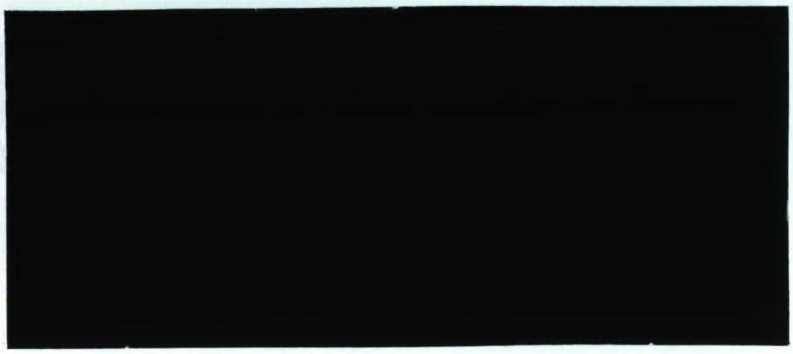
The role of explanations in inductive learning

Flach, P.A.

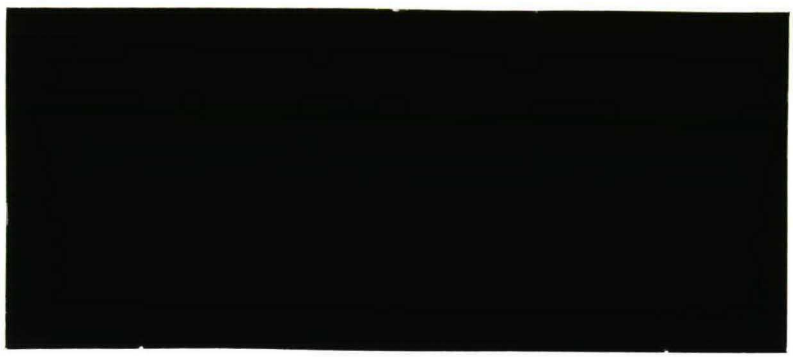
Publication date:
1991

Document Version
Publisher's PDF, also known as Version of record



ITK RESEARCH REPORT

STITUTE FOR LANGUAGE TECHNOLOGY AND ARTIFICIAL INTELLIGENCE INSTITUUT VOOR TAAL- EN KENNISTECHNOLOGIE



ITK Research Report
November 14, 1991

The role of explanations in inductive learning

Peter A. Flach

No. 30

Parts of this report have been or will be published in the following papers:

"Towards a logical theory of learning", *Proc. Sixth International Symposium on Methodologies for Intelligent Systems*, Z.W. Ras & M. Zemankova (eds.), Lecture Notes in Artificial Intelligence 542, pp. 510-519, Springer Verlag, 1991.

"The logic of explanations", *Proc. Computer Science in the Netherlands*, J. van Leeuwen (ed.), pp. 197-210, Stichting Mathematisch Centrum, 1991.

"A framework for Inductive Logic Programming", S. Muggleton (ed.), *Inductive Logic Programming*, Academic Press, 1992.

ISSN 0924-7807

The role of explanations in inductive learning

Peter A. Flach

ABSTRACT

Many AI problems involve dealing with incomplete knowledge, i.e. we have partial knowledge about an unknown model, and we have to find out what else is true about this model. Abduction and induction problems assume that our partial knowledge is given in the form of a background theory T and examples E, and the task is to extend T with a hypothesis H such that T and H together *explain* E. In this paper, we investigate this notion of explanation, concentrating on the logical properties which make it useful for induction. To this end, we set up a formal framework for analysing explanation as a kind of inference relation, analogous to the formal systems used to analyse non-monotonic inference relations. The two basic questions addressed in this paper are the following: (*i*) Given that we define explanation as logical implication wrt. some underlying logic, what properties of the underlying logic (such as monotonicity) influence the induction process? (*ii*) Are there alternative definitions of explanation, which could be used in induction? Our answer to the second question is affirmative: we suggest that *weak* explanation (logical consistency with respect to the underlying logic) is the appropriate notion to use in various induction problems, such as concept learning from incomplete examples, and induction of weak theories. On the practical side, we suggest how weak explanation can be used for deriving alternative generalisation and specialisation operators for induction of Horn theories, by manipulating the completion of such theories. Also, we show how background theory, needed for concept learning from incomplete examples, can be represented and handled on the meta-level.

Contents

1. Introduction

Many problems in AI can be characterised as *model inference* problems (Shapiro, 1981), i.e. inferring an unknown model M from a set of formulas F true in the model. Usually, the outcome of such a model inference problem is represented by an extended theory $F \cup H$ rather than a model (or a set of possible models). This type of reasoning is in general unsound: the *hypothesis* H may be false in M, if it does not follow logically from F. If nothing else is known, finding the right hypothesis is virtually impossible, because all that is known is that it must be consistent with F. Therefore, additional constraints are necessary, which depend on the problem at hand.

For instance, in diagnostic or *abductive* tasks, we have a set R of cause-effect rules and a set E of effects, and we are to find an extended theory $R \cup E \cup C$ which specifies the right causes C. We want these causes to *explain* the effects, which in general is taken to mean that the effects follow logically from the rules plus the causes: $R \cup C \models E$. This poses an additional constraint on the possible models, and thus on the possible causes. The set of possible models can be further restricted by only allowing specific formulas for R (e.g., Horn clauses), E and C (e.g., ground facts).

The task of *inductive learning* is very similar to abduction. We have a background theory T and a set of examples E, and we are to find a general rule H (e.g., a concept definition or a logic program) which *explains* the examples in some sense. Besides syntactical differences in the formulas involved, the main difference with abductive reasoning is that inductive learning is, in general, required to be incremental: several examples are supplied one at a time, and the hypothesis should be updated after each new example. This amounts to revising a theory to take new, conflicting information into account, without changing it too much; otherwise, the updated theory would now conflict with examples seen earlier. The equation *induction = abduction + revision* probably captures this viewpoint more succinctly, and also stresses the connections between these problems, which all involve reasoning with incomplete knowledge.

In this paper we concentrate on the role of explanations in inductive learning of logic programs. This subfield of Machine Learning has recently been called *Inductive Logic Programming* or ILP (Muggleton 1990, 1991). In an ILP setting, background theory T and inductive hypothesis H are sets of first-order clauses, as well as the set of examples E. Furthermore, explanation is usually defined as logical implication in the underlying logic, which is called the *base logic* in this paper. What counts as an explanation depends crucially on the properties of this base logic. For instance, if we know that Tweety is a bird, then the hypothesis 'birds fly unless they are abnormal' explains the fact that Tweety flies when interpreted as a default rule, but not when interpreted as a statement in first-order logic. In turn, the properties of explanations influence the properties of the induction process, such as convergence. In this paper, we propose a theoretical framework in which crucial properties of the induction process can be identified and related to the base logic. Throughout the paper, the consequence relation for this base logic will be denoted by $\vdash$.

This framework also allows us to introduce alternative definitions of explanation. In this paper, we investigate one alternative, i.e. consistency wrt. the base logic. So, we define: a hypothesis H is a *strong explanation* or *strong theory* for the examples E, given a background theory T, iff $T \cup H \vdash E$; H is a *weak explanation* or *weak theory* iff $T \cup H \cup E$ is consistent; we write $T \cup H \cup E \nvdash \square$. We will investigate the properties of strong and weak explanation, focussing on their utility wrt. the induction process, e.g. what property of explanation guarantees convergence? Also, we will give an abstract characterisation of

explanation, thus facilitating the comparison of weak and strong explanation (e.g., when is a weak explanation also a strong explanation). Induction based on weak/strong explanation will correspondingly be called weak/strong induction.

In the sequel, we use the following concepts and notation. We write $E \mid\!\sim_T H$ if E is explained by H given T, and $H_E = \{H \mid E \mid\!\sim_T H\}$ is the set of explanations of E (also referred to as the *Version Space* (Mitchell, 1982)). We assume the existence of a pre-order on this set, notation $H \geq_T H'$ (H is *as general as* H' wrt. T). As usual, this generality ordering is related to logical implication: $H \geq_T H'$ iff $T \cup H \models H'$. Since $\models$ is then required to be transitive (which in most cases implies monotonicity), we will not identify it with the base logic, but interpret it throughout the paper as classical two-valued logical consequence.

2. Properties of strong explanation

We start by studying the properties of strong explanation interpreted as a consequence relation. That is, we are interested in structural properties of $\mid\!\sim_T$, a binary relation between logical sentences (T is assumed to be fixed). Following other authors (Gabbay, 1985; Kraus *et al.*, 1990), these properties (such as reflexivity and transitivity) are written as inference rules in the style of Gentzen. Most of the rules studied by these authors are geared towards non-monotonic reasoning, and have limited significance in the present context. Instead, we will identify a number of new rules, particularly suited for describing properties of explanation.

