Modal Action Logics for Reasoning about Reactive Systems

SIKS Dissertation Series No. 2003-2

The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

ISBN 90-9016611-4

VRIJE UNIVERSITEIT

Modal Action Logics for Reasoning about Reactive Systems

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor aan
de Vrije Universiteit Amsterdam,
op gezag van de rector magnificus
prof. dr. T. Sminia,
in het openbaar te verdedigen
ten overstaan van de promotiecommissie
van de faculteit der Exacte Wetenschappen
op dinsdag 25 februari 2003 om 15.45 uur
in de aula van de universiteit,
De Boelelaan 1105

door

Johannes Maria Broersen

geboren te Den Helder

promotoren: prof. dr. R.J. Wieringa
prof. dr. J.-J.Ch. Meyer
prof. dr. R.P. van de Riet

Preface

Roughly, we can distinguish two types of researchers involved in logic related research: the ones that are interested in applying logic and the ones that are interested in proving theorems about logic. These groups cannot do without each other. Appliers look for logics with good theoretical properties that suit their application domain, and theorists justify their work by reference to possible future applications. This Ph.D. thesis is somewhere in between: it develops logics with a clear application domain in mind: the specification of reactive systems. On the other hand, it embarks on some investigations into the formal properties of the logics defined.

But, a Ph.D. thesis is never finished. And this holds in particular for the one you are reading now. Each of the four main chapters contains enough open questions and directions for future research to make their subjects suitable candidates for separate doctoral works. So, none of the projects started up in these chapters is actually finished. This gives some explanation for the fact that I have had great difficulty determining where to stop. But now that I did manage to put an end to it, I want to thank all the people who have played a role in this project.

First of all I thank my supervisors prof. dr. Roel Wieringa, prof. dr John-Jules Charles Meyer and prof. dr Reind van de Riet: Roel for all philosophical and directional input, and in particular, for having the confidence that I could manage to do this the way I did, John-Jules for many valuable discussions on AI-related aspects of my work (which basically means that I discussed all parts of this Ph.D. thesis with him), and Reind for pleasant talks on my work, music, my position at the university, and many other things.

Also I am very grateful to prof. dr. Wiebe van der Hoek, prof. dr. Jan Treur, dr. Yde Venema and prof. dr. Krister Segerberg, for their willingness to take a place in the reading committee for this doctoral thesis, and for their approval of the work. I am especially thankful to Wiebe van der Hoek, who came up with many good comments, and who managed to find the time and

the place to discuss them with me during a short visit to Utrecht on his way from New-Zealand to the United Kingdom.

And then there is a large group of people with whom I have had the pleasure of sharing a university room, a hotel, a deadline, a lunch, a dinner, a car, a train, an aeroplane, a drink, a laugh, etc.: Leon van der Torre, Mehdi Dastani, Joris Hulstijn, Zisheng Huang, Jeroen Scheerder, Remco Feenstra, Perry Groot, Martin Caminada, Radu Serban, Rik Eshuis, David Jansen, Rogier van Eijk, Mark Ryan, Alessio Lomuscio, Frank van Harmelen, Marta Sabou, Maarten Marx, Frank and Virginia Dignum, Henry Prakken, and many more. Some of these colleagues have actually become close friends.

I could also not have done without the support of my wonderful mother, my family, in-laws, ice-skate friends, poker friends and of Nanacht. My father encouraged me in his own way. He said he had bought a new suit for the occasion. But the occasion turned out to be a much sadder one. If I say that he would have been proud of me, I feel that I do not do him justice: his happiness for me would have outwayed his proudness by far. He was the best father anyone can wish to have. Finally, I owe the most special kind of gratitude to Mieke; our love has brought me much more than any intellectual endeavor can do.

I end this preface with two remarks. The first is that some references to the literature might suggest that I do not object against the military use of results in Artificial Intelligence. But I do. The references simply reflect that authors making other moral choices can also have ideas that are scientifically relevant. The second remark is about the use of pronouns. In the remainder of this Ph.D. thesis I use ‘he’ as a homonym for the two sorts of entities commonly referred to by the words ‘he’ and ‘she’, and similarly I use ‘we’ as a homonym for what is commonly referred to by ‘we’ and ‘I’.

Jan Broersen
Amsterdam, december 2002

Contents

Chapter 1

General introduction

This Ph.D. thesis is about the development of logics that can support a designer of reactive systems in the initial stages of modeling, where on an abstract level he is reasoning about functional properties that the system is expected or required to satisfy. In these initial stages of design, it is necessarily the case that the system under design is an abstract entity. It is abstract in the sense that it is an abstraction of a huge number of possible implementations. In any engineering project, the process we call 'design' ideally starts at this abstract level, and not at the implementation level. However, implications of design choices should be available to a system designer as soon as possible. Early inspection of implications is the best guarantee that the system being developed is actually the system the specifier intends (we call such 'checks' concerning conformity with the specifiers intentions 'validations'), and that conceptual errors and internal inconsistencies are discovered (we call such checks 'verifications') before correcting them will be too laborious. The following quote from a short (invited) paper by John Rushby called 'calculating with requirements' [161] accurately describes the issue.

> 'The reasons for favoring mathematical modeling and calculation are the same in computer science as in other engineering disciplines: they allow the consequences of requirements and the properties of design to be accurately predicted and evaluated prior to construction.'

The central paradigm in this thesis is that a reactive system can be viewed as a collection of *actions* that jointly produce the behavior desired by a reactive system specifier. We develop several action logics that enable a specifier to characterize design choices using logic formulas. Implications of design choices

thus reified, are then available as properties entailed according to the specific action logic used. The activity of checking these entailment relations is an example of *reasoning* about functional reactive system requirements (desired system properties). Another way of characterizing this type of reasoning is to make the following comparison: reasoning about requirements is to the abstract design level what execution and testing are to the implementation level. Written as an equation:

$$\begin{gathered}\text{Reasoning about requirements} : \text{Abstract Design Level} \\ = \\ \text{Execution and Testing} : \text{Implementation Level}\end{gathered}$$

The equation essentially compares two computational activities: the testing for functionality of an implemented system by executing it, and the verification of entailment relations between different types of requirement and specification properties. Both computations are examples of verifications performed on a system under development, but concern entirely different levels of abstraction. Note that the subject of what is traditionally called 'formal verification' of system implementations concerns a cross-level relation between the abstract design level and the implementation level: an implementation is formally verified to obey requirements. The languages developed in this thesis are very well suited to function as specification languages in such cross level formal verifications, but we do not develop them with this application in mind. We intend to stay entirely on the abstract design level to study the logic properties that reside there.

In the present chapter we sketch the background of our research and formulate a problem definition. First, in section 1.1, we focus in detail on the ways in which logic can be used as an inference engine for the prediction of system properties, thus helping a designer to derive consequences that he is not able or willing to deduce himself. We compare our view to the more traditional one where there is a strict dichotomy between programming and specification. One of the objectives for this comparison is to convince the reader that this Ph.D. thesis is not about programming, but about reasoning about reactive system requirements in terms of declarative action properties. This discussion enables us to formulate a preliminary, general problem definition for this Ph.D. thesis. In the sections 1.2, 1.3 and 1.4, we explain what type of action properties we deem important for system specification and what problems may arise in a logic approach to the specification of such properties. In particular, section 1.3, discusses the problem of the discrepancy between the closed worlds of implemented systems and the open worlds that logics refer to. Here we argue

that in order to bring logic closer to the closed world of implemented systems, we need closed interpretations of sets of logic formulas. Section 1.4 focuses on the problem of reasoning about requirements from a normative perspective. Requirements are desired system properties. They are in that sense norms for a system. We argue that the properties of a logic used for specification to a large extent reflect the way a specifier rationalizes his design. In particular, we show that normative (deontic) logics may serve as convenient languages to express the requirements a specifier poses to a system, and that for certain situations the use of deontic requirements is actually inevitable. Then, having described the general specification and verification setting, and the types of action reasoning involved in it that we think are important, section 1.5 gives the central problem description for this Ph.D. thesis. Finally, section 1.6 explains our approach to the central problem, and discusses semantic structures, orderings and modalities. This section also functions as an overview of the work in this Ph.D. thesis.

1.1 Reasoning about reactive system properties

A reactive system is a system that continuously interacts with its environment and whose basic functionality is to maintain a certain behavior of its environment ([88]). This is what distinguishes reactive systems from mere transformational systems, for which only the result after termination matters. Concurrency is central to reactive systems in the sense that each reactive system by definition operates concurrently with its environment. A traditional position towards reactive system design is sketched by the following picture.

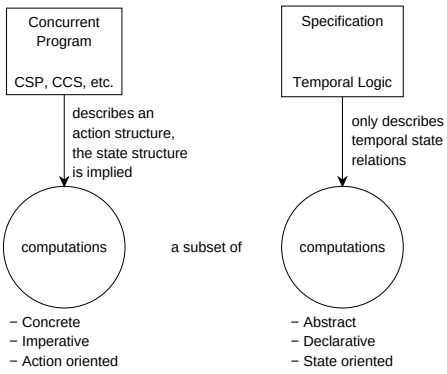


Fig 1. the traditional program-specification distinction

This is also the mental picture that Manna and Pnueli follow in their

book *The Temporal Logic of Reactive and Concurrent Systems* [124]. In the first part of this book, Manna and Pnueli define a concurrent programming language that includes constructs from many other well known languages for concurrency, such as CSP [97], CCS [138, 139], etc. This concurrent programming language contains assignment, skip and await actions at the atomic level, and selection, cooperation, while and block statements, etc. at the program level. It is interpreted over transition systems, and each program determines a set of computations (traces or trees through a transition system). Each concurrent program is meant to reflect a specific ongoing behavior in interaction with the environment. It is known that concurrent programs carry a much higher risk for unexpected and unintended behavior. This is one of the reasons why programs are verified against specifications of their intended behavior. For transformational programs, verification only concerns the relation between begin and end-states. But in case of reactive systems, we have to deal with ongoing behavior, for which end-states cannot be assumed to exist. So requirements for reactive systems may concern behavior that is infinite in the temporal dimension, which means that it cannot be verified in terms of a relation between begin- and end-states. This explains why for the specification and verification of reactive systems temporal logics are used [149]. Temporal logics can express requirements concerning infinitely ongoing behavior. Figure 1 depicts this traditional setting for reactive system verification. The program on the left determines a (set of) computation(s), by explicit prescription of an order in which actions (instructions) are executed. The structure of states (values of state variables) is ultimately defined through the transformations determined by the assignment, skip and await actions at the lowest program level. The temporal logic specification on the right determines the possible temporal (succession) relations between computation states. In these temporal specifications, nothing is said about actions or programs. A program is said to satisfy a specification if the computations interpreting the program are a subset of those interpreting the specification. There can be many different programs satisfying the same specification.

In this traditional picture, 'reasoning about requirements' is not really an issue. First of all, the verification of a program against its specification can hardly be seen as a form of reasoning, since the program is (1) not written in a logical language, and (2) constitutes the system itself: it is not a property or set of properties of the system. Second, there is no concern about entailment relations *between* temporal specification formulas (requirements).

To position the work in this thesis, in figure 2 we sketch an alternative mental picture. Our mental picture stays at a more abstract level and does

not refer directly to an implementation at all. It refers to the level where a designer thinks about his design by considering possible system properties and the logical entailment relations between them. Therefore the picture of figure 2 should be seen as a refinement, and at the same time, as an extension of the right side of figure 1, the side that is only concerned with specification properties.

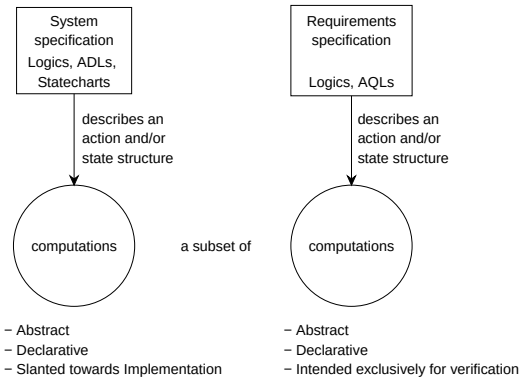


Fig 2. an abstract view on specification

System properties may entail one another, and entailed properties should not be included in a system specification. This motivates the distinction between *system specification properties* (the left side of figure 1) and *requirement properties* or *verificational properties* (the right side). Implementations are derived from system specification properties, which is why we say that the properties to the left are more slanted towards implementation. We want any representation of system specification properties to be as concise as possible. For instance, as part of our system specification we do not want properties that are logically redundant. Logically redundant properties are typically properties that might be taken in consideration as verification properties.

This picture embodies a much more liberal setting than that of figure 1. We mention a few possibilities to fill in more detail:

- The system specification and the requirements specification are stated in the same logical language. In this case, the subset relation of the computations (models) coincides with 'logical entailment'. Checking system properties for mutual entailment relations or inconsistencies falls under this picture.

- The system specification is stated in an action description language (ADL), which is a language in which to describe action domains, and the

requirements specification in an action query language (AQL), which is a language in which to specify action properties that need to be proven of an action domain description ([114]). Now the subset relation is not logical entailment in the traditional sense, since we deal with two *separate* logic languages.

- The system specification is a set of properties determining one particular transition system, and the requirements specification is a set of temporal properties. In this case the subset relation is verified by means of model checking. This picture is closest to the traditional one of figure 1, the only difference being that the transition system is not described by a concurrent program but by a (declarative) specification.

- The system specification is a statechart design (a graphical language for modeling reactive systems [84, 10]), and the requirements specification a temporal logic. This is again a setting that is close to the traditional one. Much depends on whether statecharts are considered a programming language or a specification language. We do deal with temporal properties in chapter 3, but do not treat the theory of statecharts in this Ph.D. thesis.

- The system specification is the combination of a functionality description in modal action logic combined with a description of a normative environment, and the requirements specification is a set of normative requirements stated in a deontic action logic.

As said, the situation in figure 2 can be seen as both a refinement and an extension of the specification part in figure 1. It is a *refinement* in the sense that we make a distinction between two sets of specification properties: properties from which we want to derive an implementation (the left side of the picture), and properties that are used for verification (the right side of the picture). Of course, this situation can be repeated by defining a third specification that is even closer to implementation. This way we get a series of specifications that can be thought to evolve gradually towards implementation.

The picture in figure 2 is an *extension* of the one in figure 1 in the sense that properties concerning actions and concurrency are brought under the scope of specification, that is, for specification we move from an *endogenous* logic, where actions are not explicit in the language, to an exogenous one, where actions are explicitly represented in the syntax. Temporal logics, as used in the traditional setting, do not refer explicitly to actions or to concurrency, and are thus endogenous. In the traditional setting there is a strict

division: actions and concurrency are aspects of the program side, and temporal relations between computation states are aspects of the specification side. In our picture this division is blurred: actions and concurrency are considered to be supported by the languages in which a specifier states requirements and makes verifications. This comes with several problems concerning reasoning about (true) concurrency and reasoning under, for example, frame assumptions, topics that are covered by chapters 2 and 4.

The setting of figure 2 thus covers a variety of ways in which languages of logic can assist a reactive system designer in making verifications. It accounts for checking separate requirements for mutual inconsistencies or entailment relations (redundancies). And it does not preclude situations where we want to cross-verify requirements that are stated in different languages. Having sketched this general setting, we are in a position to formulate a first, very general problem definition for this Ph.D. thesis:

> Can we develop a uniform reasoning framework that incorporates and combines the forms of reasoning involved in the different types of verificational effort that follow from the schematic setting of figure 2?

As said, this is a very general problem description. In the following subsections we say something more about the types of reasoning we deem important in this context. Then, in section 1.5 we formulate a more refined problem description.

1.2 Actions, action combinators and time

A view on reactive system behavior as series of actions to be performed by a system, provokes a close examination of the action concept itself. A first issue is how the properties of complex actions relate to properties of the more elementary actions they are composed of. Complex (compound) actions are built from atomic actions by means of action combinators. We consider the combinators that are standard in relational formalisms such as relation algebra [174] and process formalisms such as dynamic logic [152, 59, 83] and process logics [87, 90], i.e. *choice*, *sequence*, *iteration*, *converse*, and some more specific constructs like *fail*, *any* and *test*. For our purposes, two specific action combinators deserve close examination: *concurrent action composition* and *action negation*. We mentioned in the preceding section that concurrency is central to the concept of reactive systems: reactivity is considered as action taking place concurrently with environmental action or other internal action. The notion

of 'action complement' in the interpretation of reference to 'alternative action' also arises as a crucial concept in the context of action specification. For instance, it is natural to consider the property that a certain effect is brought about exclusively by a certain action. Another way of saying the same thing is that any alternative action cannot have that effect. A second example of where the action complement arises as a natural concept is that of temporal reasoning over action. For instance, we should be able to conclude that the (liveness) property that over all possible futures eventually an action a is inevitable, is logically equivalent with the property that it is not possible to perform actions alternative to a forever. And finally, we use action negation in normative statements about action; for instance: 'an obligation to perform an action a implies the absence of permission to perform any action alternative to a'.

Although in figure 2 we sketched a specification setting where, contrary to the traditional setting in figure 1, not only temporal specification properties play a role, we do not want to dispute that temporal properties are very important for specification. To model the reasoning about the temporal evolution of reactive system properties, many temporal logics have been developed [149, 9, 47, 187, 24, 124, 48], some of which have incompatible conceptions of the structure of time. In chapter 3 we present our position regarding this issue.

1.3 Open worlds and closed systems

Leaving the traditional strict dichotomy between programs (implementations) and specifications, brings with it the possibility to view an implementation as the limit of the process of specification refinement. However, although this idea of a gradual movement towards implementation sounds nice as a goal or idealization, it neglects a fundamental difference between the way we look at implementations on the one hand and declarative logic specifications on the other. Languages for implementation (programming languages) are always viewed as describing a *closed* world: a world with a clear boundary (the system boundary) and a finite number of finitely describable states in each of which a finite number of exactly defined actions are possible that display no unforeseen non-determinism. If we interpret a language of logic, on the other hand, we assume a world that is *open*: the world of logic is unbounded, involves no limitations on the number of states and possible actions, and no action occurrence or effect is precluded. In this open world, in principle, *everything* is possible. Logic formulas are then interpreted only to dictate constraints on

this open world, in order to make it only slightly less open. But in the closed world of an implemented system, in principle *nothing* is possible; the only things possible being the things enclosed by the functionality of the system as described by its program. Take as an example the implementation description (program rule) '*if* $z = c$, *then* $y := z + 1$'. The classical interpretation assumes that we are in a specific completely defined program state (implicitly represented by the place of this rule in the program) and that if at this point variable z has value c precisely the following action is performed: 1 is added to variable z and the result is assigned to y. In a first-order modal action logic we might try to express this by: '$\forall y, z.\ (z = c) \rightarrow [\textit{var-update}](y = z + 1)$'. This formula says so much as: in states where the (pre)condition $z = c$ is obeyed, performing the action *var-update* always results in a state where the (post)condition $y = z+1$ is satisfied. Here we encounter the completely different world view we presuppose when interpreting a logic formula, because this is far from equivalent to the program rule. For instance the following things are left *open* by the logic formula: (1) it does not refer to a specific state from where the action is executed, but to a whole set of states by means of a general precondition $z = c$ (2) it does not guarantee execution of *var-update* if $z = c$ is satisfied, (3) it does not exclude performance of other actions if $z = c$ is satisfied (4) it is not excluded that in states where the precondition is not satisfied, actions with effects identical to that of *var-update* are executed (5) it does not exclude actions occurring simultaneously with *var-update* (6) it does not even say that the action *var-update* is actually possible (7) it does not exclude that variables other than y or z are also changed by performing the action (8) it does not say anything about in what *way* satisfaction of the postcondition $y = z + 1$ is to be brought about by the action *var-update*. For the program interpreter this is clear: it adds 1 to the variable z and assigns the result to variable y. But for the interpreter of the logic formula it is not clear: subtracting 1 from y and assigning the result to z (a program rule would say $z := y - 1$) also satisfies it.

Of course, for this particular case, we should be able to provide additional formulas, representing properties that decide on all that is left open. But in general this is a very complicated issue, and for this discussion, this observation is beside the point. The point is that to bring logic closer to implementation, we may have to find ways to decide on all the matters that are left open in the semantics of logic in a *default* way. This is the motivation for chapter 4, where we investigate *closed* interpretations of logic formulas by adding closure *assumptions* to their semantics. It is also interesting to consider the converse direction: make closed interpretations more open. In general we might say

that there are basically two ways to reconcile the two realms: work on the interpretation of logic formulas to make them more closed or allow more freedom in the semantics of programming languages in order to make them more open. In chapter 4 of this Ph.D. thesis we take the former approach.

1.3.1 Closure and compositionality

Using logic for specification comes down to listing logic formulas representing requirements that the system ought to satisfy. An apparent advantage of this declarative approach is that it is highly *modular*, since a specification consisting of a list of properties can be easily modified by removing and adding individual items in the list. Also the task of verifying that a system specification satisfies a requirement specification (alternatively: that an implementation satisfies a specification) can be done in a modular fashion by verifying each requirement separately. The ease with which the join operation is performed in logic specifications follows from the compositionality of the semantics of the conjunction operation of logic ($\wedge$). In programming languages on the other hand, compositionality is in general much harder to achieve. In many cases, the semantics of programming languages cannot be said to be compositional; in general it is very difficult to describe the behavior of two programs that are joined together in some way in terms of the behavior of the individual programs, especially if the join involves a concurrent composition of programs.

In general we can say that ease of conjoinment (of programs and specifications) depends on the compositionality of the semantics of the conjoinment operation. As said, the conjunction operation of logic $\wedge$ satisfies this desirable property. But under a *closed* interpretation of logic formulas, as argued for in the previous section, compositionality of the semantics of $\wedge$ is *not* maintained. To explain this we return to the example '$(z = c) \rightarrow [\textit{var-update}](y = z + 1)$'. We now give an example of a *closed* interpretation of this formula by assuming that we are in a state that satisfies the precondition $(z = c)$ and (1) from this state no other actions are possible, and (2) also no other actions occur simultaneously, and (3) the action '*var-update*' is actually the only action possible, (4) other variables than y and z are not changed by performing '*var-update*', and (5) variable z is left unchanged by performing '*var-update*'. Now if we conjoin this logic sentence with for instance the sentence '$(z = c) \rightarrow [\textit{var-update}](z = c + 5)$', and we stick to a closed interpretation for both individual clauses, we run into trouble. The second formula intends to describe an extra effect of the action '*var-update*'. But this immediately conflicts with the closed interpretation of the first formula that says

that z is left unchanged by the action '*var-update*'. This means that under a closed interpretation of the two individual clauses, their conjunction has no meaningful interpretation.

The example shows that under a closed interpretation of individual formulas, the semantics of logic conjunction is no longer compositional. Our solution to this problem is to apply the closure assumptions not to individual clauses but to the complete set of requirements after they are conjoined. This makes the *closure* of an interpretation the final step of the process of building a specification by composition of individual requirements. The composition itself is performed on clauses that have an open interpretation, and only as a final step the interpretation is closed.

1.3.2 Persistency and causality

In a slightly different from, in artificial intelligence (abbreviated AI, from now on) the subject of closed interpretations of specification formulas appears as the problem of how to reason about change and causality. One of the most discussed problems in this area is the infamous frame problem. In short the original frame problem can be described as follows: when describing effects of actions declaratively, we prefer to talk exclusively about what changes as the result of an action and do not want to describe explicitly and exhaustively what properties persist. This stance towards specification of action effects embodies an important aspect of going from the open world of declarative logic specification to the closed world of programming. But, we emphasize that persistency is only one of the issues relevant for the closure of open interpretations. Recall that in section 1.3 above we mentioned eight ways in which the formula $(z = c) \rightarrow [\textit{var-update}](y = z + 1)$ can be considered open. Furthermore, note that the frame problem would actually not be a problem at all if a reasoner reasoning about how changes are brought about by actions would adopt an *open* world view, where he accepts that actions always come with other (concurrent) actions and with possible additional effects.

In the AI-community the frame problem is often directly associated with the problem of how to reason about causality, and sometimes these two problems are identified. This should not come as a surprise, because both the classical frame problem and the causality problem concern the observation that change is not arbitrary: the frame assumption says that change does not occur unless it is specified, and causality says that change does not occur without a cause. However, in our view, the problems of reasoning about causality and reasoning under frame assumptions have a different scope. A

central problem in reasoning about causality is the absence of contrapositive reasoning: if a causes b, then the negation of b does not necessarily cause the negation of a. Example: from the assertion that turning a switch to the on-position causes a room to be lightened, it does not follow that darkening the room causes the switch to turn to the off-position. This irreversibility problem concerns reasoning about causal dependency relations between conditions, and is, in our view, strictly a problem of reasoning about ramifications (secondary effects) of action effects. Thus, the ramification problem is not interpretable in terms of the difference between open and closed world views. But it is true that the problem of how to reason about causality is closely intertwined with the problem of how to reason under frame assumptions. We focus closely on these issues in chapter 4.

1.4 Normative system properties

In this section we motivate our interest in the use of normative (deontic) reasoning in the context of system specification. The relevance of deontic logic follows from our intention to specify system properties at an abstract (design) level. First of all, we argue in section 1.4.1, that on this abstract level it is convenient to allow for normative properties. Second, in section 1.4.2, we argue that in environments that concern human activity it is also necessary. We also comment on some philosophical issues raised in the area of Deontic logic, and discuss whether they are relevant for normative reactive system specification. In 1.4.3 we discuss the difference between descriptive and prescriptive models, which both play a role in our view on reactive system specification. Normative models might be used prescriptively in the specification of reactive system functionality, and descriptively in for instance the description of external bodies of norms to be used in a juridical expert system. In 1.4.4 we recall that in the literature on deontic logic it is argued that the logic of the prescriptive use of norms differs from the logic of the descriptive use of norms. Since the intended use of our logics concerns both, we should say something about this difference and about how our logics deal with it. Finally, in 1.4.5 we give our view on the role of paradoxes in deontic logic.

1.4.1 A normative stance

For an AI-researcher studying agent rationality, the picture appears to be clear: his artifacts are considered to be intelligent, and he is trying to 'capture' intelligent behavior in his models. If an AI-researcher writes down $\varphi \rightarrow O(a; b)$,

he means that he is considering the situation where his artifact has an obligation (denoted by the operator $O(.)$) to do $a;b$ in case that φ. If the researcher claims that the formula $\varphi \rightarrow [a]O(b)$ is entailed, he claims something about the rationality of his artifact. So his formulas reflect the reasoning of the artifacts he is trying to develop. If the researcher uses these formulas descriptively, we are in the realm of what is called 'knowledge representation[1]'. If the researcher uses them prescriptively, we are in the realm of 'agent specification'.

However, a reactive system designer may write down exactly the same formulas, despite the fact that a reactive system is in general not considered an intelligent artifact. The reason is that it might be convenient to look at a reactive system *as if* it has intentional capabilities. Of course, this says more about the standpoint of the specifier than about the nature of the system. If a specifier writes down $\varphi \rightarrow O(a;b)$ he means that his system is obliged to perform $a;b$ in cases where φ is satisfied. And if he checks whether $\varphi \rightarrow [a]O(b)$ is entailed, he is performing a check on the coherence of his own view on the system. This small example shows that the logic does not model the reasoning of the system under design, but that of the specifier checking his design.

Despite the fact that for AI-researchers it is much more natural to study intentional notions, essentially this same view had to be argued for extensively in the philosophy of artificial intelligence. It was Daniel Dennett who emphasized the possibility of taking an *intentional stance* [53] towards (physical, artificial) systems. The intentional stance can be described as viewing a system or object as if it is intentionally motivated, that is, as if it has goals, beliefs, desires, intentions, and as if it is susceptible to norms. We explain the intentional stance by means of an example concerning a GPS-based navigation system for cars. GPS stands for 'Global Positioning System', a world-wide implemented satellite network for determining positions on the face of the earth. We may employ the following two views on such a navigation system.

1. We may view it as a *reactive system*. The choice for this modeling paradigm implies that our basic mental picture of the system is that it continually senses its environment: through GPS its position with respect to its environment, through a mobile GSM-link (with a central database) the conditions of its environment that concern traffic-jams and diversions, and through buttons or voice recognition the commands and other triggers given by the driver. Based on its information on relevant geographic properties of its environment (highways, streets, cities, countries, routes, etc.) and its position in it, it reacts by producing responses

[1] Note that under this heading not only representations of knowledge are studied.

in the form of route information on a display or through voice processing.

2. Adopting the intentional stance, we may view it as an *artificial intelligent agent*. It communicates with satellites, the driver, and through a GSM-link with central databases to update its knowledge concerning traffic jams and diversions. It is intentional in that it has a goal in the form of the destination of the driver. It may be pro-active in the sense that if it expects to enter a certain geographical area, it may update its knowledge concerning traffic jams and diversions for this area in advance. In this sense, the system can also be attributed autonomy: the driver does not have to instruct the system to perform such checks explicitly. And typically, the system makes *plans* to reach the destination. These plans are not static: they have to be revised in case of unforeseen incidence.

The assumption underlying the intentional stance is that if we want to represent information about something we can do this in one of many mutually incommensurable ways.

The concept of the intentional stance was brought up in the philosophy of AI to avoid discussions about whether we are right in ascribing genuine mental abilities to artificial devices that display intelligent behavior. Our motivation to bring up the intentional stance here puts a slightly different emphasis. Our reason for bringing it up is that the intentional stance is really *useful* on the rather high level of abstraction we consider descriptions of reactive systems. This same point was argued for by John McCarthy [128] (long before Dennett formulated his intentional stance), who suggests that mental attributes should be ascribed to systems whenever this is useful. McCarthy writes:

> 'Our general motivation for ascribing mental qualities is the same as for ascribing any other qualities, namely to express available information about the machine and its current state.'

Embracing the intentional stance towards modeling goes hand in hand with applying certain intensional (with an 's') logics (such as epistemic logics or BDI-logics) to reason about behavior. However, we do not advocate to introduce a whole plethora of intensional modalities and intentional notions from AI into reactive system specification[2]. We restrict ourselves to the normative modalities, and call the associated stance the *normative stance*.

[2]However, note that seeing reactive system behavior as action undertaken by it can also be considered a step in the direction of taking an intentional stance.

Adopting a normative stance towards system requirements amounts to adopting terms like 'violation' and 'compensating action' in the vocabulary for describing system properties. By using this terminology one accepts a priori that systems may actually violate prescriptions, and that it needs to be specified what has to be done if the system does so. We reiterate that this is only *a stance* towards specification, one that is quite different from the one one might be used to. It is *only* a stance, since the same system might be described *not* using terms like 'violation' or 'compensating action'. In that case the states where the violation occurs are seen as normal states that might be attended by the system during a normal course of activity. So we view the normative stance as just a convenient way (in some cases) of looking at system properties, one that is close to the way we talk about system properties anyway, as is exemplified by terminology like for instance 'fault-tolerant systems'. Adopting a normative stance means adopting a terminology involving deontic notions like *permission*, *prohibition* and *obligation*. But by stating that a system is *obliged* to perform a certain step, or *prohibited* to execute a procedure, we do not claim that the system is actually susceptible to this kind of normative notions; we do not claim that it 'knows' or 'decides' how to behave in such a normative context, as humans, and maybe, artificial agents might. Our use of normative terms like 'obligation' and 'prohibition' only expresses that we prefer to describe properties of the system using the normative stance. This again emphasizes that we are only concerned with a stance, one that is especially suited for the abstract level on which we want to talk about system properties.

In deontic logic the concept of a 'soft constraint' [102, 40] embodies one of the forms of appearance of the normative stance. In deontic logic, a 'soft constraint' is defined to refer to system requirements whose violation is not considered as something that lies *outside* the scope of specification, but as just another situation for which it may be specified how the system should react. However, there is another vivid use of the term 'soft constraint', under the fundamentally different interpretation of 'weak design choice'. This distinction between weak and strong design choices concerns the relative 'strength' or 'importance' of requirements. By adopting such a distinction we enter the world of preference, probability, conflict and defeasability[3], which we view as fundamentally different from the world of normativity. If a defeasible soft constraint is violated by a design, it may be dropped altogether, because more specific, valuable or accurate information takes its place. If, on the other

[3]In chapter 4 we encounter preference and defeasability in the context of action specification.

hand, a deontic soft constraint is violated by a design, it must be preserved as a requirement containing valuable violation information for which additional, 'contrary to duty' information may be specified. To avoid confusion with the non-deontic interpretation, in this Ph.D. thesis we simply refrain from using the term 'soft constraint'.

1.4.2 Normative models of system environments

In case the *environments* of reactive systems involve *human* activity, the normative stance is the most natural view, that is, if we want to model the environment in detail. Having the choice to violate norms is one of the main characteristics that makes us call humans and other intelligent systems 'autonomous'. It would appear rather artificial to try to describe autonomous behavior of environmental entities in a non-normative way.

Modeling the environments of reactive systems is essential if we want to verify and validate global properties that are relative to the behavior of the environment. In many cases environments are populated with human operators or other autonomous agents. Autonomous agents may behave differently then prescribed, i.e. they may violate norms. We want to be able to predict how the combination of system and environment reacts on these violations. We may for instance consider assumption / guarantee properties [1] that state what is guaranteed by the system provided the environment complies to certain normative properties, and how the system reacts if those properties are violated.

1.4.3 Description versus prescription

The distinction between 'normative' and 'non-normative' is not analogous to the distinction between 'descriptive' and 'prescriptive'. Typically physicists, discovering and revealing structures that are already there, are involved in description. System designers, specifying devices that are yet to be built, are involved in a process of prescription. AI-researchers do both: they use symbolic representations both to describe intelligent behavior as observed in human (inter)action (knowledge representation) and to prescribe it in design models for artificial agents (intelligent agent specification [188]). The use of the words 'descriptive' and 'prescriptive' in these sentences exemplifies the use in this Ph.D. thesis: the words refer to the *intended use* of a model, and do not imply anything about the content of a model or the form a model takes.

The distinction between normative (deontic) properties and other intensional properties (alethic, epistemic, etc.) has another source. A deontic

property expresses an obligation, permission or prohibition. It is tempting to assume that a set of deontic properties always comprises a *prescriptive* model. And in many cases it does. But we can think of exceptions. As said, it depends on the use that is made of such a model. For instance, as part of the design of a juridical expert system a developer may need to make a deontic model of a piece of legislation. Such a model of an existing body of normative rules is, from the point of view of the developer, a descriptive model. This makes it clear that when we talk about deontic properties, we are not *necessarily* in the realm of the prescriptive models: norms *are* prescriptions but can be *used* prescriptively or descriptively. And also: sentences expressing norms *are* descriptions but can be *used* prescriptively or descriptively. Summarizing, we draw the following picture.

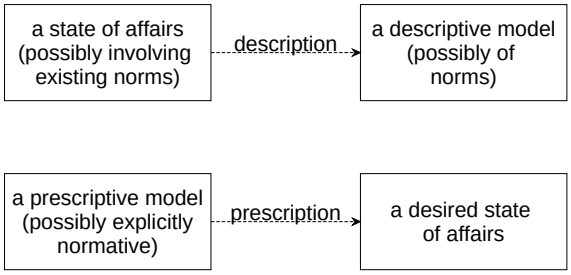


Fig 3. description and prescription

Above we stressed that system designers are essentially involved in a process of prescription, because they specify devices that are yet to be built. This holds for their enterprise as a whole. But by distinguishing between requirements properties and specification properties, as we did in section 1.1, we can nuance this picture: in the verification of requirement properties against specification properties, the former are used prescriptively, and the latter descriptively.

1.4.4 Norms versus norm-propositions

In the previous section we argued that normative models can be either descriptive or prescriptive. The intended use of the normative models in this Ph.D. thesis concerns both aspects. In deontic logic it is argued [193, 194, 2, 3] that the logic of the prescriptive use of norms differs from the logic of the descriptive use of norms. Von Wright writes [194]:

> 'An example might be "you may not park your car this side of the street". Used prescriptively it imposes a prohibition: used descrip-

> tively it gives information about existing parking regulations. In the prescriptive use the sentence does not say anything which is true or false. In the descriptive use it does.'

Von Wright uses the term 'norm' to refer to the prescriptive use of norms and 'norm-proposition' to refer to the descriptive use. Von Wright and Alchourrón emphasize that we should take the distinction serious, and that logics of norms are different from logics of norm-propositions. A new line of research that is motivated by the distinction between norms and norm-propositions is that by Makinson and Van der Torre [122, 123].

The first distinction between norm-propositions (descriptions of prescriptions) and norms (plain prescriptions) is that the former should be considered relative to an authority that has enacted them, while for the latter this is irrelevant. A distinction that is related to this observation is that norm-propositions are possibly conflicting, either because different authorities have enacted conflicting norms (e.g. national laws conflicting with international laws), or because a given body of norms (a legal system) contains conflicting information. Norms on the other hand, do never conflict: for instance, we cannot have both the norms "smoking is forbidden" and "one is obliged to smoke". These norms logically exclude each other. A second distinction concerns 'gaps'. In a context of norm-propositions, the normative information with respect to a state of affairs may be empty; norm propositions may be indifferent regarding certain situations. Norms, on the other hand, are usually assumed to be gap less; for each condition they either assume permission or prohibition. Finally, norm-propositions are said to bear truth-values, while norms are not. Norm-propositions refer to norms enacted by some authority, and may therefore be said to exist in the world. If a norm-proposition exists in the world, it is true. Norms, on the other hand, cannot be said to exist in the same way as norm-propositions do. Therefore they are said to lack truth-values. The following table summarizes the differences between norms and norm-propositions as mentioned by Von Wright and Alchourrón.

norm-propositions	*norms*
descriptive use	prescriptive use
relative to an authority that has enacted them	independant
conflicts are possible	consistent
indifference is possible	gapless
bear truth values	do not bear truth values

Table 1. norms versus norm-propositions

As said, the intended use of the deontic logics we develop is both descriptive (norm-propositions) and prescriptive (norms). Therefore we somehow have to deal with the differences mentioned in table 1. The relativity of norm-propositions with respect to an authority is not an issue that is very important in our context. The main authority is the specifier of a system, and in as far as external authorities are involved in the descriptive norms of external bodies of rules, they are not very relevant. But the issue of conflict / indifference versus consistency / gaplessness is of importance. The deontic logics of chapter 5 are adaptable to either circumstance. As for the issue of truth values, we adhere to the following standpoint. The concept of truth usually refers to states of affairs in the physical world. The deontic world is not a physical world, but a mental, abstract one. Therefore philosophers hesitate to call validity in this artificial world 'truth', notwithstanding the fact that we can formulate properties that are obeyed by the reasoning over the entities in this mental, abstract world. That is where Von Wright points at when he says ([194]) that maybe 'logic has a wider reach than truth'. We take the position that also validity in the artificial (platonic) deontic world can be regarded as truth, notwithstanding the apparent absence of the direct link with the physical world[4]. Anyway, we do not even have to adhere to the view that logic is about truth, it is enough just to say that logic is about entailment relations between meaningful sentences. Truth can be regarded an 'internal' variable of reasoning, a variable that is hidden from the reasoner all together.

1.4.5 On the paradoxes of deontic logic

To a large extent, the area of deontic logic has developed around a number of so called 'deontic paradoxes'. In as far the 'paradoxes' of deontic logic are seen as puzzles, we have no objections against this preoccupation with paradoxes. We subscribe to the viewpoint of Bertrand Russell, who writes in 'On denoting' [162]:

> 'A logical theory may be tested by its capacity for dealing with puzzles, and it is a wholesome plan, in thinking about logic, to stock the mind with as many puzzles as possible, since these serve much the same purpose as is served by experiments in physical science.'

[4]This means that we also do not support 'modal realism' [113], that says that we have to take the concept of a 'possible world' in modal logic semantics literally.

But many 'paradoxes' of deontic logic are relative to some deontic formalism in which they arise as a discrepancy with intuition or common sense. We object to calling such a discrepancy between an outcome of a model of normative reasoning and our assessment of the rationality of the outcome (common sense), 'a paradox'. We simply see these discrepancies as 'empirical[5]' refutations of the rationality of the normative reasoning model. So, in our view, many 'paradoxes' discussed in the deontic logic literature are no paradoxes, but just anomalies of the normative reasoning model under consideration. If the so called paradoxes of deontic logic would point to an inherently paradoxical nature of normative reasoning, we should give up hope of ever achieving a meaningful formalization. This motivates our choice to refer to the paradoxes of deontic logic as 'puzzles' or 'anomalies'.

The puzzles of deontic logic are relevant to us, in as far they are relevant for the more restricted context of normative reasoning about action. This means that some of the famous anomalies can be left unconsidered. For the ones that remain, the clear demarcation of our reasoning domain results in much more accurate descriptions of the problems. For instance: in many cases, such as in many formulations of the Chisholm anomaly and the Forrester anomaly, normative assertions concerning actions (ought-to-do) and situations (ought-to-be) ([64]) are intertwined, which obscures the central issue of the problem considerably. We only consider normative assertions about actions that are explicitly described in action languages.

1.5 Problem definition

The three foregoing sections on action combinators and time (section 1.2), closing interpretations of logic formulas (section 1.3) and deontic specification properties (section 1.4) discuss what type of issues and properties we deem relevant for abstract specification. With this in mind we can now reformulate the preliminary problem definition at the end of section 1.1.

> Can we define a logic framework for abstract reactive system specification that combines reasoning about all of the following aspects of action: (1) the way properties of actions that are composed by using action combinators (concurrent composition, choice, converse, etc.) relate to properties of their constituent parts (2) the relation between action and (discrete) time, (3) effects, indirect

[5]We use this word to stress the analogy with experiments in physical science, as Russell does.

> effects and enabling (executability, possibility) of actions, and (4) permissions, prohibitions and obligations regarding actions?

The development of such a formalism requires a reconciliation of the mentioned types of reasoning for the specific domain of functional reactive system requirements. The restriction to a limited reasoning domain, is clearly in support of the feasibility of the research. The general reconciliation problem seems much harder. The work by Van der Torre [179, 180] is an example of research concerning the general reconciliation problem of combining reasoning about defeasability and normativity.

But why should we aim for reconciliation? Why should we aim for a framework that indeed combines all mentioned types of reasoning? The first answer is that this is a natural implication of our intention to leave the traditional dichotomy between specification and programming. For instance, as a result of it we need to reason about concurrency, because we do not want this aspect to be the exclusive domain of the level of programs. And for exactly the same reason, we have to be able to reason about action effects, occurrences, dependencies between variables, etc. The second answer is that we simply want to be as complete as possible with respect to the types of reasoning involved in abstract system specification. In particular normative reasoning can be considered an important aspect of it, as we argued in section 1.4. The third answer is that the 4 reasoning areas described in the problem definition are not independent. An example is the Kantonian principle of 'ought implies can' and its contraposition 'cannot implies ought not'. Here we see how reasoning about possibility of actions, which concerns reasoning about action and change in the form of the qualification problem, interferes with normative reasoning. Another example is the interference between reasoning about concurrency and reasoning under frame assumptions. They interfere in the sense that separate minimal change action description assumptions for concurrent elements of a complex action cause that they cannot be conjoined concurrently. Actually for any combination of the four aspects mentioned in the problem definition, we can show dependencies of the reasoning. All of them are extensively discussed in the chapters to come.

1.6 A modal action logic approach

To account for the dependencies and interrelationships of the different reasoning types as mentioned in the problem definition, we need a common semantic ground. To this end we use modal action models (Kripke models supplemented

with action labels). In 1.6.1 we motivate this choice. We call the action logics that are interpreted on these models 'modal action logics'. The best-known modal action logic is propositional dynamic logic (PDL) [152, 59, 83]. Section 1.6.2 collects most of the modal operators that are defined throughout this Ph.D. thesis. These include operators for concurrency, time, action, and norms. In both sections 1.6.1 and 1.6.2 we refer to the chapters where the issues raised are worked out in detail. The sections can thus be read as overviews of the work in this Ph.D. thesis.

1.6.1 Semantic structures

Propositional logic (PL) is the logic of finitely many propositions about arbitrary domains. Since in PL nothing about the structure of domains is assumed, it does not bear on an abstract semantic representation of its domain of reasoning. First order logic (FOL), on the other hand, does assume a (rudimentary) structure of its reasoning domain by adopting a world view of relations and functions over arbitrary infinite sets of 'objects'. This semantic view of FOL is, in a sense, maximally abstract, and therefore, very generally applicable. The relations of FOL may be used to refer to such different real world-relations as temporal relations, spatial relations, causal relations, human relations, normative relations, authority relations, etc. This has led several influential researchers (e.g. J.A. Robinson [157]) to believe that FOL is the only logic we need in computer science.

From a strictly formal standpoint one cannot dispute the claim that FOL is all we need. But making such a claim is like claiming that for computation all we need are Turing Machines (assuming the Church-Turing thesis). There are actually two main reasons why FOL is sometimes *not* what we need. First of all it is often the case that FOL is much *more* then we need, because FOL is not decidable. Therefore, by using FOL, we cannot in general be sure that calculations terminate to reach a desired result. The second concern is the generality of FOL. As said, this is often claimed an advantage. But at the same time it is a weakness. For more specialized tasks, such as deontic, temporal and epistemic reasoning, mathematically alike semantic structures have emerged. Well-known examples of such common mathematical properties are reflexivity, transitivity, symmetry, etc. of binary relations used for the interpretation of modalities in intensional logics. Modal Kripke semantics are typically suited to model the reasoning connected to this type of formal properties of binary relations. Blackburn et al. call modal logic in their recent textbook on modal logic *the* logic of binary relations ([19]). For actions the

most used abstract semantic representation is that of a binary relation between execution and result state. This directly motivates the use of modal semantics in this Ph.D. thesis on action logics. We refer to the models we use as 'modal action models'.

Definition 1.6.1 (modal action models) *Given a countable set $\mathcal{A}$ of atomic action symbols with 'a' ranging over $\mathcal{A}$, and a countable set $\mathcal{P}$ of proposition symbols with 'P' ranging over $\mathcal{P}$, a modal action model $\mathcal{M} = (S, R^{\mathcal{A}}, V^{\mathcal{P}})$ over $\mathcal{A}$ and $\mathcal{P}$ is defined as follows:*

- *S is a non-empty set of possible states*
- *$R^{\mathcal{A}}$ is an action interpretation function $R^{\mathcal{A}} : \mathcal{A} \to 2^{(S \times S)}$, assigning a binary relation over $S \times S$ to each atomic action a in $\mathcal{A}$.*
- *$V^{\mathcal{P}}$ is an interpretation function $V^{\mathcal{P}} : \mathcal{P} \to 2^{S}$ assigning to each proposition P of $\mathcal{P}$ the subset of states in S for which P is valid.*

Any individual modal action model can be considered a high level abstract representation of a specific concrete (reactive) system. The states of a modal action model represent system states, the relations labelled with atomic action symbols represent actions taking the system from one system state to the other, and atomic propositions represent conditions that possibly change value going from one system state to the other. Note that the models are very close to transition systems, which are generally accepted as useful abstractions for reactive systems.

Throughout this Ph.D. thesis, we will use the modal action models of definition 1.6.1, with sometimes only some non-essential adaptations in cases involving initial states and norms. All logics defined in this Ph.D. thesis focus on different aspects of the models: logic properties of (1) the relation of pre- and post-conditions of complex actions with pre- and post-conditions of constituting actions (e.g. dynamic logic), (2) concurrency (intersection of accessibility relations), (3) action negation, (4) action description assumptions for effects (frame problems) and action possibilities (qualification problems), and ramifications, (5) action and (branching) time, and (6) norms (ought-to-do deontic notions).

Interpreting concurrency and complement on modal action models

A fundamental choice that has to be made in the modeling of concurrency is whether it should be regarded as *true concurrency* or be reduced to *interleaving*. The main advantage of a reduction through interleaving is that it

essentially brings back concurrency to the well-studied sequential situation. The choice for interleaving as a model for concurrency also has a very natural analogy in the actual working of computer systems: actions that on a user level seem to occur concurrently are actually interleaved on the processor level. However, on the abstract reasoning level we intend to stay, modeling concurrency by interleaving is not the most natural choice. First of all, we abstract completely from implementational issues. And whether concurrent actions are modeled through interleaving is in our view a choice regarding their implementation. Second, interleaved models of concurrency are exponentionally more extensive than truly concurrent models, which has its impact on the complexity of model-oriented reasoning tasks, such as model checking [86]. This makes it clear why we do not adopt the approach of Lodaya e.a. [117], who define a dynamic logic over modal action models in which concurrency is interpreted by interleaving.

Interestingly, it is also natural to consider the opposite: modeling interleaved activity through true concurrency. This gives independent motivation to be interested in reasoning about true concurrency. Consider a person reasoning about his time-schedule for the forthcoming week. He has to fit in several tasks in his schedule. As long as the order in which different tasks are performed during the day is not important, he can suitably apply the abstraction that a day is the minimal time unit, and that different actions performed during a day are performed concurrently. He will reason with propositions like 'I cannot go to the dentist, and attend the conference on the same day (in concurrency terms: the actions are mutually excluded)' and 'one of the days before the conference, I will have to prepare my lecture'. This means that logics that model true concurrency, such as developed in this Ph.D. thesis, also apply to reasoning tasks where we view actions *as if* they are concurrent.

We interpret true concurrency by intersection of action relations in the modal action models of definition 1.6.1. Intersection reflects true concurrency in a natural way: concurrency of actions is mirrored by the condition that the action relations that interpret them relate the same system states. But, the choice to model concurrency as intersection implies that we have to increase the expressiveness of modal languages to reason about concurrency. Standard modal logics preserve validity under bisimulation, which means that they are not strong enough to define intersection. These matters are discussed extensively in chapter 2.

The second main concern of chapter 2 is the interpretation of action complement on modal action models. In section 1.2 we motivated the importance of action complement. However, the interpretation of this notion on modal

action models is not straightforward. Semantics definitions found in the literature do not suffice. We define a notion of 'relativized action complement' that defines the relational space with respect to which the complement is taken to be relative to the action combinators in the modal action language.

Interpreting time on modal action models

As explained in section 1.1 temporal properties play a major role in system design, and are most likely to be found among the requirements properties (the right side in figure 2). The action models of definition 1.6.1 are actually fairly well-suited for the interpretation of temporal properties, although some details have to be taken care of. The states in the action models of definition 1.6.1 are primarily abstractions of system states, and not of time points. But, by looking at 'unravelings' of the action models, individual states are duplicated into possibly infinite sets that can be viewed as system states at certain points in time. A second concern is how to account for the intuition that the dimensions of time and dynamics are closely related: actions take place in time, and the notion of time is redundant without the possibility of action taking place. In chapter 3 we show for several modal action logics how to define the link between action and time by defining a next time relation R_X in terms of the action interpretation function $R^{\mathcal{A}}$. One of the problems we discuss concerns the definition of R_X for concurrent complex actions.

Interpreting norms on modal action models

The modal action models of definition 1.6.1 do not contain normative information. So if we want to interpret deontic languages, we need to introduce a normative realm in the models. There are several options. We introduce the different ideas here roughly, assuming some minimal knowledge of modal languages that can be interpreted on modal action models.

Meyer [135] defines violation states. He introduces a proposition V whose interpretation determines a set of 'violation states' for any model. Actions resulting in a violation state violate a norm, i.e. they either lack a permission, ignore a prohibition or neglect an obligation. A clear advantage of this abstraction is that it naturally reflects the intuition that the violation of a norm will bring a system in a special state: a state that under correct behavior should not have occurred. Following Meyer's approach, it follows that performing an action a from a certain state means violation of a norm if and only if in this state it holds that $\langle a \rangle V$. Van der Meyden [132] argues that this is not sufficient, and that it is more natural to associate violations with the

actions themselves rather than with the states they lead to. If we refrain from details, we may say that in Van der Meyden's approach, performing an action a from a certain state means violation of a norm if and only if in this state it holds that $\langle a^V \rangle \top$, where the action is annotated with a V to indicate that it concerns an action whose performance embodies a violation. Finally, in [36] we presented yet another approach, that simply takes the notion of 'it is a violation to perform action a' as a primitive proposition that is valuated over the set of possible states (we often call this the state-space in this Ph.D. thesis). Roughly, we may identify this with a proposition $V(a)$. Then performing an action a in a state means violation of a norm if and only if $V(a)$ holds in that state.

Now what are the differences between these approaches, and what are the relative (dis)advantages? Contrary to what is claimed by Van der Meyden, for the modal logic he is concerned with (propositional dynamic logic), his approach is equivalent to that of Meyer. There are many significant differences between Meyer's logic and Van der Meyden's logic, but these do not follow from the above choice concerning the representation of the violation primitive on the level of models. Any violation by performance of an action a in a model of Meyer corresponds one-to-one with a violation of an action a in a model of Van der Meyden. This is shown by a simple mapping for both formulas and models. Meyer's formulas are mapped to Van der Meyden's by moving violation propositions from the scope of modal operators to the atomic action 'within' them: $\langle a \rangle V$ becomes $\langle a^V \rangle \top$. This second formula is interpreted in Van der Meyden's logic in the obvious way: it is valid in a state whenever there is a V-annotated action relation leading to some other state. Now, since Van der Meyden's logic satisfies the tree model property, he is not right in claiming that his semantic representation of the deontic primitive gives rise to different logic properties: any state of a tree model that satisfies $\langle a \rangle V$ can be easily seen to correspond one-to-one with a state in a V-annotated tree model that is formed by 'moving' violations V from states to the action leading to them, and that thereby satisfies $\langle a^V \rangle \top$. This shows the equivalence of both approaches. But note that it only holds for modal action logics satisfying the tree property. In this Ph.D. thesis we consider many logics that do not satisfy this property. The main difference between the logics of Meyer and Van der Meyden does not concern the primitives, but the way normative assertions for complex actions relate to normative assertions for constituent parts. Using arguments from law, Van der Meyden chooses to define permission for a sequential composition of actions in such a way that it implies permission of all sub-actions (in chapter 5 we call such a permission a 'process permission'). But this type of choices is

independent of the choice for the semantic definition of a normative primitive.

There is a difference between the above two approaches and the approach with violation propositions of the form $V(a)$. Note that in Meyer's and Van der Meyden's approaches violation is coupled with the actual (in)possibility to perform an action. Since Meyer associates permission of a, denoted $P(a)$, with $\langle a\rangle\neg V$, the atomic action a is actually possible. In the approach with violation propositions $V(a)$, this coupling is absent. However, it can be argued that such couplings between normative assertions and qualification assertions (concerning the possibility of actions) actually reflect logic laws. We might view Kant's principle of 'ought implies can' as an example of such a law. However, we believe it is not correct to introduce this as a logical invariant. The Kantonian principle is more like an ethic directive for norm promulgation (it is morally / pragmatically wrong to impose norms that cannot possibly be obeyed), than like a property that is necessarily obeyed by any type of normative reasoning. It is not a logic necessity, because we can think of many examples where norms do not 'respect' capability (being obliged to pay one's debts, but not being able to). So, we can view the Kantonian principle as a norm about norm promulgation with roots in ethics, which means that for the application of norms to the specification of reactive system properties, we do not have to consider it to be a guiding logic principle.

All of the three mentioned approaches are adaptable to either one of the two cases discussed in section 1.4, i.e. to reasoning about norms or norm-propositions. If we want to interpret norm-propositions, we can distinguish between different types of violations: violations of a prohibition, of an obligation and of the absence of a permission. Details of these definitions are discussed in chapter 5. Also we might introduce seperate violation conditions for seperate bodies of norms or agents. But this lies outside the scope of the work presented in this Ph.D. thesis.

Closing in on modal action models

In section 1.3 we called the world of logic 'open' and the world of reactive systems 'closed'. In the open world of logic, a priori no restrictions are assumed, while in the closed system world the possibilities are restricted to that what is considered system behavior. We argued that modal action models are suitable semantic abstractions of reactive systems. In terms of modal action models, the openness of the world of logic corresponds with the view that no models are a priori precluded. The semantic equivalent of the closed world view is that only a small set of models are considered to be the actually intended

ones. Thus, under a closed view, many models of the open view are not taken into consideration because they represent all kinds of system behavior that is assumed not to occur. The most famous of these assumptions is the frame assumption. In chapter 4 we call such assumptions 'action description assumptions'. The application of action description assumptions can be seen as a form of closing the open interpretation of a set of action specification formulas. In chapter 4 we make action description assumptions explicit by defining *orderings* over modal action models. The models that reflect closed interpretations are defined as maximal or minimal elements over these orderings.

1.6.2 Modal operators

In this section we give a brief impression of the wide range of modal operators being defined throughout this Ph.D. thesis. Modal operators capture intensional notions such as possibility, necessity, obligation, globalness, etc. We divide the modal operators that are relevant to us are in three groups: (bare) modalities over complex action, temporal (action) modalities and normative action modalities.

First we consider modalities for reasoning about complex action. We call logics of action over the models of definition 1.6.1 'modal action logics' (MALs). Which action modalities are considered by a MAL depends on the action language. Other authors call logics of this type 'dynamic logics'. But in this Ph.D. thesis, we consider dynamic logic (PDL) to be a *specific* modal action logic, namely the one encompassing the action combinators sequence (;), choice ($\cup$), iteration (*), and converse ($^{\smile}$). We give names to modal action logics using expressions of the form MAL(X), where X is a specific set of action combinators. Under this convention PDL receives the name MAL($\cup, ;, ^*$). Modal operators have the form $[\alpha]\varphi$ or $\langle\alpha\rangle\varphi$, where complex actions α are built from atomic actions using the combinators specific for the action logic. Intuitively, the modal operators say: 'if it is possible to perform the complex action α it certainly results in a state that obeys φ' and 'it is possible to perform α in such a way that it results in a state where φ', respectively. Central in chapter 2 are MALs where the action language includes a *concurrency* combinator ($\cap$) and an action *complement* ($\sim$ or $\wr^I$). We consider many different action languages, and several alternative interpretations. The following quote by Krister Segerberg [166], who is one of the founding fathers of dynamic logic, supports us in our conviction that the choice for modal action logic (dynamic logic) to reason about reactive system properties is justified.

'Is dynamic logic a logic of action? It seems to this author that

> one might well say so. (...) Even so, it has to be admitted that dynamic logic lacks resources in the object language directly to express agency and ability. (...) But if dynamic logic is a logic of action, it is primarily a logic of computer action.'

The temporal operators that can be interpreted on the modal action models of definition 1.6.1 are studied in chapter 3. Since these temporal languages are interpreted on the same models as the modal action logics of chapter 2, we are in a good position to study the relations between these two types of reasoning. Apart from traditional temporal operators like $AG(\varphi)$ for 'on all possible futures it globally holds that φ', $AX(\varphi)$ for 'for all possible next moments in time it holds that φ' and $A(\varphi U \psi)$ for 'for all possible futures φ is preserved until ψ holds', we introduce $A(\varphi U \eta)$ for 'for all possible futures the condition φ is preserved until the action η occurs', $A(\eta U \varphi)$ for 'for all possible futures the action η is performed (repeatedly) until the condition φ holds', $A(\eta U \vartheta)$ for 'for all possible futures the action η is performed (repeatedly) until the action ϑ' occurs. In chapter 3 we also discuss a classification of liveness and safety properties in this context.

The deontic modalities over the modal action models we study in chapter 5 are divided into modalities for *goal norms* ($O_{\odot}(\alpha), P_{\odot}(\alpha)$ and $F_{\odot}(\alpha)$, where O, P and F stand for Obligation, Permission and Prohibition ('Forbiddeness') respectively), and modalities for *process norms* ($O_{\rightsquigarrow}(\alpha), P_{\rightsquigarrow}(\alpha)$ and $F_{\rightsquigarrow}(\alpha)$). The distinction is explained in detail in chapter 5, but roughly the distinction is that for process norms violations may occur during execution of complex (sequential) action, while for goal norms violations can only occur in the resulting states.

Chapter 2

Modal logics of action composition

In this chapter we study the modal action logics of various combinations of action combinators, over the models of definition 1.6.1. Some parts of this chapter, concerning action complement, have been published [27].

2.1 Modal action logic

The languages of modal action logic in this chapter are all of the form given in the following definition.

Definition 2.1.1 (modal syntax) *Taking 'α' to represent any compound action that can be constructed with a set of atomic action symbols $\mathcal{A}$ and a finite set of action combinators AC, and taking 'P' to represent arbitrary elements of a given countable set of proposition symbols $\mathcal{P}$, a* well-formed formula *φ of a modal action language $\mathcal{L}_{MAL}(AC)$ is defined by the following BNF:*

$$\varphi, \psi, \ldots \quad ::= \quad P \mid \top \mid \bot \mid \neg\varphi \mid \varphi \wedge \psi \mid \langle\alpha\rangle\varphi$$

This definition leaves open how compound actions α are built from atomic actions a. We consider several different sets AC of action combinators, and define the syntax of actions α for the corresponding modal action languages $\mathcal{L}_{MAL}(AC)$ separately. We define the usual syntactic extensions: $\varphi \vee \psi \equiv_{def} \neg(\neg\varphi \wedge \neg\psi)$, $[\alpha]\varphi \equiv_{def} \neg\langle\alpha\rangle\neg\varphi$, $\varphi \rightarrow \psi \equiv_{def} \neg\varphi \vee \psi$, $\varphi \leftrightarrow \psi \equiv_{def} (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$. In the following definition of the semantics of formulas

of a modal action language $\mathcal{L}_{MAL}(AC)$, we assume a general interpretation function R for the semantics of non-atomic actions α.

Definition 2.1.2 (modal semantics, validity) *The modal semantics is determined by the notion of validity of a formula φ in a state s of a model $\mathcal{M} = (S, R^{\mathcal{A}}, V^{\mathcal{P}})$, denoted $\mathcal{M}, s \models \varphi$. We implicitly assume here an extension of the interpretation function $R^{\mathcal{A}}$ for atomic actions a, to an interpretation function R for compound actions α.*

$$
\begin{array}{lll}
\mathcal{M}, s \models P & \textit{iff} & s \in V^{\mathcal{P}}(P) \\
\mathcal{M}, s \models \top & & \textit{for all } s \in S \\
\mathcal{M}, s \models \bot & & \textit{for no } s \in S \\
\mathcal{M}, s \models \neg\varphi & \textit{iff} & \textit{not } \mathcal{M}, s \models \varphi \\
\mathcal{M}, s \models \varphi \wedge \psi & \textit{iff} & \mathcal{M}, s \models \varphi \textit{ and } \mathcal{M}, s \models \psi \\
\mathcal{M}, s \models \langle\alpha\rangle\varphi & \textit{iff} & \textit{there is a } t \textit{ such that } (s,t) \in R(\alpha) \textit{ and } \mathcal{M}, t \models \varphi
\end{array}
$$

Validity on a model $\mathcal{M}$ is defined as validity in all states of the model. If φ is valid on a model $\mathcal{M}$, we say that $\mathcal{M}$ is a model for φ. General validity of a formula φ is defined as validity on all modal action models. The definition of R in terms of $R^{\mathcal{A}}$ depends on the set AC of action combinators in the modal action language. The validity relation for these separate languages is occasionally denoted $\models_{AC}$.

The semantics enforces that $[\alpha]\varphi$ holds in a state whenever all states reachable by α obey φ. From validity of formulas in a state we go via validity on a model and general validity to the definition of a modal action *logic*.

Definition 2.1.3 (modal action logic) *Given the class of modal action models of definition 1.6.1, a modal action language $\mathcal{L}_{MAL}(AC)$, and an interpretation $\mathcal{M}, s \models \varphi$ of formulas φ of this language on modal action models, the sub-set of all generally valid formulas is defined to be the (modal action) logic (relative to this language, semantics, and class of models). The logic is symbolically referred to as $MAL(AC)$.*

The important semantic notion of logic consequence (entailment) is defined as follows:

Definition 2.1.4 (entailment) *A formula φ globally entails a formula ψ (notation: $\varphi \models_G \psi$) if and only if all models for φ are also models for ψ.*

This is the global, model-based notion of entailment. Since it is model-based, it depends on the notion of validity on a model. Validity on a model is given in definition 2.1.2. For modal logics in general, there are other intuitive ways in which validity on a model can be defined. For instance, one might not want to adopt the above notion of validity because one objects against the entailment $[a]P \models_G [a][a]P$. An alternative is to define validity of models not as validity in all states, but as validity in a designated subgroup of (initial) states (or even in one particular point). For the resulting notion of entailment, that we denote by $\models_I$, the entailment $[a]P \models_I [a][a]P$ is not valid. This can also be achieved by switching to the local notion of entailment ($\models_L$): a formula φ locally entails a formula ψ if and only if state validity of φ implies state validity of ψ. For this local variant of entailment we also have $[a]P \not\models_L [a][a]P$ (for general models). In this Ph.D. thesis we mainly use the global version of entailment. When we use the entailment symbol without a subscript, we mean the global version. Pre- and postcondition reasoning typically makes use of modal action formulas of the form $\psi \to [\alpha]\varphi$. The precondition can be seen to function as a condition for the 'initial state' of the (complex) action α. Clearly we do not have $\psi \to [\alpha]\varphi \models_G \psi \to [\alpha][\alpha]\varphi$. This justifies the use of the global notion of entailment for pre- and postcondition reasoning.

In this chapter we consider a range of modal action logics, starting from the most basic one over atomic actions and its syntactic extensions, and increasing the expressive power by adding action combinators and alternative ways in which the action semantics interacts with the boolean reasoning in states. In doing so, we take a special interest in two operations on actions: *truly concurrent composition* and *action complement*. We argue that *intersection* of relations in MAL-models can be regarded an intuitive representation of true concurrency. We elaborate on the open interpretation of concurrency, and distinguish it from an open interpretation of action effects. We also briefly review some discussions in the literature on modal logics with intersection, and discuss properties of these logics. Our concern with the complement follows from its role in the expression of frame properties, temporal properties, and its relevance for the definition of deontic action notions. Extensions with complement are stronger than extensions with intersection, because for all versions of the complement we consider, the intersection operation is syntactically definable. The crucial semantic choice in adding *action complement* is to define the relational *space* with respect to which the complement is taken. A standard way to define complement is inherited from (binary) relation algebra [174]: with respect to the universal relation. We investigate the extension of modal action logics with this form of complement and discuss some of the logic prop-

erties it gives rise to. We argue that several of the resulting properties are not intuitive for modal logics of *action*. The source of the problems is that with the introduction of a complement with respect to the universal relation, we inherit the intrinsically *global* view from relation algebra. Modal logic has a *local* orientation. We define an alternative complement that is faithful to the *local* nature of modal action reasoning. The resulting (range of) modal action logics fit better with intuitions for action reasoning, and have better complexity properties.

Van Benthem [16] defined (propositional) modal logic to be exactly the bisimulation-invariant fragment of first order logic. A logic not satisfying the tree property is also not invariant under bisimulation. All logics of sections 2.4 and 2.5 of this chapter lack the standard tree property. Therefore, strictly speaking, according to Van Benthem's definition the range of logics we consider in these sections is *not* modal. Two other generalizations of the modal language to first order logic that preserve some desirable properties (e.g. decidability) are the 2 variables fragment and the (loosely) guarded fragment. But with one exception, all the logics we consider in 2.4 and 2.5 also do not belong to these fragments, because *transitive* modal operators form one of their central ingredients. It takes three variables to express transitivity in first order logic, and transitivity cannot be expressed by guarded formulas. Despite this discrepancy with traditional modal logic and its generalizations, it cannot be denied that the logics we define and study have a purely modal appearance. They all extend the minimal modal logic K, which implies (according the well-known axiom-based classification of what should be regarded as 'modal' logic), that they are all actually *normal* modal logics. All in all we conclude that it is not straightforward to find a place for the logics in the landscape of 'modal' logic categories.

2.2 Syntactic extensions of the basic language

The simplest action language is the one consisting of the countable set of atomic actions without action combinators. In the semantics for the modal action logic over this action language, we just interpret each atomic action through a relation $R^{\mathcal{A}}(a)$ in the models of definition 1.6.1. The logic is the weakest (minimal) modal logic K, extended with parameterization over modalities (each atomic action is assigned a separate modality). The logic is also known under the names 'Hennessy-Milner logic' and 'multi modal logic' (MML). In the notation of definition 2.1.3 the logic is referred to as MAL($\emptyset$) since it is the modal action logic with an empty set of action operators. The

first extension of the basic language we consider, is that with the action combinators *choice*, *sequence*, *test*, and the special actions *fail* and *skip*.

Definition 2.2.1 *Taking 'a' to represent arbitrary elements of a given set of atomic action symbols $\mathcal{A}$, the syntax of* the *syntactically definable actions is defined by the following BNF:*

$$\alpha, \beta, \ldots \quad ::= \quad a \mid \alpha \cup \beta \mid \alpha; \beta \mid \varphi? \mid fail \mid skip$$

These action combinators are given their usual interpretation.

Definition 2.2.2 *For the actions α of definition 2.2.1, the interpretation function R is given as a recursively defined extension of the interpretation function $R^{\mathcal{A}}$ for atomic actions:*

$$\begin{array}{lcll} R(a) & = & R^{\mathcal{A}}(a) & \text{for } a \in \mathcal{A} \\ R(\alpha \cup \beta) & = & R(\alpha) \cup R(\beta) & \\ R(\alpha; \beta) & = & R(\alpha) \circ R(\beta) & \\ R(\varphi?) & = & \{(s,s) \mid s \in S \text{ and } \mathcal{M}, s \models \varphi\} & \\ R(fail) & = & \emptyset & \\ R(skip) & = & \{(s,s) \mid s \in S\} & \end{array}$$

Roughly, action modalities $[\alpha]\varphi$ are interpreted by evaluating the condition φ in the 'final states' determined by the interpretation of the action α. But the semantics of the test-action $\varphi?$ breaks this principle: it defines an interaction of the action semantics with conditions in states that are possibly non-final. The test action 'generates' its own reflexive accessibility relation, and is possible whenever the tested formula holds in the current state. The empty action $fail$ is comparable to deadlock in process algebra ([7]). In process algebra deadlock is defined as the action δ that obeys the laws $x + \delta = x$ and $\delta \cdot x = \delta$. The corresponding formulas in the logic presented here are $[\alpha \cup fail]\varphi \leftrightarrow [\alpha]\varphi$ and $[fail; \alpha]\varphi \leftrightarrow [fail]\varphi$. It is not difficult to see that these are validity schemes given the syntactic extensions, which shows that the action $fail$ behaves as deadlock. Note that we have $[\alpha; fail]\varphi \leftrightarrow [\alpha][fail]\varphi \leftrightarrow [\alpha]\top \leftrightarrow \top \leftrightarrow [fail]\varphi$. The *skip* action is an action that is always possible (like the test it 'generates' its own accessibility relation) and loops in the current state, which means that it has no effect.

Proposition 2.2.1, below, implies that the basic multi-modal action logic is equivalent to the modal logic over the action combinators *choice*, *sequence*,

test, fail, and *skip*. The reason is that the operators only extend MAL(∅) syntactically, since each of them individually can be introduced as syntactic extensions. But first we define what it is to say that a modal action logic extends another logic syntactically.

Definition 2.2.3 (syntactic extensions) *Let AC and AC' be two sets of action connectives. If $\mathcal{L}_{MAL}(AC') \supset \mathcal{L}_{MAL}(AC)$, and there is a total surjective function $T : \mathcal{L}_{MAL}(AC') \to \mathcal{L}_{MAL}(AC)$ such that for any state s of any model $\mathcal{M}$ and any formula $\varphi \in \mathcal{L}_{MAL}(AC')$ it holds that $\mathcal{M}, s \models_{AC'} \varphi$ if and only if $\mathcal{M}, s \models_{AC} T(\varphi)$, we call $MAL(AC')$ a syntactic extension of $MAL(AC)$.*

Proposition 2.2.1 *$MAL(\cup, ;, \varphi?, fail, skip)$ is a syntactic extension of the logic $MAL(\emptyset)$.*

Proof

We can recursively define a total surjective function T taking formulas of $\mathcal{L}_{MAL}(\cup, ;, \varphi?, fail, skip)$ to formulas of $\mathcal{L}_{MAL}(\emptyset)$ as follows:

$$\begin{array}{lcll} T([\alpha \cup \beta]\varphi) & \equiv & [\alpha]\varphi \wedge [\beta]\varphi & \\ T([\alpha; \beta]\varphi) & \equiv & [\alpha][\beta]\varphi & \\ T([\psi?]\varphi) & \equiv & \psi \to \varphi & \\ T([fail]\varphi) & \equiv & [\bot?]\varphi & \\ T([skip]\varphi) & \equiv & [\top?]\varphi & \\ T(\varphi \wedge \psi) & \equiv & T(\varphi) \wedge T(\psi) & \\ T(\neg\varphi) & \equiv & \neg T(\varphi) & \\ T(P) & \equiv & P & \text{for } P \in \mathcal{P} \end{array}$$

Given the semantics of the operators, in definition 2.2.2, it is straightforward to verify that this translation respects the condition that for any state s of any model $\mathcal{M}$ and any formula $\varphi \in \mathcal{L}_{MAL}(\cup, ;, \varphi?, fail, skip)$ it holds that $\mathcal{M}, s \models_{\cup,;,\varphi?,fail,skip} \varphi$ if and only if $\mathcal{M}, s \models_{\emptyset} T(\varphi)$.

■

We call the operators choice, sequence, test, fail, and skip *syntactically definable* in the logic MAL(∅). We can add all action combinators that are syntactically definable in the basic modal language as a syntactic extension. But as soon as we want to add an operation on actions that is *not* syntactically definable, the possibility to define *any* of the connectives as a syntactic extension is blocked. This is what happens if for instance we add the iteration

or converse, as we do in the next section. Given the linearity of the above translation, clearly the logic with syntactic extensions has the same properties as the basic multi modal logic MAL($\emptyset$). The following table summarizes some well-known properties of this logic.

