definition 2.10 is now:

(a) For all t-propositions in τ of the form "F **holds-at** T", $H(F, T) = \mathit{true}$, and for all t-propositions of the form "$\neg F$ **holds-at** T", $H(F, T) = \mathit{false}$.

(b) For all pairs of h-propositions and q-propositions in $\eta \times \kappa$ of the form "A **happens-at** T" and "A **impossible-if** C", H does not satisfy C at T.

(c) For all r-propositions in ρ of the form "L **whenever** C", if H satisfies C at T then H satisfies $\{L\}$ at T.

B. PROPOSITION PROOFS

B.1. Proof of Proposition 2.1

PROPOSITION STATEMENT: *Let D be an occurrence-sparse domain description written in a language $\mathcal{E} = \langle\Pi, \preceq, \Delta, \Phi\rangle$, and let $T_1, T_2 \in \Pi$ be such that $T_1 \preceq T_2$. Let H*

and H' be models of D such that for all $F \in \Phi$, $H(F,T_1) = H'(F,T_1)$. Then for all $F \in \Phi$, $H(F,T_2) = H'(F,T_2)$.

PROOF: Proof is by induction on the number n of h-propositions in η of the form A **happens-at** T such that $T_1 \preceq T \prec T_2$.

Base Case: If $n = 0$ then by the first condition in the definition of a model, for all $F \in \Phi$, $H(F,T_2) = H(F,T_1) = H'(F,T_1) = H'(F,T_2)$.

Inductive Step: Suppose that $n > 0$ and that the lemma is true for all $m < n$. Let $F' \in \Phi$ be an arbitrary fluent constant. It is sufficient to show that $H(F',T_2) = H'(F',T_2)$. Since D is occurrence-sparse there exists at least one $T \in \Pi$ such that (i) $T_1 \preceq T \prec T_2$, (ii) there is at least one h-proposition in η of the form "A **happens-at** T", and (iii) there is no h-proposition in η of the form "A **happens-at** T'" such that $T \prec T' \prec T_2$. Let T_h be such a time point. By the inductive hypothesis, for all $F \in \Phi$, $H(F,T_h) = H'(F,T_h)$ and by construction there are no initiation or termination points T' (for any fluent) in H or H' such that $T_h \prec T' \prec T_2$. There are three cases to consider (since in any model, F' must be unaffected, initiated or terminated at T_h):

Case one: There is not both an h-proposition in η of the form "A **happens-at** T_h" and a c-proposition in γ either of the form "A **initiates** F' **when** C" or of the form "A **terminates** F' **when** C" such that H (and thus H') satisfies C at T_h. Hence by the first condition in the definition of a model, $H(F',T_2) = H(F',T_h) = H'(F',T_h) = H'(F',T_2)$.

Case two: There is both an h-proposition in η of the form "A **happens-at** T_h" and a c-proposition in γ of the form "A **initiates** F' **when** C" such that H (and thus H') satisfies C at T_h. Hence by the second condition in the definition of a model, $H(F',T_2) = \mathit{true} = H'(F',T_2)$.

Case three: There is both an h-proposition in η of the form "A **happens-at** T_h" and a c-proposition in γ of the form "A **terminates** F' **when** C" such that H (and thus H') satisfies C at T_h. Hence by the third condition in the definition of a model, $H(F',T_2) = \mathit{false} = H'(F',T_2)$.

B.2. Proof of Proposition 5.1

PROPOSITION STATEMENT: *Let D be an occurrence-sparse, non-converging, initially-consistent, fluent-independent projection domain description. Then D is consistent.*

PROOF: Let $D = \langle \gamma, \eta, \tau_i \rangle$ be written in the projection language $\mathcal{E} = \langle \Pi, \preceq, \Delta, \Phi \rangle$, and let T_0 be the null element of Π.

Let $M : \Phi \mapsto \{\mathit{true}, \mathit{false}\}$ be defined as follows. For each $F \in \Phi$,

- $M(F) = \mathit{true}$ if there is an i-proposition in τ_i of the form "**initially** F",
- $M(F) = \mathit{false}$ otherwise.

M will be used to construct a model H of D such that for all $F \in \Phi$, $H(F,T_0) = M(F)$. Notice that since D is initially consistent, then for each $F \in \Phi$ such that there is an i-proposition in τ_i of the form "**initially** $\neg F$", $M(F) = false$.

Since D is occurrence-sparse and non-converging, each time point $T \in \Pi$ has a unique, maximal, finite (possibly empty) sequence $T_1, \ldots, T_n$ associated with it such that $T_1 \prec \ldots \prec T_n \prec T$ and such that for each T_i there is an h-proposition in η of the form "A **happens-at** T_i". Moreover, the unique such sequence associated with T_n is $T_1, \ldots, T_{n-1}$. Therefore, it is possible to define an interpretation H of D inductively as follows. For each $T \in \Pi$ and $F \in \Phi$,

1. $H(F,T) = M(F)$ if $n = 0$ (i.e. the sequence associated with T is empty),
2. $H(F,T) = H(F,T_n)$ if $n > 0$ and there is no $A \in \Delta$ such that there is both an h-proposition in η of the form "A **happens-at** T_n" and a c-proposition in γ either of the form "A **initiates** F **when** C" or of the form "A **terminates** F **when** C" such that H satisfies C at T_n,
3. $H(F,T) = true$ if $n > 0$ and there is an $A \in \Delta$ such that there is both an h-proposition in η of the form "A **happens-at** T_n" and a c-proposition in γ of the form "A **initiates** F **when** C" such that H satisfies C at T_n,
4. $H(F,T) = false$ if $n > 0$ and there is an $A \in \Delta$ such that there is both an h-proposition in η of the form "A **happens-at** T_n" and a c-proposition in γ of the form "A **terminates** F **when** C" such that H satisfies C at T_n.

The fact that D is fluent-independent guarantees that no fluent/time-point pair (F,T) satisfies both of conditions 3 and 4 above. Hence H is well defined, and is clearly a model of D.

B.3. Proof of Proposition 6.1

PROPOSITION STATEMENT: *Let $D = \langle \gamma, \eta, \tau_i \rangle$ be an occurrence-sparse projection domain description and let τ_{ob} be an observation set. Then H is a model of $\langle \gamma, \eta, \tau_i \cup \tau_{ob} \rangle$ if and only if H is an i-model of D with τ_{ob}.*

PROOF: Let D be written in the projection language $\mathcal{E} = \langle \Pi, \preceq, \Delta, \Phi \rangle$, and let T_0 be the null element of Π.

"If" half: Suppose H is an i-model of D with τ_{ob}. Then by the definitions of an model and of an i-model there exists some set $\tau_{i\epsilon}$ of i-propositions such that H is a model of $\langle \gamma, \eta, \tau_i \cup \tau_{i\epsilon} \cup \tau_{ob} \rangle$. Hence, by the monotonicity of $\mathcal{E}$ as regards addition of t-propositions to domain descriptions (see the remarks at the end of Section 2), H is a model of $\langle \gamma, \eta, \tau_i \cup \tau_{ob} \rangle$.

"Only if" half: Suppose H is a model of $\langle \gamma, \eta, \tau_i \cup \tau_{ob} \rangle$. Let the set τ_H of i-propositions be defined as follows. For each $F \in \Phi$,

- **initially** $F \in \tau_H$ iff $H(F,T_0) = true$
- **initially** $\neg F \in \tau_H$ iff $H(F,T_0) = false$

Clearly H is a model of $\langle \gamma, \eta, \tau_i \cup \tau_H \rangle$ and, by Proposition 2.1, $\langle \gamma, \eta, \tau_i \cup \tau_H \rangle \models p$ for each $p \in \tau_{ob}$, so that τ_H is an i-explanation for τ_{ob} in D. Hence H is an i-model of D with τ_{ob}.

B.4. Proof of Proposition 7.4

PROPOSITION STATEMENT:
Let $P(\Pi, \preceq)$ be an ordering program for $\mathcal{E}$, and let D be a finite domain description. Then for any fluent literal L of $\mathcal{E}$ and any $T \in \Pi$, if

$$LP[D, P(\Pi, \preceq)] \vdash_{SLDNF} \mathit{HoldsAt}(\lambda(L), T)$$

then

$$D \models L \textbf{ holds-at } T$$

The proof of this proposition which is given below uses induction on the 'length' $\mathit{length}(\alpha)$ of the SLDNF derivation α of $\mathit{HoldsAt}(\lambda(L), T)$, where $\mathit{length}(\alpha)$ is defined in Definition B.1 below in terms of successful calls to *HappensAt*. It is defined so that each SLDNF sub-derivation of a *HoldsAt* sub-goal within α (which must have occurred within some call to *Initiates*, *Terminates*, *PossiblyInitiates* or *PossiblyTerminates*) has 'length' less than the top-level derivation α.

Definition B.1. [$\mathit{length}(\alpha)$] Let α be a successful SLDNF derivation of the goal $\mathit{HoldsAt}(\lambda(L), T)$ in $LP[D, P(\Pi, \preceq)]$. $\mathit{length}(\alpha)$ is defined inductively as follows:

$$\mathit{length}(\alpha) = S + \sum_{\beta \in B_\alpha} \mathit{size}(\beta)$$

where

- S is the number of successful calls to *HappensAt* at the top level of α, i.e. not called within a (negation-as-failure) finitely-failed subsidiary derivation of α.

- B_α is the set of all finitely-failed subsidiary derivations from negative calls (to *ClippedBetween* or *AffectedBetween*) appearing at the top level of α.

- $\mathit{size}(\beta) = 0$ if there is no successful call to *HappensAt* in any branch of β.

- $\mathit{size}(\beta) = 1 + \max(\{\mathit{length}(\alpha') \mid \alpha' \in A_\beta\})$ if there is a successful call to *HappensAt* inside β, where A_β is the set of all successful SLDNF derivations of a *HoldsAt* goal, called as a negated sub-goal in one of the branches of β[12].

PROOF OF PROPOSITION. Let α be a successful SLDNF derivation of the goal $\mathit{HoldsAt}(\lambda(L), T)$ in $LP[D, P(\Pi, \preceq)]$. We will use induction on $\mathit{length}(\alpha)$ to show that, given the fixed point I^+ as defined in Proposition 7.3, if $LP[D, P(\Pi, \preceq)] \vdash_{SLDNF} \mathit{HoldsAt}(\lambda(L), T)$ then if $L = F$ for some $F \in \Phi$ then $I^+(F, T) = \mathit{true}$, and if $L = \neg F'$ for some $F' \in \Phi$ then $I^+(F', T) = \mathit{false}$. The proposition will then follow directly from Proposition 7.3.

[12] We assume the convention that $\max(\emptyset) = 0$.

Base Case ($length(\alpha) = 0$):
Clearly, if $length(\alpha) = 0$ the query $HoldsAt(Pos(F), T)$ (respectively $HoldsAt(Neg(F'), T)$) can succeed only on clauses (LP1a), (LP1b) or (LP4), as success on (LP2) or (LP3) would require $length(\alpha) \geq 1$. We consider each of these possibilities in turn:

(i) Success on (LP1a): Clearly, the success of $Given(\lambda(L), T_1)$ for some $T_1 < T$ means that "L **holds-at** T_1" $\in D$ and so $I^+(F, T_1) = true$ (respectively $I^+(F', T_1) = false$) by rule (4) in the definition of $\mathcal{F}$. Also, since the call β to $AffectedBetween$ fails with $size(\beta) = 0$, the unground sub-goal $HappensAt(a, t_2)$ fails, so that there are no h-propositions in D. Hence there are no possible initiation points or possible termination points between T_1 and T. Therefore, since I^+ is a fixed point of $\mathcal{F}$, rule (1a) of $\mathcal{F}$ applies to give $I^+(F, T) = true$ (respectively $I^+(F', T) = false$).

(ii) Success on (LP1b): The proof is exactly analogous to (i), but using rule (1b) of $\mathcal{F}$ in place of rule (1a).

(iii) Success on (LP4): Trivially, $I^+(F, T) = true$ (respectively $I^+(F', T) = false$) by rule (4) in the definition of $\mathcal{F}$.

Inductive Step ($length(\alpha) = n$):
Suppose that the statement which we wish to prove is true for all SLDNF derivations α' of all $HoldsAt$ goals such that $length(\alpha') < n$. The query $HoldsAt(Pos(F), T)$ (respectively$HoldsAt(Neg(F'), T)$) can succeed only on clauses (LP4), (LP1a), (LP1b) or (LP2) (respectively (LP3)). Again, we consider each of these possibilities in turn:

(i) Success on (LP4): Trivially, $I^+(F, T) = true$ (respectively $I^+(F', T) = false$) by rule (4) in the definition of $\mathcal{F}$.

(ii) Success on (LP1a): Since the sub-goals $Given(\lambda(L), t_1), t_1 \prec T$ succeed, $I^+(F, T_1) = true$ (respectively $I^+(F', T_1) = false$) by rule (4) in the definition of $\mathcal{F}$ (where T_1 is the binding to t_1). It remains to show that there is no possible initiation point or possible termination point for F (respectively F') between T_1 and T, so that rule (1a) in the definition of $\mathcal{F}$ may be applied. Trivially, if the sub-goals $HappensAt(a, t_2)$, $T_1 \preceq t_2$ and $t_2 \prec T$ in the body of each $ClippedBetween$ clause collectively fail, there can be no such point. Now suppose these sub-goals succeed, binding t_2 to T_2. Since the call $AffectedBetween(T_1, \lambda(L), T_3)$ fails (so that the top level goal succeeds on clause (LP1a)), both of the calls $PossiblyInitiates(A, F, T_2)$ and $PossiblyTerminates(A, F, T_2)$ (respectively $PossiblyTerminates(A, F', T_2)$ and $PossiblyInitiates(A, F', T_2)$) fail. Hence, if there is a c-proposition in D of the form "A **initiates** F **when** C" or "A **terminates** F **when** C" (respectfully "A **terminates** F' **when** C" or "A **initiates** F' **when** C"), there is a fluent literal $L_p \in C$ such that *not* $HoldsAt(\overline{\lambda(L_p)}, T_2)$ fails, i.e. $HoldsAt(\overline{\lambda(L_p)}, T_2)$ succeeds, say with SLDNF derivation α', where $length(\alpha') < n$. By the induction hypothesis, if $L_p = F_p$ then $I^+(F_p, T_2) = false$, and if $L_p = \neg F'_p$ then $I^+(F'_p, T_2) = true$. In either case, the c-proposition is therefore not applicable in the definition of a possible initiation point or possible termination point of F (respectively F') relative to I^+. Hence rule (1a) in the definition of $\mathcal{F}$ applies to give $I^+(F, T) = true$ (respectively $I^+(F', T) = false$) as required.

(iii) Success on (LP1b): The argument in this case is exactly analogous to case (ii), but using rule (1b) (instead of rule (1a)) in the definition of $\mathcal{F}$.

(iv) Success on (LP2): In this case there exists a $T_1 \in \Pi$, $T_1 \prec T$, such that for some $A \in \Delta$ the propositions "A **happens-at** T_1" and "A **initiates** F **when**

$\{L_1, \ldots, L_k\}$" belong to D, and each of the calls $HoldsAt(\lambda(L_1), T_1), \ldots,$ $HoldsAt(\lambda(L_k), T_1)$ succeed. The successful SLDNF derivations of each of these calls are of length strictly less than n due to the successful call of $HappensAt(A, T_1)$ in the root SLDNF derivation of $HoldsAt(Pos(F), T)$. Hence by the inductive hypothesis, for each $L_i = F_i$, $I^+(F_i, T_1) = true$, and for each $L_j = \neg F_j$, $I^+(F_j, T_1) = false$. Hence T_1 is an initiation point for F relative to I^+. By an argument exactly analogous to that in case (ii) above, we can show from the finite failure of $ClippedBetween(T_1, Pos(F), T)$ that there are no possible termination points for F between T_1 and T, so that rule (2) in the definition of $\mathcal{F}$ applies to give $I^+(F, T) = true$ as required.

(v) Success on (LP3): The argument in this case is exactly analogous to case (iv), but using rule (3) (instead of rule (2)) in the definition of $\mathcal{F}$ to show that $I^+(F', T) = false$.

B.5. Proof of Proposition 8.1

Lemma B.1. Let $\mathcal{E} = \langle \Pi, \preceq, \Delta, \Phi \rangle$ be a fluent-finite, non-converging projection language, let $P(\Pi, \preceq)$ be an ordering program for $\mathcal{E}$, and let $D = \langle \gamma, \eta, \tau_i \rangle$ be a finite, initially-consistent, fluent-independent projection domain description in $\mathcal{E}$. Let H be a model of D. Let M be a ground list term such that the ground query $Member(\lambda, M)$ succeeds iff there is a fluent constant $F' \in \Phi$ such that either $\lambda = Neg(F')$ and $H(F', T_0) = false$ or $\lambda = F'$ and $H(F', T_0) = true$. Then for all $F \in \Phi$ and $T \in \Pi$,

$$EC[D, \emptyset, P(\Pi, \preceq)] \vdash_{SLDNF} HoldsAt(M, Pos(F), T)$$

if and only if $H(F, T) = true$, and

$$EC[D, \emptyset, P(\Pi, \preceq)] \vdash_{SLDNF} HoldsAt(M, Neg(F), T)$$

if and only if $H(F, T) = false$.

PROOF. Let T and F be an arbitrary time-point and fluent constant. Since D is finite and non-converging, T has a unique, maximal, finite (possibly empty) sequence $T_1, \ldots, T_n$ associated with it such that $T_1 \prec \ldots \prec T_n \prec T$ and such that for each T_i there is an h-proposition in η of the form "A **happens-at** T_i". Proof is by induction on the length n of this sequence.

Base Case:
Clearly, if $n = 0$ the queries $HoldsAt(M, Pos(F), T)$ and $HoldsAt(M, Neg(F), T)$ can succeed only on clause (EC7), and will succeed if and only if the queries $Member(Pos(F), M)$ and $Member(Neg(F), M)$ succeed respectively. By the second condition in Definition 2.10 of a model, $H(F, T) = H(F, T_0)$, so that by definition of the list term M the lemma is true in the base case.

Inductive Step:
Suppose that the lemma is true for all time-points whose associated sequences are of length $m < n$. Then in particular it is true for T_n whose associated sequence is of length $n - 1$. There are three cases to consider:

Case one: There is both an h-proposition in η of the form "A **happens-at**

T_n" and a c-proposition in γ of the form "A **initiates** F **when** C" such that H satisfies C at T_n. Hence by the third condition in the definition of a model, $H(F,T) = true$.

In this case, by the inductive hypothesis and the program definition of *Initiates*, the query $HoldsAt(M, Pos(F), T)$ will succeed on clause (EC8) with the program variable t_1 in the body of the clause bound to T_n. The query $HoldsAt(M, Neg(F), T)$ will fail on clause (EC7) because the sub-goal $ClippedBetween(M, T_0, Neg(F), T)$ will succeed on clause (EC11) with the program variable t_2 in the body of the clause bound to T_n. The query $HoldsAt(M, Neg(F), T)$ will fail on clause (EC9) because by the inductive hypothesis and fluent-independence of D it will fail on the sub-goal $Terminates(M, a, F, T_n)$ for all bindings of the variable a provided by solutions to $Happens(a, T_n)$. Hence in this case the lemma is true.

Case two: There is both an h-proposition in η of the form "A **happens-at** T_n" and a c-proposition in γ of the form "A **terminates** F **when** C" such that H satisfies C at T_n. Hence by the third condition in the definition of a model, $H(F,T) = false$.

In this case, by the inductive hypothesis and the program definition of *Initiates*, the query $HoldsAt(M, Neg(F), T)$ will succeed on clause (EC9) with the program variable t_1 in the body of the clause bound to T_n. The query $HoldsAt(M, Pos(F), T)$ will fail on clause (EC7) because the sub-goal $ClippedBetween(M, T_0, Pos(F), T)$ will succeed on clause (EC10) with the program variable t_2 in the body of the clause bound to T_n. The query $HoldsAt(M, Pos(F), T)$ will fail on clause (EC8) because by the inductive hypothesis and fluent-independence of D it will fail on the sub-goal $Initiates(M, a, F, T_n)$ for all bindings of the variable a provided by solutions to $Happens(a, T_n)$. Hence in this case the lemma is also true.

Case three: There is not both an h-proposition in η of the form "A **happens-at** T_n" and a c-proposition in γ either of the form "A **initiates** F **when** C" or of the form "A **terminates** F **when** C" such that H satisfies C at T_n. Hence by the second condition in the definition of a model, $H(F,T) = H(F,T_n)$.

Clearly in this case, by the inductive hypothesis and by the program definitions of *Initiates* and *Terminates*, the queries $HoldsAt(M, Pos(F), T)$ and $HoldsAt(M, Neg(F), T)$ can succeed on the clauses (EC8) and (EC9) respectively only with the variable t_1 in the body of each clause bound to some time-point $T_i < T_n$. Moreover, by the same argument, for all $T' < T_n$, the queries $ClippedBetween(M, T', Pos(F), T)$ and $ClippedBetween(M, T', Neg(F), T)$ succeed if and only if the queries $ClippedBetween(M, T', Pos(F), T_n)$ and $ClippedBetween(M, T', Neg(F), T_n)$ succeed respectively. Hence the queries $HoldsAt(M, Pos(F), T)$ and $HoldsAt(M, Neg(F), T)$ succeed if and only if the queries $HoldsAt(M, Pos(F), T_n)$ and $HoldsAt(M, Neg(F), T_n)$ succeed respectively. Hence in this case the lemma is also true.

Statement of Main Proposition: *Let $\mathcal{E} = \langle \Pi, \preceq, \Delta, \Phi \rangle$ be a fluent-finite, non-converging projection language, let $P(\Pi, \preceq)$ be an ordering program for $\mathcal{E}$, and let $D = \langle \gamma, \eta, \tau_i \rangle$ be a finite, initially-consistent, fluent-independent projection domain description in $\mathcal{E}$. Let τ_{ob} be a finite observation set. Then for any fluent literal L*

of $\mathcal{E}$ and any $T \in \Pi$,

$$EC[D, \tau_{ob}, P(\Pi, \preceq)] \vdash_{SLDNF} IHoldsAt(\lambda(L), T)$$

if and only if

$$D, \tau_{ob} \models_i L \textbf{ holds-at } T$$

PROOF:
Let an *initial assignment* of D be defined as a function $M : \Phi \mapsto \{true, false\}$ such that $M(F) = true$ whenever there is an i-proposition in D of the form "**initially** F", and $M(F) = false$ whenever there is an i-proposition in D of the form "**initially** $\neg F$". Since D is initially consistent there exists at least one such function, and by Propositions 2.1 and 5.1 there is a one-to-one correspondence between initial assignments of D and models of D. We may therefore unambiguously refer to the model H *generated by* the initial assignment M.

