Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

Research Report
RR-92-08

Approaches to Abductive Reasoning

- An Overview -

Gabriele Merziger

February 1992

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Postfach 20 80
D-6750 Kaiserslautern, FRG
Tel.: (+49 631) 205-3211/13
Fax: (+49 631) 205-3210

Stuhlsatzenhausweg 3
D-6600 Saarbrücken 11, FRG
Tel.: (+49 681) 302-5252
Fax: (+49 681) 302-5341

Deutsches Forschungszentrum für Künstliche Intelligenz

The German Research Center for Artificial Intelligence (Deutsches Forschungszentrum für Künstliche Intelligenz, DFKI) with sites in Kaiserslautern und Saarbrücken is a non-profit organization which was founded in 1988. The shareholder companies are Daimler Benz, Fraunhofer Gesellschaft, GMD, IBM, Insiders, Krupp-Atlas, Mannesmann-Kienzle, Philips, Sema Group Systems, Siemens and Siemens-Nixdorf. Research projects conducted at the DFKI are funded by the German Ministry for Research and Technology, by the shareholder companies, or by other industrial contracts.

The DFKI conducts application-oriented basic research in the field of artificial intelligence and other related subfields of computer science. The overall goal is to construct *systems with technical knowledge and common sense* which - by using AI methods - implement a problem solution for a selected application area. Currently, there are the following research areas at the DFKI:

- Intelligent Engineering Systems
- Intelligent User Interfaces
- Intelligent Communication Networks
- Intelligent Cooperative Systems.

The DFKI strives at making its research results available to the scientific community. There exist many contacts to domestic and foreign research institutions, both in academy and industry. The DFKI hosts technology transfer workshops for shareholders and other interested groups in order to inform about the current state of research.

From its beginning, the DFKI has provided an attractive working environment for AI researchers from Germany and from all over the world. The goal is to have a staff of about 100 researchers at the end of the building-up phase.

Prof. Dr. Gerhard Barth
Director

Approaches to Abductive Reasoning
- An Overview -

Gabriele Merziger

DFKI-RR-92-08

This work has been supported by a grant from The Federal Ministry for Research and Technology (FKZ ITW-9000 8).

Approaches to Abductive Reasoning
- An Overview -

Gabriele Merziger

German Research Center for Artificial Intelligence
Stuhlsatzenhausweg 3
W - 6600 Saarbrücken 11
e-mail: merziger@dfki.uni-sb.de

Abstract

Abduction is a form of non-monotonic reasoning that has gained increasing interest in the last few years. The key idea behind it can be represented by the following inference rule

$$\frac{\varphi \rightarrow \omega,\ \omega}{\varphi},$$

i.e., from an occurrence of ω and the rule "φ implies ω", infer an occurrence of φ as a *plausible* hypothesis or explanation for ω. Thus, in contrast to deduction, abduction is as well as induction a form of "defeasible" inference, i.e., the formulae sanctioned are plausible and submitted to verification.

In this paper, a formal description of current approaches is given. The underlying reasoning process is treated independently and divided into two parts. This includes a description of *methods for hypotheses generation* and *methods for finding the best explanations* among a set of possible ones. Furthermore, the complexity of the abductive task is surveyed in connection with its relationship to default reasoning. We conclude with the presentation of applications of the discussed approaches focusing on plan recognition and plan generation.

Contents

1 Introduction

Abductive reasoning has gained increasing interest in many fields of AI research. Its utility was first observed for diagnostic tasks (cf. [Pop73]), but as many researchers have shown it is not limited to this use. Currently under investigation or suggested are such different applications as plan recognition (e.g., [HK90]), text understanding and generation (e.g., [Sti90]), program debugging (cf. [CM85]), planning (e.g., [Esh88]), user modelling (cf. [Poo88]) or vision (cf. [CM85]).