MAL variant	complexity of sat.	modal axiom.	f. m. p.
$(;, \cup, \phi?, fail, skip)/(\emptyset)$	PSPACE [110]	[111]	yes

Table 2. some properties from the literature on MAL($\emptyset$)

The finite model property (f. m. p.) says that any satisfiable formula can be satisfied in a finite model. If additionally, the size of these finite models is bounded by the formula size, we get the small model property. The small model property implies that the number of (relevant) models for a formula is finite and bounded by its size. The small model property thus brings with it decidability of the satisfiability problem. It directly gives a naive decision procedure: check all models. This is possible in principle: the finite model property says that models are finite, and the small model property says that there are only finitely many models, given the bound with respect to the formula size. But there is another reason why the finite model property is important to us. In chapter 4 we define orderings over modal action models in order to select intended ones. In the definition of these orderings and in the results obtained for them we require models to be finite. Also, the selection of intended models is motivated by the wish to be able to construct them and perform model checking. Clearly, for construction, finiteness is a prerequisite.

2.3 Dynamic Logic

The best known modal action logic is propositional dynamic logic (PDL, [59, 83]). The original PDL, defined by Pratt [152], introduces two extra connectives to the modal action language defined in the previous section: *iteration:* * and *converse:* $^\leftarrow$.

Definition 2.3.1 *Taking 'a' to represent arbitrary elements of a given countable set of atomic action symbols $\mathcal{A}$, the syntax of regular actions is defined by the following BNF:*

$$\alpha, \beta, \ldots \quad ::= \quad a \mid \alpha \cup \beta \mid \alpha; \beta \mid \varphi? \mid \alpha^* \mid \alpha^\leftarrow$$

Definition 2.3.2 *The semantics of (c-)PDL follows from the modal semantics of definition 2.1.2, and an extension of the relational interpretation function R of definition 2.2.2 such that it includes the following equalities for * and $^{\leftarrow}$:*

$$\begin{array}{lcl} R(\alpha^*) & = & (R(\alpha))^* \\ R(\alpha^{\leftarrow}) & = & \{(s,t) \mid (t,s) \in R(\alpha)\} \end{array}$$

Both the connectives iteration and converse are not syntactically definable in the basic modal language, and can thus not be introduced through a syntactic extension.

Proposition 2.3.1 *The operators $\langle \alpha^{\leftarrow} \rangle \varphi$ and $\langle \alpha^* \rangle \varphi$ are not syntactically definable in the basic modal action logic MAL($\emptyset$).*

Proof

For the logic MAL($\emptyset$) state-validity is preserved under taking generated sub-models ([19]), that is, if a formula φ is satisfied in a state s of a model M, we can eliminate all parts of the model that are not in the reflexive transitive closure over all action relations from s, without destroying validity in s. But, for instance the formula $\langle a^{\leftarrow} \rangle P$ is not always preserved under taking a generated sub-model: if we contract a model to the part corresponding to the reflexive transitive closure over all action relations from a satisfying state s, actions a entering the state coming from another state where P holds, are eliminated.

The logic MAL($\emptyset$) is (semantically) compact. This means that any infinite set of MAL($\emptyset$)-formulas is satisfiable if all its finite sub-sets are. The operator $\langle \alpha^* \rangle \varphi$ is not (semantically) compact. Every finite sub-set of the set $\{\langle a^* \rangle \neg P, P, [a]P, [a][a]P, \dots\}$ is satisfiable, but the infinite set itself is not. From the compactness of MAL($\emptyset$) it follows that if the operator $\langle \alpha^* \rangle \varphi$ were syntactically definable in MAL($\emptyset$), the infinite set would be satisfiable. Clearly it is not.

■

In the previous section we extended the basic modal language with the action operations 'sequence', 'choice', 'test' and 'fail' through simple syntactic extensions. But as soon as *one* non-syntactically definable operator is introduced (as a real extension of the logic), *none* of these in the basic modal language syntactically definable operators can be introduced through a simple syntactic extension anymore (*skip* and $fail$ are still definable in terms of $\varphi?$,

but the test itself is not longer syntactically definable). This is seen by inspection of, for example, the formula $[(a;b)^*]\varphi$. We cannot use the reduction $[a;b]\varphi \equiv_{def} [a][b]\varphi$ for a syntactic reformulation in basic multi-modal logic, since the sequence is embedded in an iteration, which in turn cannot be reduced with the help of a syntactic extension. So as soon as a non-syntactically definable operation is allowed, properties such as $[a;b]\varphi \equiv_{def} [a][b]\varphi$ cannot any longer be used to define the program constructs through syntactic extensions. But the properties reappear as axioms in the Hilbert-style deductive system of the logic. The following axioms and rules form a sound and (weakly) complete Hilbert-style deductive system for c-PDL [144].

Axioms:
any axiomatization of
propositional logic

K	$\langle\alpha\rangle\varphi \wedge [\alpha]\psi \rightarrow \langle\alpha\rangle(\varphi \wedge \psi)$
SEQ	$[\alpha;\beta]\varphi \leftrightarrow [\alpha][\beta]\varphi$
DISJ	$\langle\alpha \cup \beta\rangle\varphi \leftrightarrow \langle\alpha\rangle\varphi \vee \langle\beta\rangle\varphi$
IT	$[\alpha^*]\varphi \rightarrow (\varphi \wedge [\alpha][\alpha^*]\varphi)$
IND	$\varphi \wedge [\alpha^*](\varphi \rightarrow [\alpha]\varphi) \rightarrow [\alpha^*]\varphi$
FORW-SYMM	$\varphi \rightarrow [\alpha]\langle\alpha^{\leftarrow}\rangle\varphi$
BCKW-SYMM	$\varphi \rightarrow [\alpha^{\leftarrow}]\langle\alpha\rangle\varphi$
TEST	$[\psi?]\varphi \leftrightarrow (\psi \rightarrow \varphi)$

Rules:
Modus ponens: $\frac{\varphi, \varphi\rightarrow\psi}{\psi}$ Modal generalization: $\frac{\varphi}{[\alpha]\varphi}$

The axiom IT is sometimes (Harel [85]) identified with $[\alpha^*]\varphi \leftrightarrow (\varphi \wedge [\alpha][\alpha^*]\varphi)$, but the axiom IT as formulated above is sufficient for a (weakly) complete axiomatization ([109]). The following table summarizes important properties of modal action logics with converse or iteration. In particular the iteration is responsible for an increase of complexity with respect to the basic case.

MAL variant	complexity of sat.	modal axiom.	f. m. p.
$(;,\cup,^{\leftarrow})$	PSPACE (thrm. 2.3.2)	[144, 108] (implicitly)	yes
$(;,\cup,^*,\varphi?)$	EXPTIME [59]	[144, 108]	yes
$(;,\cup,^{\leftarrow},^*,\varphi?)$	EXPTIME [71]	[59]	yes

Table 3. some results from the literature on dynamic logics

The logic MAL$(;,\cup,^{\leftarrow})$ is one of the simplest *real* extensions of the basic modal logic MAL$(\emptyset)$. It is a real extension because in MAL$(\emptyset)$ converse is not syntactically definable, as we saw. But the complexity of MAL$(;,\cup,^{\leftarrow})$ is not higher than that of MAL$(\emptyset)$. This result can be obtained by adapting the approach of De Giacomo [71] who showed that we can define a polynomial translation of formulas of MAL$(;,\cup,^{\leftarrow},^{*})$ (converse propositional dynamic logic, or c-PDL, in his terminology) into MAL$(;,\cup,^{*})$ (propositional dynamic logic, or PDL, for short) that preserves satisfiability.

Theorem 2.3.2 *The satisfiability problem for MAL$(;,\cup,^{\leftarrow})$ is PSPACE.*

Proof

Using the notation $C(\varphi)$ to denote the minimal set of formulas that (1) contains φ, and (2) is closed under taking sub-formulas, and $aa(\varphi)$ to denote the set of atomic actions occurring in φ, we translate formulas φ of $\mathcal{L}_{MAL}(;,\cup,^{\leftarrow})$ to formulas of $\mathcal{L}_{MAL}(;,\cup)$ as follows:

1. Translate φ into φ_1 by pushing down all occurrences of the converse operation in φ to the atomic action level according to the following rewrite rules: $(\alpha;\beta)^{\leftarrow} := \beta^{\leftarrow};\alpha^{\leftarrow}$, $(\alpha\cup\beta)^{\leftarrow} := \alpha^{\leftarrow}\cup\beta^{\leftarrow}$, $(\alpha^{\leftarrow})^{\leftarrow} := \alpha$.

2. Translate φ_1 into φ_2 by replacing all occurrences of $a^{\leftarrow}$ by new atomic action denotations a'.

3. Define $\chi(\gamma,a) \equiv_{def} (\gamma \rightarrow [a]\langle a'\rangle\gamma)\wedge(\gamma\rightarrow[a']\langle a\rangle\gamma)$ and construct the formula $\nu := \bigwedge_{a\in aa(\varphi_1)}\bigwedge_{\gamma\in C(\varphi_2)}\chi(\gamma,a)\wedge[\bigcup_{a\in aa(\varphi_1)} a\cup a'](\bigwedge_{a\in aa(\varphi_1)}\bigwedge_{\gamma\in C(\varphi_2)}\chi(\gamma,a)\wedge[\bigcup_{a\in aa(\varphi_1)} a\cup a'](\ldots))$. The nesting in this formula has to be of sufficient modal depth. To find a sufficient depth, count the number of occurrences of the sequence operation (;) in φ_2 and add to this the maximal nesting depth of modal operators in φ_2.

4. Take $\varphi' := \varphi_2\wedge\nu$.

Step 1 of the above translation clearly preserves satisfiability. Actually, from the semantics of the action connectives it follows that it even preserves state-validity. In step 2, atomic converse actions are simply seen as non-converse atomic actions with a specific name. This makes it possible to leave the language of MAL$(;,\cup,^{\leftarrow})$ and use that of MAL$(;,\cup)$. Clearly the translation after step 2 does not preserve state-validity, since completely new actions are introduced. But it is also not sufficient to preserve satisfiability, since we have *lost*

the logic laws that relate a and a' as being each others converse. This is what is added in step 3. Intuitively, the formula ν ensures that the truth conditions of the formulas $FL(\varphi)$, that all play a role in establishing the truth condition for φ, are susceptible to the logic laws that hold for converse. The formulas $\chi(\gamma, a)$ in ν are recognizable as instantiations of the standard modal logic axioms $\gamma \to [\alpha]\langle\alpha^{\leftarrow}\rangle\gamma$ and $\gamma \to [\alpha^{\leftarrow}]\langle\alpha\rangle\gamma$. Step 4 gathers the conditions of steps 2 and 3, which together ensure preservation of satisfiability. As in the proof by De Giacomo [71], this is formally proven by straightforward induction over the structure of formulas. Being linear, the above translation demonstrates that the complexity of MAL$(;, \cup, ^{\leftarrow})$ is equal to that of MAL$(;, \cup)$, and thus PSPACE.

■

Note that this does not contradict proposition 2.3.1, because only satisfiability is preserved under the translation, and not validity.

2.4 True concurrency

In this section we strengthen the languages of the previous sections by adding an action connective representing true concurrency. True concurrency is the 'uninterpreted' form of concurrency. That is, there is no reduction to for instance interleaving. Here we study logic properties of true concurrency as an action connective in modal action logics. In the introduction we emphasized that logic takes an open world view. We first consider the consequences of this open world view for the interpretation of true concurrency in modal action logics.

2.4.1 Open action interpretations

The usefulness of the concept of *open action interpretations* is recognized in the literature on Deontic Logic [55, 54, 56], the literature on logics for concurrency [70], and in the AI literature on reasoning about action and change [75]. Here we distinguish two ways in which the interpretation of actions in modal action logics can be open: openness with respect to the description of effects and openness with respect to the description of concurrency.

First we consider openness of effect descriptions. In an open action effect interpretation, the effects of actions are only partially described, by stating logic postcondition formulas representing properties that are obeyed by the effects. In a closed interpretation, effect descriptions are complete in the sense that they completely describe the state that results after performance of an

action. It is explanatory to consider this distinction in companion with the distinction between exogenous and endogenous (temporal) logics. Exogenous logics assume actions to be implicit. The difference between situations / worlds / states / time points is assumed to be brought about by actions that are not explicitly represented in the logic language. For these logics an open view with respect to action effects is the only possible view, because the distinction makes no sense if actions themselves are not represented in the language. In endogenous logics, where actions are explicit (such as the modal action logics studied here) we *do* have a choice between an open or closed action effect interpretation. Examples of modal action logics with *closed* action effect interpretations are first order dynamic logic [83], that studies the logic of program operations over first order variable assignments, and the database update logics studied by Spruit et al. [170]. In this thesis we do not study logics of this type where the effects of actions are completely described. The way in which definition 2.1.2 defines the relation between actions and their effects is open. A formula like $[\alpha]\varphi$ specifies the effect of α to be anything, provided it satisfies φ. A closed reading reverses this: the effect of α is φ, provided that no other changes take place. The subject of closing the open interpretation of sets of action effect formulas is studied in chapter 4.

In the endogenous modal action logics we study we introduce an explicit action connective for true concurrency. For the moment, we assume that this operation is denoted by &. Introduction of such an operator is in principle independent of the choice between an open or closed interpretation of action effects. But under a closed interpretation of action effects in an endogenous action logic with a true concurrency operator, the logic relation between effect properties of, say, actions α, β and $\alpha\&\beta$ is not interesting. What can be the logic relation between the (full) description of the effect of $\alpha\&\beta$ and the (full) descriptions of the effects of α and β? If α assigns $A := true$ and leaves anything else as it is, and β assigns $B := true$ while leaving anything else unchanged, and $\alpha\&\beta$ assigns $A := true$ and $B := true$ while leaving anything else unchanged, then it is not clear how to formulate general logic properties that relate the effects. We cannot, for instance, adopt the logic relation that properties of effects of constituent actions carry over to properties of effects of the concurrently composed action, since action α requires that B is left unchanged, while $\alpha\&\beta$ does not. However, an open action effect interpretation enables the possibility to define different logic relations between effects of concurrent actions and effects of their constituent actions. For instance, in concurrent dynamic logic [148, 147], the following logic relation is studied: $[\alpha]\varphi \vee [\beta]\varphi \leftrightarrow [\alpha\&\beta]\varphi$. Below we argue that this property is too strong for our

purposes. But first we discuss the second way in which action interpretations can be open.

In the study of logic relations between effects of concurrent actions and effects of their constituent actions, we may apply the adjectives 'open' and 'closed' not only to action effects, but also to the concept of concurrency itself. Under a closed concurrency interpretation an action term α is interpreted as 'the action α in isolation, i.e., not concurrently with yet other actions'. An open concurrency interpretation is 'all concurrent actions that include α as a concurrent component'. The logic property that naturally arises by adopting an open concurrency interpretation is $[\alpha]\varphi \rightarrow [\alpha\&\beta]\varphi$. This is easily seen. If $[\alpha]\varphi$ means that φ holds after any execution of concurrent actions that involves α (the open concurrency interpretation), then it holds in particular after $\alpha\&\beta$. This open concurrency interpretation is easily seen to imply an open effect interpretation of actions: if an action term α is interpreted as referring to all possible concurrent action executions that involve α as a concurrent component, then we also have to accept that the effect of α cannot in general be completely described by formulating postconditions of α alone: other concurrently performed actions may be responsible for additional effects.

We adopt the open concurrency interpretation for our modal action logics. So we adopt the principle $[\alpha]\varphi \rightarrow [\alpha\&\beta]\varphi$. We need the axiom to be able to infer that concurrently composed actions cannot be performed if effects of contributing actions contradict. If there is no logical influence of the postconditions of constituent actions on concurrently composed actions, there is no way to force non-ability of concurrent performance on grounds of conflicting postconditions of actions contributing to the concurrent execution. Suppose for example that the actions α and β have mutually inconsistent postconditions. This is for instance the case if α represents setting a flag and β un-setting it. Then from this information we want to conclude that these actions cannot be performed simultaneously. So we want to make the inference $[\alpha]\varphi \wedge [\beta]\neg\varphi \models [\alpha\&\beta]\bot$. It is not difficult to see that this inference is enabled by the schema $[\alpha]\varphi \vee [\beta]\varphi \rightarrow [\alpha\&\beta]\varphi$.

The open concurrency interpretation requires a specific 'attitude' towards the reading of MAL formulas. For instance, under the open concurrency interpretation validity of the formula $\langle\alpha\rangle\top$ does not say that it is always possible to execute an action α in isolation: it may be possible that α can only be executed if it is accompanied by, say, β. And the formula $\langle\alpha\rangle\top$ does not decide on this issue. A formula $\langle\alpha\&\beta\rangle\top$ is interpreted to mean 'It is possible to execute an action that has both α and β as concurrent components'. Note, again, that this is entirely different from the *closed* meaning a programmer would at-

tribute to the construct & for concurrency. A programmer would look at the action (instruction) α as something that is performed completely in isolation. And the construct $\alpha\&\beta$ means to him that although α is accompanied by β, the two of them together are again executed in isolation.

From now on, we refer to the combined open concurrency / open effect interpretation as the 'open action interpretation'. As an example of reasoning with open action interpretations, we recall the 'bowl of soup problem' [75]. In natural language the context of the bowl of soup reasoning problem reads:

> A bowl of soup with two handles rests on a table. An agent is about to pick up the bowl of soup. It knows that if it takes the action of lifting a side of the bowl, the effect will be that that side is actually lifted. The agent also knows that in situations where one of the sides of the bowl is lifted while the other side is not, the soup is spilled.

Apparently, the problem setting is one in which actions can be performed concurrently. The agent always has to take into account the possibility that an individual action is performed concurrently with other, possibly unknown actions: in particular he accounts for the possibility that *right-lift* is performed concurrently with *left-lift*, that *left-lift* is performed concurrently with *right-lift*, but also that *right-lift* is performed concurrently with an action *left-lift* $'$ performed by some other agent, or even concurrently with an action x that has nothing to do with the bowl of soup. This underlines that the agent adopts a world view of open actions: actions may always occur concurrently to other actions, unless, of course, the agent *knows* (or *assumes*) that other actions *do not* occur concurrently. The agent, obviously wanting to avoid that the soup is spilled, needs to be able to reason with the knowledge as described above and draw conclusions about the spilling of soup as the effect of the different (concurrent) actions it can undertake. In particular, the reasoning should allow him to arrive at the following conclusions:

- If I take the concurrent action of lifting both the left and the right side of the bowl (possibly concurrent with yet other actions), the soup will not be spilled.

- If I take the action *x-lift* (x for *right* and *left*) of lifting one side of the bowl, and if I have no further information (or assumptions) about other actions performed simultaneously, it is always possible that the soup will be spilled. The corresponding formula is: $\langle x\text{-}lift\rangle Spilled$. I cannot

draw the conclusion that soup *will* be spilled, because the effect of the other side being lifted might be caused by an action that is performed concurrently with the agents' action. This might for instance be an action of a second agent that has spotted that something is about to go wrong. Or there may be a change in the world of which the source is unknown.

An action description AD concerning the actions involved in lifting the bowl of soup is:

$$
\begin{array}{ll}
AD = \{ & \\
[\textit{left-lift}]UpLeft, & [\textit{right-lift}]UpRight, \\
UpLeft \wedge \neg UpRight \rightarrow Spilled, & \neg UpLeft \wedge UpRight \rightarrow Spilled, \\
UpLeft \wedge UpRight \rightarrow \neg Spilled, & \neg UpLeft \wedge \neg UpRight \rightarrow \neg Spilled\}
\end{array}
$$

The intuitive conclusion to be drawn from this information is that by performing both actions concurrently, we do not spill the soup:

$$
\begin{array}{r}
AD \models_G \neg Spilled \wedge \neg UpLeft \wedge \neg UpRight \rightarrow \\
[\textit{left-lift} \,\&\, \textit{right-lift}]\neg Spilled
\end{array}
$$

But it is also important to take stock of what would be non-intuitive conclusions under an open concurrency interpretation. As said, we cannot conclude that the soup will be spilled by performing for instance *right-lift*. Another non-intuitive conclusion would be that if the concurrent execution forces absence of spilling, this absence would be forced already by performances involving only one of the two lift actions. We explicitly mention this as an undesirable property because in concurrent dynamic logic [148, 147], which obeys $[\alpha]\varphi \vee [\beta]\varphi \leftrightarrow [\alpha \& \beta]\varphi$, this would be a valid inference. Below we list these properties.

$$
\begin{array}{lll}
AD & \not\models_G & [\textit{left-lift}]\neg Spilled \\
AD & \not\models_G & [\textit{right-lift}]\neg Spilled \\
AD & \not\models_G & [\textit{left-lift}\&\textit{right-lift}]\neg Spilled \rightarrow \\
 & & ([\textit{left-lift}]\neg Spilled \vee [\textit{right-lift}]\neg Spilled)
\end{array}
$$

From our discussion we distill two desirable general properties for open true concurrency: (1) effects of actions that take part in a concurrent action performance carry over to the effect of the concurrently composed actions, and

(2) effects of constituent actions of concurrent actions may enhance each other (the effect of $\alpha\&\beta$ if $[\alpha](P \rightarrow Q)$ and $[\beta]P$) or make execution of the concurrently composed action impossible (the effect of $\alpha\&\beta$ if $[\alpha]P$ and $[\beta]\neg P$). This second property has an alternative formulation in: effects of concurrently composed actions cannot always be attributed to one of the contributing actions. In formulas these properties are expressed as:

$$\models [\alpha]\varphi \rightarrow [\alpha\&\beta]\varphi$$
$$\not\models [\alpha\&\beta]\varphi \rightarrow ([\alpha]\varphi \vee [\beta]\varphi)$$

2.4.2 Intersection in dynamic logic

The operation of intersection ($\cap$) on relations in modal action models can easily be seen to satisfy both properties argued for in the previous section. Yet, plain intersection as a representation for (true) concurrency in PDL has not been considered extensively in the literature.

Definition 2.4.1 *The semantics of (c-)IPDL follows from the modal semantics of definition 2.1.2, and an extension of the relational interpretation function R of definition 2.3.2 such that it includes the following equality for $\cap$:*

$$R(\alpha \cap \beta) \quad = \quad R(\alpha) \cap R(\beta)$$

PDL can be defined using an action trace semantics (an action trace is a finite concatenation of atomic actions) for programs. Every PDL-action α corresponds to a set of traces $\Sigma(\alpha)$. Atomic actions a constitute singleton trace-sets in themselves, the action connective $\cup$ is associated with union of trace-sets, ; with the possible concatenations that can be made with elements from the trace-sets, and * with the union of all finitely repeated self-concatenations of trace-set elements, respectively. It is not difficult to verify that $\mathcal{M}, s \models \langle\alpha\rangle\varphi$ if and only if there is a state t and an action trace $\sigma = \langle a_1, a_2, \ldots, a_n\rangle$ from s to t in $\mathcal{M}$ such that $\mathcal{M}, s \models \varphi$ and $\sigma \in \Sigma(\alpha)$. Also for c-PDL trace-sets are appropriate: we can add that individual steps within a trace may follow the converse direction of atomic actions. Trace semantics of modal action logics will be important in chapter 5, where we study deontic action logics. But the semantics of modal action logics with intersection cannot be defined in terms of action traces. Therefore, in the next section we generalize traces to so called 'action graphs' with a root and a sink. First we show that intersection embodies a genuine extension of the logics encountered so far.

Proposition 2.4.1 *The intersection (concurrency) operator $\langle \alpha \cap \beta \rangle \varphi$ is not syntactically definable in the basic modal language, nor in c-PDL.*

Proof

In both the basic modal language and c-PDL state validity of formulas is preserved under unraveling models into trees (from the state in which validity is considered)[1]. For c-PDL, the unraveling also involves the converse direction of atomic actions. State validity of for instance the formula $\langle a \cap b \rangle \top$ is not preserved under this model transformation.

■

The introduction of the intersection operator enables new forms of logic inference. For instance, for modal action logics with ';', '$\cap$' and '$^{\smile}$', the following are valid entailment relations:

CycleProp1	$[(\alpha;\beta) \cap \gamma]\bot \models_G [(\alpha^{\smile};\gamma) \cap \beta]\bot$
CycleProp2	$[(\alpha^{\smile};\gamma) \cap \beta]\bot \models_G [(\gamma;\beta^{\smile}) \cap \alpha]\bot$
CycleProp3	$[(\gamma;\beta^{\smile}) \cap \alpha]\bot \models_G [(\alpha;\beta) \cap \gamma]\bot$

It is easy to verify that these properties are valid[2]. The three properties reflect the so called 'cycle rules' of relation algebra [174].

In the next section we study the expressiveness of the intersection operator by considering modal definability (not to be confused with syntactic definability of the intersection operation, as for instance discussed in section 2.2). But before going into modal definability of the intersection operation, we shortly discuss complexity properties of $\cap$-logics. A positive result concerning the complexity of intersection as an addition to PDL is due to Danecki [52] who showed that for the general, non-deterministic case, satisfiability for IPDL or, in our notation, MAL$(;,^*,\cup,\cap,\varphi?)$, is decidable. The next table summarizes some results on $\cap$-logics that can be found in the literature. Note that many (possible results on) modal action logics with intersection are absent from the table: quite some research has not yet been carried out.

[1] The standard tree property is closely connected to the property of preservation of state validity under bisimulation: the unraveling is bisimilar to the original model. The tree property is seen as an important indicator for good complexity properties [79].

[2] Note that the entailment properties do not hold for the local ($\models_L$) or initial state ($\models_I$) variants of entailment.

MAL variant	complexity of sat.	modal axiom.	f. m. p.
$(\cup, \cap)$	PSPACE-compl. [119]	[8]	yes
$(;, \cup, \cap)$	unknown	[8]	yes
$(;, ^*, \cup, \cap, \varphi?)$	decidable [52]	unknown	no

Table 4. some results from the literature on $\cap$-logics

Note that the logic MAL$(;, ^*, \cup, \cap, \varphi?)$ is decidable although it does not obey the finite model property (f. m. p.). Decidability of MAL$(;, ^*, \cup, \cap, ^{\leftarrow})$ is yet an open question. It is tempting to think that we can prove decidability for this logic by once again adapting the proof of De Giacomo for the logic MAL$(;, ^*, \cup, ^{\leftarrow})$. The semantics of the $^{\leftarrow}$ and the $\cap$ tells us that they interact according to $(\alpha \cap \beta)^{\leftarrow} := \alpha^{\leftarrow} \cap \beta^{\leftarrow}$. So, presence of the intersection does not prevent us from performing step 1 and 2 of this proof: pushing down converse to the atomic level and renaming converse atomic actions (also the iteration forms no obstacle, since $(\alpha^*)^{\leftarrow} := (\alpha^{\leftarrow})^*$). But a complication arises in the third step. It is not clear whether we can find an appropriate extension of the definition of a Fischer-Ladner closure (a generalization of closure under sub-formulas for PDL) in the presence of $\cap$. We would have to prove that addition of the closure rule 'if $\langle \alpha \cap \beta \rangle \varphi \in FL(\psi)$ then $\langle \alpha \rangle \varphi \in FL(\psi)$ and $\langle \beta \rangle \varphi \in FL(\psi)$', is sufficient.

In the next section we develop the machinery of 'action graphs' to study some aspects of the expressiveness of $\cap$-logics. Without too much work, it should be possible to use this machinery also to prove complexity results for $\cap$-logics. Nevertheless we leave complexity issues for future work.

2.4.3 Definability of classes of models and frames

Definition 2.2.3 is about syntactic extensions, that is, how to define an action operation in terms of other action operations. This notion of definability is essentially a notion that concerns an *internal* issue of the logic, and it does not say anything about the expressiveness in terms of what can be said of *models*. Here we consider definability of certain classes of models and frames. This type of definability is completely different from definability of operators in terms of others: it takes a semantic view on a logic from 'the outside'. Before we define model and frame definability, we introduce the notions of 'frame' and 'frame validity'.

Definition 2.4.2 (modal action frames) *Given a countable set $\mathcal{A}$ of action symbols with 'a' ranging over $\mathcal{A}$, a modal action frame $\mathcal{F} = (S, R^{\mathcal{A}})$ is defined as:*

- *S is a nonempty set of possible states*
- *$R^{\mathcal{A}}$ is an action interpretation function $R^{\mathcal{A}} : \mathcal{A} \to 2^{(S \times S)}$, assigning a binary relation over $S \times S$ to each atomic action a in $\mathcal{A}$.*

Is is clear that a frame is turned into a model if an interpretation of propositional atoms is added. Such a model $\mathcal{M} = (S, R^{\mathcal{A}}, V^{\mathcal{P}})$ is said to be 'based on the frame $\mathcal{F} = (S, R^{\mathcal{A}})$'.

Definition 2.4.3 (frame validity) *Frame validity $\mathcal{F} \models \varphi$ is defined as follows: a formula φ is valid on the frame $\mathcal{F}$ if and only if it is valid on all models based on $\mathcal{F}$.*

Definition 2.4.4 (definability of model and frame classes) *A class of modal action models $\mathcal{C}$ is definable in a modal action logic MAL(AC) iff there is a set of formulas Ψ in $\mathcal{L}_{MAL}(AC)$ such that for all modal action models $\mathcal{M}$ it holds that $\mathcal{M} \in \mathcal{C}$ if and only if $\mathcal{M} \models_{AC} \psi$ for all ψ in Ψ.*

A class of modal action frames $\mathcal{D}$ is definable in a modal action logic MAL(AC) iff there is a set of formulas Ψ in $\mathcal{L}_{MAL}(AC)$ such that for all modal action frames $\mathcal{F}$ it holds that $\mathcal{F} \in \mathcal{D}$ if and only if $\mathcal{F} \models_{AC} \psi$ for all ψ in Ψ.

Definability of classes of frames is one of the concerns in 'correspondence theory' [16], that studies the expressive power of modal logics by taking an 'external', first order view on their semantics over Kripke-type frames. Correspondence theory deals with two types of questions: which first order definable properties on frames are definable in the modal language (modal definability), and which modally definable properties on frames are first order definable (first-order definability). Since our main logic concern is appropriateness of semantics, we only consider definability of the first type. Furthermore, our interest deviates from the one of correspondence theory in three ways: (1) we are not interested in the first order definitions of classes of frames, (2) we do not limit ourselves to *modal* definability, because we look at languages that are strictly stronger than the languages called 'modal' by Van Benthem, and (3) we are more interested in expressiveness in terms of definability of classes of *models* than in terms of definability of classes of *frames*. This last point needs some explanation. In a sense, the expressiveness in terms of frame validity is artificially strong: implicitly, frame validity quantifies over all possible valuations in all possible states. This quantification is intrinsically second order, which explains the high expressiveness at the level of frames. The frame level

is important for the study of expressiveness as such, but for our applications there is no a priori *semantic* relevancy for being interested in the expressiveness in terms of frame validity. Our main concern is with models, since we view these as abstract semantic representations of (reactive) systems. Let's consider the example of intersection. We argued that intersection reflects true concurrency in an intuitive way. Then it makes sense to demand of our logics that they define intersection at the level of *models*, that is, that we can for instance write down a logic formula that enforces that in all models (being abstractions of a reactive system) the actions a and b can (possibly under certain conditions) be performed concurrently. Definability of intersection at the level of frames is not enough to guarantee that we can impose such a condition on models using modal action logic formulas.

First we show that intersection models and intersection frames are both not definable in dynamic logics *without* the intersection operator. Similar results (for frames, and modal languages without converse) can be found in e.g. [98].

Proposition 2.4.2 *The class of models $\mathcal{C}_{R(\gamma)=R(\alpha\cap\beta)}$ and the class of frames $\mathcal{D}_{R(\gamma)=R(\alpha\cap\beta)}$ for which $R(\gamma) = R(\alpha) \cap R(\beta)$ are not definable in dynamic logic.*

Proof

Frame validity is preserved under taking bounded morphic images ($\mathcal{F}_2 = (S_2, R_2^{\mathcal{A}})$ is a bounded morphic image of $\mathcal{F}_1 = (S_1, R_1^{\mathcal{A}})$ iff there is a surjective function $f : S_1 \to S_2$ such that (1) for all $a \in \mathcal{A}$, if $(s_1, t_1) \in R_1^{\mathcal{A}}(a)$ then $(f(s_1), f(t_1)) \in R_2^{\mathcal{A}}(a)$, and (2) for all $a \in \mathcal{A}$, if $(f(s_1), t_2) \in R_2^{\mathcal{A}}(a)$ then there is a t_1 such that $(s_1, t_1) \in R_1^{\mathcal{A}}(a)$ and $f(t_1) = t_2$, see [19] p.139 and further for details). Thus, to prove non-definability of intersection frames, it suffices to show that the class of intersection frames is *not* closed under taking bounded morphic images. However, first we have to generalize the notion of bounded morphism, such that it also preserves frame validity for formulas involving converse action. We already saw in the proof of theorem 2.3.2, that by a restriction of converse to the atomic action level, we do not get weaker modal action logics (converse can always be pushed down to the atomic level without destroying validities). This shows that it suffices to add a clause that accounts for the converse direction of relations for atomic actions to the above definition for bounded morphisms: (2') for all $a \in \mathcal{A}$, if $(s_2, f(t_1)) \in R_2^{\mathcal{A}}(a)$ then there is an s_1 such that $(s_1, t_1) \in R_1^{\mathcal{A}}(a)$ and $f(s_1) = s_2$. With this generalization, the proof proceeds as follows. Take the intersection frame $\mathcal{F}$ from the class $\mathcal{D}_{R(\gamma)=R(\alpha\cap\beta))}$, with four distinct states r, s, t and u such that $R(\alpha) = \{(r, s), (u, t)\}$, $R(\beta) = \{(r, t), (u, s)\}$, $R(\gamma) = \emptyset$. It is easy to check

that the surjective function f for which $f(r) = v, f(s) = w, f(t) = w, f(u) = v$ to the frame $\mathcal{F}'$ with $(v, w) \in R(\alpha)$, $(v, w) \in R(\beta)$, $R(\gamma) = \emptyset$ is a generalized bounded morphism. But the frame $\mathcal{F}'$ is not in the class $\mathcal{D}_{R(\gamma)=R(\alpha\cap\beta))}$, which proves the proposition.

Model validity is preserved under surjective bisimulations (see [182], and chapter 4 for definitions). Thus, to prove non-definability of intersection models, it suffices to show that the class of intersection models is not closed under surjective bisimulations. Now any bounded morphism f for frames can be turned easily into a surjective bisimulation for models based on these frames, by demanding that for these models it holds that for any state s in the source model the valuation of atomic propositions is equal to the valuation in state $f(s)$ in the target model. Note that this means that for states in the source model that have the same morphic image, valuations are equal. Creation of two such models by addition of suitable valuations to the above example, demonstrates that the class of intersection models is not closed under surjective bisimulations, which proves the proposition.

■

The above proof suggests that the reason that the modal language of dynamic logic is not strong enough to define the class of intersection models, is that it cannot distinguish confluency of paths from non-confluency. This also follows from the tree property (used to prove proposition 2.4.1) for dynamic logic: from a specific state we can always unravel models into trees without destroying validity of formulas in that state. In trees there is no confluency of paths. It is a natural thought that this 'weakness' can be remedied by *explicitly* adding the intersection operation to the modal language, as we have done in the definition of $\cap$-logics. Clearly, the intersection operator *does* enable us to express that paths are confluent. And indeed, it is easy to prove that $\cap$-logics are strong enough to define intersection frames.

Proposition 2.4.3 *In $\cap$-logics the class of frames $\mathcal{D}_{R(\gamma)=R(\alpha\cap\beta))}$, for which $R(\gamma) = R(\alpha)\cap R(\beta)$ is defined by the set of formulas (formula scheme) $\langle\gamma\rangle\varphi \leftrightarrow \langle\alpha \cap \beta\rangle\varphi$.*

Proof

($\Rightarrow$) Immediate.

($\Leftarrow$) Suppose $\mathcal{F} \notin \mathcal{D}_{R(\gamma)=R(\alpha\cap\beta))}$. We have to show that $\langle\gamma\rangle\varphi \leftrightarrow \langle\alpha \cap \beta\rangle\varphi$ is not valid on $\mathcal{F}$. There are two cases. (case 1) there is a $(s, t) \in R(\gamma)$ while $(s, t) \notin R(\alpha \cap \beta)$. Now construct a model $\mathcal{M} = (S, R^{\mathcal{A}}, V^{\mathcal{P}})$ based on $\mathcal{F}$, such that $t \in V^{\mathcal{P}}(P)$ while for all u such that $(s, u) \in R(\alpha \cap \beta)$, it holds that

$u \notin V^{\mathcal{P}}(P)$. It follows that in state s it holds that $\mathcal{M}, s \not\models \langle\gamma\rangle P \leftrightarrow \langle\alpha \cap \beta\rangle P$. Thus, the formula is also not valid on the frame $\mathcal{F}$, which proves this case. (case 2) there is a $(s,t) \in R(\alpha \cap \beta)$ while $(s,t) \notin R(\gamma)$. The proof of this case is analogous to the proof for case 1.

■

However, it turns out that in the modal action languages enriched with the $\cap$-operator it is not possible to define intersection on the level of models. To prove this, we formulate a preservation theorem for the most expressive $\cap$-logic MAL$(;, ^*, \cup, \cap, ^{\leftarrow})$ that concerns a notion of semantical equivalence that is more refined than bisimulation. We call it 'action graph bisimulation' (ag-bisimulation). The notion of 'action graph' is a generalization of the concept of a 'well-nested graph', used by Danecki in his automaton-based proof of the decidability of IPDL [52]. In the theory we develop below, we leave out the test in order to avoid unnecessary complication. But Danecki [52] shows that the test can be suitably integrated in his graphs-based reformulation of dynamic logic (IPDL) semantics.

Definition 2.4.5 (rs-graphs and basic graph operations) *Given a set of action labels $\mathcal{A}$, a graph is a tuple (N, E), with N a set of nodes, and E a relation $E : N \times \mathcal{A} \times N$ defining edges. We call graphs $G_1 = (N_1, E_1)$ and $G_2 = (N_2, E_2)$ distinct if $N_1 \cap N_2 = \emptyset$. Now let r and s be two designated elements of N called the 'root' and 'sink' of a graph, respectively. Then, we call a graph (N, E, r, s) with $r \in N$ and $s \in N$, an rs-graph. We define the following operations on rs-graphs:*

- *Sequential composition: if $G_1 = (N_1, E_1, r_1, s_1)$ and $G_2 = (N_2, E_2, r_2, s_2)$ are distinct action graphs, then the graph $G_1 \cdot G_2 = (N, E, r, s)$ is constructed by imposing that $N := N_1 \cup N_2$, $r := r_1$, $s := s_2$, $s_1 = r_2$ (the nodes referred to by s_1 and r_2 are melted together), and $E := E_1 \cup E_2$.*
- *Parallel composition: if $G_1 = (N_1, E_1, r_1, s_1)$ and $G_2 = (N_2, E_2, r_2, s_2)$ are distinct action graphs, then the graph $G_1 \parallel G_2 = (N, E, r, s)$ is constructed by imposing that $N := N_1 \cup N_2$, $r = r_1$ and $r = r_2$ (the nodes referred to by r_1 and r_2 are melted together, and are in the composed graph referred to by r), $s = s_1$ and $= s_2$ (the nodes referred to by s_1 and s_2 are melted together, and are in the composed graph referred to by s), and $E := E_1 \cup E_2$.*
- *Reverse orientation: if $G = (N, E, r', s')$ is an action graph, then the graph $G^{\circlearrowleft} = (N, E, r, s)$ is constructed by imposing that $r := s'$ and $s := r'$.*

Definition 2.4.6 (action graphs) *An action graph is any rs-graph (N, E, r, s) that is an element of the minimal set satisfying the following conditions.*

- *The point graph (N, E, r, s) with $N = \{r\} = \{s\}$ (so, $r = s$) and $E = \emptyset$ is an action graph. We denote the point graph with the symbol Υ.*
- *For any action a such that $a \in \mathcal{A}$ the graph $G = (N, E, r, s)$ with $r \neq s$ and $N = \{r, s\}$ and $E = \{(r, a, s)\}$ is an action graph. We call such a graph an elementary a-graph, and denote it by the symbol Θ_a.*
- *If G_1 and G_2 are action graphs, then also $G_1 \cdot G_2$ is an action graph.*
- *If G_1 and G_2 are action graphs, then also $G_1 \parallel G_2$ is an action graph.*
- *If G is an action graph, then also $G^{\circlearrowleft}$ is an action graph.*

Throughout this chapter, we assume that action graphs are 'finite', that is, they involve a finite number of nodes (and thus, a finite number of edges). Finite action graphs have the following features: (1) they have a root and a sink (2) from the root we can reach any point in the graph through a finite number of atomic actions (3) from any point in the graph we can reach the sink through a finite number of atomic actions (4) loops are not excluded. This last mentioned feature follows from the observation that a loop can be formed as the result of a parallel composition of a non-point graph with the point graph. The following definition shows how to interpret actions as sets of action graphs.

Definition 2.4.7 (the graph interpretation of actions) *The graph interpretation $\Gamma(\alpha)$ of any action α over the connectives $(;, ^*, \cup, \cap, ^{\leftarrow})$, is defined as the set of action graphs obeying the following conditions:*

$$
\begin{array}{lcl}
\Gamma(a) & = & \{\Theta_a\} \\
\Gamma(\alpha \cup \beta) & = & \Gamma(\alpha) \cup \Gamma(\beta) \\
\Gamma(\alpha \cap \beta) & = & \{G_1 \parallel G_2 \mid G_1 \in \Gamma(\alpha) \text{ and } G_2 \in \Gamma(\beta)\} \\
\Gamma(\alpha; \beta) & = & \{G_1 \cdot G_2 \mid G_1 \in \Gamma(\alpha) \text{ and } G_2 \in \Gamma(\beta)\} \\
\Gamma(\alpha^*) & = & \{G_0 \cup (G_0 \cdot G_1) \cup (G_0 \cdot G_1 \cdot G_2) \cup \ldots \mid G_0 = \Upsilon \\
 & & \qquad \text{and } G_i \in \Gamma(\alpha) \text{ for } i \geq 1\} \\
\Gamma(\alpha^{\leftarrow}) & = & \{G^{\circlearrowleft} \mid G \in \Gamma(\alpha)\}
\end{array}
$$

If $G \in \Gamma(\alpha)$, we call G an action graph for α.

Note that among the action graphs that interpret an action α, there may be some that are loops. For instance, in the set interpreting the action $a \cap b^*$ there is a graph that is a parallel composition of the elementary a-graph Θ_a and the point graph Υ. The result is an a-loop.

Definition 2.4.8 (ag-relatedness of states) *Given a modal action model $\mathcal{M} = (S, R^{\mathcal{A}}, V^{\mathcal{P}})$ and an action graph $G = (N, E, r, s)$, we say that state u and v (possibly $u = v$) of S are 'G-related relative to $\mathcal{M}$', notation $u(G)^{\triangleright}v$, if and only if there is a function (homomorphism) $K : N \to S$ such that:*

- *$K(r) = u$ and $K(s) = v$*
- *for all n and m such that $n \in N$ and $m \in N$, it holds that if $(n, a, m) \in E$ then $(K(n), K(m)) \in R^{\mathcal{A}}(a)$*

Note that if two points in a model are G-related, it is not necessarily the case that the 'form' of the model that is 'in between' those states corresponds one to one to the form of the graph: all connections in the graph are required to be in the model, but not the other way round. In other words: we have a homomorphism and not an isomorphism. The right pointing triangle in the notation $u(G)^{\triangleright}v$ is a symbolic reference to the fact that we deal with a homomorphism. Note also that if an action graph is not a loop, it does not follow that the relational structure between to points in a model it is homomorphic to, is also not a loop.

Definition 2.4.9 (ag-bisimulation) *Given two models $\mathcal{M}_1 = (S_1, R_1^{\mathcal{A}}, V_1^{\mathcal{P}})$ and $\mathcal{M}_2 = (S_2, R_2^{\mathcal{A}}, V_2^{\mathcal{P}})$, a non-empty relation $Z \subseteq S_1 \times S_2$ between the models is an ag-bisimulation if it obeys the following properties:*

1. *if $s_1 Z s_2$, then for all $P \in \mathcal{P}$ it holds that $s_1 \in V_1^{\mathcal{P}}(P)$ if and only if $s_2 \in V_2^{\mathcal{P}}(P)$*
2. *if $s_1 Z s_2$, and for some state t_1, and some (non-point) action graph G it holds that $s_1(G)^{\triangleright}t_1$ relative to $\mathcal{M}_1$, then there is a $t_2 \in S_2$ such that (1) $s_2(G)^{\triangleright}t_2$ relative to $\mathcal{M}_2$ and (2) $t_1 Z t_2$*
3. *if $s_2 Z s_1$, and for some state t_2, and some (non-point) action graph G it holds that $s_2(G)^{\triangleright}t_2$ relative to $\mathcal{M}_2$, then there is a $t_1 \in S_1$ such that (1) $s_1(G)^{\triangleright}t_1$ relative to $\mathcal{M}_1$ and (2) $t_2 Z t_1$*

It is not hard to see that on tree models ag-bisimulation reduces to the standard notion of bisimulation. First of all, for tree models, sets of action graphs $\Gamma(\alpha)$ interpreting an action α reduce to sets of traces. Furthermore, the assertions '$s\Theta_a, t$ relative to $\mathcal{M}$' and '$(s,t) \in R^{\mathcal{A}}$' are equivalent, which implies that the above conditions include the conditions for standard bisimulation. Then, on tree models, the only extra condition imposed by ag-bisimulation is that traces $a_1, a_2, \ldots, a_n$ also obey the forth and back conditions. But on tree models this enforces nothing new: the standard notion of bisimulation already implies that for two bisimulating states s_1 and s_2 it holds that if there is a trace τ from s_1 to t_1, then there is a state t_2 such that (1) the trace τ starts at s_2 and arrives at t_2, and (2) that t_1 and t_2 bisimulate. Another way of saying this is that bisimulation equivalence implies trace equivalence, which is a standard result from concurrency theory [77].

Proposition 2.4.4 *A formula $\langle\alpha\rangle\varphi$ of MAL$(;,^*,\cup,\cap,^{\leftarrow})$ holds in a state s of a model $\mathcal{M}$ if and only if for some (finite) action graph G and state t it holds that $G \in \Gamma(\alpha)$ and $s(G)^{\triangleright}t$ and $\mathcal{M}, t \models \varphi$. We call G a 'witness' for α in $\mathcal{M}$.*

Proof

($\Rightarrow$) By induction over the structure of actions. First we consider the atomic action case: $\mathcal{M}, s \models \langle a\rangle\varphi$. The semantic condition ensures that there is a t such that $(s,t) \in R^{\mathcal{A}}$ and $\mathcal{M}, t \models \varphi$. Then it is clear that for the elementary graph Θ_a for which by definition $\Theta_a \in \Gamma(a)$, there is a trivial homomorphism that ensures that $s(\Theta_a)^{\triangleright}t$.

The case $\mathcal{M}, s \models \langle\alpha;\beta\rangle\varphi$ follows from the semantic equivalence with $\langle\alpha\rangle\langle\beta\rangle\varphi$ and the induction hypotheses: from the graph G_1 such that $G_1 \in \Gamma(\alpha)$ and the graph G_2 such that $G_2 \in \Gamma(\alpha)$ we construct the graph $G_1 \cdot G_2$, for which it holds by definition 2.4.7 that $G_1 \cdot G_2 \in \Gamma(\alpha;\beta)$. Clearly this construction preserves the homomorphism condition.

For the case $\mathcal{M}, s \models \langle\alpha\cap\beta\rangle\varphi$ we take a closer look on its semantic condition. The semantics tells us that there is a state t such that $(s,t) \in R(\alpha)$ and $(s,t) \in R(\beta)$ and $\mathcal{M}, t \models \varphi$. But then also the formulas $\langle\alpha\rangle\varphi$ and $\langle\beta\rangle\varphi$ hold in s, and what is more, both these formulas have state t as a witness for φ. Then again, we use the induction hypotheses and construe a parallel composition of the action graphs G_1 and G_2 for α and β to obtain the witness $G_1 \parallel G_2$ for $\mathcal{M}, t \models \langle\alpha\cap\beta\rangle\varphi$. It is easy to check that the required conditions are preserved.

The case $\mathcal{M}, s \models \langle\alpha \cup \beta\rangle\varphi$ follows from the induction hypotheses in the same way, but in this case we may simply choose one of the two graphs interpreting α and β respectively.

The case $\mathcal{M}, s \models \langle \alpha^* \rangle \varphi$ is based on the following property, that follows directly from semantic conditions: $\mathcal{M}, s \models \langle \alpha^* \rangle \varphi$ if and only if (1) for some $n \in \{1, 2, \dots\}$ it holds that $\mathcal{M}, s \models \langle \alpha^n \rangle \varphi$ (where $\alpha^1 \equiv_{def} \alpha$, $\alpha^2 \equiv_{def} \alpha; \alpha$, etc.), or (2) $\mathcal{M}, s \models \varphi$. Now, case (1) is a special instance of the sequential case, and for case (2) it is easy to see that the point graph Υ is a suitable candidate.

The case $\mathcal{M}, s \models \langle \alpha^{\leftarrow} \rangle \varphi$ follows from the induction hypotheses: from the graph G such that $G \in \Gamma(\alpha)$, we construe the graph $G^{\circlearrowleft}$.

($\Leftarrow$) The 'if' direction follows from the relational interpretation of the action connectives and the graph interpretation of actions in definition 2.4.7 by straightforward induction over the structure of action graphs.

■

Theorem 2.4.5 *State validity of formulas of MAL*$(;, ^*, \cup, \cap, ^{\leftarrow})$ *is preserved by ag-bisimulations.*

Proof

By induction on the structure of formulas φ. Let s_1 and s_2 be the states such that $(s_1, s_2) \in Z$ and Z is an ag-bisimulation. For proposition letters preservation is immediate from condition 1. For the logic connectives $\wedge$ and $\neg$, the property follows from the induction hypotheses. This leaves us with the case $\langle \alpha \rangle \varphi$ for which we consider condition 2. From the 'only if' direction of proposition 2.4.4 it follows that $\mathcal{M}_1, s_1 \models \langle \alpha \rangle \varphi$ implies that there is an action graph G and a state t_1 such that $G \in \Gamma(\alpha)$ and $s_1(G)^{\triangleright} t_1$ and $\mathcal{M}_1, t_1 \models \varphi$. Now condition 2 says that there is a state t_2 such that $s_2(G)^{\triangleright} t_2$ and $\mathcal{M}_2, t_2 \models \varphi$. But then from the 'if' direction of proposition 2.4.4 it follows that $\mathcal{M}_2, s_2 \models \langle \alpha \rangle \varphi$. The converse direction follows from condition 3.

■

Proposition 2.4.6 *The class* $\mathcal{C}_{R(\gamma)=R(\alpha \cap \beta))}$ *of models for which* $R(\gamma) = R(\alpha) \cap R(\beta)$ *is not definable in MAL*$(;, ^*, \cup, \cap, ^{\leftarrow})$.

Proof

Assume (1) that there is a MAL$(;, ^*, \cup, \cap, ^{\leftarrow})$-formula ψ that defines the class of models $\mathcal{C}_{R(\gamma)=R(\alpha \cap \beta))}$. Take the model $\mathcal{M}$ from the class $\mathcal{C}_{R(\gamma)=R(\alpha \cap \beta))}$, with states s and t such that $(s, t) \in R(\alpha)$, $(s, t) \in R(\beta)$, and $(s, t) \in R(\gamma)$. From the assumption that ψ defines $\mathcal{C}_{R(\gamma)=R(\alpha \cap \beta))}$ it follows that $\mathcal{M} \models \psi$. We show how to construct a model $\mathcal{M}'$ from $\mathcal{M}$ such that $\mathcal{M}' \models \psi$, while $\mathcal{M}' \notin \mathcal{C}_{R(\gamma)=R(\alpha \cap \beta))}$. We showed that in MAL$(;, ^*, \cup, \cap, ^{\leftarrow})$, state validity

of formulas is preserved by ag-bisimulation. We construct from $\mathcal{M}$ the ag-bisimulating model $\mathcal{M}'$ by making an identical copy u of state t, and adding $(s,u) \in R(\alpha)$, $(s,u) \in R(\beta)$. Models $\mathcal{M}$ and $\mathcal{M}'$ ag-bisimulate, because for the resulting modal $\mathcal{M}'$ only an identical copy of one of the action graphs relating s and t in the original model (subgraphs count as separate graphs), is added. From preservation of formula validity under ag-bisimulation, it follows that $\mathcal{M}', s \models \psi$, and since states t and u in $\mathcal{M}'$ have isomorphic copies in $\mathcal{M}$, we have $\mathcal{M}' \models \psi$. This directly contradicts assumption 1.

■

This proves that the class $\mathcal{C}_{R(\gamma)=R(\alpha\cap\beta))}$ is not defined by for instance $\langle\gamma\rangle\varphi \leftrightarrow \langle\alpha \cap \beta\rangle\varphi$, which was sufficient to define intersection at the levels of frames. And indeed this is immediately clear: the model with states s,t,u and $(s,t) \in R(\gamma)$, $(s,u) \in R(\alpha)$, $(s,u) \in R(\beta)$, and t and u agree on all valuations, satisfies it, but is not in $\mathcal{C}_{R(\gamma)=R(\alpha\cap\beta))}$. The conclusion is that in MAL$(;,^*,\cup,\cap,^\smile)$, we can define confluency, but definability of the class $\mathcal{C}_{R(\gamma)=R(\alpha\cap\beta))}$ is prevented by the inability to express that two (compound) actions define the same relation over states. Therefore we cannot enforce that the relation $\alpha \cap \beta$ coincides with the relation γ, which would yield a definition of the class $\mathcal{C}_{R(\gamma)=R(\alpha\cap\beta))}$. The following proposition makes this conclusion explicit.

Proposition 2.4.7 *The class of models $\mathcal{C}_{R(\alpha)=R(\beta)}$, for which $R(\beta) = R(\alpha)$ is not definable in MAL$(;,^*,\cup,\cap,^\smile)$.*

Proof Analogous to the proof of proposition 2.4.5, by substituting β for $\alpha\cap\beta$.
■

All of the above developed theory for the logic MAL$(;,^*,\cup,\cap,^\smile)$ applies to $\cap$-logics in general. For instance, for $\cap$-logics without converse, we may specialize the notion of action graph by dropping closure under reverse orientation. For $\cap$-logics without sequence and iteration, we may specialize the notion of action graph by dropping closure under sequential composition, etc.

Since $\cap$-logics identify the set of models for which $R(\gamma) = R(\alpha) \cap R(\beta)$ with the set of models for which $R(\gamma) \subseteq R(\alpha) \cap R(\beta)$, as follows from the proof of proposition 2.4.6, we may relax the interpretation of the $\cap$-operator from $R(\alpha \cap \beta) = R(\alpha) \cap R(\beta)$ to $R(\alpha \cap \beta) \subseteq R(\alpha) \cap R(\beta)$, without destroying validity of formulas. So it is possible to relax the interpretation of the $\cap$-symbol and arrive at the same logic (defined as the set of all its validities, definition 2.1.3). In other words: the $\cap$-logics do not require that the interpretation of the $\cap$-symbol coincides with intersection: also a slightly relaxed interpretation

qualifies for the task. This points to a semantic inadequacy of the logics. We introduced intersection into the semantics for a good reason: because it intuitively captures the notion of true concurrency. But now it appears that we get the same logic by relaxing intersection in the aforementioned way. This means that not all aspects of reasoning under *real* intersection are captured by the logic: the logic is simply not strong enough to distinguish between real intersection and the slightly relaxed interpretation. In particular, the logic cannot encompass reasoning of the sort where from certain premises we have to logically draw the conclusion that actions are concurrent, that is, intersect.

In section 2.5 we increase expressiveness by introduction of an action complement operator. There we motivate this extension with the intended application of modal action logics to all our domains of interest: concurrency, effects, time, and norms. But the semantic inadequacy observed in this section gives independent motivation to look at stronger languages. And indeed we prove that the modal action logics with complement *do* define intersection at the level of models.

2.4.4 Related approaches to concurrency in modal action logic

Other approaches to the incorporation of concurrency into PDL were undertaken by Peleg [148, 147] and Lodaya et al. [117]. However, neither of these approaches relates concurrency to intersection.

Peleg defines generalized tree models in order to satisfy $\langle \alpha \& \beta \rangle \varphi \leftrightarrow \langle \alpha \rangle \varphi \wedge \langle \beta \rangle \varphi$. This means that concurrently composed actions inherit effect properties from their parts, while at the same time effect properties of composed actions can always be completely allocated to some constituent action. We already argued that this second aspect is not very intuitive for a language that models the reasoning of a specifier, but here we elaborate on this issue some more. If a specifier wants to prove that a specification obeys the property that α and β can be performed concurrently, he wants to prove that $\langle \alpha \& \beta \rangle \top$. However, in CPDL this is equivalent to $\langle \alpha \rangle \top \wedge \langle \beta \rangle \top$. So in CPDL a parallel composition can be performed if and only if both its parts can be performed separately. This means that in CPDL the specifier is not able to reason about the question whether the actions can be performed together: it is simply impossible that they cannot be performed whenever the constituent actions can be performed. So $\langle \alpha \& \beta \rangle \varphi \leftarrow \langle \alpha \rangle \varphi \wedge \langle \beta \rangle \varphi$ is a property that forms an obstacle for the kind of reasoning a specifier uses to derive properties of his design. In a specification language we typically specify the parts of a parallel composition separately and then want to prove or investigate whether they can be performed concurrently

in certain contexts. A non-dynamic logic approach that is worth mentioning in this context, is the one by Lifschitz. In the language C^+ [69], effects of actions add up if they are performed concurrently, *unless* the specification explicitly states that a subsuming concurrent action overrules the effect. In the extension by Thielscher [21], in addition to this, conflicting effects are reinterpreted as non-determinism. Also we mention the approach by Meyer and Doherty [136], that allows free interaction between concurrent effects.

The second approach to the incorporation of concurrency in dynamic logic focuses on a reduction to interleaving [117]. We argued in chapter 1 why we do not endorse a reduction of concurrency to interleaving in the context of reasoning about reactive system properties.

2.5 Action complement

Our study of the notion of action complement is motivated by the need to be able to refer to (the union of) actions other than a given action α. The concept of 'alternative action' arises naturally in deontic action reasoning, temporal action reasoning and reasoning about effects of actions, and reasoning about true concurrency. So the action complement plays an important role in all the central themes of this thesis, which explains the focus on this action operation in this chapter on modal action reasoning. In the two approaches to action complement we propose in this section, the intersection operation is syntactically definable. The notions of action complement, or 'action negation', we study are different from the dynamic negation, as defined by Van Benthem [17]. Dynamic negation (defined as the test concerning non-possibility to perform an action: $\sim_d \alpha \equiv_{def} ([\alpha]\bot)?$) refers to non-activity, and not to alternative action.

To explain the difference between the two approaches to action complement in this section, we quote words from Krister Segerberg [167] that address the central issue:

> 'By contrast with intersection, the question concerning complement (negative action) is intricate and involves much extra-theoretical consideration: do we humans really think in terms of complements? Does the analysis of human languages suggest that we do? Is it not the case that the choice between two actions a and $U \setminus a$ is often a choice between a and some action b that is a proper sub-set of $U \setminus a$? Before these questions have been answered, this author feels a certain unease about the unrestricted acceptance of

closure under complement.'

Segerberg uses the symbol U here to refer to the universal relation $S \times S$. He makes several points. One is that he is not convinced of the naturalness of a notion of action complement. We first show in section 2.5.1 that the action complement does appear as a natural operation in several reasoning tasks. The second important point made is that it is in a way counter-intuitive to define action complement with respect to the universal relation. In our study of complement with respect to the universal relation (2.5.2) in modal action logic, we make the uneasiness felt by Segerberg explicit by showing that it induces some properties that cannot be considered intuitive action laws. After that (2.5.3) we show how to define weaker notions of complement, that should take Segerberg's feelings of uneasiness away, and that result in logics that are more fitted to a reasoning domain of actions.

Somewhat out of line with the structure of the whole chapter, in this section about complement the order in which we consider the different extensions is from strong to weak.

2.5.1 Reasoning domains involving action complement

One of the contexts where the action complement under the interpretation of 'alternative action' naturally arises, is concerned with reasoning about effects of actions. In this reasoning context it is natural to consider the property that a certain effect is brought about exclusively by an action. Another way of saying this is that other actions cannot have that effect. Here we see how the concept of 'the other actions' or 'alternative action' can be used to reason about how effects and actions are related. If, for the moment, we denote action complement with the symbol '$-$', we can capture the property of 'exclusive effect' in a formula of the form: $\neg\varphi \rightarrow [-\alpha]\neg\varphi$. Intuitively it reads 'performing an action other than (one that involves) α from a state for which $\neg\varphi$ holds, never results in a state where φ holds'. So it expresses that changing φ from false to true can only be accomplished by performing an action (that involves) α, provided that such an action is actually possible. De Giacomo [73] observes that frame formulas of this type reflect Reiter's 'solution' [155, 20] to the frame problem. The formulas are called 'frame formulas' because they preserve the property $\neg\varphi$ over the action $-\alpha$. But note that they do not preserve properties over the action α. If for instance it is specified that $\neg P \rightarrow [-\alpha]\neg P$, then action α may or may not change P from false to true. Formulas of the form $\neg\varphi \rightarrow [-\alpha]\neg\varphi$ say nothing about what is preserved by α. To express that α preserves $\neg P$, we first have to find out which of the actions in the description

domain possibly change the value of P from false to true. Then, if β is the union of these actions, the formula $\neg P \rightarrow [-\beta]\neg P$ implies that all ways to perform α that are not also a way to perform β preserve $\neg P$. Reiter's solution to the frame problem [155] is simply to give a formula $\neg P \rightarrow [-\beta]\neg P$ for any proposition P (fluent) in the action description.

In the formalization of the bowl of soup example we can use the action negation to express that the effect UpX (X for $Right$ and $Left$) is exclusively brought about by the action *x-lift*:

$$
\begin{array}{ll}
AD = \{ & \\
[\textit{left-lift}]UpLeft, & \neg UpLeft \rightarrow [-\textit{left-lift}]\neg UpLeft, \\
[\textit{right-lift}]UpRight, & \neg UpRight \rightarrow [-\textit{right-lift}]\neg UpRight, \\
UpLeft \wedge \neg UpRight \rightarrow Spilled, & \neg UpLeft \wedge UpRight \rightarrow Spilled, \\
UpLeft \wedge UpRight \rightarrow \neg Spilled, & \neg UpLeft \wedge \neg UpRight \rightarrow \neg Spilled\}
\end{array}
$$

It is not difficult to verify the following entailment relations:

$$
\begin{array}{l}
AD \models_G \neg UpLeft \wedge \neg UpRight \rightarrow [\textit{left-lift} \cap -\textit{right-lift}]Spilled \\
AD \models_G \neg UpLeft \wedge \neg UpRight \rightarrow [-\textit{left-lift} \cap \textit{right-lift}]Spilled
\end{array}
$$

A second area where the action complement arises naturally is that of *deontic* action reasoning. Traditionally, in deontic logics that were not so much concerned with normative statements about action as with normative statements about conditions, in general the relation between *obligation* ($O(\varphi)$) and *permission* ($P(\varphi)$) is described as $O(\varphi) \leftrightarrow \neg P(\neg\varphi)$. In the work on (dynamic) deontic *action* logic, initiated by Meyer [135], this property reappears as $O(\alpha) \leftrightarrow \neg P(-\alpha)$. This identification only makes sense under a suitable interpretation of the action connective '$-$' as an action complement that refers to actions alternative to that referred to by α. Under such an interpretation the identification is read as 'obligation to perform an action a bi-implies absence of permission to perform any action not involving a (alternative action)'. We discuss this type of logics extensively in chapter 5.

The third area in which action complement, in the interpretation of alternative action, arises as a natural mode of expression is that of *temporal* reasoning over action. For instance, the (liveness) property that over all possible futures eventually an action a is inevitable, may be logically identified with the property that it is not possible to perform actions alternative to a forever. We define logics that can make this type of logical identifications in chapter 3.

Finally, we mention that reasoning about action complements is directly related to reasoning about concurrency via the identification $-(\alpha \cap \beta) \equiv -\alpha \cup -\beta$.

2.5.2 Complement with respect to the universal relation

In this section we show that in the extension of PDL with the complement as copied from relation algebra, has properties that although useful and intuitive for relations in general, do not apply to *actions*. In the next section we propose weakenings of this complement that better suit the intended interpretation as 'alternative action'

In the relation algebraic approach the natural way to define the complement is with respect to the universal relation. It is straightforward to import this notion of complement in the modal action logics we study. We denote this complement with the symbol '$\sim$'.

Definition 2.5.1 (semantics of $\sim$-logics) *$\sim$-logics are defined as modal action logics whose set AC of action combinators includes the unary action operator '$\sim$', which we call the 'universal complement'. The semantics of $\sim$-logics follows from the modal semantics of definition 2.1.2, and an extension of the relational interpretation function R of definition 2.4.1 such that it includes the following equality for $\sim$:*

$$R(\sim\alpha) \quad = \quad S \times S \setminus R(\alpha)$$

The action semantics of this complement is exemplified by the following informal meaning of formulas of the form $[\sim\alpha]\varphi$: for all states that cannot result from performing α, it holds that φ. Note that this includes the states that are not reachable at all.

Proposition 2.5.1 *The complement operator $\langle\sim\alpha\rangle\varphi$ is not syntactically definable in the basic modal language, nor in c-PDL or c-IPDL.*

Proof

We may rely on the notion of ag-bisimulation to prove this property. But it is sufficient here to consider the coarser semantic equivalence that says that state validity is preserved under contraction to generated sub-models (for the definition of generated sub-models see the proof of proposition 2.3.1 and [19]). All logics considered so far obey this property, provided that we include the converse direction of actions in case the converse operation is in the action

language. But logics containing the universal complement do not obey it. For example, state validity of the formula $\langle \sim a \rangle \top$ is not always preserved by contraction to a generated sub-model; it is not valid in a one-state model with an action a looping in it, while this model is a generated sub-model of many models that do satisfy the formula.

■

With one exception [73], in the literature known to us concerning the complement in modal action logics [68, 78, 143, 85] and related algebraic formalisms (dynamic algebras, Peirce algebra's, boolean modules), this is the version of the complement that is studied. This is not surprising given that the universal complement directly corresponds to standard negation in the first-order view on modal languages. It is straightforward to define a translation T from iteration-free $\sim$-logics involving a set of propositions $\{P, Q, \dots\}$, and a set of atomic actions $\{a, b, \dots\}$ to a first-order logic with a set of unary predicates $\{P, Q, \dots\}$, a set of binary predicates $\{R^a, R^b, \dots\}$, and a set of variables $\{x, y, \dots\}$:

$$
\begin{array}{lcllcl}
T_x(P) & \equiv & Px & T_{xy}(a) & \equiv & R^a xy \\
T_x(\bot) & \equiv & x \neq x & T_{xy}(\alpha \cup \beta) & \equiv & T_{xy}(\alpha) \vee T_{xy}(\beta) \\
T_x(\neg\varphi) & \equiv & \neg T_x(\varphi) & T_{xy}(\alpha^{\leftarrow}) & \equiv & T_{yx}(\alpha) \\
T_x(\varphi \wedge \psi) & \equiv & T_x(\varphi) \wedge T_x(\psi) & T_{xy}(\alpha;\beta) & \equiv & \exists z(T_{xz}(\alpha) \wedge T_{zy}(\beta)) \\
T_x(\langle\alpha\rangle\varphi) & \equiv & \exists y(T_{xy}(\alpha) \wedge T_y(\varphi)) & T_{xy}(\sim\alpha) & \equiv & \neg T_{xy}(\alpha)
\end{array}
$$

With induction we can prove that for any $\varphi \in \mathcal{L}_{MAL}(^{\leftarrow}, ;, \cup, \sim)$ it holds that $\mathcal{M}, s \models \varphi$ iff $\mathcal{M} \models_{FOL} T_x(\varphi)[s]$ (s is assigned to the free variable x). Iteration free $\sim$-logics are thus subsumed by first-order logic. This means that they inherit the compactness property from FOL: any non-satisfiable infinite set has a non-satisfiable finite subset. The relativized action complement we discuss in section 2.5.3 does not have this property for all modal action logics considered.

We now give a set of action connectives that can be introduced as syntactic abbreviations in terms of the complement ($\sim$) and the choice ($\cup$).

Definition 2.5.2 *The U, the fail, the intersection, the subsumption action, and the action* equivalence *are defined through the following syntactic extensions on $\cup$ and $\sim$ (We add parentheses whenever ambiguity in the reading of relational formulas may arise. But we do assume that unary operators bind stronger than binary ones.):*

$$\begin{array}{lll} \alpha \cap \beta & \equiv_{def} & \sim(\sim\alpha \cup \sim\beta) \\ \alpha \subseteq \beta & \equiv_{def} & \sim\alpha \cup \beta \\ \alpha \div \beta & \equiv_{def} & (\alpha \subseteq \beta) \cap (\beta \subseteq \alpha) \end{array} \qquad \begin{array}{lll} fail & \equiv_{def} & \alpha \cap \sim\alpha \\ U & \equiv_{def} & \alpha \cup \sim\alpha \end{array}$$

Due to these definabilities, action operations as $\cap$ and $fail$ do not have to be introduced explicitly in a modal action logic whenever complement and choice are already included (note that $fail$ can also be defined in terms of the test: $fail \equiv_{def} \bot?$). The above definition says that $\alpha \subseteq \beta$ should be read as the action whose reachability relation is the union of that of β and that of the complement of α. It may be confusing to read $\alpha \subseteq \beta$ as an action. We call an action $\alpha \subseteq \beta$ a *subsumption action*, it represents all actions for which it holds that if α is involved (as a concurrent component), than also β is involved. We call an action $\alpha \div \beta$ an *equivalence action*, since it denotes any action obeying the condition that α is involved if and only if β is involved (which includes the case that neither α or β is involved). We can turn a *subsumption action* into an *action implication* by demanding the following on the modal level.

$$\alpha \Rightarrow \beta \equiv_{def} [\sim(\alpha \subseteq \beta)]\bot$$

The right hand side says 'all actions for which it does not hold that if you do an α, you also do a β, lead to a falsum', which is to say that they cannot be there. This expresses the property that performing a transition α implies doing β at the same time (concurrently). So we can express things like 'running implies walking and walking implies moving':

$$run \Rightarrow walk \qquad\qquad walk \Rightarrow move$$

The U that is defined, includes any action relation. This motivates some authors to call it the 'any-operation'. For instance, this variant of the notion of 'any action' is studied as an explicit addition to PDL by Prendinger [153]. At the end of this section we will argue that the U is too strong to function as a notion of 'any action'. In [19] the same operator is called the *global modality*, and in [118] it is called the *universal modality*. Finally, Goranko, Gargov, Passy and Tinchev [146] call it the *universe program*. The action U, as defined above, reaches *every* state from any given state. This means that in each state we have power to say something about *all* other states of a model.

The universal complement enables the definition of the so-called window operator. Semantically the window operator $[[\alpha]]\varphi$ is defined as follows:

Definition 2.5.3

$$\mathcal{M}, s \models [|\alpha|]\varphi \quad \textit{iff} \quad \textit{for all } t, \textit{ if } \mathcal{M}, t \models \varphi \textit{ then } (s,t) \in R(\alpha)$$

The definition of the window-operator in terms of complement is: $[|\alpha|]\varphi \equiv_{def} [\sim\alpha]\neg\varphi$ [118]. This means that the window operator is just a shorthand for the formulas we already met when we motivated the introduction of a complement in modal action logic with its relevancy for the reasoning about action effects. We may now try to capture the intuition 'a is the only action that may make φ true' with:

$$\neg\varphi \rightarrow [|\alpha|]\varphi$$

In section 2.5.2, we argue that this formula is actually too strong to express the intuition.

The window operator was used by Van Benthem [15] to define an ought-to-be deontic logic (Define ought-to-be obligation as $O(\varphi) = [\ ]\varphi$, using the modal relation to interpret reachability of deontic ideal worlds. And define ought-to-be *strong* permission as $P(\varphi) = [|\ |]\varphi$, thereby obeying $P(\varphi \vee \psi) \leftrightarrow P(\varphi) \wedge P(\psi)$, which avoids the free choice permission anomaly[3]). And it was used by Humberstone [101] to define a logic of inaccessible worlds.

Some validities

With the introduction of the complement into a modal action logic encompassing the connectives of c-PDL, we arrive at a superset of the repertoire of connectives that is studied in (binary) relation algebra. The only difference is that relation algebra does not encompass iteration. All connectives of MAL($^*, ^\leftarrow, ;, \cup, \sim$) have exactly the same *interpretation* as in relation algebra. In relation algebra, axiomatization is accomplished by formulating laws that define how operations interact. A complete set of axioms for relation algebra is:

(0)	a complete set of boolean axioms		
(1)	$\alpha;(\beta;\gamma) \doteqdot (\alpha;\beta);\gamma$	(5)	$(\alpha^{\leftarrow})^{\leftarrow} \doteqdot \alpha$
(2)	$\alpha; \mathit{skip} \doteqdot \alpha \doteqdot \mathit{skip};\alpha$	(6)	$(\alpha;\beta)^{\leftarrow} \doteqdot \beta^{\leftarrow};\alpha^{\leftarrow}$
(3)	$(\alpha \cup \beta);\gamma \doteqdot \alpha;\gamma \cup \beta;\gamma$	(7)	$(\alpha \cup \beta)^{\leftarrow} \doteqdot \alpha^{\leftarrow} \cup \beta^{\leftarrow}$
(4)	$\alpha;(\beta \cup \gamma) \doteqdot \alpha;\beta \cup \alpha;\gamma$	(8)	$\sim\beta \doteqdot \sim\beta \cup (\alpha^{\leftarrow};\sim(\alpha;\beta))$

[3] Van Benthem calls it 'Ross' paradox', by mistake. Ross' paradox concerns a similar problem for obligations (see section 5.1).