Clearly, successive solutions to the sub-goals

> $Setof(l, (Initially(l)), i)$,
> $Setof(f, (Fluent(f),\ not\ Initially(Pos(f)),\ not\ Initially(Neg(f))), p)$,
> $Permutation(p, c),\ Append(c, i, m)$,

in clause (EC2) bind the variable m to an appropriate list representation of each such initial assignment in turn. Given such a ground list term M', by Lemma B.1, clause (EC6) and the definition of $Forall$, the goal

$$ConsistentWithObservations(M')$$

will succeed if and only if the model of D its corresponding initial assignment generates is consistent with each o-proposition in τ_{ob}. Hence successive solutions to the goal $IExplanation(m)$ bind the variable m to a list representation of each initial assignment which generates an i-model of D with τ_{ob}. Hence the proposition is true by clause (EC1), Lemma B.1 and the definition of $Forall$.

C. AN ORDERING PROGRAM FOR $(\Pi_\Delta, \preceq_\Delta)$

This appendix concerns the practical details of implementing Language $\mathcal{A}$ domains as Event Calculus style Prolog programs in the manner of Section 8, given the intermediate translations to $\mathcal{E}$ domain descriptions as defined in Section 3.

The following ordering program $P(\Pi_\Delta, \preceq_\Delta)$ uses the Situation Calculus style terms

$$Result(A_n, Result(\ldots, Result(A_1, S0)\ldots))$$

and

$$Branch(A', Result(A_n, Result(\ldots, Result(A_1, S0)\ldots)))$$

to represent the Δ-sequences "$A_1, \ldots, A_n$" and "$A_1, \ldots, A_n, |A'|$" respectively:

$t_1 \prec_\Delta t_2 \ \leftarrow\ ListForm(t_2, l_2),\ Append([h|r], l_1, l_2),\ ListForm(t_1, l_1).$

$t \preceq_\Delta t.$

$$t_1 \preceq_\Delta t_2 \;\leftarrow\; t_1 \prec_\Delta t_2.$$

$$\mathit{ListForm}(S0, [\,]).$$

$$\mathit{ListForm}(\mathit{Branch}(a,t), [B(a)|l]) \;\leftarrow\; \mathit{ListForm}(t,l).$$

$$\mathit{ListForm}(\mathit{Result}(a,t), [R(a), B(a)|l]) \;\leftarrow\; \mathit{ListForm}(t,l).$$

In addition, the complete occurrence set of Δ is represented by a single clause which replaces all the domain-dependent ground *HappensAt* clauses of Definition 8.2:

$$\mathit{HappensAt}(a, \mathit{Branch}(a,t)).$$

The important feature of the above ordering program is that not only does it correctly deal with ground queries of the form "$T \preceq T'$" (where T and T' are Situation Calculus style representations of Δ-sequences as described above), but it also gives all correct solutions to queries of the form "$t \preceq T'$" (where t is a variable). This enables the sub-goals in clauses (EC8)–(EC11) to be re-ordered as follows:

$$\begin{aligned}&\mathit{HoldsAt}(m, \mathit{Pos}(f), t_3) \leftarrow\\ &\quad t_1 \prec t_3,\; \mathit{HappensAt}(a,t_1),\; \mathit{Initiates}(m,a,f,t_1),\\ &\quad \mathit{not}\ \mathit{ClippedBetween}(m, t_1, \mathit{Pos}(f), t_3).\end{aligned} \tag{EC8$'$}$$

$$\begin{aligned}&\mathit{HoldsAt}(m, \mathit{Neg}(f), t_3) \leftarrow\\ &\quad t_1 \prec t_3,\; \mathit{HappensAt}(a,t_1),\; \mathit{Terminates}(m,a,f,t_1),\\ &\quad \mathit{not}\ \mathit{ClippedBetween}(m, t_1, \mathit{Neg}(f), t_3).\end{aligned} \tag{EC9$'$}$$

$$\begin{aligned}&\mathit{ClippedBetween}(m, t_1, \mathit{Pos}(f), t_3) \leftarrow\\ &\quad t_2 \prec t_3,\; t_1 \preceq t_2,\; \mathit{HappensAt}(a,t_2),\; \mathit{Terminates}(m,a,f,t_2).\end{aligned} \tag{EC10$'$}$$

$$\begin{aligned}&\mathit{ClippedBetween}(m, t_1, \mathit{Neg}(f), t_3) \leftarrow\\ &\quad t_2 \prec t_3,\; t_1 \preceq t_2,\; \mathit{Happens}(a,t_2),\; \mathit{Initiates}(m,a,f,t_2).\end{aligned} \tag{EC11$'$}$$

This re-ordering avoids problems that would otherwise arise from calls to *HappensAt* with an unground second argument.

Robotics and the Common Sense Informatic Situation

Murray Shanahan

Department of Computer Science,
Queen Mary and Westfield College,
Mile End Road,
London E1 4NS,
England.
mps@dcs.qmw.ac.uk

Abstract

Any model of the world a robot constructs on the basis of its sensor data is necessarily both incomplete, due to the robot's limited window on the world, and uncertain, due to sensor and motor noise. This paper proposes a logic-based framework in which such models are constructed through an abductive process whereby sensor data is explained by hypothesising the existence, locations, and shapes of objects. Symbols appearing in the resulting explanations acquire meaning through the theory, and yet are grounded by the robot's interaction with the world. The proposed framework draws on existing logic-based formalisms for representing action, continuous change, space, and shape. Noise is treated as a kind of non-determinism, and is dealt with by a consistency-based form of abduction.

Introduction

Without ignoring the lessons of the past, the nascent area of Cognitive Robotics [Lespérance, *et al.*, 1994] seeks to reinstate the ideals of the Shakey project [Nilsson, 1984], namely the construction of robots whose architecture is based on the idea of representing the world by sentences of formal logic and reasoning about it by manipulating those sentences. The chief benefits of this approach are,

- that it facilitates the endowment of a robot with the capacity to perform high-level reasoning tasks, such as planning, and
- that it makes it possible to formally account for the success (or otherwise) of a robot by appealing to the notions of correct reasoning and correct representation.

This paper concerns the representation of knowledge about the objects in a robot's environment, and how such knowledge is acquired. The main feature of this knowledge is its incompleteness and uncertainty, placing the robot in what McCarthy calls the *common sense informatic situation* [1989]. The treatment given in the paper is rigorously logical, but has been carried through to implementation on a real robot.

1 Assimilating Sensor Data

The key idea of this paper is to consider the process of assimilating a stream of sensor data as abduction. Given such a stream, the abductive task is to hypothesise the existence, shapes, and locations of objects which, given the output the robot has supplied to its motors, would explain that sensor data [Charniak & McDermott, 1985, page 455]. This is, in essence, the map building task for a mobile robot.

More precisely, if a stream of sensor data is represented as the conjunction Ψ of a set of observation sentences, the task is to find an explanation of Ψ in the form of a logical description (a map) Δ_M of the initial locations and shapes of a number of objects, such that,

$$\Sigma_B \wedge \Sigma_E \wedge \Delta_N \wedge \Delta_M \models \Psi$$

where,

- Σ_B is a background theory, comprising axioms for change (including continuous change), action, space, and shape,
- Σ_E is a theory relating the shapes and movements of objects (including the robot itself) to the robot's sensor data, and
- Δ_N is a logical description of the movements of objects, including the robot itself.

The exact form of these components is described in the next three sections, which present formalisms for representing and reasoning about action, change, space, and shape. In practice, as we'll see, these components will have to be split into parts for technical reasons.

Three major issues arise with this logical specification of the map building task: noisy data, incomplete information, and implementation.

- Σ_E does not have to assume a perfect correspondence between objects in the world and sensor data received from them, or a perfect correspondence between motor outputs and actual movements in the world. In practice, a noisy interface between world and robot must be assumed. Using the expressive power of first-order logic, the uncertainty resulting from such noise can be captured.

- Data in the common sense informatic situation is incomplete as well as noisy. In abductive terms, there will typically be many Δ_M's that could explain any given Ψ. For example, the robot may only receive sensor data from a small fraction of the total surface of an object, and be unable to tell whether the object is large or small. Again, using the expressive power of first-order logic, this incompleteness can be captured.
- This logical specification of the map building task must be rendered into an efficient implementation which can be executed by the on-board microprocessor of a mobile robot.

The provision of a logic-based theoretical account brings issues like noise and incompleteness into sharp focus, and permits their study within the same framework used to address wider epistemological questions in knowledge representation. It also enables the formal evaluation of algorithms for low-level motor-perception tasks by supplying a formalism in which these tasks can be precisely specified.

2 Representing Action

The formalism used in this paper to represent action and change, including continuous change, is adapted from the circumscriptive Event Calculus presented in [Shanahan, 1995b], which in turn is based loosely on the formalism of Kowalski and Sergot [1986]. However, it employs a novel solution to the frame problem, inspired by the work of Kartha and Lifschitz [1995]. The result is a considerable simplification of the formalism in [Shanahan, 1995b].

Throughout the paper, the language of many-sorted first-order predicate calculus with equality will be used, augmented with circumscription [McCarthy, 1986], [Lifschitz, 1994]. Variables in formulae begin with lower-case letters and are universally quantified with maximum scope unless indicated otherwise.

In the Event Calculus, we have sorts for fluents, actions (or events), and time points. It's assumed that time points are interpreted by the reals, and that the usual comparative predicates, arithmetic functions, and trigonometric functions are suitably defined. The formula HoldsAt(f,t) says that fluent f is true at time point t. The formulae Initiates(a,f,t) and Terminates(a,f,t) say respectively that action a makes fluent f true from time point t, and that a makes f false from t. The effects of actions are described by a collection of formulae involving Initiates and Terminates.

For example, if the term Rotate(r) denotes a robot's action of rotating r degrees about some axis passing through its body, and the term Facing(r) is a fluent representing that the robot is facing in a direction r degrees from North, then we might write the following Initiates and Terminates formulae.[1]

$$\text{Initiates(Rotate(r1),Facing(r2),t)} \leftarrow \text{HoldsAt(Facing(r3),t)} \wedge \text{r2} = \text{r3} + \text{r1} \tag{2.1}$$

$$\text{Terminates(Rotate(r1),Facing(r2),t)} \leftarrow \text{HoldsAt(Facing(r2),t)} \wedge \text{r1} \neq 0 \tag{2.2}$$

Once a fluent has been initiated or terminated by an action or event, it is subject to the common sense law of inertia, which is captured by the Event Calculus axioms to be presented shortly. This means that it retains its value (true or false) until another action or event occurs which affects that fluent.

A narrative of actions and events is described via the predicates Happens and Initially. The formula Happens(a,t) says that an action or event of type a occurred at time point t. Events are instantaneous. The formula Initially(f) says that the fluent f is true from time point 0. Here's an example narrative.

$$\text{Initially(Facing(0))} \tag{2.3}$$

$$\text{Happens(Rotate(90),10)} \tag{2.4}$$

$$\text{Happens(Rotate(}-180\text{),20)} \tag{2.5}$$

A theory will also include a pair of uniqueness-of-names axioms, one for actions and one fluents.

$$\text{UNA[Facing]} \tag{2.6}$$

$$\text{UNA[Rotate]} \tag{2.7}$$

The relationship between HoldsAt, Happens, Initiates, and Terminates is constrained by the following axioms. Note that a fluent does not hold at the time of an action or event that initiates it, but does hold at the time of an action or event that terminates it.

$$\text{HoldsAt(f,t)} \leftarrow \text{Initially(f)} \wedge \neg \text{Clipped(0,f,t)} \tag{EC1}$$

$$\text{HoldsAt(f,t2)} \leftarrow \text{Happens(a,t1)} \wedge \text{Initiates(a,f,t1)} \wedge \text{t1} < \text{t2} \wedge \neg \text{Clipped(t1,f,t2)} \tag{EC2}$$

$$\neg \text{HoldsAt(f,t2)} \leftarrow \text{Happens(a,t1)} \wedge \text{Terminates(a,f,t1)} \wedge \text{t1} < \text{t2} \wedge \neg \text{Declipped(t1,f,t2)} \tag{EC3}$$

$$\text{Clipped(t1,f,t2)} \leftrightarrow \text{Happens(a,t)} \wedge [\text{Terminates(a,f,t)} \vee \text{Releases(a,f,t)}] \wedge \text{t1} < \text{t} \wedge \text{t} < \text{t2} \tag{EC4}$$

$$\text{Declipped(t1,f,t2)} \leftrightarrow \text{Happens(a,t)} \wedge [\text{Initiates(a,f,t)} \vee \text{Releases(a,f,t)}] \wedge \text{t1} < \text{t} \wedge \text{t} < \text{t2} \tag{EC5}$$

These axioms introduce a new predicate Releases [Kartha & Lifschitz, 1994]. The formula Releases(a,f,t) says that action a exempts fluent f from the common sense law of inertia. This non-inertial status is revoked as soon as the fluent is initiated or terminated once more. The use of this predicate will be illustrated shortly in the context of continuous change.

[1] Rotation is treated as instantaneous here, and throughout the sequel.

Let the conjunction of (EC1) to (EC5) be denoted by EC. The circumscription policy to overcome the frame problem is the following. Given a conjunction of Happens and Initially formulae N, a conjunction of Initiates, Terminates and Releases formulae E, and a conjunction of uniqueness-of-names axioms U, we are interested in,

$$\begin{array}{l} \text{CIRC[N ; Happens]} \wedge \\ \quad \text{CIRC[E ; Initiates, Terminates, Releases]} \wedge \text{U} \wedge \text{EC} \end{array}$$

This formula embodies a form of the common sense law of inertia, and thereby solves the frame problem. Further details of this solution are to be found in [Shanahan, 1996]. The key to the solution is to put EC outside the scope of the circumscriptions, thus ensuring that the Hanks-McDermott problem is avoided [Hanks & McDermott, 1987]. In most cases, the two circumscriptions will yield predicate completions, making the overall formula manageable and intuitive.

For the example above, we have the following proposition. Let E be the conjunction of (2.1) with (2.2), let N be the conjunction of (2.3) to (2.5), and let U be the conjunction of (2.6) with (2.7).

Proposition 2.8.

$$\begin{array}{l} \text{CIRC[N ; Happens]} \wedge \\ \quad \text{CIRC[E ; Initiates, Terminates, Releases]} \wedge \text{U} \wedge \text{EC} \models \\ \quad\quad \text{HoldsAt(Facing(r),t)} \leftarrow \\ \quad\quad\quad [0 \leq t \leq 10 \wedge r = 0] \vee [10 < t \leq 20 \wedge r = 90] \vee \\ \quad\quad\quad\quad [20 < t \wedge r = 270]. \end{array}$$

Proof. See Appendix. □

3 Domain Constraints and Continuous Change

Two additional features of the calculus are important: the ability to represent domain constraints, and the ability to represent continuous change.

Domain constraints are straightforwardly dealt with in the proposed formalism. They are simply formulated as HoldsAt formulae with a single universally quantified time variable, and conjoined outside the scope of the circumscriptions along with EC. For example, the following domain constraint expresses the fact that the robot can only face in one direction at a time.

$$\text{HoldsAt(Facing(r1),t)} \wedge \text{HoldsAt(Facing(r2),t)} \rightarrow r1 = r2$$

In the Event Calculus, domain constraints are used to determine values for fluents that haven't been initiated or terminated by actions or events (non-inertial fluents) given the values of other fluents that have. (Domain constraints that attempt to constrain the relationship between inertial fluents can lead to inconsistency.)[1]

Following [Shanahan, 1990], continuous change is represented through the introduction of a new predicate and the addition of an extra axiom. The formula Trajectory(f1,t,f2,d) represents that, if the fluent f1 is initiated at time t, then after a period of time d the fluent f2 holds. We have the following axiom.

$$\begin{array}{l} \text{HoldsAt(f2,t2)} \leftarrow \\ \quad \text{Happens(a,t1)} \wedge \text{Initiates(a,f1,t1)} \wedge t1 < t2 \wedge \\ \quad\quad t2 = t1 + d \wedge \text{Trajectory(f1,t1,f2,d)} \wedge \\ \quad\quad\quad \neg\, \text{Clipped(t1,f1,t2)} \end{array} \qquad \text{(EC6)}$$

Let CEC denote EC $\wedge$ (EC6), and U denote the conjunction of a set of uniqueness-of-names axioms. If R is the conjunction of a set of domain constraints and T is the conjunction of set of formulae constraining Trajectory, then we are interested in,

$$\begin{array}{l} \text{CIRC[N ; Happens]} \wedge \\ \quad \text{CIRC[E ; Initiates, Terminates, Releases]} \wedge \\ \quad\quad \text{T} \wedge \text{R} \wedge \text{U} \wedge \text{CEC}. \end{array}$$

For example, suppose the robot's repertoire of actions is expanded to include the actions Go and Stop. The Go action initiates a period of continuous change in the robot's location. The Stop action terminates such a period. The robot's location will be represented by the fluent Location(Robot,p), where p is a pair of Cartesian co-ordinates the form ⟨x,y⟩. (The first argument of this fluent is there so that we can represent the locations of other objects beside the robot. This will be useful later on.) A constant velocity V is assumed in the following collection of formulae, which are intended to capture this example.

Let E be the conjunction of the following formulae.

$$\text{Initiates(Go,Moving,t)} \qquad (3.1)$$

$$\text{Releases(Go,Location(Robot,p),t)} \qquad (3.2)$$

$$\text{Terminates(Stop,Moving,t)} \qquad (3.3)$$

$$\begin{array}{l} \text{Initiates(Stop,Location(Robot,p),t)} \leftarrow \\ \quad \text{HoldsAt(Location(Robot,p),t)} \end{array} \qquad (3.4)$$

Let T be the following formula.

$$\begin{array}{l} \text{Trajectory(Moving,t,Location(Robot,}\langle x2,y2\rangle\text{),d)} \leftarrow \\ \quad \text{HoldsAt(Location(Robot,}\langle x1,y1\rangle\text{),t)} \wedge \\ \quad\quad \text{HoldsAt(Facing(r),t1)} \wedge \\ \quad\quad\quad x2 = x1 + V.d.\text{Sin}(r) \wedge y2 = y1 + V.d.\text{Cos}(r) \end{array} \qquad (3.5)$$

Let R be the following domain constraint.

$$\begin{array}{l} [\text{HoldsAt(Location(w,p1),t)} \wedge \\ \quad \text{HoldsAt(Location(w,p2),t)}] \rightarrow p1 = p2 \end{array} \qquad (3.6)$$

Let U be the conjunction of the following uniqueness-of-names axioms.

UNA[Location, Facing, Moving] (3.7)

UNA[Go, Stop] (3.8)

Let N be the following narrative description.

Initially(Location(Robot,⟨0,0⟩) (3.9)

Initially(Facing(90)) (3.10)

Happens(Go,10) (3.11)

Happens(Stop,20) (3.12)

Now, given that the circumscriptions of E and N yield the predicate completions of Happens, Initiates, Terminates, and Releases, it's a straightforward exercise to

[1] Note that Initiates(a,F1,t) $\rightarrow$ Initiates(a,F2,t) does not follow from HoldsAt(F1,t) $\rightarrow$ HoldsAt(F2,t).

show that the recommended circumscription yields what we would expect.

Proposition 3.13.

$$\begin{aligned}
&\text{CIRC}[\text{N} \,;\, \text{Happens}] \land \\
&\quad \text{CIRC}[\text{E} \,;\, \text{Initiates, Terminates, Releases}] \land \\
&\quad\quad \text{T} \land \text{R} \land \text{U} \land \text{CEC} \models \\
&\quad\quad\quad \text{HoldsAt}(\text{Location}(\text{Robot},\langle x,y\rangle),t) \leftrightarrow \\
&\quad\quad\quad\quad [0 \leq t \leq 10 \land x = 0 \land y = 0] \lor \\
&\quad\quad\quad\quad\quad [10 < t \leq 20 \land x = \text{V}.(t-10) \land y = 0] \lor \\
&\quad\quad\quad\quad\quad\quad [20 < t \land x = \text{V}.10 \land y = 0].
\end{aligned}$$

Proof. See Appendix. □

Notice that we are at liberty to include formulae which describe triggered events in N. Here's an example of such a formula, which describes conditions under which the robot will collide with a wall lying on an East-West line 100 units north of the origin.

$$\begin{aligned}
&\text{Happens}(\text{Bump},t) \leftarrow \\
&\quad \text{HoldsAt}(\text{Moving},t) \land \text{HoldsAt}(\text{Facing}(r),t) \land \\
&\quad\quad -90 < r < 90 \land \text{HoldsAt}(\text{Location}(\text{Robot},\langle x,90\rangle),t)
\end{aligned}$$

4 Representing Space and Shape

The formalism used in this paper to represent space and shape is taken from [Shanahan, 1995a]. Space is considered a real-valued co-ordinate system. For present purposes we can take space to be the plane $\mathbb{R} \times \mathbb{R}$, reflecting the fact that the robot we will consider will move only in two dimensions. A *region* is a subset of $\mathbb{R} \times \mathbb{R}$. A *point* is a member of $\mathbb{R} \times \mathbb{R}$. I will consider only interpretations in which points are interpreted as pairs of reals, in which regions are interpreted as sets of points, and in which the $\in$ predicate has its usual meaning.

A shape is represented as a region. The only shapes we will consider are open and path-connected. Every shape has a conventional centre, which is the origin $\langle 0,0\rangle$.[1] For example, an open circle of radius z units is described by following formula.

$$p \in \text{Disc}(z) \leftrightarrow \text{Distance}(p,\langle 0,0\rangle) < z \qquad \text{(Sp1)}$$

where Distance is a function yielding a positive real number, defined in the obvious way.

$$\text{Distance}(\langle x1,y1\rangle,\langle x2,y2\rangle) = \sqrt{(x1-x2)^2 + (y1-y2)^2} \qquad \text{(Sp2)}$$

The function Bearing is also useful.

$$\begin{aligned}
&\text{Bearing}(\langle x1,y1\rangle,\langle x2,y2\rangle) = r \leftarrow \qquad \text{(Sp3)} \\
&\quad z = \text{Distance}(\langle x1,y1\rangle,\langle x2,y2\rangle) \land z \neq 0 \land \\
&\quad\quad \text{Sin}(r) = \frac{x2-x1}{z} \land \text{Cos}(r) = \frac{y2-y1}{z}
\end{aligned}$$

Using Distance and Bearing we can define a straight line as follows. The term Line(p1,p2) denotes the straight line whose end points are p1 and p2. The Line function is useful in defining shapes with straight line boundaries.

$$\begin{aligned}
&p \in \text{Line}(p1,p2) \leftrightarrow \qquad \text{(Sp4)} \\
&\quad \text{Bearing}(p1,p) = \text{Bearing}(p1,p2) \land \\
&\quad\quad \text{Distance}(p1,p) \leq \text{Distance}(p1,p2)
\end{aligned}$$

Space is occupied by objects. Each object w has a unique shape denoted by the term Shape(w). If the robot is denoted by the term Robot, and if its body is circular and ten units in radius, then we can express this as follows.

$$\text{Shape}(\text{Robot}) = \text{Disc}(0{\cdot}5)$$

Spatial occupancy is represented by the fluent Occupies. The term Occupies(w,g) denotes that object w occupies region g. No object can occupy two regions at the same time. This implies, for example, that if an object occupies a region g, it doesn't occupy any subset of g nor any superset of g. We have the following domain constraints.

$$\begin{aligned}
&[\text{HoldsAt}(\text{Occupies}(w,g1),t) \land \qquad \text{(Sp5)} \\
&\quad \text{HoldsAt}(\text{Occupies}(w,g2),t)] \rightarrow g1 = g2
\end{aligned}$$

$$\begin{aligned}
&\text{HoldsAt}(\text{Occupies}(w1,g1),t) \land \qquad \text{(Sp6)} \\
&\quad \text{HoldsAt}(\text{Occupies}(w2,g2),t) \land w1 \neq w2 \rightarrow \\
&\quad\quad \neg \exists p \, [p \in g1 \land p \in g2]
\end{aligned}$$

The first of these axioms captures the uniqueness of an object's region of occupancy, and the second insists that no two objects overlap.