2.1 Using a classical base logic

In this section, we use a classical two-valued base logic: $\alpha \mid\!\sim_T \beta$ is defined as $T, \beta \models \alpha$. In (Kraus *et al.*, 1990), five elementary properties are listed, which any reasonable consequence relation should satisfy. Of these properties, the following three are satisfied by strong explanation:

- **Reflexivity:** $$\alpha \mid\!\sim_T \alpha$$
- **Left Logical Equivalence:** $$\frac{T \models \alpha \leftrightarrow \beta \,,\, \alpha \mid\!\sim_T \gamma}{\beta \mid\!\sim_T \gamma}$$
- **Cut:** $$\frac{\alpha \wedge \beta \mid\!\sim_T \gamma \,,\, \alpha \mid\!\sim_T \beta}{\alpha \mid\!\sim_T \gamma}$$

However, the properties of **Right Weakening** and **Cautious Monotonicity** do not hold, which suggests that $\mid\!\sim_T$ as defined here is too weak to be called a (deductive) consequence relation. Of course, this is as it should be, since the problem of finding explanations is in general underconstrained (for instance, alternative explanations may be jointly inconsistent).

On the other hand, strong explanation has some properties which are not shared by traditional consequence relations: these are the properties which make explanation useful for inductive learning. To describe these properties, we introduce a number of new rules.

- **Explanation Strengthening:** $$\frac{T \models \gamma \rightarrow \beta \,,\, \alpha \mid\!\sim_T \beta}{\alpha \mid\!\sim_T \gamma}$$

This rule expresses that any γ as general as some strong explanation β for a set of examples α is also a strong explanation for α. Thus, the examples provide a lower bound on the set of possible hypotheses H_E, such that a hypothesis is in H_E iff it is above the boundary. This boundary can either be represented by the least general hypotheses not yet refuted, or the most general hypotheses already refuted; together with the generality ordering, it determines the set of still possible hypotheses[1].

- **Compositionality:** $$\frac{\alpha \mid\!\sim_T \gamma \,,\, \beta \mid\!\sim_T \gamma}{\alpha \wedge \beta \mid\!\sim_T \gamma}$$

This rule means that an inductive hypothesis can be checked against a set of examples by checking it against each example separately. This is a necessary condition if we don't want to remember every example during learning. Together, Explanation Strengthening and Compositionality allow for the derivation of the following rule:

[1] Assuming there are no infinite chains in the ordering.

- **Explanation Updating:** $$\frac{T \models \gamma' \rightarrow \gamma \,,\; \alpha \mid\!\sim_T \gamma \,,\; \beta \mid\!\sim_T \gamma'}{\alpha \wedge \beta \mid\!\sim_T \gamma'}$$

Thus, if γ is a hypothesis explaining the examples seen so far α but not the next example β, it can be replaced by some γ' which is as general as γ and explains β. If we want to describe the Version Space by its lower boundary, each γ in the boundary not explaining β should be replaced by every least general γ' which satisfies this condition. This is, in essence, Mitchell's *candidate elimination* approach; the fact that this generally applicable learning approach can be described by means of the above rules, suggests that these rules are both necessary and sufficient to describe the role of strong explanations in inductive learning.

Finally, we need a rule to guarantee convergence of the induction process:

- **Convergence:** $$\frac{T \models \alpha \rightarrow \beta \,,\; \alpha \mid\!\sim_T \gamma}{\beta \mid\!\sim_T \gamma}$$

This rule generalises Left Logical Equivalence, Cut and Or, and expresses that a strong explanation for α also explains anything implied by α. Consequently, strong induction enjoys the property that if a hypothesis γ is refuted by a conjunction of examples β, it cannot be an explanation for any larger conjunction of examples α. Thus, the set of possible hypotheses shrinks monotonically when learning proceeds. Without this property learning would be very difficult.

2.2 The role of the base logic

If we now vary the base logic, we can investigate the conditions under which these three properties hold. We simply rewrite the rules for Convergence, Compositionality and Explanation Strengthening by using the identity $\alpha \mid\!\sim_T \beta = T, \beta \vdash \alpha$.

Conv:	$\dfrac{T \models \alpha \rightarrow \beta \,,\; \alpha \mid\!\sim_T \gamma}{\beta \mid\!\sim_T \gamma}$	$\Rightarrow$	$\dfrac{T \models \alpha \rightarrow \beta \,,\; T, \gamma \vdash \alpha}{T, \gamma \vdash \beta}$	**Right Weakening**
Comp:	$\dfrac{\alpha \mid\!\sim_T \gamma \,,\; \beta \mid\!\sim_T \gamma}{\alpha \wedge \beta \mid\!\sim_T \gamma}$	$\Rightarrow$	$\dfrac{T, \gamma \vdash \alpha \,,\; T, \gamma \vdash \beta}{T, \gamma \vdash \alpha \wedge \beta}$	**And**
ES:	$\dfrac{T \models \gamma \rightarrow \beta \,,\; \alpha \mid\!\sim_T \beta}{\alpha \mid\!\sim_T \gamma}$	$\Rightarrow$	$\dfrac{T \models \gamma \rightarrow \beta \,,\; T, \beta \vdash \alpha}{T, \gamma \vdash \alpha}$	**Monotonicity**

The first two of these rules, Right Weakening and And, are guaranteed by any *cumulative* logic (the weakest possible logical system according to (Kraus *et al.*, 1990)). That is, strong induction is in principle possible for any cumulative base logic.

On the other hand, the third rule shows that Explanation Strengthening requires *monotonicity* of the base logic. That is, when inducing a non-monotonic strong theory, the Version Space will contain holes. For instance, when the background theory T contains the clause `flies(X):-bird(X),not abnormal(X)`, where `not` is implemented by negation as failure, and we want to extend the theory in order to explain the example `flies(tweety)`, we can add `bird(tweety)` but not the more general hypothesis `bird(tweety) & abnormal(tweety)`. Thus, the Version Space cannot simply be represented by its boundaries wrt. the generality ordering.

Several authors have proposed the use of non-monotonic logic in Inductive Logic Programming. Bain and Muggleton (1991) propose a Closed World Specialization Algorithm, which would specialise the clause `flies(X):-bird(X)` to `flies(X):-bird(X),not abnormal(X)` when it is known that `flies(emu)` is false in the intended interpretation. Ling (1991) provides an alternative algorithm, which would include the clause `flies(X):-bird(X),X≠emu` instead. The advantage of these approaches is, that

we can list exceptions to a rule, without having to specify that normal cases are not exceptions (which would be required if we used logical negation instead). However, as shown above, the added literals require special treatment. For instance, CIGOL (Muggleton & Buntine, 1988), equipped with the Closed World Specialization Algorithm, is perfectly happy to induce the theory {`bird(tweety)`, `flies(tweety)`, `abnormal(tweety)`, `flies(X):-bird(X),not abnormal(X)`}, which does not seem to capture the intuition behind the abnormality predicate.

In this section, we have outlined a framework for describing and analysing the role of explanations in induction, by viewing explanation as a consequence relation. We have identified the main properties of strong explanation, and how these are determined by the base logic. In the next section, we will do the same for weak explanation.

3. Properties of weak explanation

In this section, we let $\kvdash_T$ stand for weak explanation. First, we investigate weak explanation as consistency wrt. a classical two-valued base logic. Then, we study how the properties of weak explanation depend on the properties of the base logic.

3.1 Using a classical base logic

Perhaps surprisingly, Reflexivity does not hold for weak explanation: an inconsistent set of examples does not have a weak explanation. Also, Compositionality is invalid, implying that for weak induction a new hypothesis must be checked against the set of all previous examples, which must therefore be remembered.

We introduce two further rules for describing the properties of weak epxlanation.

- **Symmetry:** $$\frac{\alpha \kvdash_T \beta}{\beta \kvdash_T \alpha}$$

This rule holds since examples and hypotheses can be interchanged in the definition of weak explanation. This may seem counter-intuitive, but consider the following statements:

(*i*) There exists a bird which flies

(*ii*) Every bird flies

(*iii*) 'There exists a bird which doesn't fly' is false

(*iv*) 'No bird flies' is false

The Symmetry-rule states: if you accept (*ii*) as an explanation for (*i*), then you should also accept (*iv*) as an explanation for (*iii*) (since (*i*) and (*iv*) are logically equivalent, and so are (*ii*) and (*iii*)). Note that the second statement does not strongly explain the first, since it doesn't guarantee the existence of any bird.

The 'strong' rule Explanation Strengthening is replaced by its dual

- **Explanation Weakening:** $$\frac{T \models \beta \rightarrow \gamma \,,\, \alpha \kvdash_T \beta}{\alpha \kvdash_T \gamma}$$

This rule expresses that anything as **specific** as some weak explanation for a set of examples is also a weak explanation for them. Thus, the examples provide an **upper** bound on the Version Space, such that a hypothesis is in H_E iff it is below the boundary.

This reversal of the generality ordering can be explained by noting that a strong theory describes sufficient conditions, while a weak theory describes necessary conditions. *Sufficient* conditions for a concept definition are rules that can be used to classify individuals as instances of a concept, and *necessary* conditions classify them as non-instances. For instance, a Horn theory can only specify the sufficient conditions for a concept definition[2]; the necessary conditions must be expressed in a form which allows for the derivation of negative information. Such a form can for instance be obtained by *completing* a Horn theory (Clark, 1978). Now, if we have a concept definition which is both consistent (i.e., no instance is both positively and negatively classified) en complete (i.e., each instance is positively or negatively classified), then this definition can be split into two disjoint parts *Suff* and *Nec*, such that *Suff* classifies an instance positively if and only if *Nec* does not classify it negatively. In terms of logic:

---

[2]That is, under classical semantics; adopting a minimal model semantics amounts to interpreting a Horn theory as specifying both sufficient and necessary conditions.

$$Suff \vdash I \qquad \Leftrightarrow \qquad Nec \nvdash \neg I$$

where I denotes the statement that the instance belongs to the concept. In other words, *Suff* strongly explains I if and only if *Nec* weakly explains I. Moreover, the orderings of sufficient and necessary conditions are inversely related, since a theory is more general if it can derive more positive, and hence less negative, information. The learning of necessary conditions will be further explored in section 5.