These axioms reflect properties of action relations of modal action models. As opposed to modal properties, that take the perspective of states within models, these properties can be said to result from a 'global' external view on the modal relations in models. But the above global relation algebraic properties have local modal repercussions. The following proposition is immediate.

Proposition 2.5.2 *If $\alpha \div \beta$ algebraically, the following are schemes for general validities of the logic MAL$(^*, ^\smile, ;, \cup, \sim)$:*

$$\models \langle\alpha\rangle\varphi \leftrightarrow \langle\beta\rangle\varphi \qquad\qquad \models [\sim(\alpha \div \beta)]\varphi$$

Proof

The first scheme is immediate. For the second scheme we have to expand the definition for $\div$, according to definition 2.5.2, and apply a series of semantically motivated steps to show equivalence with the scheme $[fail]\varphi$, which is trivially valid for all instantiations of φ.

■

An interesting question is whether the converse also holds: that is, is validity of all the formulas that instantiate these schemes enough to guarantee equivalence of actions in the relation algebraic sense? This issue is important for the question whether we loose expressive power by going from the global perspective of relation algebra to the local perspective of modal logics. For instance: given two action terms α and β, can we prove that $\alpha \div \beta$ algebraically by establishing (some form of) validity of some modal action logic formula? It depends on what level we consider this question: the level of models or the level of frames. It is easy to prove that at the level of frames, both schemes enforce equivalence of action relations through frame validity. This should not come as a surprise, since as we argued before, as a means to talk about frame properties, through the notion of frame validity, modal (action) logics are very strong. But in a sense it is not 'fair' to claim that the first formula proves algebraic equivalence: it only proves this relative to a certain frame. To prove relational equivalence *as such*, it is necessary to show that equivalence can be enforced at the levels of models. In proposition 2.5.6 we prove that validity of all formulas that follow the second scheme, $[\sim(\alpha \div \beta)]\varphi$, indeed accomplishes this. The first scheme does not.

As an example of the type of validity schemes that proposition 2.5.2 is concerned with, we mention the interaction of the converse and the complement. In MAL$(^*, ^\smile, ;, \cup, \sim)$, the following is a general validity scheme: NegConv-R $\models [\sim(\alpha^\smile)]\varphi \leftrightarrow [(\sim\alpha)^\smile]\varphi$. This follows immediately from the

fact that $\sim(\alpha^{\smile}) \doteq (\sim\alpha)^{\smile}$ algebraically. Note that we can use the property $NegConv - R$, in combination with the rewrite rules for converse in the proof of theorem 2.3.2, to push down converse tot the atomic level. So also for the logic MAL$(^*,^{\smile},;,\cup,\sim)$ we do not loose expressive power by syntactic restriction of the converse operation to atomic actions only. Note also that a similar property does not hold for the other unary operators: complement and iteration.

But the above types of validity do not form the only repercussion of the algebraic laws to the modal language. Examples of modal properties that are induced by, in particular, the introduction of the universal complement are:

Proposition 2.5.3 *The following are validity schemes for the U in the logic* $MAL(^*,^{\smile},;,\cup,\sim)$.

$$\begin{array}{ll} \models \varphi \to [U]\langle U\rangle\varphi & \textit{(Symmetry-U)} \\ \models \varphi \to \langle U\rangle\varphi & \textit{(Reflexivity-U)} \\ \models \langle U\rangle\langle U\rangle\varphi \to \langle U\rangle\varphi & \textit{(Transitivity-U)} \\ \models \langle U\rangle\varphi \to \langle U\rangle\langle U\rangle\varphi & \textit{(Density-U)} \end{array}$$

Proof

Straightforward semantic verification (the universal relation is an equivalence relation).

■

These properties make the U-operator an S5-modality. We will argue that these properties disqualify U as an operator that reflects the notion of 'any action'. But there are many more modal validity schemes that are not a simple reformulation of the algebraic axioms. Before we explore these modal properties, we note that there are two important differences between modal logic properties and relation algebraic properties. First of all, relation algebraic properties take a *global* view on relations, while modal properties take a *local* view. For instance, a local modal repercussion of the algebraic properties 5-7 is formed by the validity schemes: $\varphi \to [\alpha]\langle\alpha^{\smile}\rangle\varphi$ and $\varphi \to [\alpha^{\smile}]\langle\alpha\rangle\varphi$. These modal properties characterize the converse in terms of the interaction with propositional state reasoning; there is no reference to the interactions of converse with other action combinators, such as in the axioms of relation algebra.

We now explore more local modal repercussions of the algebraic properties. The repercussions of properties 1-4 are none. More interesting are

boolean (semi-)equations such as $\alpha \subseteq \alpha \cup \beta$ and $\sim\sim\alpha \div \alpha$. We did not explicitly mention such axioms in the above listing of relation algebraic laws, since there are many different ways in which to axiomatize boolean algebra. And actually, the problem of finding *minimal* sets of boolean algebraic properties that completely axiomatize boolean algebra is very hard in general. But we are not interested in this question, and rely on the assumption that boolean properties are very familiar and need no further explanation or justification. A local modal repercussion of the boolean properties of actions is the following:

Proposition 2.5.4 *The following is a validity scheme of* $MAL(^*, ^\leftarrow, ;, \cup, \sim)$.

$$\models \langle\alpha\rangle\varphi \wedge [\sim\beta]\neg\varphi \rightarrow \langle\alpha \cap \beta\rangle\varphi \qquad \textit{(K-R)}$$

Using the window operator, we may write this alternatively as $\models \langle\alpha\rangle\varphi \wedge [|\beta|]\varphi \rightarrow \langle\alpha \cap \beta\rangle\varphi$.

Proof

Consider an arbitrary model $\mathcal{M}$ and an arbitrary state s. In case that $\mathcal{M}, s \not\models \langle\alpha\rangle\varphi$ or $\mathcal{M}, s \not\models [\sim \beta]\neg\varphi$ it holds trivially that $\mathcal{M}, s \models \langle\alpha\rangle\varphi \wedge \neg\varphi \rightarrow \langle\alpha \cap \beta\rangle\varphi$. So, we assume that (1) $\mathcal{M}, s \models \langle\alpha\rangle\varphi$ and (2) $\mathcal{M}, s \models [\sim\beta]\neg\varphi$. From (1) it follows that $\alpha \neq fail$ and that there is a state t such that $(s, t) \in R(\alpha)$ and (1') $\mathcal{M}, t \models \varphi$. Now either $(s, t) \in R(\beta)$ or $(s, t) \in R(\sim\beta)$ ($\sim\beta \cup \beta$ is the universal relation). But from $(s, t) \in R(\sim\beta)$ and property 2 we would have to conclude that $\mathcal{M}, t \models \neg\varphi$, which contradicts 1'. Then, from $(s, t) \in R(\beta)$ and $(s, t) \in R(\alpha)$ it follows that $(s, t) \in R(\alpha \cap \beta)$, which together with 1' gives $\mathcal{M}, s \models \langle\alpha \cap \beta\rangle\varphi$.

■

The property K-R can be seen as the action equivalent of the basic modal property K: $\langle\alpha\rangle\varphi \wedge [\alpha]\psi \rightarrow \langle\alpha\rangle(\varphi \wedge \psi)$. When we discuss axiomatization of $\sim$-logics, we show how it can be used in deductive boolean reasoning over actions.

Now finally, we turn to the interesting question of what can be the local modal repercussions of the above algebraic property 8. We already mentioned that it gives rise to the modal validity scheme:

$$\models [\sim\beta]\varphi \leftrightarrow [\sim\beta \cup \alpha^\leftarrow; \sim(\alpha; \beta)]\varphi \qquad \text{(NegSeqConv-R)}$$

which by established modal reasoning for the $\cup$ and propositional reasoning is equivalent to:

$$\models [\sim\beta]\varphi \rightarrow [\alpha^{\leftarrow};\sim(\alpha;\beta)]\varphi \qquad \text{(NegSeqConv'-R)}$$

But this property is basically still a straightforward non-local translation of the relation algebraic property 8. And actually it is a straightforward reformulation of the appearance of this property in the original axiomatization given by Tarski [174]:

$$(8') \quad \alpha^{\leftarrow};\sim(\alpha;\beta) \subseteq \sim\beta$$

The question arises whether there are any local repercussions related to this algebraic property at all. To find related local properties we should take a local semantic view. From a local view property 8' says something like 'if I first go into backwards (converse) α-direction and after that in forward direction complementary to $\alpha;\beta$, I arrive at a place that is contained in all places that I would possibly have reached if from the beginning I would have gone directly in forward direction complementary to β'. So, from a local standpoint, the property compares the complement of $\alpha;\beta$ (after first taking one step in backwards α-direction) and the complement of β. Now it appears that from a local modal viewpoint this comparison can actually also be made without first taking one step back. The following property shows this. Its proof involves more than just a straightforward semantic verification.

Proposition 2.5.5 *The following is a validity scheme of MAL*$(^*,^{\leftarrow},;,\cup,\sim)$.

$$\models \langle\alpha\rangle[\sim\beta]\varphi \rightarrow [\sim(\alpha;\beta)]\varphi \qquad \textit{(NegSeq-R)}$$

Proof

We prove $NegSeq - R$ from negative demonstration: assume there is a model $\mathcal{M}$ with a state s such that (1) $\mathcal{M}, s \models \langle\alpha\rangle[\sim\beta]\varphi$ and (2) $\mathcal{M}, s \models \neg[\sim(\alpha;\beta)]\varphi$. From (1) it follows that $\alpha \neq fail$ and that there is a state t such that $(s,t) \in R(\alpha)$ and (1') $\mathcal{M}, t \models [\sim\beta]\varphi$. Property 2 is equivalent with (2') $\mathcal{M}, s \models \langle\sim(\alpha;\beta)\rangle\neg\varphi$. Now there are two cases: $\beta = fail$ or $\beta \neq fail$.

If $\beta = fail$ property 2' reduces to $\mathcal{M}, s \models \langle U\rangle\neg\varphi$. This is in direct contradiction with semantic property 1', that for $\beta = fail$ reduces to $\mathcal{M}, t \models [U]\varphi$. Since the U reaches all states, the contradiction strikes notwithstanding the circumstance that this second validity holds for another state.

If $\beta \neq fail$, there is a state u such that $(s,u) \in R(\sim(\alpha;\beta))$ and (3) $\mathcal{M}, u \models \neg\varphi$. Now either $(t,u) \in R(\beta)$ or $(t,u) \in R(\sim\beta)$ (again, $\sim\beta \cup \beta$ reaches all states). But from $(t,u) \in R(\sim\beta)$ and property 1' we would have to conclude that $\mathcal{M}, u \models \varphi$, which contradicts 3. And from $(t,u) \in R(\beta)$ together with $(s,t) \in R(\alpha)$ we get $(s,u) \in R(\alpha;\beta)$, which directly contradicts $(s,u) \in R(\sim(\alpha;\beta))$.

■

Note that this comparison between the two complements *without* first taking one step back cannot be made from the global orientation of relation algebra. This explains why we refer to property $NegSeq-R$ as a '*local* modal repercussion'. Also, it expresses the same intuition as property 8 using less action combinators (no converse). This seems as close as we can get to a local modal variant or consequence of algebraic property 8. We will argue that the property $NegSeq-R$ is not always intuitive for reasoning about action. In section 2.5.3 we develop several logics that do not obey it, by considering a slightly other interpretation of the complement.

Definability of classes of models and frames

The universal complement strongly enhances expressiveness. In [78] the frame expressiveness of the extension of modal logic with a universal relation complement is thoroughly investigated. One of the things mentioned there is that this complement enables expression of irreflexivity of frame relations: IRREFL: $[\sim a]\varphi \rightarrow \varphi$ expresses that the frame relation $R(a)$ is irreflexive. It accomplishes this by demanding that the complementary relation $R(\sim a)$ is reflexive. Other examples of expressible properties of frames are strict asymmetry SASYM: $\varphi \rightarrow [a]\langle\sim a\rangle\varphi$ and left linearity LeftL: $(\varphi \rightarrow \langle\sim a\rangle\psi) \vee (\langle a\rangle\psi \rightarrow (\varphi \vee \langle a\rangle\varphi))$. Together, the frame validities IRREFL and LeftL define a frame relation to be a tree. Note that we cannot express that these properties hold for all possible frame relations; we can only express that they hold for individual frame relations.

In section 2.4.3 we saw that modal action logics with intersection but without complement are strong enough to define intersection at the level of frames. And because the intersection connective is *syntactically* definable in $\sim$-logics, it should raise no surprise that also $\sim$-logics are strong enough to define intersection at the level of frames. That intersection is definable at the level of frames in logics with a universal complement is shown by several authors (e.g. [68], and [19] p.425). However, it seems to have gone unnoticed that in logics of this type (in our terminology '$\sim$-logics') we can define intersection in

a stronger sense, namely, at the level of models. First we prove that we can define relation *equivalence* at the level of models, which answers the question that arose after proposition 2.5.2.

Proposition 2.5.6 *In $\sim$-logics, the class of models $\mathcal{C}_{R(\alpha)=R(\beta)}$, for which $R(\beta) = R(\alpha)$, is defined by $[\sim(\alpha \div \beta)]\varphi$.*

Proof

($\Rightarrow$) Immediate from proposition 2.5.2.

($\Leftarrow$) First we rewrite the formula $[\sim(\alpha \div \beta)]\varphi$, by expanding the syntactic rules in definition 2.5.2 and by using semantic equivalences.

	$[\sim(\alpha \div \beta)]\varphi$
expand action abbreviations	$[\sim((\alpha \subseteq \beta) \cap (\beta \subseteq \alpha))]\varphi$
expand action abbreviations	$[\sim(\alpha \subseteq \beta) \cup \sim(\beta \subseteq \alpha)]\varphi$
boolean action reasoning	$[(\alpha \cap \sim\beta) \cup (\sim\alpha \cap \beta)]\varphi$
modal reasoning	$[\alpha \cap \sim\beta]\varphi \wedge [\sim\alpha \cap \beta]\varphi$
expand modal abbreviation	$\neg\langle\alpha \cap \sim\beta\rangle\neg\varphi \wedge \neg\langle\sim\alpha \cap \beta\rangle\neg\varphi$
subst. for props	$\neg\langle\alpha \cap \sim\beta\rangle\varphi \wedge \neg\langle\sim\alpha \cap \beta\rangle\varphi$

Now suppose $\mathcal{M} \notin \mathcal{C}_{R(\alpha)=R(\beta)}$. We have to show that $\neg\langle\alpha \cap \sim \beta\rangle\varphi \wedge \neg\langle\sim\alpha \cap \beta\rangle\varphi$ is not valid on $\mathcal{M}$. There are two cases. (case 1) There is a $(s,t) \in R(\alpha)$ while $(s,t) \notin R(\beta)$. This invalidates $\neg\langle\alpha \cap \sim\beta\rangle\varphi$, by instantiation of φ by a formula that holds in t, for which we can always use $\top$. (case 2) There is a $(s,t) \in R(\beta)$ while $(s,t) \notin R(\alpha)$. This invalidates $\neg\langle\sim\alpha \cap \beta\rangle\varphi$ by the same reasoning.

■

As a corollary we get that all action operations we considered before are definable at the level of models. As an example we mention the definitions for complement ($\mathcal{C}_{R(\beta)=R(\sim\alpha)}$, for which $R(\beta) = U \setminus R(\alpha)$) and intersection ($\mathcal{C}_{R(\gamma)=R(\alpha\cap\beta))}$, for which $R(\gamma) = R(\alpha) \cap R(\beta)$), which are defined by respectively $[\sim(\beta \div \sim\alpha)]\varphi$ and $[\sim(\gamma \div \alpha \cap \beta))]\varphi$. We argued in section 2.4.3 that the definability at the level of models of in particular intersection is a desirable property for our logics, since we view intersection as central to the formalization of true-concurrency in modal action logics, and want to incorporate all relevant aspects of reasoning with intersection. This point is in favor of the use of $\sim$-logics. But in the next sections we see some negative properties of logics of this type.

Complexity of $\sim$-logics

From a complexity perspective, $\sim$-logics are not attractive. The logics MAL($\sim$) and MAL($\cup, \sim$) (boolean modal logic) have EXPTIME-complete complexity [118, 119]. And already the extension with the sequence operation on actions, MAL($\cup, ;, \sim$), is undecidable. From the property that $\alpha \doteqdot \beta$ holds algebraically if and only if $\models [\sim(\alpha \doteqdot \beta)]\bot$ (proposition 2.5.6), it follows that $\sim$-logics are at least as complex as the algebras over their action combinators. The algebra over the combinators $\cup$, ; and $\sim$ is known to be undecidable (it is possible to encode the undecidable 'word problem' [181] in it).

Axiomatization of $\sim$-logics

De Rijke [156] studies dynamic modal logic with the universal complement (in our terminology MAL($^{\smile}, ;, \cup, \sim, \varphi?$), and gives a complete axiomatization. We do not consider this work here. But we do consider the axiomatization by Gargov and Passy [68] that concerns the logic MAL($\cup, \sim$), which they call 'boolean modal logic'. Gargov and Passy define a complete axiomatization, and show that boolean modal logic has the finite (small) model property. The logic can be seen as an extension with boolean modalities of the standard multi-modal variant of the weakest modal logic K (Hennessy-Milner logic). The relevancy of boolean reasoning over action is recognized by several authors [32, 70, 131, 54, 55]. Below we give a finite axiomatization of boolean modal logic. Gargov and Passy's deductive system relies on a recursively enumerable set of axioms.

Theorem 2.5.7 *The following axioms and rules form a sound and complete Hilbert-style deductive system for MAL($\cup, \sim$) (we freely use syntactic extensions).*

Bool1	$\varphi \to (\psi \to \varphi)$
Bool2	$(\varphi \to (\psi \to \theta)) \to ((\varphi \to \psi) \to (\varphi \to \theta))$
Bool3	$(\neg\varphi \to \psi) \to ((\neg\varphi \to \neg\psi) \to \varphi)$
K	$[\alpha](\varphi \to \psi) \to ([\alpha]\varphi \to [\alpha]\psi)$
Bool1-R	$[\sim(\alpha \subseteq (\beta \subseteq \alpha))]\varphi$
Bool2-R	$[\sim((\alpha \subseteq (\beta \subseteq \gamma)) \subseteq ((\alpha \subseteq \beta) \subseteq (\alpha \subseteq \gamma)))]\varphi$
Bool3-R	$[\sim((\sim\alpha \subseteq \beta) \subseteq ((\sim\alpha \subseteq \sim\beta) \subseteq \alpha)))]\varphi$

K'-R	$[\sim(\alpha \subseteq \beta)]\varphi \rightarrow ([\beta]\varphi \rightarrow [\alpha]\varphi)$
Disj-R	$[\alpha \cup \beta]\varphi \rightarrow [\alpha]\varphi$
Symm-U	$\varphi \rightarrow [U]\langle U\rangle\varphi$
Refl-U	$\varphi \rightarrow \langle U\rangle\varphi$
Trans-U	$\langle U\rangle\langle U\rangle\varphi \rightarrow \langle U\rangle\varphi$

Modus ponens: $\frac{\varphi, \varphi\rightarrow\psi}{\psi}$ *Modal generalization:* $\frac{\varphi}{[\alpha]\varphi}$

Proof

Soundness of $K' - R$ is proven by showing semantic equivalence with the property $K - R$ that was already proven a validity (proposition 2.5.4).

K'-R	$[\sim(\alpha \subseteq \beta)]\varphi \rightarrow ([\beta]\varphi \rightarrow [\alpha]\varphi)$
bool. reasoning over props.	$[\sim(\alpha \subseteq \beta)]\varphi \rightarrow ([\alpha]\varphi \vee \neg[\beta]\varphi)$
bool. reasoning over actions	$[\alpha \cap \sim\beta]\varphi \rightarrow ([\alpha]\varphi \vee \neg[\beta]\varphi)$
subst. for actions and for props.	$[\alpha \cap \sim\sim\beta]\neg\varphi \rightarrow ([\alpha]\neg\varphi \vee \neg[\sim\beta]\neg\varphi)$
bool. reasoning over actions	$[\alpha \cap \beta]\neg\varphi \rightarrow ([\alpha]\neg\varphi \vee \neg[\sim\beta]\neg\varphi)$
duals + bool. reasoning over props.	$\neg\langle\alpha \cap \beta\rangle\varphi \rightarrow \neg(\langle\alpha\rangle\varphi \wedge [\sim\beta]\neg\varphi)$
contraposition	$\langle\alpha\rangle\varphi \wedge [\sim\beta]\neg\varphi \rightarrow \langle\alpha \cap \beta\rangle\varphi$

Soundness of, for instance, $Bool1 - R$ follows almost immediately if the definition of $\subseteq$ is written out: $[\sim(\sim\alpha \cup (\sim\beta \cup \alpha))]\varphi$. This is semantically equivalent with $[\sim(\sim\alpha \cup \alpha))]\varphi$ and thus with $[fail]\varphi$, which is directly seen to be a validity. The point of the axioms $BoolX - R$ is that they all have the form $[\sim(BoolX)]\varphi$, where each $BoolX$ is a reformulation of an axiom for propositional boolean reasoning in relational terms. Clearly, validity follows for all formulas $[\sim(BoolX))]\varphi$ where $BoolX$ is a boolean validity reformulated in relational terms.

Completeness is shown by relating the above axiomatization to the (first) complete axiomatization defined by Gargov and Passy [68]. The difference between their axiomatization and ours is that they do not include the boolean axioms and $K' - R$, but instead propose $[fail]\varphi$, together with an infinite, but recursively enumerable set of axioms $\{[\beta]\varphi \rightarrow [\alpha]\varphi \mid \alpha \subseteq \beta$ is derivable in B$\}$, where B is the boolean algebra based on the atomic actions and the boolean

connectives. This set is enumerable because the relation $\subseteq$ for boolean algebras is decidable. Our axiomatization is equivalent with Gargov and Passy's because our axiomatization deduces all axioms in this recursively enumerable set, including $[fail]\varphi$. This follows immediately from the observation that the above axioms $BoolX - R$ together with the axiom $K' - R$ mimic a complete deductive system for boolean reasoning over relations: the axioms $BoolX - R$ are relational versions of the standard boolean axioms $BoolX$, and the axiom $K' - R$ is the relational version of the standard K-axiom for reasoning about modal states. Together with the modus ponens rule the axioms $BoolX - R$ and $K' - R$ form a deductive system that derives all formulas in the recursively enumerable set (an example of such a deduction is given below). This demonstrates that Boolean reasoning over actions is complete. If we would have used an equational axiomatization of the boolean reasoning over actions, we would need extra rules, such as the substitution rule in Gargov en Passy's second axiomatization.

■

We now list some properties that are available as theorems, deducible in the above system. We roughly sketch by which steps these deductions are made.

Proposition 2.5.8 *The following are theorems of MAL*$(\cup, \sim)$*:*

CONJ-R	$\langle \alpha \cap \beta \rangle \varphi \rightarrow \langle \alpha \rangle \varphi \wedge \langle \beta \rangle \varphi$
DISJ-R	$\langle \alpha \cup \beta \rangle \varphi \leftrightarrow \langle \alpha \rangle \varphi \vee \langle \beta \rangle \varphi$

Proof

Straightforward verification shows that these are validity schemes. Completeness then says that the properties are also derivable in the deductive system. To get an impression of such deductions we will roughly show how to deduce $\langle \alpha \cap \beta \rangle \varphi \rightarrow \langle \alpha \rangle \varphi$ from K'-R and BoolX-R axioms.

boolean axioms	$[\sim(\alpha \cap \beta \subseteq \alpha)]\varphi$
K'-R	$[\sim(\alpha \subseteq \beta)]\varphi \rightarrow ([\beta]\varphi \rightarrow [\alpha]\varphi)$
instant. K'-R	$[\sim(\alpha \cap \beta \subseteq \alpha)]\varphi \rightarrow ([\alpha]\varphi \rightarrow [\alpha \cap \beta]\varphi)$
mod. pon.	$[\alpha]\varphi \rightarrow [\alpha \cap \beta]\varphi$
duality + subst. of props.	$\langle \alpha \cap \beta \rangle \varphi \rightarrow \langle \alpha \rangle \varphi$

To deduce Disj-R, similar steps can be made.

■

Notice the correspondence of the theorems of proposition 2.5.8 with the following standard modal theorems:

CONJ	$\langle\alpha\rangle(\varphi \wedge \psi) \rightarrow \langle\alpha\rangle\varphi \wedge \langle\alpha\rangle\psi$
DISJ	$\langle\alpha\rangle(\varphi \vee \psi) \leftrightarrow \langle\alpha\rangle\varphi \vee \langle\alpha\rangle\psi$

We saw that the $\sim$-logics allow syntactic definition of the U-operator, that is symmetric, transitive and reflexive, which makes it an S5-modality. In the section after the next one we argue that the S5- properties are in general too strong for a notion that presumes to model 'any action' (the only exception being the case where all action combinators are in the action language). But first we summarize some basic logic properties of $\sim$-logics.

Summary of known $\sim$-logic properties

In the table below we summarize some results on $\sim$-logics from the literature. Note that many logics are missing from the table, which means that still many open 'standard' questions concerning $\sim$-logics abide.

MAL variant	complexity of sat.	modal axiom.	f. m. p.
$(\cup, \sim)$	EXPT.-cpl. [118]	[68] / theorem 2.5.7	yes
$(;, \cup, \sim)$	undec. [181]	[156] (sub-system)	no
$(^{\smile}, ;, \cup, \sim, \varphi?)$	undec. [156]	[156]	no

Table 5. some results from the literature on $\sim$-logics

Many properties that hold for modal logics in general do not carry over to $\sim$-logics. In particular, $\sim$-logic validity is not preserved under (1) disjoint unions, (2) generated sub-models, (3) standard bisimulation. And validity is also not preserved under the notion of ag-bisimulation, that we proved (theorem 2.4.5) to be sufficient for modal action logics involving intersection. De Rijke [156] defines a variant of bisimulation that does preserve $\sim$-logic validity.

Usefulness of $\sim$-logics as logics of action

We saw that $\sim$-logics are strong enough to define intersection and action equivalence. But we also saw that $\sim$-logics have high complexities. But the main reason to reject $\sim$-logics as appropriate for reasoning about action has a semantic nature. We argue from semantic intuitions that the universal complement is not appropriate for modeling the notion of 'alternative action'. Of course

this does not mean that it cannot be a useful construct in other applications of modal logic (e.g. epistemic reasoning).

Our first point is that the universal modality U, that is syntactically definable as $\sim \alpha \cup \alpha$, does not correspond to the notion of 'any action', an interpretation it should have if the complement were to reflect the notion of 'alternative action'. The intuition for the notion of 'any action' is that it subsumes all possible actions, but not more. But the modality U has universal power; it is capable of reaching any state, including the ones that are not reachable by any action (other than the U). In our view, this makes the U too strong for representation of the notion of 'any action'. Nevertheless, some authors on action reasoning in PDL refer to the modality U as the 'any action' [153].

The strength of the complement is also demonstrated by the property: $\langle\sim(\alpha;\beta)\rangle\varphi \rightarrow [\alpha]\langle\sim\beta\rangle\varphi$, which is the dual of $\langle\alpha\rangle[\sim\beta]\varphi \rightarrow [\sim(\alpha;\beta)]\varphi$. Let us assume that the action language does not allow for converse action. And let us retain a local perspective by situating ourselves in a state and by considering the actions α, β, their action complements, and their effects. At first sight, the property seems to describe a reasonable action law: 'if by doing an action complementary to $\alpha;\beta$ I can reach a state where φ, then necessarily, after doing α, I can reach the same state (where φ) by performing an action complementary to β'. But on closer inspection, it becomes clear that the possibility to reach the same state where φ after any possible performance of α, implies that the execution of such an α apparently has not limited our possibilities to reach certain states. This is a non-intuitive property for actions that do not encompass converse, because in general, as the result of performing such actions, the reachable state-space shrinks[4]. The property $\langle\sim(\alpha;\beta)\rangle\varphi \rightarrow [\alpha]\langle\sim\beta\rangle\varphi$ shows that it does not, because it says that all possible ways to perform α do not result in a state from which φ cannot any longer be reached by an action complementary to β.

Also, the universal complement does not combine well with temporal reasoning. For temporal reasoning over action, we assume in chapter 3 that time, in a sense, is 'realized' by the performance of actions. In any case, actions take place in time, and time evolves by the performance of actions. It follows that only if states of affairs can be achieved through action, they belong to the possible futures that have to be considered. Then it is clear that the universal operation U that reaches all possible states cannot function as an action operation, since then many states not reachable by action should have to be

[4] An analogy: at the start of one's life, much more different futures are possible than at the age of 65.

considered as possible future states of affairs.

The universal complement is also not suited to reason about action effects. In section 2.5.1 we explained that a suitable notion of action complement ('$-$') should enable the expression of frame properties. In particular we claimed that it should be possible to use formulas of the form $\neg\varphi \rightarrow [-\alpha]\neg\varphi$ to express that the condition φ can only be changed by α. But his cannot be achieved by using the universal complement. Again, it is too strong. The universal complement in the expression $\neg\varphi \rightarrow [\sim\alpha]\neg\varphi$ does not only say that actions other than (not involving) α cannot have φ as a result, it says in addition that all states in the state-space that embody the result φ can actually be reached by α. This follows from the semantics of the universal complement: any state can be reached from any other state by either α or $\sim\alpha$, and reachability by $\sim\alpha$ contradicts $[\sim\alpha]\neg\varphi$. This means that the universal complement is not suited to encode Reiter's solution to the frame problem [73, 155] into modal action logic.

Note that we do not claim that a modal *formula* of the form $[\sim\alpha]\varphi$ does not have an intuitive reading in terms of the action α. We only claim that the operator $\sim$ does not have an intuitive reading as an action operation. An intuitive reading of $[\sim\alpha]\varphi$ in terms of action is 'φ holds in all states that are not the result of performing the action α'. But this does not give a reading of $\sim$ as an action combinator, because $\sim\alpha$ is *not* an action: it may include 'transitions' that do not correspond to any action at all. In conclusion we might add that we can only use this version of the complement in a logic of action if we see the action as performed by omnipotent agents, that can actually bring about any conceivable state of affairs.

2.5.3 Relativized complement modal action logics

The universal complement introduces an aspect that is not in the spirit of modal logic: globalness. This aspect of the complement is inherited from relation algebra, where the semantic view is also global. The complement we consider in this section is faithful to the idea of locality in modal semantics. This results in a better fit with the interpretation of the modal language as a logic of action. As a side-effect we also expect better complexities for the logics with a relativized action complement. The action language comprising the new, relativized complement is only concerned with the part of the state space that is 'reachable'. It assumes that in general, from any given state at least part of the state space cannot be reached through any action, and that the part of the state space that *is* reachable may vary from state to state. Then,

the general intuition for the alternative complement, denoted $\wr^I$, is that it is taken with respect to all possible relations over this reachable state space. But what may be considered reachable, depends on what action combinators are in the action language. This explains the term 'relativized'. The complement operation is relativized with respect to the part of the state space in a modal action model that (1) is the minimal relation space containing all atomic actions and that (2) is closed under application of the action combinators of the dynamic logic language. Thus, if we allow iteration in the action language, the complement space is reflexive, and if we allow converse, the complement space is symmetric. Therefore we cannot give, as for the universal complement, a general definition of the new complement for all dynamic logics; each dynamic logic comes with its own interpretation for the complement. All in all we define six versions of the relativized complement: $\wr^K, \wr^B, \wr^{S4}, \wr^{K4}, \wr^{B4}, \wr^{S5}$, one for each dynamic logic involving a specific set of action combinators. The annotations give information about the nature of the relation space with respect to which the complement is taken, where we adopt standard terminology from modal logic to refer to transitivity, reflexivity etc. Intuitively, this relation space with respect to which the relativized complement is taken reflects the space of possible alternative complex actions. That this space is relative to the action combinators makes sense, since the syntactic complexity of alternative actions is determined by the action connectives in the action language.

As a consequence of this scattering of interpretations, some action operators that can be syntactically defined in terms of the complement also get alternative interpretations ($any^I, \subseteq^I$). Other combinators undergo no effective change in interpretation ($\cup, \cap, ;, ^*, ^\leftarrow, fail$).

To our knowledge the notion of relativized complement was not studied before in modal logic. Only in the work of De Giacomo [70] a complement can be found that is equivalent with $\wr^K$. But there it was only applied to a boolean action fragment, and not studied in detail.

Semantics of $\wr^I$-logics

We define a series of modal action logics encompassing a relativized action complement operator $\wr^I$. Each subsequent logic in the series introduces a new action operator and redefines the relativized complement operation accordingly. The justification of the definitions is the following. If a complement operation $\wr^I$ is meant to return all actions that are *alternative*, the space of actions with respect to which the complement is defined should not only contain all atomic actions, but should also be closed under the action operations

of the logic. For instance: if sequence is among the action operators of the logic, than also alternative action is possibly of a sequential nature, which means that the complement space, that is, the space with respect to which $\wr^I$ is defined, should be closed under the sequence operation.

Definition 2.5.4 (semantics of $\wr^I$-logics) *Let $\wr^K, \wr^B, \wr^{S4}, \wr^{K4}, \wr^{B4}, \wr^{S5}$, be action complement connectives for the logics $MAL(\cup, \wr^K)$, $MAL(^{\leftarrow}, \cup, \wr^B)$, $MAL(;, \cup, \wr^{K4})$, $MAL(^*, ;, \cup, \wr^{S4})$, $MAL(^{\leftarrow}, ;, \cup, \wr^{B4})$, $MAL(^*, ^{\leftarrow}, ;, \cup, \wr^{S5})$ respectively. We refer to this type of logics as $\wr^I$-logics, where I is an annotation referring to properties of the relational space with respect to which the complement $\wr^I$ is taken. The semantics of each individual $\wr^I$-logic is determined by the modal semantics of definition 2.1.2, together with a relational interpretation function R for actions, that follows by selection of the relevant clauses of definition 2.4.1 in combination with the relevant clause for the relativized action from the following list:*

$$
\begin{array}{lrcl}
\textit{for } MAL(\cup, \wr^K): & R(any^K) & = & \bigcup\limits_{a \in \mathcal{A}} R(a) \\
\textit{for } MAL(^{\leftarrow}, \cup, \wr^B): & R(any^B) & = & \bigcup\limits_{a \in \mathcal{A}} (R(a) \cup R(a^{\leftarrow})) \\
\textit{for } MAL(;, \cup, \wr^{K4}): & R(any^{K4}) & = & (\bigcup\limits_{a \in \mathcal{A}} R(a))^+ \\
\textit{for } MAL(^*, ;, \cup, \wr^{S4}): & R(any^{S4}) & = & (\bigcup\limits_{a \in \mathcal{A}} R(a))^* \\
\textit{for } MAL(^{\leftarrow}, ;, \cup, \wr^{B4}): & R(any^{B4}) & = & (\bigcup\limits_{a \in \mathcal{A}} (R(a) \cup R(a^{\leftarrow})))^+ \\
\textit{for } MAL(^*, ^{\leftarrow}, ;, \cup, \wr^{S5}): & R(any^{S5}) & = & (\bigcup\limits_{a \in \mathcal{A}} (R(a) \cup R(a^{\leftarrow})))^*
\end{array}
$$

$$R(\wr^I \alpha) = R(any^I) \setminus R(\alpha)$$

The logics of definition 2.5.4 are not the only possible $\wr^I$-logics. Given that besides the complement we consider four independent action operations (choice, sequence, converse, iteration), we have a total of $2^4 = 16$ different $\wr^I$-logics. But many of these are rather eccentric. Examples are MAL($\wr^K$), which has the same complement as MAL($\cup, \wr^K$) and MAL($^*, \cup, \wr^{S4}$), which has the same complement as MAL($^*, ;, \cup, \wr^{S4}$). We do not consider these logics, since they seem to have less relevancy for the application to reasoning about action.

We introduce the relativized *any* and the relativized *subsumption action* as syntactic extensions by defining them in terms of the relativized complement ($\wr^I$) and the choice ($\cup$). Note that we already defined the relativized any as part of the definition of the relativized action complement above. But we

identify dynamic logics by reference to their base-combinators, that is, sets of action combinators that are not definable in terms of others. We can define the *any* in terms of the complement, but not the other way round.

Definition 2.5.5 *The relativized any, and the relativized subsumption action are defined through the following syntactic extensions on* $\cup$ *and* $\wr^I$*:*

$$\begin{array}{lll} \alpha \subseteq^I \beta & \equiv_{def} & \wr^I \alpha \cup \beta \\ any^I & \equiv_{def} & \alpha \cup \wr^I \alpha \end{array}$$

The relativized versions of the *any* and the *subsumption action* differ from their non-relativized counterparts. The relativized *any* does not reach the complete state space, but only the part that is reachable through (complex) action, as determined by the action language. The next propositions states that the *intersection, equivalence action* and *fail* have the same interpretation as their relativized counterparts. In section 2.4.1, plain intersection was argued to represent concurrency in an intuitive way. Therefore this proposition is in support of the intuitive correctness of $\wr^I$-logics for reasoning about concurrent action.

Proposition 2.5.9 *For any two actions* α *and* β *in the action language of a* $\wr^I$*-logic the following holds for the relational interpretations* $R(\alpha)$ *and* $R(\beta)$ *on a modal action frame* $\mathcal{F} = (S, R^{\mathcal{A}})$*:*

$$\begin{array}{lcllll} R(\alpha \cap \beta) & = & R(\alpha \cap^I \beta) & with & \alpha \cap^I \beta & \equiv_{def} \wr^I(\wr^I \alpha \cup \wr^I \beta) \\ R(\alpha \div \beta) & = & R(\alpha \div^I \beta) & with & \alpha \div^I \beta & \equiv_{def} (\alpha \subseteq^I \beta) \cap (\beta \subseteq^I \alpha) \\ R(fail) & = & R(fail^I) & with & fail^I & \equiv_{def} \alpha \cap \wr^I \alpha \end{array}$$

Proof

We prove the first equivalence. From the action semantics for the universal complement and the relativized complements it follows that $R(\wr^I \alpha) = R(\sim(\alpha \cup \sim any^I))$. Therefore we can substitute the action $\sim(\alpha \cup \sim any^I)$ for actions $\wr^I \alpha$ in the definition for $\alpha \cap^I \beta$. This returns $\sim(\sim(\alpha \cup \sim any^I) \cup \sim(\beta \cup \sim any^I) \cup \sim any^I)$. By applying standard boolean properties we arrive at the equivalent $\sim(\sim((\alpha \cup \sim any^I) \cap any^I) \cup \sim((\beta \cup \sim any^I) \cap any^I))$. Again, by applying standard boolean properties we arrive at the equivalent action (1) $\sim(\sim((\alpha \cap any^I) \cup (\sim any^I \cap any^I)) \cup \sim((\beta \cap any^I) \cup (\sim any^I \cap any^I)))$. Now we focus on the actions (2) $\alpha \cap any^I$ and (3) $\sim any^I \cap any^I$. Action 3 is equivalent with the impossible action $fail$, as follows from the meaning of $\sim$. Action 2 is

equivalent with α, since any^I is the action that subsumes any other action: for any α it holds that $R(\alpha) \subseteq R(any^I)$. Substitution of these equivalent actions into action 1 results in $\sim(\sim\alpha \cup \sim\beta)$, which is equivalent with $\alpha \cap \beta$.

■

For any specific $\wr^I$-logic, we talk of its $\wr^I$-reduced sub-logic if we mean the logic that results by removal of the complement, and we talk of its $\sim$-logic counterpart, if we mean the corresponding logic where the $\wr^I$-operator is replaced by the $\sim$-operator. In the same way we talk about $\sim$-logic counterpart formulas, validities, validity schemes, etc.

Proposition 2.5.10 *For any $\wr^I$-logic, the relativized complement operator $\langle\wr^I\alpha\rangle\varphi$ is not syntactically definable in its $\wr^I$-reduced sub-logic.*

Proof

For all $\wr^I$-reduced sub-logics state validity is preserved under bisimulation (in specific cases generalized to deal with converse action). State validity of for instance $\langle a \cap b\rangle P$ (where we use proposition 2.5.9, saying that plain intersection is syntactically definable in any of the relativized modal action logics) is not preserved under bisimulation. The proposition can be made stronger, since for instance $\wr^{S4}$ is also not definable in MAL$(^*, ;, \cup, \cap)$, because state validity of $\langle\wr^{S4}a \cup \wr^{S4}b\rangle\neg P$ is not preserved under ag-bisimulation.

■

We will prove that $\wr^I$-logics are subsumed by their $\sim$-logic counterparts (under syntactic replacement of the complements). In this sense they are 'weaker'. But (with one exception) it is not the case that $\wr^I$-logics can be syntactically defined by their $\sim$-logic counterparts. We will prove this for the two weakest $\wr^I$-logics.

Proposition 2.5.11 *For the logics MAL$(\cup, \wr^K)$ and MAL$(\cup, ^\leftarrow, \wr^B)$, the relativized complement operator $\langle\wr^I\alpha\rangle\varphi$ is not syntactically definable in the $\sim$-logic counterparts.*

Proof

In section 2.5.2 we mentioned that iteration-free $\sim$-logics are compact. The relativized complements $\wr^K$ and $\wr^B$ are not. For the logic MAL$(\cup, \wr^K)$, it is straightforward to verify that the set $\{\langle\wr^K a\rangle\neg P, [b]P, [c]P, \dots\}$ (where $\{a, b, c, \dots\} = \mathcal{A}$), is not satisfiable while all of its finite sub-sets are. For the logic MAL$(\cup, ^\leftarrow, \wr^B)$, the absence of the compactness property follows from the set $\{\langle\wr^B(a \cup a^\leftarrow)\rangle\neg P, [b]P, [b^\leftarrow]P, [c]P, [c^\leftarrow]P, \dots\}$.

■

We will show in chapter 3 how to use the relativized action complement for temporal reasoning over action, and we will show in chapter 5 how to use it for deontic reasoning. To stress the intuitive adequacy of the complement to reason about action, we only mention here that it can be used to encode Reiter's solution of the frame problem. In section 2.5.1 we argued that the universal complement is not suitable for the encoding of Reiter's solution to the frame problem in modal action logics, because the expression $\neg\varphi \rightarrow [\sim\alpha]\neg\varphi$ does not only say that actions other than (not involving) α cannot have φ as a result, it says in addition that all states in the state-space that embody the result φ can actually be reached by α. The relativized complement version of this expression, i.e. $\neg\varphi \rightarrow [\wr^I\alpha]\neg\varphi$ does not have this unintended additional consequence: only for the states that are reachable through complex action it holds that if φ holds in them, they can be reached by α. Note that we can introduce a relativized version of the window operator of definition 2.5.3 to abbreviate $[\wr^I\alpha]\neg\varphi$.

Validities

In this section we prove that each $\wr^I$-logic represents a specific weakening of its $\sim$-logic counterpart, and that the series of $\wr^I$-logics themselves are partially ordered in strength. Also we show that this ordering is strict, by presenting validity schemes witnessing the differences in strength. First we prove:

Theorem 2.5.12 *Under syntactic replacement of $\wr^I$ with $\sim$, each validity for a given $\wr^I$-logic turns into a validity for its $\sim$-logic counterpart.*

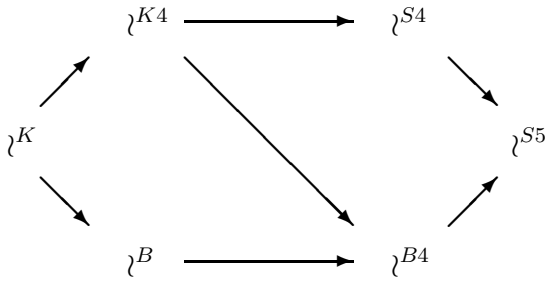
Proof

Through negative demonstration. Assume that the formula $\varphi^{\wr^I}$ is a $\wr^I$-validity (so it is not a validity scheme). Denote its $\sim$-logic counterpart formula by $\varphi^{\sim}$. Now assume that $\varphi^{\sim}$ is not a validity. Under this assumption, we show how to construct a model on which $\varphi^{\wr^I}$ is not valid, thereby proving the theorem. If $\varphi^{\sim}$ is not a validity, it follows that there is a model $\mathcal{M}$ and a state s such that $\mathcal{M}, s \not\models \varphi^{\sim}$. Define $aa(\varphi^{\sim})$ to be the set of atomic actions occurring in the formula $\varphi^{\sim}$. Now construct the model $\mathcal{M}'$, by adding to $\mathcal{M}$ the interpretation of an atomic action r such that $r \notin aa(\varphi^{\sim})$ and $R^{\mathcal{A}}(r) = S \times S \setminus any^I$. It follows that $\mathcal{M}', s \not\models \varphi^{\sim}$, because the addition of the new relation does not alter the truth-condition for $\varphi^{\sim}$: (1) all actions in $aa(\varphi^{\sim})$ are equally interpreted in $\mathcal{M}$ and $\mathcal{M}'$, and (2) all action connectives are equally interpreted, in particular, addition of $R^{\mathcal{A}}(r)$ does not in any way change the interpretation of actions of the form $\sim\alpha$, because the universal

relation is not affected, (3) all valuations of propositions remain as they are. But now, due to the addition of the extra relation, for any a such that $a \in aa(\varphi^{\sim})$, the interpretation of the relation $\wr^I a$ on $\mathcal{M}'$, is exactly the same as the interpretation of $\sim a$ on $\mathcal{M}$. It is easy to see that this does not only hold for the interpretation of negated atomic actions, but that the interpretation of any compound action $\wr^I \alpha$ on $\mathcal{M}'$, is exactly the same as the interpretation of $\sim \alpha$ on $\mathcal{M}$. But then, from the information that $\mathcal{M}'$ and $\mathcal{M}$ agree on all valuations of atomic proposition, it follows that the truth conditions for a formula $\varphi^{\wr^I}$ on $\mathcal{M}'$ is necessarily equal to the truth condition for $\varphi^{\sim}$ on $\mathcal{M}$. So it holds that $\mathcal{M}', s \not\models \varphi^{\wr^I}$. So there is a model, namely $\mathcal{M}'$, for which the formula $\varphi^{\wr^I}$ is not valid. This contradicts the assumption we started off with. ■

Second, we show that the insight used in the above proof can also be used to compare the relativized modal action logics mutually.

Theorem 2.5.13 *Under replacement of complements in formulas, the following inclusion relation between $\wr^I$-logics holds. In the picture, logics are represented by the type of complement they endorse, and arrows denote inclusion.*



Proof

We can use the same proof strategy as for theorem 2.5.12. If we want to prove that for a logic encompassing the complement $\wr^I$, the validities are included in a logic encompassing the complement $\wr^J$, we make use of the addition of a relation $R^{\mathcal{A}}(r)$ for an atomic action r, defined as $R^{\mathcal{A}}(r) = any^J \setminus any^I$. On all other aspects the proofs are completely analogous to the proof of theorem 2.5.12. This implies that inclusion of logics is determined by inclusion of the respective complement spaces. So the above partial order simply follows definition 2.5.4 by inspecting the inclusion relations between the spaces with respect to which the respective complements are defined.

■

It is clear right away that the above inclusions are strict since stronger logics encompass supersets of non-syntactically definable action connectives. But it is illustrative to see how the logics can also be distinguished by validity schemes concerning action connectives that are available in (almost) all relativized complement logics, such as the '$\cup$', the '$\wr^I$' and the ';'. We already investigated such validities for $\rightsquigarrow$-logics in section 2.5.2, and know that the logics in this section weaken these logics. So we simply check which of the validities we encountered in section 2.5.2 are affected. First we note that many validity schemes are shared by all $\wr^I$-logics and $\sim$-logics. One example is the property K-R:

$$\models \langle\alpha\rangle\varphi \wedge [\wr^I\beta]\neg\varphi \rightarrow \langle\alpha \cap \beta\rangle\varphi \qquad \text{(K-R)}$$

Other properties are valid for some $\wr^I$-logics and invalid for others. We recall the following possible validity schemes for $\wr^I$-logics:

$$\begin{array}{ll} \models \langle any^I\rangle\langle any^I\rangle\phi \rightarrow \langle any^I\rangle\phi & \text{(Trans-any)} \\ \models \phi \rightarrow [any^I]\langle any^I\rangle\phi & \text{(Symm-any)} \\ \models \phi \rightarrow \langle any^I\rangle\phi & \text{(Refl-any)} \\ \models \langle\alpha\rangle[\wr^I\beta]\phi \rightarrow [\wr^I(\alpha;\beta)]\phi & \text{(NegSeq-R)} \end{array}$$

Now the following proposition lists a table saying which properties hold for which logic.

Proposition 2.5.14 (validity schemes of $\wr^I$-logics)

MAL variant(s)	*Trans-any*	*Symm-any*	*Refl-any*	*NegSeq-R*
$(\cup, \wr^K)$	*no*	*no*	*no*	*no*
$(^{\leftarrow}, \cup, \wr^B)$	*no*	*yes*	*no*	*no*
$(;, \cup, \wr^{K4})$	*yes*	*no*	*no*	*no*
$(^*, ;, \cup, \wr^{S4})$	*yes*	*no*	*yes*	*no*
$(^{\leftarrow}, ;, \cup, \wr^{B4})$	*yes*	*yes*	*no*	*yes*
$(^*, ^{\leftarrow}, ;, \cup, \wr^{S5})$	*yes*	*yes*	*yes*	*yes*

Table 6. validity schemes for $\wr^I$-logics

Proof

Most properties follow by straightforward verification. That the property NegSeq-R is only valid for the strongest two logics takes the following insight.

First of all it takes the sequence operator to express the property, which already rules out half of the $\wr^I$-logics. Second it requires symmetry of the space with respect to which the complement $\wr^I$ is taken. Without symmetry, $\langle\alpha\rangle[\wr^I\beta]\varphi \rightarrow [\wr^I(\alpha;\beta)]\varphi$ is easily refuted in the state s of a model $\mathcal{M}$ with two states s and t such that $\mathcal{M}, t \models \neg\varphi$, and $(s,t) \in R(\alpha)$ (it is also easy to check that by making the complement space symmetric, that is, by including the converse direction of α in it, the model is no longer a counter example to the scheme, because then $\mathcal{M}, t \not\models [\wr^I\beta]\varphi$ and thus $\mathcal{M}, s \not\models \langle\alpha\rangle[\wr^I\beta]\varphi$). Symmetry of the complement space holds only for $\wr^I$-logics encompassing the converse operation.

■

It might come as a surprise that the scheme $\phi \rightarrow \langle any^I\rangle\phi$, representing reflexivity, only holds in the logic with $\wr^{S4}$ and $\wr^{S5}$, and not for the logic with $\wr^{B4}$. The complement space for the logic with $\wr^{B4}$ is both transitive and symmetric. Transitivity and symmetry together imply reflexivity. But in the present setting, transitivity and symmetry only holds for the *reachable* relation space, which results in a relativized reflexivity with respect to the *complete* state space: the reflexivity only holds for states from which action is actually possible.

The above table not only lists properties that distinguish $\wr^I$-logics mutually, it also forms a comparison of $\wr^I$-logics with their $\sim$-logic counterparts. For instance, it shows that $\mathrm{MAL}(\cup, \wr^K)$ is considerably weaker than its sibling $\mathrm{MAL}(\cup, \sim)$, because it does not support transitivity, or reflexivity or symmetry for the any^K. But the logics lower in the table are closer to their $\sim$-logic counterparts. And we now prove that the final and strongest one, $\mathrm{MAL}(^*, ^\leftarrow, ;, \cup, \wr^{S5})$, is actually equivalent with its $\sim$-logic counterpart $\mathrm{MAL}(^*, ^\leftarrow, ;, \cup, \sim)$.

Theorem 2.5.15 *Under syntactic interchange of occurrences of $\wr^{S5}$ and $\sim$, the logics $MAL(\wr^{S5}, \cup, ;, ^*, ^\leftarrow)$ and $MAL(\sim, \cup, ;, ^*, ^\leftarrow)$ encompass exactly the same validities. Thus, the logics are equivalent.*

Proof

Theorem 2.5.12 already proved that under syntactic interchange of occurrences of $\wr^{S5}$ and $\sim$, it holds that $\mathrm{MAL}(\wr^{S5}, \cup, ;, ^*, ^\leftarrow) \subseteq \mathrm{MAL}(\sim, \cup, ;, ^*, ^\leftarrow)$. Here we have to prove that also $\mathrm{MAL}(\wr^{S5}, \cup, ;, ^*, ^\leftarrow) \supseteq \mathrm{MAL}(\sim, \cup, ;, ^*, ^\leftarrow)$. Again we rely on a proof through negative demonstration. Assume that $\varphi^\sim$ is a $\sim$-validity. Denote its $\wr^{S5}$-logic counterpart formula by $\varphi^{\wr^{S5}}$. Now assume that $\varphi^{\wr^{S5}}$ is not a validity. Under this assumption, we show how to construct a model on which $\varphi^\sim$ is not valid, thereby proving the theorem. If $\varphi^{\wr^{S5}}$ is

not a validity, it follows that there is a model $\mathcal{M}$ and a state s such that $\mathcal{M}, s \not\models \varphi^{\wr^{S5}}$. Define $aa(\varphi^{\wr^{S5}})$ to be the set of atomic actions occurring in the formula $\varphi^{\wr^{S5}}$. Now construct the model $\mathcal{M}'$, by contraction to the generated sub-model of state s (simply remove all non-reachable states, where reachability also accounts for the converse direction of relations). It follows that $\mathcal{M}', s \not\models \Phi^{\wr^{S5}}$, since the truth function does not get another value by this contraction. But now, for any a such that $a \in aa(\varphi^{\wr^{S5}})$, the interpretation of the relation $\sim a$ on $\mathcal{M}'$, corresponds one to one to the interpretation of $\wr^{S5}a$ on $\mathcal{M}$. It is easy to see that actually the interpretation of any compound relation and well-formed formula using $\sim$ on $\mathcal{M}'$ corresponds one to one to the interpretation of a compound relation and well-formed formula using $\wr^{S5}$ on $\mathcal{M}$. But then it holds that $\mathcal{M}', s \not\models \varphi^{\sim}$, where $\varphi^{\sim}$ corresponds to $\varphi^{\wr^{S5}}$ with $\sim$ in place for $\wr^{S5}$. It follows that there is a model $\mathcal{M}'$ for which the scheme $\varphi^{\sim}$ is not valid. This contradicts the assumption we started off with.

■

We want to elaborate on the role of the scheme NegSec-R in the distinction between logics. The property is not relevant for the comparison of MAL$(\cup, \wr^K)$ and MAL$(\cup, \sim)$, since neither of these allow expression of the complement of sequence.[5] But for logics higher in the partial order, the property plays an interesting role. An important way in which the relativized complement differs from the universal complement is that, in case we do not have the converse in the action language, the reachable state space might change when going from one state to another. For the logics with universal complement, this does not hold, because with the universal relation, from any state in a model we can reach *any* other state. The difference between these two semantic choices is demonstrated by the way the complement interacts with the sequence. For logics with universal complement we have $\langle\alpha\rangle[\sim\beta]\varphi \rightarrow [\sim(\alpha;\beta)]\varphi$. We can read this as 'if we can do an α after which the state space not reachable through β obeys φ, then the state-space not reachable through $\alpha;\beta$ obeys φ'. So if we want to impose φ on the state-space that is complementary to the space reachable by $\alpha;\beta$, it is possible to first perform α and impose φ on the complement of states reachable by β afterwards. This is not possible in the (converse-free) logics with relativized complement. If we want to impose φ on the state-space that is complementary to the space reachable by $\alpha;\beta$, it is not possible to first perform α, and to try to impose φ on some reachable state-space φ afterwards.

[5] We might introduce in MAL$(\cup, \wr^K)$ a notion of sequence through syntactic extension ($[\alpha;\beta]\phi \equiv_{def} [\alpha][\beta]\phi$). But that does not bring us the expressiveness of the logic MAL$(;, \cup, \wr^{K4})$, because there is for instance no formula equivalent to $[\wr^{K4}(\alpha;\beta)]\phi$.

The reason is that by doing an α, we cut off reachability of many states that are in the complement of the $\alpha;\beta$-reachable states. Addition of the converse to the action language brings us back symmetry of the connected relation space. So we get symmetry of the complement (any^{B4}), and global reachability ($NegSeq - R$). This shows that from an action perspective, the converse seems to be a more powerful connective than from a mere relational perspective, because in the MAL versions with universal complement, symmetry and global reachability already holds for the variants without converse.

Definability of classes of models and frames in $\wr^I$-logics

In section 2.4.3 we saw that logics with intersection but without complement were not strong enough to define intersection and relation equivalence at the level of models. In section 2.5.2 we proved that $\sim$-logics were strong enough for this task. But also the weaker $\wr^I$-logics accomplish this.

Proposition 2.5.16 *In $\sim$-logics, the class of models $\mathcal{C}_{R(\alpha)=R(\beta)}$, for which $R(\beta) = R(\alpha)$, is defined by $[\wr^I(\alpha \div^I \beta)]\varphi$.*

Proof

The proof is completely analogous to that of proposition 2.5.6.

■

As a corollary we get that all action operations are definable at the level of models. In particular, the class of models $\mathcal{C}_{R(\gamma)=R(\alpha\cap\beta)}$, for which $R(\gamma) = R(\beta) \cap R(\alpha)$, is defined by $[\wr^I(\gamma \div^I \alpha \cap^I \beta)]\varphi$. Note that in these formulas we might replace $\div^I$ and $\cap^I$ by their non-relativized counterparts, since these were shown to have exactly the same interpretation.

Preservation of state validity

$\wr^I$-logics are stronger than $\cap$-logics. Therefore, semantic equivalence notions for $\wr^I$-logics should be obtained as strengthenings of the equivalence notion for $\cap$-logics, that is, ag-bisimulation (definition 2.4.9). Whereas in $\cap$-logics we can express that states are reachable through certain action graphs, in $\wr^I$-logics we can express additionally that certain states are *not* reachable through certain action graphs. For instance, in terms of action graphs, the formula $\langle\wr^I\alpha\rangle\varphi$ means that there is a state where φ holds that is reachable by some action graph, but not by any action graph that interprets the action α. To incorporate this non-reachability aspect, we have to generalize definition 2.4.7 concerning the graph interpretation of action. Essentially we will get a relativized notion

of semantic equivalence, i.e. semantic equivalence notion for each separate $\wr^I$-logic. The semantic equivalence notion we defined for $\cap$-logics are used in chapter 4 to base notions of 'semantic in-equivalence' on. But for $\wr^I$-logics we will not consider these. Therefore we leave semantic equivalence notions for these logics as a subject for future research.

Complexity of $\wr^I$-logics

In the proof of proposition 2.5.11 we showed that $\wr^I$-logics are not compact. This shows that the semantics of relativized complement contains a second order aspect (like the iteration). Therefore we may expect higher complexities for $\wr^I$-logics w.r.t. their $\sim$-logic counterparts. But on the other hand, relativization is known to result in better complexity properties. The work of Marx [125] on relativized relation algebra studies relativizations of relation algebras with respect to 'background' relations. He shows that relativization of relation algebras has a positive effect on their complexity. Although our form of relativization is more specific, we expect a better complexity for our $\wr^I$-logics with respect to their $\sim$-logic counterparts. Relativization can be seen as a way to adapt the semantics of formulas in order to allow more models, which should make the satisfaction problem easier. Having more models corresponds with having less validities. The validities of proposition 2.5.14 are examples of properties that hold for $\sim$-logics but not necessary for their $\wr^I$-logic counterparts.

To prove complexity results we expect that we can use the local character of the relativized complement. This local character enables us to define generalized tree properties such as defined by Marx and Venema [126]. A particular interesting question is whether the logic $\mathrm{MAL}(;, \cup, \wr^{K4})$ is decidable. The presence of sequence is likely to cause undecidability. Transitivity introduces undecidability in first order generalizations of modal logic such as the two variable and guarded fragments. But for some traditional incarnations of modal logic that were developed for the temporal and process domains, transitivity is a common feature that does not cause undecidability.

Decidability for the stronger logics $\mathrm{MAL}(\wr^{S4}, \cup, ;, ^*)$ and $\mathrm{MAL}(\wr^{B4}, \cup, ;, ^\leftarrow)$ is unlikely. But for the strongest $\wr^I$-logic, undecidability of the satisfiability problem is certain.

Theorem 2.5.17 *The satisfiability problem for* $MAL(\wr^{S5}, \cup, ;, ^*, ^\leftarrow)$ *is undecidable.*

Proof

Directly from theorem 2.5.15 saying that the logic is equivalent with the undecidable logic MAL($\sim, \cup, ;, ^*, ^\leftarrow$).

■

We conjecture that most $\wr^I$-logics have a better complexity than their $\sim$-logic counterparts, but that the difference vanishes for the strongest $\wr^I$-logics.

Axiomatization of $\wr^I$-logics

It is possible that dropping the axioms concerning the transitivity, symmetry and reflexivity of the U from the complete axiomatization of MAL($\cup, \sim$) in theorem 2.5.7, returns a (weakly[6]) complete axiomatization for MAL($\cup, \wr^K$). Axiom systems for stronger $\wr^I$-logics can then be obtained by supplementing this base system with the well-known axioms for iteration, converse, etc., and the axioms that are listed as validity schemes in proposition 2.5.14. We do however not perform this investigation here.

Summary of $\wr^I$-logic properties

In the table below we summarize properties of MAL variants with a relativized complement.

MAL variant	complexity of sat.	f. m. p.
$(\cup, \wr^K)$	PSPACE compl. (conj.)	yes
$(^\leftarrow, \cup, \wr^B)$	PSPACE compl. (conj.)	yes
$(^*, ^\leftarrow, ;, \cup, \wr^{S5})$	undec. (theorem 2.5.17)	no

Table 7. some properties of $\wr^I$-logics

The conjecture about PSPACE complexity of MAL($^\leftarrow, \cup, \wr^B$) is based on the insight from the proof of theorem 2.3.2, that we can safely push down converse operations to the atomic action level. Many $\wr^I$-logics and $\wr^I$-logic properties are missing from the table, indicating that several standard logic questions concerning $\wr^I$-logics are still open.

2.5.4 Complement and deterministic action

There are two different levels on which determinism can be assumed: (1) at the level of atomic actions, in order to obey $\langle a \rangle \varphi \rightarrow [a]\varphi$ for atomic actions

[6]Such an axiomatization is not likely to be strongly complete due to the incompactness of $MAL(\cup, \wr^K)$.

a, and (2) at the level of compound actions, in order to obey $\langle\alpha\rangle\varphi \rightarrow [\alpha]\varphi$ for complex actions α. The first strengthening corresponds to interpretations over models for which each relation $R^{\mathcal{A}}(a)$ is a partial function assigning a unique a-successor state to every state where the action a can be executed. Such models are called *deterministic models*. In general, interpretation over deterministic models leads to higher complexities. For deterministic models, addition of the global (universal) modality $[U]\varphi$ to the logic MAL$(;,\cap)$ returns an undecidable logic ([19] p.367), and addition of the 'master' modality, which is equivalent to our $[any^{S4}]\varphi$, to the logic MAL$(;,\cap)$ results in a logic that is *highly* undecidable ([19] p.371). This follows from the encoding of tiling problems. It follows that also the logic MAL$(^*,;,\cup,\wr^{S4})$ is *highly* undecidable on deterministic models.

For logics for which we assume deterministic *compound* action, the opposite holds: they have a low complexity. This is because obedience of $\langle\alpha\rangle\varphi \rightarrow [\alpha]\varphi$ can only be accomplished by imposing severe restrictions on the action syntax (very tight restrictions on the application of $\cup$ and * to exclude non-deterministic choices). We want to note that determinism on the level of compound actions does *not* mean that we only have to consider models that are traces, and that bisimulation can be replaced by trace-equivalence. The reason is that we still have a multi-modal language. Each program reaches maximally one state (up to bisimulation), but in general more than one program (up to program equivalence) is possible from a specific state. If we have such a situation, where two programs α and β with $R(\alpha) \neq R(\beta)$ are possible from a given state, *syntactically* there is no program γ allowed for which $R(\alpha) \cup R(\beta) \subseteq R(\gamma)$, because this would be a non-deterministic program. However, the notion of deterministic *compound* action completely *collapses* if *complement* enters the stage. We have to distinguish between the traditional $\sim$ and our $\wr^I$. For the logics with $\sim$, deterministic reasoning will have to obey $\langle U\rangle\varphi \rightarrow [U]\varphi$. This results in a collapse to propositional logic, since it imposes that all states in models obey exactly the same formulas. For the $\wr^I$-logics we distinguish two cases: the cases that obey reflexivity, and the cases that do not. The $\wr^I$-logics obeying reflexivity also collapse to propositional logic, since under this condition, $\langle any^I\rangle\varphi \rightarrow [any^I]\varphi$ imposes that all that holds in the current state also holds in all states reachable through (complex) action. The $\wr^I$-logics *not* obeying reflexivity collapse to modal logic over frames with maximally two states. Clearly these logics have low complexities.

2.6 Conclusions

In this chapter we studied the modal logics of action action composition that form a basis for the rest of the work in this thesis. We focussed on the notions of intersection, and action complement. Intersection was argued to represent concurrency in an intuitive way, provided we adopt the open action paradigm. Action complement was argued to be important for the combination of action reasoning with reasoning about change, temporal reasoning and with normative reasoning. Discontent with the standard notion of action complement led to the definition of a relativized action complement. Logics with this action complement feature a number of attractive properties: (1) intuitive correctness, (2) syntactic definability of the (non-relativized) intersection operation, (3) definability of intersection and of action equivalence in terms of classes of models, and (4) better complexity properties. Not all relevant aspects of modal action logics with a relativized complement were dealt with: some interesting questions concerning complexity and axiomatization were left for future research.

Chapter 3

Temporalizing modal action logics

The temporal dimension is traditionally deemed very important for reasoning about reactive system behavior [124]. The interpretation domain for temporal system properties is formed by the points in time that are attended in the behavioral history of a system. We call these points in time *moments* (other terms that can be found in the literature are: 'worlds', 'states', 'situations'). To model the reasoning about the temporal evolution of reactive system properties, many temporal logics have been developed [149, 47, 187, 24, 124, 48], some of which have incompatible conceptions of the structure of time. Temporal logics in general abstract from dynamic aspects of the structure in which moments are related, i.e. from the actions and processes through which moments 'result' from other moments. Typical temporal properties are 'liveness', expressing that certain conditions will occur eventually or even repeatedly (fairness), and 'safety', expressing that certain conditions are preserved over time.

In this chapter we study how to interpret temporal properties on modal action models, and to what extent temporal properties are already expressible by the modal action logics of chapter 2. The chapter consists mainly of definitions concerning combined action / time logics, and their motivations. Many of the relevant logic properties for the combinations follow from the individual logic fragments. We sometimes mention these properties, but the main focus will be on the semantic definitions of the combined logics. We first study how we are to define the common semantic ground for temporal formulas and action formulas.

3.1 Temporal interpretations on action models

We investigate how to interpret temporal properties on modal action models. We restrict our attention to discrete, non-dense, Ockhamist ([154, 195, 39]) time. A conception of time is 'Ockhamist' if it assumes determinism in the past direction. Non-Ockhamist time may branch in the past direction. The two other restrictions, that is, discreteness and non-density, point to the involvement of a specific temporal notion: the next moment in time. Another central temporal notion is that of the future in its global extensiveness. In order to interpret temporal operations, we have to make these two basic temporal notions concrete in modal action models. We do that by representing them by special relations in modal action models. We denote the first, the next time relation, with the relation R_X, and the second, the global time relation, with R_G. The relations represent the following information: if $(s,t) \in R_X$, then for any moment the system is in state s the next moment in time the system is possibly in state t, and if $(s,t) \in R_G$, then for any moment that the system is in state s there is a possible future moment in which the system is in state t. Note that R_X and R_G do not explicitly relate moments, but system states that may occur at certain moments. This leaves open the possibility that system states may correspond to more than one moment.

To see how R_X and R_G can be defined for the modal action logics we studied in chapter 2, we consider several requirements for a temporal view on action structures. Some have to do with desired properties for time itself, while others stem from our basic viewpoint that it is through action that, in a sense, time is 'realized'. We begin with requirements of the first type. These are requirements for R_X and R_G that do not relate to their use in an action context, but follow from temporal intuitions alone. We denote infinite sequential composition of a relation R by R^ω, and infinite transitive closure by R^∞, which gives that $R^\infty = R^\omega \cup R^*$. The temporal requirements are:

(1) $(R_X)^\infty = R_G$

(2) Let $I \equiv_{def} \{(s,s) \mid s \in S\}$ and $R_X^{irr} \equiv_{def} R_X \setminus I$. Then $R_X^{irr} \circ R_X^{irr} \subseteq (R_G \setminus R_X^{irr})$.

(3) There is no R_G^{halt} such that $R_G^{halt} \subseteq R_G$ and $R_G^{halt} \circ R_X = \emptyset$.

Requirement 1 says that the set of states that is possibly attended after the passing of a zero, finite or countably infinite number of next moments in time equals the set of possible future states, including the current state. Obedience of this requirement has two intuitive implications: (1) any state that comes

next in time belongs to a possible future, that is, $R_X \subseteq R_G$, and (2) R_G is closed under transitive composition: $R_G \circ R_G \subseteq R_G$. This reflects the general temporal law that time is transitive: 'if I am born before my brother, and my brother is born before my sister, then I am born before my sister'. Note that the requirement can be used as a definition of the relation R_G in terms of R_X. We thus only have to define the relation R_X on modal action models to enable the interpretation of temporal modalities over them. Requirement 2 follows from the observation that a state that is next in time cannot really be a state next in time if it is possible to consider states at moments in between. It says that under the condition that we cannot stay in the current system state when going to a next moment in time (irreflexivity of R_X^{irr}), we cannot reach the same system state by both one and two minimal time steps. Requirement 3 says that there cannot be future states where time comes to a halt: time goes on forever, whichever branch is followed. Note that requirement 3 does not follow from requirement 1: reflexivity of R_G does not guarantee that there are no states from where no other states are reachable through R_X. It is clear that R_X and R_G play opposite roles as the minimal time lapse and any time lapse respectively.

But should we not demand additionally that both R_X and R_G are irreflexive, in order to mirror that time is considered to be loop free? The answer is no, because logics for Ockhamist time cannot distinguish loops from their 'unravelings'. So the temporal reasoning will not run into paradoxical time circularities whenever a loop is formed by the relations R_X and R_G; the temporal languages we consider are not strong enough to 'detect' them. We may also explain this in another way: with any system state that is begin and end point of an action loop, we can associate a countably infinite set of time points, corresponding to the infinitely many times a lapse through the loop may occur. That in the model, this countably infinite set of moments is represented by one state, is justified by the fact that the time logics we consider are not strong enough to ascertain this identification of states: they only 'see' an infinite set of moments[1]. Note that modal action logics with *skip* (definition 2.2.2) and $\cap$ are strong enough to define loops. In particular, $\psi \rightarrow \langle a \cap \mathit{skip}\rangle \top$ expresses that in states where ψ holds it is possible to perform a in such a way that the current state is reached. So, we may say that in these states we have a 'circular' action (which is an action with no effect). We thus have circularity in modal action models for this formula. But as just explained, this does not

[1]Note that this is exactly why modal languages for reasoning about infinite behavior in many cases do have the finite model property: countably infinite traces can be encoded as loops in finite models.

imply that time is circular.

We now turn to the requirements that follow from the assumption that any action not only relates system states, but also moments in time. The leading principle here is that time advances by performing actions. Time is thus 'realized' by action, which means that the directional structure of actions and states in modal action models should never conflict with the temporal ordering of moments.

> (4) For all a such that $a \in \mathcal{A}$ it holds that $R^{\mathcal{A}}(a) \subseteq R_X$.
>
> (5) For all binary action combinators $\star$ it holds that if $R(\alpha) \subseteq R_G$ and $R(\beta) \subseteq R_G$, then $R(\alpha \star \beta) \subseteq R_G$. And for all unary action combinators $\bullet$ it holds that if $R(\alpha) \subseteq R_G$, then $R(\alpha^{\bullet}) \subseteq R_G$.
>
> (6) For all $(s,t) \in R_G$ it holds that there is some (possibly infinite) series of complex actions $\alpha_1, \alpha_2, \ldots$ such that $(s,t) \in R(\alpha_1) \circ R(\alpha_2) \circ \ldots$ where the α_i are complex actions in the action language of the modal action language under consideration.

Requirement 4 stipulates two things. First of all, that the relation between two system states as execution and resulting state of an atomic action also determines a temporal relation: the execution state chronologically precedes the resulting state. The second implication is that indeed atomicity of actions implies that they relate moments in time for which no intermediate moment is deemed possible. Requirement 5 shows that time is closed under action combinators in the sense that complex actions are temporal whenever their constituent parts are. This means that the definition of R_G (and thus R_X) will have to depend on which action combinators are in the language. So again, we encounter the concept of 'relativization' with respect to action combinators. For instance, if converse is in the action language, time can be advanced by performing a converse action. We discuss this particular example in a separate subsection below. Finally, requirement 6 refers to the observation that we do not consider moments that never occur in any course of action. For reasoning about system behavior this is important. We are not interested to reason about moments that will never occur in a history of system behavior. But, note that by 'never' we do not mean 'not within a finite number of time steps', because we do consider infinities.