An object's location is represented by the fluent Location. A further domain constraint is required which relates Location to Occupies. The term Location(w,p), which we've already encountered, denotes that the object w is located at point p. This means that the region it occupies is the result of displacing the conventional centre of its shape by x units east and y units north, where $p = \langle x,y\rangle$. If the object's shape is the region g, then the result of this displacement is denoted by the term Displace(g,p).

$$\begin{aligned}
&\text{HoldsAt}(\text{Occupies}(w,\text{Displace}(g,p)),t) \leftrightarrow \qquad \text{(Sp7)} \\
&\quad \exists g \, [\text{Shape}(w,g) \land \text{HoldsAt}(\text{Location}(w,p),t)]
\end{aligned}$$

$$\langle x1,y1\rangle \in \text{Displace}(g,\langle x2,y2\rangle) \leftrightarrow \langle x1-x2,y1-y2\rangle \in g \text{(Sp8)}$$

Using the Displace function, shapes can be conveniently combined to form new shapes by taking their union (via a disjunction). The following formula defines a shape a little like the field of view through a pair of binoculars, formed from two overlapping circles.

$$\begin{aligned}
&p \in \text{TwoDiscs}(x) \leftrightarrow \\
&\quad p \in \text{Displace}(\text{Disc}(x),\langle -\tfrac{x}{2},0\rangle) \lor \\
&\quad\quad p \in \text{Displace}(\text{Disc}(x),\langle \tfrac{x}{2},0\rangle)
\end{aligned}$$

The incorporation of rotations in this formalism is extremely straightforward. In the present context, however, the only moving objects we'll encounter are circular, so the possibility of rotating a shape has been ignored.

The final component of the framework is a means of default reasoning about spatial occupancy [Shanahan, 1995a]. Shortly, a theory of continuous motion will be described. This theory insists that, in order for an object to follow a trajectory in space, that trajectory must be clear. Accordingly, as well as capturing which regions of space are occupied, our theory of space and shape must capture which regions are unoccupied.

[1] This conventional "centre" is just a reference point, and doesn't even have to be inside the shape in question.

A suitable strategy for now is to make space empty by default. It's sufficient to apply this default just to the situation at time 0 — the common sense law of inertia will effectively carry it over to later times. The following axiom is required, which can be thought of as a *common sense law of spatial occupancy.*

$$\mathrm{AbSpace}(w) \leftarrow \mathrm{Initially}(\mathrm{Location}(w,p)) \qquad \text{(Oc1)}$$

The predicate AbSpace needs to be minimised, with Initially allowed to vary.

Where previously we were interested in CIRC[N ; Happens], it's now convenient to split this circumscription into two, and to distribute Initially formulae in two places. Given,

- the conjunction O of Axioms (Sp1) to (Sp8) with Axiom (Oc1),
- a conjunction M of Initially formulae which mention only the spatial fluents Location and Occupies, and
- a conjunction N of Happens formulae and Initially formulae which don't mention the spatial fluents Location and Occupies, and
- conjunctions E, T, R, U, and CEC as described in the last section,

we are now interested in,

$$\begin{aligned}
&\mathrm{CIRC}\,[\mathrm{O} \wedge \mathrm{M}\,;\, \mathrm{AbSpace}\,;\, \mathrm{Initially}] \wedge \\
&\quad \mathrm{CIRC}[\mathrm{N}\,;\, \mathrm{Happens}] \wedge \\
&\qquad \mathrm{CIRC}[\mathrm{E}\,;\, \mathrm{Initiates, Terminates, Releases}] \wedge \\
&\qquad\quad \mathrm{T} \wedge \mathrm{R} \wedge \mathrm{U} \wedge \mathrm{CEC}.
\end{aligned}$$

5 Sensors and Motors: The Theory Σ_E

We now have the logical apparatus required to construct a formal theory of the relationship between a robot's motor activity, the world, and the robot's sensor data. For now we will assume perfect motors and perfect sensors. The issue of noise is dealt with in Section 7.

The robot used as an example throughout the rest of the paper is one of the simplest and cheapest commercially available mobile robotic platforms at the time of writing, namely the Rug Warrior described by Jones and Flynn [1993] (Figure 5a). This is a small, wheeled robot with a 68000 series microprocessor plus 32K RAM on board. It has a very simple collection of sensors. These include three bump switches arranged around its circumference, which will be our main concern here. In particular, we will confine our attention to the two forward bump switches, which, in combination, can deliver three possible values for the direction of a collision.

Needless to say, each different kind of sensor gives rise to its own particular set of problems when it comes to constructing Σ_E. The question of noise is largely irrelevant when it comes to bump sensors. With infra-red proximity detectors, noise plays a small part. With sonar, the significance of noise is much greater. The use of cameras gives rise to a whole set of issues which are beyond the scope of this paper.

The central idea of this paper is the assimilation of sensor data through abduction. This is in accordance with the principle, "prediction is deduction but explanation is abduction" [Shanahan, 1989]. To begin with, we'll be looking at the predictive capabilities of the framework described. The conjunction of our general theory of action, change, space, and shape with the theory Σ_E, along with a description of the initial locations and shapes of objects in the world and a description of the robot's actions, should yield a description of the robot's expected sensory input. If prediction works properly using deduction in this way, the reverse operation of explaining a given stream of sensor data by hypothesising the locations and shapes of objects in the world is already defined. It is simply abduction using the same logical framework.

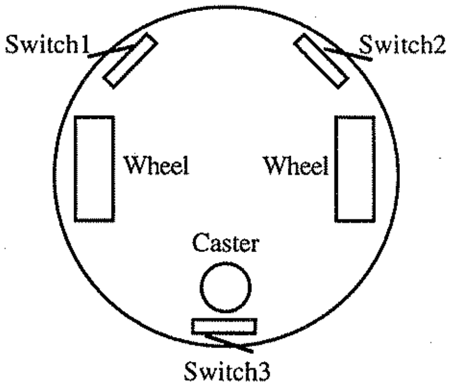


Figure 5a:
The Rug Warrior Robot from Above

In the caricature of the task of assimilating sensor data presented in Section 1, the realtionship between motor activity and sensor data was described by Σ_E. In practice, this theory is split into parts and distributed across different circumscriptions (see Section 3).

First, we have a collection of formulae which are outside the scope of any circumscription. Let B be the conjunction of CEC with Axioms (B1) to (B5) below. The robot is assumed to travel at a velocity of one unit of distance per unit of time.

$$\mathrm{UNA[Occupies, Location, Facing, Moving]} \qquad \text{(B1)}$$

$$\mathrm{UNA[Rotate, Go, Stop]} \qquad \text{(B2)}$$

$$\begin{aligned}
&\mathrm{Trajectory(Moving,t,Location(Robot,\langle x2,y2\rangle),d)} \leftarrow \qquad \text{(B3)} \\
&\quad \mathrm{HoldsAt(Location(Robot,\langle x1,y1\rangle),t)} \wedge \\
&\qquad \mathrm{HoldsAt(Facing(r),t)} \wedge \\
&\qquad\quad \mathrm{x2 = x1 + d.Sin(r)} \wedge \mathrm{y2 = y1 + d.Cos(r)}
\end{aligned}$$

$$\begin{aligned}
&\mathrm{HoldsAt(Blocked(w1,w2,r),t)} \leftrightarrow \qquad \text{(B4)} \\
&\quad \mathrm{HoldsAt(Occupies(w1,g1),t)} \wedge \\
&\qquad \mathrm{HoldsAt(Occupies(w2,g2),t)} \wedge \mathrm{w1} \neq \mathrm{w2} \wedge \\
&\qquad\quad \mathrm{HoldsAt(Location(w1,p1),t)} \wedge \\
&\qquad\qquad \neg \exists\, \mathrm{z1}\, [\mathrm{z1} > 0 \wedge \forall\, \mathrm{z2}\, [\mathrm{z2} \leq \mathrm{z1} \wedge \\
&\qquad\qquad\quad \mathrm{Bearing(p1,p2,r)} \wedge \mathrm{Distance(p1,p2,z2)} \rightarrow \\
&\qquad\qquad\qquad \neg \exists\, \mathrm{p}\, [\mathrm{p} \in \mathrm{g2} \wedge \mathrm{p} \in \mathrm{Displace(g1,p2)}]]]
\end{aligned}$$

$$\begin{array}{l}\mathrm{HoldsAt(Touching(w1,w2,p),t)} \leftrightarrow \\ \quad \mathrm{HoldsAt(Occupies(w1,g1),t)} \wedge \\ \qquad \mathrm{HoldsAt(Occupies(w2,g2),t)} \wedge \mathrm{w1} \neq \mathrm{w2} \wedge \\ \qquad\quad \exists\, \mathrm{p1, p2}\ [\mathrm{p} \in \mathrm{Line(p1,p2)} \wedge \mathrm{p} \neq \mathrm{p1} \wedge \mathrm{p} \neq \mathrm{p2} \wedge \\ \qquad\qquad \forall\, \mathrm{p3}\ [[\mathrm{p3} \in \mathrm{Line(p1,p)} \wedge \mathrm{p3} \neq \mathrm{p}] \rightarrow \\ \qquad\qquad\quad \mathrm{p3} \in \mathrm{g1}] \wedge \\ \qquad\qquad\qquad \forall\, \mathrm{p3}\ [[\mathrm{p3} \in \mathrm{Line(p,p2)} \wedge \mathrm{p3} \neq \mathrm{p}] \rightarrow \\ \qquad\qquad\qquad\quad \mathrm{p3} \in \mathrm{g2}]].\end{array} \tag{B5}$$

The fluent Blocked(w1,w2,r) holds if object w1 cannot move any distance at all in direction r without overlapping with another object. The fluent Touching(w1,w2,p) holds if w1 and w2 are touching at point p. This is true if a straight line exists from p1 to p2 at a bearing r which includes a point p3 such that every point between p1 and p3 apart from p3 itself is in g1 and every point from p2 to p3 apart from p3 itself is in g2.

Next we have a collection of Initiates, Terminates, and Releases formulae. Let E be the conjunction of the following axioms (E1) to (E6). A Bump event occurs when the robot collides with something.

$$\begin{array}{l}\mathrm{Initiates(Rotate(r1),Facing(r1+r2),t)} \leftarrow \\ \quad \mathrm{HoldsAt(Facing(r2),t)}\end{array} \tag{E1}$$

$$\begin{array}{l}\mathrm{Releases(Rotate(r1),Facing(r2),t)} \leftarrow \\ \quad \mathrm{HoldsAt(Facing(r2),t)} \wedge \mathrm{r1} \neq 0\end{array} \tag{E2}$$

$$\mathrm{Initiates(Go,Moving,t)} \tag{E3}$$

$$\mathrm{Releases(Go,Location(Robot,p),t)} \tag{E4}$$

$$\begin{array}{l}\mathrm{Terminates(a,Moving,t)} \leftarrow \\ \quad \mathrm{a = Stop} \vee \mathrm{a = Bump} \vee \mathrm{a = Rotate(r)}\end{array} \tag{E5}$$

$$\begin{array}{l}\mathrm{Initiates(a,Location(Robot,p),t)} \leftarrow \\ \quad [\mathrm{a = Stop} \vee \mathrm{a = Bump}] \wedge \\ \qquad \mathrm{HoldsAt(Location(Robot,p),t)}\end{array} \tag{E6}$$

Now we have a collection of formulae concerning the narrative of actions and events we're interested in. This collection has two parts. Let N be N1 ∧ N2. The first component part concerns triggered events. The events Switch1 and Switch2 occur when the robot's forward bump switches are tripped (see Figure 5a). Let N1 be the conjunction of Axioms (H1) to (H3) below.[1]

$$\begin{array}{l}\mathrm{Happens(Bump,t)} \leftarrow \\ \quad [\mathrm{HoldsAt(Moving,t)} \vee \mathrm{Happens(Go,t)}] \wedge \\ \qquad \mathrm{HoldsAt(Facing(r),t)} \wedge \\ \qquad\quad \mathrm{HoldsAt(Blocked(Robot,w,r),t)}\end{array} \tag{H1}$$

$$\begin{array}{l}\mathrm{Happens(Switch1,t)} \leftarrow \\ \quad \mathrm{Happens(Bump,t)} \wedge \mathrm{HoldsAt(Facing(r),t)} \wedge \\ \qquad \mathrm{HoldsAt(Location(Robot,p1),t)} \wedge \\ \qquad\quad \mathrm{HoldsAt(Touching(Robot,w,p2),t)} \wedge \\ \qquad\qquad \mathrm{r}-90 \leq \mathrm{Bearing(p1,p2)} < \mathrm{r}+12\end{array} \tag{H2}$$

$$\begin{array}{l}\mathrm{Happens(Switch2,t)} \leftarrow \\ \quad \mathrm{Happens(Bump,t)} \wedge \mathrm{HoldsAt(Facing(r),t)} \wedge \\ \qquad \mathrm{HoldsAt(Location(Robot,p1),t)} \wedge \\ \qquad\quad \mathrm{HoldsAt(Touching(Robot,w,p2),t)} \wedge \\ \qquad\qquad \mathrm{r}-12 \leq \mathrm{Bearing(p1,p2)} < \mathrm{r}+90\end{array} \tag{H3}$$

Note that Axiom (H1) caters for occasions on which the robot attempts to move when it is already blocked, as well as for occasions on which the robot's motion causes it to collide with something. In the former case, an immediate Bump event occurs, and the robot accordingly moves no distance at all.

For present purposes, the Bump event is somewhat redundant. In Axioms (E5) and (E6) it could be replaced by Switch1 and Switch2 events, and in Axioms (H2) and (H3) it could be simplified away. One reason not to abolish the Bump event is that, in principle, a collision could occur without the attendant sensor event — if one of the bump switches were broken, say. Similarly, a sensor event could occur without a collision as its cause — if a rain drop were to momentarily short a connection, for example.

Another reason is that abolishing the Bump event would violate a basic principle of the present approach, according to which the assumption of an external world governed by certain physical laws, a world to which its sensors have imperfect access, is built in to the robot. The robot's task is to do its best to explain its sensor data in terms of a model of the physics governing that world. In any such model, incoming sensor data is the end of the line, causally speaking. In the physical world, it's not a sensor event that stops the robot but a collision with a solid object.

The second component of N is a description of the robot's actions. Suppose the robot behaves as illustrated in Figure 5b. Let N2 be the conjunction of the following formulae, which represent the robot's actions up to the moment when it bumps into obstacle A.

$$\mathrm{Happens(Go,0)} \tag{5.1}$$

$$\mathrm{Happens(Stop,2{\cdot}8)} \tag{5.2}$$

$$\mathrm{Happens(Rotate(-90),3{\cdot}3)} \tag{5.3}$$

$$\mathrm{Happens(Go,3{\cdot}8)} \tag{5.4}$$

The final component of our theory is O ∧ M, where M is a map of the robot's world and O is the conjunction of Axioms (Sp1) to (Sp8) with Axiom (Oc1). Like N, M is conveniently divided into two parts. Let M be M1 ∧ M2, where M1 is a description of the initial locations, shapes, and orientations (where applicable) of known objects, including the robot itself. For the example of Figure 5b, M1 would be the conjunction of the following formulae.

$$\mathrm{Initially(Facing(80))} \tag{5.5}$$

$$\mathrm{Initially(Location(Robot,\langle 1,1\rangle))} \tag{5.6}$$

$$\mathrm{Shape(Robot) = Disc(0{\cdot}5)} \tag{5.7}$$

The form of M2 is the same as that of M1. However, when assimilating sensor data, M2 is supplied by abduction. For now though, let's look at the predictive capabilities of this framework, and supply M2 directly. Let M2 be the following formula, which describes the obstacle in Figure 5b.

$$\begin{array}{l}\mathrm{Initially(Location(A,\langle 2,4\rangle))} \wedge \\ \quad \forall\, \mathrm{x, y}\ [\langle \mathrm{x,y}\rangle \in \mathrm{Shape(A)} \leftrightarrow -1 < \mathrm{x} < 1 \wedge \\ \qquad -0{\cdot}5 < \mathrm{y} < 0{\cdot}5]\end{array} \tag{5.8}$$

[1] Both forward bump switches are tripped if the collision point is within approximately 12° of the robot's bearing.

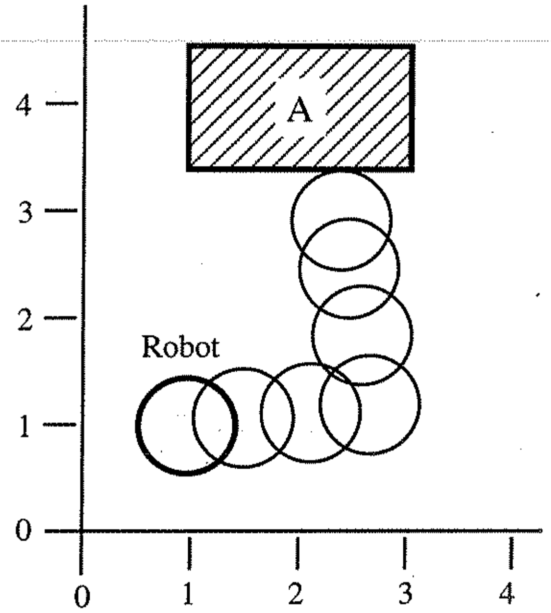


Figure 5b: A Sequence of Robot Actions

The following proposition says that, according to the formalisation, bump switch two is tripped at approximately time 5·5 (owing to a collision with obstacle A), and that the bump switches are not tripped at any other time.

Proposition 5.9.

$$\begin{aligned}&\text{CIRC}\,[\text{O} \wedge \text{M1} \wedge \text{M2}\,;\,\text{AbSpace}\,;\,\text{Initially}] \wedge\\ &\quad \text{CIRC}[\text{N1} \wedge \text{N2}\,;\,\text{Happens}] \wedge\\ &\qquad \text{CIRC}[\text{E}\,;\,\text{Initiates, Terminates, Releases}] \wedge \text{B} \vDash\\ &\qquad\quad \text{Happens}(\text{Switch1},T_{\text{bump}}) \wedge\\ &\qquad\qquad \text{Happens}(\text{Switch2},T_{\text{bump}}) \wedge\\ &\qquad\qquad\quad [[\text{Happens}(\text{Switch1},t) \vee\\ &\qquad\qquad\qquad \text{Happens}(\text{Switch2},t)] \rightarrow t = T_{\text{bump}}]\end{aligned}$$

where $T_{\text{bump}} = \dfrac{2\cdot5 + 2\cdot8.\text{Cos}(80)}{\text{Cos}(-10)} + 3\cdot8$.

Proof. See Appendix. □

The process of assimilating sensor data is the reverse of that of predicting sensor data. As outlined in Section 1, the task is to postulate the existence, location, and shape of a collection of objects which would explain the robot's sensor data, given its motor activity.[1]

Let Ψ be the conjunction of a set of formulae of the form Happens(Switch1,τ) or Happens(Switch2,τ) where τ is a time point. What we want to explain is the *partial completion* of this formula, for reasons that will be made clear shortly. The only-if half of this completion is defined as follows.

Definition 5.10.

$$\begin{aligned}&\text{COMP}[\Psi] \equiv_{\text{def}}\\ &\quad [\text{Happens}(a,t) \wedge [a = \text{Switch1} \vee a = \text{Switch2}]] \rightarrow\\ &\qquad \bigvee_{\langle\alpha,\tau\rangle \in \Gamma} [a = \alpha \wedge t = \tau]\end{aligned}$$

where $\Gamma = \{\langle\alpha,\tau\rangle \mid \text{Happens}(\alpha,\tau) \in \Psi\}$. □

Given Ψ, we're interested in finding conjunctions M2 of formulae in which each conjunct has the form,

$$\text{Initially}(\text{Location}(\omega,\rho)) \wedge \forall\, p\; [p \in \text{Shape}(\omega) \leftrightarrow \Pi]$$

where ρ is a point constant, ω is an object constant, and Π is any formula in which p is free, such that O ∧ M1 ∧ M2 is consistent and,

$$\begin{aligned}&\text{CIRC}[\text{O} \wedge \text{M1} \wedge \text{M2}\,;\,\text{AbSpace}\,;\,\text{Initially}] \wedge\\ &\quad \text{CIRC}[\text{N1} \wedge \text{N2}\,;\,\text{Happens}] \wedge\\ &\qquad \text{CIRC}[\text{E}\,;\,\text{Initiates, Terminates, Releases}] \wedge \text{B} \vDash\\ &\qquad\quad \Psi \wedge \text{COMP}[\Psi].\end{aligned}$$

The partially completed form of the Happens formula on the right-hand-side of the turnstile eliminates anomalous explanations in which, for example, the robot encounters a phantom extra obstacle before the time of the first event in Ψ. If Ψ on its own were used instead of this partially completed formula, it would be possible to construct such explanations by shifting all the obstacles that appear in a proper explanation into new positions which take account of the premature interuption in the robot's path caused by the phantom obstacle.

Clearly, from Proposition 5.9, if Ψ is,

$$\text{Happens}(\text{Switch1},T_{\text{bump}}) \wedge \text{Happens}(\text{Switch2},T_{\text{bump}})$$

then (5.8) is an explanation that meets this specification.[2] Note that the symbol A in (5.8) (or rather its computational counterpart in the actual robot), when generated through the abductive assimilation of sensor data, is *grounded* in Harnad's sense of the term [Harnad, 1990], at the same time as acquiring *meaning* through the theory. Furthermore, the theoretical framework within which such explanations are understood,

- Links the symbols that appear in them directly to a level of representation at which high-level reasoning tasks can be performed, and
- Licenses an account of the robot's success (or otherwise) at performing its tasks which appeals to the correctness of its representations and its reasoning processes.

However, (5.8) is just one among infinitely many possible explanations of this Ψ of the required form. A bizarre example of an alternative explanation would be that the whole of space was occupied by a single object with a tunnel bored in it whose shape exactly matched that of the robot's path up to time T_{bump}.

In the specification of an abductive task like this, the set of explanations of the required form will be referred to as the *hypothesis space*. It's clear, in the present case, that some constraints must be imposed on the hypothesis space to eliminate bizarre explanations. Furthermore, the set of all explanations of the suggested form for a given stream of

[1] In the present paper, it is assumed that all sensor data require explanation. To take account of glitches (as opposed to just noise), this requirement can be relaxed.