3.2 The role of the base logic

If we again vary the base logic, we can investigate the conditions under which Convergence and Explanation Weakening hold. We can save some work by noting that each one of these can be rewritten into the other using Symmetry. Rewriting the rule for Convergence by using the identity $\alpha \mid\!\sim_T \beta = T, \alpha, \beta \nvdash \Box$[3] and taking the contrapositive, we obtain:

$$\textbf{Conv:} \qquad \frac{T \models \alpha \rightarrow \beta \,,\, \alpha \mid\!\sim_T \gamma}{\beta \mid\!\sim_T \gamma} \qquad \Rightarrow \qquad \frac{T \models \alpha \rightarrow \beta \,,\, T, \beta, \gamma \vdash \Box}{T, \alpha, \gamma \vdash \Box}$$

That is, T, β, γ does not have a model implies T, α, γ does not have a model. A sufficient condition for this is $T, \alpha \vdash \beta$, since then any model of T, α, γ is a model of T, β, γ, and this is in turn implied by $T \models \alpha \rightarrow \beta$, provided the base logic satisfies Reflexivity and Right Weakening (which any cumulative logic does).

We conclude that induction of weak theories in a cumulative base logic is theoretically always possible; moreover, the Version Space can always be represented by its boundaries (there are no holes). On the other hand, we lose Compositionality, implying that a new hypothesis must be checked against the entire set of previous examples. Since we have seen that strong induction of non-monotonic theories results in a Version Space with holes, weak induction provides an interesting alternative.

[3]Note that the identity $T, \alpha, \beta \mid\!\sim \Box = T, \alpha \mid\!\sim \neg\beta$ requires Contraposition (hence a classical monotonic base logic).

4. Abstract analysis

The purpose of this section is to study different logical systems of explanation and their relationships. As in (Kraus *et al.*, 1990), we define such systems in an abstract way by means of structural rules, without reference to the underlying base logic. We will define the systems **SC** (strong explanation wrt. cumulative base logic), **SM** (strong explanation wrt. monotonic base logic), **W** (weak explanation), and **CC** (which combines **SM** and **W**). The latter system is particularly interesting, since it relates strong and weak explanation. We will show how to use weak explanation to generate strong explanations, and also that weak explanation gives a more useful interpretation to a specific kind of examples.

4.1 Logical systems for strong explanation

The two lemmas in this section are duals of results by Kraus *et al.*, and have been obtained by rewriting $T, \alpha \vdash \beta$ to $\beta \mid\!\sim_T \alpha$. We start by noting that Reflexivity and Convergence together imply $T \models \alpha \rightarrow \beta \Rightarrow \beta \mid\!\sim_T \alpha$, i.e. a hypothesis explains all its logical consequences given T. Our first system will be called **SC**, for *strong cumulative explanation*, i.e. strong explanation wrt. a cumulative base logic. It consists of Reflexivity, Convergence, and the following three new rules:

- **Right Logical Equivalence**: $\dfrac{T \models \beta \leftrightarrow \gamma,\ \alpha \mid\!\sim_T \beta}{\alpha \mid\!\sim_T \gamma}$
- **Right Cut**: $\dfrac{\alpha \mid\!\sim_T \beta \wedge \gamma,\ \beta \mid\!\sim_T \gamma}{\alpha \mid\!\sim_T \gamma}$
- **Right Extension**: $\dfrac{\alpha \mid\!\sim_T \gamma,\ \beta \mid\!\sim_T \gamma}{\alpha \mid\!\sim_T \beta \wedge \gamma}$

Right Logical Equivalence states that logically equivalent explanations explain exactly the same things. **Right Cut** expresses that a part of an explanation, which is itself explained by another part, may be cut away from the explanation. **Right Extension** states that an explanation may be extended by anything it explains. Together, these latter two rules imply Compositionality.

LEMMA 1 (Kraus *et al.*, 1990). *Compositionality is a derived rule in SC.*
Proof. Suppose $\alpha \mid\!\sim_T \gamma$ and $\beta \mid\!\sim_T \gamma$; by Right Extension we have $\alpha \mid\!\sim_T \beta \wedge \gamma$. Also, because $\alpha \wedge \beta \wedge \gamma \models \alpha \wedge \beta$, we have $\alpha \wedge \beta \mid\!\sim_T \alpha \wedge \beta \wedge \gamma$. Using Right Cut gives $\alpha \wedge \beta \mid\!\sim_T \beta \wedge \gamma$, and since by assumption $\beta \mid\!\sim_T \gamma$, we can cut away β from the explanation to get $\alpha \wedge \beta \mid\!\sim_T \gamma$. □

As was shown before, assuming a cumulative monotonic base logic (satisfying Monotonicity) guarantees Explanation Strengthening. However, Kraus *et al.* show that a cumulative monotonic logic is strictly weaker than classical logic, satisfying Contraposition. Our next system **SM**, for *strong monotonic explanation*, models strong explanation wrt. a classical monotonic base logic. It consists of the rules of **SC** plus the following rule:

- **Contraposition**: $\dfrac{\alpha \mid\!\sim_T \beta}{\neg\beta \mid\!\sim_T \neg\alpha}$

LEMMA 2 (Kraus *et al.*, 1990). *Explanation Strengthening is a derived rule in SM.*
Proof. Suppose $T \models \gamma \rightarrow \beta$ and $\alpha \mid\!\sim_T \beta$; by Contraposition, it follows that $\neg\beta \mid\!\sim_T \neg\alpha$. Convergence gives $\neg\gamma \mid\!\sim_T \neg\alpha$, which finally results in $\alpha \mid\!\sim_T \gamma$ by Contraposition. □

4.2 Logical systems for weak explanation

In this section we develop a logical system for weak explanation. As in the previous section, this system is obtained by rewriting structural properties of the base logic into structural properties of explanation, in this case by rewriting $T, \alpha \vdash \beta$ to $\neg\beta \not\mid\!\sim_T \alpha$[4].

The system **W** consists of the previously introduced rules Symmetry, Convergence, Right Logical Equivalence, plus the following three *weak* counterparts of rules in **SC**:

- **Weak Reflexivity:** $\neg\alpha \not\mid\!\sim_T \alpha$
- **Weak Right Cut:** $\dfrac{\alpha \mid\!\sim_T \beta \wedge \gamma,\ \neg\beta \not\mid\!\sim_T \gamma}{\alpha \mid\!\sim_T \gamma}$
- **Weak Right Extension:** $\dfrac{\alpha \mid\!\sim_T \gamma,\ \neg\beta \not\mid\!\sim_T \gamma}{\alpha \mid\!\sim_T \beta \wedge \gamma}$

Weak Reflexivity and Convergence together imply $T \models \alpha \rightarrow \beta \Rightarrow \neg\beta \not\mid\!\sim_T \alpha$, i.e. no hypothesis explains the negation of any of its logical consequences, given T. We have the following result.

LEMMA 3. *Explanation Weakening is a derived rule in W.*
Proof. Suppose $T \models \beta \rightarrow \gamma$ and $\alpha \mid\!\sim_T \beta$; by Symmetry, it follows that $\beta \mid\!\sim_T \alpha$. Convergence gives $\gamma \mid\!\sim_T \alpha$, which finally results in $\alpha \mid\!\sim_T \gamma$ by Symmetry. □

Assuming Monotonicity of the base logic does not enrich this logical system, because it does not translate into rules previously underivable. We conclude that for weak explanation the precise nature of the base logic is immaterial, as long as it is cumulative.

4.3 Negative examples

It is customary in inductive learning to have, besides the set of *positive* examples P which are to be explained, a set of *negative* examples N, to be explained in a different way. Thus, an inductive learning problem requires two notions of explanation, $\mid\!\sim^+$ and $\mid\!\sim^-$, for positive and negative explanation, respectively (we omit the suffix T for readability). Usually, these two notions satisfy the equivalence

$$\alpha \mid\!\sim^+ \beta \Leftrightarrow \alpha \not\mid\!\sim^- \beta \qquad (*)$$

which means that, for a given hypothesis, each example is either positive or negative.

Note that if $\mid\!\sim$ denotes some notion of strong explanation used for positive examples, then its associated definition of weak explanation, $\mid\!\approx$, can be used for negative examples, because (*) then yields $\alpha \not\mid\!\sim^- \beta \Leftrightarrow \neg\alpha \mid\!\approx \beta$. This can be formalised as follows. Rewriting the rules for **SC**, **SM** and **W** using $\alpha \mid\!\sim \beta \Rightarrow \alpha \not\mid\!\sim \beta$, we obtain the

[4]Hence, $\neg$ should be interpreted as strong negation.

systems SC$^-$, SM$^-$ and W$^-$. Notice that Explanation Strengthening rewrites to Explanation Weakening and *vice versa*, just as we would expect[5]. We then have the following result.