In the sections to come, we investigate for various modal action logics defined in chapter 2 how relations R_X and R_G that satisfy the above six requirements can be defined. For the modal action logics of the previous chapter that are most useful for our purposes, that is, the $\cap$-logics and the

$\wr^I$-logics, we will run into problems. We discuss these problems in section 3.4, and propose solutions. But first we discuss three general issues regarding the interpretation of temporal formulas on modal action models.

Determinism and linear temporal logics

An important issue is the relation between linear time and determinism of action. First we shortly explain the difference between linear time and branching time, in terms of properties of R_X. Assume that the relation R_X interprets a temporal operator N, with the informal meaning 'next moment in time'. Now, in linear time temporal logic it holds that $N\varphi \rightarrow \neg N\neg\varphi$, meaning that if in a state for the next moment φ holds, it cannot be that there are other possible states for the next moment where φ does not hold. So in the linear conception of time moments have unique successor moments. Linear time temporal logic is thus appropriate for reasoning domains where the future is (in principle) completely determined by the present[2]. In branching time temporal logic on the other hand, we a priori assume two dimensions of time: a dimension that corresponds with *duration*, and a dimension that corresponds with the different *courses* of time. If N is a branching time temporal operator, $N\varphi \rightarrow \neg N\neg\varphi$ is not a validity. Branching time temporal reasoning is appropriate for reasoning about temporal domains where we a priori assume that the future is not determined by the present. An example is reasoning about autonomous behavior of agents: at any possible future moment an agent has a *choice* about what future to pursue.

It is a natural thought that there should be a strong relation between the notion of linear time and the two notions of determinism we encountered in chapter 2, i.e. (1) determinism at the level of atomic action (characterized by the formula $\langle a\rangle\varphi \rightarrow [a]\varphi$), and (2) determinism at the level of general complex action (characterized by the formula $\langle\alpha\rangle\varphi \rightarrow [\alpha]\varphi$). But that is not the case. The six requirements we formulated do not enforce this. And indeed, they should not. Consider the second notion of determinism for modal action logics. It imposes that from each system state, by performing a specific complex action it is only possible to reach exactly one resulting state. So, the limitation we have is that we cannot perform an action whose outcome is not certain. But no limitations hold for the choice between separate actions. So still, at each moment, several compound deterministic actions are possible,

[2]Note that this does not mean that we assume to always know what the future will be: we may not have enough information about the present or about which moments temporally follow-up which other moments. Model theoretically: we know there is a linear time model reflecting reality, but we do not know which one.

which leads to a *branching* notion of time. It is clear that also for determinism of the first type, and for standard non-deterministic action, we arrive at a branching notion of time.

Future and past

We do not discuss past temporal formulas. In many cases the past is definable in terms of future 'Until' formulas. This leads to the well-known separation theorem's [66, 65].

The only comment we want to make about past modalities is not to confuse them with converse action. We do not make the a priori assumption saying that converse action as interpreted by $R(\alpha^{\leftarrow})$ should somehow be identified with a converse time direction. Conceptually there is a crucial difference between temporal past operators and the action converse: past operators look back in time, converse actions do not. A converse action is like an 'undo': time advances, but the state of affairs returns to a previously encountered state. This difference also follows from the observation that for inverse operations no restrictions apply, while past temporal operators are usually thought to be non-branching. The realms of converse / non-converse action and future / past temporal operation are thus completely distinct. This means that we can have past operators looking back in time over actions and even over converse actions.

Initial states

Many temporal logics define validity on a model with respect to a set of *initial states*. For temporal logics this is often considered more convenient, because we are very used to reason about time from our own perspective on it, that is, from the point we usually refer to by 'now'. The definition of validity on a model as validity in all initial states of the model affects logic entailment, as we discussed in section 2. The entailment $[a]\varphi \models_G [a][a]\varphi$ holds for modal action models. But if validity on a model is defined with respect to initial states, it does not hold.

By defining model validity as validity in all initial states of a model, we do not loose or gain validities: if a formula is valid in all states of models, it is also valid in certain distinguished states of models, and vice versa. However, the allocation of initial states in models paves the way to impose an optional additional constraint on the temporal structure of models, namely, that initial states have no past. For such models we would have formulas like $\neg P\varphi$, where 'P' is a past modality, as general validities. A similar (but not equivalent,

see above) condition for modal action models with initial states would be that from initial states no converse actions can be performed. This can for instance be used to model that compensating actions are not possible initially. For such models we would have $\neg\langle a^{\leftarrow}\rangle\top$ as a general validity. Note that these validities only hold if we restrict accessibility from initial states such that no past or reverse access to other states is possible; by the allocation of certain distinguished 'initial' states as such, no such properties are introduced. The main message of this section is thus that we do not have to worry that the introduction of initial states will affect the validities of a logic. We can thus add them without any difficulty if we want to adapt the notion of entailment. In section 4.3.2 we will use this observation in the analysis of frame properties for sequential actions.

3.2 Combining basic modal action logic with CTL

We show how for basic modal action logic, the six requirements for R_X and R_G on modal action models can be satisfied. First we define $R_X \equiv_{def} R(any^K)$, which by requirement 1 also defines R_G. Recall that in definition 2.5.4 we defined the semantics of any^K as $R(any^K) \equiv_{def} \bigcup_{a\in\mathcal{A}} R(a)$. In section 2.5.3 we used this operator to define the space with respect to which to define an action complement.

The definition $R_X \equiv_{def} R(any^K)$ satisfies all requirements except for requirement 3: time may come to a halt whenever we reach a state where no atomic actions from $\mathcal{A}$ are possible. With requirements 4, 5 and 6 we have explicitly linked the flow of time to the performance of actions, which has as a consequence that time cannot advance if no actions are possible. We discuss two ways out of this problem. The first is to introduce a special action that is only possible in states where no actions from the set $\mathcal{A}$ are possible, and that loops in the current state. For this purpose we can use the test action $([any^K]\bot)?$. This test (definition 2.2.2) checks the possibility to perform any atomic action in the current state. If no atomic actions are possible, the test succeeds. This test is equivalent with the 'dynamic negation' as defined by Van Benthem [17], applied to the union of atomic actions. Since it loops in the current state, the action $([any^K]\bot)?$ has no effect (just like the *skip*). But, we may define that is does advance the time to a next moment in time. Thus, we may define $R_X \equiv_{def} R(([any^K]\bot)?) \cup R(any^K)$. Note that we cannot use the action *skip* (for which proposition 2.2.1 says that it can be syntactically defined according to $skip \equiv_{def} \top?$) for this purpose. If we would include *skip*

as an action that contributes to the next time relation R_X we would get that the current system state is always one of the system states that is possible the next moment in time. Then, we would have to accept that in any system state there is a possible future where the dynamics comes to a standstill while time goes on forever. Also we would get that if something holds for any system state that is next in time, it also holds for the current system state.

Strictly speaking, the above solution is no solution for basic modal action logic, since it requires an extension of the language with dynamic negation or test. The other way out is to assume that it is no problem to violate requirement 3. We simply accept this violation as a strong consequence of our view that time is 'realized' through action. So we stick to the definition $R_X \equiv_{def} R(any^K)$, and accept that time comes to a halt in the subset of states of the state space where no actions are possible. Note that this is not such a strange viewpoint after all. If time leads us to a state where the whole future is frozen due to the absence of any action possibility, the role of time has become redundant[3]. We show that the violation of requirement 3 does not prevent us from defining a version of the well-known branching time temporal logic CTL [47, 57] in terms of R_X. We do not explicitly define the syntax of the combined logic MAL($\emptyset$) / CTL, because this is straightforward: both languages are freely mixed.

Definition 3.2.1 (the syntax of CTL) *Taking 'P' to represent arbitrary elements of a given countable set of proposition symbols $\mathcal{P}$, and 'a' to represent arbitrary elements of a given countable set of action symbols $\mathcal{A}$, a well-formed formula φ of the temporal language $\mathcal{L}_{CTL}$ is defined by:*

$$\varphi, \psi, \ldots \quad := \quad P \mid \neg\varphi \mid \varphi \wedge \psi \mid \langle a \rangle \varphi \mid E(\varphi U \psi) \mid A(\varphi U \psi)$$

where φ, ψ represent arbitrary well-formed formulas. Furthermore, the following abbreviations are applied:

$$\begin{array}{ll} EX\varphi \equiv_{def} E(\bot U \varphi) & AX\varphi \equiv_{def} \neg EX\neg\varphi \\ EF\varphi \equiv_{def} E(\top U \varphi) & AG\varphi \equiv_{def} \neg EF\neg\varphi \\ AF\varphi \equiv_{def} A(\top U \varphi) & EG\varphi \equiv_{def} \neg AF\neg\varphi \end{array}$$

[3]There is an analogy with physics, where it is a common viewpoint that time emerged with the first action (the big bang) and stops with the final action (the great collapse), would it occur.

The CTL-operators have the following informal meanings:

$E(\varphi U \psi)$	:	there is a possible future course of action after which ψ will hold, while φ holds until then
$A(\varphi U \psi)$	:	for all possible future courses of action eventually ψ will hold, while φ holds until then
$EX\varphi$	:	there is an atomic action after which φ will hold
$AX\varphi$	:	after application of any atomic action φ will hold
$EF\varphi$	:	there is a possible future course of action after which φ will hold
$AG\varphi$	:	for all possible future courses of action φ will be preserved
$AF\varphi$	:	for all possible future courses of action eventually φ will hold
$EG\varphi$	:	there is a possible future course of action that preserves φ

The formal semantics of the CTL-operators is stated in terms of the notions of 'maximal state trace' and 'finite sub-trace'.

Definition 3.2.2 (state traces) *For any modal action model $\mathcal{M} = (S, R^{\mathcal{A}}, V^{\mathcal{P}})$, and a definition of the temporal relation $R_X \subseteq S \times S$ in terms of $R^{\mathcal{A}}$, a finite state trace σ from a world s of $\mathcal{M}$ is defined as a series of states $< s_0, s_1, \ldots, s_n >$ such that $s_0 = s$, and $n \geq 1$, and for all i such that $0 \leq i < n$ it holds that $(s_i, s_{i+1}) \in R_X$. We define $\Sigma_+(\mathcal{M}, s)$ to be the set of all finite state traces from s in $\mathcal{M}$. Similarly, countably infinite state traces are infinite series $< s_0, s_1, \ldots >$ such that for all i such that $i \geq 0$ it holds that $(s_i, s_{i+1}) \in R_X$. We define $\Sigma_\omega(\mathcal{M}, s)$ to be the set of all countably infinite state traces from s in $\mathcal{M}$.*

Definition 3.2.3 (finite sub-traces) *For any countably infinite state trace $\sigma = s_0, s_1, \ldots, s_n, \ldots$ and any finite state trace $\sigma' = s_0, s_1, \ldots, s_n, \ldots, s_k$ with $n \geq 1$ and possibly $n = k$, we call the state trace $\sigma'' = s_0, s_1, \ldots, s_n$ a finite sub-trace of σ and σ' respectively. We denote that σ' is a finite sub-trace of any finite or countably infinite state trace σ by $\sigma' \preccurlyeq \sigma$.*

Definition 3.2.4 (maximal state traces) *The set $\Sigma_m(\mathcal{M}, s)$ of all maximal state traces from s in the model $\mathcal{M}$ is defined by $\Sigma_m(\mathcal{M}, s) = \{\sigma \mid$ either $\sigma \in \Sigma_\omega(\mathcal{M}, s)$ or $\sigma \in \Sigma_+(\mathcal{M}, s)$ and there is no $\sigma' \in \Sigma_+(\mathcal{M}, s)$ such that $\sigma \preccurlyeq \sigma'$ and $\sigma' \not\preccurlyeq \sigma\}$.*

We now define the semantics of the CTL-operators.

Definition 3.2.5 (CTL-semantics, validity, logic) *Validity $\mathcal{M}, s \models \varphi$, of a CTL-formula φ in a world s of a modal action model $\mathcal{M}$ is defined as:*

$\mathcal{M}, s \models E(\varphi U \psi)$ *iff* $\exists \sigma \in \Sigma_+(\mathcal{M}, s)$ *with* $\sigma =< s_0, s_1, \ldots, s_n >$ *such that*
(1) $\mathcal{M}, s_n \models \psi$ *and*
(2) $\forall s_i$ *such that* $s_0 < s_i < s_n$ *it holds that* $\mathcal{M}, s_i \models \varphi$

$\mathcal{M}, s \models A(\varphi U \psi)$ *iff* $\forall \sigma \in \Sigma_m(\mathcal{M}, s)$, *there is a* $\sigma' =< s_0, s_1, \ldots, s_n >$ *such that*
(1) $\sigma' \preccurlyeq \sigma$ *and*
(2) $\mathcal{M}, s_n \models \psi$ *and*
(3) $\forall s_i$ *such that* $s_0 < s_i < s_n$ *it holds that* $\mathcal{M}, s_i \models \varphi$

Validity on a model $\mathcal{M}$ is defined as validity in all worlds of the model. If φ is valid on a model $\mathcal{M}$, we say that $\mathcal{M}$ is a model for φ. General validity of a formula φ is defined as validity on all modal action models. The logic CTL is determined by the set of all general validities of $\mathcal{L}_{CTL}$ over the class of standard modal action models.

Note that the notion of 'maximal state trace' is only needed to give semantics to the $A(\varphi U \psi)$ operator. Note also that reflexive temporal operators can be introduced as syntactic extensions: $E_r(\varphi U \psi) \equiv_{def} \psi \vee E(\varphi U \psi)$ and $A_r(\varphi U \psi) \equiv_{def} \psi \vee A(\varphi U \psi)$.

Together with definition 2.1.2, this defines the mixed dynamic / temporal logic MAL($\emptyset$) / CTL. The logic MAL($\emptyset$) / CTL combines reasoning about actions and time and is close to the logic ACTL [141, 140]. However, the interaction between action and time reasoning in this combined language is very limited. It is expressed by the scheme $AX\varphi \rightarrow [a]\varphi$. In the next section we will increase the strength of the time-action relation by going from the basic modal action logic to dynamic logic.

3.3 Temporalizing dynamic logic

There have been numerous attempts to add temporal expressiveness to PDL. Process logic [87] and computation path logic [90] subsume many of these approaches. But process logic and its variants are themselves subsumed by the modal μ^m-calculus as defined by Bradfield and Stirling [25] (we denote their μ-calculus with the superscript '^{m}', because it is based on the rudimentary action language $m, n, \ldots := a_1, a_2, \ldots, a_n \mid -(a_1, a_2, \ldots, a_n)$ for $a_i \in \mathcal{A}$, where the

comma stands for choice and the dash for complement). In section 3.5 we define an extension of Bradfield and Stirling's μ^m-calculus by extending the language of actions m to a language of more complex actions η. In the present section we discuss how for dynamic logic the formulated requirements can be met. With respect to the basic modal action language, c-PDL (converse PDL, see section 2.3) introduces two extra action connectives: iteration and converse. The iterated action does not force us to alter our previous definitions of R_X. Requirement 5 is the only requirement that needs closer attention. It says that the global time relation should be closed under the iteration action. But it already is closed under the iteration action due to requirement 1. But in case of the converse action, requirement 5 does force us to adapt the definitions of R_X. To ensure that the global time relation is closed under taking the converse of actions, we need to include converse atomic actions in the next time relation R_X. This means that if performing α advances the time, now also performing $\alpha^{\leftarrow}$ advances the time. We discussed this in section 3.1. Note that this results in a natural view on converse action. A converse action is like an 'undo' action: an action with opposite effect. Examples include: (1) the undo in text processors, (2) the operation of returning a consumer product that does not meet expectations and asking the money back.

From the above considerations it follows that for c-PDL we have the following two alternatives for the definition of the next time relation on modal action models: $R_X \equiv_{def} R(([any^B]\bot)?) \cup R(any^B)$ and $R_X \equiv_{def} R(any^B)$. As for the basic modal action logic case of the previous section, we can use the first definition if we want to obey requirement 3, and the second if we do not. Note that we now use the any^B action (definition 2.5.4) instead of the any^K action. As explained above, this is because time can be advanced by performing a converse action in c-PDL. For PDL (i.e. without the converse) we have the alternatives $R_X \equiv_{def} R(([any^K]\bot)?) \cup R(any^K)$ and $R_X \equiv_{def} R(any^K)$.

In the same way as we did for the basic modal action language and CTL, we can use the relation R_X to interpret CTL on the same models as (c-)PDL. This defines the combined logic (c-)PDL / CTL. In the next section we investigate the interactions between the action reasoning in the PDL-variant PDL-x and the temporal reasoning in CTL.

Combining PDL and CTL

To enable a comparison between some aspects of the expressive powers of CTL and PDL, we define a slight extension of PDL that we call PDL-x. We introduce a 'general action' x as a syntactic element in the action language of

PDL, and define its relational semantics as $R(x) \equiv_{def} R_X$. For R_X we can use both defined alternatives. With the definition we get directly that the CTL-operator $AX(.)$ and the PDL-x-operator $[x](.)$ have equivalent interpretations. This enables us to investigate the interactions between temporal operators and dynamic operators in the combined logic CTL / PDL-x. The interactions are expressed in the following proposition:

Proposition 3.3.1 *For the combined logic CTL/PDL-x it holds that:*

$$\begin{array}{lcl} AX\varphi & \rightarrow & [a]\varphi \\ AX\varphi & \leftrightarrow & [x]\varphi \\ EX\varphi & \leftrightarrow & \langle x\rangle\varphi \\ E(\varphi U\psi) & \leftrightarrow & \langle x;(\varphi?;x)^*\rangle\psi \end{array}$$

But in PDL-x we cannot express the conditions on models expressed by CTL-formulas of the forms $EG\varphi$ and $AF\varphi$. Consequently we cannot express $A(\varphi U\psi)$ either.

We do not prove this in detail, but discuss the important intuitions behind the proof. Validity of the four formulas comparing PDL-x and CTL-operators is easily verified. To show the inability to express $EG\varphi$ and $AF\varphi$ in PDL-x, we construct two models which can be distinguished by CTL-formulas of this form, but not by any formula of PDL-x. In both PDL-x and CTL, state-validity of formulas is preserved under standard bisimulation. So the models we look for do not bisimulate, are nevertheless indistinguishable for PDL-x, and at the same time *can* be distinguished by CTL. Figure 4 below shows two such models. By default, all states that are not annotated with $-p$ (which stands for $\neg p$) satisfy p. Relations are not annotated with action symbols, since we abstract from actions by means of the general 'atomic' action x.

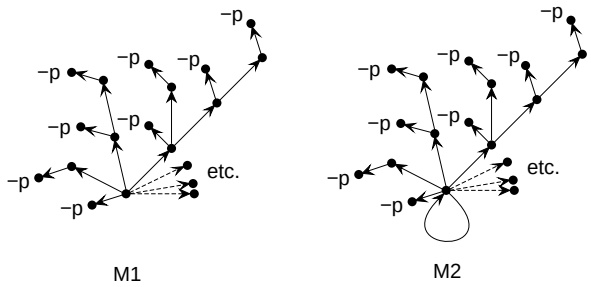


Fig 4. two models distinguishable by CTL, but indistinguishable for PDL-x

The model $\mathcal{M}_1$ has a 'fractal'-like structure: by performing one atomic step from the root into an arbitrary branch, we arrive at a complete copy of the model part that *from the root* is seen to reside at the left hand side of the chosen branch. The strings 'etc.' in the figure mean that the root has infinitely many such branches. The fractal-like structure ensures that for any finitely branching 'slice' of the model as seen from the root, we can always perform a (series of) step(s) from the root and arrive at a state where we see exactly the same finitely branching 'slice'. We make it plausible that this property ensures that PDL-x is unable to distinguish the models. Model $\mathcal{M}_2$ only differs from $\mathcal{M}_1$ in that a loop is added to the root. This loop is responsible for the fact that the models can be distinguished by CTL: $\mathcal{M}_1$ satisfies $AF\neg p$ and does not satisfy EGp, while $\mathcal{M}_2$ satisfies EGp (through the loop) and does not satisfy $AF\neg p$ (because of the loop). Now we have to show that PDL-x does not distinguish the models. If we want to do this thoroughly, we have to show that the roots of the above models bisimulate in the PDL-x ultra-filter extensions (see [19]) of the above models. But to avoid mathematical burden and to concentrate on the intuitions, we will not introduce the notion of an ultra-filter extension. Instead we choose to make the claim plausible by taking a somewhat ad hoc semantic view on PDL-x expressiveness. The only difference between the models is the loop at the root. This loop induces the existence of a countably infinite p-state trace. Now, it is not true that in PDL-x we cannot say anything about the existence of such countably infinite p-state traces. For instance $p \wedge [x^*](p \rightarrow \langle x\rangle p)$ says that there has to be at least one such a state trace. But model $\mathcal{M}_2$ exemplifies that not every model with a countably infinite p-state trace satisfies this formula. So we cannot use the formula to distinguish $\mathcal{M}_1$ from $\mathcal{M}_2$. On the other hand, it is clear that if we look for formulas to say something about the countably infinite state trace, we have to consider formulas with the box operator $[x^*](.)$ as the primary construct to talk about these state traces. Attempts to capture the existence of the state trace through formulas of the form $\langle x^*\rangle\varphi$ have to fail because they only concern properties φ that are at a finite distance from the root. But the problem with formulas of the form $[x^*]\varphi$ is that they do not discriminate between separate state traces, which is exactly the reason why we can 'fool' the property $p \wedge [x^*](p \rightarrow \langle x\rangle p)$. Model $\mathcal{M}_2$ does not satisfy it because at any modal depth there is a branch that starts by going to a p-state, but that eventually encounters a $\neg p$-state. The reader is invited to check that the fractal-like structure ensures that no other PDL-x formulas distinguish the models (note that the example concerns PDL without converse).

In PDL-x we can define a weaker version of the temporal operator $A(\varphi\, U\, \psi)$. The weaker version differs in the sense that it is not necessary that the condition ψ is actually met after finitely many next moments in time. This weaker temporal operator can be defined in PDL-x as $A(\varphi\, U_w\, \psi) \equiv_{def} [(x; \neg\psi?)^+]\varphi$ (where $\alpha^+ \equiv_{def} \alpha; \alpha^*$). Note that also formulas of this form do not distinguish between the above two example models.

The conclusion is that in PDL with the help of the action x, we can express $AG\varphi$ and its negated dual $EF\psi$, but not $AF\varphi$ or $EG\psi$. PDL *couples* the orientations A and E of the branching dimension to respectively the orientations G and F of the duration dimension. Proposition 3.3.1 gives a good impression of the relative expressive powers of PDL and CTL. But several questions remain. For instance, can any PDL-x expressible temporal property by captured by CTL? This seems very likely, and it would mean that the temporal expressiveness of the combined logic PDL-x/CTL is equivalent with that of CTL. Finally, we could study the comparison between PDL-x and CTL on image finite models. In these models no infinite branching is allowed. Can we, for this type of models define an equivalent PDL-x formula for every CTL-formula? We leave these questions for future research.

3.4 Temporalizing logics of concurrent action

So far, we did not encounter any serious problems in meeting the requirements. But this changes when modal action logics are strengthened with intersection of actions (concurrency) or action complement. Even though $\cap$-logics do not define intersection at the level of models, the temporalization of $\cap$-logics is problematic, because they are strong enough to express the inclusion $R(\alpha \cap \beta) \subseteq R(\alpha) \cap R(\beta)$. Therefore, in $\cap$-logics we can easily define a model for which requirement 2 cannot be satisfied. Take for instance a model for which in some state s the formula $\langle a \cap (b; c)\rangle P$ is satisfied. This formula enforces (proposition 2.4.4) that there is a state t that satisfies P and an action graph $\Theta_a \parallel (\Theta_b \cdot \Theta_c)$ such that there is a homomorphism that maps the root of the graph to s and the sink to t. This shows that for instance for the $\cap$-logic MAL$(\cup, \cap)$, it is not correct to define the next time relation as $R_X \equiv_{def} R(any^K)$. This definition violates requirement 2: in the example we have that state t corresponds both to a next moment in time (through atomic action a) and to a second next moment in time (through action b, followed by action c). Note that this is not due to reflexivity of R_X in s or t. The action complement causes similar problems, which follows immediately from the syntactic definability of intersection in terms of complement and choice.

Having established the problem, we consider three directions in which to proceed. The most drastic possibility is to abandon the Ockhamist conception of time. This requires a revision of the way in which we view modal action models as abstract representations of time structures, and an alternative view on time itself. We might for instance switch to the particular non-Ockhamist conception of time where the *same* future time-point can be reached through *separate* alternative courses of time (see e.g. the logic of since and until in [126]) Under this conception of the structure of time the above example gets a completely new interpretation: the system state t corresponds to exactly one time point that is reachable through alternative time-routes, corresponding to, on the one hand action a, and on the other hand the sequence of actions b and c. In such logics the 'until'-formula $\varphi U \psi$ needs not be prefixed by a path (state trace) quantifier in order to interpret it on modal action models. The formula means 'there is some future point where (1) ψ holds, and (2) for which all states of state traces leading to it satisfy φ'. This notion of until is not preserved under bisimulation, which marks the difference with branching time logics such as CTL, CTL*, and the modal μ-calculus, that all do preserve validity under bisimulation.

Several examples of non-Ockhamist temporal logics can be found in the literature [187, 172, 126]. Although there are possible applications for non-Ockhamist conceptions of time, we believe that for system specification purposes this is not the most intuitive perspective to take. But our main reason to reject it is that it is not compatible with the use of relational intersection in modal action structures as the abstract representation for concurrency. Since we have chosen to interpret concurrency by relational intersection, we cannot at the same time see intersection as a representation for alternative courses of time: each concurrent execution of actions concerns a single course of time.

The second possibility is to stick to Ockhamist time and intersection as the formal counterpart of concurrency, but to drop requirement 4, saying that any atomic action in $\mathcal{A}$ contributes to the relation R_X. The idea is to lift the time perspective on atomicity one level up, to the level of action graphs that (1) are not the result of the concatenation of two other action graphs, and (2) have a maximal number of edges, as base for the minimal units of time. The above mentioned graph $\Theta_a \parallel (\Theta_b \cdot \Theta_c)$ is one that satisfies these conditions. As a whole such graphs can be viewed as single coherent actions, that are built up out of concurrently and sequentially composed atomic actions. Within these coherent assemblings of atomic actions, it is not clear how we should deal with time. But we might define that the coherent graphs themselves determine minimal time lapses. This view combines well with modeling concurrency with

intersection. We will not give a formal definition, but the idea is clear; the root and sink of each action graph satisfying the above two conditions, determine an element of the relation R_X. Requirement 4 is the only requirement that cannot be obeyed by this solution. Indeed, the role of atomic actions as minimal units of time is now played by the 'minimal' graphs just mentioned.

The third possibility is to adapt the action language to ensure that the action language is not both closed under, on the one hand sequence or iteration, and on the other hand intersection or complement. The following action syntax is an example.

Definition 3.4.1 (A leveled action syntax) *Taking 'a' to represent arbitrary elements of a given set of atomic action symbols $\mathcal{A}$, a levelled action syntax enabling satisfaction of the six requirements for temporal reasoning over modal actions models is defined as:*

$$\begin{array}{lcl} \eta, \vartheta, \ldots & := & a \mid \eta \cup \vartheta \mid \wr^B \eta \mid \eta^{\leftarrow} \\ \alpha, \beta, \ldots & := & \eta \mid \alpha \cup \beta \mid \alpha; \beta \mid \alpha^* \mid \varphi? \end{array}$$

Only the lower level of 'non-sequential' actions η is closed under complement, converse and (implicitly) intersection. Restriction of converse to the lower action level does not cost us expressiveness: we saw in the proof of theorem 2.3.2 that under the standard interpretation of action connectives, converse can always be pushed down to the atomic action level. Restriction of complement and intersection to the lower action level does cost us expressiveness. But it still enables us to give definitions for operations such as $fail \equiv_{def} \wr^B(\eta \cup \wr^B \eta)$ and $skip \equiv_{def} \top?$. For the definition of the next time relation R_X we get exactly the same alternatives as for PDL: $R_X \equiv_{def} R(([any^B]\bot)?) \cup R(any^B)$ and $R_X \equiv_{def} R(any^B)$. Again, we can use the first definition if we want to obey requirement 3, and the second if we do not. Note that the any^B action is actually definable as $any^B \equiv_{def} \wr^B \alpha \cup \alpha$ in the modal action logics over this levelled action syntax. So, if we choose the second option for the next time relation, i.e. $R_X \equiv_{def} R(any^B)$, we have the next time relation also as an action (any^K) in the action language. For PDL we had to introduce the special general action x for this purpose.

By distinguishing two levels of action, and restricting complement to the lower level, we avoid the problems concerning how to interpret temporal modalities over modal action models for this action language. We show how to define in the modal action logic over this levelled action syntax some mixed time/action operators of the form $A((.)U_w(.))$, where the empty places are

either taken by conditions or actions. The intuitive meaning of for instance $A(\varphi\, U_w\, \eta)$ is 'on all state traces condition φ is preserved at least until an action η occurs'. The subscript w indicates that this is a weak notion of until, for which it is not required that after finitely many next moments in time the action η actually occurs.

Definition 3.4.2 (weak mixed action / time until operators)

$$\begin{array}{lll} A(\varphi\, U_w\, \eta) & \equiv_{def} & [(\wr^B \eta)^+]\varphi \\ A(\eta\, U_w\, \varphi) & \equiv_{def} & [(\eta; \neg\varphi?)^*][\wr^B \eta]\bot \\ A(\eta\, U_w\, \vartheta) & \equiv_{def} & [\wr^B \eta]\bot \wedge [\eta^+][\wr^B(\eta \cup \vartheta)]\bot \end{array}$$

The strong version of the operators are not expressible in the modal action logic over the leveled syntax for the same reason that the strong until operator $A(\psi\, U\, \varphi)$ is not expressible in PDL (see section 3.3). But we can increase the expressiveness of the higher action level in order to define the strong version of the operators. This is the subject of the next section, where for the higher level we take a modal μ-calculus.

3.5 The μ^η-calculus

Informally the modal μ-calculus can be described as basic modal logic (usually referred to as the modal logic K) extended with a minimal and a maximal fixed-point operator. The combination of fixed-point operators and standard modal operators $\Diamond\varphi$ and $\Box\varphi$ enables the expression of a wide range of temporal properties. The μ-calculus is known to subsume, among others, the temporal logics CTL, LTL, CTL*, ACTL, ECTL* [51]. By considering fixed-point operators in combination with the basic modal action operators $\langle a\rangle\varphi$ and $[a]\varphi$, we get what we call the 'μ^a-calculus', that among others, subsumes ACTL and PDL [22]. In this section we go one step further, and combine fixed-point operators with the modal action operators $\langle\eta\rangle\varphi$ and $[\eta]\varphi$, where the actions η are the actions from the lower action level in definition 3.4.1. We call the resulting calculus the 'μ^η-calculus'. All examples of μ-calculi over modal action models in the literature [107, 184, 173, 23, 72, 25] have a less expressive action fragment. The μ-calculus of De Giacomo and Chen [72], that lacks the converse operation, comes closest to the one defined in this section.

Definition 3.5.1 (syntax μ^η-calculus) *Taking 'a' to represent arbitrary elements of a given set of atomic action symbols $\mathcal{A}$, and 'P' to represent arbitrary elements of a given set of proposition symbols $\mathcal{P}$, and 'Z' to represent*

arbitrary elements of a given set of state-set variables[4] $\mathcal{Z}$*, the well-formed formulas* $\varphi, \psi, \ldots$ *of the* μ^η*-calculus are defined by the following BNF:*

$$\begin{array}{lcl} \eta, \vartheta, \ldots & := & a \mid \eta \cup \vartheta \mid \wr^B \eta \mid \eta^{\leftarrow} \\ \varphi, \psi, \ldots & := & P \mid Z \mid \top \mid \bot \mid \neg\varphi \mid \varphi \wedge \psi \mid [\eta]\varphi \mid \mu Z.\, \varphi \end{array}$$

The set of all well-formed formulas is denoted $\mathcal{L}(\mu^\eta)$*.*

The standard syntactic extensions are applied: $\varphi \vee \psi \equiv_{def} \neg(\neg\varphi \wedge \neg\psi)$, $\langle\eta\rangle\varphi \equiv_{def} \neg[\eta]\neg\varphi$, $\varphi \rightarrow \psi \equiv_{def} \neg(\varphi \wedge \neg\psi)$, $\varphi \leftrightarrow \psi \equiv_{def} \neg(\varphi \wedge \neg\psi) \wedge \neg(\psi \wedge \neg\varphi)$. State-set variables from $\mathcal{Z}$ are introduced as sub-formulas of general well-formed formulas φ to be able to view these as a functions $\varphi(Z) : 2^S \rightarrow 2^S$ on sets of states. This in turn makes it possible to define the semantics of $\mu Z.\, \varphi(Z)$ as a minimal fixed-point (a minimal set of states Z such that $\varphi(Z) = Z$) of this function. We apply the usual restriction that a variable Z only appears within the scope of an even number of negations in bounded formulas. The restriction guarantees monotonicity of functions $\varphi(Z) : 2^S \rightarrow 2^S$, which in turn guarantees a solution to the fixed-point equation $\varphi(Z) = Z$. After this short 'look ahead' into the semantics of the modal μ^η-calculus, we now give the formal description of it. After that, we elaborate on how a fixed-point μ^η-calculus formula can best be read.

The semantics is defined by extending the interpretation function $V^{\mathcal{P}} : \mathcal{P} \rightarrow 2^S$ to the interpretation function $V_\varepsilon : \mathcal{L}(\mu^\eta) \rightarrow 2^S$, where ε is an assignment function $\varepsilon : \mathcal{Z} \rightarrow 2^S$ assigning values form the set 2^S to variables form the set $\mathcal{Z}$ which are possibly present in formulas φ of $\mathcal{L}(\mu^\eta)$. We only consider formulas of $\mathcal{L}(\mu^\eta)$ where all variables are bound by either a μ or ν. To determine the assignment ε for the interpretation on a particular model $\mathcal{M}$ of a formula φ containing a variable Z as a sub-formula, we view the interpretation $V_\varepsilon(\varphi)$ as a function $V_\varepsilon : 2^S \rightarrow 2^S$ from state sets ranged over by Z to state sets $V_\varepsilon(\varphi)$. We make this function explicit by writing: $\lambda Z.\, V_\varepsilon(\varphi)$[5]. The value assigned to Z by ε is then obtained as the least or greatest fixed-point (depending on whether Z is bound by a μ or ν respectively) of this function. The role of the identities in definition 3.5.2 is thus twofold: (1) they define $V_\varepsilon(\varphi)$ recursively in the structure of formulas φ, and (2) they define fixed-point equations over functions $\lambda Z.\, V_\varepsilon(\varphi)$, whose minimal or maximal solution contributes to the assignment function ε. If we would allow variables to occur

[4] Usually these are called 'state variables'. But we feel that 'state-set variable' is more appropriate, since the variables range over sets of states.

[5] This is just a more accurate notation for the before used $\phi(Z)$.

free in formulas, the definition would allow us to assign arbitrary values to them, leading to under-determination of the interpretation of formulas.

Definition 3.5.2 (semantics μ^η-calculus) *Given a model $\mathcal{M} = (S, R^{\mathcal{A}}, V^{\mathcal{P}})$, the interpretation $V_\varepsilon(\varphi)$ of a well-formed formula φ on a model $\mathcal{M}$ and an assignment ε of bound state-variables in φ are defined by:*

$$
\begin{array}{lcll}
R(a) & = & R^{\mathcal{A}}(a) & \text{for } a \in \mathcal{A} \\
R(\eta \cup \vartheta) & = & R(\eta) \cup R(\vartheta) & \\
R(\eta^{\leftarrow}) & = & \{(s,t) \mid (t,s) \in R(\eta)\} & \\
R(\wr^B \eta) & = & (\bigcup_{a \in \mathcal{A}} R(a) \cup R(a^{\leftarrow})) \setminus R(\eta) & \\
V_\varepsilon(\top) & = & S & \\
V_\varepsilon(\bot) & = & \emptyset & \\
V_\varepsilon(P) & = & V^{\mathcal{P}}(P) & \text{for } P \in \mathcal{P} \\
V_\varepsilon(Z) & = & \varepsilon(Z) & \\
V_\varepsilon(\varphi \wedge \psi) & = & V_\varepsilon(\varphi) \cap V_\varepsilon(\psi) & \\
V_\varepsilon(\neg\varphi) & = & S \setminus V_\varepsilon(\varphi) & \\
V_\varepsilon([\eta]\varphi) & = & \{s \mid \forall s', (s,s') \in R(\eta) \text{ implies } s' \in V_\varepsilon(\varphi)\} & \\
V_\varepsilon(\mu Z.\varphi) & = & \text{the least fixed-point of the function } \lambda Z.\ V_\varepsilon(\varphi) & \\
V_\varepsilon(\nu Z.\varphi) & = & \text{the greatest fixed-point of the function } \lambda Z.\ V_\varepsilon(\varphi) &
\end{array}
$$

A formula φ is valid in state s of a model $\mathcal{M} = (S, R^{\mathcal{A}}, V^{\mathcal{P}})$ if and only if $s \in V_\varepsilon(\varphi)$ and valid on a model $\mathcal{M} = (S, R^{\mathcal{A}}, V^{\mathcal{P}})$ if and only if $V_\varepsilon(\varphi) = S$. A formula is generally valid if it is valid on all models.

Due to Tarski [175] the definition of the least fixed-point of the monotonic function $\lambda Z.\ V_\varepsilon(\varphi)$ can be written as $\bigcap\{Z \subseteq S \mid \lambda Z.\ V_\varepsilon(\varphi) \subseteq Z\}$. In the definition of the semantics we prefer just to write 'the least fixed-point of the function $\lambda Z.\ V_\varepsilon(\varphi)$', because the characterization due to Tarski is less intuitive.

To understand the intuitive meaning of properties described by μ-calculus formulas, it is useful to consider equivalent infinitary expansions. We have to distinguish between the expansion of ν-formulas and the expansion of μ-formulas. On standard models, satisfaction of a μ-calculus formula $\mu Z.\ \varphi(Z)$ corresponds to satisfaction of the infinitary formula $\varphi(\bot) \vee \varphi(\varphi(\bot)) \vee \ldots$[6]. This means that a formula like $\mu Z.\ \varphi \wedge [a]Z$ can be 'read' as the infinitary formula $(\varphi \wedge [a]\bot) \vee (\varphi \wedge [a](\varphi \wedge [a]\bot)) \vee (\varphi \wedge [a](\varphi \wedge [a](\varphi \wedge [a]\bot))) \vee \ldots$, which helps in understanding its intuitive meaning. Satisfaction of a μ-calculus

[6] This identification assumes $\vee$-continuity of $\varphi(Z)$.

formula $\nu Z.\ \varphi(Z)$ corresponds to satisfaction of the infinitary formula $\varphi(\top) \wedge \varphi(\varphi(\top)) \wedge \ldots$ [7]. And, with the help of the property $\top \leftarrow \varphi(\top) \leftarrow \varphi(\varphi(\top)) \ldots$, that follows from the monotonicity of $\varphi(Z)$, we can see that $\nu Z.\ \varphi(Z)$ is even equivalent with the infinitary formula $\varphi(\varphi(\ldots \varphi(\top) \ldots))$[8]. Then, a formula like $\nu Z.\ \varphi \wedge [a]Z$ can be 'read' as the infinitary formula $\varphi \wedge [a](\varphi \wedge [a](\varphi \wedge \ldots))$.

Both PDL and CTL can be translated to the μ^η-calculus without any difficulty. We first show how to translate converse-free PDL, MAL$(\psi?, \cup, ;, ^*)$ in our notation, to the μ^η-calculus by means of a syntactic translation function T ([22]). This PDL-translation makes no use of the lower action level of actions η.

Proposition 3.5.1 *The recursive function* $T : \mathcal{L}_{MAL}(\psi?, \cup, ;, ^*) \rightarrow \mathcal{L}(\mu^\eta)$ *that determines a map from PDL-formulas to equivalent* μ^η*-calculus formulas, is defined as follows:*

$$
\begin{array}{lcllcl}
T(P) & \equiv & P & T(Z) & \equiv & Z \\
T(\varphi \wedge \psi) & \equiv & T(\varphi) \wedge T(\psi) & T(\neg\varphi) & \equiv & \neg T(\varphi) \\
T([a]\varphi) & \equiv & [a]T(\varphi) & T([\alpha \cup \beta]\varphi) & \equiv & T([\alpha]\varphi) \wedge T([\beta]\varphi) \\
T([\alpha^*]\varphi) & \equiv & \nu Z.\ T(\varphi) \wedge T([\alpha]Z) & T([\alpha;\beta]\varphi) & \equiv & T([\alpha][\beta]\varphi) \\
T([\psi?]\varphi) & \equiv & T(\psi) \rightarrow T(\varphi) & & &
\end{array}
$$

Note that also state-set variables occur in the translation. This is because they may appear as the result of translating formulas like $[a^*]\varphi$. With the help of the infinitary expansion of formulas it is easy to check that the formula $\langle a^* \rangle \varphi$, which translates to the formula $\mu Z.\ \varphi \vee \langle a \rangle Z$ refers to all states in a model where by a finite number of executions of a, we reach a state where φ holds, and that $[a^*]\varphi$, which translates to $\nu Z.\ \varphi \wedge [a]Z$ means that φ has to hold after any number of executions of a.

The following proposition shows how we can define a translation from the CTL-version of section 3.2 to the μ^η-calculus. Note that the translation takes into account that the operators $E(\varphi U \psi)$ and $A(\varphi U \psi)$ defined in section 3.2 are not reflexive.

Proposition 3.5.2 *The recursive function* $T : \mathcal{L}_{CTL} \rightarrow \mathcal{L}(\mu^\eta)$ *that determines a map from CTL-formulas to equivalent* μ^η*-calculus formulas, is defined as follows:*

[7] Assuming $\wedge$-continuity.

[8] Monotonicity also implies $\bot \rightarrow \varphi(\bot) \rightarrow \varphi(\varphi(\bot)) \ldots$, but this is of no use in the simplification of the reading of $\mu Z.\ \varphi$.

$$
\begin{array}{lcl}
T(P) & \equiv & P \\
T(\langle a\rangle\varphi) & \equiv & \langle a\rangle T(\varphi) \\
T(\varphi \wedge \psi) & \equiv & T(\varphi) \wedge T(\psi) \\
T(E(\varphi U \psi)) & \equiv & \langle any^B\rangle \mu Z.\, T(\neg\psi) \rightarrow (T(\varphi) \wedge \langle any^B\rangle Z) \\
T(\neg\varphi) & \equiv & \neg T(\varphi) \\
T(A(\varphi U \psi)) & \equiv & [any^B]\, \mu Z.\, T(\neg\psi) \rightarrow (T(\varphi) \wedge [any^B] Z)
\end{array}
$$

Also more expressive branching time temporal logics, such as CTL* and ECTL* can be translated to μ-calculi [51]. But their translation is less straightforward.

The above two translations give an impression of how logics for dynamics (PDL) and logics for branching time (CTL) are subsumed by the μ^η-calculus. But also, the calculus enables the expression of mixed dynamic / temporal properties. We show how to define the strong versions of the mixed action / time operators of definition 3.5.3.

Definition 3.5.3 (strong mixed action / time until operators)

$$
\begin{array}{lcl}
A(\varphi\, U\, \eta) & \equiv_{def} & [\wr^B \eta]\, \mu Z.\, \varphi \wedge [\wr^B \eta] Z \\
A(\eta\, U\, \varphi) & \equiv_{def} & [\wr^B \eta]\bot \wedge [\eta]\, \mu Z.\, \neg\varphi \rightarrow ([\wr^B \eta]\bot \wedge [\eta] Z) \\
A(\eta\, U\, \vartheta) & \equiv_{def} & [\wr^B \eta]\bot \wedge [\eta]\, \mu Z.\, [\wr^B (\eta \cup \vartheta)]\bot \wedge [\eta] Z
\end{array}
$$

By applying the translation of proposition 3.5.1 it is easy to check that the weak versions of this definition, obtained by replacing the μ's by ν's, are equivalent to the weak mixed action / time until operators of definition 3.5.3. Note that these properties give evidence for the claim of section 2.5.1 that in order to reason about the interactions of time and action, we need the notion of 'action complement'. In particular, the above until operators cannot be defined without the relativized action complement '$\wr^B$'.

Finally, we show that the any^B-construct of the μ^η-calculus can be used to define some important classes of temporal properties. A well-known classification of temporal properties for system specification concerns the distinction between liveness and safety [4]. More refined classifications, including such properties as 'reactivity', 'response', 'persistence'[9], 'obligation'[10], and 'guarantee' have also been given [45]. Traditionally, safety and liveness are

[9] This notion of persistence is distinct from the notion of minimal change we discuss in chapter 4.

[10] This notion of obligation is distinct from the deontic notion of obligation we discuss in chapter 5.

considered linear time temporal notions, which means that they are defined as formal properties of runs (which are equivalent with the 'maximal state traces' of definition 3.2.4). Alpern and Schneider [4] characterize liveness and safety in terms of structural properties of Büchi-automata, which have expressive power equal to that of linear temporal logic. In our setting we assume branching time. We saw that even determinism of action is not sufficient to guarantee linearity of the temporal dimension. Stirling [173] mentions that in such a branching time setting, we have to distinguish between strong and weak notions of liveness and safety. 'Strong' means here that the property holds for all runs through models / trees interpreting a branching time formula, while 'weak' means that there is at least one such run. Strong liveness is thus liveness with respect to all runs through a model. And in addition, we have weak liveness that is defined as liveness of at least one run through a model. Stirling [173] shows how to characterize these properties in terms of the modal μ^m-calculus. Below we adapt his definitions to the μ^η-calculus. Intuitively, liveness means that eventually a 'good' condition is met. Safety means that a good condition is preserved over time. The definitions below abstract from specific good and bad conditions by introduction of the propositions *Good* and *Bad*, for which $\models_G Bad \leftrightarrow \neg Good$. First we show how to express strong and weak liveness, that say that on all traces (respectively some trace) through tree models eventually the condition $\neg Bad$ will hold.

strong liveness wrt state conditions: $\mu Z.\ \text{Bad} \rightarrow (\langle any^B \rangle \top \wedge [any^B] Z)$

weak liveness wrt state conditions: $\mu Z.\ \text{Bad} \rightarrow \langle any^B \rangle Z$

The notions of weak and strong safety, expressing that the condition *Good* is preserved over respectively some trace / all traces, is defined next. If we interchange the role of the special conditions *Good* and *Bad*, weak safety is the negation of strong liveness, and strong safety is the negation of weak liveness.

weak safety wrt state conditions: $\nu Z.\ \text{Good} \wedge ([any^B] \bot \vee \langle any^B \rangle Z)$

strong safety wrt state conditions: $\nu Z.\ \text{Good} \wedge [any^B] Z$

The above expressions for liveness and safety in a branching time setting concern only state conditions. But it is a reasonable wish to be able to express the liveness property that eventually a certain action is performed. Expression of this type of properties requires the introduction of good and bad *actions*. We abstract from particular good and bad actions by introducing the actions

good and *bad* such that always $[\wr^B(bad = \wr^B good)]\bot$. Now we define liveness and safety with respect to action performances as follows:

strong liveness wrt action performance: $\mu Z.\ \langle any^B\rangle\top \wedge [bad]Z$

weak liveness wrt action performance: $\mu Z.\ [good]\bot \rightarrow \langle any^B\rangle Z$

Note that weak liveness with respect to action performance follows from weak liveness with respect to state conditions by substitution of $[good]\bot$ as the bad state condition. But strong liveness with respect to action performance cannot be obtained from strong liveness with respect to state conditions. Note also that to define the difference between the strong and weak notions in this action oriented setting, the relativized action negation $\wr^B$ is again crucial. In section 2.5.1 we used this observation as one of the motivations to develop an intuitive notion of action negation. As for the state-condition case, we may define strong and weak safety properties. And again, these safety properties are obtained as negations (after interchanging the role of the special actions *good* and *bad*) of the liveness properties:

weak safety wrt action performance: $\nu Z.\ \langle any^B\rangle\top \rightarrow \langle good\rangle Z$

strong safety wrt action performance: $\nu Z.\ [bad]\bot \wedge [any^B]Z$

3.6 Conclusions

In this chapter we formulated requirements based on intuitions for the interpretation of temporal operators on modal action models. We showed how to obey these requirements and how to interpret some well-known branching time logics on modal action models. This enabled us to study some properties of the interaction of time and action. Difficulties were encountered in case modal action logics are strong enough to express intersection of action relations, i.e. concurrency. Actions corresponding to minimal time steps may be concurrently composed with (sequences of) actions corresponding to non-minimal time steps, which gives conflicting information for the duration of concurrent actions. Seeing intersection of action relations as confluency of courses of (non-Ockhamist) time was rejected, since intersection is already used to model concurrency. As a possible solution we proposed to restrict concurrency to non-sequential actions, which resulted in a two-layered action syntax, where the lower level involves action complement, intersection (implicitly) and converse, and the higher level the sequential action combinators such as sequence and iteration. We mentioned that in this language we can define

some weak mixed temporal / dynamic modalities. To be able to express the strong versions of these operators we increased the expressiveness of the higher action level, resulting in the μ^{η}-calculus. The μ^{η}-calculus features a number of good properties: (1) it enables the expression of strong mixed action / time operators, (2) due to its high expressive power it subsumes many branching time temporal logics, (3) due to the underlying level of actions η its expressive power also extends to dynamic reasoning, (4) the dynamic reasoning encompasses a relativized action negation and converse. In section 5.3 we exploit this ability of the μ^{η}-calculus to express dynamic properties, to define a deontic modal logic of regular action. The μ^{a}-calculus is decidable and has a complete axiomatization [184]. Also there are several model checkers available for both the μ^{a}-calculus [171, 49] and the μ-calculus [18]. In section 2.5.3 we studied the modal action logic over actions η, and saw that the stronger $\sim$-logic counterpart is decidable and complete. These observations justify the assumption that also the μ^{η}-calculus is decidable and complete.

Chapter 4

Intended modal action models

The relevancy of the subject of this chapter, the definition of intended modal action models for sets of modal action formulas, follows directly from our viewpoint that a specifier may view the system he develops as a set of actions that together are to produce the reactive behavior he has in mind. As a consequence, we have to face some well-known problems from the area of reasoning about action and change: the frame problem, the qualification problem and the ramification problem. But why exactly do we have to face these problems in the context of system specification? We want to use sets of modal action logic formulas for the description of reactive behavior. As discussed in the introduction chapter, logic assumes an open view on dynamics. One of the implications is that an action description not only has to explicitly mention every condition that is changed by the performance of an action, also everything that does not change has to be specified explicitly. If action descriptions are large, and are often extended or updated (in the development stage, a specifier is expected to modify his action descriptions frequently), this task of having to specify everything that does not change can be a cumbersome process. One would like to have ways in which to generate such properties as extensions of the original action description automatically. Definitions of such extensions, which are also called 'completions' are examples of the 'syntactic approach'[1] to the frame problem and other problems of the same type. Syntactic approaches have semantic counterparts. The formulas used for extension, representing frame and other properties, have a semantic counterpart in the concept of intended models for the non-completed action description.

[1] Another example of a syntactic approach is that of 'circumscription', which adds an explicit expression of the frame assumption to action descriptions. The disadvantage is that the expression of the frame assumption requires a higher order language.

The intended models of a non-completed action description are precisely those models that are standard models of the completed action description. Now the semantic approach to the mentioned problems can be described as finding 'intuitive' characterizations of such intended models. Sandewall [163] gives the following description of the approaches to reasoning with action (and change):

> 'If Γ is a set of propositions (specifying action properties), we write $[[\Gamma]]$ for the set of classical models for Γ and $Int([[\Gamma]])$ for the set of intended models[2]. (...) Conventional logic provides definitions and the means of using $[[\Gamma]]$. The research problem is how to obtain $Int([[\Gamma]])$ or the corresponding conclusions (formulae true in all members of $Int([[\Gamma]])$) in terms of operations on formulae in Γ.'

The problem of defining and obtaining $Int([[\Gamma]])$ forms the base objective of the semantic approach. Sandewall's problem description also gives the central problem of this chapter, provided we take into account that Sandewall has the models of first order logic in mind, while we work with modal action models. The central research question for this chapter is thus to establish intuitive intended model semantics for modal action descriptions. This question falls apart in two subquestions. First: what do we base ourselves on with respect to the intuitiveness of such semantics? Checking the semantics against one or two examples can hardly be enough evidence for calling a semantics intuitive. And this is exactly what many (first-order logic-based) work on semantics in reasoning about action and change has been criticized for (see [163, 81]). Although we do test our solutions against some benchmark examples, we have independent reason to call our semantics correct. For the frame problem, the qualification problem, and the mutual exclusion problem, a problem that was not described before in the literature on reasoning about action and change, we obtain intended modal action models as minimal, maximal and maximal elements respectively, in orderings of such models. Evidence for the intuitive correctness of the solutions is provided by the way the orderings for change, qualification and mutual exclusion are obtained as variations on semantic equivalence relations for models of the modal action description language that is used. We believe this strategy may contribute to the solution of the problem of anomalous models in reasoning about action and change. Some aspects of this approach were explored by us in [33, 31, 35, 37].

The second question is how to combine the proposed semantic solutions to the frame, qualification and mutual exclusion problem. In section 4.6 we

[2]We have adapted Sandewall's notation $\Sigma(\Gamma)$ here to $Int([[\Gamma]])$.

show that the two orders in which to apply the two main orderings subsequently result in the same intended models. An interesting question is for which modal action description languages these intended models are unique. If intended models are unique for an action description, the task of (automatically) completing the description in order to reason (through theorem proving) about entailed properties using the standard semantics, can be replaced by the process of explicit generation of the intended model in combination with efficient modal model checking. In section 4.7 we define a modal action language for concurrent actions for which intended models are unique up to semantic equivalence.

4.1 Three related problems for action specification

In the literature [163] many claims can be found with respect to the relationships between the problems with reasoning about action and change. In this section we give our standpoint regarding this issue. We hope this will help the reader to understand (1) the source of the interdependencies between the problems, and (2) our use of terminology.

The frame problem When specifying effects of actions we do not want to involve ourselves in describing explicitly and exhaustively for each individual action what conditions do not change as the result of it.

The ramification problem We do not want to specify explicitly and exhaustively for each individual action what does change as the result of executing it; we want to have the possibility to globally specify that certain conditions (effects) are caused by other conditions (effects).

The qualification problem When specifying under what conditions actions are possible, we do not want to involve ourselves in describing exhaustively what circumstances prevent the possibility (qualification) of an action.

The above problems have a strong parallel in problems with the modeling of common sense reasoning about knowledge of action. An agent may know about some effect of a certain action, and be ignorant about other effects. It is a common reasoning pattern for such agents to assume that effects they do not know about, do not occur (the closed world assumption). And indeed, agents assume that other agents do exactly the same thing, witness such communications as ‘turning this switch will shut down that computer’, which will make

the addressee of the message assume that any other computer in the room is not shut down or started by turning the switch. So, when reasoning about knowledge of action, agents may apply a very similar criterion of minimal change. The epistemic assumption that conditions do not change whenever it is not known that they will change, is called the 'frame assumption'. But we want to emphasize that agents can also adopt completely opposite assumptions when reasoning about action and their effects. It might be the case that an agent knows that a certain action does not change a certain condition, and assumes that other conditions might possibly change (non-deterministically). This assumption is the opposite of a frame assumption. As an example, consider the action *bungeejump*. It is perfectly normal for an agent to (1) know that the action will preserve the condition *Alive*, and (2) assume that certain other conditions of his body might change. Such an assumption should than be called a 'non-frame assumption'. Because of this confusion with *epistemic* assumptions, we will explicitly talk about '*action description* assumptions' when we refer to frame assumptions etc. in the context of action specification. We pursue to incorporate these action description assumptions in the action description semantics. So for the frame problem, the goal is to define an action semantics that includes the description assumption of minimal change, stipulating that changes that are not specified, are not possible. We do so in section 4.3.

As two sub-problem areas of the frame problem, the *representational problem* and the *over-commitment problem* are often mentioned. The representational (or combinatorial) frame problem concerns the question of how to make frame properties explicit as elements of an action language. So the representational frame problem is not a semantic problem, but a problem of the syntactic approach. The over-commitment problem concerns the fact that minimization (of change) policies often are too strict, and rule out possibilities that intuitively should not be disallowed. This problem is thus semantic. The syntactic counterpart of this problem is that extension formulas that are added to an action description to select the intended models, are too strict, and rule out models that are intended. In section 4.3.1 we call such extensions not 'intention-safe'.

The two mentioned problems are usually classified as subproblems of the frame problem exclusively. The reason is probably that the term 'frame problem' is often used as a common name for all problems of this type. We use the term 'frame problem' only in the restricted sense, referring to the problem of *persistency*. It should be clear then that the qualification problem can be thought to have its own representational and over-commitment problem.

A modal action logic-based solution to the ramification problem involves a careful formulation of causation rules in modal action logic. We see the ramification problem as a problem of a nature that is very different from that of the frame or qualification problem. We argue that the ramification problem does not involve a description assumption. The ramification problem only calls for supplementary expressive power of the action language; expressive power that concerns the ability to express and reason correctly with *derived effects*. So, in our view, the ramification problem is a pure representational problem. We discuss this in section 4.5.

As a solution to the qualification problem we pursue an action semantics that incorporates the description assumption stipulating that actions are qualified (alternative terminology: 'are possible', 'are enabled'), unless it is specified that they are not. We call this description assumption 'maximal qualification'. For concurrent actions, we also propose a second, somewhat speculative description assumption: minimal mutual exclusion. We observe that for concurrent action, the standard semantic solution to the qualification problem, that is, to maximize qualifications and specify only the conditions under which actions cannot take place (in the form of necessary preconditions), still requires the specification of an unreasonable amount of (non-)qualification information. For each concurrent composition of atomic actions that appear in an action description, we have to specify explicitly which conditions prevent its possible occurrence. We argue that this problem calls for a new default interpretation in the semantics of action descriptions: one that minimizes or maximizes concurrent qualifications relative to qualifications of concurrent constituent parts. The qualification problem, and its concurrent variant that we call 'the mutual exclusion problem', are treated in section 4.4.

Of course we want a solution that solves all of the above described problems at the same time. But it is not obvious that solutions for the separate problems can be added to arrive at one global solution. For instance, the frame problem and the qualification problem seem to point in opposite directions in the influence they have on action. Minimal change constrains action possibilities by preferring ones that change less. Maximal reachability, 'encourages' action possibilities by maximizing enablings. These are opposite directions. And the frame and ramification problem meet where persistency must be overridden in case of ramification. But at the same time ramification should not be responsible for unintended extra effects, which is to say that the frame assumption also has to be applied to ramifications (derived effects). A similar overlap exists between the qualification problem and the ramification problem: ramifications should also be subject to the possibility of preventing

the qualification of an action [176]. Despite all these dependencies between the three main problems, we will be able to separate them by providing 'orthogonal' solutions. In section 4.6 we show that the solutions we give for the problems can be combined freely.

4.2 From semantic equivalence to orderings

We propose a two step approach to the definition of intended models for modal action languages. The first step is to establish a model-based semantic equivalence notion for the modal action logic. The second step is then to adapt, for both the frame problem and the qualification problem, the notion of semantic equivalence in a minimal way, such that it is turned into a preference ordering of modal action models. In case of the frame problem, the adaptation implements the criterion of minimal change. In case of the qualification problem, the adaptation implements maximal qualification (actions are possible, unless the action description contains explicit information to the contrary). A justification for calling the orderings thus obtained intuitively appropriate is the following. The fact that the resulting change ordering, as a solution to the frame problem, is obtained as a minimal adaptation of the notion of semantic equivalence, makes it possible to control that the ordering compares models *only* on the aspect of minimal change, while all other information contained in models is kept invariant modulo the expressive power of the modal action logic. And the fact that the resulting qualification ordering, as a solution to the qualification problem, is obtained as a minimal adaptation of the notion of semantic equivalence, makes it possible to control that the ordering compares models only on the aspect of action qualification (possibility) in certain states. So by following this approach we can convince ourselves that the orderings compare models exclusively on the relevant aspects (minimal change, maximal qualification, etc.), and that no unintended properties for intended models sneak into the minimization (or maximization) strategy. Of course, this approach cannot guarantee that no unintended models are selected: the addition of the minimal change and maximal qualification principles to semantic equivalence notions can be done in a non-intuitive way. But taking the semantic equivalence notions as a starting point for the definition of the orderings at least gives us a certain level of confidence that no anomalous intended models result.

We assume throughout this chapter that the modal actions logics used for making action descriptions obey the finite model property. The equivalences and orderings are thus relations between models with finite sets of states. In

chapter 2 we mentioned that non-finite models only arise in case of intersection and / or complement in combination with iteration. We will not consider such strong logics in this chapter.

4.3 The frame problem

To single out the modal action models of an action description (AD) that obey a criterion of minimal change we need to compare models on the aspect of minimal change, while all other aspects of models are kept invariant modulo the expressive power of the ADL. In particular, we do not want to compare models on the condition of possibility of action in states. The only other change ordering of modal action models proposed in the literature [62] does minimize action possibilities as a side effect. We discuss in section 4.8 that this forces a specifier to give preconditions that are sufficient for the possibility of an action. This means that a solution to the qualification problem is blocked by the ordering in [62]. Following the strategy described in the previous section, we define orderings based on semantic equivalence notions. For the frame problem we discuss three types of modal action description languages: ADLs where (1) nesting of modalities is syntactically disallowed, and that are not strong enough to express concurrency, (2) ADLs that do allow nesting of modalities, but cannot express concurrency, and (3) ADLs that can express concurrency of action.

4.3.1 Change over non-sequential action

In this section we define change orderings for ADLs whose model validities are preserved under (total surjective) 1-bisimulation, an instance of n-bisimulation, whose general definition can be found in several textbooks (e.g. [19]). This equivalence is adequate for modal action logics that do not allow nesting of modalities, and that are not strong enough to enforce intersection of action relations in modal action models. An example is the basic modal action logic MAL($\emptyset$), with the syntactic restriction that the maximal modal depth of formulas is 1. The following is a model-based semantic equivalence relation for such logics.

Definition 4.3.1 (total surjective 1-bisimulation) *Let* $\mathcal{M}_1 = (S_1, R_1^{\mathcal{A}}, V_1^{\mathcal{P}})$ *and* $\mathcal{M}_2 = (S_2, R_2^{\mathcal{A}}, V_2^{\mathcal{P}})$ *be two models over* $\mathcal{A}$ *and* $\mathcal{P}$*. Then* $\mathcal{M}_1 \simeq_{1b} \mathcal{M}_2$ *(the subscript '1b' for '1-bisimulation') if and only if there is a total surjective relation* $H \subseteq S_1 \times S_2$*, such that for all* s_1 *and* s_2 *for which* $(s_1, s_2) \in H$ *it holds that:*

1. $s_1 \in V_1^{\mathcal{P}}(P)$ *if and only if* $s_2 \in V_2^{\mathcal{P}}(P)$ *for all* P
2. *if there is a* t_1 *such that* $(s_1, t_1) \in R_1^{\mathcal{A}}(a)$*, then there is a* t_2 *such that* $(s_2, t_2) \in R_2^{\mathcal{A}}(a)$ *and* $t_1 \in V_1^{\mathcal{P}}(P)$ *if and only if* $t_2 \in V_2^{\mathcal{P}}(P)$ *for all* P
3. *if there is a* t_2 *such that* $(s_2, t_2) \in R_2^{\mathcal{A}}(a)$*, then there is a* t_1 *such that* $(s_1, t_1) \in R_1^{\mathcal{A}}(a)$ *and* $t_1 \in V_1^{\mathcal{P}}(P)$ *if and only if* $t_2 \in V_2^{\mathcal{P}}(P)$ *for all* P

This notion of semantic equivalence between models differs from (total surjective variant of) plain bisimulation [145, 19] in the sense that it is non-recursive. This corresponds to the absence of nested modalities in the ADLs for which this equivalence notion is appropriate.

Theorem 4.3.1 $\mathcal{M}_1 \simeq_{1b} \mathcal{M}_2$ *if and only if for all formulas* φ *of* $\mathcal{L}_{MAL}(\emptyset)$ *with maximal modal depth 1, it holds that* $\mathcal{M}_1 \leftrightsquigarrow_{\emptyset} \mathcal{M}_2$*, which stands for the property that* φ *is valid on* $\mathcal{M}_1$ *if and only if it is valid on* $\mathcal{M}_2$*.*

Proof

First we prove by induction on the structure of formulas φ that $\mathcal{M}_1 \simeq_{1b} \mathcal{M}_2$ implies $\mathcal{M}_1 \leftrightsquigarrow_{\emptyset} \mathcal{M}_2$. For proposition letters preservation is immediate from condition 1. For the logic connectives $\wedge$ and $\neg$, the property follows from the induction hypotheses. This leaves us with the case $\langle a \rangle \varphi$. From the totality and surjectivity of the relation H it follows that model validity of $\langle a \rangle \varphi$ on $\mathcal{M}_1$ implies that for any state s_1 of $\mathcal{M}_1$ there has to be a witness for a in the form of a state t_1 such that $(s_1, t_1) \in R(a)$ and $\mathcal{M}_1, t_1 \models \varphi$. Now condition 2 says that for any state s_2 of $\mathcal{M}_2$ there is a state t_2 such that $(s_2, t_2) \in R_2(a)$ and $\mathcal{M}_2, t_2 \models \varphi$. But then $\langle a \rangle \varphi$ is valid on $\mathcal{M}_2$. This proofs the 'only if' direction. The 'if' part is proven in the same way using condition 3.

Second we prove that $\mathcal{M}_1 \leftrightsquigarrow_{\emptyset} \mathcal{M}_2$ implies $\mathcal{M}_1 \simeq_{1b} \mathcal{M}_2$. We prove the contraposition. Assume that $\mathcal{M}_1 \not\simeq_{1b} \mathcal{M}_2$. This means that either in $\mathcal{M}_1$ or in $\mathcal{M}_2$ there is a state that is not 1-bisimilar to a state in the other model. Assume this state is s_1 in model $\mathcal{M}_1$. Now we consider the following cases.

(case 1) s_1 is not 1-bisimilar with any state of $\mathcal{M}_2$ because there is no state s_2 in $\mathcal{M}_2$ that meets condition 1 of definition 4.3.1, that is, there is no state having the same valuation of propositional atoms. This means that for any state s_2 in $\mathcal{M}_2$ we can find a literal L, being a formula of the form P or $\neg P$, such that L holds in s_2 but not in s_1. Then the formula $L_1 \vee L_2 \vee \ldots \vee L_n$ holds on $\mathcal{M}_2$ but not on $\mathcal{M}_1$, because it is not satisfied in state s_1 of $\mathcal{M}_1$. Since we assume the finite model property in this chapter, the disjunction is finite.

(case 2) s_1 is not 1-bisimilar with any state of $\mathcal{M}_2$ because among the states in $\mathcal{M}_2$ with the same valuation of propositional atoms there is not one that meets the first part of condition 2 in definition 4.3.1. Let $\{s_2^i \mid 1 \leq i \leq n\}$ be the set of states in $\mathcal{M}_2$ with the same valuation of propositional atoms as for s_1. Then for each s_2^i there is an action a^i and a state t_1^i such that $(s_1, t_1^i) \in R_1(a^i)$ while there is no state t_2^i such that $(s_2^i, t_2^i) \in R_2(a^i)$. Now let ψ be a propositional formula that in model M_2 distinguishes the valuation of atomic propositions of the states s_2^i from the valuation of atomic propositions in other states. Then the formula $\psi \rightarrow (\neg\langle a^1\rangle\top \vee \ldots \vee \neg\langle a^n\rangle\top)$ holds on $\mathcal{M}_2$ but not on $\mathcal{M}_1$, because it is not satisfied in state s_1 of $\mathcal{M}_1$.

(case 3) s_1 is not 1-bisimilar with any state of $\mathcal{M}_2$ because among the states in $\mathcal{M}_2$ satisfying condition 1 and the first part of condition 2 in definition 4.3.1, there is not one satisfying the second part of condition 2. Now let $\{s_2^i \mid 1 \leq i \leq n\}$ be the set of states in $\mathcal{M}_2$ with the same valuation of propositional atoms and the same set of possible actions as for s_1. Then for each s_2^i there is an action a^i, a state t_1^i and a literal L_i such that $(s_1, t_1^i) \in R_1(a^i)$ and L_i holds in t_1^i, while there is no state t_2^i such that $(s_2^i, t_2^i) \in R_2(a^i)$ and L_i holds in t_2^i. Let ψ again be a propositional formula that in model M_2 distinguishes the valuation of atomic propositions of the states s_2^i from the valuation of atomic propositions in other states. Then the formula $\psi \rightarrow (\neg\langle a^1\rangle L_1 \vee \ldots \vee \neg\langle a^n\rangle L_n)$ holds on $\mathcal{M}_2$ but not on $\mathcal{M}_1$, because it is not satisfied in state s_1 of $\mathcal{M}_1$.

(case 4 + 5) Analogous to case 2 and 3, but now for condition 3 of definition 4.3.1.

■

The relation H relates models that are semantically equivalent. Now we add the aspect of minimal change, thereby transforming the relation H into an ordering. But first we need to establish a criterion of change. We define the valuation change for two states to be the set of propositions whose valuation in one state differs from that in the other state.

Definition 4.3.2 (valuation change) *For any interpretation $V^{\mathcal{P}}$ of propositional atoms $\mathcal{P}$, the valuation change $\delta(s_1, s_2)$ with respect to the states s_1 and s_2 is defined as the set of propositions $\{P \mid (s_1 \in V^{\mathcal{P}}(P)$ and $s_2 \notin V^{\mathcal{P}}(P))$ or $(s_1 \notin V^{\mathcal{P}}(P)$ and $s_2 \in V^{\mathcal{P}}(P))\}$.*

Now we can add the change condition to the equivalence notion, thereby turning it into an ordering $\sqsubseteq_{1b}^{ch}$ (the superscript 'ch' for 'change') of modal action models.

Definition 4.3.3 (depth 1 change ordering) *Let $\mathcal{M}_1 = (S_1, R_1^{\mathcal{A}}, V_1^{\mathcal{P}})$ and $\mathcal{M}_2 = (S_2, R_2^{\mathcal{A}}, V_2^{\mathcal{P}})$ be two models over $\mathcal{A}$ and $\mathcal{P}$. Then $\mathcal{M}_1 \sqsubseteq_{1b}^{ch} \mathcal{M}_2$ if and only if there is a total surjective relation $H \subseteq S_1 \times S_2$, such that for all s_1 and s_2 for which $(s_1, s_2) \in H$ it holds that:*

1. *$s_1 \in V_1^{\mathcal{P}}(P)$ if and only if $s_2 \in V_2^{\mathcal{P}}(P)$ for all P*
2. *if there is a t_1 such that $(s_1, t_1) \in R_1^{\mathcal{A}}(a)$, then there is a t_2 such that $(s_2, t_2) \in R_2^{\mathcal{A}}(a)$ and $\delta(s_1, t_1) \subseteq \delta(s_2, t_2)$*
3. *if there is a t_2 such that $(s_2, t_2) \in R_2^{\mathcal{A}}(a)$, then there is a t_1 such that $(s_1, t_1) \in R_1^{\mathcal{A}}(a)$ and $\delta(s_2, t_2) \supseteq \delta(s_1, t_1)$*

Metaphorically speaking, going down in the $\sqsubseteq_{1b}^{ch}$-ordering of models for an action description AD, transitions from states look for 'closer' other states, that is, states for which it only takes a subset of the current changes to reach them. To get an impression of how this ordering compares modal action models, we look at some models of the example formula $\neg A \wedge \neg B \wedge \neg C \rightarrow [k](A \vee B)$.