[2] It is assumed that our language includes an arbitrarily large set of unused constant symbols, from which ω is drawn.

sensor data is hard to reason about, and computing a useful representation of such a set is infeasible. This problem is tackled in the full paper by adopting a *boundary-based* representation of shape (see [Davis, 1990, Chapter 6]). Space limitations preclude further discussion of this topic here.

6 Noise

The hallmark of the common sense informatic situation for a mobile robot is incomplete and uncertain knowledge of a spatially extended world of middle-sized objects. Incompleteness is a consequence of the robot's limited window on the world, and uncertainty results from noise in its sensors and actuators. This section deals with noise.

Both noisy sensors and noisy actuators can be captured using non-determinism. (An alternative is to use probability [Bacchus, *et al.*, 1995]). Here we'll only look at the uncertainty in the robot's location that results from its noisy motors. The robot's motors are "noisy" for various reasons. For example, the two wheels might rotate at slightly different speeds when the robot is trying to travel in a straight line, or the robot might be moving on a slope or a slippery surface. Motor noise of this kind can be captured using a non-deterministic Trajectory formula, such as the following replacement for Axiom (B3).[1]

$$\begin{array}{l}\exists\, \text{x1, y1 [Trajectory(Moving,t,} \\ \quad \text{Location(Robot,}\langle\text{x1,y1}\rangle\text{),d)} \wedge \\ \quad\quad \text{Distance(}\langle\text{x1,y1}\rangle,\langle\text{x2,y2}\rangle\text{)} \leq \text{d.}\varepsilon\text{]} \leftarrow \\ \quad\quad\quad \text{HoldsAt(Location(Robot,}\langle\text{x3,y3}\rangle\text{),t)} \wedge \\ \quad\quad\quad\quad \text{HoldsAt(Facing(r),t)} \wedge \\ \quad\quad\quad\quad\quad \text{x2 = x3 + d.Sin(r)} \wedge \text{y2 = y3 + d.Cos(r)}\end{array} \qquad \text{(B6)}$$

In effect, Axiom (B6) constrains the robot's location to be within an ever-expanding *circle of uncertainty* centred on the location it would be in if its motors weren't noisy.[2] The constant ε determines the rate at which this circle grows. Axiom (B7) below ensures that there are no discontinuities in the robot's trajectory. Without this axiom the robot would be able to leap over any obstacle which didn't completely cover the circle of uncertainty for its location. The term Abs(d) denotes the absolute value of d.

$$\begin{array}{l}\text{Trajectory(f,t,Location(x,p1),d1)} \rightarrow \\ \quad \forall\, \text{z [z > 0} \rightarrow \exists\, \text{d} \;\forall\, \text{d2, p2 [[d2 > 0} \wedge \\ \quad\quad \text{Abs(d2–d1) < d} \wedge \\ \quad\quad\quad \text{Trajectory(f,t,Location(x,p2),d2)]} \rightarrow \\ \quad\quad\quad\quad \text{Distance(p1,p2) < z]]}\end{array} \qquad \text{(B7)}$$

Figure 6a shows the robot exploring the corner of an obstacle. Figure 6b shows the evolution of the corresponding circle of uncertainty, highlighting the points where the robot changes direction.

Figure 6b is somewhat misleading, however. Consider Figure 6c. On the top left, the evolution of the circle of uncertainty for the robot's location is shown. To the right, three potential locations are shown for the three changes of direction. Although these locations all fall within the relevant circles of uncertainty, the robot could never get to the third location from the second. This is because, as depicted at the bottom of the figure, in any given model the circle of uncertainty for the robot's location at the end of a period of continuous motion can only be defined relative to its actual location at the start of that period. This can be verified by inspecting Axioms (B6) and (B7).

The relative nature of the evolution of the circle of uncertainty means that the robot can acquire a detailed knowledge of some area A1 of its environment, then move to another area A2 which is some distance from A1, and acquire an equally detailed knowledge of A2. The accumulated uncertainty entails only that the robot is uncertain of where A1 is relative to A2. This natural feature of the formalisation conforms with what we would intuitively expect given the robot's informatic situation.

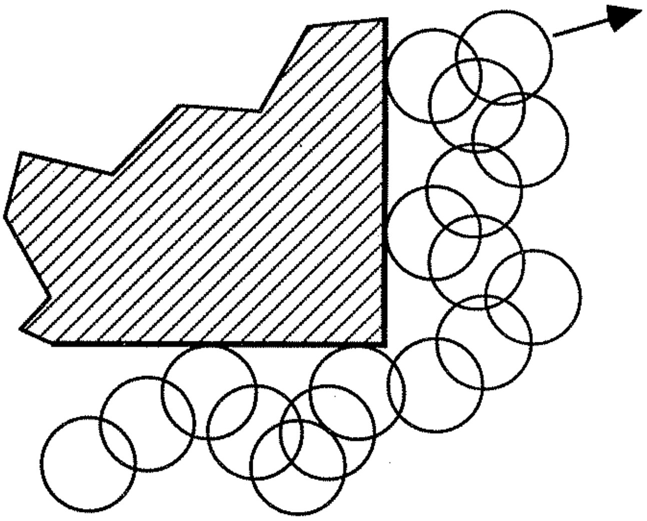


Figure 6a: The Robot Explores a Corner

Non-determinism is a potential source of difficulty for the abductive approach to explanation. Even with a precise and complete description of the initial state of the world, including all its objects and their shapes, a non-deterministic theory incorporating a formula like the one above will not yield the exact times at which collision events occur. Yet the sensor data that has to be assimilated has precise times attached to it. Fortunately we can recast the task of assimilating sensor data as a form of *weak abduction* so that it yields the required results. Intuitively what we want to capture is the fact that without the hypothesised objects, the sensor data could not have been received. This is analogous to the consistency-based approach to diagnosis proposed by Reiter [1987].

[1] The Rotate action could also be made non-deterministic.

[2] Note that, while objects occupy open subsets of $\mathbb{R}^2$, regions of uncertainty are closed.

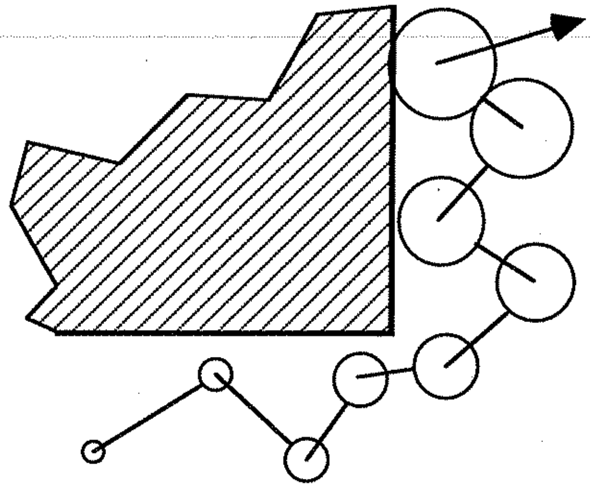


Figure 6b: The Evolution of the Circle of Uncertainty

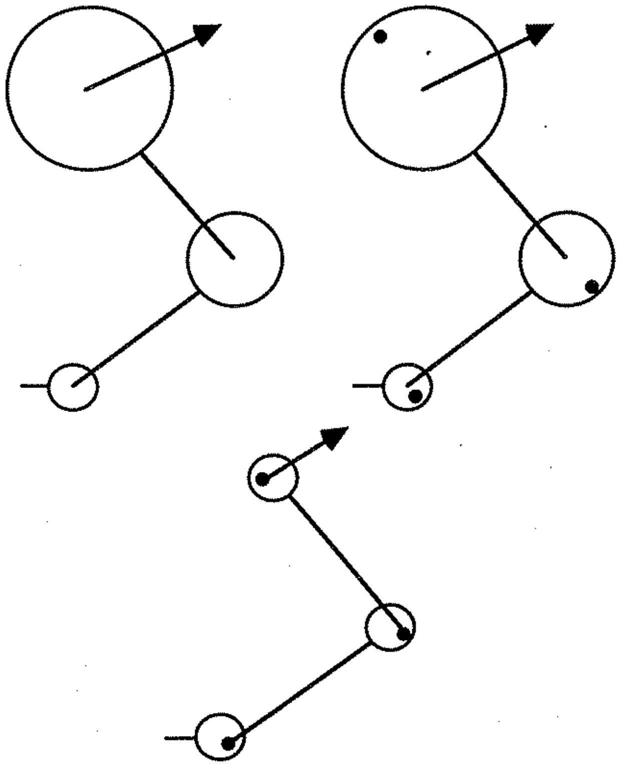


Figure 6c: The Circle of Uncertainty Is Relative Not Absolute

Definition 6.1. Given,

- the conjunction B of CEC with Axioms (B1), (B2), and (B4) to (B7),
- the conjunction E of Axioms (E1) to (E6),
- the conjunction O of Axioms (Sp1) to (Sp8) with Axiom (Oc1),
- a conjunction M1 of Initially and Shape formulae describing the initial locations, shapes, and orientations of known objects, including the robot itself,
- the conjunction N1 of Axioms (H1) to (H3),
- a conjunction N2 of Happens formulae describing the robot's actions, and
- a conjunction Ψ of formulae of the form Happens(Switch1,τ) or Happens(Switch2,τ),

an *explanation* of Ψ is a conjunction M2 of formulae in which each conjunct has the form,

$$\text{Initially}(\text{Location}(\omega,\rho)) \wedge \forall\, p\ [p \in \text{Shape}(\omega) \leftrightarrow \Pi]$$

where ρ is a point constant, ω is an object constant, and Π is any formula in which p is free, such that O ∧ M1 ∧ M2 is consistent, and,

$$\begin{array}{l} \text{CIRC}[O \wedge M1 \wedge M2 \,;\, \text{AbSpace} \,;\, \text{Initially}] \wedge \\ \quad \text{CIRC}[N1 \wedge N2 \,;\, \text{Happens}] \wedge \\ \quad\quad \text{CIRC}[E \,;\, \text{Initiates, Terminates, Releases}] \wedge B \\ \quad\quad\quad \not\models \neg\, [\Psi \wedge \text{COMP}[\Psi]]. \end{array}$$ □

There will, naturally, be many explanations for any given Ψ which meet this definition, even using the boundary-based representation of shape adopted in the full version of the paper. A standard way to treat multiple explanations in abductive knowledge assimilation is to adopt their disjunction. This has the effect of smothering any explanations which are stronger than necessary, such as those which postulate superfluous obstacles. The disjunction of all explanations of Ψ is the *cautious explanation* of Ψ.

A variety of *preference relations* over explanations can also be introduced. For example, it might be reasonable to assume that nearby collision points indicate the presence of a single object. Such preference relations are a topic for further study.

The following theorem establishes that the above definition of an explanation is equivalent to the deterministic specification offered in the last section when ε is 0. Let B_{det} be the conjunction of CEC with Axioms (B1) to (B5).

Definition 6.2. A formula M is a *complete spatial description* if the location and shape of every object mentioned in M is the same in every model of,

$$\text{CIRC}[O \wedge M \,;\, \text{AbSpace} \,;\, \text{Initially}].$$ □

Theorem 6.3. If ε = 0 and M1 is a complete spatial description, then M2 is an explanation of Ψ if and only if O ∧ M1 ∧ M2 is consistent and,

$$\begin{array}{l} \text{CIRC}[O \wedge M1 \wedge M2 \,;\, \text{AbSpace} \,;\, \text{Initially}] \wedge \\ \quad \text{CIRC}[N1 \wedge N2 \,;\, \text{Happens}] \wedge \\ \quad\quad \text{CIRC}[E \,;\, \text{Initiates, Terminates, Releases}] \wedge B_{det} \models \\ \quad\quad\quad \Psi \wedge \text{COMP}[\Psi]. \end{array}$$

Proof. See Appendix. □

A considerable amount of further work has been carried out, which is reported in the full version of the paper, but which it is only possible to present in outline here. Two further theorems have been established which characterise the abductive explanations defined above in terms which appeal more directly to the information available to any map-building algorithm which might be executed on board the robot. These theorems have been used to prove the correctness, with respect to the abductive specification given, of an algorithm for sensor data assimilation which constructs an *occupancy array* [Davis, 1990, Section 6.2.1].

This algorithm forms the core of an implementation in C, which runs on data acquired by the robot in the real world. Some preliminary experiments have been conducted in which the robot, under the control of a behaviour-based architecture [Brooks, 1986], explores an enclosure, and makes a record of its actions and sensor data for subsequent processing using the algorithm.

Concluding Remarks

In the paper accompanying his 1991 Computers and Thought Award Lecture, Brooks remarked that,

> [The field of Knowledge Representation] concentrates much of its energies on anomalies within formal systems which are never used for any practical task.
> [Brooks, 1991, page 578]

This paper should be construed as an answer to Brooks. According to the logical account given in this paper, a robot's incoming sensor data is filtered through an abductive process based on a framework of innate concepts, namely space, time, and causality. The development of a rigorous, formal account of this process bridges the gap between theoretical work in Knowledge Representation and practical work in robotics, and opens up a great many possibilities for further research. The following three issues are particularly pressing.

- The assimilation of sensor data from moving objects, such as humans, animals, or other robots. Movable obstacles should also be on the agenda.
- The assimilation of richer sensor data than that supplied by the Rug Warrior's simple bump switches.
- The control of the robot via the model of the world it acquires through abduction. Existing work in the Cognitive Robotics vein is likely to be influential here [Lespérance, *et al.*, 1994], [Kowalski, 1995], [Poole, 1995].

Acknowledgements

The inspiration for Cognitive Robotics comes from Ray Reiter and his colleagues at the University of Toronto. Thanks to Neelakantan Kartha and Rob Miller. The author is an EPSRC Advanced Research Fellow.

References

[Bacchus, *et al.*, 1995] F.Bacchus, J.Y.Halpren, and H.J.Levesque, Reasoning about Noisy Sensors in the Situation Calculus, *Proceedings IJCAI 95*, pages 1933-1940.

[Brooks, 1986] R.A.Brooks, A Robust Layered Control System for a Mobile Robot, *IEEE Journal of Robotics and Automation*, vol 2, no 1 (1986), pages 14-23.

[Brooks, 1991] R.A.Brooks, Intelligence Without Reason, *Proceedings IJCAI 91*, pages 569-595.

[Charniak & McDermott, 1985] E.Charniak and D.McDermott, *Introduction to Artificial Intelligence*, Addison-Wesley (1985).

[Davis, 1990] E.Davis, *Representations of Commonsense Knowledge*, Morgan Kaufmann (1990).

[Hanks & McDermott, 1987] S.Hanks and D.McDermott, Nonmonotonic Logic and Temporal Projection, *Artificial Intelligence*, vol 33 (1987), pages 379-412.

[Harnad, 1990] S.Harnad, The Symbol Grounding Problem, *Physica D*, vol 42 (1990), pages 335-346.

[Jones & Flynn, 1993] J.L.Jones and A.M.Flynn, *Mobile Robots: Inspiration to Implementation*, A.K.Peters (1993).

[Kartha & Lifschitz, 1994] G.N.Kartha and V.Lifschitz, Actions with Indirect Effects (Preliminary Report), *Proceedings 1994 Knowledge Representation Conference*, pages 341-350.

[Kartha & Lifschitz, 1995] G.N.Kartha and V.Lifschitz, A Simple Formalization of Actions Using Circumscription, *Proceedings IJCAI 95*, pages 1970-1975.

[Kowalski, 1995] R.A.Kowalski, Using Meta-Logic to Reconcile Reactive with Rational Agents, in *Meta-Logics and Logic Programming*, ed. K.R.Apt and F.Turini, MIT Press (1995), pages 227-242.

[Kowalski & Sergot, 1986] R.A.Kowalski and M.J.Sergot, A Logic-Based Calculus of Events, *New Generation Computing*, vol 4 (1986), pages 67-95.

[Lespérance, *et al.*, 1994] Y.Lespérance, H.J.Levesque, F.Lin, D.Marcu, R.Reiter, and R.B.Scherl, A Logical Approach to High-Level Robot Programming: A Progress Report, in *Control of the Physical World by Intelligent Systems: Papers from the 1994 AAAI Fall Syposium*, ed. B.Kuipers, New Orleans (1994), pages 79-85.

[Lifschitz, 1994] V.Lifschitz, Circumscription, in *The Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 3: Nonmonotonic Reasoning and Uncertain Reasoning*, ed. D.M.Gabbay, C.J.Hogger and J.A.Robinson, Oxford University Press (1994), pages 297-352.

[McCarthy, 1986] J.McCarthy, Applications of Circumscription to Formalizing Common Sense Knowledge, *Artificial Intelligence*, vol 26 (1986), pages 89-116.

[McCarthy, 1989] J.McCarthy, Artificial Intelligence, Logic and Formalizing Common Sense, in *Philosophical Logic and Artificial Intelligence*, ed. R.Thomason, Kluwer Academic (1989), pages ???-???.

[Nilsson, 1984] N.J.Nilsson, ed., *Shakey the Robot*, SRI Technical Note no. 323 (1984), SRI, Menlo Park, California.

[Poole, 1995] D.Poole, Logic Programming for Robot Control, *Proceedings IJCAI 95*, pages 150-157.

[Reiter, 1987] R.Reiter, A Theory of Diagnosis from First Principles, *Artificial Intelligence*, vol 32 (1987), pages 57-95.

[Shanahan, 1989] M.P.Shanahan, Prediction Is Deduction but Explanation Is Abduction, *Proceedings IJCAI 89*, pages 1055-1060.

[Shanahan, 1990] M.P.Shanahan, Representing Continuous Change in the Event Calculus, *Proceedings ECAI 90*, pages 598-603.

[Shanahan, 1995a] M.P.Shanahan, Default Reasoning about Spatial Occupancy, *Artificial Intelligence*, vol 74 (1995), pages 147-163.

[Shanahan, 1995b] M.P.Shanahan, A Circumscriptive Calculus of Events, *Artificial Intelligence*, vol 77 (1995), pages 249-284.

[Shanahan, 1996] M.P.Shanahan, *Solving the Frame Problem: A Mathematical Investigation of the Common Sense Law of Inertia*, MIT Press (1996), to appear.

Appendix A: Proofs

Proof of Proposition 2.8. From CIRC[N ; Happens], we get,

$$\begin{aligned}&\text{Happens}(a,t) \leftrightarrow && [\text{A.1}]\\ &\quad [[a = \text{Rotate}(90) \land t = 10] \lor \\ &\qquad [a = \text{Rotate}(-180) \land t = 20]].\end{aligned}$$

From CIRC[E ; Initiates, Terminates, Releases], we get,

$$\begin{aligned}&\text{Initiates}(a,f,t) \leftrightarrow && [\text{A.2}]\\ &\quad a = \text{Rotate}(r1) \land f = \text{Facing}(r2) \land \\ &\qquad \text{HoldsAt}(\text{Facing}(r3),t) \land r2 = r3 + r1\end{aligned}$$

$$\begin{aligned}&\text{Terminates}(a,f,t) \leftrightarrow && [\text{A.3}]\\ &\quad a = \text{Rotate}(r1) \land f = \text{Facing}(r2) \land \\ &\qquad \text{HoldsAt}(\text{Facing}(r2),t) \land r1 \neq 0\end{aligned}$$

$$\neg \exists\, a,f,t\ [\text{Releases}(a,f,t)]. \qquad [\text{A.4}]$$

From [A.1] and (EC4), we get,

$$\neg\, \text{Clipped}(0,\text{Facing}(0),t) \leftarrow 0 \leq t \leq 10$$

which, given (EC1) and (2.3), yields,

$$\text{HoldsAt}(\text{Facing}(0),t) \leftarrow 0 \leq t \leq 10. \qquad [\text{A.5}]$$

From [A.2] and [A.5], we get,

$$\text{Initiates}(\text{Rotate}(90),\text{Facing}(90),10). \qquad [\text{A.6}]$$

From [A.1] and (EC4), we get,

$$\neg\, \text{Clipped}(10,\text{Facing}(90),t) \leftarrow 10 < t \leq 20$$

which, given [A.1], [A.6] and (EC2), yields,

$$\text{HoldsAt}(\text{Facing}(90),t) \leftarrow 0 \leq t \leq 10. \qquad [\text{A.7}]$$

From [A.2] and [A.5], we get,

$$\text{Initiates}(\text{Rotate}(-180),\text{Facing}(270),20). \qquad [\text{A.8}]$$

From [A.1] and (EC4), we get,

$$\neg\, \text{Clipped}(20,\text{Facing}(270),t) \leftarrow 20 < t$$

which, given [A.1], [A.8] and (EC2), yields,

$$\text{HoldsAt}(\text{Facing}(270),t) \leftarrow 20 < t. \qquad [\text{A.9}]$$

The proposition follows from [A.5], [A.7], and [A.9]. □

Proof of Proposition 3.13. From CIRC[N ; Happens], we get,

$$\begin{aligned}&\text{Happens}(a,t) \leftrightarrow && [\text{A.10}]\\ &\quad [[a = \text{Go} \land t = 10] \lor [a = \text{Stop} \land t = 20]].\end{aligned}$$

From CIRC[E ; Initiates, Terminates, Releases], we get,

$$\begin{aligned}&\text{Initiates}(a,f,t) \leftrightarrow && [\text{A.11}]\\ &\quad [a = \text{Go} \land f = \text{Moving}] \lor \\ &\qquad [a = \text{Stop} \land f = \text{Location}(\text{Robot},p) \land \\ &\qquad\quad \text{HoldsAt}(\text{Location}(\text{Robot},p),t)]\end{aligned}$$

$$\text{Terminates}(a,f,t) \leftrightarrow a = \text{Stop} \land f = \text{Moving} \qquad [\text{A.12}]$$

$$\begin{aligned}&\text{Releases}(a,f,t) \leftrightarrow && [\text{A.13}]\\ &\quad a = \text{Go} \land f = \text{Location}(\text{Robot},p).\end{aligned}$$

From [A.10] and (EC4), we get,

$$\neg\, \text{Clipped}(0,\text{Location}(\text{Robot},\langle 0,0\rangle),t) \leftarrow 0 \leq t \leq 10$$

which, from (EC1) and (3.9), yields,

$$\text{HoldsAt}(\text{Location}(\text{Robot},\langle 0,0\rangle),t) \leftarrow 0 \leq t \leq 10. \qquad [\text{A.14}]$$

Similarly, we can show,

$$\text{HoldsAt}(\text{Facing}(90),10). \qquad [\text{A.15}]$$

From [A.11] we have,

$$\text{Initiates}(\text{Go},\text{Moving},10). \qquad [\text{A.16}]$$

From [A.10] and (EC4), we get,

$$\neg\, \text{Clipped}(10,\text{Moving},t) \leftarrow 10 < t \leq 20. \qquad [\text{A.17}]$$

From [A.14], [A.15], and (3.5), we get,

$$\begin{aligned}&\text{Trajectory}(\text{Moving},10,\text{Location}(\text{Robot},\langle x,0\rangle),d) \leftarrow \\ &\quad x = V.d\end{aligned}$$

which, given [A.10], [A.16], [A.17] and (EC6), yields,

$$\begin{aligned}&\text{HoldsAt}(\text{Location}(\text{Robot},\langle x,0\rangle),t) \leftarrow && [\text{A.18}]\\ &\quad 10 < t \leq 20 \land x = V.(t-10).\end{aligned}$$