LEMMA 4. *Define* $\neg\alpha \mid\!\sim^- \beta$ *iff* $\alpha \not\ll \beta$, *then* $\mid\!\sim^-$ *satisfies the rules of* **SM**$^-$ *iff* $\not\ll$ *satisfies the rules of* **W**.

Proof. Left to the reader. □

Not explaining α is different from (positively) explaining $\neg\alpha$, as will be clear from an example: if T = {`sparrow(sparky)`, `penguin(tweety)`}, then the Horn theory H_1 = {`sparrow(X)`$\rightarrow$`flies(X)`} does not explain `flies(tweety)`, but it does not explain $\neg$`flies(tweety)` either (using strong monotonic explanation). On the other hand, the theory H_2 = {`sparrow(X)`$\rightarrow$`flies(X)`, `penguin(X)`$\rightarrow\neg$`flies(X)`} does explain $\neg$`flies(tweety)`. Alternatively, using weak explanation, H_1 explains both `flies(tweety)` and $\neg$`flies(tweety)`, while H_2 explains $\neg$`flies(tweety)` but not `flies(tweety)`. In fact, learning weak explanations can be simulated by negating the examples, interchanging positive and negative examples, and then learning a strong explanation.

We would like to suggest that it can be fruitful to weaken equivalence (*), i.e. to allow one and the same example to be both positive and negative. For instance, suppose we are learning the concept 'sparrow', and we have two examples: Sparky the small, brown sparrow, and Flap the big, brown falcon. That is, we are looking for a theory positively explaining `small`$\wedge$`brown`$\rightarrow$`sparrow` and negatively explaining `big`$\wedge$`brown`$\rightarrow$`sparrow`. Now, if the teacher describes both Sparky and Flap incompletely, omitting their size, then the positive and negative example become identical. Thus, we are looking for a theory both positively and negatively explaining `brown`$\rightarrow$`sparrow`. In the case of weak explanation, this requirement can be fulfilled. In section 6, we will have a closer look at learning from such incompletely specified examples.

An analogous situation arises when we use strong explanation wrt. a non-monotonic base logic, i.e. $\alpha \mid\!\sim^+ \beta \Leftrightarrow \beta \vdash \alpha$. In that case, we can define either $\alpha \mid\!\sim^- \beta \Leftrightarrow \beta \not\vdash \alpha$ or $\alpha \mid\!\sim \beta \Leftrightarrow \beta, \neg\alpha \not\vdash \Box$. In the first case, we clearly have equivalence (*), but not in the second case. As an example, let T = {`bird(opus)`, `bird(tweety)`}, p = `flies(opus)`, n = `flies(tweety)`, and H = {`flies(X) :-bird(X), not abnormal(X)`}, then $p \mid\!\sim^+ H$ as desired, and $n \mid\!\sim^- H$ only for the second definition (because there is a model in which Tweety is abnormal, hence doesn't fly). Thus, we use the first definition if we explicitly want to include the abnormality of the negative examples in our theory, and the second definition otherwise.

4.4 Combining weak and strong explanation

In this section, we explore the relation between weak and strong explanation. It is shown that the system W can be obtained from SM by a simple transformation, and *vice versa*. That is, each notion of strong explanation defines a notion of weak explanation, and conversely. Next, we show that under certain conditions these corresponding notions of explanation are equivalent, i.e. a hypothesis strongly explains a set of examples iff it explains them weakly. Thus, weak explanation provides an alternative way of checking a potential explanation. However, the conditions for the equivalence of weak and strong explanation also mean that the generality ordering will not be very useful. Fortunately, there are ways to overcome this problem, such that weak explanation can be used for checking strong explanation, without losing the generality ordering.

---

[5]Thus, combining SM and SM$^-$ or W and W$^-$ results in the celebrated *Version Space* model (Mitchell, 1982): the set of possible hypotheses is $H_{P,N} = H_P \cap \overline{H_N}$, and if H_P has a lower/upper boundary, $H_{P,N}$ has both a lower and an upper boundary.

The following lemma shows how each notion of strong explanation defines a corresponding notion of weak explanation, and conversely. Throughout this section, we assume that $\neg$ stands for strong negation (i.e., satisfying idempotence: $\neg\neg\alpha \equiv \alpha$).

LEMMA 5. *Define* $\alpha \mid\!\!\sim_T \beta$ *iff* $\neg\alpha \not\vdash_T \beta$, *then* $\vdash_T$ *satisfies the rules of SM iff* $\mid\!\!\sim_T$ *satisfies the rules of W.*

Proof. Using the rewrite rule $\alpha \vdash_T \beta \Rightarrow \neg\alpha \not\mid\!\!\sim_T \beta$[6], each rule of **SM** rewrites (after re-arranging) to a rule of **W**: Convergence and Right Logical Equivalence rewrite to themselves, Reflexivity rewrites to Weak Reflexivity, Right Cut to Weak Right Extension, Right Extension to Weak Right Cut, and Contraposition rewrites to Symmetry. □

The correspondence between strong and weak explanation suggests the following question: are there conditions under which these two notions are equivalent (i.e. $\alpha \vdash_T \beta \Leftrightarrow \alpha \mid\!\!\sim_T \beta$)? In terms of our logical systems, is there a system stronger than both **SM** and **W**? The answer can be obtained by comparing Reflexivity, Right Cut and Right Extension in **SM** to their weak counterparts in **W**. The former can be derived from the latter if the following rule is added to **W**:

- **Consistent Explanation:** $\dfrac{\alpha \vdash_T \beta}{\neg\alpha \not\vdash_T \beta}$

Consistent Explanation expresses that a hypothesis cannot both explain an example and its negation. Conversely, the weak rules can be derived from their strong counterparts under the following rule:

- **Complete Explanation:** $\dfrac{\neg\alpha \not\vdash_T \beta}{\alpha \vdash_T \beta}$

which states that any hypothesis always explains an example or its negation.

The following result shows that adding both Consistent Explanation and Complete Explanation to either **W** or **SM** results in a system **CC**, for *Complete Consistent Explanation*, which is strictly stronger than both.

LEMMA 6. *In the presence of both Consistent Explanation and Complete Explanation, Contraposition and Symmetry are equivalent.*

Proof. Suppose $\alpha \vdash_T \beta$; Consistent Explanation implies $\neg\alpha \not\vdash_T \beta$, Symmetry implies $\beta \not\vdash_T \neg\alpha$, and Complete Explanation implies $\neg\beta \vdash_T \neg\alpha$. Conversely, $\alpha \vdash_T \beta$ implies $\neg\alpha \not\vdash_T \beta$ by Consistent Explanation, Contraposition implies $\neg\beta \not\vdash_T \alpha$, and Complete Explanation implies $\beta \vdash_T \alpha$.□

CC is interesting because of the equivalence $\alpha \vdash_T \beta \Leftrightarrow \neg\alpha \not\vdash_T \beta$, thus providing two different ways of checking explanations. However, **CC** itself is not very useful for induction, for the following reason. Since **CC** is stronger than both **W** and **SM**, every rule of the latter two is a rule of **CC**. In particular, **both** Explanation Strengthening and Explanation Weakening are rules of **CC**. That is, if $\alpha \vdash_T \beta$, then for every γ which is comparable to β wrt. $\geq$ (i.e., $\beta \models \gamma$ or $\gamma \models \beta$), $\alpha \vdash_T \gamma$. Thus, if we want to change an explanation in order to accomodate for a new example, the new explanation will be **incomparable** to the original explanation! We can get around this by using different representations for strong and weak explanations, as will be shown in section 5.

---

[6]Note that this transformation from strong to weak explanation takes the same form as its inverse; i.e., corresponding notions of strong and weak explanation are completely dual.

The system CC can also be constructed by means of the following two rules.

- **Consistent Example:** $\dfrac{\alpha \mid\!\sim_T \beta}{\alpha \not\mid\!\sim_T \neg\beta}$

Consistent Example expresses that an example cannot be explained by both a hypothesis and its negation.

- **Complete Example:** $\dfrac{\alpha \not\mid\!\sim_T \neg\beta}{\alpha \mid\!\sim_T \beta}$

Complete Example states that any example is always explained by a hypothesis or its negation.

LEMMA 7. *In the presence of either Symmetry or Contraposition, Consistent Example and Consistent Explanation are equivalent; so are Complete Example and Complete Explanation. In the presence of both Consistent Example and Complete Example, Contraposition and Symmetry are equivalent.*

Proof. Analogous to the proof of Lemma 6. □

In section 6, we give an example of a learning problem in which weak explanation provides an alternative way of checking explanations for complete examples. However, in the case of *incomplete* examples, weak explanation provides an alternative way of **interpreting** the examples.

5. Learning necessary conditions by weak explanation

In the previous section, we showed that in the system CC, every weak explanation is a strong explanation and *vice versa*. The intuitive interpretation of an explanation in CC is, that it consists of necessary and sufficient conditions which are equal. As was shown however, this reduces the generality ordering to the trivial ordering in which each explanation is only comparable with the empty explanation and the inconsistent explanation. Therefore, it is better to keep weak and strong explanations separate, and to transform one into the other if necessary. This will be discussed in the present section.