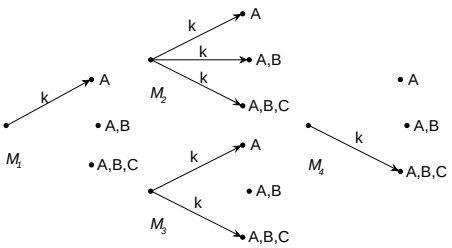


Fig 5. a comparison of some models of $\neg A \wedge \neg B \wedge \neg C \rightarrow [k](A \vee B)$

All models in figure 5 are models of $\neg A \wedge \neg B \wedge \neg C \rightarrow [k](A \vee B)$. Of course there are many other models. Models with other combinations of interpretations of atomic propositions: models where k leads to states where A does not hold, models where k is possible nowhere or in states with other valuations. But the above set of models is the smallest displaying the relevant issues, namely (1) that for atoms that are not specified as an effect (the atom C is an example) the change is minimized to zero, that (2) for atoms that appear disjunctive in an effect (the atoms A and B are examples) change is minimized such that only one of the atoms is changed, and (3) that models equivalent in the $\sqsubseteq_{1b}^{ch}$-ordering are not necessarily semantically equivalent. Let us look closely at the comparison of the above models under the $\sqsubseteq_{1b}^{ch}$-ordering. All models in figure 5 can be mutually compared in the $\sqsubseteq_{1b}^{ch}$-ordering: for

any comparison between two models we can choose the relation H such that states with corresponding positions (and valuations) are related. It is easy to check that by this choice for H, both state valuations and action possibilities are preserved. First we define $\mathcal{M} \sqsubset^{ch}_{1b} \mathcal{M}'$ as $\mathcal{M} \sqsubseteq^{ch}_{1b} \mathcal{M}'$ and $\mathcal{M}' \not\sqsubseteq^{ch}_{1b} \mathcal{M}$, and $\mathcal{M} \equiv^{ch}_{1b} \mathcal{M}'$ as $\mathcal{M} \sqsubseteq^{ch}_{1b} \mathcal{M}'$ and $\mathcal{M}' \sqsubseteq^{ch}_{1b} \mathcal{M}$. Now the following relations hold for the displayed models of the example formula: $\mathcal{M}_1 \sqsubset^{ch}_{1b} \mathcal{M}_2$, $\mathcal{M}_1 \sqsubset^{ch}_{1b} \mathcal{M}_3$, $\mathcal{M}_2 \sqsubset^{ch}_{1b} \mathcal{M}_4$, $\mathcal{M}_3 \sqsubset^{ch}_{1b} \mathcal{M}_4$ and $\mathcal{M}_2 \equiv^{ch}_{1b} \mathcal{M}_3$. So, model $\mathcal{M}_1$ is a minimal model under the $\sqsubseteq^{ch}_{1b}$-ordering, $\mathcal{M}_2$ and $\mathcal{M}_3$ are above $\mathcal{M}_1$ and are mutually indistinguishable in the ordering, and $\mathcal{M}_4$ is the model that forms the top of the ordering. Clearly $\mathcal{M}_1$ is not the only minimal model for $\neg A \wedge \neg B \wedge \neg C \rightarrow [k](A \vee B)$ under the $\sqsubseteq^{ch}_{1b}$-ordering, also the model where action k only makes proposition B true is minimal. And $\mathcal{M}_2$ and $\mathcal{M}_3$ are clearly not the only models that are in between the top model and the minimal models. Figure 5 thus shows only a fragment of the ordering of models.

From this example, we might get the impression that minimization over the $\sqsubseteq^{ch}_{1b}$-ordering always goes hand in hand with a reduction of non-determinism: all ways to perform an action that change more than explicitly described by the action description formula $\neg A \wedge \neg B \wedge \neg C \rightarrow [k](A \vee B)$ are ruled out in the above example. But this reduction of non-determinism is due to the special form of this description formula. We now give an example that enforces non-determinism as the result of minimization over the $\sqsubseteq^{ch}_{1b}$-ordering. The example formula is $\neg A \rightarrow \langle k \rangle A$.

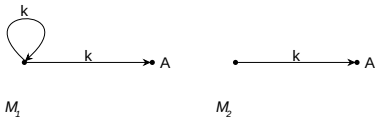


Fig 6. models for which $\mathcal{M}_1 \sqsubset^{ch}_{1b} \mathcal{M}_2$

The action k in the model on the left is 'more non-deterministic' than the action k in the model on the right. But also, the model on the left is below the model on the right in the $\sqsubseteq^{ch}_{1b}$-ordering. And intuitively, this makes sense. A natural language transcription of the formula $\neg A \rightarrow \langle k \rangle A$ is 'it is possible to change the value of A from false to true through execution of k'. Under a minimal change description assumption, such an assertion only makes sense if it is also possible to perform k in such a way that the value of A is not changed, because if not, all ways to perform k would bring about a change from $\neg A$ to A, which means that we should have described (additionally) that $\neg A \rightarrow [k]A$.

The example of figure 5 shows how to specify a lower bound for the changes brought about by an action k. The example of figure 6 shows how to specify an upper bound. So, in the example $\{\neg A \wedge \neg B \wedge \neg C \rightarrow \langle k \rangle (A \wedge B \wedge C), \neg A \wedge \neg B \wedge \neg C \rightarrow [k]A\}$, we get that the minimum (the lower bound) of change for k-actions is one that brings about A, and the maximum (the upper bound) for change brings about A, B and C. So, by using formulas of the forms $\psi \rightarrow [\alpha]\varphi$ and $\psi \rightarrow \langle \alpha \rangle \varphi$, we can specify a lower and an upper bound respectively, for the change brought about by an action α under the condition ψ.

We define $\equiv^{ch}_{1b}$-equivalence classes as sets of models that are $\equiv^{ch}_{1b}$-equivalent. If $\mathcal{M}$ is a model, then $|\mathcal{M}|^{ch}_{1b}$ is its $\equiv^{ch}_{1b}$-equivalence class. The $\sqsubseteq^{ch}_{1b}$-ordering is a pre-order on models: reflexifivity and transitivity are immediate from the definition. We extend this pre-order on models to a partial order over the $\equiv^{ch}_{1b}$-equivalence classes in the standard way. Anti-symmetry of this extended $\sqsubseteq^{ch}_{1b}$-ordering, follows directly from the extension: if $|\mathcal{M}|^{ch}_{1b} \sqsubseteq^{ch}_{1b} |\mathcal{M}'|^{ch}_{1b}$ and $|\mathcal{M}'|^{ch}_{1b} \sqsubseteq^{ch}_{1b} |\mathcal{M}|^{ch}_{1b}$, then $|\mathcal{M}'|^{ch}_{1b} = |\mathcal{M}|^{ch}_{1b}$. Now the following proposition says that semantically equivalent models are indistinguishable in the $\equiv^{ch}_{1b}$-ordering.

Proposition 4.3.2 *For any two models $\mathcal{M}$ and $\mathcal{M}'$ we have that $\mathcal{M} \simeq_{1b} \mathcal{M}'$ implies $\mathcal{M} \equiv^{ch}_{1b} \mathcal{M}'$.*

Proof

From the structural correspondence of the definitions for the $\sqsubseteq^{ch}_{1b}$-ordering and $\simeq_{1b}$-equivalence. If we replace the $\subseteq$ and $\supseteq$ symbols in the second and third condition of definition 4.3.3 by the equal sign '=', we arrive exactly at an alternative formulation of the semantic equivalence notion of definition 4.3.1. This proves the proposition, since absence of change (represented by substitution of the equal sign '=') complies to the conditions $\delta(s_1, t_1) \subseteq \delta(s_2, t_2)$ and $\delta(s_2, t_2) \supseteq \delta(s_1, t_1)$.

■

That the contraposition of this proposition does not hold, follows from the example of figure 5. It holds that $\mathcal{M}_2 \sqsubseteq^{ch}_{1b} \mathcal{M}_3$ and $\mathcal{M}_3 \sqsubseteq^{ch}_{1b} \mathcal{M}_2$. But there is no $\simeq_{1b}$-equivalence, since the 'middle' transition in $\mathcal{M}_2$ has no equivalent in $\mathcal{M}_3$. But in section 4.7 we define a modal action description language for which minimal change models are semantically equivalent if and only if they cannot be distinguished in the change ordering for that action description language. This property also holds for the example action description of figure 5. We do not prove that formally, but note that it is in agreement with the fact that $\mathcal{M}_2$ and $\mathcal{M}_3$ are $\equiv^{ch}_{1b}$-equivalent, but not $\simeq_{1b}$-equivalent, which can only be because they are non-minimal.

Proposition 4.3.2 (and its proof) shows that the change ordering stays very close to the equivalence notion, which is in support of the claim that it minimizes nothing but change. We explained in section 4.2 why this is important. But the property is important also for another reason. It implies that models that are not equivalent in the $\equiv^{ch}_{1b}$-ordering, i.e. models that have different change properties, are not semantically equivalent. And since theorem 4.3.1 states that semantic inequivalence implies modal inequivalence[3], we have that the models can be distinguished by formulas of the ADL. This means that we do not have to increase expressiveness of the ADL to express frame properties: if models differ in the change ordering, there must be formulas in the ADL (that corresponds to the semantic equivalence relation the ordering is based on) that separates them. And with the property that the number of relevant models (modulo 1-bisimilarity) is bounded by the size of action descriptions (which follows from the small model property) we get that we can always define a finite extension to select a particular minimal model. This is also sufficient to guarantee that each action description can be completed by one particular set F that selects exactly all minimal models: for F we can take a disjunction over all conjunctions of extension formulas for separate minimal models. The finiteness of the number of relevant models ensures that F is finite.

The following theorem shows how for action descriptions with formulas of the form $\xi \rightarrow [\alpha]\chi$, extensions look like. Formulas of this form can express (1) conditional effect information for actions: ξ *is a condition under which performing an action* α *brings about* χ, and (2) necessary preconditions $\neg\xi \rightarrow [\alpha]\bot$: *action* α *can only take place if the condition* ξ *is satisfied*. First we introduce the notions 'intention-safe' and 'intention-sufficient' for extensions. An extension is intention-safe whenever the formulas of the extension do not exclude any intended model. An extension is intention-sufficient whenever the formulas in the extension exclude all non-intended models. Note that for any consistent action description the inconsistent extension is always intention-sufficient but never intention-safe.

Definition 4.3.4 *Using the notation* $Int([[AD]])$ *to denote the set of intended models, and* $[[AD]]$ *to denote the set of standard modal action models (definition 1.6.1) of an action description AD, we call a set of extension formulas C* intention-safe *if and only if* $Int([[AD]]) \subseteq [[AD \cup C]]$, *and* intention-sufficient *if and only if* $[[AD \cup C]] \subseteq Int([[AD]])$.

[3]For standard bisimulation, this relation between semantic equivalence and modal equivalence is known as the Hennessy-Milner property [19].

Theorem 4.3.3 *For any finite action description AD containing only action description formulas of the form $\xi \rightarrow [\alpha]\chi$, where formulas ξ and χ do not contain modalities, and actions α are from an action language that does not have sequence $(;)$, iteration $(^*)$, or intersection $(\cap)$, there is a finite set F of extension formulas of the form $\psi \wedge \varphi \rightarrow [\alpha]\varphi$ (φ and ψ do not contain modalities), that is intention-safe and sufficient with respect to the $\sqsubseteq^{ch}_{1b}$-minimal models of AD.*

Proof

Consider an action description AD for which not all standard models $[[AD]]$ are $\sqsubseteq^{ch}_{1b}$-minimal models. Take an arbitrary set F that is intention-safe but not sufficient with respect to $\sqsubseteq^{ch}_{1b}$-minimal models (the empty set suffices). The lack of sufficiency for F implies that there is a model $\mathcal{M}_2$ of $AD \cup F$ that is not a $\sqsubseteq^{ch}_{1b}$-minimal model of AD. But then there is a $\sqsubseteq^{ch}_{1b}$-minimal model $\mathcal{M}_1$ of AD, such that $\mathcal{M}_1 \sqsubset^{ch}_{1b} \mathcal{M}_2$. So $\mathcal{M}_1 \sqsubseteq^{ch}_{1b} \mathcal{M}_2$ and $\mathcal{M}_2 \not\sqsubseteq^{ch}_{1b} \mathcal{M}_1$. Let H be a total surjective relation that is a witness for the assertion $\mathcal{M}_1 \sqsubseteq^{ch}_{1b} \mathcal{M}_2$. Then, from $\mathcal{M}_2 \not\sqsubseteq^{ch}_{1b} \mathcal{M}_1$ it follows that there is a non-zero, finite number of 'strict change differences' between the models $\mathcal{M}_2$ and $\mathcal{M}_1$, which are identified as elements $< s_1, t_1, s_2, t_2, a >$ obeying the properties $(s_1, s_2) \in H$ and $(s_1, t_1) \in R_1^{\mathcal{A}}(a)$ and $(s_2, t_2) \in R_2^{\mathcal{A}}(a)$ and $\delta(s_1, t_1) \subset \delta(s_2, t_2)$. If there were not such strict change differences, the relation H would also be a witness for the assertion $\mathcal{M}_2 \sqsubseteq^{ch}_{1b} \mathcal{M}_1$, which contradicts $\mathcal{M}_2 \not\sqsubseteq^{ch}_{1b} \mathcal{M}_1$. Now we demonstrate that for each strict change difference we can construct a formula of the form $\psi \wedge \varphi \rightarrow [\alpha]\varphi$ that is valid on $\mathcal{M}_1$ and not on $\mathcal{M}_2$. This means that we can extend the extension F with a conjunction of such formulas to exclude the non-minimal change model $\mathcal{M}_2$, while leaving model validity of the minimal model $\mathcal{M}_1$ intact. For each specific strict change difference, we first obtain a set NMC which is defined as the set of atoms $\delta(s_2, t_2) \setminus \delta(s_1, t_1)$. These are the atoms that are responsible for the non-minimal change through action a in state s_2 of the model $\mathcal{M}_2$. Now let φ be the conjunction of literals $L_1 \wedge L_2 \wedge \ldots \wedge L_n$ for which it holds that $\{P \mid P = L_i \text{ or } \neg P = L_i \text{ for } 1 \leq i \leq n\} = NMC$ and $\mathcal{M}_2, s_2 \models L_1 \wedge L_2 \wedge \ldots \wedge L_n$ (and thus $\mathcal{M}_1, s_1 \models L_1 \wedge L_2 \wedge \ldots \wedge L_n$). And let ψ be a propositional formula that distinguishes the valuation of atomic propositions in s_2 (and thus also in the state s_1) from other valuations of atomic propositions in states (on finite models it is always possible to find such a formula). Then the formula $\psi \wedge \varphi \rightarrow [a]\varphi$ is valid on $\mathcal{M}_1$, and not on $\mathcal{M}_2$. Non-validity of the formula on $\mathcal{M}_2$ follows directly from the existence of the strict change difference that implies that the formula is not satisfied in s_2. For validity of the formula on $\mathcal{M}_1$ we have to show three things:

(1) That it is valid in s_1. Clearly we have $\mathcal{M}_1, s_1 \models \psi \wedge \varphi$, because ψ and φ are constructed to be valid in s_1. Then the formula demands that all ways to perform a result in a state where φ. This follows from the minimality of $\mathcal{M}_1$ and the form of the specification formulas $(\xi \rightarrow [\alpha]\chi)$.

(2) That the formula is valid in all states whose valuation of atomic propositions differs from that in s_1. This holds because these states invalidate ψ.

(3) That in states s_1' with a valuation of atomic propositions identical to the valuation for s_1, the formula is valid. For this case, the argument is similar to the argument for case 1.

■

If we drop the restriction for action description formulas, and go to a stronger ADL for which $\simeq_{1b}$-equivalence holds, frame formulas of the form $\psi \wedge \varphi \rightarrow [\alpha]\varphi$ might no longer suffice to select the $\sqsubseteq^{ch}_{1b}$-minimal models. The example of figure 6 showed that minimization can introduce non-determinism. Therefore we also need frame formulas of the form $\psi \wedge \varphi \rightarrow \langle\alpha\rangle\varphi$ for the general case. We conjecture that for any action description language for which $\simeq_{1b}$-equivalence holds, we can construct a conjunction of frame formulas of the forms $\psi \wedge \varphi \rightarrow [\alpha]\varphi$ and $\psi \wedge \varphi \rightarrow \langle\alpha\rangle\varphi$ that is intention-safe and sufficient with respect to $\sqsubseteq^{ch}_{1b}$-minimal models. In such extensions formulas of the form $\psi \wedge \varphi \rightarrow [\alpha]\varphi$ are used to enforce that actions do not change more than intended, and formulas of the form $\psi \wedge \varphi \rightarrow \langle\alpha\rangle\varphi$ are used to ensure that ways to perform an action that change less than other ways to perform it, are actually possible.

If we allow stronger languages for the extensions, some frame properties can be expressed more concisely. In section 2.5.3 we mentioned that in particular the relativized action negation enables an economic expression of frame properties: the formula $\psi \wedge \neg\varphi \rightarrow [\wr^K \alpha]\neg\varphi$ expresses that under the condition ψ, the condition φ can only be brought about by the action α. In section 2.5.1 we explained how we can use this type of formulas to encode Reiter's solution to the frame problem [155] in modal action logics. We do not go into the question whether with formulas of this type together with formulas of the form $\neg\varphi \rightarrow \langle\wr^K \alpha\rangle\neg\varphi$, we can build intention-safe and sufficient extensions.

The minimal elements in the $\sqsubseteq^{ch}_{1b}$-ordering define an 'intended' minimal change semantics of action descriptions. The minimal change semantics of an action description AD is the set of $\sqsubseteq^{ch}_{1b}$-minimal models of AD. Preferential entailment under an ordering $\sqsubseteq^{pref}$ is defined as follows.

Definition 4.3.5 (preferential entailment) *An action description AD preferentially entails ψ, notation $AD \models^{pref} \psi$, if and only if all $\sqsubseteq^{pref}$-intended*

(maximal or minimal) models for AD are also models for ψ.

The Yale Shooting

The Yale shooting problem (YSP) [82] is a problem concerning the correct behavior of fluent values along possible action traces through models for formulas describing action effects at the atomic action level. In modal action logic the relevant information of the action scenario is specified by:

$$\begin{array}{lcl} \neg Loaded & \rightarrow & [load]Loaded \\ Loaded & \rightarrow & [shoot]\neg Loaded \\ Alive \wedge Loaded & \rightarrow & [shoot]\neg Alive \\ \top & \rightarrow & [wait]\top \end{array}$$

The formula $\top \rightarrow [wait]\top$ is valid in any model, and thus superfluous from a logic point of view. It is only added to the description to ensure that the action *wait* is in the description signature. Now the intended, and the unintended conclusion for the YSP are as follows (where we assume the use of a modal action logic with sequence as an action query language):

The intended conclusion:

$$\begin{array}{l} (Alive \wedge \neg Loaded \rightarrow [load](Alive \wedge Loaded)) \wedge \\ (Alive \wedge \neg Loaded \rightarrow [load\,;wait](Alive \wedge Loaded)) \wedge \\ (Alive \wedge \neg Loaded \rightarrow [load\,;wait\,;shoot](\neg Alive \wedge \neg Loaded)) \end{array}$$

The undesired conclusion:

$$\begin{array}{l} (Alive \wedge \neg Loaded \rightarrow [load](Alive \wedge Loaded)) \wedge \\ (Alive \wedge \neg Loaded \rightarrow [load\,;wait](Alive \wedge \neg Loaded)) \wedge \\ (Alive \wedge \neg Loaded \rightarrow [load\,;wait\,;shoot](Alive \wedge \neg Loaded)) \end{array}$$

Before we show that the intended conclusion is entailed, and the unintended conclusion is not entailed by the $\sqsubseteq^{ch}_{1b}$-minimal models, we take a closer look at the YSP action description itself. Note that the description does not contain qualification information. We claim that to show the correctness of the change ordering for the YSP, it is important that indeed we leave qualification information out. Other approaches that claim to solve the YSP [44, 91, 62] do provide qualification information in their YSP action description (using sufficient precondition formulas of the form $\psi \rightarrow \langle a \rangle \top$). Foo et al. explicitly add

$\langle load \rangle \top, \langle wait \rangle \top$, and $\langle shoot \rangle \top$. But since their minimal change semantics also minimizes action possibilities, they have to add these explicit qualifications in order to make their entailment relation behave well. Of course, we could also add qualification information to the above action description. This would not destroy the correct behavior of our preferential entailment relation. But our point is that it should not be necessary to add qualification information in order to make the entailment relation behave well. We discuss the ordering of Foo et al. in section 4.8.

We now take a closer look at the $\sqsubseteq^{ch}_{=1b}$-minimal models of the YSP action description. There are only two fluents in the description, which implies that $\sqsubseteq^{ch}_{=1b}$-minimal models maximally involve four different valuations of propositions in states. In each state of $\sqsubseteq^{ch}_{=1b}$-minimal models, actions *load*, *wait* and *shoot* are either possible or not. The following is a $\sqsubseteq^{ch}_{=1b}$-minimal model for the YSP, where in each state *all* of the actions are possible. Transitions with more than one label are used to abbreviate separate transitions relating the same states, and the fluents *Loaded* and *Alive* are abbreviated to respectively L and A.

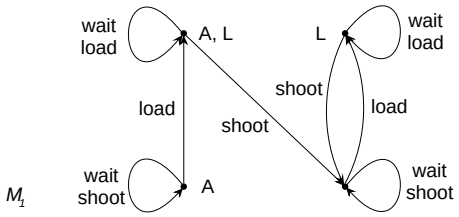


Fig 7. a minimal model for the Yale shooting scenario

This is by far not the only $\sqsubseteq^{ch}_{=1b}$-minimal model. First of all there are infinitely many models that are semantically equivalent (in the sense of definition 4.3.1) with this model. Such models can be constructed by, for instance, 'unraveling' the loops. Because the YSP action description does not contain qualification information, each of the transitions in the above model can be left out to obtain other minimal models. Leaving out transitions in the model results in minimal models that are not comparable to the above one and to each other (two models $\mathcal{M}$ and $\mathcal{M}'$ are not comparable if $\mathcal{M} \not\sqsubseteq^{ch}_{=1b} \mathcal{M}'$ and $\mathcal{M}' \not\sqsubseteq^{ch}_{=1b} \mathcal{M}$), because the models will not be equal in the action sequences that are possible. But the model of figure 7 is 'canonical' in the sense that minimal models are either semantically equivalent with it or with minimal

models that can be formed by leaving out transitions or states. Theorem 4.3.3 states that there has to be an extension that exactly selects all of these $\sqsubseteq^{ch}_{=1b}$-minimal models. A possibility for such an intention-safe and sufficient extension with respect to the $\sqsubseteq^{ch}_{=1b}$-minimal models, is:

$$
\begin{array}{lcl}
\neg Alive & \rightarrow & [wait]\neg Alive \\
Alive & \rightarrow & [wait]Alive \\
\neg Loaded & \rightarrow & [wait]\neg Loaded \\
Loaded & \rightarrow & [wait]Loaded \\
\neg Alive & \rightarrow & [load]\neg Alive \\
Alive & \rightarrow & [load]Alive \\
Loaded & \rightarrow & [load]Loadedn \\
\neg Alive & \rightarrow & [shoot]\neg Alive \\
\neg Loaded & \rightarrow & [shoot]\neg Loaded \\
Loaded \wedge \neg Alive & \rightarrow & [shoot]\neg Alive \\
Alive \wedge \neg Loaded & \rightarrow & [shoot](Alive \wedge \neg Loaded)
\end{array}
$$

So, for the very small action description of the YSP, we already need 11 frame formulas. If we provide frame formulas by hand, it is very easy to overlook certain persistencies. An example is the formula $Loaded \wedge \neg Alive \rightarrow [shoot]\neg Alive$. It says that the action *shoot*, in the context *Loaded* preserves the value of $\neg Alive$. This particular formula also shows that it does not suffice to provide only frame formulas of the form $\varphi \rightarrow [a]\varphi$. Actions can have different effects in different situations (contexts), which means that frame formulas have to be made conditional on action preconditions. Therefore we require the form $\psi \wedge \varphi \rightarrow [a]\varphi$ for frame formulas.

It is clear from inspection of the 'canonical' minimal model of figure 7, and from the extension given above, that the intended conclusion is entailed and the unintended conclusion is not entailed by the $\sqsubseteq^{ch}_{=1b}$-minimal models of the YSP description. Most other solutions to the YSP try to deal with the trade-off between the minimizations concerning subsequent actions. The trade-off is due to the fact that one tries to accomplish minimal change of effects of separate actions through the minimization of a single abnormality predicate. This results in two extensions for the YSP scenario: the intended one in which the change of the wait action is none, and that of the shoot action is fatal, and the unintended one in which the change in the shoot-action is none and that of the wait-action is, surprisingly, non-empty (the gun becomes unloaded). In our setting, this second extension is non-existent, because the minimal change in the wait action is not 'traded' against minimal change in the shoot action.

The solution sketched by Meyer and Doherty [136] behaves well on the YSP for similar reasons.

4.3.2 Change over sequential action

In this section we extend expressiveness of ADLs by allowing the sequence action operation and nested modalities. A typical example of such an ADL is MAL($;,\cup,^*$), in the literature known as PDL (without converse). Therefore we take MAL($;,\cup,^*$) as the central action description language in this section. For other ADLs that allow reasoning about sequential action and exclude reasoning about concurrency (intersection), slight adaptations of the definitions may be required.

The presence of nested modalities complicates the definition of minimal change models considerably. The difficulty is caused by the sequentiality of actions. We will have to define a notion of minimal change that in a way distributes over series of actions when performed one after the other. The central issue is exemplified by the 'stolen car problem' (SCP) [106]. The YSP and the SCP are dual. The YSP is about specifying change locally (the atomic *load*, *shoot* and *wait* actions) and deriving the global change (after the sequence *load* ; *wait* ; *shoot*), the SCP is about specifying a global change over several actions and deriving possible local changes. In the SCP the change concerns the condition that a car becomes stolen. Initially it is not stolen. Three sequentially performed actions lead from the state where the car is not stolen to the state where it is stolen, but no information is provided concerning which of the actions is 'responsible' for the stealing. The criterion of minimal change then imposes that it is either the first, second or third action, while the possibility that the condition changes three times is excluded. The action description information of the SCP-scenario is specified by the modal action logic formula $\neg Stolen \rightarrow [a1;a2;a3]Stolen$. The intended conclusion is that maximally one of the three actions is responsible for the change. The intended conclusion thus excludes that $a1$ changes the condition to $Stolen$, $a2$ changes it back to $\neg Stolen$, and $a3$ changes it again to $Stolen$. An intended conclusion is thus: $\neg Stolen \rightarrow \neg\langle a1\rangle(Stolen \wedge \langle a2\rangle(\neg Stolen \wedge \langle a3\rangle Stolen))$. This conclusion can certainly not be inferred under the standard semantics, since nothing excludes models of $\neg Stolen \rightarrow [a1;a2;a3]Stolen$ where the condition $Stolen$ changes value three times during the execution of $a1;a2;a3$. So, with the SCP, we encounter a completely new aspect of the notion of minimal change, an aspect that cannot be approached by minimizing change locally for non-sequential action only. We have to adopt a more global view on minimal

change.

Minimizing change, under preservation of all other information that is stored in models, is much more difficult if the language we use to talk about the models allows nesting of modalities or contains modalities over sequential actions. The problem is that for sequences of actions, qualification information and change information is not 'encoded' independently in models. For the *non*-sequential case of the previous section we had a clear separation: if action $a1$ is qualified in a state s, and if the change for $a1$ from s has to be minimized, we can simply arrange that from s the action $a1$ reaches a 'closer' state, without altering any of the qualification information in any of the states of the model. But now consider the problem of minimizing the change of a sequence of actions $a1;a2$ from a state s, while (1) the valuation in the start state s is left unchanged, and (2) qualification information (throughout the model) is left unchanged. Let s be the starting state, t the intermediate state, and u and the end state. If the change in action $a1$ is not minimal and we want to minimize it, while leaving the information that the sequential action $a1;a2$ is possible from s unchanged, we cannot avoid to use an alternative 'half-way' state t' and an alternative end state u' for the action $a1;a2$: the half-way state t' is closer to s than the state t, and the end state u' is as far from t' as u from t (the alternative end state is thus required to ensure that the amount of change for the action $a2$ is left unaltered). It is clear that the new situation mixes up the qualification information of the original model: for instance, $a2$ might not be qualified for execution in t'. To steer clear from these dependency problems we work with models where initial conditions (qualification information) and process conditions (change information) are clearly separated: tree models with initial states.

Definition 4.3.6 (tree models and initial states) *A modal action model with initial states $\mathcal{M} = (I, S, R^{\mathcal{A}}, V^{\mathcal{P}})$ is a standard modal action model $\mathcal{M} = (S, R^{\mathcal{A}}, V^{\mathcal{P}})$ for which a set of initial states I such that $I \subseteq S$ is distinguished. A modal action tree-model with initial states $\mathcal{M} = (I, S, T^{\mathcal{A}}, V^{\mathcal{P}})$ is a modal action model with initial states where the interpretation function $T^{\mathcal{A}}$ for atomic actions is such that the actions form a set of trees with the initial states as roots.*

For models with initial states, satisfiability is defined as satisfiability in an initial state, and validity on a model is defined as validity in all initial states. We said in section 3.1 that by the restriction to models with initial states, a modal action logic does not become stronger or weaker. As pointed out in chapter 2 the difference reveals itself in the properties for entailment: for the

global notion of entailment ($\models_G$) we have on standard models that $[a]P \models_G [a][a]P$, and on models with initial states, that $[a]P \not\models_I [a][a]P$. However, for the modal action logics we defined in chapter 2 there is a relation between the two notions of entailment: $AD \models_G \varphi$ if and only if $[any^{S5}](\bigwedge AD) \models_I \varphi$, where $\models_G$ represents standard (global) entailment, and $\models_I$ entailment based on validity in initial states. This property will be useful at the end of this section, where we compare the sequential minimal change solution to the non-sequential one.

Definition 4.3.7 (total surjective bisimulation for initial states) *Let $\mathcal{M}_1 = (I_1, S_1, R_1^{\mathcal{A}}, V_1^{\mathcal{P}})$ and $\mathcal{M}_2 = (I_2, S_2, R_2^{\mathcal{A}}, V_2^{\mathcal{P}})$ be two MA-models with initial states. Then $\mathcal{M}_1 \simeq_b \mathcal{M}_2$ (the subscript 'b' for 'bisimulation') if and only if there is a total surjective relation $I \subseteq I_1 \times I_2$ and a relation $H \subseteq S_1 \times S_2$, such that:*

- $I \subseteq H$

and for all s_1 and s_2 for which $(s_1, s_2) \in H$ it holds that:

1. *$s_1 \in V_1^{\mathcal{P}}(P)$ if and only if $s_2 \in V_2^{\mathcal{P}}(P)$ for all P*
2. *if there is a t_1 such that $(s_1, t_1) \in R_1^{\mathcal{A}}(a)$, then there is a t_2 such that $(s_2, t_2) \in R_2^{\mathcal{A}}(a)$ and $(t_1, t_2) \in H$*
3. *if there is a t_2 such that $(s_2, t_2) \in R_2^{\mathcal{A}}(a)$, then there is a t_1 such that $(s_1, t_1) \in R_1^{\mathcal{A}}(a)$ and $(t_1, t_2) \in H$*

The following theorem is a minor variant on standard results. An important difference with the corresponding theorem for the non-sequential case (theorem 4.3.1) is that the we do not have the direction saying that modal equivalence implies semantic equivalence.

Theorem 4.3.4 *If for two models with initial states $\mathcal{M}_1$ and $\mathcal{M}_2$ it holds that $\mathcal{M}_1 \simeq_b \mathcal{M}_2$, then for all formulas φ of MAL$(;, \cup, ^*)$ it holds that φ is valid on $\mathcal{M}_1$ if and only if it is valid on $\mathcal{M}_2$ (notation $\mathcal{M}_1 \leftrightsquigarrow_{;,\cup,^*} \mathcal{M}_2$).*

Proof

By straightforward generalization of the proof for standard bisimulation (see for instance [19]) to the total surjective, initial state case.

■

The recursion guarantees semantic equivalence of states under nesting of modalities and for modalities over sequential actions. This recursion is inherited by the change ordering we obtain by adding the criterion of change to the equivalence notion. The recursion in the change ordering ensures that minimal change is distributed 'fairly' over sequence of action.

Definition 4.3.8 (sequential change ordering) *Let* $\mathcal{M}_1 = (I_1, S_1, T_1^{\mathcal{A}}, V_1^{\mathcal{P}})$ *and* $\mathcal{M}_2 = (I_2, S_2, T_2^{\mathcal{A}}, V_2^{\mathcal{P}})$ *be two tree models with initial states. Then* $\mathcal{M}_1 \sqsubseteq_b^{ch} \mathcal{M}_2$ *if and only if there is a total surjective relation* $I \subseteq I_1 \times I_2$ *and a relation* $H \subseteq S_1 \times S_2$*, such that it holds that:*

- $I \subseteq H$

and for all s_1 *and* s_2 *such that* $(s_1, s_2) \in I$ *it holds that:*

- $s_1 \in V_1^{\mathcal{P}}(P)$ *if and only if* $s_2 \in V_2^{\mathcal{P}}(P)$ *for all* P

and for all s_1 *and* s_2 *for which* $(s_1, s_2) \in H$ *it holds that:*

1. *if there is a* t_1 *such that* $(s_1, t_1) \in T_1(a)$*, then there is a* t_2 *such that* $(s_2, t_2) \in T_2(a)$ *and* $\delta(s_1, t_1) \subseteq \delta(s_2, t_2)$ *and* $(t_1, t_2) \in H$
2. *if there is a* t_2 *such that* $(s_2, t_2) \in T_2(a)$*, then there is a* t_1 *such that* $(s_1, t_1) \in T_1(a)$ *and* $\delta(s_2, t_2) \supseteq \delta(s_1, t_1)$ *and* $(t_1, t_2) \in H$

Just as for the non-sequential case, we define $\equiv_b^{ch}$-equivalence classes as sets of models that are $\equiv_b^{ch}$-equivalent. If $\mathcal{M}$ is a model, then $|\mathcal{M}|_b^{ch}$ is its $\equiv_b^{ch}$-equivalence class. The $\sqsubseteq_b^{ch}$-ordering is a pre-order on models: reflexifivity and transitivity are immediate from the definition. We extend this pre-order on models to a partial order over the $\equiv_b^{ch}$-equivalence classes in the standard way. Anti-symmetry of this extended $\sqsubseteq_b^{ch}$-ordering, follows directly. As for the non-sequential case, the ordering is very close to the equivalence notion.

Proposition 4.3.5 *For any two models* $\mathcal{M}$ *and* $\mathcal{M}'$ *it holds that* $\mathcal{M} \simeq_b \mathcal{M}'$ *implies* $\mathcal{M} \equiv_b^{ch} \mathcal{M}'$*.*

Proof

As in the proof for proposition 4.3.2, we replace the $\subseteq$ and $\supseteq$ symbols in the forth and back clauses by the equal sign '='. Then the recursion in the definition ensures that the difference in change over any sequential composition of action relations in the two compared models is zero. Together with

the clause that there is no difference in initial valuations either, it follows that by this substitution, similarity in valuations is preserved over sequence of actions. Therefore we arrive exactly at an alternative formulation of the semantic equivalence notion of definition 4.3.7.

■

Since theorem 4.3.4 is only one way, we do not have, as for the non-sequential case, that intention-safe and sufficient extensions matching the minimal models over the ordering of definition 4.3.8, always exist. An extra complication is that for a logic such as MAL($;, \cup, ^*$), tree models need not be finite. An example is the formula $p \wedge [a^*](p \rightarrow \langle a \rangle p)$, that cannot be satisfied on a finite tree model. However, we conjecture that if we would weaken the sequential case by starting with (total surjective) n-bisimilarity as the semantic equivalence notion, the results for the non-sequential case can be generalized. Such an alternative change ordering for the sequential case could not be used for modal action logics that have iteration, since this operation is not 'safe' for n-bisimulations. In the next section we discuss the stolen car scenario, and show that minimization over the $\sqsubseteq_b^{ch}$-ordering distributes minimal change over sequence of action correctly.

The Stolen Car

We argue that the ordering of definition 4.3.8 sorts out the models that display the intended notion of minimal change in case of the stolen car problem (SCP). The type of problems exemplified by the SCP are an important test for the semantics, since they are about the minimal distribution of changes over sequences of actions. An SCP action description should give no information about possibility of actions, since, as argued before, absence of such information is a prerequisite for assessing the correctness of the defined notion of minimal change. With this restriction, the SCP scenario is described by:

$$\neg Stolen \rightarrow [a1; a2; a3]Stolen$$

Consider the following three models for this action description, where the left-most states are initial states.

Fig 8. models for the stolen car scenario

To compare the models, we need to establish the 'comparison' relations I and H. The relation I compares initial states, which for this example, are the left-most states in the models. The relation H then can be chosen such that the first, second, third and fourth states of any pair of separate models are compared. The models $\mathcal{M}_1$ and $\mathcal{M}_2$ are both minimal for the SCP action description under the $\sqsubseteq_b^{ch}$-ordering. That one of the models is not below or above the other in the $\sqsubseteq_b^{ch}$-ordering is seen as follows. Both contain the action sequence $a1, a2, a3$. In $\mathcal{M}_1$ the first two of these actions change less than the corresponding ones in $\mathcal{M}_2$, but the third one changes more. So we have that $\mathcal{M}_1 \not\sqsubseteq_b^{ch} \mathcal{M}_2$ and $\mathcal{M}_2 \not\sqsubseteq_b^{ch} \mathcal{M}_1$, which means that $\mathcal{M}_1$ and $\mathcal{M}_2$ are mutually not ordered. That they are both minimal is seen by comparing them to model $\mathcal{M}_3$. In $\mathcal{M}_3$, after the first action the car is stolen, after the second it has become non-stolen mysteriously, and after the third it is stolen again. Clearly here the change from stolen to not stolen is not minimally distributed over the sequence of three actions. Comparison of $\mathcal{M}_1$ and $\mathcal{M}_3$ gives that both the first and second action change less, while the third changes as much as the corresponding action in $\mathcal{M}_3$. This means that $\mathcal{M}_1 \sqsubseteq_b^{ch} \mathcal{M}_3$ and $\mathcal{M}_3 \not\sqsubseteq_b^{ch} \mathcal{M}_1$, and thus $\mathcal{M}_1$ is below $\mathcal{M}_3$ in the ordering. A similar argumentation can be given for the claim that at $\mathcal{M}_2$ is below $\mathcal{M}_3$ in the ordering. A third minimal model of this type is of course the one where the stealing takes place during the second action.

Of course, models can be much more extensive than the ones of figure 8. In particular we can have infinite tree models. In figure 9 below we represent two such tree models implicitly: the tree models are obtained by unraveling the models from the left most, initial states.

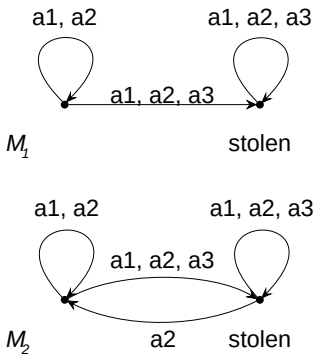


Fig 9. more models for the stolen car scenario

Model $\mathcal{M}_1$ is $\sqsubseteq_b^{ch}$-minimal for the SCP action description. We exemplify this by showing that model $\mathcal{M}_2$ is above $\mathcal{M}_1$ in the $\sqsubseteq_b^{ch}$-ordering. The relation I is again simple: it relates the initial (left most) states of the models. But, it is rather difficult to describe in words what the relation H that is a witness for the claim that $\mathcal{M}_1 \sqsubseteq_b^{ch} \mathcal{M}_2$ looks like. But it should be clear that H can be taken such that all action trees that start with the $a2$-branch that in model $\mathcal{M}_2$ goes back to the state for which $\neg Stolen$, are strictly 'above' the corresponding $a2$-branch in model $\mathcal{M}_1$ that results in the state where $Stolen$.

Back to the Yale shooting

The SCP example suggests that the ordering of definition 4.3.8 performs well for ADLs with nested modalities and modalities for sequential action. And except for the addition of the concept of 'initial states', definition 4.3.8 looks like a neat generalization of the corresponding definition for non-sequential actions. But it turns out that the ordering for nested modalities does not perform well on the example for the non-sequential case: the Yale shooting problem. Let us again consider dynamic logic, or, in our terminology, MAL$(;, \cup, ^*)$ as an ADL. If we want to give an action description of the YSP in this logic, using the ordering of definition 4.3.8 to sort out the minimal change models, we have to account for the fact that we have moved to a system with initial states and a corresponding alternative notion of entailment. To get the same reasoning, we use the earlier mentioned connection: $AD \models_G \varphi$ if and only if $[any](\bigwedge AD) \models_I \varphi$ ($\models_I$ is the entailment notion based on initial states). So, to get the same reasoning as in section 4.3.1, we have to prefix the YSP action description formulas with a modality that ensures that they hold in all states reachable from the initial state. In PDL we do not have the *any* modality.

But for individual action descriptions, it is not hard to simulate it: we prefix the formulas of the YSP scenario description of section 4.3.1 with formulas $[(load \cup wait \cup shoot)^*](.)$. Using the 'simulated' *any*, the YSP scenario can be expressed as:

$$[(load \cup wait \cup shoot)^*](\neg Loaded \rightarrow [load]Loaded)$$
$$[(load \cup wait \cup shoot)^*](Loaded \rightarrow [shoot]\neg Loaded)$$
$$[(load \cup wait \cup shoot)^*](Alive \wedge Loaded \rightarrow [shoot]\neg Alive)$$
$$[(load \cup wait \cup shoot)^*](\top \rightarrow [wait]\top)$$

Now we ask whether the $\sqsubseteq^{ch}_{=b}$-minimal models of this action description correspond to the $\sqsubseteq^{ch}_{=1b}$-minimal models of description in section 4.3.1. It turns out that this is not the case. The following two models provide a counterexample.

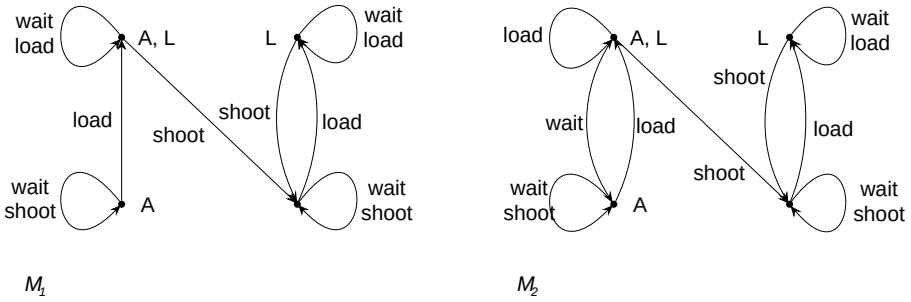


Fig 10. two models for the YSP scenario

The state in the lower-left corner of both models is the initial state, from where the models are thought to be unraveled into infinite trees. Under local minimization, that is, by using the $\sqsubseteq^{ch}_{=1b}$-ordering, it holds that $\mathcal{M}_1 \sqsubset^{ch}_{1b} \mathcal{M}_2$. This follows directly from the observation that the atomic *wait* action from the upper-left state in model $\mathcal{M}_2$ performs a non-minimal change. In the $\sqsubseteq^{ch}_{=b}$-ordering we do not minimize over atomic actions but over sequences of actions from the initial states. From the initial state in model $\mathcal{M}_2$, the sequence $load\,;wait\,;shoot$ performs a change in the *wait* action, while in model $\mathcal{M}_1$, the same sequence performs a change in the *shoot* action. The problem is the ordering of definition 4.3.8 does not prefer one of both sequences: the change in the *wait* action of model $\mathcal{M}_1$ is strictly less than the change in model $\mathcal{M}_2$, but the change in the *shoot* action of model $\mathcal{M}_2$ is strictly less than the change in model $\mathcal{M}_1$. This means that with definition 4.3.8 that enables us to minimize over sequential action, we have brought back the Yale shooting problem.

It is tempting to assume that it should be possible to reconcile the orderings of the previous and present section and arrive at one definition that provides a semantic solution to both the YSP and the SCP. But we argue that the analysis actually shows that both problems cannot be solved at the same time, since they reflect incompatible intended interpretations (action description assumptions). The YSP was solved by the minimization strategy for non-sequential action in the previous section. We may describe the strategy for this non-sequential case as a minimization that is global with respect to states (it is imposed for all states in models) and local with respect to action (it is aimed at minimizing changes over atomic actions). The strategy of the present section is in a sense the opposite: it is local with respect to states (only for initial states), and global with respect to action (minimization of change distributes over sequence of action). Although opposites, both strategies seem exactly right for their task. To solve the YSP, we need minimization that is local to atomic actions: as soon as we allow a 'trade off' between minimizations for separate atomic actions, such as for the global minimization strategy in this section, or as for many standard approaches in the literature (such as minimization with respect to one abnormality predicate), we get back two alternatives to obey minimal change, one of which is the famous counter-intuitive one. But on the other hand, for a solution to the SCP, such a trade off between minimizations for atomic actions is exactly what is needed: if the first action brings about the change, the second and third do not, but if the second brings it about, the first and the third do not, etc. A solution to the SCP thus requires that on the local level of individual atomic actions, change is not minimized 'blindly': minimization depends on the minimization of change for other actions in the sequence. The solutions to the YSP and the SCP thus require incompatible minimization strategies, which means that we cannot combine them.

4.3.3 Change over concurrent action

The principle of minimal change has to be applied with much more caution if concurrent actions are involved. Imposing separate minimal change description assumptions for separate actions that each bring about a change, results in the impossibility to perform the actions concurrently: each action would violate the minimal change criterion for the other actions, since effects of constituent concurrent parts add up in modal action logic. We thus have to apply the minimal change criterion to the concurrent actions themselves, and not to the individual actions that constitute a concurrent action. This follows automatically by applying the approach described in section 4.2. In this section

we take the modal action logic MAL$(\cup, \cap)$ as a paradigmatic example. Definitions for other modal action logics that can express intersection follow by generalizations. In order not to get involved in the issue of sequential concurrent minimal change, we assume that in the ADL modalities are not allowed to occur nested. For such an ADL, we can define a notion of semantic equivalence between models by specialization of definition 2.4.9. But first we give a definition that specializes definition 2.4.8 concerning strict graph relatedness of states in models, for the use of these concepts for the logic MAL$(\cup, \cap)$. For this simpler logic, the notion of action graph reduces to that of a 'step', which is the parallel composition of a set of atomic actions.

Definition 4.3.9 (step relatedness) *For any given model $\mathcal{M} = (S, R^{\mathcal{A}}, V^{\mathcal{P}})$ and set of atomic actions T, we say that states s and t are T related relative to $\mathcal{M}$, notation '$s(T)^{\triangleright}t$', if and only if $T \subseteq \{a \mid (s,t) \in R^{\mathcal{A}}(a)\}$.*

Then the following definition specializes definition 2.4.9 for the case MAL$(\cup, \cap)$, and at the same time generalizes it by lifting the comparison of states to a comparison of models.

Definition 4.3.10 (total surjective step 1-bisimulation) *Let $\mathcal{M}_1 = (S_1, R_1^{\mathcal{A}}, V_1^{\mathcal{P}})$ and $\mathcal{M}_2 = (S_2, R_2^{\mathcal{A}}, V_2^{\mathcal{P}})$ be two MA-models. Then $\mathcal{M}_1 \simeq_{s1b} \mathcal{M}_2$ (the subscript '1sb' for 'step 1-bisimulation') if and only if there is a total surjective relation $H \subseteq S_1 \times S_2$, such that for all s_1 and s_2 for which $(s_1, s_2) \in H$ it holds that:*

1. *$s_1 \in V_1^{\mathcal{P}}(P)$ if and only if $s_2 \in V_2^{\mathcal{P}}(P)$ for all P*
2. *if there is a t_1 such that for some step T it holds that $s_1(T)^{\triangleright}t_1$ relative to $\mathcal{M}_1$, then there is a t_2 such that $s_2(T)^{\triangleright}t_2$ relative to $\mathcal{M}_2$ and $t_1 \in V_1^{\mathcal{P}}(P)$ if and only if $t_2 \in V_2^{\mathcal{P}}(P)$ for all P*
3. *if there is a t_2 such that for some step T it holds that $s_2(T)^{\triangleright}t_2$ relative to $\mathcal{M}_2$, then there is a t_1 such that $s_1(T)^{\triangleright}t_1$ relative to $\mathcal{M}_1$ and $t_2 \in V_2^{\mathcal{P}}(P)$ if and only if $t_1 \in V_1^{\mathcal{P}}(P)$ for all P*

Theorem 4.3.6 *$\mathcal{M}_1 \simeq_{s1b} \mathcal{M}_2$ if and only if for all formulas φ of MAL$(\cup, \cap)$ with maximal modal depth 1, it holds that φ is valid on $\mathcal{M}_1$ if and only if it is valid on $\mathcal{M}_2$ (notation $\mathcal{M}_1 \leftrightsquigarrow_{\cup,\cap} \mathcal{M}_2$).*

Proof

In section 2.4.3 in theorem 2.4.5 we proved for the more general case of action graph bisimulation, that validity is preserved in states. It is straightforward to generalize this result to the total surjective case (while we need only the case for MAL$(\cup, \cap)$ to prove the present theorem). The other direction is similar to the proof for theorem 4.3.1 under replacement of atomic actions a by steps T.

■

The turn from equivalence to the ordering is very much like the turn for the non-sequential non-concurrent case. The only difference is that we now take concurrent steps instead of atomic actions as the unit of action over which change is minimized. This results in the following ordering:

Definition 4.3.11 *Let $\mathcal{M}_1 = (S_1, R_1^{\mathcal{A}}, V_1^{\mathcal{P}})$ and $\mathcal{M}_2 = (S_2, R_2^{\mathcal{A}}, V_2^{\mathcal{P}})$ be two MA-models. Then $\mathcal{M}_1 \sqsubseteq_{s1b}^{ch} \mathcal{M}_2$ if and only if there is a total surjective relation $H \subseteq S_1 \times S_2$, such that for all s_1 and s_2 for which $(s_1, s_2) \in H$ it holds that:*

1. *$s_1 \in V_1^{\mathcal{P}}(P)$ if and only if $s_2 \in V_2^{\mathcal{P}}(P)$ for all P*
2. *if there is a t_1 such that for some step T it holds that $s_1(T)^{\triangleright}t_1$ relative to $\mathcal{M}_1$, then there is a t_2 such that $s_2(T)^{\triangleright}t_2$ relative to $\mathcal{M}_2$ and $\delta(s_1, t_1) \subseteq \delta(s_2, t_2)$*
3. *if there is a t_2 such that for some step T it holds that $s_2(T)^{\triangleright}t_2$ relative to $\mathcal{M}_2$, then there is a t_1 such that $s_1(T)^{\triangleright}t_1$ relative to $\mathcal{M}_1$ and $\delta(s_2, t_2) \supseteq \delta(s_1, t_1)$*

Results for the non-concurrent, non-sequential case of section 4.3.1 are straightforwardly generalized to the concurrent case. We thus have that finite intention-safe and sufficient extension (using formulas of the same logic) exist, and that for description formulas of the form $\xi \rightarrow [\alpha]\chi$, it suffices to use extension formulas of the form $\psi \wedge \varphi \rightarrow [\alpha]\varphi$.

The bowl of soup

To show that the ordering performs well on concurrent action descriptions, we again consider the bowl of soup problem we discussed in sections 2.4.1 and 2.5.1.

$AD = \{$
$[\textit{left-lift}]UpLeft$, $UpLeft \wedge \neg UpRight \rightarrow Spilled$,
$[\textit{right-lift}]UpRight$, $UpRight \wedge \neg UpLeft \rightarrow Spilled$,
$UpLeft \wedge UpRight \rightarrow \neg Spilled$, $\neg UpLeft \wedge \neg UpRight \rightarrow \neg Spilled\}$

To keep the example as simple as possible, we model the ramification of *Spilled* with the help of global constraints. To check whether the minimization of change is performed correctly over concurrent action, we question the minimal change models of this small action description with formulas of the strictly stronger language MAL$(\cup, \wr^K)$. Note that also in the YSP example, we used a stronger language, namely PDL, as an action query language. It is easily verified that the intended conclusions $\neg UpLeft \wedge \neg UpRight \rightarrow [\textit{right-lift} \cap \wr^K \textit{left-lift}]Spilled$ and $\neg UpLeft \wedge \neg UpRight \rightarrow [\textit{left-lift} \cap \wr^K \textit{right-lift}]Spilled$ hold on $\sqsubseteq^{ch}_{s1b}$-minimal models of the action description. In section 2.5.1 we argued that the same intended conclusions can be drawn after extension of the description with the frame formulas $\neg UpLeft \rightarrow [\wr^K \textit{left-lift}]\neg UpLeft$ and $\neg UpRight \rightarrow [\wr^K \textit{right-lift}]\neg UpRight$. Thus, if we allow that the language for extensions of action descriptions is stronger than the language for the descriptions themselves, we can use frame formulas of the form $\psi \wedge \neg\varphi \rightarrow [\wr^K \alpha]\neg\varphi$, instead of frame formulas of the form $\psi \wedge \varphi \rightarrow [\alpha]\varphi$ to express persistency information more economically.

The correct conclusions are drawn while staying within a reasoning context of open actions, because $[\textit{right-lift} \cap \wr^K \textit{left-lift}]Spilled$ says that *Spilled* holds after any concurrent action containing *right-lift* as a concurrent component and excluding *left-lift*. This contradicts the claim by Giordano et al. [75] saying that to model the correct reasoning in this example, it is necessary to introduce closed actions.

4.4 The qualification problem

In section 4.1 we described the qualification problem as the problem how to avoid having to specify sufficient conditions for the possibility of actions. That ψ is a sufficient precondition for an action α is expressed as $\psi \rightarrow \langle\alpha\rangle\top$. But obtaining correct information of this type, i.e. establishing conditions ψ for a certain action domain, is usually very hard for two reasons. First of all, when making action descriptions, it is very difficult to extract all relevant information concerning conditions that prevent the possibility of an action from a description domain. A standard example is from McCarthy [129] who

describes what conditions may prevent the possibility of a boat crossing a river. It may be leak, the weather may be bad, the peddles may be broken, etc. Second, if logic action descriptions are large, the difficulty may also come from 'inside' the description. It may for instance be the case that the specifier of an action domain does not notice that effect information ($\psi \rightarrow [\alpha]\varphi$) and static information (χ), specified as part of an extensive action description AD, conspires ($AD \wedge \varphi \wedge \chi$ is not consistent) to logically imply that α cannot take place under the condition ψ (i.e. $\psi \rightarrow [\alpha]\bot$). In such a situation addition of a formula $\psi \rightarrow \langle\alpha\rangle\top$ results in the global entailment of $\neg\psi$, which might not be intended.

A semantic solution to these problems is provided by the principle of maximal qualification, that takes a default interpretation with respect to qualification information: an action is possible in a certain situation whenever it does not follow from the action description that it is not possible. So, under an intended maximal qualification interpretation of action descriptions, we may only provide necessary preconditions for the possibility of actions. We call this default interpretation 'maximal qualification'. Such a default interpretation is also assumed in the languages $\mathcal{A}$ and $\mathcal{C}$ developed by Gelfond and Lifschitz [69], where it is called 'qualification completeness'.

For the concept of maximal qualification we follow the same approach as for minimal change: we take the equivalence notion for a class of logics and adapt it such that it compares models on the aspect of qualification. For minimal change, we did not want to compare models on action possibilities, which meant that it was necessary to leave the back and forth structure of the equivalence notion in tact, and only focus on the change of valuations in states. For maximal qualification we have the opposite. We want to compare models only on the possibility to do certain actions in states. This means we have to eliminate either the forth or back clause in equivalence notions, such that action possibilities are preserved only in one 'direction'. Furthermore, we do not have to look at valuations in states reached by actions. Qualification is not about the decision whether non-deterministic actions can be performed in a certain way, but about the decision whether non-deterministic actions can be performed at all. If maximization of qualifications would involve maximization of all ways to perform an action, it would as a side effect maximize non-determinism.

4.4.1 Qualification of non-sequential action

For non-sequential action we start with the equivalence notion of definition 4.3.1. Elimination of the 'back' condition and of the comparison of post-conditions, results in the qualification ordering $\sqsubseteq^{ql}_{1b}$ (the superscript 'ql' for 'qualification') for non-sequential actions.

Definition 4.4.1 (qualification ordering) *Let $\mathcal{M}_1 = (S_1, R_1^{\mathcal{A}}, V_1^{\mathcal{P}})$ and $\mathcal{M}_2 = (S_2, R_2^{\mathcal{A}}, V_2^{\mathcal{P}})$ be two models over $\mathcal{A}$ and $\mathcal{P}$. Then $\mathcal{M}_2 \sqsubseteq^{ql}_{1b} \mathcal{M}_1$ if and only if there is a total surjective relation $H \subseteq S_1 \times S_2$, such that for all s_1 and s_2 for which $(s_1, s_2) \in H$ it holds that:*

1. *$s_1 \in V_1^{\mathcal{P}}(P)$ if and only if $s_2 \in V_2^{\mathcal{P}}(P)$ for all P*
2. *if there is a t_2 such that $(s_2, t_2) \in R_2^{\mathcal{A}}(a)$, then there is a t_1 such that $(s_1, t_1) \in R_1^{\mathcal{A}}(a)$*

Analogous to what we defined for the change orderings, we define $\mathcal{M} \equiv^{ql}_{1b} \mathcal{M}'$ as $\mathcal{M} \sqsubseteq^{ql}_{1b} \mathcal{M}'$ and $\mathcal{M}' \sqsubseteq^{ql}_{1b} \mathcal{M}$, and $\equiv^{ql}_{1b}$-equivalence classes as sets of models that are $\equiv^{ql}_{1b}$-equivalent. If $\mathcal{M}$ is a model, then $|\mathcal{M}|^{ql}_{1b}$ is its $\equiv^{ql}_{1b}$-equivalence class. The $\sqsubseteq^{ql}_{1b}$-ordering is a pre-order on models: reflexifivity and transitivity are immediate from the definition. As for the change orderings, we extend this pre-order on models to a partial order over the $\equiv^{ql}_{1b}$-equivalence classes in the standard way. Preferential entailment is defined as in definition 4.3.5. The following proposition says that semantically equivalent models are indistinguishable in the $\equiv^{ql}_{1b}$-ordering.

Proposition 4.4.1 *For any two models $\mathcal{M}$ and $\mathcal{M}'$ it holds that $\mathcal{M} \simeq_{1b} \mathcal{M}'$ implies $\mathcal{M} \equiv^{ql}_{1b} \mathcal{M}'$.*

Proof

Directly from the correspondence between definition 4.4.1 for the $\sqsubseteq^{ql}_{1b}$-ordering and definition 4.3.1 for $\simeq_{1b}$-equivalence. All the conditions in the ordering are also conditions of the semantic equivalence.

■

So, models that are not equivalent in the $\equiv^{ql}_{1b}$-ordering, i.e. models that have different qualification properties, are not semantically equivalent. Then, it follows from proposition 4.3.1 that the models can be distinguished by formulas of MAL($\emptyset$). The following theorem states that we can always define an extension of a finite action description AD that selects exactly the models maximal under the $\sqsubseteq^{ql}_{1b}$-ordering, by using formulas of the form $\psi \rightarrow \langle\alpha\rangle\top$.

Theorem 4.4.2 *For any finite action description AD in an action description language for which $\simeq_{1b}$-equivalence holds, there is a finite set F of extension formulas of the form $\psi \rightarrow \langle\alpha\rangle\top$, that is intention-safe and sufficient with respect to the $\sqsubseteq^{ql}_{1b}$-maximal models of AD.*

Proof

Consider an action description AD for which not all standard models $[[AD]]$ are $\sqsubseteq^{ql}_{1b}$-maximal models. Take an extension set F that is intention-safe but not sufficient with respect to $\sqsubseteq^{ql}_{1b}$-maximal models (the empty set suffices). So there is a model $\mathcal{M}_2$ of $AD \cup F$ that is not a $\sqsubseteq^{ql}_{1b}$-maximal model of AD. But then there is a $\sqsubseteq^{ql}_{1b}$-maximal model $\mathcal{M}_1$ of AD, such that $\mathcal{M}_2 \sqsubset^{ql}_{1b} \mathcal{M}_1$. So $\mathcal{M}_2 \sqsubseteq^{ql}_{1b} \mathcal{M}_1$ and $\mathcal{M}_1 \not\sqsubseteq^{ql}_{1b} \mathcal{M}_2$. Let H be a total surjective relation that is a witness for the assertion $\mathcal{M}_2 \sqsubseteq^{ql}_{1b} \mathcal{M}_1$. Then, from $\mathcal{M}_1 \not\sqsubseteq^{ql}_{1b} \mathcal{M}_2$ it follows that there is a non-zero, finite number of 'strict qualification differences' between the models $\mathcal{M}_2$ and $\mathcal{M}_1$, which are identified as tuples $< s_1, s_2, a >$ obeying the conditions $(s_1, s_2) \in H$ and there is a t_1 such that $(s_1, t_1) \in R^{\mathcal{A}}_1(a)$ while there is no t_2 such that $(s_2, t_2) \in R^{\mathcal{A}}_2(a)$. If there were not such strict qualification differences, the relation H would also be a witness for the assertion $\mathcal{M}_1 \sqsubseteq^{ql}_{1b} \mathcal{M}_2$, which contradicts $\mathcal{M}_1 \not\sqsubseteq^{ql}_{1b} \mathcal{M}_2$. Now we show that for each strict qualification difference we can give a formula of the form $\psi \rightarrow \langle\alpha\rangle\top$ that is valid on $\mathcal{M}_1$ and not on $\mathcal{M}_2$. This means that we can extend the extension F with a conjunction of such formulas to exclude the non-maximal qualification model $\mathcal{M}_2$, while leaving model validity of $\mathcal{M}_1$ intact. Let for any strict qualification difference $< s_1, s_2, a >$, ψ be a propositional formula that distinguishes the valuation of atomic propositions in s_2 (and thus also in the state s_1) from other valuations of atomic propositions in states (on finite models it is always possible to find such a formula). We show that the formula $\psi \rightarrow \langle a\rangle\top$ is valid on $\mathcal{M}_1$, and not on $\mathcal{M}_2$. Non-validity of the formula on $\mathcal{M}_2$ follows directly from the existence of the strict qualification difference that implies that the formula is not satisfied in s_2. For validity of the formula on $\mathcal{M}_1$ we have to show three things:

(1) That it is valid in s_1. This follows directly from the existence of the strict qualification difference.

(2) That the formula is valid in all states whose valuation of atomic propositions differs from that in s_1. This holds because these states invalidate ψ.

(3) That in states s'_1 with a valuation of atomic propositions identical to the valuation for s_1, the formula is valid. This follows from the maximality of $\mathcal{M}_1$ in the $\sqsubseteq^{ql}_{1b}$-ordering.

■

4.4.2 Qualification of sequential action

By extending the expressiveness of ADLs with modalities over sequential action, we arrive at languages for which model validity is preserved under the semantic equivalence notion of definition 4.3.7. For these languages we define the following qualification ordering.

Definition 4.4.2 (sequential qualification ordering) *Let* $\mathcal{M}_1 = (I_1, S_1, T_1^{\mathcal{A}}, V_1^{\mathcal{P}})$ *and* $\mathcal{M}_2 = (I_2, S_2, T_2^{\mathcal{A}}, V_2^{\mathcal{P}})$ *be two tree models with initial states. Then* $\mathcal{M}_2 \sqsubseteq_b^{ql} \mathcal{M}_1$ *if and only if there is a total surjective relation* $I \subseteq S_1 \times S_2$ *and a relation* $H \subseteq S_1 \times S_2$*, such that it holds that:*

- $I \subseteq H$

and for all s_1 *and* s_2 *such that* $(s_1, s_2) \in I$ *it holds that:*

- $s_1 \in V_1^{\mathcal{P}}(P)$ *if and only if* $s_2 \in V_2^{\mathcal{P}}(P)$ *for all* P

and for all s_1 *and* s_2 *for which* $(s_1, s_2) \in H$ *it holds that:*

- *if there is a* t_2 *such that* $(s_2, t_2) \in T_2(a)$*, then there is a* t_1 *such that* $(s_1, t_1) \in T_1(a)$ *and* $(t_1, t_2) \in H$

It is straightforward to generalize proposition 4.4.1 to this sequential case. However, as for the sequential minimal change case, we do not have that intention-safe and sufficient extensions matching the intended (maximal) models always exist.

4.4.3 Qualification of concurrent action and the mutual exclusion problem

As for the frame problem, for the case of concurrent action we restrict ourselves to modal action logics over steps (concurrent atomic actions). The generalization of the theory is straightforward and deserves no further explication. We get the following qualification ordering for steps.

Definition 4.4.3 *Let* $\mathcal{M}_1 = (S_1, R_1^{\mathcal{A}}, V_1^{\mathcal{P}})$ *and* $\mathcal{M}_2 = (S_2, R_2^{\mathcal{A}}, V_2^{\mathcal{P}})$ *be two MA-models. Then* $\mathcal{M}_2 \sqsubseteq_{s1b}^{ql} \mathcal{M}_1$ *if and only if there is a total surjective relation* $H \subseteq S_1 \times S_2$*, such that for all* s_1 *and* s_2 *for which* $(s_1, s_2) \in H$ *it holds that:*

1. $s_1 \in V_1^{\mathcal{P}}(P)$ *if and only if* $s_2 \in V_2^{\mathcal{P}}(P)$ *for all* P

2. *if there is a* t_1 *such that for some step* T *it holds that* $s_2(T)^{\triangleright}t_2$ *relative to* $\mathcal{M}_2$*, then there is a* t_1 *such that* $s_1(T)^{\triangleright}t_1$ *relative to* $\mathcal{M}_1$

Maximization over the above ordering reflects the description assumption that concurrent actions are possible whenever an action description contains no information to the contrary. Definitions, propositions and theorems for the non-concurrent (and non-sequential) case generalize smoothly to the concurrent case, which is why we do not present them. But we do want to discuss an extra complication for the specification of qualification information, that only occurs in the context of concurrent action. We call it 'the mutual exclusion problem' [37]. Our investigation of this problem is still preliminary, and should be considered more as a possible direction for future research, than as a worked out theory.

The mutual exclusion problem

We argue that for concurrent action, the qualification problem, as it is described in section 4.1, becomes considerably harder. This is because the number of possible different concurrent actions grows exponentially with the number of atomic actions used for action description. This means that for concurrent actions the qualification problem is exponentially intensified: for any possible concurrent composition of actions a specifier has to decide on the qualification information when making an action description. The principle of maximal qualification does not help us very much: it stipulates that all of the concurrent possibilities are included by default. A specifier then has to provide formulas of the form $\neg\psi \rightarrow [a_1 \cap a_2 \cap \ldots \cap a_n]\bot$ saying that in states where $\neg\psi$ holds it is not possible to perform the atomic actions $a_1 \cap a_2 \cap \ldots \cap a_n$ concurrently. Following standard terminology from concurrency theory, we refer to such formulas as 'mutual exclusions'. The problem of having to specify all mutual exclusions is thus 'caused' by the principle of maximal qualification in combination with the incorporation of concurrent actions in the description language.

The decision task for the specifier is partially alleviated by an 'add-up' property for qualifications of concurrent actions. In the modal action logics we study, it is the case that if ψ is a necessary precondition for α, then it is also a necessary precondition for $\alpha \cap \beta$. This follows from the semantics defined in section 2.4.2: $\neg\psi \rightarrow [\alpha]\bot \models \neg\psi \rightarrow [\alpha \cap \beta]\bot$ is a valid conclusion. Thus a specifier does not have to decide on all exponentially many qualification possibilities for concurrent actions: if he decides to exclude $a \cap b$, he implicitly decides to exclude $a \cap b \cap c$, etc. Yet, this only gives a partial alleviation of the prob-

lem. The space of actions for which the specifier has to make a qualification decision is still exponential: that he can prune this space quite effectively due to the add-up principle, does not take this problem away. We argue that this problem of having to specify all mutual exclusions is an important problem in itself, that requires a solution independent from the qualification problem.

The mutual exclusion problem When specifying preconditions for actions, we do not want to involve ourselves in describing exhaustively together with which other actions, actions may or may not occur concurrently.

We suggest that preferably one would like to assume a default interpretation with respect to the issue raised by the mutual exclusion problem, a position of the same type as the minimal change principle for the frame problem and the maximal qualification principle for the qualification problem. Just as for these other famous problems, we would like to assume something general about the action description that alleviates the problem of having to specify all mutual exclusion. We propose the following description assumption: if actions are both possible concurrently and in isolation, we assume the concurrent possibility at the cost of the isolated possibilities. That is, we want to minimize mutual exclusion, or equivalently, maximize concurrency, and only specify the exceptions as explicit mutual exclusions. This characterizes the mutual exclusion problem as a problem similar to the frame and qualification problem.

Minimal mutual exclusion thus deals with the problem whether isolated and concurrent execution of an action can both be qualified or that only one of the two is (preferably) qualified, and it decides this choice in favor of concurrent executions. The difference with the standard qualification problem is that minimal mutual exclusion is not a default interpretation about individual actions as such, but about mutual concurrent in- or exclusions of actions. Maximal qualification just maximizes all possible occurrences, either isolated or concurrent.

The problem of wanting to engage in a default interpretation with respect to possibilities of concurrent actions relative to constituent concurrent parts also arises in the context of the semantics of statecharts, a graphical specification formalism for reactive systems [89, 150, 10]. A basic problem in assigning intuitive semantics to statecharts is how to deal with the concurrent executions of parallel components of the statechart. Often a criterion of 'maximal steps' or 'maximal parallelism' is assumed ([100]). Another argument for taking this particular default interpretation is related to the problem of the state space explosion that is encountered if we try to apply model checking techniques to the verification of concurrent system specifications [86]. The state space

explosion is a name for the problem that the number of concurrent actions grows exponentially with the number of atomic actions. Now the state space is most drastically reduced if we adopt a default interpretation of minimal mutual exclusion. It is perceivable that by doing so, we can derive interesting properties of the concurrent system that would otherwise haven taken too much computation time to calculate.

We now show how we might approach the mutual exclusion problem in the same way as we approached the frame and the qualification problem. We only consider the non-sequential case. We adapt the equivalence notion of definition 4.3.1 such that it compares models on the aspect of mutual exclusions of actions. The forth condition is changed such that (concurrent) actions that are qualified in lower models are either (1) qualified in higher models, or are a concurrent composition of actions that are qualified in higher models. The back condition guarantees that steps of actions that are qualified in higher models are either (1) also qualified in lower models, or (2) are concurrently subsumed by steps that are qualified in lower models.

Definition 4.4.4 (mutual exclusion ordering) *Let* $\mathcal{M}_1 = (S_1, R_1^{\mathcal{A}}, V_1^{\mathcal{P}})$ *and* $\mathcal{M}_2 = (S_2, R_2^{\mathcal{A}}, V_2^{\mathcal{P}})$ *be two MA-models. Then* $\mathcal{M}_1 \sqsubseteq^{me}_{s1b} \mathcal{M}_2$ *if and only if there is a total surjective relation* $H \subseteq S_1 \times S_2$*, such that for all* s_1 *and* s_2 *for which* $(s_1, s_2) \in H$ *it holds that:*

1. $s_1 \in V_1^{\mathcal{P}}(P)$ *if and only if* $s_2 \in V_2^{\mathcal{P}}(P)$ *for all* P

2. *if there is a* t_1 *such that for some step* T' *it holds that* $s_1(T')^{\triangleright}t_1$ *relative to* $\mathcal{M}_1$*, then there are states* $t_2^1, t_2^2, \ldots, t_2^n$ *and there are steps* $T^1, T^2, \ldots, T^n$ *such that* $s_2(T^1)^{\triangleright}t_2^1, s_2(T^2)^{\triangleright}t_2^2, \ldots, s_2(T^n)^{\triangleright}t_2^n$ *relative to* $\mathcal{M}_2$*, and* $T' = T^1 \cup T^2 \cup \ldots \cup T^n$

3. *if there is a* t_2 *such that for some step* T *it holds that* $s_2(T)^{\triangleright}t_2$ *relative to* $\mathcal{M}_2$*, then there is a* t_1 *such that for some step* T' *it holds that* $T \subseteq T'$*, and* $s_1(T')^{\triangleright}t_1$ *relative to* $\mathcal{M}_1$

Metaphorically speaking, going down in the mutual exclusion ordering, actions look around for concurrent partners. If an action has found a partner, it sticks to it in the models lower down in the ordering. And concurrent parts of actions in models lower in the ordering always originate from isolated actions in higher models. To give some evidence that this ordering does what it promises to do, we consider two examples where models are compared under the $\sqsubseteq^{me}_{s1b}$-ordering.

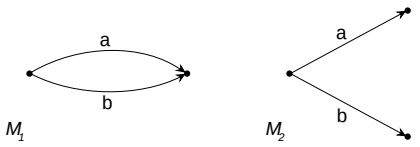


Fig 11. models for which $\mathcal{M}_1 \sqsubset^{me}_{s1b} \mathcal{M}_2$

Assuming that the relation H relates corresponding positions in the models, it follows that $\mathcal{M}_1 \sqsubseteq^{me}_{s1b} \mathcal{M}_2$ and $\mathcal{M}_2 \not\sqsubseteq^{me}_{s1b} \mathcal{M}_1$. Thus, if two different actions performed in isolation can also be performed concurrently, models with the concurrent possibility are preferred, if we minimize over the $\sqsubseteq^{me}_{s1b}$-ordering.

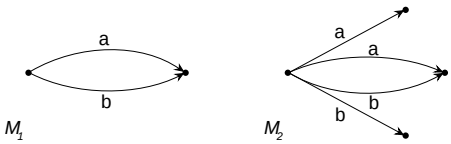


Fig 12. models for which $\mathcal{M}_1 \sqsubset^{me}_{s1b} \mathcal{M}_2$

Again it holds that $\mathcal{M}_1 \sqsubseteq^{me}_{s1b} \mathcal{M}_2$ and $\mathcal{M}_2 \not\sqsubseteq^{me}_{s1b} \mathcal{M}_1$. Thus, situations where actions can only be performed concurrently are favored (by minimization over the $\sqsubseteq^{me}_{s1b}$-ordering) over situations where there is a choice to perform actions concurrently or in isolation. The two examples thus make it clear that whenever an action description specifies compatible postconditions for two actions, in $\sqsubseteq^{me}_{s1b}$-minimal models for the description they are only possible concurrently.