From [A.11] and [A.18], we have,

$$\begin{aligned}&\text{Initiates}(\text{Stop},\text{Location}(\text{Robot},\langle x,0\rangle),20) \leftarrow && [\text{A.19}]\\ &\quad x = V.10.\end{aligned}$$

From [A.10] and (EC4), we get,

$$\begin{aligned}&\neg\, \text{Clipped}(20,\text{Location}(\text{Robot},\langle x,0\rangle),t) \leftarrow \\ &\quad 20 < t \land x = V.10\end{aligned}$$

which, given [A.10], [A.18], [A.19] and (EC2), yields,

$$\begin{aligned}&\text{HoldsAt}(\text{Location}(\text{Robot},\langle x,0\rangle),t) \leftarrow && [\text{A.20}]\\ &\quad 20 < t \land x = V.10.\end{aligned}$$

From [A.14], [A.18] and [A.20], we arrive at,

$$\begin{aligned}&\text{HoldsAt}(\text{Location}(\text{Robot},\langle x,y\rangle),t) \leftarrow \\ &\quad [0 \leq t \leq 10 \land x = 0 \land y = 0] \lor \\ &\qquad [10 < t \leq 20 \land x = V.(t-10) \land y = 0] \lor \\ &\qquad\quad [20 < t \land x = V.10 \land y = 0].\end{aligned}$$

The proposition follows from this and the domain constraint (3.6). □

Proof of Proposition 5.9. From CIRC[N1 $\land$ N2 ; Happens], we get,

$$\begin{aligned}
&\mathrm{Happens}(a,t) \leftrightarrow && [A.21]\\
&\quad \mathrm{H1}(a,t) \lor \mathrm{H2}(a,t) \lor \mathrm{H3}(a,t) \lor \mathrm{H4}(a,t)
\end{aligned}$$

where,

$$\begin{aligned}
&\mathrm{H1}(a,t) \equiv_{def}\\
&\quad [a = \mathrm{Go} \land t = 0] \lor [a = \mathrm{Stop} \land t = 2{\cdot}8] \lor\\
&\qquad [a = \mathrm{Rotate}(-90) \land t = 3{\cdot}3] \lor [a = \mathrm{Go} \land t = 3{\cdot}8]\\
&\mathrm{H2}(a,t) \equiv_{def}\\
&\quad \exists\, w,r\ [a = \mathrm{Bump} \land [\mathrm{HoldsAt}(\mathrm{Moving},t) \lor t = 3{\cdot}8] \land\\
&\qquad \mathrm{HoldsAt}(\mathrm{Facing}(r),t) \land\\
&\qquad\quad \mathrm{HoldsAt}(\mathrm{Blocked}(\mathrm{Robot},w,r),t)]\\
&\mathrm{H3}(a,t) \equiv_{def}\\
&\quad \exists\, w,r\ [a = \mathrm{Switch1} \land [\mathrm{HoldsAt}(\mathrm{Moving},t) \lor t = 3{\cdot}8] \land\\
&\qquad \mathrm{HoldsAt}(\mathrm{Facing}(r),t) \land\\
&\qquad\quad \mathrm{HoldsAt}(\mathrm{Location}(\mathrm{Robot},p1),t) \land\\
&\qquad\qquad \mathrm{HoldsAt}(\mathrm{Touching}(\mathrm{Robot},w,p2),t) \land\\
&\qquad\qquad\quad r{-}150 < \mathrm{Bearing}(p1,p2) < r{+}30]\\
&\mathrm{H4}(a,t) \equiv_{def}\\
&\quad \exists\, w,r\ [a = \mathrm{Switch2} \land [\mathrm{HoldsAt}(\mathrm{Moving},t) \lor t = 3{\cdot}8] \land\\
&\qquad \mathrm{HoldsAt}(\mathrm{Facing}(r),t) \land\\
&\qquad\quad \mathrm{HoldsAt}(\mathrm{Location}(\mathrm{Robot},p1),t) \land\\
&\qquad\qquad \mathrm{HoldsAt}(\mathrm{Touching}(\mathrm{Robot},w,p2),t) \land\\
&\qquad\qquad\quad r{-}30 < \mathrm{Bearing}(p1,p2) < r{+}150].
\end{aligned}$$

From CIRC[E ; Initiates, Terminates, Releases], we get,

$$\begin{aligned}
&\mathrm{Initiates}(a,f,t) \leftrightarrow && [A.22]\\
&\quad [a = \mathrm{Rotate}(r1) \land f = \mathrm{Facing}(r1{+}r2) \land\\
&\qquad \mathrm{HoldsAt}(\mathrm{Facing}(r2),t)] \lor [a = \mathrm{Go} \land f = \mathrm{Moving}] \lor\\
&\qquad\quad [[a = \mathrm{Stop} \lor a = \mathrm{Bump}(r)] \land\\
&\qquad\qquad f = \mathrm{Location}(\mathrm{Robot},p) \land\\
&\qquad\qquad\quad \mathrm{HoldsAt}(\mathrm{Location}(\mathrm{Robot},p),t)]
\end{aligned}$$

$$\begin{aligned}
&\mathrm{Terminates}(a,f,t) \leftrightarrow && [A.23]\\
&\quad [a = \mathrm{Stop} \lor a = \mathrm{Bump}(r) \lor a = \mathrm{Rotate}(r)] \land\\
&\qquad f = \mathrm{Moving}
\end{aligned}$$

$$\begin{aligned}
&\mathrm{Releases}(a,f,t) \leftrightarrow && [A.24]\\
&\quad a = \mathrm{Go} \land f = \mathrm{Location}(\mathrm{Robot},p).
\end{aligned}$$

From CIRC [O $\land$ M1 $\land$ M2 ; AbSpace ; Initially] we get,

$$\mathrm{Initially}(\mathrm{Location}(x,p)) \rightarrow x = A \lor x = \mathrm{Robot}. \qquad [A.25]$$

It can easily be shown that A retains its initial location for all time. Let $X_{turn} = 1 + 2{\cdot}8.\mathrm{Sin}(80)$, $Y_{turn} = 1 + 2{\cdot}8.\mathrm{Cos}(80)$. From [A.25] and (5.8), using Axioms (Sp7), (Sp8) and (B4), it can be confirmed that,

$$\begin{aligned}
&\neg\, \mathrm{HoldsAt}(\mathrm{Blocked}(\mathrm{Robot},w,r),t) \leftarrow && [A.26]\\
&\quad \mathrm{HoldsAt}(\mathrm{Location}(\mathrm{Robot},\langle x,y\rangle),t) \land\\
&\qquad \exists\, d\ [[0 \le d \le 2{\cdot}8 \land x = 1 + d.\mathrm{Sin}(80) \land\\
&\qquad\quad y = 1 + d.\mathrm{Cos}(80)] \lor\\
&\qquad\qquad [3{\cdot}8 < d < T_{bump} \land\\
&\qquad\qquad\quad x = X_{turn} + (d - 3{\cdot}8).\mathrm{Sin}(-10) \land\\
&\qquad\qquad\qquad y = Y_{turn} + (d - 3{\cdot}8).\mathrm{Cos}(-10)]].
\end{aligned}$$

It can similarly be confirmed that,

$$\begin{aligned}
&\neg\, \mathrm{HoldsAt}(\mathrm{Touching}(\mathrm{Robot},w,p),t) \leftarrow && [A.27]\\
&\quad \mathrm{HoldsAt}(\mathrm{Location}(\mathrm{Robot},\langle x,y\rangle),t) \land\\
&\qquad \exists\, d\ [[0 \le d \le 2{\cdot}8 \land x = 1 + d.\mathrm{Sin}(80) \land\\
&\qquad\quad y = 1 + d.\mathrm{Cos}(80)] \lor\\
&\qquad\qquad [3{\cdot}8 < d < T_{bump} \land\\
&\qquad\qquad\quad x = X_{turn} + (d - 3{\cdot}8).\mathrm{Sin}(-10) \land\\
&\qquad\qquad\qquad y = Y_{turn} + (d - 3{\cdot}8).\mathrm{Cos}(-10)]].
\end{aligned}$$

Given [A.26], from [A.21] to [A.24], using a similar procedure to that employed in the proof of Proposition 3.13, we can show,

$$\begin{aligned}
&\mathrm{HoldsAt}(\mathrm{Location}(\mathrm{Robot},\langle x,y\rangle),t) \leftarrow && [A.28]\\
&\quad [0 \le t \le 2{\cdot}8 \land x = 1 + t.\mathrm{Sin}(80) \land\\
&\qquad y = 1 + t.\mathrm{Cos}(80)] \lor\\
&\qquad\quad [2{\cdot}8 < t \le 3.8 \land x = X_{turn} \land y = Y_{turn}] \lor\\
&\qquad\qquad [3{\cdot}8 < t \le T_{bump} \land\\
&\qquad\qquad\quad x = X_{turn} + (t - 3{\cdot}8).\mathrm{Sin}(-10) \land\\
&\qquad\qquad\qquad y = Y_{turn} + (t - 3{\cdot}8).\mathrm{Cos}(-10)].
\end{aligned}$$

Given that A retains its initial location, from [A.28], [A.25] and (5.8), using Axioms (Sp7) and (Sp8), we can show,

$$\mathrm{HoldsAt}(\mathrm{Blocked}(\mathrm{Robot},A,-10),T_{bump}). \qquad [A.29]$$

We can also show,

$$\mathrm{HoldsAt}(\mathrm{Facing}(-10),T_{bump}). \qquad [A.30]$$

From [A.29] and [A.30], using Axiom (B4), we get,

$$\mathrm{Happens}(\mathrm{Bump},T_{bump}). \qquad [A.31]$$

Given that A retains its initial location, from [A.28], [A.25] and (5.8), using Axioms (Sp7) and (Sp8), we can show,

$$\begin{aligned}
&\exists\, p1, p2\ [&& [A.32]\\
&\quad \mathrm{HoldsAt}(\mathrm{Touching}(\mathrm{Robot},A,p1),T_{bump}) \land\\
&\qquad \mathrm{HoldsAt}(\mathrm{Location}(\mathrm{Robot},p2),T_{bump}) \land\\
&\qquad\quad \mathrm{Bearing}(p1,p2) = 0].
\end{aligned}$$

From [A.21] and [A.30] to [A.32] we get,

$$\begin{aligned}
&\mathrm{Happens}(\mathrm{Switch1},T_{bump}) \land && [A.33]\\
&\quad \mathrm{Happens}(\mathrm{Switch2},T_{bump}).
\end{aligned}$$

From [A.21], [A.27] and Axiom (B5) we get,

$$\begin{aligned}
&[\mathrm{Happens}(\mathrm{Switch1},t) \lor \mathrm{Happens}(\mathrm{Switch2},t)] \rightarrow && [A.34]\\
&\quad t = T_{bump}.
\end{aligned}$$

The proposition follows directly from [A.33] and [A.34]. □

Proof of Theorem 7.5. We only need to consider Ψ since the definition of an explanation caters for COMP[Ψ] automatically. The theorem follows from the fact that Axioms (B3) and (B5) are equivalent if ε is 0, and the fact that (B3) ensures that the robot's path is deterministic in the sense that at any given time its location is the same in every model of,

$$\begin{aligned}
&\mathrm{CIRC}[O \land M1 \land M2 ;\ \mathrm{AbSpace} ;\ \mathrm{Initially}] \land\\
&\quad \mathrm{CIRC}[N1 \land N2 ;\ \mathrm{Happens}] \land\\
&\qquad \mathrm{CIRC}[E ;\ \mathrm{Initiates, Terminates, Releases}] \land B_{det}.
\end{aligned}$$

To see that the theorem follows, consider that, if the robot's path is deterministic according to a formula Γ and the locations and shapes of objects are the same in every model of Γ (as they must be in the above formula since M1 and M2 are complete spatial descriptions), then $\Gamma \nvDash \neg\, \Psi$ if and only if $\Gamma \vDash \Psi$. □

Safety Logics I: Absolute Safety

Zhisheng Huang and John Bell*
Applied Logic Group
Department of Computer Science
Queen Mary and Westfield College
University of London
email: {huang, jb}@dcs.qmw.ac.uk

Abstract

In this paper we distinguish between absolute safety and normative safety, and develop a formal theory of absolute safety. We start with the idea of a *disastrous state* (a state in which some disastrous fact is true) and define a *disastrous action* to be an action which always leads to a disastrous state. We then define *dangerous actions* to be actions which may lead to disastrous states, and *dangerous states* to be states in which every possible action is dangerous. Finally, *safe actions* (*states*) are defined to be actions (states) which are not dangerous. Using Dynamic Logic as a basis, we develop a formal possible-worlds semantics for our theory of safety, and give the basic safety logic **SL** and several extensions of it. We then give an example of reasoning about safety in nuclear power stations, and conclude with an indication of how we are building on this work to produce a formal account of normative safety.

1 Introduction

The task of planning for rational agents is to find plans which will achieve their goals, given an appropriate theory of action and change. In safety-critical systems, agents have to consider not only how they can achieve their goals, they also have to consider how they can achieve them safely. One solution is constraint-based planning, however the constraints have to be *specified* in advance, and there is no principled attempt to *reason* about them. As a result, the limitations of this approach will make themselves felt as the the application domains increase in complexity. The alternative is to develop formal logics which agents can reason about safety. This is a complex task, as reasoning about safety may involve commonsense reasoning about actions and change, time, beliefs, goals, rationality, etc. We begin the task in this paper by abstracting away from many of these features, and developing a formal theory of what we consider to be the essential features of safety, and which we will subsequently extend.

We start with disasters and the idea of a *disastrous state*. A disastrous state is one in which some fact, which the agent considers to be disastrous, is true. It is assumed that such states are abhorrent to the agent and that the agent tries to avoid them if at all

*This research forms part of the RED project, *Decision Support Systems, Safety and Liability*, and is supported by the United Kingdom Engineering and Physical Sciences Research Council under grant number GR/H 12348.

possible. For instance, a typical medical safety-critical system would consider states in which the patient dies as a result of treatment to be disastrous, and would try to avoid them at all costs. We further assume that the application domain is such that there is no question as to which states the agent considers to be disastrous. Of course, what counts as a disastrous state may be context-dependent. For example, the agent might consider a state in which it has missed a plane to be disastrous if it has a non-transferrable ticket, but it need not do so if there is another flight and the ticket is transferrable. However, in some domains, disasters may not by context dependent. We will refer to disastrous states of this kind as *stable* disastrous states. However, in most applications disasters are *unstable*. Furthermore, as will become clear in our formal analysis, stable disastrous states can be viewed as a special case. Consequently, when talking about disastrous states we will usually take them to be unstable. A *disastrous action* can now be defined to be an action which *always* leads to a disastrous state.

We then define a *dangerous action* to be an action which *may* lead to a disastrous state, and a *dangerous state* to be state in which every action which the agent can perform – every action which is open to the agent – is dangerous. Note that disastrous actions (states) are thus dangerous actions (states).

Finally, we define a *safe action* (a *safe state*) to be an action (a state) which is not dangerous.

As an example of these definitions consider an agent who is learning to swim, and suppose that the agent is alone in the pool. The agent's goal is to enjoy its swim, and, naturally, the agent considers that its death by drowning is a disaster; that is, the agent takes states in which it has drowned to be disastrous states. The agent enjoys swimming, regardless of whether it is swimming in shallow or deep water. Swimming is shallow water is safe; the agent is never in danger of drowning because it can always stand up. However, if the agent is in deep water this action is no longer open to it and all its attempts to stay afloat amount to helpless floundering. Consequently it is in danger of drowning no matter what action it takes, and will do so unless someone comes to its aid. Swimming in deep water is therefore dangerous (or unsafe) for the agent. Note that being in deep water is only unsafe because *all* of the actions open to the agent are dangerous. If, for example, the agent could capture a buoy whenever it wanted – that is, if a safe action were available to the agent – then swimming in the deep end would be safe.

We have defined an *absolute* notion of safety, in which an action is safe only if there is absolutely no possibility of it leading to a dangerous state. In practice, safety is a *normative* concept. That is, in everyday life almost all actions are dangerous, as there is always *some* (perhaps remote) possibility that the action will lead to a disastrous state. So in practice we are interested in which actions are *normally* safe; that is, in actions which do not normally lead to disastrous states. For example, it is not absolutely safe to fly by major airlines (because, there is always the possibility, however remote, of a crash) but it is normally safe to do so (as flights do not normally end in disaster). However, it is useful to start with the formalisation of absolute safety for two reasons. Firstly, the formalisation of reasoning about absolute safety is central to the development of safety-critical systems. Systems of this kind involve domains in which many of the dangers of daily life are removed, or are not considered to be possibilities. So it makes sense to define (absolutely) safe actions and states, and then to aim to maintain safety by trying to ensure that only safe actions are performed. The second reason for starting with absolute safety is methodological: the formalisation of absolute safety provides a conceptual and theoretical basis on which further work, for example on the formalization of normative safety can build.

We have chosen to formalize absolute safety in Dynamic Logic [3], as this offers a powerful tool for formalizing actions using classical logic and Kripke semantics. In particular, states are represented as possible worlds, and actions are represented as accessibility relations between worlds. The crucial distinction between the necessary and the possible consequences of actions in our informal analysis is thus readily and naturally formalized. In the next section we define $\mathcal{SL}$, our language of safety, and give formal semantics for it. We then turn to logics of safety and give our basic safety logic **SL** and discuss several extensions of it. An example of reasoning about safe formulas in nuclear power stations is then given. Finally, we give a brief indication of ongoing work on formalising normative safety and of how this builds on the present work.

2 The Language $\mathcal{SL}$

Let Φ_0 be a set of primitive propositions, and PA be a set of primitive actions. We will use the Roman letters $a, b, \ldots$ (with or without subscripts or superscripts) to denote actions, and lower case Greek letters $\phi, \psi, \ldots$ (with or without subscripts or superscripts) to denote formulas.

Definition 1 (Actions) *The set of (composite) actions ACTION is, as usual, defined to be the smallest set which is closed under the following syntactic rules:*

- *If $a \in PA$ then $a \in ACTION$*
- *If $a, b \in ACTION$ then $(a \cup b), (a; b) \in ACTION$*

Here $\cup$ is the non-deterministic choice operator (so $a \cup b$ means "do either a or b non-deterministically"), and ';' is the sequence operator (so $a; b$ means "do a and then do b"). The definition can be extended to include further action operators, such as the "Kleene star", '$*$', and the '**skip**' operator. However, their inclusion is not required for present purposes, and is thus left for future work.

Definition 2 (The language $\mathcal{SL}$) *The language $\mathcal{SL}$ is the minimal set of formulas which satisfies the following conditions:*

- *If $p \in \Phi_0$ then $p \in \mathcal{SL}$.*
- *If $\phi, \psi \in \mathcal{SL}$ then $\neg\phi \in \mathcal{SL}$ and $\phi \wedge \psi \in \mathcal{SL}$.*
- *If $a \in ACTION$ and $\phi \in \mathcal{SL}$ then $\langle a \rangle \phi \in \mathcal{SL}$.*
- *If $\phi \in \mathcal{SL}$ then* $\mathbf{Dis}\phi \in \mathcal{SL}$ *and* $\mathbf{Dan}\phi \in \mathcal{SL}$*.*
- *If $a \in ACTION$ then* $\mathbf{Dis}(a) \in \mathcal{SL}$ *and* $\mathbf{Dan}(a) \in \mathcal{SL}$*.*

Informally, $\langle a \rangle \phi$ states that doing a makes ϕ possible, that is ϕ *may* be true after action a is taken. Then $[a]\phi$ is defined as $\neg\langle a \rangle \neg\phi$. Thus $[a]\phi$ states that doing a makes ϕ inevitable; that is, ϕ *will* be true after action a is taken. A sentence of the form $\mathbf{Dis}\phi$ ($\mathbf{Dan}\phi$) means that ϕ is disastrous (dangerous). Similarly $\mathbf{Dis}(a)$ ($\mathbf{Dan}(a)$) means that action a is disastrous (dangerous). The safety operators are then introduced by definition as follows:

- **Safe**$(a) \overset{\text{def}}{\Longleftrightarrow} \neg$**Dan**$(a)$
- **Safe**$\phi \overset{\text{def}}{\Longleftrightarrow} \neg$**Dan**$\phi$

The falsum '$\perp$' and the remaining Boolean connectives '$\vee$', '$\rightarrow$' and '$\leftrightarrow$' are defined using the negation and conjunction operators in the usual way.

Recall that our analysis of safety started with disastrous states. We therefore extend the standard Kripke models of propositional Dynamic Logic by adding the component D in order to represent disastrous states. In the simplest case D can just be a set of worlds. This would give a model with *stable* disastrous states; that is, it would give a model in which what counts as a disaster is independent of any particular world. However, as the airline-ticket example illustrated, disasters are typically unstable. Consequently, we prefer to model unstable disasters, and to treat stable disasters as a special case; see Section 5.

Definition 3 ($\mathcal{SL}$-models) *A* model for $\mathcal{SL}$ *is a tuple* $M = \langle W, \{R^a\}_{a \in PA}, D, V \rangle$, *where*

- W *is a set of possible worlds,*
- $R^a \subseteq W \times W$ *is a binary accessibility relation for each primitive action* $a \in PA$,
- $D : W \rightarrow \mathcal{P}(W)$ *assigns to each world* w *the set* $D(w)$ *of worlds which are disastrous with respect to* w, *and*
- $V : \Phi_0 \rightarrow \mathcal{P}(W)$ *is the usual valuation function.*

As usual, we extend the accessibility relations for primitive actions to complex actions as follows:

- $R^a_+ = R^a$ *(where a is primitive)*
- $R^{a \cup b}_+ = R^a_+ \cup R^b_+$
- $R^{a;b}_+ = R^a_+ \circ R^b_+$ *(composition of relations)*

We will usually write R^a *rather than* R^a_+ *when there is no danger of ambiguity.*

Finally, we require that the global accessibility relation $R = \{\langle w, w' \rangle : wR^a_+w'$ *for some* $a\}$ *is a chronology – that is, that* R *is a partial order which is required to be backwards-linear (or anti-convergent).*

Intuitively wR^aw' means that world w' is one possible outcome of doing action a in world w.

The states of our informal analysis are thus represented by worlds, and actions are represented by accessibility relations amongst them. In order to capture the idea of the consequences of an action *leading to* a disastrous state, we define the *trace* of action each a from each world w. The trace of a from w, is a tree rooted at w each branch of which represents a possible sequence of events resulting from doing a at w; the trace of a from w is thus the tree of all states that doing a at w can lead to. Thus a leads to a disastrous world (state) if there is such a world on some branch of the trace of a from w. Formally, let $R^aw = \{w' \in W : \langle w, w' \rangle \in R^a_+\}$ be the set of all possible outcomes of doing the (possibly compound) action a at w. Then $Trace(a, w)$ is defined as follows.