As before, let $\models_T$ denote strong explanation and $\models\!\!\!\ll_T$ denote the corresponding notion of weak explanation (i.e., $\alpha \models_T \beta \Leftrightarrow \neg\alpha \not\models\!\!\!\ll_T \beta$), and suppose every complete and consistent explanation can be split into two disjoint parts β and β^*, such that $\alpha \models_T \beta$ iff $\neg\alpha \not\models_T \beta^*$. If in addition * is idempotent, we have $\alpha \models_T \beta \Leftrightarrow \alpha \models\!\!\!\ll_T \beta^*$; thus, we can check $\alpha \models_T \beta$ by first transforming β to β^*, and then checking whether β^* weakly explains α. Furthermore, if $T \models \alpha \rightarrow \beta$ implies $T \models \beta^* \rightarrow \alpha^*$, we can generalise β by specialising β^*. For instance, let β be a Horn theory, and let β^* denote the augmentation obtained by predicate completion, i.e. the only-if parts of predicate definitions, then we can check whether α is strongly explained (logically implied) by β by checking whether α is weakly explained by (consistent with) β^*. The following example illustrates this correspondence.

Let `element(E,L)` be a predicate with intended interpretation: `E` occurs in the list `L`. Consider the hypothesis {`element(X,[X|Y])`}, stating that `element(X,Y)` is true if `X` is the first element of the list `Y`. This hypothesis is not a strong explanation for the example `element(2,[1,2,3])`. This can be proved by constructing the completion `∃Y1:Y=[X|Y1]:-element(X,Y)` of the initial hypothesis, and proving that the completed hypothesis is not a weak explanation of the example. Thus, we prove that they are logically inconsistent, by resolving `∃Y1:Y=[X|Y1]:-element(X,Y)`[7] with `element(2,[1,2,3])`, yielding the

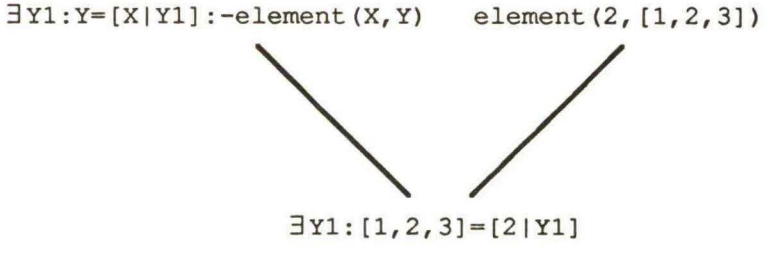


Figure 1. Proving the inconsistency of a completed hypothesis with an example.

formula `∃Y1:[1,2,3]=[2|Y1]` (fig. 1). This formula is unsatisfiable under the standard interpretation of = as syntactical identity (i.e. unification). A possible way to make the fomula satisfiable is by disjoining it with `∃Z∃Y2:[1,2,3]=[Z,2|Y2]` (which is `∃Z∃Y2:Y=[Z,X|Y2]` under the substitution {`X`→`2`, `Y`→`[1,2,3]`}). This amounts to specialising the completed formula to

[7] We retain the existential quantifier in order to avoid Skolem functors.

```
∃Y1∃Z∃Y2:Y=[X|Y1];Y=[Z,X|Y2]:-element(X,Y)
```

which in turn amounts to generalising the original hypothesis to {`element(X,[X|Y])`, `element(X,[Z,X|Y])`}.

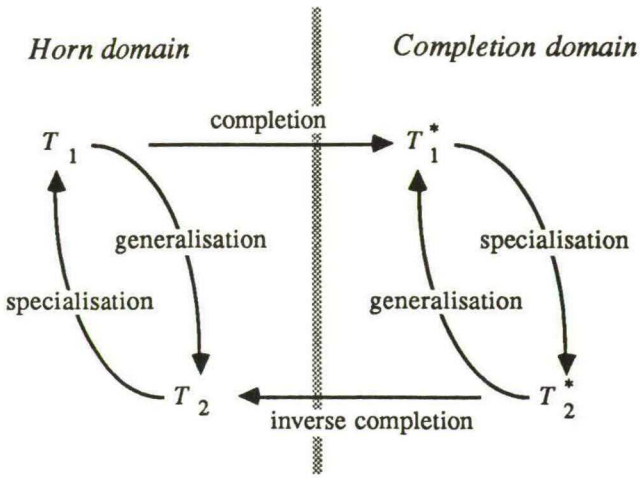


Figure 2. The duality between Horn theories and completed theories.

Thus, generalisation and specialisation can be related to each other by means of predicate completion, as depicted in fig. 2. This duality can be exploited in various ways. First of all, an operator for Horn theories could be applied to completed theories, yielding a new operator when transformed back to the Horn domain. For instance, a general-purpose specialisation operator for clausal theories such as Shapiro's (1981) refinement operator could be turned into a generalisation operator by performing completion-specialisation-inverse completion. Instead of implementing both generalisation and specialisation operators, a system might include only one type and realise the other by means such transformations.

Secondly, the completion domain might suggest specific operators which yield new and interesting operators when transformed to the Horn domain. For instance, in the above example the completed theory is specialised by adding a literal to a clause, which corresponds to adding a clause to the corresponding Horn theory. This also shows that the relation between specialisation in one domain and generalising in the other is not trivial and needs further investigation. Such a study will also lead to a better understanding of the relation between operators which change theories on the clause level (such as the Absorption operator (Muggleton & Buntine, 1988)), and those which operate on the literal level (such as refinement operators).

6. Concept learning from incomplete examples

In the previous section, we saw how weak explanation could be useful in the context of strong explanation, by assigning a second interpretation to Horn theories (as expressed by their completions) in order to achieve completeness of explanations. As was shown in section 4.4, the same effect can be achieved by assuming completeness of examples,which is expressed by the rule Complete Example. This rule states that any instance is always correctly classified by either a hypothesis or its negation. If this property does not hold, then some relevant information has been omitted from the description of the instance. Given this property, any weak explanation is also a strong explanation, and we might again use the former notion to implement the latter. However, in this section we are interested in *incomplete examples*: we will show that weak explanation assigns the right interpretation to incomplete examples, and we will show that complete examples can be represented just as incomplete examples, provided we add certain axioms to our background theory.

Consider a universe of birds, described by the properties colour (black, brown, golden) and size (small, big). By means of these properties, we can distinguish between blackbirds, falcons, sparrows and eagles[8] (fig. 3). In this figure, a *concept* is any subset of the universe. Throughout, we will assume that the concept can be described by means of the given properties, that is, we consider only unions of the smallest blocks. An *example* is a description of a member of the universe in terms of the given properties. Clearly, an example also corresponds to one or more blocks[9]. An example is *complete* if it corresponds to exactly one smallest block, and *incomplete* otherwise. A concept explains a complete example if and only if it contains the block associated with the example. We call this the *inclusion* condition.

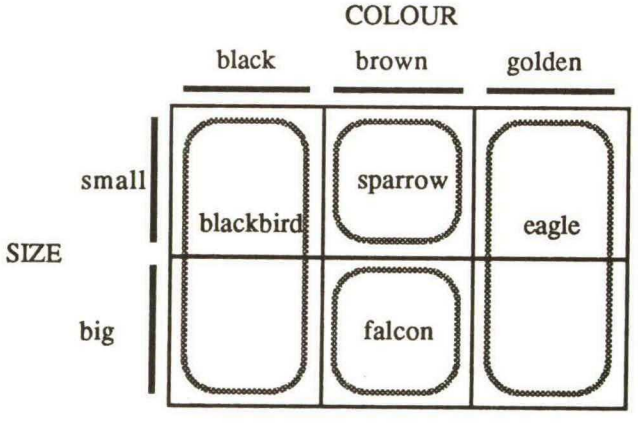


Figure 3. A universe of birds.

Now, what about incomplete examples? Suppose we are learning the concept of sparrow. As a positive example, the teacher describes Sparky the sparrow as a brown bird. This example is incomplete, because it corresponds to two blocks. The teacher clearly forgot to mention that Sparky is a **small** bird. Still, we can use the incomplete example to conclude that the rule 'all sparrows are black', which describes a necessary

[8]This representation is taken from (Banerji, 1969), and the example is borrowed from (Muggleton, 1987).
[9]The rule Consistent Example ensures that an example corresponds to at least one block.

condition for sparrowness, must be false in the intended interpretation. That is, a concept explains an incomplete example if and only if it contains *at least one* of the blocks associated with the example, which means that the concept and the union of blocks associated with the example have a non-empty intersection. This we call the *intersection* condition.

The main point now is, that this second definition of explanation also works for complete examples: a concept contains a block if and only if it has a non-empty intersection with it. This can be reformulated in our theoretical framework as follows. The intersection condition means that the concept is a weak explanation for the example, because it does not necessarily assign the same classification to each instance of the example. The inclusion condition, on the other hand, means that the concept is a strong explanation for the example. Thus we have, for complete examples, that a concept is a strong explanation if it is a weak explanation.