The $\sqsubseteq^{me}_{s1b}$-ordering is reflexive and transitive. In the same way as for the other orderings, we may define $\equiv^{me}_{s1b}$-equivalence classes and a generalized partial $\sqsubseteq^{me}_{s1b}$-ordering over these classes. From the construction of the $\sqsubseteq^{me}_{s1b}$-ordering as an adaptation of $\simeq_{1b}$-equivalence, it is not difficult to conclude that $\mathcal{M} \simeq \mathcal{M}'$ if and only if models $\mathcal{M}$ and $\mathcal{M}'$ belong to the same $\equiv^{me}_{s1b}$-equivalence class. The minimal mutual exclusion semantics of an action description AD is the set of $\sqsubseteq^{me}_{s1b}$-minimal models of AD. Preferential entailment under minimal mutual exclusion is defined analogous to preferential entailment under minimal change, and maximal qualification.

That the mutual exclusion problem is more than just the qualification problem for concurrent action, follows from the observation that we cannot use the extension formulas $\psi \rightarrow \langle\alpha\rangle\top$ to define intention-safe and sufficient extensions. If, for instance, we want to enforce that in the situations where ψ, atomic actions a and b are only possible concurrent but not in isolation,

it is not sufficient to specify $\psi \rightarrow \langle a \cap b \rangle \top$: this formula ensures that action $a \cap b$ is possible, but does not exclude that a and b are not also possible in isolation. But also we cannot add $\psi \rightarrow \langle a \rangle \bot$ and $\psi \rightarrow \langle b \rangle \bot$, since then the action is not longer possible concurrently. What we need, to ensure that the action is only possible concurrently is the following formula $\psi \rightarrow [\wr^K(a \cap b)]\bot$. This formula excludes models where from states where ψ holds only an a, only a b or no a or b are possible: it only allows a and b concurrent. We thus need the extra expressiveness of the relativized action negation as defined in section 2.5.3 to define extensions: the expressiveness of the ADL itself (in this case modal action logics for which validity is preserved under total surjective 1-bisimulation) does not suffice. We conjecture that in general minimization over the mutual exclusion ordering of definition 4.4.4 is intention-safe and sufficient with respect to extensions with formulas of the form $\psi \rightarrow [\wr^K \alpha]\bot$.

4.5 The ramification problem

The ramification problem was described in section 4.1 as the problem of how to specify that certain effect properties are the secondary result of other effects being brought about. We refer to such relations between effects / conditions as *causal* relations. But in the literature on action and change, the term 'causation' is somewhat overloaded with meaning. The term is also used for effects being 'caused' by actions, and for actions being 'caused' (triggered) by conditions becoming true. We use the term 'causation' exclusively to refer to a dependency between (post) conditions (and actions are not conditions, although in the deontic logic literature they have been viewed as such, as we will recall in section 5.4). A main characteristic of causal relations (in the interpretation as a dependency between conditions) is the impossibility to use them contrapositively, that is, if the condition A is being caused by the condition B, then it is not necessarily the case that the condition $\neg B$ is being caused by the condition $\neg A$. This means that we cannot use the material implication of propositional logic to specify causal dependencies between conditions. This is illustrated by the two switches example [115, 177]. In this example we have two switches that both need to be switched on to cause a room to be illuminated[4]. The static domain constraint $Sw_1 \wedge Sw_2 \rightarrow Light$, under its standard propositional logic semantics, does not suffice to model the causal relations in this

[4]Thielscher's [177] example is slightly different, since it concerns the light of a lamp in an electric circuit, which ensures that we have $Sw_1 \wedge Sw_2 \leftrightarrow Light$. We adapt the example in order to emphasize that the problem concerns the contrapositive use of the material implication.

example. If we make Sw_2 to hold, coming from a state where (1) Sw_1 is on, (2) Sw_2 is off and (3) $Light$ is off, we have two equivalent possibilities to obey minimal change. The first one makes $Light$ true, and is associated with the propositional logic inference $\frac{Sw_1, Sw_2, Sw_1 \wedge Sw_2 \rightarrow Light}{Light}$. The second makes Sw_1 false, and is associated with the inference $\frac{Sw_2, \neg Light, \neg Light \rightarrow \neg(Sw_1 \wedge Sw_2)}{\neg Sw_1}$. The first inference is intended. The second inference, that involves contrapositive reasoning with the material implication, is not intended. The static constraint $Sw_1 \wedge Sw_2 \rightarrow Light$ is thus not strong enough to rule out the non-causal contrapositive inference. The formula can be said to lack information (see also [74]) about 'in what way' its integrity is to be preserved.

Foo and Zhang [61] give a solution to this problem that can be applied to the modal action logics we defined. The solution involves an extension of the modal action languages with modalities $[\psi]\varphi$, which are read as 'condition ψ causes condition φ'. The formal interpretation of these modalities involves an extension of modal action models with a function $R : L \rightarrow 2^{S \times S}$, with L the set of well-formed propositional formulas, such that (1) if $\mathcal{M}, s \models \varphi$ then $(s, s) \in R(\varphi)$ for any $\varphi \in L$ and (2) if $\models \psi \rightarrow \varphi$, then $R(\psi) \subseteq R(\varphi)$. This means that the modality $[\psi]\varphi$ is a weakening of the test-modality $[\psi?]\varphi$. The difference is that $[\psi]\varphi$ can be invalid in a state where $\psi \rightarrow \varphi$, while $[\psi?]\varphi$ cannot. The modality $[\psi]\varphi$ does not obey the scheme $[\psi]\varphi \rightarrow [\neg\varphi]\neg\psi$, which shows that it meets the important criterion that causal relations cannot be used contrapositively. Foo et al. show that the causal information in the two switches example can be modeled correctly with the formulas $[Sw_1 \wedge Sw_2]Light$ and $[\neg Sw_1 \vee \neg Sw_2]\neg Light$. These properties imply the static constraint $Sw_1 \wedge Sw_2 \rightarrow Light$. In general it holds that $[\psi]\varphi \rightarrow (\psi \rightarrow \varphi)$. This follows directly from the semantic condition that if $\mathcal{M}, s \models \varphi$ then $(s, s) \in R(\varphi)$ for any $\varphi \in L$. This feature is criticized by Giordano and Schwind [76], who claim that it reintroduces the problems connected to modeling causality with the material implication. We do not agree. Above we argued that indeed, the global constraint $Sw_1 \wedge Sw_2 \rightarrow Light$ is not strong enough. We need a stronger expression that in addition encodes in what way $Sw_1 \wedge Sw_2 \rightarrow Light$ has to be preserved. Since this expression has to be stronger, it should imply the static constraint.

We do not elaborate any further on this solution to the ramification problem. For details, we refer to the papers of Foo and Zhang.

4.6 Orthogonality of the problems

In previous sections we studied the frame problem, the qualification problem and the ramification problem in isolation. Here we investigate how to combine the solutions. We show that for the non-sequential, concurrent case the solutions are orthogonal, that is, we can apply the solutions for the separate problems independently. Orthogonality is important, since it shows that the intended semantics is decomposed into two independent action description assumptions.

Theorem 4.6.1 *For any set of modal action models Ω we denote the set of $\sqsubseteq^{ch}_{s1b}$-minimal models by $MinCh(\Omega)$, and we denote the set of $\sqsubseteq^{ql}_{s1b}$-maximal models by $MaxQl(\Omega)$. Then for any action description AD in a modal action logic for which model validity is preserved under total surjective 1-bisimulation (definition 4.3.1) it holds that $MaxQl(MinCh([[AD]])) = MinCh(MaxQl([[AD]]))$.*

Proof

It is easy to check that definitions 4.3.11 and 4.4.3 ensure that $\mathcal{M}_1 \sqsubseteq^{ch}_{s1b} \mathcal{M}_2$ only if $\mathcal{M}_1 \in |\mathcal{M}_2|^{ql}_{s1b}$ (and thus $\mathcal{M}_2 \in |\mathcal{M}_1|^{ql}_{s1b}$). So the qualification ordering partitions the models of an action description into separate qualification equivalence classes, while the change ordering always minimizes models within these qualification equivalence classes. Thus, it does not matter which order we assume. If we first minimize the change of the models for AD, due to the above property we always minimize within each qualification equivalence class. Subsequent maximization then selects only those minimal change equivalence classes that are part of a maximal qualification equivalence class. If we start with maximizing qualification, we obtain the maximal qualification equivalence classes first. Then minimization of change for the maximal qualification equivalence classes results in the same minimal change equivalence classes within these maximal qualification equivalence classes.

■

Clearly, also for the combined ordering we have that semantic equivalence implies equivalence for the orderings. The other direction does not hold. Even the models in the classes that are minimal in the change ordering and maximal in the qualification ordering, are not semantically equivalent. A simple counterexample is the set we gave in section 4.3.1: $\{\neg A \wedge \neg B \wedge \neg C \rightarrow \langle k \rangle (A \wedge B \wedge C), \neg A \wedge \neg B \wedge \neg C \rightarrow [k]A\}$. The formulas specify that the lower bound for the change brought about by action k is $\{A\}$, and the upper bound

$\{A, B, C\}$. Due to maximal qualification, all the (valuation maximal) models contain the action k. But in some models k brings about only A, in others A and B, and in yet others A, B and C. This means that these models are not semantically equivalent in the sense of definition 4.3.1 (and its generalization to concurrent action in definition 4.3.10).

In section 4.4.3 we identified a problem concerning concurrent qualifications: the mutual exclusion problem. We argued that a possible solution to this problem is to minimize mutual exclusions. As a solution we proposed the mutual exclusion ordering of definition 4.4.4. Minimization over the ordering favors mutual inclusions over mutual exclusions. However, this solution is not independent of the qualification ordering. It makes no sense to apply maximal qualification first: the possibilities of both (1) concurrent actions and (2) their concurrent constituents will be maximized, and a further selection of models that favors inclusions over exclusions is not possible anymore. Thus, if we want to adopt the action description assumption that corresponds with minimization over the ordering of definition 4.4.4, we have to apply minimal mutual exclusion before maximal qualification in the subsequent application of orderings.

The solution to the ramification problem we discussed in section 4.5 does not involve an action description assumption. Therefore, the solution to the ramification problem does not interfere with the minimizations and maximizations over orderings that were given as solutions to the other problems.

4.7 Unique intended models

We call the models of an action description that obey both minimal change and maximal qualification the 'intended models'. In the previous section we showed that intended models can be 'captured' by extensions. Here we investigate how strongly we have to restrict the ADL in order to arrive at a language for which intended models for action descriptions are unique up to semantic equivalence. This makes it possible to generate intended models and calculate entailed properties through model checking. Although we have to restrict ADLs quite severely to reach this goal, still most common action description properties can be expressed, including properties of concurrent actions. First we investigate some obstacles for the uniqueness of intended models.

A condition that has to be satisfied to enable a comparison of two models in any of the orderings discussed in the previous sections, is that for any pair of states s_1, s_2 related by the comparison relation H it holds that $s_1 \in V_1^{\mathcal{P}}(P)$ if and only if $s_2 \in V_2^{\mathcal{P}}(P)$ for all P in an action description. Thus, models of

action descriptions can only be compared if the same diversity of valuations of atomic propositions is present in both models. This means that models that differ in this respect never belong to the same equivalence classes for the combined minimal change / maximal qualification ordering. We thus get separate intended models for each subset of the admissible valuations for atomic propositions in an action descriptions AD. This implies that even for action descriptions with no modalities at all, intended models are not unique: for instance, the action description consisting of the single static formula $A \vee B$ has seven intended modal action models: we can construct 3 different modal action models with one world, 3 with two worlds, and 1 with three worlds. But, the model with three worlds, each corresponding to a separate propositional model of the formula, is the one that already encodes all relevant information. We would like to consider only such 'valuation maximal' models, and prove uniqueness of intended models with respect to them. We prove that preferential entailment is not affected by the restriction to valuation maximal models.

Definition 4.7.1 (valuation maximal models) *A valuation maximal model $\mathcal{M} = (S, R^{\mathcal{A}}, V^{\mathcal{P}})$ of an action description AD is a model of AD that is maximal in the following ordering. Let $\mathcal{M}_1 = (S_1, R_1^{\mathcal{A}}, V_1^{\mathcal{P}})$ and $\mathcal{M}_2 = (S_2, R_2^{\mathcal{A}}, V_2^{\mathcal{P}})$ be two models over $\mathcal{A}$ and $\mathcal{P}$. Then $\mathcal{M}_1 \sqsubseteq^v \mathcal{M}_2$ if and only if there is a total function $H : S_1 \to S_2$, such that for all P $s_1 \in V_1^{\mathcal{P}}(P)$ if and only if $H(s_1) \in V_2^{\mathcal{P}}(P)$.*

As exemplified above, for a formula that contains no modalities, a typical valuation maximal (modal action) model is the one where each propositional logic model of the formula corresponds with a valuation of propositions in one of the states of the model. Valuation maximal models have also been called 'full models' [170].

Definition 4.7.2 (valuation maximal global entailment) *An action description AD vm-globally entails ψ, notation $AD \models_G^{vm} \psi$, if and only if all valuation maximal models for AD are also models for ψ.*

Proposition 4.7.1 *$\Phi \models_G \psi$ if and only if $\Phi \models_G^{vm} \psi$.*

Proof

($\Rightarrow$) Immediate from definition 2.1.4 for entailment, and definition 4.7.2 for valuation maximal global entailment that says that valuation maximal models are a subset of the set of all models for an action description.

($\Leftarrow$) From negative demonstration. Assume that $\Phi \models_G^{vm} \psi$ and $\Phi \not\models_G \psi$. Then there is a model $\mathcal{M}_1$ for Φ that is not valuation maximal, and that is not a model for ψ. Now take an arbitrary valuation maximal model $\mathcal{M}_2$ for Φ. Then, the disjoint union $\mathcal{M}_1 \uplus \mathcal{M}_2$ is also valuation maximal. For disjoint unions it holds that a finite set of formulas is globally valid on $\mathcal{M}_1$ and on $\mathcal{M}_2$ if and only if it is globally valid on $\mathcal{M}_1 \uplus \mathcal{M}_2$. But then, with $\Phi \models_G^{vm} \psi$, we get that ψ is globally valid on $\mathcal{M}_1 \uplus \mathcal{M}_2$, and thus also on $\mathcal{M}_1$. This contradicts that $\mathcal{M}_1$ is not a model for ψ.

■

So, it is justified to consider only valuation maximal models, since this does not affect the reasoning. Therefore, we also consider the uniqueness question (up to semantic equivalence) only with respect to valuation maximal models. We now consider some examples for which valuation maximal intended models are not unique (in the sequel we simply talk of 'intended models', in order not to have to repeat the adjective 'valuation maximal' every time).

For the action description $\{\neg A \wedge \neg B \rightarrow [a](A \vee B)\}$ there are intended models that are not semantically equivalent. In particular, the intended model where there is exactly one way to perform a, that brings about A, but not B, and the intended model where there is exactly one way to perform a, that brings about B, but not A, are not semantically equivalent. The models are both intended since they are maximal in the qualification ordering (in both models a is possible under the condition $\neg A$), and minimal in the change ordering. Obviously, the non-uniqueness is caused by the disjunctive effect information.

In the action description $\{\neg A \rightarrow [a]A, \neg(\neg A \wedge \neg B \wedge \neg C) \rightarrow [a]\bot, A \rightarrow (B \vee C)\}$ the absence of disjunctive information in the effect formulas $\neg A \rightarrow [a]A$ is not sufficient to ensure determinism of action: the non-determinism is due to non-deterministic indirect effects that are specified by the static constraint. There is an intended model where action a brings about A and B, and there is an intended model where action a brings about A and C.

For the action description $\{\neg A \wedge \neg B \wedge \neg C \rightarrow [a]A, \neg A \wedge \neg B \wedge \neg C \rightarrow \langle a\rangle(B \wedge C)\}$ we have that in intended models, the action a in the situation where $\neg A \wedge \neg B \wedge \neg C$ brings about at least A and at most A and B and C. But then the model where from a state where $\neg A \wedge \neg B \wedge \neg C$ the action a has two transitions, one corresponding to the minimum change that brings about A, and one corresponding to the maximum change bringing about A and B and C, and the model that in addition has the middle transition that brings about A and B, are both intended but not semantically equivalent.

The action description $\{\neg A \rightarrow ([a]\bot \vee [b]\bot)\}$ expresses that either a or

b is not qualified under the condition $\neg A$. Maximal qualification thus gives two intended models: one for which under the condition $\neg A$ the action a is possible, and one where b is possible (minimal change information plays no role in this example).

We define the syntax of an action description language that avoids the causes for non-uniqueness of intended models in the above examples.

Definition 4.7.3 (restricted ADL syntax) *Given a set $\mathcal{A}$ of action symbols with $a \in \mathcal{A}$, and a set $\mathcal{P}$ of proposition symbols with $P \in \mathcal{P}$, the syntax of an action description formula σ is defined as:*

$$\begin{array}{lcl} \sigma, \tau, \ldots & ::= & (L_1 \wedge \ldots \wedge L_u) \vee (L_1 \wedge \ldots \wedge L_v) \mid \varphi \rightarrow [\alpha](L_1 \wedge \ldots \wedge L_w) \\ L & ::= & P \mid \neg P \\ \varphi, \psi, \ldots & ::= & P \mid \neg\varphi \mid \varphi \wedge \psi \\ \alpha, \beta, \ldots & ::= & a \mid \alpha \cap \beta \end{array}$$

This language for action descriptions AD is a subset of the language of the logic MAL($\cap$), that (1) limits the use of modalities to formulas of the form $\varphi \rightarrow [\alpha](L_1 \wedge \ldots \wedge L_w)$, thereby excluding nestings of modalities, and (2) constrains the form of non-modal formulas (static constraints) to the form $(L_1 \wedge \ldots \wedge L_u) \vee (L_1 \wedge \ldots \wedge L_v)$. It is easy to check that none of the examples with non-unique intended models is expressible. The semantics for the ADL follows from the semantics of the individual operators defined in chapter 2. Note that the effect formulas only enable the specification of lower bounds for changes. In combination with the minimization of change we thus get that effects of concurrent actions are completely determined (we make use of this property in the proof for the next theorem). This gives rise to the following property, that we give without a proof: $\mathcal{M}_1 \simeq^{ch}_{s1b} \mathcal{M}_2$ if and only if $\mathcal{M}_1 \equiv^{ch}_{s1b} \mathcal{M}_2$. The following theorem says that for the defined ADL intended models are unique.

Theorem 4.7.2 *For any action description AD in the language of definition 4.7.3, there is a valuation maximal, $\sqsubseteq^{ch}_{s1b}$-minimal, $\sqsubseteq^{ql}_{s1b}$-maximal model that is unique up to $\simeq_{s1b}$-equivalence.*

Proof

By negative demonstration. Assume that there are two intended valuation maximal models $\mathcal{M}_1$ and $\mathcal{M}_2$ for AD, that by definition 4.3.10 are not semantically equivalent. The definition gives three possible reasons for two models not to be semantically equivalent.

The first case concerns the situation where there is a state in one of the models that cannot be matched (by a relation H) to a state with an identical valuation of atomic propositions in the other model. But this contradicts the valuation maximality of both models.

The second case concerns the situation where models do agree on the diversity of valuations of atomic propositions in states, but not on all qualifications of concurrent actions in these states. Thus, without loss of generality we may assume that for some pair of states (s_1, s_2) with $s_1 \in S_1$ and $s_2 \in S_2$ obeying $s_1 \in V_1^{\mathcal{P}}(P)$ if and only if $s_2 \in V_2^{\mathcal{P}}(P)$ for all P, it holds that there is a concurrent step T and some t_1 such that $s_1(T)^{\triangleright}t_1$ relative to $\mathcal{M}_1$, while there is no t_2 such that $s_2(T)^{\triangleright}t_2$ relative to $\mathcal{M}_2$. So, although s_1 and s_2 have the same valuation of atomic propositions, there is a strict qualification difference between them: step T is possible in s_1 but not in s_2. Since $\mathcal{M}_2$ is maximal in the $\sqsubseteq^{ql}_{s1b}$-ordering, there has to be a set of formulas Φ of the form $\varphi \rightarrow [\bigcap A](L_1 \wedge \ldots \wedge L_w)$ (with A a set of atomic actions) in AD such that $\mathcal{M}_2, s_2 \models \varphi$ and $A \subseteq T$, such that the set Λ of all effect literals appearing in effects of formulas in Φ in combination with the set Ψ of static constraints of the form $(L_1 \wedge \ldots \wedge L_u) \vee (L_1 \wedge \ldots \wedge L_v)$ in AD form an inconsistent set. But then, the sets Φ and Ψ also prevent the qualification of T in state s_1 of model $\mathcal{M}_1$, which contradicts the presence of a strict qualification difference between s_1 and s_2.

The third case concerns the situation where the models agree on the diversity of valuations of atomic propositions in states, and on all qualifications of concurrent actions, but not on all postconditions for concurrent actions. Thus, for some pair of states (s_1, s_2) and step T obeying $s_1 \in S_1$ and $s_2 \in S_2$ and $s_1 \in V_1^{\mathcal{P}}(P)$ if and only if $s_2 \in V_2^{\mathcal{P}}(P)$ for all P and there is state t_1 such that it holds that $s_1(T)^{\triangleright}t_1$ relative to $\mathcal{M}_1$, it holds that for some state t_2 such that $s_2(T)^{\triangleright}t_2$ relative to $\mathcal{M}_2$ it is not the case that $t_1 \in V_1^{\mathcal{P}}(P)$ if and only if $t_2 \in V_2^{\mathcal{P}}(P)$ for all P. This contradicts the property that for the above ADL, the $\sqsubseteq^{ch}_{s1b}$-ordering determines a unique successor valuation (not a unique state, see example below) for any state and qualified concurrent action T. We show how to construct the unique successor valuation for step T in s_1. Again we consider the set of effect formulas Φ in AD that is relevant for this step in this state, and the set Λ of corresponding effects (in the form of literals whose validity is brought about). We consider two cases:

Case 1: The update of the valuation of atomic proposition in s_1 with the literals of Λ (this update is defined in an obvious way) results in a valuation for which there is a state t_1 in S_1. This means that we have found a unique successor valuation. It is unique, since the minimal change enforced by the

effect formulas in Φ is uniquely described by Λ.

Case 2 The update of the valuation of atomic proposition in s_1 with the literals of Λ results in a valuation for which there is no state t_1 in S_1. This means that the static constraints give rise to secondary effects. Assume, without loss of generality, that there is a subset Ξ of static constraints $(L_1 \wedge \ldots \wedge L_u) \vee (L_1 \wedge \ldots \wedge L_v)$ in AD for which the left conjunction of literals is inconsistent with the update of the valuation for s_1 with the literals in Λ. It cannot be that for such formulas also the right conjunction of literals gives an inconsistency in this sense, for then step T would not be qualified. So, for all static constraints in Ξ, we may add the literals of the right side to the 'effect literals' in Λ to obtain Λ', and update the valuation of s_1 with Λ'. Then the process repeats itself. We check whether there is a state with a valuation corresponding to the new update, and if not, we again select a set of static constraints Ξ' to obtain an 'extended' update. Since the step T is qualified, and since there are only finitely many static constraints, the process will stop after a finite amount of repetitions, such that eventually a unique update set Λ'' is obtained. The unique minimal change determined by the update set Λ'' gives a unique successor valuation for the performance of T in s_1.

■

The proof shows that the language enables a quite straightforward construction of intended models. Such models are valuation maximal, but not valuation unique, i.e. it is not the case that each state in the unique intended model has a unique valuation of atomic propositions. A counter example is $AD = \{\neg A \rightarrow [a \cap b]\bot, \neg A \rightarrow [a]A, \neg A \rightarrow [b]A\}$. Actions a and b cannot be performed concurrently, but have the same effect. Under minimal change this means that they update the state where $\neg A$ holds in exactly the same way, but that they have separate resulting states. These two separate resulting states thus have identical valuations of atomic propositions.

4.8 Related work

In this section we discuss some approaches that are related to ours. We distinguish two categories: the purely syntactic approaches based on the definition of extensions, and the semantic modal approaches.

4.8.1 Approaches to extension construction in modal action logics

We discus some approaches to the frame problem that focus on the construction of extensions. In several approaches it is assumed that it is possible to express persistency in modal action logics by defining extensions with the help of formulas of the type $L \rightarrow [a]L$, with L a positive or negated propositional constant, and a an atomic action. For instance, Castilho, Gasquet and Herzig [91, 44] make use of an explication of the dependency of fluents on action occurrences, and use this explication to define how formulas of the form $L \rightarrow [a]L$ should be added to action descriptions to obtain the intended semantics. Another approach that attempts to define extensions with formulas of the form $L \rightarrow [a]L$ is that by Giordano, Martelli and Schwind [75]. Just as for the language of Castilho, Gasquet and Herzig, the method only applies to the fragment of modal action logic without iteration. But the language of Giordano, Martelli and Schwind does include concurrency. The basic idea is to define extensions of action descriptions by adding formulas $L \rightarrow [\alpha]L$ in such a way that it is not derivable from the resulting completed description that action α does change the value of the literal L. Weaknesses of the approach are the appearance of multiple extensions and the absence of solutions to related problems, such as the qualification and ramification problem.

We have some general objections against approaching the frame problem by focusing on formulas of the form $L \rightarrow [a]L$. First of all, for more general modal action logic formulas $\psi \rightarrow [\alpha]\varphi$, the actions nor the properties that change value are required to be atomic. So in general the question of how to explicitly construct such extensions correctly and systematically is far from trivial, especially if there is no formal intended semantics with respect to which we can prove intention-safety and -sufficiency. A second problem for this approach is that indeed extensions with formulas of the form $L \rightarrow [a]L$ are in general not intention-sufficient. First of all, action effect information can be conditional on pre-conditions, which means that also persistency information has to be conditional on action pre-conditions. But also we sometimes need formulas that express that it is *possible* to perform an action in such a way that it preserves a certain value. In section 4.3.1 we mentioned that for examples where lower bounds and upper bounds for changes are specified, we need frame formulas of both the forms $\psi \wedge \varphi \rightarrow [\alpha]\varphi$ and $\psi \wedge \varphi \rightarrow \langle\alpha\rangle\varphi$.

Also stronger languages for the expressions of frame properties have been considered in the literature. First of all there is De Giacomo's encoding [73] of Reiter's monotonic solution [155, 20] to the frame problem that uses the notion of action complement. We mentioned this solution in section 2.5.1,

where we used it to justify the importance of the notion of action complement for reasoning about action and persistency. The main idea of this approach is to first take stock of all actions in an action description that may influence a certain atomic property. This way the dependency of value changes of fluents on execution of actions is described explicitly. After this process is completed, formulas (successor state axioms) saying that certain formulas are exclusively influenced by certain actions can be added to impose the intended semantics. For his approach De Giacomo uses an action negation similar to the relativized action negation $\wr^K$ that we defined in section 2.5.3. Formulas of the form $\psi \wedge L \rightarrow [\wr^K \alpha]L$ and $\psi \wedge \neg L \rightarrow [\wr^K \alpha]\neg L$ can be used to express that performance of other actions than α leave the value of L unchanged. De Giacomo does not define a general procedure for addition of formulas of the form $\psi \wedge \neg L \rightarrow [\wr^K \alpha]\neg L$. The main contribution is thus that it is shown that with the help of action complement we can express frame properties succinctly.

Prendinger [153] focuses on how to express frame properties in an easy, intuitive and economic way in PDL. His main contributions are the addition of the following operators to PDL: (1) terminal preservation expressed by $tpress(\varphi, \alpha)$, with the intended meaning that φ holds before and after execution of α, (2) chronological preservation expressed by $cpress(\varphi, \alpha)$ with the intended meaning that φ is preserved throughout the execution of α. Prendinger defines the semantics of these constructs together with their axiomatization, and gives classical soundness and completeness results for the resulting logic. But the notions of terminal and chronological preservation can also be given as a definitional extension.

Proposition 4.8.1 *The notions of terminal preservation $tpres(\varphi, \alpha)$ and chronological preservation $cpres(\varphi, \alpha)$ of φ over α, as defined by Prendinger [153] can be defined in PDL.*

Proof

We define a translation T of formulas $tpres(\varphi, \alpha)$ and $cpres(\varphi, \alpha)$ to plain PDL:

$$
\begin{array}{lcll}
T(tpres(\varphi, \alpha)) & \equiv & \varphi \rightarrow [\alpha]\varphi & \\
T(cpres(\varphi, a)) & \equiv & T(tpres(\varphi, a)) & \text{for } a \in \mathcal{A} \\
T(cpres(\varphi, \alpha \cup \beta)) & \equiv & T(cpres(\varphi, \alpha)) \wedge T(cpres(\varphi, \beta)) & \\
T(cpres(\varphi, \alpha; \beta)) & \equiv & T(cpres(\varphi, \alpha)) \wedge [\alpha]T(cpres(\varphi, \beta)) & \\
T(cpres(\varphi, \alpha^*)) & \equiv & [\alpha^*]T(cpres(\varphi, \alpha)) &
\end{array}
$$

It is straightforward to check that each of the above translation clauses preserves Prendinger's semantics of $tpres(\varphi, \alpha)$ and $cpres(\varphi, \alpha)$ on modal action models.

■

An implication of the proposition is that the properties $tpress(\varphi, \alpha)$ and $cpress(\varphi, \alpha)$ do not add expressiveness to PDL, but are just convenient abbreviations for complex PDL formulas. Only in this way the preservation constructs contribute to the representational aspect of the frame problem. Another implication is a refinement of the commonly made claim (e.g. [90]) that in PDL we cannot talk about properties that hold 'along the way' when executing a program: it is true that we cannot say in PDL for instance 'there *is* a possibility to successfully execute α while preserving φ along the way', but definition 4.8.1 demonstrates that we can say 'for all possible executions of α we preserve φ along the way'.

A natural question is whether extensions using the formulas $tpress(\varphi, \alpha)$ and $cpress(\varphi, \alpha)$ can be proven intention-safe and sufficient with respect to the semantics for sequential minimal change we gave in section 4.3.2. This seems very difficult. The extension will have to account for the delicate way in which minimal change is distributed over sequences of actions. The constructs $cpress(\varphi, \alpha)$ and $cpress(\varphi, \alpha)$ do not help much, because in such examples as the stolen car scenario a construct that defines persistency 'along the way' cannot be used to impose that the change is brought about by either the first, second or third action.

4.8.2 Semantic modal approaches

It is surprising that most existing approaches to the problems with reasoning about action and change focus on representational issues and extension procedures. This is even more surprising if we realize that the emphasis in most first-order-based approaches to the same problems (e.g. [155, 163, 80, 178]) is on the semantics, that is, on the selection of the intended first-order models of action descriptions. An exception to this is the work by Foo e.a. [62] who give a syntactic and a semantic characterization of their modal action logic-based solution to the frame problem, and prove that it is intention-safe and intention-sufficient (in our terminology). We already mentioned in section 4.3.1 that their minimization strategy for change does not only minimize change, but also action possibilities. Therefore, in action descriptions, it is required to give sufficient preconditions for all actions, such that action possibilities are not minimized by the change ordering. We argued that it should

not be required to give qualification information in order to guarantee that the minimization strategy for change works correctly.

Definition 4.8.1 (minimal change models of Foo e.a. [62]) *Let $Chg(\mathcal{M})$ be defined by: $(a, P, s) \in Chg(\mathcal{M})$ if and only if there exists an accessible state s' to s on action a such that the truth value of P is different at s and s'. Then, for any $\mathcal{M}_1 = (S_1, R_1^{\mathcal{A}}, V_1^{\mathcal{P}})$ and $\mathcal{M}_2 = (S_2, R_2^{\mathcal{A}}, V_2^{\mathcal{P}})$ satisfying a modal action description, it holds that $\mathcal{M}_1 \sqsubset_{Foo} \mathcal{M}_2$ if and only if:*

1. $S_1 = S_2$
2. $V_1^{\mathcal{P}}(P) = V_2^{\mathcal{P}}(P)$ *for all* P
3. $Chg(M_1) \subset Chg(M_2)$

Intuitively, $\mathcal{M}_1 \sqsubset_{Foo} \mathcal{M}_2$ means $\mathcal{M}_1$ has less state change than $\mathcal{M}_2$.

In this ordering, change is not separated from action possibility (qualification). Now assume that we use this ordering to select minimal change models for the YSP action description we gave in section 4.3.1. This action description did not contain any qualification information. This means that there are models of the YSP action description for which the relation structure is empty. Because the above ordering minimizes action qualifications, we get such models as the result of minimization. Foo et al. explicitly add $\langle load \rangle \top, \langle wait \rangle \top$, and $\langle shoot \rangle \top$ to prevent this. So they have to add explicit qualifications in order to make their minimal change entailment relation behave well.

The second semantic approach we want to mention is the one embodied by the languages $\mathcal{A}$ and $\mathcal{C}$, initiated by Gelfond and Lifschitz [114, 69]. This approach shares some basic concepts with ours, such as the distinction between an action description language (ADL) and an action query language (AQL) and the maximization of qualifications. A notable difference is that our approach stays within the tradition of (multi) modal logic. An advantage of a modal approach is that we can use established modal techniques and results to tackle the problems of action reasoning. The fact that we support the intuitiveness of our semantics with its relation to (variations on) well-known semantic equivalence notions for modal logics (i.e. bisimulations), is a clear example of this. Another example is the possibility to use well established modal model checking techniques for the calculation of properties of unique intended models.

4.9 Conclusions

In this chapter we concentrated on the classical problems of reasoning about action and change. We argued that these problems are relevant for system specification with the modal action logics we defined in chapter 2. The three classical problems we discussed are the frame problem, the qualification problem and the ramification problem. For the ramification problem we discussed a solution of Foo and Zhang [61] that can be imported in our setting without much difficulty. We proposed semantic solutions to the frame and qualification problem for non-sequential, sequential and concurrent action. We defined orderings over modal action models, where we take the model-based semantic equivalence notions for ADLs as the starting point. To our knowledge, the relevancy of semantic equivalence notions was not recognized before in the area of reasoning about action and change. Extensions that are intention-safe and sufficient with respect to intended models defined by minimizations and maximizations over the orderings were shown to exist, but not explicitly constructed. For concurrent action we also identified a possible problem not mentioned before in the literature on reasoning about action and change: the mutual exclusion problem. Finally, we showed how to combine the solutions for the separate problems, and investigated for modal action logic with concurrency, how much we have to restrict the language in order to arrive at an ADL for which intended models are unique up to semantic equivalence.

A possible point of criticism to our work is that it is mainly semantic: for some ADLs, the form of extension formulas is established and the existence of intention-safe and sufficient extensions with this formulas is proven, but extensions were not explicitly generated. However, one should realize that establishing a correct (intuitive) intended semantics is a most important part of the research program that is concerned with the modeling of reasoning about action and change. As an example of the kind of important issues raised by such semantic studies, we mention our claim that the stolen car problem and the Yale shooting problem concern incompatible action descriptions assumptions, which means that they cannot be solved by one and the same intended semantics. We believe that the conceptually attractive abstract representation of action domains as modal action structures enables us to approach such conceptual difficulties more fruitfully.

A main point of criticism on many other semantic solutions in the literature, to the problems discussed in this chapter is that they lack foundational ideas and are mainly example-driven. Sandewall ([163], page 468) says the following about this:

> 'The most important weakness, however, is that various logics have been proposed to be 'solutions to the frame problem' just on basis of intuitions and a few examples, only to be very quickly refuted by the arrival of more examples.'

We argue that this criticism does not apply to our approach. The present approach is motivated from first principles (minimal change, maximal qualification) and modal theory (bisimulations and related semantic equivalences). Therefore, our approach is not mainly example-driven: the examples only play a role in the testing of intuitions, but do not form the starting point of our research.

We did not consider solutions to the problems for ADLs with action negation. In section 2.5.3 we mentioned that for such languages we need semantic equivalence relations that are stronger than those for $\cap$-logics. Since we did not yet work out the theory of such equivalence notions for languages with action complement fully, we leave this as a subject for future work.

Chapter 5

Deontic modal action logic

In this chapter we are concerned with normative reasoning about action. The chapter consists of two parts: one about what we call *goal norms* and a second about *process norms*. This classification, that was not explicitly defined before in the literature, typically applies to the kind of norms we study: norms over actions from an explicit action language. The class of goal norms takes an action norm to be a norm about the result / goal / effect of an action, while normative repercussions for sequential sub-actions are absent. Thus, goal norms may only be violated in the result state of an action; violations cannot occur in the process of reaching this state. We denote norms of this types with the symbols $O_{\odot}(\alpha)$, $P_{\odot}(\alpha)$ and $F_{\odot}(\alpha)$, where O, P and F stand for Obligation, Permission and Prohibition ('Forbiddeness') respectively, and the subscript '$\odot$' depicts that norms concern action goals. For goal norms it holds, for instance, that each of the formulas $P_{\odot}(a;b) \wedge \neg P_{\odot}(a)$, $F_{\odot}(a) \wedge \neg F_{\odot}(a;b)$, and $O_{\odot}(a;b) \wedge \neg O_{\odot}(a)$ is consistent. Thus, it can be permitted to perform $a;b$, while at the same time a permission to perform a is lacking. It may even be the case that a is explicitly forbidden[1], meaning that a brings us to a state satisfying conditions that are not allowed. The point is that it is the result of the action that matters: action $a;b$ is permitted, because it brings about a result that is allowed. This view on norms as compliance to action goals was first adopted by Meyer [135]. Some of our work on deontic action logic takes the same perspective [30, 29, 26, 28]. The view is closely related to views on dealing with planning problems in dynamic logic. For planning in AI, the problem is to find compound actions whose execution results in a compliance to a certain goal (see e.g. [169]). In section 5.2 we develop the goal directed view on action norms. The definition of the logics in this section hinges heavily

[1]Note that we do not assume that $P(\alpha) \leftrightarrow \neg F(\alpha)$, but only that $\neg(P(\alpha) \wedge F(\alpha))$.

on the definition of the relativized action negation of section 2.5.3.

The other class we consider, is the class of action process norms. We denote norms of this types with the symbols $O_{\rightsquigarrow}(\alpha), P_{\rightsquigarrow}(\alpha)$ and $F_{\rightsquigarrow}(\alpha)$, where the subscript '$\rightsquigarrow$' depicts that the norms concern the whole process. For norms of this class, we take the alternative position, that is, that each of the formulas $P_{\rightsquigarrow}(a;b) \wedge \neg P_{\rightsquigarrow}(a)$, $F_{\rightsquigarrow}(a) \wedge \neg F_{\rightsquigarrow}(a;b)$, and $O_{\rightsquigarrow}(a;b) \wedge \neg O_{\rightsquigarrow}(a)$ is inconsistent. So, a permission to perform a sequential compound action requires permissions for all sub-actions. In general we can say that for this type of norms violations may occur at any point during the process of action performance. This view was first adopted by Van der Meyden [132], who criticized the goal view for not being intuitive in general. Our standpoint is that both views may coexist, as long as a specifier does not confuse the two alternative interpretations. Van der Meyden does not deal with the deontic notion of *obligation*. In section 5.3, we study this type of norms. The material in this section hinges heavily on the temporal view, as developed in chapter 3. The section is a reflection of the work we presented in [34, 36].

For both goal norms and process norms we define reductions of deontic operators to modal action logic operators. Due to these reductions, normative assertions are reduced to statements about violation conditions of pre- and postconditions of actions. Thus, normative assertions are interpreted on the same models as assertions about action, and, as we saw in chapter 3, temporal assertions. The goal norm reduction reduces action norms to violation conditions of action postconditions. This is explained in section 5.2. The process norm reduction does not only look at the postcondition of compound actions, but requires obedience of violation conditions during the entire process. To express this, we use (variants of) the modal μ^{η}-calculus, as defined in section 3.5. This reduction is defined in section 5.3. But first we discuss in section 5.1 an issue that is relevant for both types of norms: the distinction between free and imposed choice.

5.1 Free choice versus imposed choice

The distinction between goal and process norms concerns two alternative views on the deontic properties of sequence. A similar situation arises for the choice combinator: do we assume that the pairs $P(a \cup b)$ and $\neg P(a)$, $F(a)$ and $\neg F(a \cup b)$, and $O(a \cup b)$ and $\neg O(a)$ are consistent, or not? We need a coherent view on the interpretation of the choice combinator, that decides all these questions from one central and clear intuition. This view needs to serve the use as specification logics we envision for deontic modal action logics.

We have to ask ourselves in what sense non-deterministic choice actually is non-deterministic. Who or what is it, lacking determination of choice? In our application of deontic reasoning, a system designer enacts norms in order to exercise control over the behavior of a reactive system. The choice in actions $\alpha \cup \beta$ is then non-determinism that reflects an explicit renunciation of the possibility to exercise control. In other words, non-determinism of action reflects deliberate under-specification: the actual outcome of the action is not specified; only that the outcome satisfies certain properties. The specifying agent does not control, and also does not want to control (decide on) the choice in $\alpha \cup \beta$. In particular, he cannot rely on the system being subject to the norm to perform the non-deterministic action $\alpha \cup \beta$ in such a way that no norms are violated. So, if he considers $F(\alpha \cup \beta)$, he means that there is a way to perform $\alpha \cup \beta$ that leads to a violation. And, since he does not (want to) control the choice, this means that the non-deterministic action $\alpha \cup \beta$ is forbidden. In other words: we consider $F(a) \wedge \neg F(a \cup b)$ to be inconsistent. We can draw a parallel with the situation where a father is 'specifying' desired behavior of his child. Because of the non-determinism in the child's activities, he cannot control the behavior of his child completely, even if the child is fully obedient. Assume the father considers the action of climbing the stairs. He knows that his child is capable of climbing the stairs without falling, but he also knows that there is the possibility that he will fall. Because of this 'violation' that is associated with a particular way to perform the stair climbing action, the father declares it to be forbidden to climb the stairs. When an actual execution of a forbidden action does not result in a state that embodies a violation, we just have a lucky coincidence. It does not mean that it was not right after all to have called the action forbidden. The action was forbidden because it could have resulted in a violation state. We give an example on free choice prohibition in section 5.3 on process norms.

For permission, we have a dual situation: specifying $P(\alpha \cup \beta)$, without (wanting) the control over the choice represented by $\cup$, means that any way to perform the action $\alpha \cup \beta$ is permitted. Thus, $P(a \cup b) \wedge \neg P(a)$ is inconsistent.

If a specifier enacts $O(a \cup b)$ without (wanting) to enforce any control over the way in which $\alpha \cup \beta$ is performed, he merely means that any action c for which $R(c) \subseteq R(a \cup b)$ complies to the obligation, while any action c such that $R(c) \setminus R(a \cup b) \neq \emptyset$ does not comply to the obligation, since it has possible outcomes that violate it. This shows that for obligation the situation is somewhat more complicated. At this stage it is too early to go into the semantics of obligation in detail. But we have that $O(a \cup b) \wedge \neg O(a)$ is actually consistent under the proposed interpretation of choice. Also our view on choice

does not imply inconsistency of $O(a \cup b) \wedge O(a)$. The free choice semantics for obligation we define does actually not impose any logical relation on formulas $O(a \cup b)$ and $O(a)$ and $O(b)$. The logics in this chapter will consider any one of these three expressions as *primitive*.

Alternatively, we can take the viewpoint of the system being subject to a choice norm. We do not think of reactive systems as 'having control' or 'not having control', since we do not view them as autonomous. But let us, for now, assume that we do specify autonomous agents. Then, we may say that the explicit renunciation of control on the specification (enactment) part implies that the subject agent is free to choose. This is why we adopt the term 'free choice' for our view on choice. We use this name in order to be consistent with the (ought-to-be) deontic logic literature [103, 92, 121] where the property $P(\psi \vee \varphi) \leftrightarrow P(\psi) \wedge P(\varphi)$ stands for 'free choice permission'. But, we saw that the free choice view is not limited to permissions alone: it extends to prohibition and obligation.

The alternative position, we call it 'imposed choice', is studied most extensively in the literature. In an imposed choice view on norms, $P(a \cup b) \wedge \neg P(a)$ is consistent. Also the work on dynamic deontic logic initiated by Meyer [135, 56] has developed according to the imposed choice view. However, we have some difficulty accepting this view as a conceptual alternative. The imposed choice interpretation seems to suffer from an internal confusion: expressions $P(a \cup b)$ are interpreted to mean that the action $a \cup b$ is permitted, but it is not excluded that we have a violation when performing $a \cup b$: the action is permitted if performed in a certain way, and possibly not permitted if performed in some other way. What justification is there for calling an action $a \cup b$ permitted, if there are possible ways to perform it that are not permitted? The imposed choice principle is defended with examples like the following: 'I permit you to drive my car' does not imply that 'I permit you to drive my car and drink (a concurrent action that by the open interpretation of concurrency as explained in section 2.4.1, counts as a way to perform the driving action)'. But the problem with this is that it does not attack the principle of free choice, but only reveals the incompleteness of normative assertions in normal discourse. In normal discourse, the actual meaning of assertions involves many implicit default assumptions. The agent enacting this norm meant 'I permit you to drive my car, unless at the same time you drink, use your telephone, violate traffic regulations, etc.'. With such an exhaustive set of exceptions added to it, the choice that is present in the car driving action is indeed free. But, in general, such exceptions are not spelled out explicitly: they are assumed as general background knowledge. For a study on how to deal with default

information in deontic logic, we refer to the work of Van der Torre [179].

The distinction between free and imposed choice operators is very much like the distinction between weak and strong deontic operators in the literature. However, it is not right to associate 'free' with 'strong and 'imposed' with 'weak'. The notions of free choice permission [103, 121, 55, 56], free choice prohibition and free choice obligation we consider in this chapter correspond to strong permission [191, 92, 192, 6, 160], weak prohibition [183] and strong obligation [38, 183] respectively.

In section 5.3.1 we discuss that for process norms, the notion of free choice has to be generalized in order to apply to choices that are made during execution of compound sequential actions. This results in the notions of 'free process choice' and 'imposed process choice'. In the above mentioned literature on the formalization of the deontic properties of choice, these concepts are not identified.

Finally we note that the free/imposed distinction has an analogy in the distinction between internal and external choice for process algebras. The connection between external choice and imposed choice is that the lack of choice of an agent that is performing a trace, can be viewed as choice that is externally forced upon the agent by the environment. Internal choice corresponds to free choice, because all choices during execution can be thought of as internal to, or 'under control' of the agent.

5.1.1 The ought-to-be case

The deontic interpretation of choice for ought-to-be deontic logics, has been the subject of quite some controversy. In most ought-to-be deontic logics, $P(\varphi \vee \psi) \wedge \neg P(\varphi)$ is consistent. However, many authors have claimed that this is counterintuitive, since in many examples it seems that a permission for a choice intuitively implies permission of the individual alternatives that make up a choice. This then leads to the property $P(\varphi \vee \psi) \rightarrow P(\varphi) \wedge P(\psi)$, a principle that is known under the name 'free choice permission'. But for ought-to-be deontic logics we do not consider this to be a valid principle. We agree with those authors (e.g. [60]) that claim that the free choice permission 'paradox' for ought-to-be deontic logics is simply a misunderstanding of the intended semantic content of ought-to-be deontic expressions. Indeed, if $P(\varphi)$ is correctly interpreted as 'it is permitted to satisfy the condition φ' it makes perfect sense to say that this entails that 'it is permitted to satisfy the condition $\varphi \vee \psi$', because the condition in this latter assertion is a (propositional) logic consequence of the former condition. For an idealized reasoner it is absurd

to be permitted to satisfy a certain condition while not being permitted its (propositional) logic consequences[2]. This gives a strong justification for the ought-to-be deontic implication $P(\varphi) \vee P(\psi) \rightarrow P(\varphi \vee \psi)$[3]. And indeed, this reading excludes a free choice permission reading of $P(\varphi \vee \psi)$, because we would then have both $P(\varphi) \vee P(\psi) \rightarrow P(\varphi \vee \psi)$, and $P(\varphi \vee \psi) \rightarrow P(\varphi) \wedge P(\psi)$, and thus $P(\varphi) \vee P(\psi) \rightarrow P(\varphi) \wedge P(\psi)$. This leads to a degenerate logic where the permission to satisfy some condition implies the permission to satisfy any condition.

It is argued [93, 121] that for the ought-to-be case, the source of the free choice confusion resides in people's linguistic extra-logical habit to 'abbreviate' $P(\varphi) \wedge P(\psi)$ (or even $P(\varphi) \wedge P(\psi) \wedge P(\varphi \wedge \psi)$) by $P(\varphi \vee \psi)$. We argue that a possible other source of the confusion is that ought-to-do permissions are often mistaken for ought-to-be norms. As an example, consider the norm $P_{\odot}(\textit{go-north} \cup \textit{go-west})$. It is clearly a norm about actions, the actions of going north and going west. And it is very natural to give it a free choice interpretation: an agent who considers himself to be subject to this norm, and that is free to choose, has permission to go north and permission to go west. The above argumentation concerning 'absurdity of not being permitted logical consequences', is irrelevant, because of the absence of a logical connection: the action *go-west* cannot be said to 'logically entail' the action $\textit{go-north} \cup \textit{go-west}$, simply because they are not propositions. In the modal action logics we study in this thesis, actions are what Castañeda [41, 42, 43] calls 'practitions'. A practition is not a proposition, that is, not something that can be true of a world / situation / state / moment. But we can have propositions that are about practitions. In the modal action logics we study in this thesis, the basic forms of the propositions concerning actions (practitions) α are $\langle\alpha\rangle\varphi$ and $[\alpha]\varphi$.

5.2 Action goal norms

In this section, we study reductions of deontic operators representing goal norms to the modal action operators $\langle\alpha\rangle\varphi$ and $[\alpha]\varphi$ and violation conditions. Such a reduction was first proposed by Meyer [135]. We define our version of such a reduction in section 5.2.1, and demonstrate its intuitiveness in two

[2]Note that this is closely related to the omni-science problem [96] for reasoning about knowledge, that is in a sense dual: reasoners are in general not ideal, that is, they do not know all the logic consequences of a proposition. Also, the same problem arises in reasoning about intentions and desires [50].

[3]But note that it gives *no* justification for the property $P(p) \vee P(q) \leftrightarrow P(p \vee q)$, obeyed by many deontic systems.

ways: first, we show that it gives rise to intuitive deontic validities, and second, we show that it avoids well-known anomalies from the literature. Hilpinen [94] argues that Castañeda's distinction between practitions and propositions about practitions provides solutions to many of the contrary to duty benchmark examples, but not to the one about the 'gentle murderer'. We show that our approach avoids this problem.

In the reduction proposed by Meyer the violation condition corresponds with validity of the proposition V, and the negation of an action α is denoted by $\overline{\alpha}$.

Definition 5.2.1 (Meyer's reduction to modal action logic)

$$P_{\odot}(\alpha) \equiv_{def} \langle\alpha\rangle\neg V$$
$$F_{\odot}(\alpha) \equiv_{def} [\alpha]V$$
$$O_{\odot}(\alpha) \equiv_{def} [\overline{\alpha}]V$$

The reduction is faithful to the goal norm intuition in the sense that violations are modeled as postconditions of compound actions α. Meyer's action negation is part of an action algebra that obeys axioms that are considered intuitively desired. It is shown that these axioms are consistent by providing a model theory. The algebra is then used as an interpretation for dynamic logic actions: syntactically, the actions of the algebra are identified with the actions within the modal box of dynamic logic, and semantically, a connection is made between the algebraic semantics and the relational semantics on dynamic logic models. However, the connection between the modal part and the algebraic part leaves room for alternative interpretations, which makes it unclear how to lift the algebraic axioms to the modal level. Also, it is not clear how to generalize the action negation $\overline{\alpha}$ such that it encompasses iteration and converse of action. Another possible drawback of the action algebra is that its notion of action negation seems to conflict with the goal norm intuition. The action negation obeys the following axion of the action algebra: $\overline{\alpha;\beta} = \overline{\alpha} \cup \alpha;\overline{\beta}$[4]. This in turn, with the above reduction, gives rise to the following property for obligations: $O_{\odot}(\alpha;\beta) \leftrightarrow O_{\odot}(\alpha) \wedge [\alpha]O_{\odot}(\beta)$. However, this property is not consistent with the goal view on action norms: the implication $O_{\odot}(\alpha;\beta) \rightarrow O_{\odot}(\alpha)$ shows a normative repercussion for the sequential sub-action α, because the norm $O_{\odot}(\alpha;\beta)$ is already violated if α is not performed. We deal with such possible violations for sequential sub-actions in section 5.3, where we study the process view on norms.

[4]Note that there seems to be some structural correspondence with the validity $\langle\imath^I(\alpha;\beta)\rangle\varphi \rightarrow [\alpha]\langle\imath^I\beta\rangle\varphi$ that we met in chapter 2.

5.2.1 A cautious reduction

We deviate from Meyer's reduction in definition 5.2.1 on four points. First of all, we use the relativized action negation as defined in 2.5.3, thereby avoiding the above mentioned problems with the negation $\overline{\alpha}$. In section 2.5.3 we mentioned that the relativized action negation aims at modeling the notion of 'alternative action'. This is exactly the interpretation that is required for deontic modal action logic, as we will see.

The second aspect on which we deviate from Meyer, is the interpretation of choice. The reduction of definition 5.2.1 takes the imposed view on choice: an action is permitted if and only if there is a way to perform it that does not lead to a violation. In the previous section we argued that we adopt the free choice paradigm in this chapter.

The third aspect concerns the interdefinability of the operators in the reduction: equivalently, we could have defined $F_{\odot}(\alpha) \equiv_{def} \neg P_{\odot}(\alpha)$, and $O_{\odot}(\alpha) \equiv_{def} F_{\odot}(\wr^{I}\alpha)$. In the reduction we define below, we avoid these strong interdefinabilities. The definition $P_{\odot}(\alpha) \equiv_{def} \neg F_{\odot}(\alpha)$ is avoided in order to leave room for action not being normed. In section 1.4.4 we argued this is important to allow for the possibility of 'gaps'. We avoid $O_{\odot}(\alpha) \equiv_{def} F_{\odot}(\wr^{I}\alpha)$, because we assume that it is not always the case that $O_{\odot}(\alpha) \leftarrow F_{\odot}(\wr^{I}\alpha)$ holds. We thus assume that for an obligation to perform an action it sometimes takes more than a prohibition to perform any other action. Obligations that follow from prohibitions for alternative action can said to be 'negatively' motivated. The problem then is, that nothing is said about possible violations after performance of the negatively motivated action itself. For instance, a prohibition to do anything but wait, does not imply the free choice obligation to wait, because the sub-action of waiting and smoking may also be forbidden. Also in the other branches of ought-to-do deontic logics, the interdefinabilities have been disputed. Hintikka [95] and Maibaum [120], avoid interdefinabilities $F_{\odot}(\neg p) \equiv_{def} O_{\odot}(p)$ and $\neg P_{\odot}(\neg p) \equiv_{def} O_{\odot}(p)$ for reasons similar to ours. And Von Wright emphasized [194] that since 'norm and action' ([190]), he has considered obligation and permission to be not interdefinable. We avoid the interdefinabilities between permission, prohibition and obligation by introducing a 'violation proposition' for each of them: V_F for the violation of a prohibition, V_O for the violation of an obligation, and V_P for lack of explicit permission.

Finally, we also deviate from Meyer in that we impose the 'ought implies may' principle [6, 142]. This is the standpoint that it is inconsistent to assert that $O(\alpha) \wedge \neg P(\alpha)$. This says that enacting a (free choice) obligation for an action α necessarily (logically) subsumes the enactment of a (free choice)

permission for the same action. This seems not only intuitively valid, it also blocks Ross' 'paradox' [159, 133], which is embodied by the undesirable property $O(\alpha) \rightarrow O(\alpha \cup \beta)$. Assume that we have $O(\alpha)$. We show that applying both the 'ought implies may' principle and Ross' property leads to an inconsistent logic. Intuitively, the obligation $O(\alpha)$ implies that action other than α is forbidden. In particular, under the free interpretation of choice, for any action β for which $R(\beta) \setminus R(\alpha) \neq \emptyset$ it holds that $F(\beta)$. Then, if we conclude $O(\alpha \cup \beta)$ by following Ross' property, we run into a contradiction: with 'ought implies may' we conclude from $O(\alpha \cup \beta)$ that $P(\alpha \cup \beta)$, which under the free choice interpretation contradicts with $F(\beta)$.

With the above four deviations from Meyer's definition, we arrive at the following reductions for the operators for permission, prohibition and obligation, where (a) free choice permission for α is identified as the absence of a possibility to do α in a way that results in a permission violation, (b) free choice prohibition for α is identified as the possibility to do α in a way that results in a prohibition violation, and (c) free choice obligation for α is identified as the conjunction of free choice permission for α and the occurrence of an obligation violation after any action performance not involving α:

Definition 5.2.2 (A cautious reduction to modal action logic)

$$\begin{array}{ll} (a) & P_{\odot}(\alpha) \equiv_{def} \neg\langle\alpha\rangle V_P \\ (b) & F_{\odot}(\alpha) \equiv_{def} \langle\alpha\rangle V_F \\ (c) & O_{\odot}(\alpha) \equiv_{def} P_{\odot}(\alpha) \wedge [\wr^I \alpha] V_O \end{array}$$

Except for $O_{\odot}(\alpha) \rightarrow P_{\odot}(\alpha)$, this reduction to modal action logic does not impose any relation between the deontic modalities. But we can introduce these relations by defining dependencies between the different types of violation. We may for instance define, as in Meyer [135], that there is no difference between the violation of an obligation, the violation of a prohibition, and the absence of accordance with a permission: $V_O \leftrightarrow V_F \leftrightarrow V_P$. But then, we get the strong interdefinabilities back. We can be much more cautious. We require, that we cannot at the same time (1) be in accordance with a permission and violate a prohibition, and (2) violate an obligation and not violate a prohibition. In formulas:

Definition 5.2.3 (Violation dependencies)

$$\begin{array}{ll} (d) & \neg(\neg V_P \wedge V_F) \\ (e) & \neg(V_O \wedge \neg V_F) \end{array}$$

Note that under these constraints states may satisfy $\neg V_P \wedge \neg V_O \wedge \neg V_F$, indicating that there is room for indifference with respect to normation (see section 1.4.4). The reduction of definition 5.2.2 is easily adapted in order to satisfy other requirements. We not only can get back the strong interdefinabilities between deontic modalities, we can also adapt the reduction in such a way that we exchange free choice for imposed choice. Several variations are possible. In the next section we discuss the deontic properties this definition gives rise to.

5.2.2 Some deontic properties

We give some deontic logic validities induced by the reduction of definition 5.2.2 and 5.2.3. To assess the intuitive correctness of the properties it is important that choice is to be read as 'free choice', and that concurrency has an open interpretation, as explained in section 2.4.1. The first four properties say that: an action cannot be (1) at the same time permitted and forbidden, (2) at the same time be obliged and not permitted, (3) obliged, while alternative actions. i.e. actions with a possible (remember free choice) different outcome, are not forbidden, (4) obliged, while an action whose set of possible outcomes is different is also obliged. In formulas:

Proposition 5.2.1

(1) $\neg(P_\odot(\alpha) \wedge F_\odot(\alpha))$
(2) $\neg(O_\odot(\alpha) \wedge \neg P_\odot(\alpha))$
(3) $\langle \beta \cap \wr^I \alpha \rangle \top \rightarrow \neg(O_\odot(\alpha) \wedge \neg F_\odot(\beta))$
(4) $\langle \wr^I(\alpha \div \beta) \rangle \top \rightarrow \neg(O_\odot(\alpha) \wedge O_\odot(\beta))$

Proof

Logic property 1 follows easily from the properties a, b and d. Property 2 follows directly from c. To show that property 3 holds, we apply the reduction to obtain $\langle \beta \cap \wr^I \alpha \rangle \top \rightarrow \neg(\neg\langle\alpha\rangle V_P \wedge [\wr^I \alpha] V_O \wedge \neg\langle\beta\rangle V_F)$. We rewrite this as $\langle \beta \cap \wr^I \alpha \rangle \top \rightarrow \langle\alpha\rangle V_P \vee \langle \wr^I \alpha \rangle \neg V_O \vee \langle\beta\rangle V_F$. Then, with $V_P \vee \neg V_O$ that follows from (d) and (e) this is easily seen to be valid in any $\wr^I$-logic. Property 4 is verified in the same way. It reduces to $\langle \wr^I(\alpha \div \beta) \rangle \top \rightarrow \neg(\neg\langle\alpha\rangle V_P \wedge [\wr^I \alpha] V_O \wedge \neg\langle\beta\rangle V_P \wedge [\wr^I \beta] V_O)$, which is also easily checked with the property $\neg(V_O \wedge \neg V_P)$, that follows from (d) and (e).

■

The first two properties together imply the property $\neg(O_\odot(\alpha) \wedge F_\odot(\alpha))$. Note that the properties 3 and 4 are stronger than respectively the properties

$O_\odot(\alpha) \rightarrow F_\odot(\wr^I\alpha)$ and $O_\odot(\alpha) \rightarrow \neg O_\odot(\wr^I\alpha)$. That is, $O_\odot(\alpha)$ not only implies the prohibition of the action $\wr^I\alpha$, it implies the prohibition of any action whose outcome (effect) might (recall that choice is free) violate the obligation. And $O_\odot(\alpha)$ not only implies the absence of the obligation to do the action $\wr^I\alpha$, it implies the absence of any obligation to do an action whose set of possible outcome states is different. A simple consequence of this property is $[\alpha \cap \beta]\bot \rightarrow \neg(O_\odot(\alpha) \wedge O_\odot(\beta))$, saying that if two actions cannot be performed concurrently, they cannot be both obliged. Property 4 reflects that obligation is a much stronger operator than permission or prohibition. An obligation for α divides the reachable state-space in two categories: the states reachable by α obey $\neg V_P \wedge \neg V_O \wedge \neg V_F$, and the states that form the complement of α with respect to what is reachable by any compound action in the action language (the relativized action negation), obey $V_P \wedge V_O \wedge V_F$. Permission and prohibition impose much weaker conditions on the state-space, in particular they do not completely determine the value of all three violation conditions V_F, V_O and V_P in each state in the state-space.

The above properties mainly concern the interactions between separate deontic modalities. Below we formulate for each modality individually how it interacts with (free) choice and (open action) concurrency. It holds that: (5) permission to choose between α and β is equivalent to permission to do α together with permission to perform β, (6) permission to perform α implies permission to perform α concurrent with β (remember the open action interpretation and free choice), (7) prohibition to choose between α and β is equivalent to prohibition to do α or prohibition to perform β, (8) prohibition to perform α and β simultaneously implies prohibition to perform α and prohibition to perform β, (9) the obligation to perform α and the obligation to perform β together are equivalent with the obligation to perform α and β simultaneously. In formulas:

Proposition 5.2.2

$$
\begin{array}{ll}
(5) & P_\odot(\alpha \cup \beta) \leftrightarrow P_\odot(\alpha) \wedge P_\odot(\beta) \\
(6) & P_\odot(\alpha \cap \beta) \leftarrow P_\odot(\alpha) \vee P_\odot(\beta) \\
(7) & F_\odot(\alpha \cup \beta) \leftrightarrow F_\odot(\alpha) \vee F_\odot(\beta) \\
(8) & F_\odot(\alpha \cap \beta) \rightarrow F_\odot(\alpha) \wedge F_\odot(\beta) \\
(9) & O_\odot(\alpha \cap \beta) \leftrightarrow O_\odot(\alpha) \wedge O_\odot(\beta)
\end{array}
$$

Proof

The association of the permission operator with a dynamic logic box operator (a) ensures that permission obeys the properties for free choice. The

properties for choice and concurrency all follow from the choice for either a box or a diamond operation for these modalities.

■

We cannot avoid to test the above deontic properties for choice and concurrency against the notorious benchmark examples for them in the literature. The two famous Benchmark examples for the deontic properties of choice are the paradox of free choice permission and Ross' paradox [159] concerning choice and obligation. We already discussed both problems. The free choice problem was discussed in section 5.2, where we explained that we intend to design our logics such that they avoid the problem. Property 5 shows that indeed our version of the reduction does enable free choice permission.

Ross' anomaly concerns the problem of how to avoid the undesirable property $O_{\odot}(\alpha) \rightarrow O_{\odot}(\alpha \cup \beta)$, that is often instantiated with the sentence 'if I am obliged to post the letter, I am obliged to post or burn it'. We already discussed this is in section 5.2.1. The reduction of section 5.2.1 avoids it by imposing the 'ought implies may' principle. Let us give an example of how application of the 'ought implies may' principle achieves avoidance of Ross' property. We take the instantiation: $O_{\odot}(\mathit{listen}) \rightarrow O_{\odot}(\mathit{listen} \cup \mathit{leave})$. This is intuitively undesirable, and is not valid according to the reduction. With the background information that $[\mathit{listen} \cap \mathit{leave}]\bot$ (it is not possible to listen and leave simultaneously), it follows that $O_{\odot}(\mathit{listen})$ implies $F_{\odot}(\mathit{leave})$. And if $F_{\odot}(\mathit{leave})$ and $O_{\odot}(\mathit{listen} \cup \mathit{leave})$ can hold at the same time, we would violate the 'ought implies may' principle $O_{\odot}(\alpha) \rightarrow P_{\odot}(\alpha)$. Actually, among the members of the above list of properties, there is not one that characterizes a logic relation between between $O_{\odot}(\alpha \cup \beta)$ and the components $O_{\odot}(\alpha)$ and $O_{\odot}(\beta)$. So not only Ross' property is avoided, also the reverse implication is.

Finally we mention some deontic properties of sequence, that reflect the goal orientedness of the norms.

Proposition 5.2.3

$$
\begin{array}{ll}
(10) & P(\alpha;\beta) \leftrightarrow [\alpha]P(\beta) \\
(11) & F(\alpha;\beta) \leftrightarrow \langle\alpha\rangle F(\beta) \\
(12) & O(\alpha;\beta) \leftrightarrow [\alpha]O(\beta)
\end{array}
$$

Property 12 holds due to the implementation of the ought implies may principle in the definition $O(\alpha) \equiv_{def} P(\alpha) \wedge [\wr^I \alpha]V_O$. If we would have defined $O(\alpha) \equiv_{def} [\wr^I \alpha]V_O$ we would not have property 12, but for the two strongest $\wr^I$-logics we defined in section 2.5.3, due to the property NegSeq-R: $\langle\alpha\rangle[\wr^I\beta]\varphi \rightarrow [\wr^I(\alpha;\beta)]\varphi$, we would have $\langle\alpha\rangle O(\beta) \rightarrow O(\alpha;\beta)$.

Of course, the above selection of deontic properties is not necessarily complete in the sense that together they capture the intended deontic reasoning. But that they do not need to, since we have a reduction to $\wr^I$-logics. All of the properties are just relativized negation modal action logic properties in disguise: the deontic reasoning can be carried out completely in these modal action logics.

5.2.3 Contrary to duty goal norms

A contrary to duty norm (CTD-norm, for short) with respect to some primary norm N is a norm that expresses additional normative information for the cases where N is violated. The most problematic anomalies of deontic logic concern CTD-norms [46, 112, 63]. Before we discuss one of the examples discussed in the literature, we explain the temporal interpretation of CTD-norms that naturally arises in the modal action logic context. This temporal view will be our dominant view on CTD-norms.

In the view on norms as embodied by the reduction in section 5.2.1, violating a norm corresponds to attending a violation state. The temporal view on CTD-norms is that they stipulate what norms are valid in such violation states. This gives a clear and unproblematic conception of CTD-norms. The interpretation of actions as binary relations over states naturally leads us to distinguish between the normative conditions that hold before and after execution of an action. Normative conditions that are in force before execution, are only violated in the state that results after execution. Therefore, in the temporal view, CTD-norms, and the norms that they are the CTD of (the primary norms), are never in force in the same state. A temporal solution to the famous anomalies of CTD-reasoning involves the addition of a temporal dimension to the reasoning domain and the recognition that the examples can be formalized such that the situation where the primary norm is in force and the situation where the CTD-norm is in force can be temporally separated. In the temporal view, contrary to duty norms typically concern some compensating action: an action that 'makes up' for the violation. Take the following example. I am obliged to pay a debt. If I do not pay the debt, I am obliged to pay the debt plus a penalty. In a formula: $O_{\odot}(\mathit{pay\text{-}debt}) \wedge [\wr^I \mathit{pay\text{-}debt}](O_{\odot}(\mathit{pay\text{-}debt}) \wedge O_{\odot}(\mathit{pay\text{-}penalty}))$. From property 9 of section 5.2.2, it follows that this is equivalent with $O_{\odot}(\mathit{pay\text{-}debt}) \wedge [\wr^I \mathit{pay\text{-}debt}]O_{\odot}(\mathit{pay\text{-}debt} \cap \mathit{pay\text{-}penalty})$, which means that it is obliged to pay debt plus penalty simultaneously, in other words: the debt is raised with a penalty. An example from the ought-to-be deontic literature that describes a situation where a temporal solution is ap-

propriate is the 'good Samaritan problem [112]: one ought not to be robbed, but if one is robbed one ought to be helped.

In the deontic logic literature, only those CTD-norms are studied for which the primary norm is an obligation. But in most of these studies prohibition and permission are definable in terms of obligations. Since we have separated violations concerning obligations, prohibitions and permissions, we can actually distinguish three different types of CTD: one for each deontic modality. The above example concerning the payment of debts gives a CTD-norm with respect to obligation. CTD-norms with respect to prohibitions $F_{\odot}(\alpha)$ are more problematic. The point is that for the temporal interpretation of such CTD-norms, we need to able to refer to the states where the violation V_F occurs, in order to specify what CTD-norm holds in these states. But the violation states for a norm $F_{\odot}(\alpha)$, in the weak interpretation we gave it, can be any of the states reached by α: we do not know for which of these states V_F holds and for which not. The CTD with respect to permissions $P_{\odot}(\alpha)$ is equally problematic. We may assume that probably in some of the states not reachable through α the violation V_P occurs, but we cannot say in general which states these are.

In the deontic logic literature, examples have been formulated where temporalization of the reasoning domain is claimed not to disentangle the problems of CTD-reasoning. One of these examples is the Chisholm anomaly [46], that reverses the temporal order: the primary norm violation is thought to occur after the CTD-norm situation. The following CTD-norm appears in Chisholm's example: 'It ought to be that Jones goes to the aid of his neighbors, but if Jones does not go to the aid of his neighbors, then he ought not to tell them he is coming'. The obligation to help is the primary obligation, the obligation not to tell is the CTD-norm. We have the impression that the sentence is a somewhat forced attempt to place the violation condition temporally after the coming into force of the CTD-norm. And in our opinion, it does not succeed in doing so: it makes sense to say that Jones' CTD-obligation not to tell actually refers to the moment at which he decides not to come. Then, this moment of decision coincides with the point of violation and with the point of coming into force of the CTD. The reasoning (that corresponds with concluding to a logically rational decision) thus has to deal with all information on a 'temporally equivalent' basis, that is, we can no longer view the primary norm and the CTD-norm as referring to separate points in time: both apply to the moment of decision.

The view that primary norms and their CTD-norms are not temporally separated, but refer to the same state / moment, is dominant for deontic action

logics that represent actions by propositions or unary act predicates. For such logics we thus have no a priori distinction between execution and result states of actions. This may be the reason that in standard possible worlds semantics for ought-to-do deontic logics of this type, such as for standard deontic logic (SDL) [189, 5, 93], a norm and its CTD-norm are usually thought to be in force in the same state. This semantically puzzling situation has been approached in various ways. A natural approach is to associate CTD-norms with preferences. One ought to satisfy φ, but if one does not (decides not to), one should satisfy ψ. Obedience of the first norm is preferred, but obedience of the second is second best. This leads to concepts such as 'sub-optimal worlds (the best worlds among the bad worlds)'. A famous example designed to enforce time-simultaneity of CTD-norms is Forrester's 'gentle murder' example [63]. The gentle murderer anomaly arises in the modeling of the following sentences.

1. It is forbidden for Smith to kill his mother.
2. If Smith kills his mother, he ought to kill her gently.
3. Smith kills his mother.

Sentence 1 expresses the primary norm, and sentence 2 the CTD-norm. A temporal solution is not appropriate, since both the primary norm and the CTD-norm apply to the same moment: the moment of killing. Since the example is formulated in terms of ought-to-do norms, we can formulate it in deontic modal action logic. Surprisingly, the property that is easiest to express in deontic logics of the ought-to-be type, is the hardest to express in modal action logic, namely, the simple fact that Smith kills his mother. This is a fact about the occurrence of an action. In modal action logics we can only talk about possibility (enabling) of action. Then, the following is as close as we can get: the kill action is possible, and other actions are not. We formalize the information as follows:

1. It is forbidden for Smith to kill his mother: $F_{\odot}(\mathit{kill})$
2. If Smith kills, he is obliged to perform it in such a way that it is a gentle kill: $O_{\odot}(\mathit{kill} \subseteq^{I} \mathit{kill\text{-}gentle})$
3. It is possible for Smith to kill his mother, and it is not possible for Smith not to kill his mother: $\langle \mathit{kill} \rangle \top \wedge [\wr^{I} \mathit{kill}] \bot$
4. Background information: gently killing is a way of killing. In other words: an action cannot simultaneously be a gentle kill and not a kill. In our formalization: $[\mathit{kill\text{-}gentle} \cap \wr^{I} \mathit{kill}] \bot$

It is not hard to assess that the formulas are consistent. The (non-deterministic) kill action is possible, and at least one of the ways to kill leads to a violation. At the same time, all ways of killing that are not gentle, also lead to a violation. The 'ought implies may' principle does not destroy consistency: it just imposes that all states that do not violate the obligation are also permitted. We thus have arrived at a consistent, non-temporalized interpretation of the CTD-example in deontic modal action logic. A possible objection against this formalization of the example is that it is only consistent due to the weakness of the prohibition. The free choice prohibition $F_{\odot}(kill)$ only says that there are ways of killing that are forbidden. Meyer [134] gives an alternative consistent formalization of this puzzle in deontic modal action logic.