- $Trace(a, w) = \{\langle w, w\rangle\} \cup \{\langle w, w'\rangle : w' \in R^a w\}$ where a is primitive
- $Trace((a \cup b), w) = Trace(a, w) \cup Trace(b, w)$
- $Trace((a; b), w) = Trace(a, w) \cup \bigcup_{w' \in R^a w} Trace(b, w') \cup$ $(Trace(a, w) \circ \bigcup_{w' \in R^a w} Trace(b, w'))$

For brevity's sake we will write $w_1 R^a w w_2$ if $\langle w_1, w_2\rangle \in Trace(a, w)$; thus, $w_1 R^a w w_2$ states that w_1 occurs no later than w_2 on a branch in the trace of a from w.

Definition 4 (Truth conditions for $\mathcal{SL}$) *Let $M = \langle W, \{R^a\}_{a \in PA}, D, V\rangle$ be an $\mathcal{SL}$-model. Then a sentence ϕ is true at a world w in M (written $M, w \models \phi$, or, equivalently, $w \in [\![\phi]\!]^M$) as follows.*

$$\begin{array}{lll}
M, w \models p & \textit{iff} & w \in V(p) \textit{ where } p \textit{ primitive} \\
M, w \models \neg\psi & \textit{iff} & M, w \not\models \psi \\
M, w \models \psi \wedge \chi & \textit{iff} & M, w \models \psi \textit{ and } M, w \models \chi \\
M, w \models \langle a\rangle\psi & \textit{iff} & \exists w'(w' \in R^a w \textit{ and } w' \in [\![\psi]\!]^M) \\
M, w \models \mathbf{Dis}\psi & \textit{iff} & [\![\psi]\!]^M \subseteq D(w) \\
M, w \models \mathbf{Dis}(a) & \textit{iff} & \forall w_1(w R^a w w_1 \Rightarrow \\
 & & \exists w_2((w_1 R^a w w_2 \textit{ or } w_2 R^a w w_1) \textit{ and } w_2 \in D(w))) \\
M, w \models \mathbf{Dan}(a) & \textit{iff} & \exists w'(w R^a w w' \textit{ and } w' \in D(w)) \\
M, w \models \mathbf{Dan}\psi & \textit{iff} & \forall w_1(w_1 \in [\![\psi]\!]^M \Rightarrow \forall a(R^a w_1 \neq \emptyset \Rightarrow \\
 & & \exists w_2(w_1 R^a w_1 w_2 \textit{ and } w_2 \in D(w))))
\end{array}$$

As usual, a sentence ϕ is said to be true in a model M (written $M \models \phi$) if $M, w \models \phi$ for all worlds w in M, and ϕ is said to be valid (written $\models \phi$) if ϕ is true in all models.

The truth condition for dangerous states is a bit tricky. The condition $R^a w_1 \neq \emptyset$ means that action a *can* be taken (is open) at world w_1; a new CAN operator is therefore not necessary. Let $Rw = \{a \in ACTION : R^a w \neq \emptyset\}$ be the set of actions which are open to the agent at w. For a set of possible worlds X, we define the *zone* of X as follows:

$Zone(X) = \{w : \forall a \in Rw \exists w'(w R^a w w' \text{ and } w' \in X)\};$

thus $w \in Zone(X)$ if every possible action in w might lead to a world in X. The truth condition for dangerous states can then be re-stated as:

$M, w \models \mathbf{Dan}\phi$ iff $[\![\phi]\!]^M \subseteq Zone(D(w))$.

We can then define the three kinds of worlds as follows.

- w' is a disastrous world with respect to w iff $w' \in D(w)$.
- w' is a dangerous world with respect to w iff $w' \in Zone(D(w))$.
- w' is a safe world with respect to w iff $w' \in W/Zone(D(w))$.

The possibility operator $\langle\cdot\rangle$ can be extended to the more general operator $\langle\cdot\rangle^*$. Informally, $\langle a\rangle^*\phi$ means that ϕ is possible after some sub-action of action a. The new operator is defined as follows:

- $\langle a\rangle^*\phi \overset{\text{def}}{\Longleftrightarrow} \langle a\rangle\phi \vee \phi$ where a is primitive.
- $\langle a\cup b\rangle^*\phi \overset{\text{def}}{\Longleftrightarrow} \langle a\cup b\rangle\phi \vee (\langle a\rangle^*\phi \vee \langle b\rangle^*\phi) \vee \phi$
- $\langle a;b\rangle^*\phi \overset{\text{def}}{\Longleftrightarrow} \langle a;b\rangle\phi \vee \langle a\rangle^*\phi \vee \langle a\rangle\langle b\rangle^*\phi \vee \phi$

3 The Safety Logic SL

The logic of safety, **SL**, consists of the following axioms, definitions and inference rules.[1]

Axioms

(PC) All propositional tautologies
(A1) $\neg\langle a\rangle\bot$
(A2) $\langle a\rangle(\phi\vee\psi)\rightarrow\langle a\rangle\phi\vee\langle a\rangle\psi$
(A3) $\langle a;b\rangle\phi\leftrightarrow\langle a\rangle\langle b\rangle\phi$
(A4) $\langle a\cup b\rangle\phi\leftrightarrow\langle a\rangle\phi\vee\langle b\rangle\phi$

(D1) $\mathbf{Dis}\bot$
(D2) $\mathbf{Dis}(\phi\vee\psi)\rightarrow\mathbf{Dis}\phi\vee\mathbf{Dis}\psi$
(D3) $\mathbf{Dis}\phi\wedge\mathbf{Dis}\psi\rightarrow\mathbf{Dis}(\phi\vee\psi)$
(D4) $\mathbf{Dis}(a\cup b)\leftrightarrow\mathbf{Dis}(a)\wedge\mathbf{Dis}(b)$

(DA1) $\mathbf{Dan}\phi\wedge\mathbf{Dan}\psi\rightarrow\mathbf{Dan}(\phi\wedge\psi)$
(DA2) $\langle a\rangle\phi\wedge\mathbf{Dis}\phi\rightarrow\mathbf{Dan}(a)$
(DA3) $\mathbf{Dan}(a\cup b)\leftrightarrow\mathbf{Dan}(a)\vee\mathbf{Dan}(b)$
(DA4) $\mathbf{Dan}(a)\vee(\langle a;b\rangle\phi\wedge\mathbf{Dan}\phi)\rightarrow\mathbf{Dan}(a;b)$
(DA5) $\mathbf{Dan}\phi\wedge\phi\wedge\neg[a]\bot\rightarrow\mathbf{Dan}(a)$

(DD1) $\mathbf{Dis}\phi\rightarrow\mathbf{Dan}\phi$
(DD2) $\mathbf{Dis}(a)\rightarrow\mathbf{Dan}(a)$

Definitions

(A5) $\langle a\rangle^*\phi\leftrightarrow\langle a\rangle\phi\vee\phi$ (where a is primitive)
(A6) $\langle a\cup b\rangle^*\phi\leftrightarrow\langle a\cup b\rangle\phi\vee(\langle a\rangle^*\phi\vee\langle b\rangle^*\phi)\vee\phi$
(A7) $\langle a;b\rangle^*\phi\leftrightarrow\langle a;b\rangle\phi\vee\langle a\rangle^*\phi\vee\langle a\rangle\langle b\rangle^*\phi\vee\phi$
(A8) $[a]\phi\leftrightarrow\neg\langle a\rangle\neg\phi$

(SAdf) $\mathbf{Safe}(a)\leftrightarrow\neg\mathbf{Dan}(a)$
(SSdf) $\mathbf{Safe}\phi\leftrightarrow\neg\mathbf{Dan}\phi$

Inference Rules

(MP) From ϕ and $\phi\rightarrow\psi$ infer ψ
(NECA) From ϕ infer $[a]\phi$

[1] Some of the axioms may be redundant. We leave this question for further work.

(MONA) From $\langle a\rangle\phi$ and $\phi \rightarrow \psi$ infer $\langle a\rangle\psi$
(SPD) From $\psi \rightarrow \phi$ infer $\mathbf{Dis}\phi \rightarrow \mathbf{Dis}\psi$
(SPDS) From $\psi \rightarrow \phi$ infer $\mathbf{Dan}\phi \rightarrow \mathbf{Dan}\psi$

Theorems and Derived Rules

(SPS) From $\phi \rightarrow \psi$ infer $\mathbf{Safe}\phi \rightarrow \mathbf{Safe}\psi$
(SS1) $\neg\mathbf{Safe}\bot$
(SS2) $\mathbf{Safe}(\phi \wedge \psi) \leftrightarrow \mathbf{Safe}\phi \wedge \mathbf{Safe}\psi$
(SS3) $\mathbf{Safe}(\phi \wedge \psi) \rightarrow \mathbf{Safe}(\phi) \vee \mathbf{Safe}(\psi)$
(SA1) $\mathbf{Safe}(a \cup b) \leftrightarrow \mathbf{Safe}(a) \wedge \mathbf{Safe}(b)$
(SA2) $\mathbf{Safe}(a; b) \rightarrow \mathbf{Safe}(a)$
(SA3) $\mathbf{Safe}(a) \wedge \langle a\rangle^*\phi \rightarrow \neg\mathbf{Dis}\phi$
(TA1) $\phi \rightarrow \langle a\rangle^*\phi$

The axioms (A1)-(A4) and the inference rules (NECA) and (MONA) are those of Dynamic Logic. Axiom (D1) states that logical inconsistency is a disaster. (D2) states that disasters are decomposable under disjunction. (D3) states that if two states are disastrous (simultaneously), then one of them must be disastrous. (D4) states that action $a \cup b$ is disastrous exactly when actions a and b are disastrous. (DA1) states that dangerous states are closed under conjunction. (DA2) states that if one possible outcome of the action is disastrous, then the action is dangerous. (DA3) reiterates that dangerous actions are closed under disjunction. (DA4) states that if a sub-action is dangerous, then the whole sequence of the action is dangerous as well. (DA5) says that all actions which can be taken in a dangerous state are dangerous. (DD1) and (DD2) state that all disastrous states and actions are dangerous. The inference rule (SPD) states that any state which implies a disastrous state is also disastrous, and (SPDS) states that the same applies to dangerous states. Some of the properties of safe actions and sates are listed as theorems. (SA2) and (SA3) are of particular interest. (SA2) states that if an action sequence $a; b$ is safe, then action a is safe. It does *not* generally follow that it is safe to do b; as doing a may lead to dangers which are not otherwise apparent. This question is considered further in Section 5.2. Similarly, (SA3) states that a safe action can never lead to a disastrous state; although it may lead to a dangerous one.

Theorem 1 (Soundness of SL) *The logic* **SL** *is sound for the class of $\mathcal{SL}$-models.*

PROOF: The proofs for most axioms and inference rules are straightforward from the definitions. For example, proof for (DD1) is as follows. Suppose that $M, w \models \mathbf{Dis}\phi$, then, by the truth condition, we have $[\![\phi]\!]^M \subseteq D(w)$. For any $w_1 \in [\![\phi]\!]^M$ and any action a, we have $w_1 Raw_1 w_1$ by the definition of a trace. So $w_1 Raw_1 w_1$ and $w_1 \in D(w)$. So, for any $w_1 \in [\![\phi]\!]^M$ and any action a, there exists a world w_1 such that $w_1 Raw_1 w_1$ and $w_1 \in D(w)$. So, by the truth condition for dangerous states, we have $M, w \models \mathbf{Dan}\phi$. □

We have not yet proved the completeness of **SL**. There are two difficulties, which arise because the language $\mathcal{SL}$ lacks the required expressive power.

1. From the truth condition for dangerous actions, we know that if an action a is dangerous, then there exists a state ψ such that ψ is accessible via the action a (written

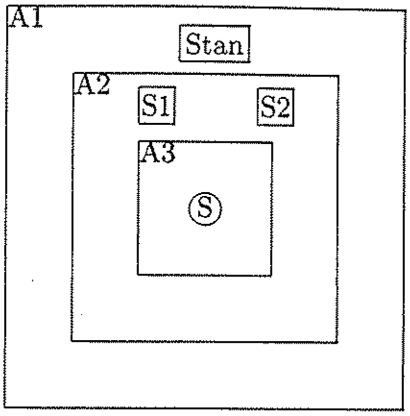


Figure 1: Power station Example

$\langle a\rangle^*\psi$), and ψ is disastrous, namely, the following "axiom" holds:

(DA$^+$) $\mathbf{Dan}(a) \rightarrow \langle a\rangle^*\psi \wedge \mathbf{Dis}\psi$ for some formula ψ.

However, the second-order existential quantification in this axiom cannot be expressed in $\mathcal{SL}$, or in a sensible rule schema.

2. The truth condition for dangerous states involves universal quantification over actions. However, $\mathcal{SL}$ does not permit this.

4 Reasoning about Safety: An Example

In order to illustrate how **SL** can be used to reason about safety, this section discusses an example of safety-critical reasoning in the imaginary nuclear power station pictured in Figure 1. The area $A1$ represents those parts of the power station which should be kept free of radiation. This area surrounds the area $A3$ which contains the nuclear reactor, and which is thus an area of high radioactivity. The area $A2$ acts as a "radiation lock" between $A1$ and $A3$; it provides access between the two areas, and is designed to prevent radioactivity spreading from area $A3$ to area $A1$. Area $A2$ contains safety suits $S1$ and $S2$ which are designed to protect against radioactivity. If an agent is wearing one of these suits, the agent can go into area $A3$ safely. But if the agent goes into area $A3$ without a suit, the agent will be exposed to a lethal dose of radiation. Agents who are wearing safety suits are not allowed to enter area $A1$, as the residual radiation on the suits would contaminate area $A1$. An accident has occurred, the reactor is out of control and is heading towards melt-down. The only remaining way of disabling it is by means of switch S in area $A3$. Our agent, *Stan*, who is currently located in area $A1$, has to form and execute a plan for doing this safely.

The initial state can be described as follows;

(*Init*) $Agent(Stan) \wedge Suit(S1) \wedge Suit(S2) \wedge Switch(S) \wedge Object(S1) \wedge Object(S2) \wedge Object(S) \wedge Area(A1) \wedge Area(A2) \wedge Area(A3) \wedge In(Stan, A1) \wedge In(S1, A2) \wedge In(S2, A2) \wedge In(S, A3) \wedge \neg Radiated(Stan) \wedge \neg Radioactive(A1)$

The safety constraints can be represented as follows:

$(C1)\quad In(Stan, A3) \wedge \neg(Wearing(Stan, S1) \vee Wearing(Stan, S2)) \rightarrow Radiated(Stan)$

$(C2)\quad In(S1, A1) \vee In(S2, A1) \rightarrow Radioactive(A1)$

Thus *Stan* gets a lethal dose of radiation if he enters $A3$ without a suit, and area $A1$ becomes radioactive if a suit is brought into it.

We will thus say that *Stan*'s actions are safe if they do not violate $(C1)$ and $(C2)$.

$(SA)\quad \mathbf{Safe}(a) \leftrightarrow \neg\langle a\rangle^{*} Radiated(Stan) \wedge \neg\langle a\rangle^{*} Radioactive(A1)$

And a safe shutdown state is defined as follows:

$(SS)\quad \mathbf{Safe}Shutdown \leftrightarrow Shutdown \wedge In(Stan, A1) \wedge \neg Radiated(Stan) \wedge \neg Radioactive(A1)$

In order to reason about action and change, we will, for present purposes, adopt the extended STRIPS-approach used in SIPE [2, 5]. The available actions (operators) are as follows. An agent can move between areas, or move to an object in an area, or move away from an object in an area.

- $Move(a, a1, a2)$. Preconditions: $Agent(a) \wedge Area(a1) \wedge Area(a2) \wedge In(a, a1) \wedge \neg\exists x NextTo(a, x)$.
  Additions: $In(a, a2)$. Deletions: $In(a, a1)$.
- $Move(a, a1, x)$. Preconds: $Agent(a) \wedge Area(a1) \wedge Object(x) \wedge In(a, a1) \wedge In(x, a1)$.
  Additions: $NextTo(a, x)$.
- $Move(a, x, a1)$. Preconds: $Agent(a) \wedge Object(x) \wedge Area(a1) \wedge In(a, a1) \wedge In(x, a1)$.
  Deletions: $NextTo(a, x)$.

An agent can also put on or take off a protective suit.

- $Enrobe(a, s)$. Preconds: $Agent(a) \wedge Suit(s) \wedge NextTo(a, s)$.
  Additions: $Wearing(a, s)$. Deletions: $NextTo(a, s)$.
- $Disrobe(a, s)$. Preconds: $Agent(a) \wedge Suit(s) \wedge Wearing(a, s)$.
  Additions: $NextTo(a, s)$. Delete list: $Wearing(a, s)$.

Finally, an agent can disable the reactor.

- $Disable(a, s)$. Preconds: $Agent(a) \wedge Switch(s) \wedge NextTo(a, s)$. Additions: $Shutdown$.

In STRIPS, an action a occurs in a state Σ. If the preconditions of a are true when a occurs, then a succeeds. The resulting state Σ' is obtained by adding the additions of a to Σ and removing the deletions of a from Σ. In SIPE, Σ may, in effect, also contain domain rules; such as the safety constraints $(C1)$ and $(C2)$. Indeed, in the present example we also need the following domain rules to describe a ramification of the move-area action:

$(C3)\quad In(a, a1) \wedge Wearing(a, s) \rightarrow In(s, a1)$

$(C4) \quad Object(o) \wedge In(o, a1) \wedge a1 \neq a2 \rightarrow \neg In(o, a2)$

Thus, if an agent is wearing a safety suit when it moves from one area to another, then the suit should go with the agent.[2] The presence of domain rules complicates the update process; as it is necessary, after the earlier revisions, to add the consequents of any domain rules that are applicable, and then delete any literals which were in Σ and whose complements have been added by the domain rules. We will refer to this process as a "SIPE update".

Let us assume that *Stan* produces the following non-linear plan:

$(\pi) \quad Move(Stan, A1, A2); (\pi_1 \cup \pi_2); Move(Stan, A2, A1)$

where

$\pi_1 = Move(Stan, A2, S1); Enrobe(Stan, S1);$
$Move(Stan, A2, A3); Move(Stan, A3, S);$
$Disable(Stan, S); Move(Stan, S, A3);$
$Move(Stan, A3, A2); Disrobe(Stan, S1);$
$Move(Stan, S1, A2)$

$\pi_2 = Move(Stan, A2, S2); Enrobe(Stan, S2);$
$Move(Stan, A2, A3); Move(Stan, A3, S);$
$Disable(Stan, S); Move(Stan, S, A3);$
$Move(Stan, A3, A2); Disrobe(Stan, S2);$
$Move(Stan, S2, A2)$

Our task is to formalise this plan and to show that it is safe; that is, that it satisfies (SA) and (SS). We can represent an action a in $\mathcal{SL}$ as the sentence $\Sigma \rightarrow [a]\Sigma'$; where Σ is a conjunction of sentences of $\mathcal{SL}$ which describes the state in which a occurs and which implies the preconditions of a, and Σ' is the appropriate SIPE-update of Σ. We will not attempt to formalise the inference process which produces the SIPE-update. We will thus assume, $(SIPE)$, that for each state Σ which is produced by the plan and which implies the preconditions of an action a in the plan, we have the sentence $\Sigma \rightarrow [a]\Sigma'$; in this example it is clear what Σ' is in each case. Given Σ, a can thus be applied, $(SIPE)$ and (MP), to give $[a]\Sigma'$. For conciseness, we will abbreviate $\Sigma \rightarrow [a]\Sigma'$ to $\Sigma[a]\Sigma'$. Our task is thus to prove that

$(SAS) \quad \mathbf{Safe}(\pi) \wedge \Sigma[\pi]\Sigma' \wedge (\Sigma' \rightarrow \mathbf{Safe}Shutdown)$

where Σ is the initial state description $(Init)$ conjoined with the domain rules $(C1)-(C4)$.

Here is an outline of the proof. The plan π in effect consists of a plan involving suit $S1$ and a plan involving suit $S2$:

$\pi_{S1} = Move(Stan, A1, A2); \pi_1; Move(Stan, A2, A1)$
$\pi_{S2} = Move(Stan, A1, A2); \pi_2; Move(Stan, A2, A1)$

[2] For convenience we have occasionally used first-order formulae in the prepresentation of the problem. These are easily translated into suitable, if long-winded, sentences of $\mathcal{SL}$.

It is sufficient to prove that π_{S1} and π_{S2} both satisfy (SAS); as it then follows from the theorem (SA1) and the theorem:

$(SA4)$ $[a]\phi \wedge [b]\phi \rightarrow [a \cup b]\phi$

that π satisfies (SAS). As the two plans are similar, we will outline the proof of the case for π_{S1} only.

We have the following action-application rule:

(App) From Σ and $\Sigma[a]\Sigma'$ infer $[a]\Sigma'$

And, by means of (MONA), (A3), (A8) and propositional reasoning, we can derive the following action-sequence rule:

(Seq) From $\Sigma[a]\Sigma'$ and $\Sigma'[b]\Sigma''$ infer $\Sigma[a;b]\Sigma''$.

In order to show that **Safe**(π_{S1}), we can use (MONA), (App), (Seq), $(SIPE)$ and Σ to successively show that:

$[Move(Stan, A1, A2)](\neg Radiated(Stan) \wedge \neg Radioactive(A1))$
$[Move(Stan, A1, A2); Move(Stan, A2, S1)](\neg Radiated(Stan) \wedge \neg Radioactive(A1))$
...
$[\pi_{S1}](\neg Radiated(Stan) \wedge \neg Radioactive(A1))$.

Then, by (MONA), (A5)-(A8) and propositional reasoning, we can obtain the desired conclusion.

$\neg\langle\pi_{S1}\rangle^* Radiated(Stan) \wedge \neg\langle\pi_{S1}\rangle^* Radiated(A1)$.

To complete the proof for π_{S1}, we can repeatedly use (Seq), and $(SIPE)$ to show that in $\Sigma[\pi_{S1}]\Sigma'$, Σ' is $\Sigma \wedge Shutdown$. And so, by propositional reasoning and (SS), we have $\Sigma' \rightarrow$ **Safe**$Shutdown$ as required.

5 Extensions of SL

Imposing various conditions on $\mathcal{SL}$-models results in different safety logics which include **SL**. Some examples are given in this section.

5.1 Stable disastrous states

An $\mathcal{SL}$-model $M = \langle W, \{R^a\}_{a \in PA}, D, V\rangle$ has stable disastrous states if D satisfies the condition $\forall w \forall w'(D(w) = D(w'))$. The corresponding logic is obtained by adding the following axioms to **SL**.

(Ds1) **Dis**$\phi \leftrightarrow [a]$**Dis**ϕ

(Ds2) $\mathbf{Dan}\phi \leftrightarrow [a]\mathbf{Dan}\phi$

(Dsl) and (Ds2) state that disastrous (dangerous) states are not changed no matter what action is taken. These yield the following extra properties.