6.1 A logical reformulation

The points made so far will now be given a more rigorous treatment by means of a reformulation in logic. The interpretation assigned by the intersection condition to incomplete examples (i.e., interpreting missing attribute values as DON'T KNOW rather than as DON'T CARE) means that the examples should be represented by existential statements like `sparrow(sparky) & brown(sparky)` rather than by universal statements such as `sparrow(X):-brown(X)` (which is false in the intended interpretation). On the other hand, for complete examples the universal statement is always true in the intended interpretation. That is, we must extend our background theory T such that for example `sparrow(sparky) & brown(sparky) & small(sparky)` implies `sparrow(X):-brown(X),small(X)`, given T.

We show how to do this by means of an example. Suppose that the teacher already informed us that Sparky is a brown sparrow, without telling whether Sparky is small or big. Now she adds that Flap is a big brown falcon, that is, a non-sparrow. Looking at fig. 3, we should be able to conclude that Sparky must be small (assuming that the concept of sparrow is consistent). We can handle this line of reasoning deductively, if we add certain axioms to our background theory (fig. 4).

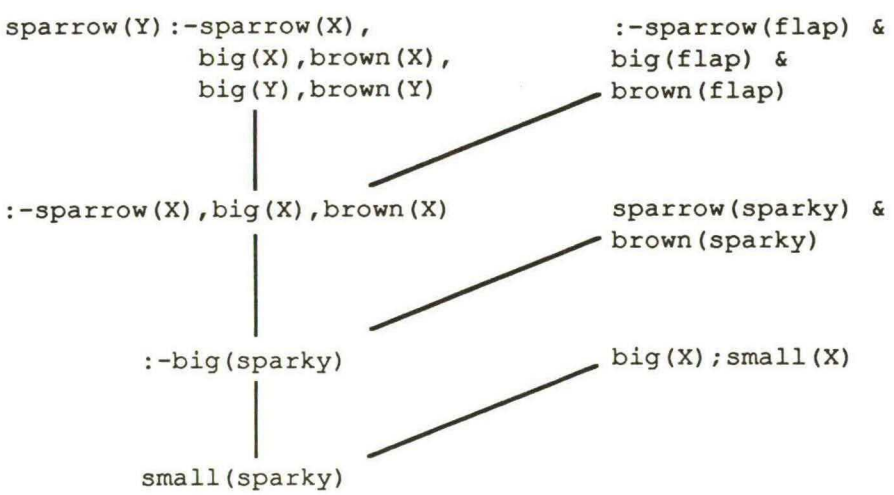


Figure 4. Sparky is small.

The crucial point in fig. 4 is the top-left axiom. It is in fact an instance of a second-order axiom `P(Y):-big(X),brown(X),P(X),big(Y),brown(Y)`, which states that *whatever is true or false about one big brown bird, must be equally true or false about every big brown bird.* This is an expression of the

fact, that big&brown is an undividable block in fig. 3. Put differently: it indicates the limitations of the concept language in a declarative way. We need one such axiom for every block in fig. 3. In addition, we need axioms stating that for every property, each bird can be assigned exactly one of its values. For instance, for the property colour we would have `black(X);brown(X);golden(X)`, `:-black(X),brown(X)`, `:-black(X),golden(X)`, and `:-brown(X),golden(X)`. Note, that all these axioms can be automatically derived, if we know which properties there are, and which values each property has. That is, a large amount of first-order axioms can be represented by a few higher-order axioms. For instance, the axiom `:-black(X),brown(X)` is a logical consequence of the following second-order theory:

$\forall$`P,V1,V2,X: property(P)`$\wedge$`P(V1)`$\wedge$`P(V2)`$\rightarrow\neg$`(V1(X)`$\wedge$`V2(X))`

```
property(colour)
colour(black)
colour(brown)
```

Since a general-purpose theorem prover for second-order logic will be too inefficient, we prefer a specialised meta-interpreter instead. In the following sections, we describe a meta-level representation of the necessary axioms, and two meta-interpreters for reasoning with them.

6.2 Representation of background axioms

The above theory is represented on the meta-level as follows:

```
axiom1(([]:-[V1(X),V2(X)])):-
        property(P,Values),
        select_two(Values,V1,V2).
select_two(List,First,Second):-
        select(List,First,List1),
        select(List1,Second,List2).
select([X|Xs],X,Xs).
select([X|Xs],Y,Zs):-
        select(Xs,Y,Zs).
property(colour,[black,brown,golden]).
```

Here, clauses (not necessarily Horn) are represented on the meta-level by terms `Head:-Body`, where `Head` and `Body` are lists of literals. We use the possibility, provided by some Prologs, of using variables in functor position[10]. As expected, `axiom1(([]:-[black(X),brown(X)]))` is a logical consequence of this program.

The following program represents axioms like `black(X);brown(X);golden(X)`:

```
axiom2((Head:-[])):-
        property(P,Values),
        construct(Values,X,Head).
construct([],X,[]).
construct([V|Vs],X,[V(X)|Rest]):-
        construct(Vs,X,Rest).
property(colour,[black,brown,golden]).
```

Second-order axioms like `P(Y):-P(X),big(X),brown(X),big(Y),brown(Y)` are represented by the following program, which allows for the instantiation of the predicate variable `P`:

[10]Alternatively, we could use `=..` to construct the terms explicitly.

```
axiom3(Axiom):-
        to_be_learned(P),
        axiom3(P,Axiom).
axiom3(P,([P(X)]:-[P(Y)|Body])):-
        properties(Props),
        prop_values(Props,Values),
        construct(Values,X,XBody),
        construct(Values,Y,YBody),
        append(XBody,YBody,Body).
prop_values([],[]).
prop_values([P|Props],[V|Values]):-
        property(P,PValues),
        element(V,PValues)
        prop_values(Props,Values).
to_be_learned(sparrow).
properties([colour,size]).
property(colour,[black,brown,golden]).
property(size,[small,big]).
```

In the remaining two sections, we describe two meta-interpreters for reasoning with background axioms which are represented in this way.

6.3 Extending incomplete examples

The following program is a meta-interpreter for carrying out proofs as in fig. 4. The predicate `extend/3` takes two examples, one positive and one negative, and yields an extension for the incomplete example, if possible. The program uses the predicate `resolve/3`, which implements a resolution step for full clausal logic.

```
extend(PosNeg,NegPos,Extension):-
        prove_with_axiom3(PosNeg,F1),
        prove_list(NegPos,F1,F2),
        prove_with_axiom2(F2,Extension).
prove_with_axiom3(Example,F):-
        axiom3(Axiom),
        prove_list(Example,Axiom,F).
prove_list([],F,F).
prove_list([H|T],A,F):-
        resolve(H,A,R),
        prove_list(T,R,F).
prove_with_axiom2(In,Out):-
        axiom2(Axiom),
        resolve(In,Axiom,Out).
```

The following three queries illustrate how the program works. The first query shows that the program is able to handle the proof of fig. 4.

```
?-extend([ ([]:-[sparrow(flap)]), ([big(flap)]:-[]),
           ([brown(flap)]:-[]) ],
         [ ([sparrow(sparky)]:-[]), ([brown(sparky)]:-[]) ],
         Extension).
Extension = [small(sparky)]:-[]
```

The second query shows, that the program works equally well for an incomplete negative example and a complete positive example (in fact, the order of the first two arguments is immaterial). Furthermore, it shows that the extension need not be a definite clause.

```
?-extend([ ([]:-[sparrow(bruce)]), ([small(bruce)]:-[]) ],
         [ ([sparrow(sparky)]:-[]), ([brown(sparky)]:-[]),
           ([small(sparky)]:-[]) ],
         Extension).
Extension = [black(bruce),golden(bruce)]:-[]
```

Finally, we show that some additional information can be deduced, even if both examples are incomplete. The query now has several answers, depending on the colours of Bruce and Sparky.

```
?-extend([ ([]:-[sparrow(bruce)]), ([small(bruce)]:-[]) ],
         [ ([sparrow(sparky)]:-[]), ([small(sparky)]:-[]) ],
         Extension).
Extension = [brown(sparky),golden(sparky)]:-[black(bruce)] ;
Extension = [brown(bruce),golden(bruce)]:-[black(sparky)] ;
Extension = [black(sparky),golden(sparky)]:-[brown(bruce)] ;
Extension = [black(bruce),golden(bruce)]:-[brown(sparky)] ;
Extension = [black(sparky),brown(sparky)]:-[golden(bruce)] ;
Extension = [black(bruce),brown(bruce)]:-[golden(sparky)]
```

6.4 Classifying unseen instances

The following program is a meta-interpreter for classifying new birds on the basis of previous examples. The first clause handles the trivial case, where the complete description of the new bird matches a previous complete example. Alternatively, if the example is incomplete, we can try to extend it.

```
classify(Examples,Desc,Class):-
     element(Ex1,Examples),
     prove_with_axiom3(Ex1,F1),
     prove_list(Desc,F1,Class).
classify(Examples,Desc,Class):-
     element(Ex1,Examples),
     prove_with_axiom3(Ex1,F1),
     prove_list(Desc,F1,F2),
     element(Ex2,Examples),
     extend(Ex1,Ex2,Extension),
     resolve(F2,Extension,Class).
```

Again, we illustrate the operation of the program with three queries. The first query asks for a classification of a small, brown bird, given that Sparky is a brown sparrow, while Flap is a big brown non-sparrow. There are two alternative answers, depending on whether the incomplete example is extended or not:

```
?-classify([ [([sparrow(sparky)]:-[]),([brown(sparky)]:-[])],
             [([]:-[sparrow(flap)]), ([brown(flap)]:-[]),
              ([big(flap)]:-[])] ],
           [ ([brown(bird)]:-[]), ([small(bird)]:-[]) ],
           Class)
Class = [sparrow(bird)]:-[small(sparky)] ;
Class = [sparrow(bird)]:-[]
```