Castañeda [41, 42, 43] has argued that many anomalies of CTD-reasoning are not inherited by deontic logics distinguishing actions (he calls them 'practitions') from act propositions (conditions), such as in the modal action setting. However, Hilpinen [94] showed that the gentle murderer reasoning example is not solved by Castañeda's proposal. Hilpinen discusses that Castañeda's approach in particular fails to model the 'gentle mode' of the killing. Above we showed that in our modal action logic setting there is a natural and consistent interpretation of the Chisholm example. In our formalism, 'gentle' is just a certain way of killing: there are many ways in which to kill, which is expressed by the non-determinism of the action. One of the ways of killing is killing gently.

5.3 Process norms

In this section we consider the class of action 'process' norms. We denote these norms with the subscript '$\rightsquigarrow$'. As said in the introduction, for process norms we take the position that the formulas $P_{\rightsquigarrow}(a;b) \wedge \neg P_{\rightsquigarrow}(a)$, $F_{\rightsquigarrow}(a) \wedge \neg F_{\rightsquigarrow}(a;b)$, and $O_{\rightsquigarrow}(a;b) \wedge \neg O_{\rightsquigarrow}(a)$ are all inconsistent. So, a permission to perform a sequential compound action requires permissions for all sub-actions. In general we can say that for this type of norms violations may occur at any point during the process of action performance. In this section we show that with formulas of μ-calculi we can enforce the right (non)-violation conditions along the execution traces of actions. We define reductions of formulas $P_{\rightsquigarrow}(\alpha)$, and $F_{\rightsquigarrow}(\alpha)$, and $O_{\rightsquigarrow}(\alpha)$, with α a 'regular' action, to formulas of μ-calculi.

But let us first give some preliminary intuitions concerning process oriented deontic modalities over sequence (;) and (free) choice ($\cup$), starting with 'Prohibition'. If a doctor forbids the sequence *eat*; *sport*, prohibition of *eat*; *sport*; *sleep* should be implied, but not necessarily prohibition of *eat*. Also, being forbidden

the free choice over *steal*∪*buy* (where we assume that the agent or system subject to this norm has full control over the choice denoted by ∪, as discussed in section 5.1) should imply being forbidden the free choice over *steal*∪*buy*∪*ask*, but not necessarily being forbidden the action *buy*. Intuitions for prohibition of iteration (*) over an action a follow directly from the intuitions for sequence and choice, since a^* stands for the choice over all possible finite sequences of actions a. Generalizing, we may say that (1) forbidden actions can never serve as constituent parts of actions that are not forbidden, and that (2) constituent actions of forbidden compound actions are not necessarily themselves forbidden. These intuitions comply with the following formulas:

$$F_{\rightsquigarrow}(a;b) \leftrightarrow F_{\rightsquigarrow}(a) \vee \langle a \rangle F_{\rightsquigarrow}(b)$$
$$F_{\rightsquigarrow}(a \cup b) \leftrightarrow F_{\rightsquigarrow}(a) \vee F_{\rightsquigarrow}(b)$$
$$F_{\rightsquigarrow}(a^*) \leftrightarrow \langle a^* \rangle F_{\rightsquigarrow}(a)$$

An intuition for the permission of sequence is given by a possible regulation for crossing a border to some country saying that it is permitted to perform the sequence *cross-border*; *buy-liquor*. This regulation should imply permission of *cross-border*, but not necessarily permission of *cross-border*; *buy-liquor*; *cross-border*, because this comes down to importing liquor, which in this example is assumed to be forbidden. For choice we have the intuition that being permitted the choice *eat*∪*drink* should imply being permitted the action *eat*, but not being permitted the choice *eat*∪*drink*∪*smoke*. In general we can say that any action that is a constituent of a permitted compound action should itself be permitted, and that permission of a constituent action is not sufficient to guarantee permission of a compound action. Deontic modal action logic formulas reflecting these intuitions are:

$$P_{\rightsquigarrow}(a;b) \leftrightarrow P_{\rightsquigarrow}(a) \wedge [a] P_{\rightsquigarrow}(b)$$
$$P_{\rightsquigarrow}(a \cup b) \leftrightarrow P_{\rightsquigarrow}(a) \wedge P_{\rightsquigarrow}(b)$$
$$P_{\rightsquigarrow}(a^*) \leftrightarrow [a^*] P_{\rightsquigarrow}(a)$$

Note that if we define $P_{\rightsquigarrow}(\alpha) = \neg F_{\rightsquigarrow}(\alpha)$, the above formulas for permission are equivalent with those for prohibition. For obligation, intuitions are more complicated. For the obligation of sequence the same intuitions as for permission apply: any action that is a sub-action of an obliged sequence of actions should itself be obliged. For instance, if we are obliged to perform $LookLeft; LookRight; LookLeft; CrossStreet$ then first we are obliged to perform $LookLeft$, and when we have performed $LookLeft$, we are obliged

to perform $LookRight; LookLeft; CrossStreet$, etc. But for the obligation of choice, no intuitive reduction seems possible. Take as an example a norm for students in class: being obliged to *listen* or *leave*, denoted $O_{\rightsquigarrow}(listen \cup leave)$. This obligation cannot be identified with $O_{\rightsquigarrow}(listen) \wedge O_{\rightsquigarrow}(leave)$, because intuitively this says that there is an obligation to do both actions, which contradicts the interpretation that we can obey $O_{\rightsquigarrow}(listen \cup leave)$ by performing either the action *listen* or the action *leave*. But also $O_{\rightsquigarrow}(listen) \vee O_{\rightsquigarrow}(leave)$ is no alternative, since this formula is implied by $O_{\rightsquigarrow}(listen)$ in propositional logic. This means that we would have to accept $O_{\rightsquigarrow}(a) \rightarrow O_{\rightsquigarrow}(a \cup b)$ as a validity, which is Ross' anomaly [159] again.

Since the iteration operator also encompasses choice, the irreducibility of obligation of choice also affects obligation of iteration. This means that for obligation we only have the intuition concerning sequence:

$$O_{\rightsquigarrow}(a; b) \leftrightarrow O_{\rightsquigarrow}(a) \wedge [a]O_{\rightsquigarrow}(b)$$

Thus, obligation is a difficult case. But it turns out that the irreducibility of obligation of choice occurs only on the level of atomic action. In the full process-type deontic logic of regular action we define, obligations concerning choice between compound actions are reduced to obligations over choice between atomic actions. A preliminary example of this reduction is given by the formula $O_{\rightsquigarrow}((a; b) \cup (c; d)) \leftrightarrow (O_{\rightsquigarrow}(a \cup c) \wedge [a]O_{\rightsquigarrow}(b) \wedge [c]O_{\rightsquigarrow}(d))$, where the obligation over the choice between the two compound actions $a; b$ and $c; d$ is expressed in terms of the obligation over the choice between the atomic actions a and c[5].

In section 5.3.1 we give semantic characterizations of deontic modalities over regular actions in terms of conditions on modal action models. The conditions on models are defined in terms of deontic primitives on choices over atomic actions. In section 5.3.2, we define μ^a-calculus characterizations of the semantic conditions. In section 5.3.3 we study the compositionality of the defined notions in the regular action combinators sequence, choice and iteration. It will turn out that permission and prohibition are compositional, and that obligation is not compositional in these action combinators. In section 5.3.4 we show how to go from the μ^a-calculus (the fragment of the μ^η-calculus defined in section 3.5 where actions η are atomic) characterization over atomic choice actions to reductions that completely translate deontic formulas to (1)

[5]We assume here that $a \neq c$. For situations where a and c are the same action, our definitions give $O_{\rightsquigarrow}((e; b) \cup (e; d)) \leftrightarrow (O_{\rightsquigarrow}(e) \wedge [e]O_{\rightsquigarrow}(b \cup d))$.

the μ^m-calculus of Bradfield and Stirling, and (2) the μ^η-calculus of section 3.5. In section 5.3.5 we discuss contrary to duty process norms.

5.3.1 Semantic conditions and free process choice

In this section we work towards semantic characterizations of the deontic notions $P_{\rightsquigarrow}(\alpha)$, $F_{\rightsquigarrow}(\alpha)$ and $O_{\rightsquigarrow}(\alpha)$ for regular actions α in terms of conditions on modal action models. In subsequent sections we capture these conditions with the help of μ^a-calculus expressions. Regular actions encompass choice $\cup$, sequence ; and iteration *, which are the combinators of PDL (section 2.3), without the converse and the test.

Definition 5.3.1 (regular action) *Taking 'a' to represent arbitrary elements of a given set of atomic action symbols $\mathcal{A}$, the syntax of regular actions is defined by the following BNF:*

$$\alpha, \beta, \ldots \quad := \quad a \mid \mathit{skip} \mid \alpha \cup \beta \mid \alpha; \beta \mid \alpha^*$$

We interpret a regular action by a (possibly infinite) set of action traces, which are finite concatenations of atomic actions and states. Atomic actions a constitute elementary action traces in themselves, and *skip* refers the 'point' trace consisting of one individual state. The action combinator $\cup$ is associated with union of action traces, ; with concatenation, and * with union of all finitely repeated self-concatenations, respectively. The trace interpretation of regular action is a specialization of the notion of 'action graph' as defined in section 2.4.3 in definition 2.4.7: action traces are thus action graphs for actions that do not encompass the concurrency, action complement, and converse combinators. We use the notation introduced in section 2.4.3, i.e. we denote that a trace (action graph) G interprets α by $G \in \Gamma(\alpha)$. However, in this chapter we use $\langle a_1, a_2, \ldots, a_n \rangle$ as a shorthand for the action trace (graph) $\Theta_{a_1} \circ \Theta_{a_2} \circ \ldots \circ \Theta_{a_n}$.

We saw in section 5.2.1 how to incorporate the deontic realm in modal action models with the help of violation propositions, in the style of Meyer [135]: if V is a violation proposition, $\langle \alpha \rangle V$ represents that α is (weakly) forbidden ($F(\alpha)$) because it may lead to a violation. Van der Meyden [132] assigns violation labels to action traces instead of to postconditions: if $\langle a \rangle \top$, and there is an action trace for α that is marked as a violation trace, we have that α is (weakly) forbidden. On a modal action logic for which validity is preserved under standard bisimulation, these two views are equivalent, as was argued in section 1.6.1 of the introductory chapter. There is however a straightforward third way to introduce the deontic realm in modal action models. For all

atomic actions $a \in \mathcal{A}$, we can interpret atomic propositions $F(a)$ (it is forbidden to perform a) in states of models. This approach seems the one that stays closest to the objective of the logic, namely, to model the entailment relations between normative expressions over complex (regular) actions and normative expressions over more elementary and, in particular, atomic actions.

However, we do not define models that valuate atomic propositions $F(a)$. We start by investigating the deontic logic over a subset of complex actions, namely, actions that are more complex than atomic choice actions of the form $a_1 \cup a_2 \cup \ldots \cup a_n$. We show that normative expressions over such complex actions can be reduced to normative expressions over atomic choice actions. The logic that defines the relation between normative expressions over atomic choice actions and atomic actions, is postponed to section 5.3.4. To enable a semantic, model based investigation of the logic over actions that are more complex than atomic choice actions, we define models that include interpretations of all deontic primitives of the form $P_{\rightsquigarrow}(a_1 \cup a_2 \cup \ldots \cup a_n)$, $F_{\rightsquigarrow}(a_1 \cup a_2 \cup \ldots \cup a_n)$ and $O_{\rightsquigarrow}(a_1 \cup a_2 \cup \ldots \cup a_n)$ (it is permitted / prohibited /obliged to perform the choice of atomic actions $a_1 \cup a_2 \cup \ldots \cup a_n$). Since we postpone the logic for atomic choice actions to section 5.3.4, these interpretations say nothing about the logic relation between for instance $P_{\rightsquigarrow}(a \cup b)$ and $P_{\rightsquigarrow}(a)$: it may for instance be true in a state of a model that $P_{\rightsquigarrow}(a \cup b) \wedge \neg P_{\rightsquigarrow}(a)$, while in other states $P_{\rightsquigarrow}(a \cup b) \wedge P_{\rightsquigarrow}(a)$.

The inclusion of separate primitives for each deontic modality shows that, as in section 5.2.1, we are cautious, and do not a priori impose interdefinabilities for permission, prohibition and obligation. We concentrate on the logic properties of the individual modalities, and show that for each separate modality we can reduce the normative properties of complex actions to normative properties for atomic choice actions. This means that also the interactions between the modalities depend on the interactions at the level of atomic choice actions. So also the interactions are postponed to section 5.3.4 where we discuss possible logics for atomic choice actions.

Definition 5.3.2 (normative choice models) *Given a countable set $\mathcal{A}$ of atomic action symbols and a countable set $\mathcal{P}$ of atomic proposition symbols, a normative choice model $\mathcal{M} = (S, R^{\mathcal{A}}, P^{\mathcal{A}}, F^{\mathcal{A}}, O^{\mathcal{A}}, V^{\mathcal{P}})$ over $\mathcal{A}$ and $\mathcal{P}$ is defined as follows:*

- *S is a non-empty set of possible states.*
- *$R^{\mathcal{A}}$ is an action interpretation function $R^{\mathcal{A}} : \mathcal{A} \to 2^{(S \times S)}$, assigning a binary relation over $S \times S$ to each atomic action a in $\mathcal{A}$.*

- *$P^{\mathcal{A}}$ is an interpretation function $P^{\mathcal{A}} : 2^{\mathcal{A}} \to 2^S$, assigning to each set of actions $\{a_1, a_2, \ldots, a_n\}$ the subset of states in S for which $P_{\rightsquigarrow}(a_1 \cup a_2 \cup \ldots \cup a_n)$ is valid.*
- *$F^{\mathcal{A}}$ is an interpretation function $F^{\mathcal{A}} : 2^{\mathcal{A}} \to 2^S$, assigning to each set of actions $\{a_1, a_2, \ldots, a_n\}$ the subset of states in S for which $F_{\rightsquigarrow}(a_1 \cup a_2 \cup \ldots \cup a_n)$ is valid.*
- *$O^{\mathcal{A}}$ is an interpretation function $O^{\mathcal{A}} : 2^{\mathcal{A}} \to 2^S$, assigning to each set of actions $\{a_1, a_2, \ldots, a_n\}$ the subset of states in S for which $O_{\rightsquigarrow}(a_1 \cup a_2 \cup \ldots \cup a_n)$ is valid.*
- *$V^{\mathcal{P}}$ is an interpretation function $V^{\mathcal{P}} : \mathcal{P} \to 2^S$ assigning to each proposition P of $\mathcal{P}$ the subset of states in S for which P is valid.*

Having defined regular actions together with their semantics and modal action models, we can formulate our central question(s) more accurately: are we able to give an intuitive meaning to the process norms $P_{\rightsquigarrow}(\alpha)$, $F_{\rightsquigarrow}(\alpha)$ and $O_{\rightsquigarrow}(\alpha)$ for general regular actions α, in terms of the primitive interpretations of $P_{\rightsquigarrow}(a_1 \cup a_2 \cup \ldots \cup a_n)$ and $O_{\rightsquigarrow}(a_1 \cup a_2 \cup \ldots \cup a_n)$ for atomic choice actions $a_1 \cup a_2 \cup \ldots \cup a_n$ using some form of modal action logic, and if so, what are the implications for the logical relations between (1) $P_{\rightsquigarrow}(\alpha)$, $F_{\rightsquigarrow}(\alpha)$ and $O_{\rightsquigarrow}(\alpha)$ and corresponding deontic properties of constituting parts of α, and (2) $P_{\rightsquigarrow}(\alpha)$, $F_{\rightsquigarrow}(\alpha)$ and $O_{\rightsquigarrow}(\alpha)$ mutually? In the next two subsections, we focus on the possible meaning of $P_{\rightsquigarrow}(\alpha)$, $F_{\rightsquigarrow}(\alpha)$ and $O_{\rightsquigarrow}(\alpha)$ in terms of conditions on trajectories through the models of definition 5.3.2.

Free process choice semantics for permission and prohibition

We start by formulating a base-intuition concerning the free choice semantics for the notions of permission and prohibition of regular action.

$P_{\rightsquigarrow}(\alpha) \equiv$ it is permitted to perform any action trace that interprets α

$F_{\rightsquigarrow}(\alpha) \equiv$ there is an action trace that interprets α that is forbidden

Our starting point is thus that being permitted a (regular) action is being permitted *all* possibilities (all action traces in the action trace-set) to perform the action, which corresponds with the free-choice view on permission, as discussed in section 5.1. In order to define a formal semantics of $P_{\rightsquigarrow}(\alpha)$ and $F_{\rightsquigarrow}(\alpha)$, we have to capture this intuition in terms of a condition on modal action models.

Action traces for α have to be related to trajectories from a state s in a model $\mathcal{M}$ where the property $P_{\rightsquigarrow}(\alpha)$, respectively $F_{\rightsquigarrow}(\alpha)$ is thought to hold. Let us first formally define the distinction between action traces and trajectories. A trajectory through a model $\mathcal{M} = (S, R^{\mathcal{A}}, P^{\mathcal{A}}, F^{\mathcal{A}}, O^{\mathcal{A}}, V^{\mathcal{P}})$ is a series of states and actions: $\tau = \langle s_0, a_1, s_1, \ldots, s_{n-1}, a_n, s_n \rangle$, where possibly $s_i = s_j$ and/or $a_i = a_j$ for some i and j such that $i \neq j$, and where $\forall i$ such that $0 \leq i \leq n$ it holds that $s_i \in S$, and $\forall i$ such that $1 \leq i \leq n$ it holds that $(s_{i-1}, s_i) \in R^{\mathcal{A}}(a_i)$. Then, given a trajectory $\tau = \langle s_0, a_1, s_1, \ldots, s_{n-1}, a_n, s_n \rangle$ we denote the action trace $\langle a_1, a_2, \ldots, a_n \rangle$, i.e. the concatenation of all elementary a_i-graphs (remember that $\langle a_1, a_2, \ldots, a_n \rangle$ is a shorthand for $\Theta_{a_1} \circ \Theta_{a_2} \circ \ldots \circ \Theta_{a_n}$), by $G(\tau)$.

Using the notions of trajectory and action trace, it seems straightforward to reformulate the above intuition as a formal condition on models. To guarantee that all action traces of an action α are permitted, in a given state s_0 it seems sufficient to demand that when following a trajectory corresponding to an action trace for α, on all states of the trajectory, the next action a in the action trace is permitted, that is, if $P_{\rightsquigarrow}(a_1 \cup \ldots a_n)$ holds in that state, for some i it holds that $a_i = a$. But it turns out that this characterization is naive: we will have to be more precise about the semantics of $P_{\rightsquigarrow}(\alpha)$ and $F_{\rightsquigarrow}(\alpha)$, because the characterization leaves room for more than one interpretation, as can be seen from the following example models for $P_{\rightsquigarrow}(a; (b \cup c))$.

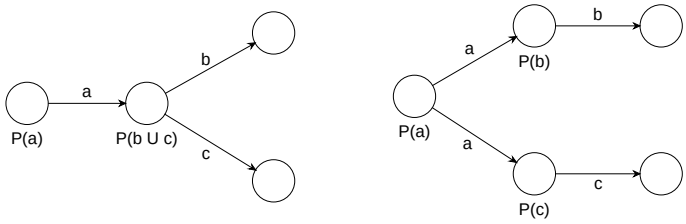


Fig 13. (1) free process choice (2) partially imposed process choice

Both models, in their left most state, obey the 'naive' semantic characterization if applied to $P_{\rightsquigarrow}(a; (b \cup c))$, since in both models, both action traces $\langle a, b \rangle$ and $\langle a, c \rangle$ for $a; (b \cup c)$ follow trajectories where on all states the next action in the trace is permitted. But from a modal point of view, the models are not equivalent since they do not bisimulate: model 1 satisfies $P_{\rightsquigarrow}(a) \wedge [a] P_{\rightsquigarrow}(b \cup c)$, and model 2 only satisfies the weaker $P_{\rightsquigarrow}(a) \wedge [a](P_{\rightsquigarrow}(b) \vee P_{\rightsquigarrow}(c))$. These two formulas correspond with two different views on how choice is dealt with during the process of action execution. If we think of the notion of permission as part of the logic governing an agent, we can say that model 1 corresponds to

the logic of an agent that reasons about a future course of events α where it has full control over all choices between actions during execution of α. The agent reasons 'if initially I am permitted to perform either ab or ac, then after I have done a, I can choose either b or c, which implies that after I have performed a, I am permitted both b and c'. This is exactly what is reflected by the formula $P_{\rightsquigarrow}(a) \wedge [a](P_{\rightsquigarrow}(b \cup c))$. We refer to this interpretation of permission as the 'free process choice' semantics.

Model 2 corresponds to the logic of an agent that reasons about a future course of events where it has only partial control over choice. During execution it has no control over choice: after it has performed a, it is only permitted (say, by its environment) to perform either b or c. But at the start, the agent is free to choose, since it is permitted both traces ab and ac. The agent may reason 'initially I am permitted to perform either ab or ac, but I am not permitted to choose between b and c *after* I have done a, which means that after I have performed an a, I will either be permitted b or c'. This is exactly what is reflected by the formula $P_{\rightsquigarrow}(a) \wedge [a](P_{\rightsquigarrow}(b) \vee P_{\rightsquigarrow}(c))$. Another way of looking at this mixed free initial choice / partially imposed process choice situation is to say that under this semantics for $P_{\rightsquigarrow}(a;(b \cup c))$, the agent has to make up his mind about future choices in advance: it is permitted both ab or ac, but before it performs an a, it has to choose what it wants to do afterwards, because it is not allowed a choice at 'runtime', in particular, once it has performed the a. Under the free process choice interpretation (free at the start and free during the rest of the execution) of $P_{\rightsquigarrow}(a;(b \cup c))$ it is allowed to change its mind at 'runtime', in particular, after the a has been performed.

In model 2 we thus have a mixture of free and imposed choice: free choice at the start, and imposed choice for all subsequent actions. We can also imagine a situation where during the complete process, including the start, the agent is deprived of choice. When we interpret choice according to this view, we speak of an 'imposed choice semantics'. The recognition of the difference between free process choice and imposed process choice thus generalizes the notions of free choice and imposed choice as discussed in section 5.1 to the process case. The need for such a generalization does not come as a surprise: for process norms, violations may occur on all occasions during execution, which means that the distinction between free and imposed choice also applies to all possibilities of choice during execution.

Now, if we want to define a deontic process logic, we will have to decide which semantics we pursue: free process choice or imposed process choice. In other work on process norms in a modal logic setting [130, 132] one seems not to be aware of this distinction, but the choice implicitly made is that in

favor of free process choice. Here we also pursue this semantics. However, the imposed process choice semantics, in its most stringent form, where the deprivation of choice occurs at all stages of action execution, is characterizable by a simple adaptation of the μ^a-calculus constructions we give.

It is also conceivable, as in model 2 of the previous example, that the deprivation of runtime choice is only partial. But, the separation of free choice and imposed choice points need not be as clear as in model 2. An agent might be deprived of choice only at certain specific points in the future course of events described by a regular action α. We can easily give examples of more extensive models in which this is the case. In such models, each point where a non-deterministic atomic action appears represents a point where a choice concerning subsequent actions is forced upon an agent from the outside (in model 2 of the previous example, we have that a is non-deterministic in the initial state, which means that a choice concerning the subsequent actions b and c is forced upon the agent after a has been executed). Each of these models would have a separate modal action logic formula characterizing it. But we do not define such a logic. A logic that enables the expression of 'partially imposed process choice', would have to be based on an action language that is more expressive than the language of regular action. Such an action language would have to represent and distinguish both situations of figure 13, for instance by relying on trees instead of action traces as interpretations for action symbols.

The example shows that to reflect the free process choice intuition in terms of conditions on models, it is not enough to simply demand that on all points on trajectories, the right primitive normative condition holds for the next action of the action trace. Adoption of a free process choice semantics means that in any state s_i of a trajectory τ we are not only permitted to do the next action a_{i+1} in the action trace $G(\tau)$ for α (if we follow such an action trace); we are also permitted to do any action being a next action in any other action trace G for α that shares an identical i-prefix with the action trace $G(\tau)$.

Definition 5.3.3 (free process choice permission and prohibition) *The semantic characterization of the notions $P_{\rightsquigarrow}(\alpha)$ and $F_{\rightsquigarrow}(\alpha)$, both in words and as a formal condition on models $\mathcal{M} = (S, R^{\mathcal{A}}, P^{\mathcal{A}}, F^{\mathcal{A}}, O^{\mathcal{A}}, V^{\mathcal{P}})$, is defined as:*

> *$P_{\rightsquigarrow}(\alpha)$ holds in a state s iff on any state s_i on trajectories from s that follow an action trace for α, it holds that $P_{\rightsquigarrow}(b_1 \cup b_2 \cup \ldots \cup b_n)$ for the set of actions $B = \{b_1, b_2, \ldots b_n\}$ that keeps us 'within' the action trace-set for α*

$\mathcal{M}, s_0 \models P_{\rightsquigarrow}(\alpha)$ *iff for all* $\tau = \langle s_0, a_1, s_1, \ldots, s_{i-1}, a_i, s_i\rangle$ *through* $\mathcal{M}$*, it holds that if the set* $B_{i+1} = \{a_{i+1} \mid \langle a_1, \ldots, a_i, a_{i+1}, \ldots, a_n\rangle \in \Gamma(\alpha)\}$ *is non-empty, then* $s_i \in P^{\mathcal{A}}(B_{i+1})$

$F_{\rightsquigarrow}(\alpha)$ *holds in a state* s *iff there is a state* s_i *on a trajectory from* s *that follows an action trace for* α*, for which it holds that* $F_{\rightsquigarrow}(b_1 \cup b_2 \cup \ldots \cup b_n)$ *for the set of actions* $B = \{b_1, b_2, \ldots b_n\}$ *while performance of one of the actions* b_i *keeps us 'within' the action trace-set for* α

$\mathcal{M}, s_0 \models F_{\rightsquigarrow}(\alpha)$ *iff there is a* $\tau = \langle s_0, a_1, s_1, \ldots, s_{i-1}, a_i, s_i\rangle$ *through* $\mathcal{M}$*, such that the set* $B_{i+1} = \{a_{i+1} \mid \langle a_1, \ldots, a_i, a_{i+1}, \ldots, a_n\rangle \in \Gamma(\alpha)\}$ *is non-empty and* $s_i \in F^{\mathcal{A}}(B_{i+1})$

The reader may check that according to this semantic definition of $P_{\rightsquigarrow}(\alpha)$ the formula $P_{\rightsquigarrow}(a; (b \cup c))$ holds in the left-most state of model 1 of the example, but not in model 2.

Free process choice semantics for obligation

For obligation, the base-intuition can be formulated as:

$O_{\rightsquigarrow}(\alpha) \equiv$ it is obliged to perform an action trace for α

But as for permission (and its counterpart prohibition) we have to become more precise. Being obliged α can be interpreted under free process choice or under imposed process choice. In the first, weaker case, an agent reasons about a situation where it has committed itself to perform an action trace interpreting α, and where it has free process choice, or in other words, can 'change traces' during runtime. In the second case it reasons about a future course of events under the condition that it has to choose and commit itself to one particular trace from the beginning, which points to a considerably stronger obligation. Note that compared to the situation for permission, the role of the terms 'strong' and 'weak' is interchanged. Except for this shift in perspective, the discussion on free/imposed process choice is analogous to that for permission. Therefore we immediately turn to a precise characterization of free process choice obligation, in words, and in more formal terms.

Definition 5.3.4 (the semantics of free process choice obligation) *The semantic characterization of the notion* $O_{\rightsquigarrow}(\alpha)$*, both in words and as a formal condition on models* $\mathcal{M} = (S, R^{\mathcal{A}}, P^{\mathcal{A}}, F^{\mathcal{A}}, O^{\mathcal{A}}, V^{\mathcal{P}})$*, is defined as:*

$O_{\rightsquigarrow}(\alpha)$ holds in a state s iff on any state s_i on trajectories from s that follow an action trace for α, that do not correspond with a point where an action trace for α has been executed completely, it holds that $O_{\rightsquigarrow}(b_1 \cup b_2 \cup \ldots \cup b_n)$ for the set of actions $B = \{b_1, b_2, \ldots b_n\}$ that keeps us 'within' the action trace-set for α

$\mathcal{M}, s_0 \models O_{\rightsquigarrow}(\alpha)$ iff for all $\tau = \langle s_0, a_1, s_1, \ldots, s_{i-1}, a_i, s_i \rangle$ through $\mathcal{M}$ for which $\langle a_1, \ldots, a_i \rangle \notin \Gamma(\alpha)$, it holds that if the set $B_{i+1} = \{a_{i+1} \mid \langle a_1, \ldots, a_i, a_{i+1}, \ldots, a_n \rangle \in \Gamma(\alpha)\}$ is non-empty, then $s_i \in O^{\mathcal{A}}(B_{i+1})$

Note that this definition is very close to the one for permission. The only structural difference is the condition concerning the complete execution of an action trace. The characterization says that the condition $O_{\rightsquigarrow}(b_1 \cup b_2 \cup \ldots \cup b_n)$ is not imposed on all states of trajectories through the model that correspond to an action α, but only on states that are not the final state of the execution of an action trace interpreting α. This marks an intrinsic difference between the process versions of the notions of permission and obligation. The difference is exemplified by the meaning of the two expressions $P_{\rightsquigarrow}(b; a^*)$ and $O_{\rightsquigarrow}(b; a^*)$. The trace-set $\Gamma(b; a^*)$ for the regular action $b; a^*$ is $\{\langle b \rangle, \langle b, a \rangle, \langle b, a, a \rangle, \langle b, a, a, a \rangle, \ldots\}$. The semantic characterization for permission says that in states of trajectories that result from the execution of the final action of such traces we are permitted to perform the action a, since it keeps us within the trace-set. Intuitively this is what should hold, since we are permitted any of the traces. But, for obligation a similar condition should not hold: if in states of trajectories that result from the execution of the final action of such a trace we would be obliged to do an action that keeps us within the trace-set, we would be forced always to 'jump' to longer traces by doing an extra a. This obligation would force us to keep on performing a's forever, and we would thus be obliged to perform an a-loop. This does not correspond to the intuition that $O_{\rightsquigarrow}(b; a^*)$ stands for an obligation to perform one of the traces interpreting $b; a^*$ (with freedom to switch between traces for $b; a^*$ during execution). By demanding in definition 5.3.4 that only on pre-final states we are forbidden to do actions that bring us out of the trace-set, we allow for the possibility to do an 'escaping' action in the final state. That is exactly what is intended: we have fulfilled our obligation, since we have performed one of the action traces for α. And when we have fulfilled the obligation, we should not any longer be constrained in the performance of actions[6]. The

[6]Note that the fulfillment of an obligation does not 'cause' an 'obligation to *stop*'.

semantics of definition 5.3.4 implies that $O_{\rightsquigarrow}(a^*) = \top$ for any atomic action a, and $O_{\rightsquigarrow}(skip) = \top$, since we can already comply to these obligations by doing nothing, which is always a possibility. This makes these obligations void.

Now that we have come to the point that the deontic notions we want to formalize are characterized as semantic conditions on modal action models, we may concentrate on the question how to capture the defined notions with the help of modal action logic formulas. We have already seen, for the example property $P_{\rightsquigarrow}(a;(b \cup c))$, how this can be achieved as $P_{\rightsquigarrow}(a) \wedge [a](P_{\rightsquigarrow}(b \cup c))$. But for general regular actions that involve iteration, a simple modal action logic formula does not suffice.

5.3.2 μ^a-calculus characterizations through DFAs

We now turn to the μ^a-calculus characterizations of the deontic notions of section 5.3.1. These characterizations constitute a recursive composition from atomic deontic notions, governed by the structure of deterministic finite automatons (DFAs) of a regular action. The role of DFAs in the μ^a-calculus characterization is twofold. On the one hand, DFAs are used as a sort of action models that interpret μ^a-calculus formulas in their starting state, and on the other hand DFAs are used to define a syntactical, recursive relation with μ^a-calculus formulas. The μ^a-calculus is the fragment of the μ^η-calculus defined in section 3.5 where actions η are atomic. Thus, from now on we assume that the μ^η-calculus syntax of definition 3.5.1 is restricted to atomic actions a, and is extended to include the primitives $P_{\rightsquigarrow}(a_1 \cup a_2 \cup \ldots \cup a_n)$, $F_{\rightsquigarrow}(a_1 \cup a_2 \cup \ldots \cup a_n)$ and $O_{\rightsquigarrow}(a_1 \cup a_2 \cup \ldots \cup a_n)$. The resulting language is what we call 'the μ^a-calculus'.

μ^a-calculus characterizations for permission and prohibition

The first step in the μ^a-calculus definition of deontic notions of regular action is to associate a regular action with a deterministic finite automaton (DFA) that describes the same set of action traces. The second step is to express $P_{\rightsquigarrow}(\alpha)$ and $F_{\rightsquigarrow}(\alpha)$ completely in terms of the primitive notion $P_{\rightsquigarrow}(a_1 \cup a_2 \cup \ldots \cup a_n)$ by building a μ^a-calculus formula based on the DFA. There is always more than one DFA for a given regular action, but we show that μ^a-calculus translations of different DFAs of the same regular action are logically equivalent. To get a first impression of this approach, two initial examples are given.

Example 5.3.1 *Consider the properties: $P_{\rightsquigarrow}(a^*;b)$ and $P_{\rightsquigarrow}((a;b^*)^*;c)$. DFAs of the regular actions $a^*;b$ and $(a;b^*)^*;c$ are:*

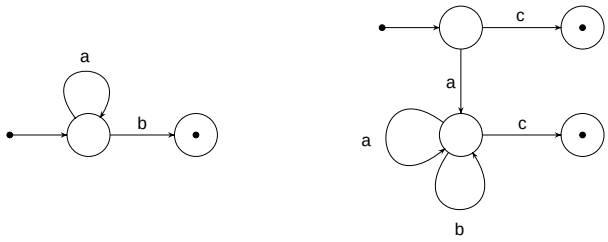


Fig 14. two example DFAs

The property $P_{\rightsquigarrow}(a^*;b)$ expressed in terms of $P_{\rightsquigarrow}(a)$ and $P_{\rightsquigarrow}(b)$ by means of a μ^a-calculus formula: $\nu Z.\ P_{\rightsquigarrow}(a \cup b) \wedge [a]Z$. In words: "the permission to perform the action $a^*;b$ equals the permission to initially choose between the atomic actions a and b, and if a is chosen to reach this same deontic state again".

The property $P_{\rightsquigarrow}((a;b^*)^*;c)$ expressed in terms of $P_{\rightsquigarrow}(a)$, $P_{\rightsquigarrow}(b)$ and $P_{\rightsquigarrow}(c)$ by means of a μ^a-calculus formula: $P_{\rightsquigarrow}(a \cup c) \wedge [a](\nu Z.\ P_{\rightsquigarrow}(a) \wedge P_{\rightsquigarrow}(b \cup c) \wedge [a]Z \wedge [b]Z)$[7]. In words: "the permission to perform the action $P_{\rightsquigarrow}((a;b^*)^*;c)$ equals the permission to initially choose between the atomic actions a and c, and if a is chosen, to reach a state where one is permitted to choose between the atomic actions a, b and c, and where if one chooses either an a or a b one reaches the same deontic state again".

The examples show that the μ^a-calculus formula expressing the deontic notion is related directly to the automaton. We now give the precise definition of how the formula is built. First the notions of 'grounded loop' and 'automaton return state' for DFAs are defined.

Definition 5.3.5 *A grounded loop of a deterministic finite automaton $DFA = (Q, A, N, q^i, T)$, with Q a set of automaton states, A a set of automaton actions, $N : Q \times A \to Q$ the transition function, $q^i \in Q$ the initial automaton state and $T \subset Q$ the set of terminal automaton states, is a sequence of edges $e_1, e_2, \ldots, e_n$ ($e_i \in E$ and $E \subset Q \times A \times Q$) such that:*

- *e_1 leaves the initial automaton state q^i*
- *each e_i leaves the automaton state that is entered by e_{i-1}*

[7]Note that in this example the regular action does not contain the choice operator $\cup$ but that in the μ^a-calculus translation to permissions over atomic actions, many non-deterministic choices appear. This is the non-determinism that is part of the semantics of the iteration.

- *there is an e_i with $i < n$ such that e_i and e_n enter the same automaton state*

- *there is no other pair of edges e_j and e_k that enter the same automaton state*

An automaton state that is entered by the final edge e_n of a grounded loop is called an 'automaton return state'. In the following, the set of automaton return states of a DFA is denoted by R.

Proposition 5.3.1 *Given a deterministic finite automaton, there are finitely many grounded loops, and each grounded loop involves a finite number of edges.*

Proof

From negative demonstration: an infinite number of grounded loops can only be realized with an infinite number of edges and thus with an infinite automaton, and a grounded loop of infinite length can only be realized with an infinite number of edges in combination with an infinite number of automaton states, since in a grounded loop we cannot visit an automaton state for the second time (the only exception is the automaton return state at the end of a loop, but there the loop ends).

■

In general there are many grounded loops, and many automaton return states of these loops coincide. In the procedure defined next, the procedure that builds a μ^a-calculus formula from a given deterministic finite state machine, each automaton state that is the automaton return state of one or more grounded loops is assigned a separate μ^a-calculus state variable.

Definition 5.3.6 *Let α be a regular action, and $U^\alpha = (Q, A, N, q^i, T)$ a corresponding deterministic finite automaton with Q a set of automaton states, A a set of automaton actions, $N : Q \times A \to Q$ the transition function, $q^i \in Q$ the initial automaton state, and $T \subset Q$ the set of terminal automaton states. Furthermore, let $R \subseteq Q$ be the set of automaton return states of the DFA, let Z_q be a μ^a-calculus state variable associated with an automaton return state $q \in R$, and let $Out(q)$ be the set of outgoing automaton actions of an automaton state $q \in Q$. Then a μ^a-calculus formula for $P_{\rightsquigarrow}(\alpha)$ is built with the help of the following recursive function f that associates a μ^a-calculus formula to each automaton state:*

$$
\begin{array}{llcl}
\textit{if } q \in Q \setminus R, & f(q) & = & P_{\rightsquigarrow}(\bigcup Out(q)) \wedge \bigwedge\limits_{a \in Out(q)} [a] f(N(q,a)) \\
\textit{if } q \in R, & f(q) & = & \nu Z_q.\ P_{\rightsquigarrow}(\bigcup Out(q)) \wedge \bigwedge\limits_{a \in Out(q)} [a] f(N(q,a)) \\
\textit{if } q \in R, & f'(q) & = & Z_q
\end{array}
$$

An automaton return state $q \in R$ has two associated formulas: $f(q)$ and $f'(q)$. The value $f(q)$ is used if in a thread of recursive calls of the function f, the automaton return state q is visited for the first time, and the value $f'(q)$ is used if in this same thread the automaton state is attended for the second time. If automaton states have no outgoing automaton actions, their associated formula is $\top$. The μ^a-calculus characterization of $P_{\rightsquigarrow}(\alpha)$ is defined as the formula associated to the initial automaton state of the automaton: $P_{\rightsquigarrow}(\alpha) = f(q^i)$.

The reader is invited to check that this recursive procedure, applied to the deontic notions of example 5.3.1, returns the μ^a-calculus formulas mentioned in the example. For prohibition $F_{\rightsquigarrow}(\alpha)$ we can define a separate recursive function with $\vee$ instead of $\wedge$, $\langle\ \rangle$ instead of $[\ ]$ and μ instead of ν. Because of this straightforward duality, we do not give an explicit definition of the μ^a-calculus reduction of prohibition.

Proposition 5.3.2 *The recursive function f in definition 5.3.6 always returns a finite, well-formed μ^a-calculus formula.*

Proof

Each 'thread' of recursive calls of the function f either follows a grounded loop through the finite automaton or ends in a terminal automaton state of the finite automaton. Since (1) a finite automaton contains only finitely many grounded loops, and (2) the loops of a finite automaton involve only a finite number of automaton edges, and (3) on second attendance of an automaton return state in the recursive evaluation of f no further function calls are invoked, the recursive calls eventually stop. The well-formedness follows directly from the well-formedness of the separate cases distinguished in the definition of the function f.

■

We now turn to the claim that this μ^a-calculus characterization captures the intuition of free process choice permission of section 5.3.1.

Theorem 5.3.3 *The μ^a-calculus characterization of definition 5.3.6 is correct and sufficient with respect to the semantics of $P_{\rightsquigarrow}(\alpha)$ and $F_{\rightsquigarrow}(\alpha)$ of definition 5.3.3.*

Sketch of a proof

If we denote the initial automaton state q^i of a deterministic finite automaton U^α corresponding to a regular action α by $q^i(U^\alpha)$, correctness says that in states of models where the semantic condition for $P_{\rightsquigarrow}(\alpha)$ of definition 5.3.3 holds, the formula $f(q^i(U^\alpha))$ is valid. We first consider a special DFA-based model for $P_{\rightsquigarrow}(\alpha)$ that satisfies the semantic conditions, and that by definition satisfies the formula $f(q^i(U^\alpha))$. The second step is to show that any other state of any other model satisfying the semantic conditions for $P_{\rightsquigarrow}(\alpha)$ is related to the initial state of the special DFA-model by a mapping that preserves validity of $f(q^i(U^\alpha))$. The DFA-model we consider is based on the DFA for α: automaton states Q are turned into modal action model states S, and the transition function N for automaton actions A is turned into an action interpretation function $R^{\mathcal{A}}$. Furthermore, we define that for the interpretation $P^{\mathcal{A}}$ it holds that $s \in P^{\mathcal{A}}(Out(q))$, where q is the automaton state associated to the modal action model state s. We first have to show that in its initial state (the state corresponding to the automaton state q^i) this model satisfies the semantic conditions for $P_{\rightsquigarrow}(\alpha)$. The semantic condition of definition 5.3.3 imposes primitive normative conditions only on trajectories that correspond to sub-traces of action traces that interpret α. Clearly, in the DFA-model, all trajectories correspond to action traces that are sub-traces of action traces interpreting α. The semantic condition is that on all states of trajectories, those actions that keep us within the trace set are permitted. In the DFA-model, these are exactly the outgoing actions of a state. Since for the DFA-model we have defined that $s \in P^{\mathcal{A}}(Out(q))$, the semantic condition is obeyed (Note that this explains why we use DFAs and not NDFAs: for NDFAs, the outgoing actions of automaton states may be only a subset of the actions that keep us within the action trace set described by the automaton). Now we have to ascertain that the μ^a-calculus formula $f(q^i(U^\alpha))$ holds in the initial state of the special DFA-model. The formula $f(q^i(U^\alpha))$ only contains ν-modalities. In section 3.5 we explained that satisfaction of a ν-formula is equivalent with satisfaction of its infinitary expansion. In the construction of $f(q^i(U^\alpha))$, these infinitary expansions can be formed by dropping the condition that if a return state is attended for the second time, a μ-calculus state-variable is introduced: instead we simply get another call of the function f. Also if return states are attended for the third, fourth, etc. time we keep calling the function f, and the formula $f(q^i(U^\alpha))$ is infinitely expanded. Now it is fairly easy to see, that

by induction over the structure of the DFA-model from the initial state, the infinitary expansion of the formula $f(q^i(U^\alpha))$ is satisfied in the initial state. It remains to be shown that models other than the DFA-model, satisfying the same semantic conditions, also satisfy $f(q^i(U^\alpha))$. We claim that any other model satisfying the semantic condition of definition 5.3.3 can be constructed from the special DFA-model by model transformations that do not affect validity of the formula $f(q^i(U^\alpha))$ in the initial state. In what way can models that satisfy the semantic condition of definition 5.3.3 differ from the special DFA-model? First of all there might be many trajectories from the initial state that are not sub-traces of action traces interpreting α. Clearly, the presence of such trajectories does not affect validity of $f(q^i(U^\alpha))$, since the modalities in the formula do not concern actions that are not in the action trace set for α. Second, the tree structure of the set of trajectories in the DFA-model that do follow an action trace from α, may be different in other models satisfying the semantic condition. In particular, it need not be that in models satisfying the semantic condition, common prefixes of action traces for α follow the same trajectory in a model (such as in the DFA-model). But this also does not affect validity of $f(q^i(U^\alpha))$, since it contains only non-negated nested ν-modalities, that cannot distinguish a tree from a set of traces through the tree. Third, models satisfying the semantic condition might differ form the DFA-model in the sense that trajectories from the initial state are completely or partially absent. Again, due to the fact that it contains only non-negated nested ν-modalities, this does not affect validity of $f(q^i(U^\alpha))$.

We call the opposite direction to prove, 'sufficiency'. To prove sufficiency, we have to show that the formula $f(q^i(U^\alpha))$ imposes the conditions of definition 5.3.3 on traces. This follows from the μ^a-calculus semantics of the formula $f(q^i(U^\alpha))$, but is also ascertained by the standard semantics for infinite expansion of the formula $f(q^i(U^\alpha))$.

■

A regular action is equivalent with many different DFAs. According to definition 5.3.6, all of these DFAs can be used to form a μ^a-calculus expression for a property $P_{\rightsquigarrow}(\alpha)$. The following proposition states that if for two different DFAs for the same α, by applying definition 5.3.6 two different μ^a-calculus characterizations follow, the two formulas are logically equivalent.

Proposition 5.3.4 *μ^a-calculus translations of different DFAs describing the same set of traces are logically equivalent.*

Sketch of a proof

We do not prove this in detail, but reveal the intuition behind the proof. From automaton theory it is known that for any DFA (and even FA) there is a unique minimal DFA (MDFA) that describes the same trace-set, and, that DFAs differ from the MDFA only in the sense that some MDFA-states have equivalent copies. The recursive function of definition 5.3.6 is not able to distinguish between DFAs that only differ in the sense that certain automaton states have copies. This is illustrated by a small example .

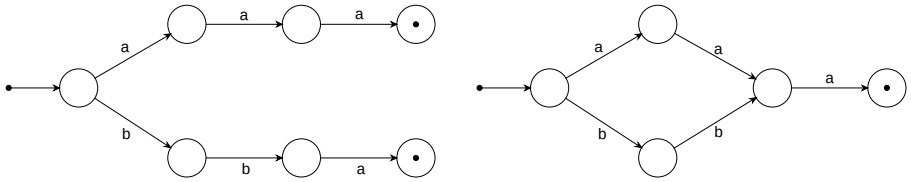


Fig 15. two trace-equivalent DFAs

Both the DFAs of the example represent the trace-set $\{\langle a,a,a\rangle, \langle a,b,a\rangle\}$. The right DFA is minimal. The left DFA contains copies of the end and before-end state of the minimal DFA. This is a very simple example of copies of automaton states. In general also automaton return states can be copied, which may lead to DFAs that are not easily recognizable as equivalent to a minimal form. It is not difficult to see that for both DFAs the function f of definition 5.3.6 returns exactly the same μ^a-calculus formula. If copies of automaton return states are involved, μ^a-calculus formulas may differ in the names of state-set variables Z_s with $s \in R$. But this does not have any influence on the validity of the formula.

■

A μ^a-calculus characterization for obligation

In section 5.3.1 we saw that the semantic characterization of obligation closely resembles the one for permission. The main difference is that primitive normative conditions are absent in states where an action trace has been executed completely. This difference comes back in the automaton based μ^a-calculus characterization for obligation, defined below.

Definition 5.3.7 *Let α be a regular action, and $U^\alpha = (Q, A, N, q^i, T)$ a corresponding deterministic finite automaton with Q a set of automaton states, A a set of automaton actions, $N : Q \times A \to Q$ the transition function, $q^i \in Q$ the initial automaton state, and $T \subset Q$ the set of terminal automaton states.*

Furthermore, let $R \subseteq Q$ be the set of automaton return states of the DFA, let Z_q be a μ^a-calculus state variable associated with the automaton return state $q \in R$, and let $Out(q)$ be the set of outgoing automaton actions of an automaton state q of Q. Then a μ^a-calculus formula for $O_{\rightsquigarrow}(\alpha)$ is built with the help of a recursive function f that associates a μ^a-calculus formula to each automaton state:

$$
\begin{array}{lrcl}
\textit{if } q \in Q \setminus (R \cup T), & f(q) & = & O_{\rightsquigarrow}(\bigcup Out(q)) \wedge \bigwedge\limits_{a \in Out(q)} [a] f(N(q,a)) \\
\textit{if } q \in R \setminus T, & f(q) & = & \nu Z_q.\, O_{\rightsquigarrow}(\bigcup Out(q)) \wedge \bigwedge\limits_{a \in Out(q)} [a] f(N(q,a)) \\
\textit{if } q \in T \setminus R, & f(q) & = & \bigwedge\limits_{a \in Out(q)} [a] f(N(q,a)) \\
\textit{if } q \in R \cap T, & f(q) & = & \nu Z_q.\, \bigwedge\limits_{a \in Out(q)} [a] f(N(q,a)) \\
\textit{if } q \in R, & f'(q) & = & Z_q
\end{array}
$$

The distinction between primed and unprimed values is the same as in definition 5.3.6. The μ^a-calculus characterization of $O_{\rightsquigarrow}(\alpha)$ is defined as the formula associated to the initial state of the automaton: $O_{\rightsquigarrow}(\alpha) = f(q^i)$.

The differences with the definition of the function f for permission follow directly from the semantic choices discussed in section 5.3.1. First, on the level of atomic actions, we have of course $O_{\rightsquigarrow}(\bigcup Out(q))$ instead of $P_{\rightsquigarrow}(\bigcup Out(q))$. Second, the assignment of μ^a-calculus formulas to terminal automaton states $q \in T$ is different from the assignment to non-terminal automaton states. There are even two cases, one describing the formula $f(q)$ for terminal states that are not at the same time return states ($q \in T \setminus R$), and one for terminal states that are return states ($q \in R \cap T$). The difference with the definition of $f(q)$ for the non-terminal automaton states $q \in Q \setminus (R \cup T)$ and $q \in R \setminus T$ is that the obligations $O_{\rightsquigarrow}(\bigcup Out(q))$ are left out, which corresponds to the semantic choice that in the states of trajectories that correspond to the complete execution of an action trace (terminal states in the automatons) no atomic obligations should hold. We now formulate the theorem that says that this characterization is equivalent to the semantic characterization of obligation in definition 5.3.4.

Theorem 5.3.5 *The translation $O_{\rightsquigarrow}(\alpha) = f(q^i(U^\alpha))$, is correct and sufficient with respect to the semantic characterization of definition 5.3.4.*

Sketch of a proof

The proof of the correctness and sufficiency differs only in details from the one for $P_{\rightsquigarrow}(\alpha)$, and follows from the correspondence of the semantic characterization in terms of traces, and the structure of automatons, seen as action models for $O_{\rightsquigarrow}(\alpha)$.

■

We conclude this section with two examples.

Example 5.3.2 *Consider the properties:* $O_{\rightsquigarrow}((a;b)^*)$ *and* $O_{\rightsquigarrow}((a \cup b)^*;b;a)$. *DFAs of the regular actions* $(a;b)^*$ *and* $(a \cup b)^*;b;a$ *are:*

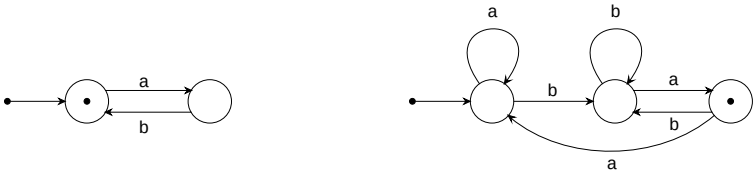


Fig 16. two example DFAs

The μ^a*-calculus expression for* $O_{\rightsquigarrow}((a;b)^*)$ *is:* $\nu Z.\,[a](O_{\rightsquigarrow}(b) \wedge [b]Z)$. *Note that* $\not\models O_{\rightsquigarrow}((a;b)^*) \rightarrow O_{\rightsquigarrow}(a)$, *while for permission:* $\models P_{\rightsquigarrow}((a;b)^*) \rightarrow P_{\rightsquigarrow}(a)$. *Again this shows the difference between permission and obligation with respect to normative properties holding in states where an action trace has been executed completely.*

The μ^a*-calculus expression for* $O_{\rightsquigarrow}((a \cup b)^*;b;a)$ *is:* $\nu Z.\, O_{\rightsquigarrow}(a \cup b) \wedge [a]Z \wedge [b](\nu Y.\, O_{\rightsquigarrow}(a \cup b) \wedge [b]Y \wedge [a]([b]Y \wedge [a]Z))$. *This example shows that it is not always straightforward to recognize that a regular action and a DFA describe the same set of traces. Consequently, a* μ^a*-calculus expression is not always easily recognized as describing a certain deontic action notion.*

5.3.3 Compositionality in action combinators

The μ^a-calculus characterization of $P_{\rightsquigarrow}(\alpha)$ and $O_{\rightsquigarrow}(\alpha)$ shows a particular form of compositionality of the semantics of these notions, namely, in terms of the primitives $P_{\rightsquigarrow}(a_1 \cup a_2 \cup \ldots \cup a_n)$ and $O_{\rightsquigarrow}(a_1 \cup a_2 \cup \ldots \cup a_n)$, through an automaton-based translation. In this section we discuss the compositionality of the defined semantics with respect to the regular action combinators choice, sequence and iteration.

Compositionality of permission and prohibition

We formulate properties that define the compositionality of permission and prohibition with respect to the regular action combinators. For permission,

we first observe that according to the free choice interpretation, the primitive $P_{\rightsquigarrow}(a_1 \cup a_2 \cup \ldots \cup a_n)$ is actually no primitive, since it is intuitive to decompose it into $P_{\rightsquigarrow}(a_1) \wedge P_{\rightsquigarrow}(a_2) \wedge \ldots \wedge P_{\rightsquigarrow}(a_n)$. This means that for permission, we could have taken the notion $P_{\rightsquigarrow}(a)$ for atomic actions a as primitive. We did not do that in order to stress the similarities between the automaton-based μ^a-calculus characterizations of the notions of permission and obligation for actions that are more complex than atomic choice actions of the form $a_1 \cup a_2 \cup \ldots \cup a_n$. However, in this section about compositionality, we assume that indeed normative atoms of the form $P_{\rightsquigarrow}(a)$ are primitive. The semantic characterizations of definition 5.3.3, are adapted accordingly, by making the identification $P_{\rightsquigarrow}(a_1 \cup a_2 \cup \ldots \cup a_n) \equiv P_{\rightsquigarrow}(a_1) \wedge P_{\rightsquigarrow}(a_2) \wedge \ldots \wedge P_{\rightsquigarrow}(a_n)$. Assuming this identification, we give some examples of the compositionality of the semantics for permission in terms of the regular action combinators. We start with compositionality with respect to $\cup$ for which we again consider the example $P_{\rightsquigarrow}((a;b) \cup (c;d))$. The corresponding μ^a-calculus expression is $P_{\rightsquigarrow}(a \cup c) \wedge [a]P_{\rightsquigarrow}(b) \wedge [c]P_{\rightsquigarrow}(d)$. Now $\cup$-compositionality should guarantee that this is expressible in terms of $P_{\rightsquigarrow}(a;b)$ and $P_{\rightsquigarrow}(c;d)$, for which the μ^a-calculus expressions are $P_{\rightsquigarrow}(a) \wedge [a]P_{\rightsquigarrow}(b)$ and $P_{\rightsquigarrow}(c) \wedge [c]P_{\rightsquigarrow}(d)$. With the identification $P_{\rightsquigarrow}(a_1 \cup a_2 \cup \ldots \cup a_n) \equiv P_{\rightsquigarrow}(a_1) \wedge P_{\rightsquigarrow}(a_2) \wedge \ldots \wedge P_{\rightsquigarrow}(a_n)$ it follows immediately that $P_{\rightsquigarrow}((a;b) \cup (c;d)) \equiv P_{\rightsquigarrow}(a;b) \wedge P_{\rightsquigarrow}(c;d)$. To investigate *-compositionality, we consider the formula $P_{\rightsquigarrow}(a^*)$ as an example. The μ^a-calculus characterization is $\nu Z.\, P_{\rightsquigarrow}(a) \wedge [\alpha]Z$, which shows how $P_{\rightsquigarrow}(a^*)$ is decomposed in terms of $P_{\rightsquigarrow}(a)$. Finally, it is easy to check that $P_{\rightsquigarrow}(a;b)$ is decomposed into $P_{\rightsquigarrow}(a) \wedge [a]P_{\rightsquigarrow}(b)$, which exemplifies ;-compositionality.

Proposition 5.3.6 *With the identification $P_{\rightsquigarrow}(a_1) \wedge P_{\rightsquigarrow}(a_2) \wedge \ldots \wedge P_{\rightsquigarrow}(a_n) \equiv P_{\rightsquigarrow}(a_1 \cup a_2 \cup \ldots \cup a_n)$, the following properties are correct and sufficient with respect to the semantic characterization of definition 5.3.3:*

$$
\begin{array}{lcllcl}
P_{\rightsquigarrow}(skip) & \leftrightarrow & \top & F_{\rightsquigarrow}(skip) & \leftrightarrow & \bot \\
P_{\rightsquigarrow}(\alpha;\beta) & \leftrightarrow & P_{\rightsquigarrow}(\alpha) \wedge [\alpha]P_{\rightsquigarrow}(\beta) & F_{\rightsquigarrow}(\alpha;\beta) & \leftrightarrow & F_{\rightsquigarrow}(\alpha) \vee \langle\alpha\rangle F_{\rightsquigarrow}(\beta) \\
P_{\rightsquigarrow}(\alpha \cup \beta) & \leftrightarrow & P_{\rightsquigarrow}(\alpha) \wedge P_{\rightsquigarrow}(\beta) & F_{\rightsquigarrow}(\alpha \cup \beta) & \leftrightarrow & F_{\rightsquigarrow}(\alpha) \vee F_{\rightsquigarrow}(\beta) \\
P_{\rightsquigarrow}(\alpha^*) & \leftrightarrow & \nu Z.\, P_{\rightsquigarrow}(\alpha) \wedge [\alpha]Z & F_{\rightsquigarrow}(\alpha^*) & \leftrightarrow & \mu Z.\, F_{\rightsquigarrow}(\alpha) \vee \langle\alpha\rangle Z
\end{array}
$$

Note that fix-point formulas only appear in the decomposition of $P_{\rightsquigarrow}(\alpha^*)$ and $F_{\rightsquigarrow}(\alpha^*)$. In some of the above decompositions, formulas of the form $[\alpha]\varphi$ and $\langle\alpha\rangle\varphi$ appear, which for a further decomposition rely on the μ^a-calculus translation of PDL from section 2.3.

Sketch of a proof

For correctness of $P_{\rightsquigarrow}(\alpha;\beta) \leftrightarrow P_{\rightsquigarrow}(\alpha) \wedge [\alpha]P_{\rightsquigarrow}(\beta)$ we have to prove that this formula holds given that for the action traces for $P_{\rightsquigarrow}(\alpha;\beta)$, $P_{\rightsquigarrow}(\alpha)$, and $P_{\rightsquigarrow}(\beta)$ the conditions of definition 5.3.3 hold. By way of example, we prove the direction from left to right. Now consider a state in a model where the semantic conditions for $P_{\rightsquigarrow}(\alpha;\beta)$ hold. These semantic conditions concern all states on trajectories corresponding to the action traces from $\Gamma(\alpha;\beta)$. Then these same semantic conditions ensure that also $P_{\rightsquigarrow}(\alpha)\wedge[\alpha]P_{\rightsquigarrow}(\beta)$ holds. Recall (section 2.4.3) that the action traces from $\Gamma(\alpha;\beta)$ are concatenations of traces from $\Gamma(\beta)$ to action traces from $\Gamma(\alpha)$. Then, the conditions for $P_{\rightsquigarrow}(\alpha)$ are obeyed on the trajectories that correspond to the first part of concatenated action traces. And, since $[\alpha]\varphi$ means that φ holds after all trajectories that follow an action trace from $\Gamma(\alpha)$, the property $[\alpha]P_{\rightsquigarrow}(\beta)$ is obeyed, because it reflects that on the second part of concatenated trajectories the semantic conditions are met. It is illustrative to verify this for the example models for $P_{\rightsquigarrow}(a;(b\cup c))$ given earlier. The property 'decomposes' $P_{\rightsquigarrow}(a;(b\cup c))$ into $P_{\rightsquigarrow}(a)\wedge[a]P_{\rightsquigarrow}(b\cup c)$. The right hand side states that in both traces ab and ac, after a is performed, the permission property $P_{\rightsquigarrow}(b\cup c)$ must hold. Correctness of $P_{\rightsquigarrow}(\alpha\cup\beta) \leftrightarrow P_{\rightsquigarrow}(\alpha)\wedge P_{\rightsquigarrow}(\beta)$ and $P_{\rightsquigarrow}(\alpha^*) \leftrightarrow \nu Z.\ P_{\rightsquigarrow}(\alpha)\wedge[\alpha]Z$ can be ascertained by similar arguments.

To prove sufficiency we have to go the other way: given the validity of the formulas we have to prove that the right conditions on traces are imposed. This can be seen as follows. The properties, together with the μ^a-calculus expressions for PDL of definition 3.5.1, can be used to 'break down' any formula $P_{\rightsquigarrow}(\alpha)$ and $F_{\rightsquigarrow}(\alpha)$ into formulas containing only primitive deontic formulas of the form $P_{\rightsquigarrow}(a)$ and ν-modalities over atomic actions a. What we actually get by following this procedure, is an alternative, but equivalent μ^a-calculus characterization. We can then use the same prove technique as for the sufficiency of the μ^a-calculus characterization of definition 5.3.6, that is, we can consider the infinite expansion of the μ^a-calculus characterization, and ascertain that it imposes exactly the right conditions on models, by an induction argument over the formula structure.

■

Compositionality of obligation

For permission we observed that the free choice intuition forced us to accept that formulas $P_{\rightsquigarrow}(a_1\cup a_2\cup\ldots\cup a_n)$ we used as primitives, can actually be broken down. However, for obligation, the free choice interpretation does not give us that the normative elements $O_{\rightsquigarrow}(a_1\cup a_2\cup\ldots\cup a_n)$ hide more elementary

primitives. A decomposition into $O_{\rightsquigarrow}(a_1) \wedge O_{\rightsquigarrow}(a_2) \wedge \ldots \wedge O_{\rightsquigarrow}(a_n)$ is simply not intuitive, as discussed in section 5.3. This non-compositionality of choice at the atomic level, causes a non-compositionality of choice at the general action level. Consider the example $O_{\rightsquigarrow}((a;b) \cup (c;d))$. The corresponding μ^a-calculus expression is $O_{\rightsquigarrow}(a \cup c) \wedge [a]O_{\rightsquigarrow}(b) \wedge [c]O_{\rightsquigarrow}(d)$. Now $\cup$-compositionality would guarantee that this is expressible in terms of $O_{\rightsquigarrow}(a;b)$ and $O_{\rightsquigarrow}(c;d)$, for which the μ^a-calculus expressions are $O_{\rightsquigarrow}(a) \wedge [a]O_{\rightsquigarrow}(b)$ and $O_{\rightsquigarrow}(c) \wedge [c]O_{\rightsquigarrow}(d)$. A reduction would thus require that the obligation $O_{\rightsquigarrow}(a \cup c)$ is reduced to $O_{\rightsquigarrow}(a)$ and $O_{\rightsquigarrow}(c)$. This is not possible, since all three these expressions are primitives.

The notion of process obligation is also not compositional in the iteration combinator. But, for a different reason. In the semantics for $O_{\rightsquigarrow}(\alpha^*)$, we defined that α is not obliged at any state that corresponds to a complete execution of one of the traces for α^*. This means, for instance, that the obligation $O_{\rightsquigarrow}(a^*)$ for an atomic action a is void (equivalent with $\top$): the states where a has been executed zero, one or more times, all correspond to a complete execution of one of the action traces, which means that no conditions are imposed whatsoever. To show that in general $O_{\rightsquigarrow}(\alpha^*)$ cannot be decomposed in terms of $O_{\rightsquigarrow}(\alpha)$, we look at the example: $O_{\rightsquigarrow}(a;b)^*$. A corresponding μ^a-calculus expression is $\nu Z.\ [a](O_{\rightsquigarrow}(b) \wedge [b]Z)$. This is not expressible in terms of $O_{\rightsquigarrow}(a;b)$, which is identified with $O_{\rightsquigarrow}(a) \wedge [a]O_{\rightsquigarrow}(b)$, because it encompasses the primitive notion $O_{\rightsquigarrow}(a)$ that does not play any role in $\nu Z.\ [a](O_{\rightsquigarrow}(b) \wedge [b]Z)$.

Finally we argue that the notion of process obligation is also not compositional in the sequence combinator. Consider the example $O_{\rightsquigarrow}(a^*;b)$. This formula is not expressible in terms of $O_{\rightsquigarrow}(a^*)$ and $[a^*]O_{\rightsquigarrow}(b)$. The μ^a-calculus formulas characterizations are $\nu Z.\ O_{\rightsquigarrow}(a \cup b) \wedge [a]Z$, $\top$ and $\nu Z.\ O_{\rightsquigarrow}(b) \wedge [a]Z$. We cannot express the first of these formulas in terms of the other two because all three formulas $O_{\rightsquigarrow}(a \cup b)$, $O_{\rightsquigarrow}(a)$ and $O_{\rightsquigarrow}(b)$ are primitive. And even if we would assume, as for permission, that we could break down $O_{\rightsquigarrow}(a \cup b)$, the formulas would still be incomparable, because obligations concerning the atomic action a are completely absent in the second two formulas. Note however, that this counterexample only proves non-compositionality with respect to the compound operation '*;'. Indeed, for regular actions in which the ; is never preceded by a *, we can prove that $O_{\rightsquigarrow}(\alpha;\beta) \leftrightarrow O_{\rightsquigarrow}(\alpha) \wedge [\alpha]O_{\rightsquigarrow}(\beta)$ holds. The non-compositionality with respect to sequence is thus 'caused' by the non-compositionality with respect to iteration.

Summarizing we may say that the non-compositionality of obligation, with respect to the regular action syntax, has two main causes. The first is that

obligation of (free) choice is not compositional, which traces back to non-compositionality at the level of choice over atomic action. The second is that obligation of iteration is non-compositional, which is caused by the fact that we can comply with an obligation concerning an iteration by doing nothing. This relates to the absence of primitive obligations in states that correspond to complete execution of action traces.

5.3.4 Reductions to the μ^m- and the μ^η-calculus

in section 5.3.1 we introduced the notions $P_{\rightsquigarrow}(a_1 \cup a_2 \cup \ldots \cup a_n)$, $F_{\rightsquigarrow}(a_1 \cup a_2 \cup \ldots \cup a_n)$ and $O_{\rightsquigarrow}(a_1 \cup a_2 \cup \ldots \cup a_n)$ as primitives for deontic process reasoning. By introducing these primitives, we temporarily discarded the deontic logic of atomic choice actions themselves, and also, the logic of the interactions between the separate normative modalities. Since we have shown in definitions 5.3.6 and 5.3.7 how to reduce normative expressions concerning complex regular actions to μ^a-calculus expressions over the above mentioned primitives, the interactions of the modalities are completely determined by the interactions at the atomic choice level. We give two possible definitions for the logic of atomic choice actions and the interactions.

The first possibility involves a shift from the normative choice models of definition 5.3.2 to standard modal action models, where we introduce normativity in the same way as Meyer does, that is, by distinguishing a special violation proposition V_F. This shift goes hand in hand with a shift in language, from the μ^a-calculus to the μ^m-calculus as defined by Bradfield and Stirling [23, 173, 25]. The μ^m-calculus extends the μ^a-calculus with the elementary action language $m, n, \ldots := a_1, a_2, \ldots, a_n \mid -(a_1, a_2, \ldots, a_n)$ for $a_i \in \mathcal{A}$ (this explains why we use the term 'μ^m-calculus' for the calculus defined by Bradfield and Stirling, while the authors themselves refer to their calculus simply as 'the modal μ-calculus'). The actions $a_1, a_2, \ldots, a_n$ are interpreted as a choice over the atomic actions a_i. So they correspond to what we have called 'atomic choice actions'. An action $-(a_1, a_2, \ldots, a_n)$ is interpreted as the choice over all actions other than $a_1, a_2, \ldots, a_n$, that is, over the actions in $\mathcal{A} \backslash \{a_1, a_2, \ldots, a_n\}$. So the μ^m-calculus actually contains a rudimentary notion of action complement. For the reduction to normative expressions over atomic choice actions, we did not need the action complement. But now that it comes down to defining the relation between separate modalities, we need it again. In the μ^m-calculus, the primitives $P_{\rightsquigarrow}(a_1 \cup a_2 \cup \ldots \cup a_n)$, $F_{\rightsquigarrow}(a_1 \cup a_2 \cup \ldots \cup a_n)$ and $O_{\rightsquigarrow}(a_1 \cup a_2 \cup \ldots \cup a_n)$ can be defined as follows.

Definition 5.3.8 (a μ^m-calculus reduction) *Reductions for the normative primitives in the μ^m-calculus:*

$$\begin{array}{lll} F_{\rightsquigarrow}(a_1 \cup a_2 \cup \ldots \cup a_n) & \equiv_{def} & \langle a_1, a_2, \ldots, a_n \rangle V_F \\ P_{\rightsquigarrow}(a_1 \cup a_2 \cup \ldots \cup a_n) & \equiv_{def} & [a_1, a_2, \ldots, a_n] \neg V_F \\ O_{\rightsquigarrow}(a_1 \cup a_2 \cup \ldots \cup a_n) & \equiv_{def} & [-(a_1, a_2, \ldots, a_n)] V_F \end{array}$$

Note that this definition imposes the same relation between obligation and prohibition as the reduction for goal norms in definition 5.2.2. Thus, $O_{\rightsquigarrow}(m)$ not only implies the prohibition of the action $-m$, it also implies the prohibition of any action n whose outcome (effect) is possibly different than any possible outcome for m. It is not too difficult to see that this property for the atomic choice level, also holds for the general regular action level. This property could thus be seen as the process norm version of the deontic principle that 'being obliged' implies 'being forbidden not to'.

With definition 5.3.8 we have completed a full reduction of expressions $P_{\rightsquigarrow}(\alpha)$, $F_{\rightsquigarrow}(\alpha)$, and $O_{\rightsquigarrow}(\alpha)$ for regular actions α to the standard μ^m-calculus defined by Stirling and Bradfield [25] extended with one distinguished violation proposition V_F. Normative expressions are consistent iff their μ^m-calculus translations are consistent, entailment relations between normative expressions (possibly involving different deontic modalities) exist if and only if entailment relations between their μ^m-calculus translations exist, etc.

The second possibility for defining the logic of the primitives $P_{\rightsquigarrow}(a_1 \cup a_2 \cup \ldots \cup a_n)$, $F_{\rightsquigarrow}(a_1 \cup a_2 \cup \ldots \cup a_n)$ and $O_{\rightsquigarrow}(a_1 \cup a_2 \cup \ldots \cup a_n)$ follows from the observation that atomic choice actions do not involve sequence or iteration, which means that the difference between goal and process norms should not give rice to different logics for such actions. We can thus base ourselves on the insights developed in section 5.2, and take a goal norm logic to function as the logic of the primitives for process norms. As for the μ^m-calculus approach defined above, this involves a (small) shift of models and of language. For the models we take standard modal action models, and introduce normativity in these models by distinguishing the violation propositions V_P, V_F and V_O, as in section 5.2.1. And, as for the μ^m-calculus approach defined above, we add an action language, which in this case is much more powerful. We add the language $\eta, \vartheta, \ldots := a \mid \eta \cup \vartheta \mid \wr^B \eta \mid \eta^{\leftarrow}$. With this addition we get the μ^η-calculus, as defined in section 3.5. Recall that in the μ^η-calculus we distinguish two action layers: the layer $\eta, \vartheta, \ldots := a \mid \eta \cup \vartheta \mid \wr^B \eta \mid \eta^{\leftarrow}$, and a layer on top of that, that in this case is layer of regular actions over actions η: $\alpha, \beta, \ldots := \eta \mid skip \mid \alpha \cup \beta \mid \alpha; \beta \mid \alpha^*$. This second layer is not explicitly defined

in the syntax of the μ^η-calculus, since it is syntactically definable in terms of μ^η-calculus formulas, as shown in section 3.5. Then, the deontic logic over the lower level actions η, can be taken as in section 5.2.1. In the μ^η-calculus, the primitives $P_{\rightsquigarrow}(a_1 \cup a_2 \cup \ldots \cup a_n)$, $F_{\rightsquigarrow}(a_1 \cup a_2 \cup \ldots \cup a_n)$ and $O_{\rightsquigarrow}(a_1 \cup a_2 \cup \ldots \cup a_n)$ can thus be defined as follows.

Definition 5.3.9 (a μ^η-calculus reduction) *μ^η-calculus reductions for the normative primitives $N_{\rightsquigarrow}(\eta)$ over atomic choice actions $\eta = a_1 \cup a_2 \cup \ldots \cup a_n$:*

$$\begin{array}{llll} F_{\rightsquigarrow}(\eta) & \equiv_{def} & F_{\odot}(\eta) & \equiv_{def} \neg\langle\eta\rangle V_P \\ P_{\rightsquigarrow}(\eta) & \equiv_{def} & P_{\odot}(\eta) & \equiv_{def} \langle\eta\rangle V_F \\ O_{\rightsquigarrow}(\eta) & \equiv_{def} & O_{\odot}(\eta) & \equiv_{def} P_{\odot}(\eta) \wedge [\wr^B \eta] V_O \end{array}$$

With this definition, we have completed a full reduction of expressions $P_{\rightsquigarrow}(\alpha)$, $F_{\rightsquigarrow}(\alpha)$, and $O_{\rightsquigarrow}(\alpha)$ for regular actions α to the standard μ^η-calculus defined in section 3.5. Note that with the introduction of the converse at the lower action level, we have implicitly introduced the converse also at the process level. Given an arbitrary action term $\alpha^{\leftarrow}$ where α is an action of the two layered syntax mentioned above (and in definition 3.4.1), we can use the reduction rules defined in the proof of theorem 2.3.2 to the level of actions η, where it is well-defined.