(SD) $\mathbf{Safe}\phi \leftrightarrow [a]\mathbf{Safe}\phi$

5.2 Locally stable disastrous states

The condition is: $\forall w \forall w' \forall a (w' \in R^a w \Rightarrow D(w') \subseteq D(w))$, and the corresponding axiom is $\langle a \rangle \mathbf{Dan}(b) \to \mathbf{Dan}(a; b)$, giving the theorem $\mathbf{Safe}(a; b) \to [a]\mathbf{Safe}(b)$.

The condition means that there would be no more new disasters which can be created by taking any action. Therefore, the corresponding axiom says that if an action $a; b$ is safe (with respect to the current situation), then the the action b always is safe with respect to the situations after taking action a.

A typical example for this axiom is: if, having drunk alcohol, it is dangerous to drive, then it is dangerous to drink alcohol and then drive.

5.3 Logical inconsistency is the only disaster

Adding the condition $\forall w (D(w) = \emptyset)$ to $\mathcal{SL}$-models gives the corresponding axiom $\mathbf{Dis}\phi \to (\phi \leftrightarrow \bot)$, which states that logical inconsistency is the only disaster.

5.4 Disasters are other-world oriented

Adding the condition $\forall w (w \notin D(w))$ to $\mathcal{SL}$-models gives the corresponding axiom $\mathbf{Dis}\phi \to \neg\phi$, which states that disasters occur only in other worlds.

5.5 Dangerous world system

A dangerous world system is one in which all actions can be taken are dangerous. The condition is $\forall w \forall a (R^a w \neq \emptyset \Rightarrow w \in D(w))$, and the corresponding axiom is: $\neg[a]\bot \to \mathbf{Dan}(a)$.

5.6 Miserable world system

A miserable world system is one in which all worlds are disastrous. The condition is $\forall w (D(w) = W)$, and the corresponding axiom is $\mathbf{Dis}\top$.

6 Normative Safety

In the introduction we distinguished between absolute safety and normative safety, and have subsequently concentrated on formalising absolute safety. In a companion paper [1] we extend the formal machinery developed here to give a theory of normative safety, and the remainder of this section gives an indication of how this is done. The intuitive idea is that an action (state) is safe if it is not *normally* dangerous; where, for example, an action is not normally dangerous if it does not normally lead to a disastrous state. In order to formalise this, we add an abnormal-action operator to $\mathcal{SL}$. Informally, for action a, $Ab(a)$

states that action a has ended abnormally. For example, if we want to say that a flight always ends abnormally if the plane crashes, we can write:

$[fly](Crash \rightarrow Ab(fly))$,

In order to formalise this, we add an abnormal-action operator to $\mathcal{SL}$. Informally, for action a, $Ab(a)$

states that action a has ended abnormally. For example, if we want to say that a flight always ends abnormally if the plane crashes, we can write:

$[fly](Crash \rightarrow Ab(fly))$,

Formally, $\mathcal{SL}$-models are extended by adding a function $AB : W \rightarrow \mathcal{P}(ACTION)$ which assigns to each world w the set $AB(w)$ of all actions which end abnormally at w. Thus:

$M, w \models Ab(a)$ *iff* $a \in AB(w)$.

The Ab-operator can be used to define a "normal outcome" operator $[\cdot]_N$ as follows:

$$[a]_N\phi \overset{\text{def}}{\Longleftrightarrow} [a](\neg Ab(a) \rightarrow \phi)$$

Thus $[a]_N\phi$ states that ϕ holds in all *normal* outcomes of a. Formal semantics can be given for this operator and for the safety-related operators using the idea of the *normal trace* of an action. Informally, $NTrace(a, w)$ is the sub-tree of $Trace(a, w)$ consisting of all branches of $Trace(a, w)$ in which events unfold normally, that is, of all branches along which contain no worlds in which an action ends abnormally. Reasoning in the resulting normative safety logic $\mathbf{SL_N}$ can be illustrated by the flying example discussed in the introduction. Let Θ be the theory:

(1) $\mathbf{Safe_N}(fly) \leftrightarrow [fly]_N \neg Crash$
(2) $\mathbf{Safe}(fly) \leftrightarrow [fly]\neg Crash$
(3) $\langle fly \rangle Crash$
(4) $[fly](Crash \rightarrow Ab(fly))$

Thus, (1) it is normally safe to fly iff the plane doesn't normally crash, (2) it is absolutely safe to fly iff the plane never crashes, (3) it is possible that the plane will crash, and (4) the flight will end abnormally if the plane crashes. Then clearly $\Theta \models \mathbf{Safe_N}(fly) \wedge \neg\mathbf{Safe}(fly)$.

In order to be able to conclude that actions end normally, we define a preference order on models. Intuitively, the preferred models are those in which abnormalities occur as late as possible, that is, the preferred models are those which are chronologically most normal. We then define a preferential relation in the usual way [4]; a set of sentences Θ preferentially entails a sentence ϕ (written $\Theta \mid\!\approx \phi$) iff ϕ is true in all preferred models of Θ. In the example we then have $\Theta' \mid\!\approx \neg[fly]_N Crash$ as required.

7 Concluding Remarks

In this paper we distinguished between absolute safety and normative safety, and developed a formal theory of absolute safety. We started with the idea of a disastrous state (a state in which some disastrous fact is true) and defined a disastrous action to be an action which always leads to a disastrous state. We then defined dangerous actions to be actions which may lead to disastrous states, and dangerous states to be states in which every possible action is dangerous. Finally, safe actions (states) were defined to be actions (states) which are not dangerous. Using Dynamic Logic as a basis, we developed a formal possible-worlds semantics for our theory of safety, and defined the basic safety logic **SL** and several extensions of it. We then gave an example of reasoning about safety in nuclear power stations, and concluded with an indication of how we are building on this work to produce a formal account of normative safety.

References

[1] John Bell and Zhisheng Huang, Safety Logics II: Normative Safety. Proceedings of ECAI-96, 1996.

[2] Fikes, R., and Nilsson, N., STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving, *Artificial Intelligence* 2, 1971, pp. 189-209.

[3] Harel, D., Dynamic Logic, in: D. Gabbay and F. Guenthner, (eds.), *Handbook of Philosophical Logic*, Vol.II, (D. Reidel publishing company, 1984), 497-604.

[4] Shoham, Y. *Reasoning About Change*, M.I.T. Press, Cambridge, Massachusetts, 1988.

[5] D. Wilkins. *Practical Planning.* Morgan Kaufmann, San Mateo, California, 1988.

Safety Logics II: Normative Safety

John Bell and Zhisheng Huang [1]

Abstract. In this paper we continue our analysis and formalisation of reasoning about safety. In a companion paper we distinguish between *absolute* safety and *normative* safety, and use Dynamic Logic to develop a formal possible-worlds semantics and a logic for absolute safety. In this paper, we begin by outlining this theory. We then introduce *Defeasible Dynamic Logic* in order to give possible-worlds semantics and a logic for normative safety. We then define a preferential entailment relation defined in order to represent commonsense reasoning about the normal termination of actions. We conclude with a discussion of the relationship between safety, obligation, rationality and risk, and outline some extensions to the present work.

1 Introduction

Attempting to develop a formal theory of reasoning about safety is a complex task; as reasoning of this kind may involve commonsense reasoning about action and change, time, beliefs, goals, obligations, rationality, etc. In an earlier paper we began this task by abstracting away from many of these features in order to develop a formal theory which captures what we consider to be the essential features of what we call *absolute safety* [9]. In order to make this paper self-contained, we outline the essential features of this theory in the next section. We then extend this theory to include *normative safety*. We begin by introducing *Defeasible Dynamic Logic* and use this to give a possible-worlds semantics for normative safety. We then present a logic of normative safety. We then show how commonsense reasoning about the normal termination of actions can be represented by means of a preferential entailment relation. In conclusion, we discuss the relationships between safety, obligation, rationality and risk, and outline some extensions to the present work.

2 Absolute Safety

We start with disasters and the idea of a *disastrous state*. A disastrous state is one in which some fact, which the agent considers to be disastrous, is true. It is assumed that such states are abhorrent to the agent and that the agent tries to avoid them if at all possible. For instance, a typical medical safety-critical system would consider states in which the patient dies as a result of treatment to be disastrous, and would try to avoid them at all costs. We further assume that the application domain is such that there is no question as to which states the agent considers to be disastrous. Of course, what counts as a disastrous state may be context-dependent. For example, the agent might consider a state in which it has missed a plane to be disastrous if it has a non-transferrable ticket, but it need not do so if there is another flight and the ticket is transferrable. A *disastrous action* can now be defined to be an action which *always* leads to a disastrous state. We then define a *dangerous action* to be an action which *may* lead to a disastrous state, and a *dangerous state* to be state in which every action which the agent can perform – every action which is open to the agent – is dangerous. Note that disastrous actions (states) are thus dangerous actions (states). Finally, we define a *safe action* (a *safe state*) to be an action (a state) which is not dangerous.

We have chosen to formalize safety in Dynamic Logic [6], as this offers a powerful tool for formalizing actions using classical logic and Kripke semantics. In particular, states are represented as possible worlds, and actions are represented as accessibility relations between worlds. The crucial distinction between the necessary and the possible consequences of actions in our informal analysis is thus readily and naturally formalized. We begin by defining the language $\mathcal{SL}$. Let Φ_0 be a set of primitive propositions, and PA be a set of primitive actions. We will use the Roman letters $a, b, \ldots$ (with or without subscripts or superscripts) to denote actions, and lower case Greek letters $\phi, \psi, \ldots$ (with or without subscripts or superscripts) to denote formulas.

Definition 1 (Actions) *The set of (composite) actions $ACTION$ is, as usual, defined to be the smallest set which is closed under the following syntactic rules:*

- *If $a \in PA$ then $a \in ACTION$*
- *If $a, b \in ACTION$ then $(a \cup b), (a; b) \in ACTION$*

Here $\cup$ is the non-deterministic choice operator (so $a \cup b$ means "do either a or b non-deterministically"), and ';' is the sequence operator (so $a; b$ means "do a and then do b").

Definition 2 (The language $\mathcal{SL}$) *The language $\mathcal{SL}$ is the minimal set of formulas which satisfies the following conditions:*

- *If $p \in \Phi_0$ then $p \in \mathcal{SL}$.*
- *If $\phi, \psi \in \mathcal{SL}$ then $\neg\phi \in \mathcal{SL}$ and $\phi \wedge \psi \in \mathcal{SL}$.*
- *If $a \in ACTION$ and $\phi \in \mathcal{SL}$ then $\langle a\rangle\phi \in \mathcal{SL}$.*
- *If $\phi \in \mathcal{SL}$ then* **Dis**$\phi \in \mathcal{SL}$ *and* **Dan**$\phi \in \mathcal{SL}$.
- *If $a \in ACTION$ then* **Dis**$(a) \in \mathcal{SL}$ *and* **Dan**$(a) \in \mathcal{SL}$.

Informally, $\langle a\rangle\phi$ states that doing a makes ϕ possible, that is ϕ *may* be true after action a is taken. Then $[a]\phi$ is defined as $\neg\langle a\rangle\neg\phi$. Thus $[a]\phi$ states that doing a makes ϕ inevitable; that is, ϕ *will* be true after action a is taken. A sentence of the form **Dis**ϕ (**Dan**ϕ) means that ϕ is disastrous (dangerous). Similarly **Dis**(a) (**Dan**(a)) means that action a is disastrous (dangerous). The safety operators are then introduced by definition:

$$\mathbf{Safe}(a) \overset{\text{def}}{\Longleftrightarrow} \neg\mathbf{Dan}(a)$$

$$\mathbf{Safe}\phi \overset{\text{def}}{\Longleftrightarrow} \neg\mathbf{Dan}\phi$$

Recall that our analysis of safety started with disastrous states. We therefore extend the standard Kripke models of propositional Dynamic Logic by adding a disastrous-worlds function. In our analysis

[1] Applied Logic Group, Department of Computer Science, Queen Mary and Westfield College, University of London

ECAI 96. *12th European Conference on Artificial Intelligence*
Edited by W. Wahlster
Published in 1996 by John Wiley & Sons, Ltd.

we also want to view actions as being temporally forwards-directed, with each action leading from an earlier world to a set of later worlds. On this view, an action followed by its inverse (e.g. the *pickup* and *putdown* operators in STRIPS) can be conceived as leading to a new world which differs from the original one only in that the time stamp has changed; that is, time has moved on. It also seems natural to require that the underlying model of time is a tree, in which each branch represents a possible outcome of a sequence of actions.

Definition 3 ($\mathcal{SL}$-models) *A model for $\mathcal{SL}$ is a tuple $M = \langle W, \{R^a\}_{a \in PA}, D, V \rangle$ where*

- *W is a (non-empty) set of possible worlds,*
- *$R^a \subseteq W \times W$ is a binary accessibility relation for each primitive action $a \in PA$,*
- *$D : W \rightarrow \mathcal{P}(W)$ assigns to each world w the set $D(w)$ of worlds which are disastrous with respect to w, and*
- *$V : \Phi_0 \rightarrow \mathcal{P}(W)$ is the usual valuation function.*

As usual, we extend the accessibility relations for primitive actions to (composite) actions R^a_+ (usually written as R^a where there is no danger of ambiguity) as follows:

- *$R^a_+ = R^a$ (where a is primitive)*
- *$R^{a \cup b}_+ = R^a_+ \cup R^b_+$*
- *$R^{a;b}_+ = R^a_+ \circ R^b_+$ (composition of relations)*

Finally, we require that the global accessibility relation $R = \{\langle w, w' \rangle : wR^a_+ w'$ for some $a\}$ is a chronology *– that is, that R is a backwards-linear (or anti-convergent) partial order.*

Intuitively wR^a_+w' means that world w' is one possible outcome of doing action a in world w. In order to capture the idea of the consequences of an action *leading to* a disastrous state, we define the *trace* of each action a from each world w. The trace of a from w, is a tree rooted at w, each branch of which represents a course of events which might result from doing a at w. Thus a may lead to a disastrous world (state) if there is such a world on some branch of the trace of a from w. Formally, let $R^a w = \{w' \in W : \langle w, w' \rangle \in R^a_+\}$ be the set of all possible outcomes of doing the (possibly compound) action a at w. Then $Trace(a, w)$ is defined as follows.

- $Trace(a, w) = \langle w, w \rangle \cup \{\langle w, w' \rangle : w' \in R^a w\}$ where a is primitive
- $Trace((a \cup b), w) = Trace(a, w) \cup Trace(b, w)$
- $Trace((a; b), w) = Trace(a, w) \cup \bigcup_{w' \in R^a w} Trace(b, w') \cup (Trace(a, w) \circ \bigcup_{w' \in R^a w} Trace(b, w'))$

For brevity's sake we will write $w_1 R^a w w_2$ if $\langle w_1, w_2 \rangle \in Trace(a, w)$; thus, $w_1 R^a w w_2$ states that w_1 occurs no later than w_2 on a branch in the trace of a from w.

Definition 4 (Truth conditions for $\mathcal{SL}$) *Let $M = \langle W, \{R^a\}_{a \in PA}, D, V \rangle$ be an $\mathcal{SL}$-model. Then a sentence ϕ is true at a world w in M (written $M, w \models \phi$, or, equivalently, $w \in [\![\phi]\!]^M$) as follows.*

$$\begin{array}{lll}
M, w \models p & \textit{iff} & w \in V(p) \textit{ where } p \textit{ is primitive} \\
M, w \models \neg\psi & \textit{iff} & M, w \not\models \psi \\
M, w \models \psi \wedge \chi & \textit{iff} & M, w \models \psi \textit{ and } M, w \models \chi \\
M, w \models \langle a \rangle \psi & \textit{iff} & \exists w'(w' \in R^a w \textit{ and } w' \in [\![\psi]\!]^M) \\
M, w \models \mathbf{Dis}\psi & \textit{iff} & [\![\psi]\!]^M \subseteq D(w) \\
M, w \models \mathbf{Dis}(a) & \textit{iff} & \forall w_1(wR^a ww_1 \Rightarrow \exists w_2((w_1 R^a ww_2 \\
 & & \textit{or } w_2 R^a ww_1) \textit{ and } w_2 \in D(w))) \\
M, w \models \mathbf{Dan}(a) & \textit{iff} & \exists w'(wR^a ww' \textit{ and } w' \in D(w)) \\
M, w \models \mathbf{Dan}\psi & \textit{iff} & \forall w_1(w_1 \in [\![\psi]\!]^M \Rightarrow (\forall a)(R^a w_1 \\
 & & \neq \emptyset \Rightarrow \exists w_2(w_1 R^a w_1 w_2 \textit{ and } \\
 & & w_2 \in D(w))))
\end{array}$$

As usual, a sentence ϕ is said to be true in a model M (written $M \models \phi$) if $M, w \models \phi$ for all worlds w in M, and ϕ is said to be valid (written $\models \phi$) if ϕ is true in all models.

The logic of absolute safety, **SL**, is given in the companion paper [9], along with a detailed example showing how it can be used to reason about safety in a nuclear power station.

3 Normative Safety

We have defined an *absolute* notion of safety, in which an action is safe only if there is absolutely no possibility of it leading to a dangerous state. In practice, safety is a *normative* concept. That is, in everyday life almost all actions are dangerous, as there is always *some* (perhaps remote) possibility that the action will lead to a disastrous state. So in practice we are interested in actions which are *normally* safe; that is, in actions which do not normally lead to disastrous states. For example, it is not absolutely safe to fly by major airlines (because, there is always the possibility, however remote, of a crash) but it is normally safe to do so (as flights do not normally crash). In this section we extend the theory of safety to reasoning of this kind by adding a normative dimension to the concepts introduced in the previous section. For example, a *normally* dangerous action is one which, in the normal course of events, may lead to a disastrous state, and a *normally* dangerous state is one in which every action which is open to the agent is, in the normal course of events, dangerous.

3.1 The Language $\mathcal{SL}_\mathcal{N}$

In order to formalise the normative aspect, we introduce *Defeasible Dynamic Logic* by adding an abnormality operator, Ab, and a defeasible necessity operator, $[\cdot]_N$ to $\mathcal{SL}$.

Definition 5 (The language $\mathcal{SL}_\mathcal{N}$) *$\mathcal{SL}_\mathcal{N}$ is defined by replacing $\mathcal{SL}$ with $\mathcal{SL}_\mathcal{N}$ everywhere in Definition 2, and by adding the clauses:*

- *If $a \in ACTION$ then $Ab(a) \in \mathcal{SL}_\mathcal{N}$.*
- *If $a \in ACTION$ and $\phi \in \mathcal{SL}_\mathcal{N}$ then $[a]_N \phi \in \mathcal{SL}_\mathcal{N}$.*
- *If $\phi \in \mathcal{SL}_\mathcal{N}$ then $\mathbf{Dis}_\mathrm{N}\phi \in \mathcal{SL}_\mathcal{N}$ and $\mathbf{Dan}_\mathrm{N}\phi \in \mathcal{SL}_\mathcal{N}$.*
- *If $a \in ACTION$ then $\mathbf{Dis}_\mathrm{N}(a) \in \mathcal{SL}_\mathcal{N}$ and $\mathbf{Dan}_\mathrm{N}(a) \in \mathcal{SL}_\mathcal{N}$.*

Intuitively, $Ab(a)$ means that action a has ended abnormally, $[a]_N \phi$ states that ϕ holds in all normal outcomes of a, $\mathbf{Dis}_\mathrm{N}\phi$ ($\mathbf{Dan}_\mathrm{N}\phi$) states that ϕ is normally disastrous (normally dangerous), and $\mathbf{Dis}_\mathrm{N}(a)$ ($\mathbf{Dan}_\mathrm{N}(a)$) states that action a is normally disastrous (normally dangerous). The additional operators are then defined as follows:

$$\langle a \rangle_N \phi \overset{\text{def}}{\iff} \neg[a]_N \neg\phi$$

$$\mathbf{Safe}_\mathrm{N}(a) \overset{\text{def}}{\iff} \neg\mathbf{Dan}_\mathrm{N}(a)$$

$$\mathbf{Safe}_{\mathrm{N}}\phi \overset{\mathrm{def}}{\Longleftrightarrow} \neg\mathbf{Dan}_{\mathrm{N}}\phi$$

Thus $\langle a\rangle_N\phi$ states that ϕ holds in some normal outcome of a, and $\mathbf{Safe}_{\mathrm{N}}(a)$ ($\mathbf{Safe}_{\mathrm{N}}\phi$) states that action a (state ϕ) is normally safe.

$\mathcal{SL}$-models are extended to $\mathcal{SL}_{\mathcal{N}}$-models by adding the function AB which returns the set of actions which have ended abnormally at each world.

Definition 6 ($\mathcal{SL}_{\mathcal{N}}$-models) *An $\mathcal{SL}_{\mathcal{N}}$-model is a tuple $M = \langle W, \{R^a\}_{a\in PA}, AB, D, V\rangle$, where:*

- *W, $\{R^a\}_{a\in PA}$, D and V are as in Definition 3, and*
- *$AB : W \rightarrow \mathcal{P}(ACTION)$ assigns to each world w the set of actions which have ended abnormally at w. AB is required to satisfy the following conditions on compound actions:*
  - *$a;b \in AB(w)$ iff $\exists w', w''(w''R^a w'$ and $w'R^b w$ and $b \in AB(w))$, and*
  - *$a\cup b \in AB(w)$ iff $\exists w'(w'R^a w$ and $a \in AB(w)$, or $w'R^b w$ and $b \in AB(w))$.*

The conditions on AB ensure that if an action $a;b$ has ended abnormally at a world w then b has ended abnormally at w, and that if an action $a \cup b$ has ended abnormally at w then one of the actions a or b has ended abnormally at w.

The truth conditions for the normative safety operators are defined in terms of the notion of the *normative trace*, $Trace_N(a,w)$, of an action a from a world w. Informally, $Trace_N(a,w)$ is the subtree of $Trace(a,w)$ consisting of all branches of $Trace(a,w)$ in which events unfold normally, that is, of all branches which contain no worlds in which any sub-action of a ends abnormally. Formally, $Trace_N(a,w)$ is defined analogously to $Trace(a,w)$, except that $R^a_N w = \{w' \in W : \langle w,w'\rangle \in R^a_+$ and $a \notin AB(w')\}$ replaces $R^a w$ everywhere in the definition. In like manner we also abbreviate $\langle w_1,w_2\rangle \in Trace_N(a,w)$ to $w_1 R_N w w_2$.