The second query repeats the first one, except that this time the description of the new bird is also incomplete: we only know that it is small. In this case, it can only be classified if we assume that it is brown:

```
?-classify([ [([sparrow(sparky)]:-[]),([brown(sparky)]:-[])],
             [([]:-[sparrow(flap)]), ([brown(flap)]:-[]),
              ([big(flap)]:-[])] ],
           [([small(bird)]:-[])],
           Class)
Class = [sparrow(bird)]:-[small(sparky), brown(bird)] ;
Class = [sparrow(bird)]:-[brown(bird)]
```

The situation changes if we leave the size of the new bird unspecified, while stating that it is brown. In this case, we can assign two alternative classifications, depending on the size of the new bird.

```
?-classify([ [([sparrow(sparky)]:-[]),([brown(sparky)]:-[])],
             [([]:-[sparrow(flap)]), ([brown(flap)]:-[]),
              ([big(flap)]:-[])] ],
           [([brown(bird)]:-[])],
           Class)
Class = [sparrow(bird)]:-[small(sparky), small(bird)] ;
Class = [sparrow(bird)]:-[big(sparky), big(bird)] ;
Class = []:-[sparrow(bird), big(bird)] ;
Class = [sparrow(bird)]:-[small(bird)]
```

Note, that the second answer is in fact true but useless: if Sparky were big, the two examples would be logically inconsistent. On the other hand, the third answer correctly states that if the new bird is big, then it must be a non-sparrow.

Since the above program performs classification of unseen instances, it seems that it is already a learning program. However, this is not the case, since classification is only based upon previous examples, and not on inductive generalisations. Every inference step (such as inferring missing attribute values) is deductively justified by the background theory. In this respect, the program resembles the Version Space method, which maintains every possible hypothesis, such that classification is always deductively justified by the examples and the language bias. Also note that, while Version Space classification is three-valued (instance of the concept, not an instance of the concept, unknown), the above program can give conditional classifications, depending on missing attribute values.

In conclusion, we note that the framework sketched above also enables a systematic approach to the incorporation of additional, domain-dependent background knowledge into the learning process. Such background knowledge expresses dependencies between properties: if I know something about the values an object has for these properties, than I also know something about the possible values for that property. For instance, in fig. 3 we could add the property shade with values bright (all golden birds and some brown birds) and dark (all black birds and some brown birds). This means that bright black birds and dark golden birds do

not exist. Instead of 2*3*2=12 blocks, there are only 8 blocks. This is true in general: background knowledge reduces the number of atoms in the concept lattice. The higher-order logical framework alluded to above should contain mechanisms to express such background knowledge. See (Flach & Veelenturf, 1989) for an initial study of such mechanisms.

7. Finding structure in data

In the previous two sections, we gave examples of induction problems in which both weak and strong explanations occurred. In this section, we discuss cases in which hypotheses are never strong enough to be more than weak explanations. Such hypotheses are called *weak* theories. A simple example of this type of learning is provided by the problem of inducing type hierarchies from typed individuals (Flach, 1990a). For example, from `bird(sparky)`, `sparrow(sparky)`, `bird(flap)` and `:-sparrow(flap)`, we might induce the theory {`bird(X) :-sparrow(X)`}, but not {`sparrow(X) :-bird(X)`}, because the latter is inconsistent with what we know about Flap. The first theory does not imply any of the examples, but is consistent with them. Therefore, it is a weak explanation but not a strong one. Weak theories are not meant to subsume the examples; rather, they describe certain structural properties found in the examples.

More contrived induction problems of this type can be found in the field of databases (Flach, 1990b). Let R be a database relation concerning beers, bars, and drinkers. A typical tuple from R would be `r(jones,heineken,jimmys)`, meaning that Jones drinks Heineken at Jimmy's. Suppose that every bar serves only one beer; this is expressed as

```
Beer1=Beer2:-r(Drinker1,Beer1,Bar),r(Drinker2,Beer2,Bar)
```

This is known as a *functional dependency* in database theory. It shows that the two tuples `r(jones,heineken,jimmys)` and `r(smith,heineken,jimmys)` contain redundant information: if Jones drinks Heineken at Jimmy's, and Smith goes to Jimmy's, she can only drink Heineken. Thus, R can be decomposed into two separate relations, one about drinkers and the beers they drink, and one about bars and the beer they serve.

Alternatively, suppose that if a drinker drinks a beer, she drinks it in every bar in which it is served. This is expressed as

```
r(Drinker1,Beer,Bar2):-r(Drinker1,Beer,Bar1),r(Drinker2,Beer,Bar2)
```

and it is known as a *multivalued dependency*. In this case, R contains redundant information also: if Jones drinks Heineken at Jimmy's, and Smith drinks Heineken in the Pink Panther, we know that Jones also drinks Heineken in the Pink Panther, and Smith also drinks Heineken at Jimmy's. Again, R can be decomposed into two relations.

If we want to find out what functional dependencies (fds) hold for R, we could proceed as follows. Initially, FD contains all possible fds. For each fd in FD, look for two tuples which refute it; if found, remove it from FD. This is a rather naive approach, because more general fds such as

```
Beer1=Beer2:-r(Drinker1,Beer1,Bar),r(Drinker2,Beer2,Bar)
```

imply more specific ones like

```
Beer1=Beer2:-r(Drinker,Beer1,Bar),r(Drinker,Beer2,Bar)
```

Thus, if the latter is refuted by two tuples, the former is also. Instead of maintaining the full set of fds, we keep only their most general elements; if they are refuted, we look at the refuting tuples in order to see how far they must be specialised. Full details can be found in (Flach, 1990b).

The important point here is, that this process can be reformulated as induction of a weak theory. Tuples are positive examples, which are supplied one at a time. A hypothesis is a set of fds. The relation between tuples and fds is logical consistency, not implication. The set of possible hypotheses is bounded from above by one most general hypothesis (since Explanation Weakening holds instead of Explanation Strengthening). This most general explanation can be specialised by specialising its elements that are found inconsistent.

If we want to learn what multivalued dependencies (mvds) hold for R, we could proceed in a similar way. There is, however, an important difference between fds and mvds: fds are always refuted by tuples in the relation, while mvds can only be refuted by two tuples which are in the relation, and one tuple which is not. For example, the mvd

```
r(Drinker1,Beer,Bar2):-r(Drinker1,Beer,Bar1),r(Drinker2,Beer,Bar2)
```

can be refuted by demonstrating that `r(smith,heineken,jimmys)` and `r(jones,heineken,pinkpanther)` are in the relation, but `r(smith,heineken,pinkpanther)` is not. We could use the Closed World Assumption and assume that if a tuple is not known to be in the relation, then it is not in the relation. However, this is only possible if the entire relation is presented to the learner at once.

If we want to induce mvds incrementally, negative information must be explicitly available during the induction process. Since it is somewhat unrealistic to demand that every possible tuple should be marked as being in the relation or not, we suggested a *querying* approach in (Flach, 1990b): if `r(smith,heineken,pinkpanther)` has not yet been presented by the teacher, we just ask her whether it is in the relation or not. If it is not, the mvd is refuted; if it is, we add it as a positive example and try to refute the mvd in a different way (for instance by asking whether `r(jones,heineken,jimmys)` is in the relation).

Note that Compositionality does not hold when inducing weak theories. Thus, a hypothesis weakly explaining examples separately does not necessarily weakly explain their conjunction. For example, if H=`{sparrow(X):-bird(X)}`, E_1=`bird(flap)` and E_2=`:-sparrow(flap)`, we have $E_1 \not\models_T H$, $E_2 \not\models_T H$, but not $E_1 \wedge E_2 \not\models_T H$. The reason here is, that different examples are related to each other by way of the constants they contain. For the same reason, Compositionality does not hold for inducing fds and mvds. Therefore, these procedures must remember every example. After each new example, hypotheses must be tested against the entire set of examples.

8. Related work

In this paper, we view *explanation* as the central notion in induction. This relation bears some similarity with the *covers* relation introduced by De Raedt in his Generic Concept Learning algorithm *GENCOL* (de Raedt, 1991). However, since De Raedt defines generality in terms of the covers relation (concept C_1 is more general than C_2 iff the set of examples covered by C_1 contains those covered by C_2), this in fact defines *covers*(C, e) as $\models C \rightarrow e$, i.e. strong explanation. In our framework, we achieve additional freedom by defining explanation and generality seperately.

Explanation is a central notion in many forms of reasoning, such as abduction and diagnosis. Each of these types of reasoning can be characterised as a model inference problem, i.e. extending a partial theory (or preferring certain models), such that it entails some given facts[11]. Zadrozny (1990) describes the role of explanations in abduction in a way very similar to the analysis presented in this paper, by means of structural properties. Poole (1989) describes two models of diagnosis: consistency-based diagnosis, resulting in a minimal description of abnormal components which is consistent with the observations, and abductive diagnosis, resulting in a minimal diagnosis which implies the observations. Obviously, these two models can be reformulated in terms of weak and strong explanation. Also, Poole notes that the two models use different descriptions of normal and/or abnormal behaviour, whereas in our model we showed that weak and strong explanations differ with respect to the kind of condition (necessary/sufficient). It seems that a further integration of models of induction, abduction and diagnosis is possible and desirable.