For the deontic logic over actions η at the lower action level, all the issues for goal norms mentioned in section 5.2.1 are relevant. The choices made at this level have repercussions for the logic of process norms at the higher action level. One such issue is the incorporation of the 'ought implies may' principle. In definition 5.3.9 above, we included it to emphasize that we can use the goal norm definitions of section 5.2.1 here directly as definitions for the logic of the non-sequential process norm primitives. The reader might check that the inclusion in definition 5.3.9 of the property $O_{\odot}(a_1 \cup a_2 \cup \ldots \cup a_n) \rightarrow P_{\odot}(a_1 \cup a_2 \cup \ldots \cup a_n)$, does not mean that the ought implies may principle also holds at the process norm level. This is due to the difference between the μ^a-calculus reductions for process permission and process obligation of definitions 5.3.6 and 5.3.7. In particular, we do not have that $O_{\rightsquigarrow}(a^*) \rightarrow P_{\rightsquigarrow}(a^*)$, since $O_{\rightsquigarrow}(a^*)$ is obeyed in any state of any model and $P_{\rightsquigarrow}(a^*)$ is not.

5.3.5 Contrary to duty process norms

In section 5.2.3 we discussed that we can adopt two different views on the interpretation of CTD-norms: the temporal view, and the synchronous view. For process norms, it seems very hard to find a satisfying interpretation of

CTD-norms in the synchronous view. The intuitions for a temporal view on process norm CTD-norms are much clearer. We only consider CTD-norms for primary norms that are obligations. We mentioned in section 5.2.3 the possibility of notions of contrary to duty for prohibition and permission. We do not consider these, and leave their investigation for future research. The temporal view on CTD-norms encompasses that CTD-norms hold in states where a primary norm is violated. So, if we want to use this view in the definition of CTD-norms for process norms, we need to be able to refer to the states where primary norms are violated. These are exactly the states where the violation proposition V_O holds. So, if we want to express that a norm $N_{\rightsquigarrow}(\alpha)$ comes into force at the point where a primary norm $O_{\rightsquigarrow}(\alpha)$ is violated, we can take the μ^m-calculus or μ^η-calculus translation of the norm $O_{\rightsquigarrow}(\alpha)$ and substitute the formula $N_{\rightsquigarrow}(\alpha)$ for every occurrence of the proposition V_O. But then, we could also substitute the μ^m-calculus or μ^η-calculus translation of the norm $N_{\rightsquigarrow}(\alpha)$ for V_O. This simple procedure thus shows how we can also translate CTD-norms into the μ^m-calculus or μ^η-calculus.

5.4 Related work

Our work can be suitably given a place in the tradition on dynamic deontic logic [135, 185, 137, 186, 132, 56]. In section 5.2.1 we already discussed the work of Meyer, who initiated this line of work. In this section we want to give an impression of the situatedness of the work on dynamic deontic logic in the extensive deontic logic literature.

A well-established classification of deontic logics is the distinction between ought-to-do and ought-to-be logics. Ought-to-be logics take conditions referring to properties of situations, system states, worlds or states of affairs as arguments for deontic modalities. So in ought-to-be deontic logics, we typically have a modal expression $P(A)$ saying 'it is permitted to be in a static situation that satisfies the condition A'. In ought-to-do logics, on the other hand, deontic modalities take *actions* or *act-propositions* as their arguments. So in an ought-to-do logic, an expression $P(act)$ roughly means 'it is permitted to perform activity *act*'. The deontic logics in this chapter are all of the ought-to-do type. Both Segerberg [167] and Von Wright [190, 194] emphasize that any ought-to-do deontic logic should be based on a proper logic of action. For this purpose, we used the modal action logics developed in chapters 2 and 3. But it is good to realize that in the literature, action theories of a quite different nature have functioned as a basis for ought-to-do deontic logics. There are at least three important ways in the literature, in which the deontic

expressions $O(act), P(act)$ and $F(act)$, with *act* a term representing action in some way, are interpreted as normative assertions about action.

In his 1951-work [189], Von Wright uses a standard propositional logic formalism for *act* and reads atomic expressions p as 'act predicates': $F(p)$ means that any act is prohibited that is in the extension of the predicate p. This leads to the view ([94]) that individual actions form the domain of interpretation of compound act-formulas (using the standard logical connectives), and that it is possible to view deontic operators as quantifiers that bind variables of act-predicates to the domain of individual actions. Hintikka [95] has argued that under this reading of deontic formulas, many traditional deontic principles, such as the interdefinabilities $O(p) = F(\neg p)$, and $O(p) = \neg P(\neg p)$, do *not* follow. He says:

> 'I would say that the logical structure of obligations, prohibitions and permissions pertaining to kinds of acts is more complicated than what is usually assumed.'

The well-studied STIT (Seeing To It That) logics [11, 166, 168, 99, 12], which are close to older work on, what we call, BIAT (Bringing It About That) logics [104, 105, 116, 151], consider elements *act* to be of the form $E_i p$, representing 'agent i sees to it that p'. Terms $E_i p$, are studied as operators in their own right and are often referred to as 'action modalities'. But whereas in Von Wright's 1951-view, actions are explicit in the interpretation of act predicates, in the STIT-approach actions are left implicit all together. The logic of the operator $E_i p$ is much more concerned with a normative view on *agency* (the parameter i in $E_i p$) and the bringing about, through (implicit) action, of *conditions* (the propositions p in $E_i p$), than with a normative view on 'activity' itself. For instance, STIT theory does not enable a distinction between being allowed to bring about condition A through action α, but not through action β.

STIT-approaches include a notion of 'not seeing to it' which is often called 'refraining'. Refraining is conceptually related to action complement: refraining from an action means that alternative actions are performed. But the correspondence with the notion of relativized action negation of section 2.5.3 is restricted to the conceptual level, since STIT-formalisms do not have an explicit action language. Furthermore, refraining for STIT-formalisms is not an non-controversial concept, and it has several alternative formalizations [151, 12].

Von Wright [192, 194] notes that deontic action logics that are not based on an explicit action language have a problem in distinguishing between 'seeing to it that' and 'preventing that it will occur that not'. He argues that

this calls for an explicit action language and norms pertaining directly to action. Observations similar to those made by Von Wright, inspired Castañeda to divide ought-to-do norms into those pertaining directly to 'practitions', and those pertaining to action propositions, being conditions under which actions may occur. A practition is not a predicate describing some 'act property', or a condition being brought about by an act performed by some agent, but the act (event) itself. Castañeda [41, 42, 43] observed that many anomalies of deontic reasoning are not inherited by logics that interpret normative assertions as taking 'practitions' as an argument. It was argued by Hilpinen [94] that the group of dynamic deontic logics [135, 185, 132, 56, 36] can be viewed as deontic 'practition' logics. A short comment about terminology is required here. Hilpinen, Meyer, and several other researchers view the term 'dynamic logic' as referring to the whole group of modal logics that is characterized by the insertion of action terms into the modal box and diamond. Under this definition, all of the modal action logics discussed in chapter 2 are dynamic logics. However, we consider dynamic logic to be a *specific* modal action logic: the one with the action combinators sequence, choice iteration and converse (see 2.3). So, where Hilpinen and several other authors speak of 'dynamic deontic logics', we speak of 'deontic modal action logics'. This is the category of deontic logics studied in this chapter. And, following the view presented by Hilpinen [94], we might refer to the logics as '*practition*-type deontic modal action logics'.

Dynamic deontic logics study normative notions over complex actions. This gives rise to a whole range of questions not considered before in ought-to-do deontic logic. We saw in section 5.1.1 that, for instance, the deontic logic properties of choice were also studied in other settings. To some extent, the same holds for concurrency. But the study of the deontic logic properties of sequence, iteration, and converse is characteristic for the work on dynamic deontic logics. A good example is the work by Van der Meyden on the dynamic logic of permissions [132]. He defines two notions of permission for regular action: a free choice variant, and a variant he calls 'not forbidden'. This distinction is similar to our distinction between free process choice and imposed process choice. But Van der Meyden does not discuss the free choice intuitions thoroughly, does not recognize the possibility of partial imposed choice, and does not explain the connection with free choice semantics for non-sequential actions. And most importantly, Van der Meyden does not deal with obligation.

The three mentioned paradigms, i.e. the act-predicate approach, the STIT / BIAT-approaches, and the practition-type approaches are not as distinct as may seem at first sight. An interesting approach to the possible reconciliation

of the BIAT and practition views is proposed by Segerberg [165], who introduces actions $\delta_i p$ in dynamic logic representing 'agent i brings it about that p'. Also the act-predicate view closes in on the practition view if we consider act-predicates to be of the binary type, with possible states as the domain of interpretation.

5.5 Conclusions

In the introduction to this chapter we introduced a distinction between goal norms and process norms. For both types of action norm categories we defined reductions to modal action logics: for goal norms we defined reductions to the modal action logics with a relativized action complement of section 2.5.3, and for process norms we defined reductions to the μ^η-calculus defined in section 3.5. The reductions show that ought-to-do normativity can be given an purely dynamic interpretation. The first conclusion to be drawn from the possibility of such reductions, is that it gives 'proof' for the normative stance we formulated in section 1.4.1. The normative stance is essentially that if we adopt another viewpoint, we can see dynamic behavior in a normative manner. The reductions we defined in the present chapter can then be seen as the formal counterparts of going back from a normative viewpoint to a non-normative one.

A central theme for both reductions is the treatment of choice. We treat choice as under control, in the sense that a system (or agent) that is subject to the specified norms has the exclusive power to decide on the choice in complex actions α. A system can thus freely choose among the possibilities available according to the norms imposed on it; in particular it cannot be that it performs a permitted choice action and run into a violation. This alludes to the concept of 'free choice permission'. Our work shows that it is actually possible to define deontic logics of free choice permission. In the deontic logic literature it is generally assumed that the introduction of free choice permission leads to inconsistent or degenerate logics [121, 158].

For action goal norms we defined a 'cautious' reduction to modal action logic that deviates from the reduction defined by Meyer [135] in many respects: we do not assume the strong interdefinabilities between the separate deontic modalities, we avoid the free choice anomaly, and finally, the resulting deontic dynamic logics deal with converse and iteration of action. The definition was shown to be intuitive: it obeys many desired logic properties for ought-to-do dynamic deontic logics, and performs well on the standard reasoning examples concerning choice and contrary to duty norms.

For process norms the concept of free choice had to be refined, which led to the concept of 'free process choice'. Semantic characterizations for fee process choice norms were defined, and a DFA-based translation to μ-calculi was shown to correspond to this semantics. The analyses revealed that some detailed semantic choices have to be made in the formalization of the process norm versions of obligation and permission. In particular, we mentioned two important differences between the process notions of permission and obligation: the latter behaves differently with respect to choice over atomic action, and with respect to iteration.

Chapter 6

Discussion and conclusion

In chapter 1 we formulated as the central theme of research for this Ph.D. thesis, the development of a logic framework that combines reasoning about (1) action composition (using action combinators), (2) (discrete) time, (3) action effects, action possibilities, and (4) norms on action performances. In the preceding chapters we focussed on each of these individual reasoning tasks: in chapter 2 we focussed on concurrent action composition and action negation, in chapter 3 on temporal reasoning, in chapter 4 on reasoning about effects and qualifications accounting for action description assumptions, and in chapter 5 on reasoning about action norms. In this concluding chapter we investigate in how far we have reached the goal of designing a logic framework that *combines* the mentioned types of reasoning.

Combining (modal) logics is an area of research in itself [14, 67]. The first concern when combining two logics is how to combine them syntactically. There are several options. One is to merge the languages of separate logics without any restrictions. This is the most rigorous route; it needs a generalized semantic structure, that allows interpretation of both types of formulas, and indeed, of merged formulas. As an attractive possibility for such generalized semantic structures we mention multi-dimensional modal structures [127, 164, 13]. Due to interactions between modalities, the complexities of combined logics are in general much higher then the sum of the complexities of the individual logics. Therefore all kinds of weaker forms of combining logics are studied. A natural step is to consider layered logics: formulas of one modal type may occur within the scope of modalities of another modal type, but not the other way round.

We defined many different syntaxes in the preceding chapters. However, there is no reason whatsoever to consider constraints on the way in which the

syntaxes are combined. This holds in particular for the combination of 'bare' action modalities (as studied in chapter 2) and normative action modalities: we saw that goal oriented and process oriented normative expressions over complex actions can be *syntactically reduced* to $\langle I \rangle$-logics and the μ^{η}-calculus, respectively.

But, there is also a general reason for not having to restrict the way in which to combine the languages. The logics in this Ph.D. thesis are designed around a common semantic ground, namely, modal action models. In chapter 2 we showed how we can interpret concurrency and action complement on modal action models. In chapter 3 we showed how to interpret the temporal dimension on modal action models. In chapter 4 we showed how action description assumptions can be interpreted in terms of orderings of modal action models. Finally, in chapter 5, we showed how two separate types of action norms, i.e. goal norms and process norms, can be interpreted on modal action models. The common semantic ground enables us to interpret the logics that are formed by merging the separate languages, without generalizing the semantic structures. Thus, by using modal action models for all three types of reasoning, we have made a non-trivial step in achieving the goal mentioned in the central problem definition. In the sections below we focus on some specific interactions of the separate types of reasoning.

6.1 Action and time

In chapter 3 we showed how to combine action and time reasoning on modal action models. We introduced modalities that explicitly refer to both action and time. This enables us to express properties such as 'action α is performed until condition φ is met', and 'actions alternative to α are performed forever'. In our view, the relation between action and time is close, which is why time and action do not need separate modal accessibility relations (see e.g. the work on STIT-logics [12] for a different opinion on this issue). In chapter 2 we showed how to formalize the intuition that time, in a sense, is 'realized' by action. The main conclusion to be drawn is that reasoning about action and time can be combined fairly well in our framework. Difficulties were encountered in case modal action logics are strong enough to express intersection of action relations, i.e. concurrency. Atomic actions that take a minimal time-step may be concurrently composed with actions taking non-minimal time steps, which results in conflicting information for the duration of such concurrent compositions. The possibility to see intersection of action relations as confluency of courses of (non-Ockhamist) time is rejected, because intersection cannot

be used to model concurrency and time-confluency at the same time. As a possible solution to the problem concerning minimal time-steps for concurrent action, we proposed to restrict concurrency and action complement to non-sequential actions, which resulted in a two-layered action syntax. We showed how to define mixed temporal / dynamic modalities over modal action models, using this syntax.

6.2 Action description assumptions and time

In this section we shortly discuss the interactions between the solutions for the incorporation of description assumptions we proposed in chapter 4, and the solutions concerning the expression of temporal properties, as developed in chapter 3.

In chapter 3 we showed for several modal action logics how to define the basic temporal relations on models in order to define mixed action / time modal action logics. If we abstract from the details and the exceptions we discussed in chapter 3, we may roughly identify the time-line with the transitive closure of all action relations. In chapter 4 we showed that to implement the action description assumptions of minimal change and maximal qualification, we can filter the set of modal action models and arrive at the intended ones, i.e. the ones obeying these description assumptions. For instance, minimization of change filters out models where individual action applications change less. We showed for several sub-formalisms that the application of action description assumptions corresponds to the application of sets of extension formulas in the same language. This brings back any action description under an intended model interpretation to an extended action description under a standard modal action logic semantics. It is clear that on any model of such a set of standard modal action logic formulas, we can interpret the temporal formulas defined in chapter 3. In this sense, no problems arise.

Note however that adopting an intended interpretation with respect to action description formulas may have temporal consequences. For instance, minimization of change may cause some action of a sequence of actions to reach a closer state, from where the successor action in the sequence is not possible any longer. This shows that minimization of change may alter the possible course of events, and thereby, the temporal information stored in models. Also maximization with respect to qualifications has an effect on the temporal information contained in models: maximization of qualifications actually means that actions are possible by default. So, along with the maximization of action possibilities, we may say that possible futures are maximized. A

specifier using the action description assumptions should be aware of these temporal consequences of intended interpretations.

6.3 Action description assumptions and norms

We defined deontic action logics as reductions to plain modal action logics using violation propositions V_P, V_F and V_O. We see no problems in combining this approach to normative reasoning with the intended model approach we developed for reasoning under action description assumptions in chapter 4. We can combine the approaches freely, as long as the violation propositions V_P, V_F and V_O are left out of the minimization strategy for minimal change. Then, also in extensions of action descriptions corresponding to these minimization and maximization strategies, these propositions are not allowed to occur.

However, it might be interesting to consider what happens if we do allow the interference with minimal change. Then we would have a situation where by default violations and non-violations are preserved.

6.4 Action norms and time

We can easily combine the temporal logics of chapter 3 and the normative logics of chapter 5. For instance, in the μ^η-calculus we can reason about the mix of dynamic, temporal and normative properties: in chapter 3 we showed how to express mixed dynamic / temporal reasoning in this calculus, and in chapter 5 we showed how to reduce deontic reasoning to reasoning in the calculus. But, we should not be so uncareful as to claim that we can suitably model all aspects of this combined reasoning domain. After all, the interplay between (action) norm-violation and time is a delicate one. And there may be properties in this domain that are not expressible using our approach. In the following we undertake a short investigation.

The relation between norms and time is a central theme in the work of Maibaum. In an overview paper [120] of his (joint) work in deontic logic he defines two logic systems. The first uses a, what he calls, 'immediate' notion of obligation. An obligation is immediate if its fulfillment is not postponable to a future point in time, i.e. the obligation has to be fulfilled *now*, and we may not defer it and first do another action. In our setting, the goal obligation $O_{\odot}(\alpha)$ and the process obligation $O_{\rightsquigarrow}(\alpha)$ are also immediate obligations. Maibaum argues that the immediate notion of obligation is not very useful for specification. He says that, referring to industrial trials to the application

of his logic, 'in most situations, a requirement to perform some action in the future was more appropriate'.

Clearly, our approach enables us to specify non-immediate obligations. We can for instance specify that an obligation only holds after performance of some other action.

> 'if ψ holds, it is obliged to perform α immediately after (conditional on performance of) β': $\psi \rightarrow [\beta]O(\alpha)$

Or, an obligation holds at certain specific future moments:

> 'whatever (atomic) action is performed first, it is always obliged to perform α next': $AXO(\alpha)$
>
> 'if ψ holds, there is a possible future course of events where performing α is obliged forever'[1]: $\psi \rightarrow EGO(\alpha)$

In these examples we left unspecified which of the two action norm variants of chapter 5 we have in mind: effect norms or process norms. Note that we freely use the branching time operators of CTL: goal norms can be reduced to the plain relativized action modalities of chapter 2, for which we showed in chapter 3 how to combine them with branching time temporal logics such as CTL. And, for process norms we defined a reduction to the modal μ^{η}-calculus, that also subsumes CTL (definition 3.5.2).

An interesting question is whether these are actually the kind of properties referred to by Maibaum when he claims that 'in most situations, a requirement to perform some action in the future was more appropriate'. The central question is whether we consider the following two normative assertions to be equivalent or not: 'after some finite number of actions it is obliged to perform α' and 'it is obliged to perform α after some finite number of actions'. The first assertion fits neatly to the formula $AFO(\alpha)$ (on all paths, some time in future it is obliged to perform α) of our logic framework. But it is not clear how to represent the second assertion. We claim that this is because it embodies a semantic confusion.

Maibaum attempts a formalization of the second assertion. His approach involves the direct association of this type of obligations with liveness properties and the direct association of permissions with safety properties. This association is not strange. The obligation featuring in the above assertion is

[1] Note that this does not necessarily mean that on this path where α is obliged globally, no violations can occur.

essentially an obligation to comply to a liveness property: the obligation will have to be discharged at 'some' (finitely reachable) point in future, which is the 'good' thing liveness properties talk about (see the end of section 3.5). Likewise, a permission might be thought of as a property that, if it is not explicitly withdrawn at some point, is preserved over time. But as Maibaum indicates in the conclusion of his paper [120], this association with liveness and safety does not allow contrary to duty specification. For, if an obligation is considered to be a liveness property (or an obligation to obey a liveness property), its violation cannot occur after any finite number of actions. This points to an 'internal' semantic confusion of assertions of the type 'it is obliged to perform α after some finite number of actions'. The problem is that nothing is said about *at what point* in the future the obligation has to be discharged. This makes such 'obligations' too weak to be useful, since at any point during a future course of events reactive systems (agents) can postpone the obligation to yet another future point[2].

What is missing from assertions of the mentioned type is a *deadline* for the obligation. We take the notion of a 'deadline' here in a general sense, meaning that it is not necessarily specified in what in computer science is called 'real time'. A deadline states, for instance, that the obligation has to be fulfilled before a certain action or condition occurs. Note that for immediate obligations, we also have a deadline of this type: the obligation has to be fulfilled *now*, i.e. before any other action is performed. Adding a deadline to the mentioned assertion, gives something like: 'it is obliged to perform η before the condition φ occurs' (since we do not want to get involved in the delicate issue of what it means if the condition φ occurs *during* the execution of the action, from this point on we restrict ourselves to non-sequential actions η). It is an interesting question whether or not we can express this in our logic system (see also [58] for an approach to the specification of normative deadlines in deontic modal action logics). In a first attempt we may make a semantic identification with the following assertion:

> 'for all possible future courses of events, either η is obliged now, the last moment before φ occurs, or any moment in between': $\neg E(\neg O(\eta) U \varphi)$

A natural language description that is closer to the form of the formula is: 'it is not the case that there is a possible future course of events such that for all time points in between now and the point where φ occurs, there is no

[2] I am sure that some readers will suspect that α stands for 'finish ones PhD-thesis'.

obligation to perform η'. However, intuitively there is something wrong with the identification of 'it is obliged to perform η before the condition φ occurs' with 'either η is obliged now, the next moment, etc., or the last moment before φ occurs'. This can be seen by looking at the violation conditions of $\neg E(\neg O(\eta) U \varphi)$. In most models of the formula $\neg E(\neg O(\eta) U \varphi)$, violations may occur in states that are not pre-final with respect to the occurrence of φ. A model where η is obliged immediately, while the condition φ is *not* obeyed in the next state, satisfies it. Therefore, in this model the obligation can also be violated by the first action. This violation is counter-intuitive, since the obligation we are trying to find an expression for only demands that we have to perform η before φ occurs. Then, if φ does not occur yet, why should there be a violation?

We may ask ourselves then, what the 'right' violation conditions for the assertion 'it is obliged to perform η before the condition φ occurs' are. Apparently a violation can only occur in states that are pre-final with respect to the occurrence of the condition φ, since the performance of η can always be postponed to these points without violating the norm. It is thus justified to semantically identify the assertion 'for all possible future courses of events it is obliged to perform η before the condition φ occurs' with the following assertion:

> 'for all possible future courses of events it holds that if the action η has not yet been done in the state before φ occurs, there is an immediate obligation in that state to perform it': $\neg E(\wr^B \eta \, U \, (EX\varphi \wedge \neg O(\eta)))$

The natural language description that is closer to the form of the formula is: 'there is no future course of events where we can perform actions alternative to η until we reach a state where it is possible to do an action with φ as a possible result and where it is not obliged to perform η immediately'. In this property three of the four major research themes of this Ph.D. thesis meet: (1) we need the action complement of chapter 2 to make the obligation in the state that is pre-final with respect to the occurrence of φ conditional on the absence of actions η in the course of action leading to this state[3], (2) we need the temporal / dynamic until-operator of chapter 3 to express the temporal conditions of the involved actions, and (3) we need the obligation operators of chapter 5 to account for the normative conditions in the pre-final state.

[3]Note that this is not the only aspect of the deadline property where the action complement plays a role. In chapters 3 and 5 we discussed its crucial role in the specification of mixed dynamic / temporal modalities as such, and in obligation modalities, respectively.

Therefore, in a nutshell, this single formula demonstrates the usefulness of much of the work in this Ph.D. thesis, which makes this a good point to end it.

6.5 Final remarks

A very general conclusion to be drawn from this doctoral thesis concerns the importance of semantic analysis. The analysis of deontic deadlines in the previous section reveals that semantic content can be much more subtle than expected. These semantic issues may seem rather specialized material. But it is actually at the semantic level where design faults are easily made: formulas or programs may turn out to give rise to slightly other behavior than was intended or expected by the specifier. This emphasizes the importance of a logic framework in which these semantic issues can be represented and analyzed. And using a formal logic system with a clear model-theoretic semantics to specify such semantically critical system properties, may help to reduce the tension that can exist between what is intended by a specifier and the actual behavior displayed by his system design.

Finally, we want to confess that this Ph.D. thesis has one apparent omission: a worked out case study that demonstrates that the developed logic framework is actually useful in practical situations. We decided not to include such a worked out example of the specification and verification of a reactive system, since we felt that demonstrating a serious application of the theory developed in this thesis is enough work for a separate Ph.D. thesis.

Bibliography

[1] M. Abadi and L. Lamport. Conjoining specifications. *ACM Transactions on Programming Languages and Systems*, 17(3):507–534, 1995.

[2] C.E. Alchourrón. A sketch of logic without truth. In *The 2nd International Conference on Artificial Intelligence and Law*, pages 165–179. ACM Press, 1989.

[3] C.E. Alchourrón. Philosophical foundations of deontic logic and the logic of defeasible conditionals. In J.-J.Ch. Meyer and R.J. Wieringa, editors, *Deontic Logic in Computer Science: Normative System Specification*, pages 43–84. John Wiley and Sons, 1993.

[4] B. Alpern and F.B. Schneider. Recognizing safety and liveness. *Distributed Computing*, 2:117–126, 1987.

[5] A.R. Anderson. A reduction of deontic logic to alethic modal logic. *Mind*, 67:100–103, 1958.

[6] L. Åqvist. Deontic logic. In D.M. Gabbay and F. Guenthner, editors, *Handbook of philosophical logic, vol. II*, pages 605–714. D. Reidel Publishing Company, 1984.

[7] J.C.M. Baeten and W.P. Weijland. *Process Algebra*, volume 18 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1990.

[8] P. Balbiani and D. Vakarelov. Iteration-free PDL with intersection: a complete axiomatization. *Fundamenta Informaticae*, 45(3):173–194, 2001.

[9] H. Barringer, R. Kuiper, and A. Pnueli. Now you may compose temporal logic specifications. In *Proceedings of the 16th Annual ACM Symposium on Theory of Computing*, pages 51–63, 1984.

[10] M. von der Beeck. A comparison of Statecharts variants. In H. Langmaack, W.P. de Roever, and J. Vytopil, editors, *Formal Techniques in Real-Time and Fault-Tolerant Systems*, volume 863 of *Lecture Notes in Computer Science*, pages 128–148. Springer, 1994.

[11] N. Belnap and M. Perloff. Seeing to it that: a canonical form for agentives. *Theoria*, 54(3):175–199, 1988.

[12] N. Belnap, M. Perloff, and M. Xu. *Facing the future*. Oxford University Press, 2001.

[13] B. Bennet, A.G. Cohn, F. Wolter, and M. Zakharyaschev. Multi-dimensional multi-modal logics as a framework for spatio-temporal reasoning. *Applied Intelligence*. forthcoming.

[14] B. Bennett, C. Dixon, M. Fisher, E. Franconi, I. Horrocks, U. Hustadt, and M. de Rijke. Combinations of modal logics. *Journal of AI Reviews*, 17(1):1–20, 2002.

[15] J.F.A.K. van Benthem. Minimal deontic logics. *Bulletin of the Section of Logic*, 8(1):36–42, 1979.

[16] J.F.A.K. van Benthem. Correspondence theory. In D.M. Gabbay and F. Guenthner, editors, *Handbook of philosophical logic, vol. II*. D. Reidel Publishing Company, 1984.

[17] J.F.A.K. van Benthem. Programming operations that are safe for bisimulation. *Studia Logica*, 60(2):311–330, 1998.

[18] A. Biere. mu-cke - efficient mu-calculus model checking. In O. Grumberg, editor, *International Conference on Computer-Aided Verification (CAV'97)*, volume 1254 of *Lecture Notes in Computer Science*, pages 468–471. Springer, 1997.

[19] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2001.

[20] A. Borgida, J. Mylopoulos, and R. Reiter. On the frame problem in procedure specifications. *IEEE Transactions on Software Engineering*, 21:785–798, 1995.

[21] S.-E. Bornscheuer and M. Thielscher. Representing concurrent actions and solving conflicts. *Journal of the Interest Group in Pure and Applied Logics (IGPL). Special issue of the ESPRIT project MEDLAR*, 4(3):355–368, 1996.

[22] J.C. Bradfield. *Verifying Temporal Properties of Systems*. Birkhäuser Boston, Massachussetts, 1992.

[23] J.C. Bradfield. On the expressivity of the modal mu-calculus. In C. Puech and R. Reischuk, editors, *Proceedings 13th Annual Symposium on Theoretical Aspects of Computer Science (STACS '96)*, volume 1046 of *Lecture Notes in Computer Science*, pages 479–490. Springer, 1996.

[24] J.C. Bradfield and C. Stirling. Verifying temporal properties of processes. In J.C.M. Baeten and J.W. Klop, editors, *CONCUR '90: Theories of Concurrency: Unification and Extension*, volume 458 of *Lecture Notes in Computer Science*, pages 115–125. Springer, 1990.

[25] J.C. Bradfield and C. Stirling. Modal logics and mu-calculi: An introduction. In J.A. Bergstra, A. Ponse, and S.A. Smolka, editors, *Handbook of Process Algebra*, pages 291–330. Elsevier Science, 2001.

[26] J.M. Broersen. A new action-base for dynamic deontic logics. In J. Horty and A.J.I. Jones, editors, *Pre-proceedings 6th International Workshop on Deontic Logic in Computer Science (DEON'02)*, pages 21–37, 2002.

[27] J.M. Broersen. Relativized action negation for dynamic logics. In *Pre-proceedings Advances in Modal Logic (AiML 2002)*, 2002. submitted to World Scientific.

[28] J.M. Broersen, M. Dastani, Z. Huang, and L.W.N. van der Torre. Trust and commitment in dynamic logic. In H. Shafazand and A. Min Tjoa, editors, *Eurasia-ICT 2002: Information and Communication Technology*, volume 2510 of *Lecture Notes in Computer Science*, pages 677–684. Springer, 2002.

[29] J.M. Broersen, M. Dastani, and L.W.N. van der Torre. Commitment and trust in dynamic logic. In A van den Bosch and H. Weigand, editors, *Proceedings of the 12th Belgium-Netherlands Artificial Intelligence Conference*, pages 3–11, 2000.

[30] J.M. Broersen, M. Dastani, and L.W.N van der Torre. Leveled commitment and trust in negotiation. In *Proceedings of the Autonomous Agents 2000 Workshop on Deception, Fraud and Trust in Agent Societies*, 2000.

[31] J.M. Broersen and R.J. Wieringa. Preferential semantics for action specifications in first-order modal action logic. In *Proceedings of the ECAI'98 Workshop on Practical Reasoning and Rationality (PRR'98)*, 1998.

[32] J.M. Broersen and R.J. Wieringa. A logic for the specification of multi-object systems. In P. Ciancarini, A. Fantechi, and R. Gorrieri, editors, *Formal Methods for Open Object-Based Distributed Systems*, pages 241–258. Kluwer Academic Publishers, 1999.

[33] J.M. Broersen, R.J. Wieringa, and R.B. Feenstra. Minimal semantics for action specifications in PDL. In J. Engelfriet and M. Spaan, editors, *Proceedings Accolade '96*, pages 15–30, Department of Mathematics and Computer Science, University of Amsterdam, 1997. Dutch Graduate School in Logic.

[34] J.M. Broersen, R.J. Wieringa, and J.-J.Ch. Meyer. Mu-calculus-based deontic logic for regular actions. In R. Demolombe and R. Hilpinen, editors, *Pre-proceedings 5th International Workshop on Deontic Logic in Computer Science (DEON'00)*, pages 43–61, 2000.

[35] J.M. Broersen, R.J. Wieringa, and J.-J.Ch. Meyer. A semantics for persistency in propositional dynamic logic. In J. Lloyd, V. Dahl, U. Furbach, M. Kerber, K.-K. Lau, C. Palamidessi, L. Moniz Pereira, Y. Sagiv, and P.J. Stuckey, editors, *Proceedings First International Conference on Computational Logic (CL2000)*, volume 1861 of *Lecture Notes in Artificial Intelligence*, pages 912–925. Springer, 2000.

[36] J.M. Broersen, R.J. Wieringa, and J.-J.Ch. Meyer. A fixed-point characterization of a deontic logic of regular action. *Fundamenta Informaticae*, 48(2, 3):107–128, 2001. Special Issue on Deontic Logic in Computer Science.

[37] J.M. Broersen, R.J. Wieringa, and J.-J.Ch. Meyer. The mutual exclusion problem in reasoning about action and change. In P. Doherty and M. Thielscher, editors, *Pre-proceedings NMR2002*, 2002.

[38] M.A. Brown. Doing as we ought: towards a logic of simply dischargeable obligations. In M.A. Brown and J. Carmo, editors, *Deontic logic, agency, and normative systems. Proceedings DEON '96*, pages 47–65. Springer, 1996.

[39] M.A. Brown and V. Goranko. An extended branching-time Ockhamist temporal logic. *Journal of Logic, Language, and Information*, 8(2):143–166, 1999.

[40] J. Carmo. Deontic database constraints, violation and recovery. *Studia Logica*, 1/2 (57):139–165, 1996.

[41] H.-N. Castañeda. *Thinking and Doing. The Philosophical Foundations of Institutions.* D. Reidel Publishing Company, 1975.

[42] H.-N. Castañeda. The paradoxes of deontic logic: the simplest solution to all of them in one fell swoop. In R. Hilpinen, editor, *New Studies in Deontic Logic: Norms, Actions and the Foundations of Ethics*, pages 37–85. D. Reidel Publishing Company, 1981.

[43] H.-N. Castañeda. Aspectual actions and davidson's theory of events. In B.P. McLaughlin E. LePore, editor, *Actions and Events: Perspectives on the Pholosophy of Donald Davidson*, pages 294–310. Basil-Blackwell, 1985.

[44] M.A. Castilho, O. Gasquet, and A. Herzig. Formalizing action and change in modal logic I: the frame problem. *Journal of Logic and Computation*, 9(5), 1999.

[45] E.S. Chang, Z. Manna, and A. Pnueli. Characterization of temporal property classes. In W. Kuich, editor, *Proceedings of the 19th International Colloquium on Automata, Languages, and Programming (ICALP 1992)*, volume 623 of *Lecture Notes in Computer Science*, pages 474–486. Springer, 1992.

[46] R.M. Chisholm. Contrary-to-duty imperatives and deontic logic. *Analysis*, 24:33–36, 1963.

[47] E.M. Clarke, E.A. Emerson, and A.P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 8(2), 1986.

[48] E.M. Clarke, O. Grumberg, and D. Long. Verification tools for finite-state concurrent systems. In *A decade of concurrency*, volume 803 of *Lecture Notes in Computer Science*, pages 124–175. Springer, 1993.

[49] R. Cleaveland and S. Sims. The NCSU concurrency workbench. In R. Alur and T. Henzinger, editors, *Computer-Aided Verification (CAV '96)*, volume 1102 of *Lecture Notes in Computer Science*, pages 394–397. Springer, 1996.

[50] P.R. Cohen and H.J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42(3):213–261, 1990.

[51] M. Dam. CTL* and ECTL* as fragments of the modal mu-calculus. *Theoretical Computer Science*, 126:77–96, 1994.

[52] R. Danecki. Nondeterministic propositional dynamic logic with intersection is decidable. In A. Skowron, editor, *Proceedings of the 5th Symposium on Computation Theory*, volume 208 of *Lecture Notes in Computer Science*, pages 34–53, 1984.

[53] D.C. Dennett. *The intentional stance*. The MIT Press, 1987.

[54] F.P.M. Dignum and J.-J.Ch. Meyer. Negations of transactions and their use in the specification of dynamic and deontic integrity constraints. In M.Z. Kwiatkowska, M.W. Shields, and R.M. Thomas, editors, *Semantics for Concurrency*, pages 61–80. Springer, 1990.

[55] F.P.M. Dignum, J.-J.Ch. Meyer, and R.J. Wieringa. Contextual permission: A solution to the free choice paradox. In A.J.I. Jones and M. Sergot, editors, *2nd International Workshop on Deontic Logic in Computer Science (DEON'94)*, pages 107–130. Norwegian Research Center for Computers and Law, 1994.

[56] F.P.M. Dignum, J.-J.Ch. Meyer, and R.J. Wieringa. Free choice and contextually permitted actions. *Studia Logica*, 57:193–220, 1996.

[57] E.A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, volume B: Formal Models and Semantics*, chapter 14, pages 996–1072. Elsevier Science, 1990.

[58] H. Weigand F. Dignum and E. Verharen. Meeting the deadline: on the formal specification of temporal deontic constraints. In Z.W. Ras and M. Michalewicz, editors, *Foundations of Intelligent Systems*, number 1079 in Lecture Notes in Artificial Intelligence, pages 243–252. Springer, 1996.

[59] M.J. Fischer and R.E. Ladner. Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences*, 18(2):194–211, 1979.

[60] D. Føllesdal and R. Hilpinen. Deontic logic: An introduction. In R. Hilpinen, editor, *Deontic Logic: Introductory and Systematic Readings*, pages 1–35. D. Reidel Publishing Company, 1971.

[61] N. Foo and D. Zhang. Extended propositional dynamic logic for expressing the indirect effect of actions. In *Advances in Modal Logic*, volume 3. CSLI Publications, 2001.

[62] N. Foo, D. Zhang, Y. Zhang, S. Chopra, and B.Q. Vo. Encoding solutions of the frame problem in dynamic logic. In T. Eiter, W. Faber, and M. Truszczyński, editors, *Logic Programming and Nonmonotonic Reasoning (LPNMR'01)*, volume 2173 of *Lecture Notes in Artificial Intelligence*, pages 240–253. Springer, 2001.

[63] J.W. Forrester. Gentle murder, or the adverbial Samaritan. *Journal of Philosophy*, 81(4):193–197, 1984.

[64] J.W. Forrester. *Being Good and Being Logical: Philosophical Groundwork for a New Deontic Logic*. M.E. Sharpe, 1996.

[65] D.M. Gabbay. The declarative past and imperative future: executable temporal logic for interactive systems. In *Temporal logic in specification*, volume 398 of *Lecture Notes in Computer Science*, pages 409–448. Springer, 1989.

[66] D.M. Gabbay, A. Pnueli, S. Shela, and J. Stavi. On the temporal analysis of fairness. In *Proceedings 7th ACM symposium on principles of programming languages*, pages 163–173, 1980.

[67] D.M. Gabbay and M. de Rijke, editors. *Frontiers of Combining Systems 2*, volume 7 of *Studies in Logic and Computation*. John Wiley and Sons, 2000.

[68] G. Gargov and S. Passy. A note on boolean modal logic. In P. Petkov, editor, *Mathematical Logic. Proceedings of The Summer School and Conference "Heyting'88"*, pages 311–321. Plenum Press, 1990.

[69] M. Gelfond and V. Lifschitz. Action languages. *Electronic Transaction on AI*, 16(3), 1998.

[70] G. De Giacomo. *Decidability of class-based knowledge representation formalisms*. PhD thesis, Universita' di Roma "La Sapienza", 1995.

[71] G. De Giacomo. Eliminating "converse" from converse PDL. *Journal of Logic, Language and Information*, 5:193–208, 1996.

[72] G. De Giacomo and X.J. Chen. Reasoning about nondeterministic and concurrent actions: A process algebra approach. In *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI'96)*, pages 658–663. The MIT Press, 1996.

[73] G. De Giacomo and M. Lenzerini. PDL-based framework for reasoning about actions. In *Proceedings of the 4th Congress of the Italian Association for Artificial Intelligence (AI*IA'95)*, volume 992 of *Lecture Notes in Artificial Intelligence*, pages 103–114. Springer, 1995.

[74] M.L. Ginsberg and D.E. Smith. Reasoning about action II: The qualification problem. *Artificial Intelligence*, 35:311–342, 1988.

[75] L. Giordano, A. Martelli, and C. Schwind. Dealing with concurrent actions in modal action logic. In H. Prade, editor, *Proceedings 13th European Conference on Artificial Intelligence (ECAI'98)*, 1998.

[76] L. Giordano and C. Schwind. Towards a conditional logic of actions and causation. In P. Doherty and M. Thielscher, editors, *Pre-proceedings NMR2002*, 2002.

[77] R.J. van Glabbeek. *Comparative Concurrency Semantics and Refinement of Actions*, volume 109 of *CWI Tract*. CWI, Amsterdam, 1996. Second edition of dissertation.

[78] V. Goranko. Modal definability in enriched languages. *Notre Dame Journal of Formal Logic*, 31(1):81–105, 1990.

[79] E. Grädel. Why are modal logics so robustly decidable. *Bulletin of the EATCS*, 68:90–103, 1999.

[80] P. Grünwald. Causation and nonmonotonic temporal reasoning. In G. Brewka, C. Habel, and B. Nebel, editors, *KI-97: Advances in Artificial Intelligence*, volume 1303 of *Lecture Notes in Artificial Intelligence*, pages 159–170. Springer, 1997.

[81] P. Grünwald. *The Minimum Description Length Principle and Reasoning under Uncertainty*. PhD thesis, Universiteit van Amsterdam, Institute for Logic, Language and Computation, ILLC Dissertation Series 1998-03, 1998.

[82] S. Hanks and D. Mc Dermott. Default reasoning, nonmonotonic logics, and the frame problem. In *Proceedings of the National Conference on Artificial Intelligence (AAAI86)*, pages 328–333. Morgan Kaufmann Publishers, 1986.

[83] D. Harel. *First Order Dynamic Logic*, volume 68 of *Lecture Notes in Computer Science*. Springer, 1979.

[84] D. Harel. Statecharts: a visual formalism for complex systems. *Science of Computer Programming*, 8:231–274, 1987.

[85] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. The MIT Press, 2000.

[86] D. Harel, O. Kupferman, and M.Y. Vardi. On the complexity of verifying concurrent transition systems. In A.W. Mazurkiewicz and J. Winkowski, editors, *Proceedings 8th International Conference on Concurrency Theory (CONCUR '97)*, volume 1243 of *Lecture Notes in Computer Science*, pages 258–272. Springer, 1997.

[87] D. Harel and D. Peleg. Process logic with regular formulas. *Theoretical Computer Science*, pages 307–322, 1985.

[88] D. Harel and A. Pnueli. On the development of reactive systems. In K.R. Apt, editor, *Logics and Models of Concurrent Systems, volume F-13 of NATO ASI Series*, pages 477–498. Springer, 1985.

[89] D. Harel, A. Pnueli, J.P. Schmidt, and R. Sherman. On the formal semantics of statecharts. In *Proceedings Symposium on Logic in Computer Science*, pages 54–64. Computer Science Press, 1987.

[90] D. Harel and E. Singerman. Computation paths logic: An expressive, yet elementary, process logic (abridged version). In P. Degano, R. Gorrieri, and A. Marchetti-Spaccamela, editors, *Proceedings 24th International Colloquium on Automata, Languages and Programming (ICALP'97)*, volume 1256 of *Lecture Notes in Computer Science*, pages 408–418. Springer, 1997.

[91] A. Herzig and O. Rifi. Propositional belief base update and minimal change. *Artificial Intelligence*, 115(1):107–138, 1999.

[92] R. Hilpinen. Conditionals and possible worlds. In G. Fløstad, editor, *Contemporary Philosophy, a New Survey*, volume 1, pages 299–335. Martinus Nijhoff, 1981.

[93] R. Hilpinen. New studies in deontic logic. D. Reidel Publishing Company, 1981.

[94] R. Hilpinen. Actions in deontic logic. In J.-J.Ch. Meyer and R.J. Wieringa, editors, *Deontic Logic in Computer Science: Normative System Specification*, pages 85–100. John Wiley and Sons, 1993.

[95] J. Hintikka. Some main problems of deontic logic. In R. Hilpinen, editor, *Deontic Logic: Introductory and Systematic Readings*, pages 59–104. D. Reidel Publishing Company, 1971.

[96] J. Hintikka. Impossible worlds vindicated. *Journal of Philosophical Logic*, 4:475–484, 1975.

[97] C.A.R. Hoare. *Communicating Sequential Processes*. Prentice-Hall, 1985.

[98] W. van der Hoek. *Modalities for Reasoning about Knowledge and Quantities*. PhD thesis, Faculteit der Wiskunde en Informatica, Vrije Universiteit Amsterdam, 1992.

[99] J.F. Horty. *Agency and Deontic Logic*. Oxford University Press, 2001.

[100] C. Huizing, R. Gerth, and W.P. de Roever. Modeling Statecharts in a fully abstract way. In M. Dauchet and M. Nivat, editors, *13th Colloquium on Trees in Algebra and Programming (AAP'88)*, volume 299 of *Lecture Notes in Computer Science*, pages 271–294. Springer, 1988.

[101] L. Humberstone. Inaccessible worlds. *Notre Dame Journal of Formal Logic*, 24:346–352, 1983.

[102] A.J.I. Jones and M. Sergot. On the characterization of law and computer systems: The normative sytems perspective. In J.-J.Ch. Meyer and R.J. Wieringa, editors, *Deontic Logic in Computer Science: Normative System Specification*, pages 275–307. John Wiley and Sons, 1993.

[103] H. Kamp. Free choice permission. *Aristotelian Society Proceedings N.S.*, 74:57–74, 1973-1974.

[104] S. Kanger. New foundations for ethical theory. In R. Hilpinen, editor, *Deontic Logic: Introductory and Systematic Readings*, pages 36–58. D. Reidel Publishing Company, 1971.

[105] S. Kanger. Law and logic. *Theoria*, 38(3):105–132, 1972.

[106] H.A. Kautz. The logic of persistence. In *Proceedings of the National Conference on Artificial Intelligence (AAAI86)*, pages 401–405. Morgan Kaufmann Publishers, 1986.

[107] D. Kozen. Results on the propositional mu-calculus. *Theoretical Computer Science*, 27:333–354, 1983.

[108] D. Kozen and R. Parikh. An elementary proof of the completeness of PDL. *Theoretical Computer Science*, 14:113–118, 1981.

[109] D. Kozen and J. Tiuryn. Logics of programs. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, volume B: Formal Models and Semantics*, pages 789–840. Elsevier Science, 1990.

[110] R. Ladner. The computational complexitiy of provability in systems of propositional modal logic. *SIAM Journal on Computing*, 6:467–480, 1977.

[111] E.J. Lemmon and D.S. Scott. *The 'Lemmon Notes': an introduction to modal logic*. Blackwell, 1977.

[112] D. Lewis. Semantic analysis for dyadic deontic logic. In S. Stunland, editor, *Logical Theory and Semantical Analysis*, pages 1–14. D. Reidel Publishing Company, 1974.

[113] D. Lewis. *On the Plurality of Worlds*. Basil Blackwell, 1986.

[114] V. Lifschitz. Two components of an action language. *Annals of Mathematics and Artificial Intelligence*, 21(1), 1997.

[115] F. Lin and R. Reiter. State Constraints Revisited. *Journal of Logic and Computation*, 4(5):655–678, 1994. Special Issue on Action and Processes.

[116] L. Lindahl. *Position and Change - A Study in Law and Logic*. Number 112 in Synthese Library. D. Reidel Publishing Company, 1977.

[117] K. Lodaya, R. Parikh, R. Ramanujan, and P.S. Thiagarajan. A logical study of distributed transition systems. *Information and Computation*, 119:91–118, 1995.

[118] C. Lutz and U. Sattler. The complexity of reasoning with boolean modal logic. In *Advances in Modal Logic*, volume 3. CSLI Publications, 2000.

[119] C. Lutz and U. Sattler. The complexity of reasoning with boolean modal logic (extended version). Technical Report LTCS-Report 00-02, Aachen University of Technoligy, 2001.

[120] T.S.E. Maibaum. Temporal reasoning over deontic specifications. In J.-J. Ch. Meyer and R.J. Wieringa, editors, *Deontic Logic in Computer Science: Normative System Specification*, pages 141–202. John Wiley and Sons, 1993.

[121] D. Makinson. Stenius' approach to disjunctive permission. *Theoria*, 50:138–147, 1984.

[122] D. Makinson. On a fundamental problem of deontic logic. In P. McNamara and H. Prakken, editors, *Norms, Logics and Information Systems. New Studies on Deontic Logic and Computer Science*, pages 29–53. IOS Press, 1998.

[123] D. Makinson and L.W.N. van der Torre. Input-output logics. *Journal of Philosophical Logic*, 29:383–408, 2000.

[124] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems: Specification*. Springer, 1992.

[125] M. Marx. Relativized relation algebras. *Algebra Universalis*, 41:23–45, 1999.

[126] M. Marx and Y. Venema. Local variations on a loose theme: modal logic and decidability. In M.Y. Vardi and S. Weinstein, editors, *Finite Model Theory and its Applications*. Springer. forthcoming.

[127] M. Marx and Y. Venema. *Multi-Dimensional Modal Logic*, volume 4 of *Applied Logic Series*. Kluwer Academic Publishers, 1997.

[128] J. McCarthy. Ascribing mental qualities to machines. *Philosophical Perspectives in Artificial Intelligence*, 1979.

[129] J. McCarthy. Circumscription—a form of non-monotonic reasoning. *Artificial Intelligence*, 13:27–39, 1980.

[130] L.T. McCarty. Modalities over actions I. Model theory. In J. Doyle, E. Sandewall, and P. Torasso, editors, *Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning (KR'94)*, pages 437–448. Morgan Kaufmann Publishers, 1994.

[131] R. van der Meyden. A clausal logic for deontic action specification. In *Proceedings International Logic Programming Symposium*. The MIT Press, 1991.

[132] R. van der Meyden. The dynamic logic of permission. *Journal of Logic and Computation*, 6(3):465–479, 1996.

[133] J.-J.Ch Meyer. Free choice permissions and ross' paradox: internal vs external non-determinism. Technical Report IR-130, Faculty of Mathematics and Computer Science, Vrije Universiteit, Amsterdam, 1987.

[134] J.-J.Ch Meyer. A simple solution to the "deepest" paradox in deontic logic. *Logique et Analyse*, (117-118):81–90, 1987.

[135] J.-J.Ch. Meyer. A different approach to deontic logic: Deontic logic viewed as a variant of dynamic logic. *Notre Dame Journal of Formal Logic*, 29:109–136, 1988.

[136] J.-J.Ch. Meyer and P. Doherty. Preferential action semantics (preliminary report). In J.-J.Ch. Meyer and P.-Y. Schobbens, editors, *Formal Models of Agents*, volume 1760 of *Lecture Notes in Artificial Intelligence*, pages 187–201. Springer, 1999.

[137] J.-J.Ch. Meyer and R.J. Wieringa. Deontic logic: a concise overview. In J.-J.Ch. Meyer and R.J. Wieringa, editors, *Deontic Logic in Computer Science: Normative System Specification*, pages 3–16. John Wiley and Sons, 1993.

[138] R. Milner. *A Calculus of Communicating Systems*, volume 92 of *Lecture Notes in Computer Science*. Springer, 1980.

[139] R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.

[140] R. De Nicola, A. Fantechi, S. Gnesi, and G. Ristori. An action-based framework for verifying logical and behavioural properties of concurrent systems. *Computer Networks and ISDN Systems*, 25:761–778, 1993.

[141] R. De Nicola and F. Vaandrager. Action versus state based logics for transition systems. In I. Guessarian, editor, *Semantics of Concurrency*, volume 469 of *Lecture Notes in Computer Science*, pages 407–419. Springer, 1990.

[142] D. Nute. Apparent obligation. In D. Nute, editor, *Defeasible Deontic Logic*, pages 287–315. Kluwer Academic Publishers, 1997.

[143] E. Orlowska. Dynamic logic with program specification and its relational proof system. *Journal of Applied Non-Classical Logics*, 3(2):147–171, 1993.

[144] R. Parikh. The completeness of propositional dynamic logic. In *Proceedings 7th Symposium on Mathematical Foundations of Computer Science*, volume 64 of *Lecture Notes in Computer Science*, pages 403–415. Springer, 1978.

[145] D. Park. Concurrency and automata on infinite sequences. In P. Deussen, editor, *Proceedings 5th GI-Conference on Theoretical Computer Science*, volume 104 of *Lecture Notes in Computer Science*, pages 167–183. Springer, 1981.

[146] S. Passy and T. Tinchev. An essay in combinatory dynamic logic. *Information and Computation*, 93:263–332, 1991.

[147] D. Peleg. Communication in concurrent dynamic logic. *Journal of Computer and System Sciences*, 35:23–58, 1987.

[148] D. Peleg. Concurrent dynamic logic. *Journal of the ACM*, 34:450–479, 1987.

[149] A. Pnueli. The temporal logic of programs. In *Proceedings of the 18th IEEE Symposium on the Foundations of Computer Science (FOCS-77)*, pages 46–57. IEEE Computer Society Press, 1977.

[150] A. Pnueli and M. Shalev. What is in a step: on the semantics of statecharts. In T. Ito and A.R. Meyer, editors, *Theoretical Aspects of Computer Software*, volume 526 of *Lecture Notes in Computer Science*, pages 244–264. Springer, 1991.

[151] I. Pörn. *Action Theory and Social Science: Some Formal Models*. D. Reidel Publishing Company, 1977.

[152] V.R. Pratt. Semantical considerations on Floyd-Hoare logic. In *Proceedings 17th IEEE Symposium on the Foundations of Computer Science*, pages 109–121. IEEE Computer Society Press, 1976.

[153] H. Prendinger and G. Schurz. Reasoning about action and change, a dynamic logic approach. *Journal of Logic, Language and Information*, 5:209–245, 1996.

[154] A.N. Prior. *Past, Present, and Future*. Clarendon Press, 1967.

[155] R. Reiter. The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In V. Lifschitz, editor, *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy*. Academic Press, 1991.

[156] M. de Rijke. A system of dynamic modal logic. *Journal of Philosophical Logic*, 27:109–142, 1998.

[157] J.A. Robinson. Computational logic: Memories of the past and challanges for the future. In J. Lloyd, V. Dahl, U. Furbach, M. Kerber, K.-K. Lau, C. Palamidessi, L. Moniz Pereira, Y. Sagiv, and P.J. Stuckey, editors, *Proceedings First International Conference on Computational Logic (CL2000)*, volume 1861 of *Lecture Notes in Artificial Intelligence*, pages 1–24. Springer, 2000.

[158] R. van Rooy. Permission to change. *Journal of Semantics*, 17(2), 2000.

[159] A. Ross. Imperatives and logic. *Theoria*, 7:53–71, 1941.

[160] L. Royakkers and F.P.M. Dignum. Giving permission implies giving choice. In E. Schweighofer, editor, *Proceedings 8th international conference and workshop on database and expert system applications*, 1997.

[161] J. Rushby. Calculating with requirements. In *Proceedings 3rd IEEE International Symposium on Requirements Engineering*, pages 144–146. IEEE Computer Society Press, 1992.

[162] B. Russell. On denoting. *Mind*, 14:479–493, 1905. Reprinted in Bertrand Russell, Essays in Analysis, London: Allen & Unwin, 103-119, 1973.

[163] E. Sandewall and Y. Shoham. Non-monotonic temporal reasoning. In D.M. Gabbay, C.J. Hogger, and J.A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming-Epistemic and Temporal reasoning (Volume 4)*, pages 439–498. Clarendon Press, 1994.

[164] K. Segerberg. Two-dimensional modal logic. *Journal of Philosophical Logic*, 2:77–96, 1973.

[165] K. Segerberg. Bringing it about. *Journal of Philosophical Logic*, 18(4):327–347, 1989.

[166] K. Segerberg. Getting started: Beginnings in the logic of action. *Studia Logica*, 51, 1992.

[167] K. Segerberg. Outline of a logic of action. Technical Report 5–2000, Department of Philosophy University of Uppsala, 2000.

[168] M. Sergot and F. Richards. On the representation of action and agency in the theory of normative positions. *Fundamenta Informaticae*, 34, 2001. Special issue, proceedings DEON2000.

[169] L. Spalazzi and P. Traverso. A dynamic logic for acting, sensing and planning. *Journal of Logic and Computation*, 10(6):787–821, 2000.

[170] P.A. Spruit, R.J. Wieringa, and J.-J.Ch. Meyer. Axiomatization, declarative semantics and operational semantics of passive and active updates in logic databases. *Journal of Logic and Computation*, 5(1):27–50, 1995.

[171] B. Steffen, R. Cleaveland, and J. Parrow. The concurrency workbench: A semantics–based verification tool for finite state systems. *ACM Transactions on Programming Languages and Systems, TOPLAS*, 15:36–72, 1993.

[172] C. Stirling. Modal and temporal logics. In S. Abramsky, D.M. Gabbay, and T.S.E. Maibaum, editors, *Handbook of Logic in Computer Science*, pages 477–563. Oxford University Press, 1992.

[173] C. Stirling. Modal and temporal logics for processes. In *Banff Higher Order Workshop*, volume 1043 of *Lecture Notes in Computer Science*, pages 149–237. Springer, 1996.

[174] A. Tarski. On the calculus of relations. *Journal of Symbolic Logic*, 6:73–89, 1941.

[175] A. Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5, 1955.

[176] M. Thielscher. Qualified ramifications. In B. Kuipers and B. Webber, editors, *Proceedings of the 14th National Conference on Artificial Intelligence (AAAI)*. The MIT Press, 1997.

[177] M. Thielscher. Ramification and causality. *Artificial Intelligence*, 89(1–2):317–364, 1997.

[178] M. Thielscher. From situation calculus to fluent calculus: State update axioms as a solution to the inferential frame problem. *Artificial Intelligence*, 111(1–2):277–299, 1999.

[179] L.W.N. van der Torre. *Reasoning about Obligations: Defeasibility in Preference-based Deontic Logic*. PhD thesis, Erasmus University Rotterdam, 1997.

[180] L.W.N. van der Torre and Y.H. Tan. The many faces of defeasibility in defeasible deontic logic. In D. Nute, editor, *Defeasible Deontic Logic*, pages 79–121. Kluwer Academic Publishers, 1997.

[181] A.M. Turing. The word problem in semi-groups with cancellation. 52, 1950.

[182] Y. Venema. Modal definability, purely modal. In J. Gerbrandy, M. Marx, M. de Rijke, and Y. Venema, editors, *JFAK. Essays dedicated to Johan van Benthem on the occasion of his 50th birthday (CD-Rom)*. Vossiuspers AUP, 1999.

[183] P. Vranas. New foundations for deontic logic: a preliminary sketch. In J. Horty and A.J.I. Jones, editors, *Pre-proceedings 6th International Workshop on Deontic Logic in Computer Science (DEON'02)*, 2002.

[184] I. Walukiewicz. Completeness of Kozen's axiomatisation of the propositional μ-calculus. In *Proceedings LICS'95*, pages 14–24, 1995.

[185] R.J. Wieringa and J.-J.Ch. Meyer. Actors, actions, and initiative in normative system specification. *Annals of Mathematics and Artificial Intelligence*, 7:289–346, 1993.

[186] R.J. Wieringa and J.-J.Ch. Meyer. Applications of deontic logic in computer science: A concise overview. In J.-J.Ch. Meyer and R.J. Wieringa, editors, *Deontic Logic in Computer Science: Normative System Specification*, pages 17–40. John Wiley and Sons, 1993.

[187] P. Wolper. On the relation of programs and computations to models of temporal logic. In B. Banieqbal, H. Barringer, and A. Pnueli, editors, *Temporal Logic in Specification*, volume 398 of *Lecture Notes in Computer Science*, pages 75–123. Springer, 1989.

[188] M. Wooldridge. *Reasoning about rational agents*. Intelligent robotics and autonomous agents. The MIT Press, 2000.

[189] G.H. von Wright. Deontic logic. *Mind*, 60:1–15, 1951.

[190] G.H. von Wright. *Norm and action; a logical enquiry*. International Library of Philosophy and Scientific Method. Routledge & Kegan Paul, 1963.

[191] G.H. von Wright. Deontic logic and the theory of conditions. In R. Hilpinen, editor, *Deontic Logic: Introductory and Systematic Readings*, pages 159–177. D. Reidel Publishing Company, 1971.

[192] G.H. von Wright. On the logic of norms and actions. In R. Hilpinen, editor, *New Studies in Deontic Logic*, pages 3–35. D. Reidel Publishing Company, 1981.

[193] G.H. von Wright. Norms, truth and logic. In A.A. Martino, editor, *Deontic Logic, Computational Linguistics and Legal Information Systems (Volume 2)*, pages 3–20. North-Holland Publishing Company, 1982.

[194] G.H. von Wright. Deontic logic - as I see it. In P. McNamara and H. Prakken, editors, *Norms, Logics and Information Systems. New Studies on Deontic Logic and Computer Science*, pages 15–25. IOS Press, 1999.

[195] A. Zanardo and J. Carmo. Ockhamist computational logic: Past-sensitive necessitation in CTL. *Journal of Logic and Computation*, 3(3):249–268, 1993.

Abstract

This Ph.D. thesis focuses on modal logics whose inference mechanisms can be of assistance in the process of design and verification of reactive systems. We argue that the intuitions for such logics follow from the specific 'view' on system properties adopted by a specifier. The first and most important aspect of this 'specifier view' we assume, is that a system under development can be seen as a collection of coherent actions that together are to produce the system behavior the specifier has in mind. This choice implies that to model the reasoning involved in the initial stages of design, we should consider logics of action.

We consider four aspects of reasoning about action that we deem relevant for the system specification domain: (1) action composition, (2) action and time, (3) action description assumptions (such as the frame problem), and (4) normative properties of action. Action composition is studied by considering certain specific action combinators. We concentrate on concurrent action composition and action complement. Concurrency is central to reactive systems in that, by definition, they operate concurrently with their environment. But, more in general we want to be able to reason about the effects of concurrent actions as related to the effects of their constituent parts. Action complement is studied in detail, because its definition forms a prerequisite for establishing logics of action and time, the economic specification of frame properties, and for the logics of action and norms. Existing definitions for action complement in the literature are discussed and we argue that they do not fit our purposes. As an alternative we define a relativized version of the action complement.

Traditionally, the temporal domain is deemed important for system specification. We investigate how in a modal action logic context, reasoning about action and reasoning about time can be combined. This results in mixed languages that enable the specification of mixed properties such as 'for all possible futures only actions α occur until the condition φ is met'.

A less well-considered problem for system specification is the modeling

of 'action description assumptions'. The most famous example of such an assumption is the frame assumption. We propose semantic solutions to the frame problem and the qualification problem, for different types of modal action description languages. We consider sequence and concurrency of action in this context.

We argue that the normative view is important for reasoning about reactive systems because it enables a specifier to represent valuable information about his system in a convenient way. A typical normative property has the form 'under condition ψ the system is obliged to perform α, but if it fails to do so, it should perform β to compensate for not having done α'. The formulation of such properties requires a certain 'stance' towards the specification of system properties. We call this stance the 'normative stance'. We develop two types of normative action logics: one for which only the result of an action is normed, and one for which also the way the result is obtained is normed. For both types of normative action logics we show how to define reductions of normative expressions to purely dynamic modalities. These reductions can be seen as formal counterparts of the normative stance.

The normative dimension, the action dimension and the temporal dimension are all interpreted on the same relational structures in modal action models. This allows free combination and nesting of temporal, dynamic (action), and normative formulas. In the conclusion of this Ph.D. thesis we show how this enables us to analyze the concept of a 'deontic deadline', which combines elements from each of these three reasoning domains.

Samenvatting (Dutch abstract)

Dit proefschrift behandelt modale actielogica's waarvan de inferentiemechanismes gebruikt kunnen worden voor de verificatie en het ondersteunen van het ontwerpproces van reactieve systemen. We stellen dat de intuïties voor deze logica's volgen uit de manier van kijken van degene die het systeem beschrijft. Het eerste, meest belangrijke door ons aangenomen aspect van deze manier van kijken, is dat een systeem kan worden gezien als een coherente verzameling van acties die samen het door de beschrijver gewenste gedrag produceren. Deze keuze impliceert dat we voor het modelleren van het soort van redeneren dat belangrijk is in de fase van het initiële systeemontwerp, moeten kijken naar actielogica's.

We onderzoeken vier aspecten van het redeneren over acties die ons belangrijk lijken voor het systeembeschrijvingsdomein: (1) actiecompositie, (2) actie en tijd, (3) actiebeschrijvingsaannames (zoals het 'frame'-probleem), en (4) normatieve eigenschappen van acties. We bestuderen actiecompositie door te kijken naar specifieke actiecombinatoren. We concentreren ons daarbij op de noties van parallellisme en actiecomplement. Parallellisme staat centraal in de studie van reactieve systemen, omdat een reactief systeem per definitie in parallelle samenhang met zijn omgeving opereert. Meer in het algemeen willen we in staat zijn te redeneren over de manier waarop effecten van parallelle acties samenhangen met de effecten van deelacties. We bestuderen het actiecomplement tot in detail, omdat een goede definitie daarvan een voorwaarde vormt voor het formuleren van logica's van actie en tijd, de economische expressie van frame-aannames, en de logica's van actie en normen. Bestaande definities voor het actiecomplement in de literatuur worden onder de loep genomen en ongeschikt bevonden. Als alternatief definiëren we een gerelativiseerde versie van het actiecomplement.

Traditioneel wordt het temporele domein als zeer belangrijk beschouwd

voor systeemspecificatie. We onderzoeken hoe in de context van modale actielogica redeneren over actie en tijd kan worden gecombineerd. Dit resulteert in gemengde talen die de specificatie van gemengde eigenschappen mogelijk maken, zoals 'voor alle mogelijke tijdslijnen geldt dat de acties α worden uitgevoerd totdat aan de conditie φ wordt voldaan'.

Een probleem dat minder aandacht heeft gekregen in de literatuur over systeemspecificatie is het modelleren van 'actiebeschrijvingsaannames'. Het meest bekende voorbeeld van zo'n aanname is de frame-aanname. We ontwikkelen semantische oplossingen voor het frame-probleem en het kwalificatieprobleem, voor verschillende soorten modale actiebeschrijvingstalen. We kijken naar sequentie en parallellisme van acties in deze context.

We stellen dat de normatieve manier van kijken belangrijk is voor het redeneren over reactieve systemen, omdat het een beschrijver in staat stelt op een handige manier waardevolle informatie over het systeem te representeren. Een typische normatieve eigenschap zoals die gebruikt kan worden bij systeemspecificatie, heeft de vorm 'wanneer aan de conditie ψ voldaan wordt, dan is het systeem verplicht de actie α uit te voeren, maar wanneer het dat niet doet, dan moet het daarvoor ter compensatie β uitvoeren'. Het formuleren van dit soort eigenschappen vergt een bepaalde 'houding' tegenover het specificeren van systeemeigenschappen. We noemen deze houding de 'normatieve houding'. We ontwikkelen twee soorten normatieve actielogica's: een waarvoor geldt dat alleen het resultaat van acties is genormeerd, en een waarvoor ook de manier waarop tot een resultaat wordt gekomen is genormeerd. Voor beide soorten logica's laten we zien hoe er een reductie van normatieve expressies naar puur dynamische modaliteiten gedefinieerd kan worden. Deze reducties kunnen worden gezien als de formele tegenhangers van 'de normatieve houding'.

De normatieve dimensie, de actiedimensie en de temporele dimensie worden alle geïnterpreteerd in termen van dezelfde relationele structuren in modale actiemodellen. Daardoor kunnen we temporele, dynamische en normatieve formules vrijelijk combineren. In de conclusie van dit proefschrift laten we zien dat dit ons in staat stelt het concept 'deontische deadline' te analyseren, een concept dat elementen van elk van de drie genoemde redeneerdomeinen in zich draagt.

SIKS Dissertation Series

1998-1	Johan van den Akker (CWI) DEGAS - An Active, Temporal Database of Autonomous Objects
1998-2	Floris Wiesman (UM) Information Retrieval by Graphically Browsing Meta-Information
1998-3	Ans Steuten (TUD) A Contribution to the Linguistic Analysis of Business Conversations within the Language/Action Perspective
1998-4	Dennis Breuker (UM) Memory versus Search in Games
1998-5	E.W. Oskamp (RUL) Computerondersteuning bij Straftoemeting
1999-1	Mark Sloof (VU) Physiology of Quality Change Modelling; Automated modelling of Quality Change of Agricultural Products
1999-2	Rob Potharst (EUR) Classification using decision trees and neural nets
1999-3	Don Beal (UM) The Nature of Minimax Search
1999-4	Jacques Penders (UM) The practical Art of Moving Physical Objects
1999-5	Aldo de Moor (KUB) Empowering Communities: A Method for the Legitimate User-Driven Specification of Network Information Systems
1999-6	Niek J.E. Wijngaards (VU) Re-design of compositional systems
1999-7	David Spelt (UT) Verification support for object database design
1999-8	Jacques H.J. Lenting (UM) Informed Gambling: Conception and Analysis of a Multi-Agent Mechanism for Discrete Reallocation.
2000-1	Frank Niessink (VU) Perspectives on Improving Software Maintenance
2000-2	Koen Holtman (TUE) Prototyping of CMS Storage Management
2000-3	Carolien M.T. Metselaar (UVA) Sociaal-organisatorische gevolgen van kennistechnologie; een procesbenadering en actorperspectief.

2000-4 Geert de Haan (VU) ETAG, A Formal Model of Competence Knowledge for User Interface Design

2000-5 Ruud van der Pol (UM) Knowledge-based Query Formulation in Information Retrieval.

2000-6 Rogier van Eijk (UU) Programming Languages for Agent Communication

2000-7 Niels Peek (UU) Decision-theoretic Planning of Clinical Patient Management

2000-8 Veerle Coupé (EUR) Sensitivity Analyis of Decision-Theoretic Networks

2000-9 Florian Waas (CWI) Principles of Probabilistic Query Optimization

2000-10 Niels Nes (CWI) Image Database Management System Design Considerations, Algorithms and Architecture

2000-11 Jonas Karlsson (CWI) Scalable Distributed Data Structures for Database Management

2001-1 Silja Renooij (UU) Qualitative Approaches to Quantifying Probabilistic Networks

2001-2 Koen Hindriks (UU) Agent Programming Languages: Programming with Mental Models

2001-3 Maarten van Someren (UvA) Learning as problem solving

2001-4 Evgueni Smirnov (UM) Conjunctive and Disjunctive Version Spaces with Instance-Based Boundary Sets

2001-5 Jacco van Ossenbruggen (VU) Processing Structured Hypermedia: A Matter of Style

2001-6 Martijn van Welie (VU) Task-based User Interface Design

2001-7 Bastiaan Schonhage (VU) Diva: Architectural Perspectives on Information Visualization

2001-8 Pascal van Eck (VU) A Compositional Semantic Structure for Multi-Agent Systems Dynamics.

2001-9 Pieter Jan 't Hoen (RUL) Towards Distributed Development of Large Object-Oriented Models, Views of Packages as Classes

2001-10 Maarten Sierhuis (UvA) Modeling and Simulating Work Practice BRAHMS: a multiagent modeling and simulation language for work practice analysis and design

2001-11 Tom M. van Engers (VUA) Knowledge Management: The Role of Mental Models in Business Systems Design

2002-01 Nico Lassing (VU) Architecture-Level Modifiability Analysis

2002-02 Roelof van Zwol (UT) Modelling and searching web-based document collections

2002-03 Henk Ernst Blok (UT) Database Optimization Aspects for Information Retrieval

2002-04 Juan Roberto Castelo Valdueza (UU) The Discrete Acyclic Digraph Markov Model in Data Mining

2002-05 Radu Serban (VU) The Private Cyberspace Modeling Electronic Environments inhabited by Privacy-concerned Agents

2002-06 Laurens Mommers (UL) Applied legal epistemology; Building a knowledge-based ontology of the legal domain

2002-07 Peter Boncz (CWI) Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications

2002-08 Jaap Gordijn (VU) Value Based Requirements Engineering: Exploring Innovative E-Commerce Ideas

2002-09 Willem-Jan van den Heuvel (KUB) Integrating Modern Business Applications with Objectified Legacy Systems

2002-10 Brian Sheppard (UM) Towards Perfect Play of Scrabble

2002-11 Wouter C.A. Wijngaards (VU) Agent Based Modelling of Dynamics: Biological and Organisational Applications

2002-12 Albrecht Schmidt (Uva) Processing XML in Database Systems

2002-13 Hongjing Wu (TUE) A Reference Architecture for Adaptive Hypermedia Applications

2002-14 Wieke de Vries (UU) Agent Interaction: Abstract Approaches to Modelling, Programming and Verifying Multi-Agent Systems

2002-15 Rik Eshuis (UT) Semantics and Verification of UML Activity Diagrams for Workflow Modelling

2002-16 Pieter van Langen (VU) The Anatomy of Design: Foundations, Models and Applications

2002-17 Stefan Manegold (UVA) Understanding, Modeling, and Improving Main-Memory Database Performance

2003-1 Heiner Stuckenschmidt (VU) Onotology-Based Information Sharing In Weakly Structured Environments