Definition 7 (Truth conditions for $\mathcal{SL}_{\mathcal{N}}$) *Let $M = \langle W, \{R^a\}_{a\in PA}, AB, D, V\rangle$ be an $\mathcal{SL}_{\mathcal{N}}$-model. Then a sentence ϕ is true at a world w in M (written $M,w \models \phi$, or, equivalently, $w \in [\![\phi]\!]^M$) as in Definition 4 with the following additions.*

$$\begin{array}{lll}
M,w \models Ab(a) & \textit{iff} & a \in AB(w) \\
M,w \models [a]_N\psi & \textit{iff} & \forall w'(w' \in R^a_N w \Rightarrow w' \in [\![\psi]\!]^M) \\
M,w \models \mathbf{Dis}_{\mathrm{N}}\psi & \textit{iff} & M,w \models \mathbf{Dis}\psi \\
M,w \models \mathbf{Dis}_{\mathrm{N}}(a) & \textit{iff} & \forall w_1(wR^a_N ww_1 \Rightarrow \exists w_2((w_1R^a_N ww_2 \\
 & & \textit{or } w_2R^a_N ww_1) \textit{ and } w_2 \in D(w))) \\
M,w \models \mathbf{Dan}_{\mathrm{N}}(a) & \textit{iff} & \exists w'(wR^a_N ww' \textit{ and } w' \in D(w)) \\
M,w \models \mathbf{Dan}_{\mathrm{N}}\psi & \textit{iff} & \forall w_1(w_1 \in [\![\psi]\!]^M \Rightarrow (\forall a)(R^a w_1 \\
 & & \neq \emptyset \Rightarrow \exists w_2(w_1R^a_N w_1w_2 \textit{ and} \\
 & & w_2 \in D(w))))
\end{array}$$

3.2 The Safety Logic $\mathbf{SL}_{\mathbf{N}}$

The logic of (absolute and) normative safety, $\mathbf{SL}_{\mathbf{N}}$, is obtained by adding the following axioms and inference rules to the absolute safety logic **SL** [9].

Axioms

(AN1) $[a]_N\top$
(AN2) $[a]_N(\phi\wedge\psi) \leftrightarrow [a]_N\phi \wedge [a]_N\psi$
(AN3) $[a;b]_N\phi \leftrightarrow [a]_N[b]_N\phi$
(AN4) $[a\cup b]_N\phi \leftrightarrow [a]_N\phi \wedge [b]_N\phi$
(AN5) $[a]_N\phi \leftrightarrow [a](\neg Ab(a) \rightarrow \phi)$

(AB1) $Ab(a;b) \rightarrow Ab(b)$
(AB2) $Ab(a\cup b) \leftrightarrow Ab(a) \vee Ab(b)$

(DN1) $\mathbf{Dis}_{\mathrm{N}}\phi \leftrightarrow \mathbf{Dis}\phi$
(DN2) $\mathbf{Dis}_{\mathrm{N}}(a\cup b) \leftrightarrow \mathbf{Dis}_{\mathrm{N}}(a) \wedge \mathbf{Dis}_{\mathrm{N}}(b)$

(DAN1) $\mathbf{Dan}_{\mathrm{N}}\phi \wedge \mathbf{Dan}_{\mathrm{N}}\psi \rightarrow \mathbf{Dan}_{\mathrm{N}}(\phi\wedge\psi)$
(DAN2) $\langle a\rangle_N\phi \wedge \mathbf{Dis}_{\mathrm{N}}\phi \rightarrow \mathbf{Dan}_{\mathrm{N}}(a)$
(DAN3) $\mathbf{Dan}_{\mathrm{N}}(a\cup b) \leftrightarrow \mathbf{Dan}_{\mathrm{N}}(a) \vee \mathbf{Dan}_{\mathrm{N}}(b)$
(DAN4) $\mathbf{Dan}_{\mathrm{N}}(a) \vee (\langle a;b\rangle_N\phi \wedge \mathbf{Dan}_{\mathrm{N}}\phi) \rightarrow \mathbf{Dan}_{\mathrm{N}}(a;b)$

(DDN1) $\mathbf{Dis}_{\mathrm{N}}\phi \rightarrow \mathbf{Dan}_{\mathrm{N}}\phi$
(DDN2) $\mathbf{Dis}_{\mathrm{N}}(a) \rightarrow \mathbf{Dan}_{\mathrm{N}}(a)$

(DDAN1) $\mathbf{Dis}(a) \rightarrow \mathbf{Dis}_{\mathrm{N}}(a)$
(DDAN2) $\mathbf{Dan}_{\mathrm{N}}(a) \rightarrow \mathbf{Dan}(a)$
(DDAN3) $\mathbf{Dan}_{\mathrm{N}}\phi \rightarrow \mathbf{Dan}\phi$

Definitions

(PosN) $\langle a\rangle_N\phi \leftrightarrow \neg[a]_N\neg\phi$
(SAdf) $\mathbf{Safe}_{\mathrm{N}}(a) \leftrightarrow \neg\mathbf{Dan}_{\mathrm{N}}(a)$
(SSdf) $\mathbf{Safe}_{\mathrm{N}}\phi \leftrightarrow \neg\mathbf{Dan}_{\mathrm{N}}\phi$

Inference Rules

(MONAN) From $[a]_N\phi$ and $\phi \rightarrow \psi$ infer $[a]_N\psi$
(SPDSN) From $\psi \rightarrow \phi$ infer $\mathbf{Dan}_{\mathrm{N}}\phi \rightarrow \mathbf{Dan}_{\mathrm{N}}\psi$

Theorems and Derived Rules

(PAN) $\langle a\rangle_N\phi \leftrightarrow \langle a\rangle(\phi\wedge\neg Ab(a))$
(SPSN) From $\phi \rightarrow \psi$ infer $\mathbf{Safe}_{\mathrm{N}}\phi \rightarrow \mathbf{Safe}_{\mathrm{N}}\psi$
(SSN1) $\neg\mathbf{Safe}_{\mathrm{N}}\bot$
(SSN2) $\mathbf{Safe}_{\mathrm{N}}(\phi\wedge\psi) \leftrightarrow \mathbf{Safe}_{\mathrm{N}}\phi \wedge \mathbf{Safe}_{\mathrm{N}}\psi$
(SSN3) $\mathbf{Safe}_{\mathrm{N}}(\phi\wedge\psi) \rightarrow \mathbf{Safe}_{\mathrm{N}}\phi \vee \mathbf{Safe}_{\mathrm{N}}\psi$
(SAN1) $\mathbf{Safe}_{\mathrm{N}}(a\cup b) \leftrightarrow \mathbf{Safe}_{\mathrm{N}}(a) \wedge \mathbf{Safe}_{\mathrm{N}}(b)$
(SAN2) $\mathbf{Safe}_{\mathrm{N}}(a;b) \rightarrow \mathbf{Safe}_{\mathrm{N}}(a)$

The axioms (AN1)-(AN5), (AB1), (AB2), (PosN) and the inference rule (MONAN) give our Defeasible Dynamic Logic. The remaining axioms and inference rule state properties of the defeasible safety operators and the relationship between them and the absolute safety operators. Some of the properties of normally safe actions and sates are listed as theorems.

Theorem 1 (Soundness of $\mathbf{SL}_{\mathbf{N}}$) *The logic $\mathbf{SL}_{\mathbf{N}}$ is sound for the class of $\mathcal{SL}_{\mathcal{N}}$-models.*

It is not possible to prove the completeness of $\mathbf{SL}_{\mathbf{N}}$ by means of the standard method of canonical models. In on-going work we are looking at other methods and at proving the independence of the axioms.

Example 1 *Let Θ be the theory:*

$$\mathbf{Safe}(fly) \leftrightarrow [fly]Arrive \quad (1)$$
$$\mathbf{Safe}_{\mathrm{N}}(fly) \leftrightarrow [fly]_N Arrive \quad (2)$$
$$Preconds(fly) \rightarrow$$

$$[fly]_N Arrive \wedge \langle fly \rangle Arrive \wedge \langle fly \rangle \neg Arrive \quad (3)$$

$$Preconds(fly) \quad (4)$$

$$Ab(fly) \leftrightarrow Crashed \vee Hijacked \vee \ldots \quad (5)$$

That is, the flight is absolutely safe if, no matter what happens, it arrives at the destination (1), the flight is normally safe if it arrives at the destination in the normal course of events (2), if the usual preconditions hold then, in the normal course of events, the flight will arrive at the destination but there is always the possibility that it does not (3), the usual preconditions hold (4), and, finally the flight proceeds normally if the plane does not crash, is not hijacked, etc. (5). Then $\Theta \models$ **Safe**$_N(fly) \wedge \neg$**Safe**(fly). *That is, given* Θ *we can infer that it is normally safe to fly, but not absolutely safe to do so. However, we cannot infer that it is possible for the flight to arrive safely. As is clear from (AN5), the conclusion* $[fly]_N Arrive$ *only guarantees arrival whenever the flight ends normally, it does not guarantee that the flight does ever end normally.*

3.3 Chronologically Minimising Abnormalities

While it is now possible to distinguish between absolute safety and normative safety and to reason about normative safety, we still do not have a complete account of commonsense reasoning about it. For example, in reasoning that it is normally safe to fly, we make the default assumption that, given that the usual preconditions hold (the plane is airworthy, has been fuelled, etc.), the flight will be a normal one. It should thus at least be possible that the flight ends normally. In order to represent this aspect of commonsense reasoning about actions, we will, following [11], define a preferential entailment relation. Intuitively, we prefer models in which events unfold normally. Technically, we restrict consideration to models in which abnormalities occur as late as possible; that is, to models which are chronologically most normal.

Definition 8 *Let* $M = \langle W, \{R^a\}_{a \in PA}, AB, D, V \rangle$ *and* $M' = \langle W', \{R^{a'}\}_{a \in PA}, AB', D', V' \rangle$ *be* $\mathcal{SL_N}$*-models with respective chronologies* R *and* R'. M *and* M' *are said to be* comparable *if* $W = W'$, $R = R'$ *and* $D = D'$. M *is said to be* chronologically more normal than M' *(written* $M \prec M'$*) if* M *and* M' *are comparable and there is a world* $w \in W$ *such that:*

- *for any world* $w' \in W$ *such that* $w \neq w'$ *and not* wRw',
  - $AB(w') = AB'(w')$ *and*
  - $\{p \in \Phi_0 : w' \in V(p)\} = \{p \in \Phi_0 : w' \in V'(p)\}$,
- *for any action* a, $M', w \models Ab(a)$ *if* $M, w \models Ab(a)$, *and*
- *for some action* a, $M', w \models Ab(a)$ *and* $M, w \not\models Ab(a)$.

Thus $M \prec M'$ if M and M' are comparable, M and M' agree on all worlds which precede or are unrelated to some world w in their common chronology, every action that ends abnormally at w in M also ends abnormally at w in M', and there is at least one action which ends normally at w in M and which ends abnormally at w in M'.

Definition 9 *An* $\mathcal{SL_N}$*-model* M *is a* chronologically most normal model *(a* c.m.n.-model*) of a sentence* ϕ *if* $M \models \phi$ *and there is no model* M' *such that* $M' \models \phi$ *and* $M' \prec M$. *Similarly,* M *is a c.m.n.-model of a set of sentences* Θ *if* $M \models \Theta$ *and there is no model* M' *such that* $M' \models \Theta$ *and* $M' \prec M$.

Definition 10 (Normal entailment) *A set of sentences* Θ normally entails *a sentence* ϕ *(written* $\Theta \approx\models \phi$*) if* $M \models \phi$ *for any c.m.n.-model* M *of* Θ.

Example 2 *Let* Θ *be as in Example 1. Then:* $\Theta \approx\models$ **Safe**$_N(fly) \wedge \neg$**Safe**$(fly) \wedge \langle fly \rangle_N Arrive$. *Given* Θ, *we can conclude that it is normally safe to fly, but not absolutely safe to do so, and, moreover, we can also conclude by default that it is possible that the flight ends normally; that is, that there is at least one outcome of the flying action which does not involve an abnormality. (In outline, the proof is as follows. By chronological minimisation of abnormalities we have* $\neg[fly]Ab(fly)$, *that is* $\langle fly \rangle \neg Ab(fly)$. *This together with* $[fly]_N Arrive$, *that is,* $[fly](\neg Ab(fly) \rightarrow Arrive)$, *(AN5), gives* $\langle fly \rangle (\neg Ab(fly) \wedge Arrive)$, *that is* $\langle fly \rangle_N Arrive$, *(PAN).)*

A general argument for the use of chronological minimisation in *predictive* commonsense reasoning is given in [1, 2].

4 Safety, Obligation, Rationality and Risk

In this section we discuss some the relationships between safety, obligation, rationality and risk, and indicate how the present work might be extended to include these.

It is interesting to consider the relationship between obligation and safety. A central idea here is that of a *safety procedure* or *protocol*. Protocols are designed to maintain safety and have to be revised if they are found to lead to dangerous states. Protocols impose deontic restrictions on agents, as the agents involved are (normally) obliged to follow them. For example, in hospitals medical staff are (normally) obliged to follow protocols when treating patients; see, e.g, [5]. In order to formalise these constraints we could add the deontic operators **Obl** and **Per** to $\mathcal{SL_N}$. Intuitively, **Obl**(a) states that action a is obligatory, and **Per**(a) states that action a is permitted. Of course we have to provide appropriate formal semantics for these operators. One trick might be to introduce an action operator, *Do*, and then use the standard possible-world semantics for deontic operators; for example, $\mathbf{Obl}(a) \overset{\text{def}}{\Longleftrightarrow} \mathbf{Obl}Do(a)$. We can then, for example, state that only procedures which are normally safe can be followed:

$$\mathbf{Per}(a) \rightarrow \mathbf{Safe}_N(a).$$

The problem with this approach is that it may not always be *rational* to do only safe actions. For example if a patient has reached a critical stage, it may be that an *unsafe* treatment is the only option. So, in exceptional circumstances agents must be able to override protocols. Moreover, in domains where safety is not considered critical, it is often rational to do things which are not even normally safe. For example, an investor may risk money on an unsafe investment as this promises to bring in a greater return than any safe alternative, or an adult may risk their life trying to save that of a child. Indeed, in much of everyday life it is rational to take risks; that is to do dangerous things. In the case of protocols, we thus want the imperative to sate that only rational actions are permitted:

$$\mathbf{Per}(a) \rightarrow \mathbf{Rat}(a).$$

The problem now is to give a semantic account of the rationality operator. One starting point is the AI-planning theory of practical rationality outlined in [3] which begins with the following definition:

> A resource-bounded agent behaves rationally if it reasons and acts so as to achieve as many of its goals, in their comparative order of importance to the agent, as is possible given the resources available to it and the constraints in force.

In this paper we will simplify the theory dramatically. The relevant components are an agenda of goals, a planner which produces plans for these goals, and a scheduler which chooses the best plan to execute.

The agenda can be formalised using the preference semantics developed in [7, 8]. In order to incorporate them, two components need to be added to our models:

- $cw : W \times \mathcal{P}(W) \rightarrow \mathcal{P}(W)$ is a function, which selects the set $cw(w, [\![\phi]\!]^M)$ of closest worlds to w in which ϕ is true, and
- $\succ \subseteq \mathcal{P}(W) \times \mathcal{P}(W)$ is a preference relation on sets of worlds.

We can then introduce the preference operator $\mathbf{P}$ and give its truth conditions as follows:

$$M, w \models \phi \mathbf{P} \psi \text{ iff } cw(w, [\![\phi \wedge \neg\psi]\!]^M) \succ cw(w, [\![\psi \wedge \neg\phi]\!]^M).$$

So $\phi\mathbf{P}\psi$ is true at w if every closest $\phi \wedge \neg\psi$-world to w is preferred to every closest $\psi \wedge \neg\phi$-world to w. The truth condition thus formalises von Wright's *conjunction expansion principle* for preferences. We can now formalise the agenda by writing $\phi\mathbf{P}\phi'$ to indicate that goal ϕ is preferred to goal ϕ'.

In order to represent the planner we will, for present purposes, simplify and assume that there is only one possible plan (action) which will achieve each goal. We can therefore define the plan a which will achieve goal ϕ as:

$$Plan(a, \phi) \overset{\text{def}}{\Longleftrightarrow} [a]\phi \wedge \forall b(b \neq a \rightarrow \neg[b]\phi).$$

We can then define a preference ordering on plans according to the desirability of the goals they are designed to achieve:

$$a \succ_d b \overset{\text{def}}{\Longleftrightarrow} Plan(a, \phi) \wedge Plan(b, \psi) \wedge \phi\mathbf{P}\psi$$

A plan (action) can then be defined to be executable if it is possible given the resources available: $Exec(a) \overset{\text{def}}{\Longleftrightarrow} \neg[a]\bot$. The scheduler should then execute the most preferred executable plan:

$$\mathbf{Rat}(a) \overset{\text{def}}{\Longleftrightarrow} Exec(a) \wedge \neg\exists b(Exec(b) \wedge b \succ_d a).$$

It may be important to reason about degrees of risk; for example, an investor should take reasonable risks. In order to represent this kind of reasoning, the operator Sat can be added to the language, where, intuitively, $Sat(a)$ means that the risk associated with action a is acceptable. Models can be extended to include a function which associates an appropriate level of risk with each action at each world:

$$SR : W \times ACTION \rightarrow \omega$$

Then, letting $Risk(a, w)$ be the number of branches in the (normative) trace of a from w which contain at least one disastrous world, we can supply truth conditions for the new operator as follows.

$$M, w \models Sat(a) \quad \textit{iff} \quad Risk(a, w) \leq SR(a, w)$$

The most preferred, executable action could then be defined to be rational if the degree of risk involved is acceptable:

$$\mathbf{Rat}(a) \overset{\text{def}}{\Longleftrightarrow} Exec(a) \wedge \neg\exists b(Exec(b) \wedge b \succ_d a) \wedge Sat(a).$$

5 Concluding Remarks

In this paper we extended our analysis and formalisation of absolute safety to normative safety. We extended the formalism by introducing a defeasible extension of dynamic logic in order to give possible-worlds semantics and a logic for normative safety. We then defined a preferential entailment relation in order to be able to represent commonsense reasoning about actions. We then discussed the relationship between safety, obligation and rationality, and gave an indication of how the present work might be extended to produce a formal theory of safety, obligation, rationality, and risk.

As a result of the review process we have become aware of the work on reasoning about safety in theoretical Computer Science, for example that of [10], and of Dunin-Keplicz and Radzikowska's extension of Dynamic Logic to incorporate actions with typical effects [4].

ACKNOWLEDGEMENTS

This research formed part of the RED project, *Decision Support Systems, Safety and Liability*, and was supported by the United Kingdom Engineering and Physical Sciences Research Council under grant number GR/H 12348. We would also like to thank the reviewers and everyone else who has commented on it.

REFERENCES

[1] Bell, J., Prediction Theories and Explanation Theories. Unpublished manuscript. Available from ftp.dcs.qmw.ac.uk in a directory named applied_logic/bell as a file pt&et.ps.

[2] Bell, J., Remarks on the Evaluation of Formal Theories of Causal Reasoning. Working Notes of the IJCAI-95 Workshop on *Nonmonotonic Reasoning, Action and Change*, Williams, M., (ed.), pp. 14-21.

[3] Bell, J., A Planning Theory of Practical Rationality. Proceedings of the AAAI-95 Fall Symposium on Rational Agency, M.I.T., 1995. Fehling, M. (ed.), pp. 1-4.

[4] Dunin-Keplicz, B., and Radzikowska, A., Epistemic Approach to Actions with Typical Effects. Proceedings of *ECSQARU'95*. Froidevaux, C., and Kholas, J., (Eds). Lecture Notes in Artificial Intelligence No. 946. Springer, Berlin, 1995, pp. 180-188.

[5] Hammond, P., and Sergot, M., Computer Support for Protocol-Based Treatment of Cancer. Proceedings of the 2nd International Conference on the Practical Applications of Prolog, London, 1994.

[6] Harel, D., Dynamic Logic, in: D. Gabbay and F. Guenthner, (eds.), *Handbook of Philosophical Logic*, Vol.II, (D. Reidel publishing company, 1984), 497-604.

[7] Huang, Z., *Logics for Agents with Bounded Rationality*, ILLC Dissertation series 1994-10, University of Amsterdam, 1994.

[8] Huang, Z., Masuch, M., and Pólos, L., ALX: an action logic for agents with bounded rationality, *Artificial Intelligence* **82** (1996), pp. 101-153.

[9] Huang, Z., and Bell, J., Safety Logics I : Absolute Safety. Proceedings of Common Sense '96. Buvac, S., and Costello, T., (eds.), pp. 59-66.

[10] Manna, Z., and Pneuli, A., *The Temporal Logic of Reactive and Concurrent Systems: Specification*, Springer Verlag, New York, 1991.

[11] Shoham, Y. *Reasoning About Change*, M.I.T. Press, Cambridge, Massachusetts, 1988.

[12] von Wright, G., *The Logic of Preference*, (Edinburgh, 1963).

List of Participants

John Bell, Dept. of Computer Science, Queen Mary and Westfield College, University of London, London, E1 4NS, UK, jb@dcs.qmw.ac.uk

Paul Dongha, Dept of Computation, UMIST, P.O. Box 88, Manchester, M60 1QD, UK, dongha@sna.co.umist.ac.uk

John Fox, Advanced Computation Laboratory, Imperial Cancer Research Fund, London WC2A 2PX, UK, jf@acl.lif.icnet.uk

Wiebe van der Hoek, Utrecht University, Department of Computer Science, Padualaan 14, De Uithof, P.O. Box 80.089, 3508 TB Utrecht, The Netherlands, wiebe@cs.ruu.nl

Lisa Marie Hogg, Dept. of Electronic Engineering, Queen Mary and Westfield College, University of London, London E1 4NS, UK, L.M.Hogg@qmw.ac.uk

Zhisheng Huang, Dept. of Computer Science, Queen Mary and Westfield College, University of London, London, E1 4NS, UK, huang@dcs.qmw.ac.uk

Tony Hunter, Department of Computing, Imperial College, 180 Queen's Gate, London, SW7 2BZ, UK, abh@doc.ic.ac.uk

Paul Krause, Advanced Computation Laboratory, Imperial Cancer Research Fund, London, WC2A 2PX, UK, pjk@acl.lif.icnet.uk

Bernd van Linder, Utrecht University, Department of Computer Science, Padualaan 14, De Uithof, P.O. Box 80.089, 3508 TB Utrecht, The Netherlands, bernd@cs.ruu.nl

Jerome Mengin, Department of Computer Science, Oxford-Brookes University, Oxford, UK, p0071586@brookes.ac.uk

Rob Miller, Department of Computing, Imperial College, 180 Queen's Gate, London, SW7 2BZ, UK, rsm@doc.ic.ac.uk

Simon Parsons, Dept. of Electronic Engineering, Queen Mary and Westfield College, University of London, London, E1 4NS, UK, S.Parsons@qmw.ac.uk

Mark Ryan, Department of Computer Sciemce, University of Birmingham, UK, M.D.Ryan@cs.bham.ac.uk

Murray Shanahan, Dept. of Computer Science, Queen Mary and Westfield College, University of London, London, E1 4NS, UK, mps@dcs.qmw.ac.uk

Pierre-Yves Schobbens, Institut d'Informatique, Facultes Universitaires de Namur, Rue, Grandgagnage 21, 5000 Namur, Belgium, pys@info.fundp.ac.be

Yao-Hua Tan, Department of Computer Science, Erasmus University Rotterdam, P.O. Box 1738, NL-3000 DR Rotterdam, The Netherlands, ytan@sjaan.fbk.eur.nl

Nic Wilson, Department of Computer Science, Oxford-Brookes University, Oxford, UK, p0071587@brookes.ac.uk