Induction is also related to theory revision: how to change a given theory in order to take new information into account. Gärdenfors (1988) distinguishes three revision operators: *expansion* incorporates non-conflicting information in the theory, *revision* takes care of new information which contradicts the current theory, and *contraction* changes the theory such that it no longer implies some facts. These operators are abstractly defined by their structural properties. There is a strong relationship between revision operators and explanation: for instance, the expansion of a theory T with new information A can be defined as the set $\{B \mid B \mid\!\sim_T A\}$, i.e. the set of formulas explained by A, given T. It can then be shown that structural properties of strong explanation translate to Gärdenfors' *rationality postulates* for expansion operators. Another interesting parallel concerns his notion of *epistemic entrenchment*, which is an ordering on formulas used to define revision and contraction operators: this ordering seems to be closely related to the generality ordering used in inductive learning. Whereas our framework describes the *static* properties of explanations, Gärdenfors' framework can be used to describe the *dynamics* of explanations when taking new examples into account. On the other hand, revision operators are based on classical logic; it would be interesting to redefine them in terms of explanations, thereby gaining the flexibility of varying the underlying logic.

[11] Note the correspondence with non-monotonic reasoning.

9. Concluding remarks

In this paper, we have outlined a logical framework for analysing notions of explanation. We believe such a framework to be useful, because explanation is a central concept in many AI problems, such as diagnosis and learning. The framework allows the study of the crucial properties of explanations in an abstract setting, without too much reference to the underlying languages. More specifically, the framework offers an analysis of

- the properties that make explanation useful for inductive learning (such as Convergence);
- the relation between these properties and the logical properties of the base logic (like monotonicity);
- abstract definitions of explanation (the logical systems **SC**, **SM**, **W**, and **CC**);
- alternative definitions of explanation, and their interrelations.

Furthermore, we have applied this framework to various topics in inductive learning. We have shown that the duality between weak and strong explanation has a parallel in the duality between learning necessary and sufficient conditions, which can be expressed by means of predicate completion. We claim that a better understanding of this parallel will increase our knowledge of operators for generalisation and specialisation. Other applications include handling missing attribute values in logic, and discovering structure in data.

We have outlined some of the parallels between our framework and existing models of abduction and theory revision, and we are presently pursuing a further study of these parallels. Furthermore, since we presented explanation as an inference relation, it would be nice to have a model-theoretic characterisation of it, including a representation theorem (soundness and completeness). This would allow us to prove statements like 'in the system **SM**, we have $\alpha \mid\!\sim_T \beta$ iff $T \models \beta \rightarrow \alpha$'. Finally, we are working on an implementation of some of the ideas illustrated in this paper, in particular the use of predicate completion, and induction of non-monotonic theories.

References

M. BAIN & S. MUGGLETON (1991), 'Non-monotonic learning'. In *Machine Intelligence 12*, J.E. Hayes, D. Michie & E. Tyugu (eds.), pp. 105-119, Oxford University Press, Oxford.

R.B. BANERJI (1969), *Theory of problem solving: an approach to Artificial Intelligence*, Elsevier, New York.

K.L. CLARK (1978), 'Negation as failure'. In *Logic and Databases*, H. Gallaire & J. Minker (eds.), pp. 293-322, Plenum Press, New York.

P.A. FLACH & L.P.J. VEELENTURF (1989), 'Concept learning from examples: theoretical foundations', ITK Research Report no. 2, Institute for Language Technology & Artificial Intelligence, Tilburg University, the Netherlands.

P.A. FLACH (1990a), 'Second-order inductive learning', ITK Research Report no. 10, Institute for Language Technology & Artificial Intelligence, Tilburg University, the Netherlands, January. A preliminary version of this paper appeared in Analogical and Inductive Inference AII'89, K.P. Jantke (ed.), Lecture Notes in Computer Science 397, Springer Verlag, Berlin, 1989, pp. 202-216.

P.A. FLACH (1990b), 'Inductive characterisation of database relations'. In *Proc. International Symposium on Methodologies for Intelligent Systems*, Z.W. Ras, M. Zemankowa & M.L. Emrich (eds.), pp. 371-378, North-Holland, Amsterdam. Full version appeared as ITK Research Report no. 23.

D.M. GABBAY (1985), 'Theoretical foundations for non-monotonic reasoning in expert systems'. In *Logics and Models of Concurrent Systems*, K.R. Apt (ed.), pp. 439-457, Springer Verlag, Berlin.

P. GÄRDENFORS (1988), *Knowledge in flux*, MIT Press, Cambridge, Massachusetts.

S. KRAUS, D. LEHMANN & M. MAGIDOR (1990), 'Nonmonotonic reasoning, preferential models and cumulative logics', *Artificial Intelligence* **44**, pp. 167-207.

C. LING (1991), 'Non-monotonic specialisation (preliminary version)'. In *Proc. First International Workshop on Inductive Logic Programming*, S. Muggleton (ed.), pp. 59-68, Viana de Castelo, Portugal.

T.M. MITCHELL (1982), 'Generalization as search', *Artificial Intelligence* **18**:2, pp. 203-226.

S. MUGGLETON (1987), 'Duce, an oracle based approach to constructive induction'. In *Proc. Tenth International Joint Conference on Artificial Intelligence*, pp. 287-292, Morgan Kaufmann, Los Altos, CA.

SUMMARY OF ITK RESEARCH REPORTS

No	Author	Title
1	H.C. Bunt	On-line Interpretation in Speech Understanding and Dialogue Sytems
2	P.A. Flach	Concept Learning from Examples Theoretical Foundations
3	O. De Troyer	RIDL*: A Tool for the Computer-Assisted Engineering of Large Databases in the Presence of Integrity Constraints
4	E. Thijsse	Something you might want to know about "wanting to know"
5	H.C. Bunt	A Model-theoretic Approach to Multi-Database Knowledge Representation
6	E.J. v.d. Linden	Lambek theorem proving and feature unification
7	H.C. Bunt	DPSG and its use in sentence generation from meaning representations
8	R. Berndsen en H. Daniels	Qualitative Economics in Prolog
9	P.A. Flach	A simple concept learner and its implementation
10	P.A. Flach	Second-order inductive learning
11	E. Thijsse	Partical logic and modal logic: a systematic survey
12	F. Dols	The Representation of Definite Description
13	R.J. Beun	The recognition of Declarative Questions in Information Dialogues
14	H.C. Bunt	Language Understanding by Computer: Developments on the Theoretical Side
15	H.C. Bunt	DIT Dynamic Interpretation in Text and dialogue
16	R. Ahn en H.P. Kolb	Discourse Representation meets Constructive Mathematics

No	Author	Title
17	G. Minnen en E.J. v.d. Linden	Algorithmen for generation in lambek theorem proving
18	H.C. Bunt	DPSG and its use in parsing
19	H.P. Kolb	Levels and Empty? Categories in a Principles and Parameters Approach to Parsing
20	H.C. Bunt	Modular Incremental Modelling Belief and Intention
21	F. Dols en H. Daniels	Nog niet verschenen
22	F. Dols	Nog niet verschenen
23	P.A. Flach	Inductive characterisation of database relations
24	E. Thijsse H. Daniels	Definability in partial logic: the propositional part
25	H. Weigand	Modelling Documents
26	O. De Troyer	Object Oriented methods in data engineering
27	O. De Troyer	The O-O Binary Relationship Model
28	E. Thijsse	On total awareness logics
29	E. Aarts	Recognition for Acyclic Context Sensitive Grammars is NP-complete
30	P.A. Flach	The role of explanations in inductive learning
31	W. Daelemans, K. De Smedt en J. de Graaf	Default inheritance in an object-oriented representation of linguistic categories

S. MUGGLETON & W. BUNTINE (1988), 'Machine invention of first-order predicates by inverting resolution'. In *Proc. Fifth International Conference on Machine Learning*, J. Laird (ed.), pp. 339-352, Morgan Kaufmann, San Mateo.

S. MUGGLETON (1990), 'Inductive Logic Programming'. In *Proc. First Conference on Algorithmic Learning Theory*, Ohmsha, Tokyo.

S. MUGGLETON, ED. (1991), *Proc. First International Workshop on Inductive Logic Programming*, Viana de Castelo, Portugal.

D. POOLE (1989), 'Normality and faults in logic-based diagnosis'. In *Proc. Eleventh International Joint Conference on Artificial Intelligence*, pp. 1304-1310, Morgan Kaufmann, Los Altos, CA.

L. DE RAEDT (1991), *Interactive concept-learning*, PhD thesis, Catholic University Leuven.

E.Y. SHAPIRO (1981), *Inductive inference of theories from facts*, Techn. rep. 192, Comp. Sc. Dep., Yale University.

W. ZADROZNY (1990), 'The logic of abduction (preliminary report)'. In *Proc. First International Workshop on Principles of Diagnosis*, pp. 8-17, Stanford University.

ITK: P.O. BOX 90153 5000 LE TILBURG THE NETHERLANDS