The aim of this paper is to summarize the underlying formal models of abduction and investigate different methods for arriving at abductive hypotheses. Only in the last section will we regard possible applications, restricted to the field of plan recognition and plan generation.

Abduction is used to generate explanations, i.e., abducible sentences, whereas deduction is normally used for testing derivability. In a deduction step, each consequence is obtained by applying a logically correct inference rule, so the deduced formulae are logical consequences of the theory under consideration. Reasoning by abduction can be roughly described by the following inference rule:

$$\frac{\varphi \rightarrow \omega,\ \omega}{\varphi}$$

This rule corresponds to some kind of inversion of Modus Ponens. Nevertheless, concerning the implication there is an important difference. Modus Ponens requires material implication as it is used in classical logic, but abduction allows one to characterize the relationship between φ and ω with a much wider degree of freedom. Usually, it is thought of as some kind of causal relationship, as e.g., φ *is the reason for* ω *being true*, but as Levesque (cf. [Lev89]) shows, not all abduction has to be concerned with cause and effect in the strict sense. He suggests the extension of the notion of explanation in order to grasp the case that φ is sufficient, not only necessary, to sanction the belief in ω, i.e., there does not have to be a direct causal relationship between both formulae but, in connection with what is known, φ is enough for ω to be true.

Most of the current approaches do not consider these representation problems. Normally, the rules used for abduction are written down using material implication that is implicitly, i.e., by its use, interpreted as some kind of causal relationship.

In this paper, we will first examine the origin of the notion of abduction in philosophical logic, where it is considered in the context of the two other important inference rules, namely deduction and induction. To give a more extensive description of abductive reasoning we will divide the reasoning process into three parts that are examined independently. First, we will give a *formal description* of the abductive task that determines the underlying model. We discuss the set-cover-based approach, the logic-based approach and the knowledge-level approach. Methods for *hypothesis generation* are presented in section 3. In particular, we treat generation by resolution and generation by an Assumption-based Truth Maintenance System. As these methods produce *sets* of feasible hypotheses, it is convenient to further restrict these sets by applying heuristics. Approaches to *finding the "best" hypotheses* are investigated in section 4. In section 5, we will briefly look at the relationship between default reasoning and abduction. By determining a common

subtask some remarks on the complexity of both abductive and default reasoning can be made. We conclude by presenting in section 6 some applications of abductive reasoning in the field of planning and plan recognition, respectively: the approaches of Eshghi (see [Esh88]), and Helft and Konolige (see [HK90]).

1.1 Philosophical Background

The notion of abduction was first introduced by the philosopher Peirce (see [Pei58]). In the field of AI it was taken up in 1973 by Pople (cf. [Pop73]) and the research in this area was revived by Charniak and McDermott. In [CM85] they stress the importance of abduction as a third form of inference besides induction and deduction.

In philosophical logic, abduction has always been seen in close connection with induction. In his earlier papers (see [Pei58]), Peirce sees abduction as a reasoning process from effect to cause, thus yielding explanations. In contrast to this, induction conjectures general laws from particulars, i.e., instead of synthesizing explanations, induction classifies and thus does not add new knowledge to the theory. Later, Peirce emphasizes the connection between both inference methods in the following way: "Induction is an argument which sets out from a hypothesis, resulting from previous Abduction ... " (see p. 198 in [Gou50]). This means that abduction should be used to synthesize an explanation, the induction axiom, that thereafter is verified by induction. Only the use of both processes together yields an acceptable explanation.

An important feature of abduction and as well of induction is their reducibility to a valid deduction, if the abductive process is sound. If φ explains ω abductively in connection with theory $\mathcal{T}$, then ω must be derivable from $\varphi \cup \mathcal{T}$. This connection gave rise to one method of constructing abductive hypotheses, namely by resolving the negated observation ω with theory $\mathcal{T}$. The dead-ends reached are candidate hypotheses (see section 3). Finding an explanation in such a way by deductive means thus ensures the soundness of the process.

In general, there are several possible abductive hypotheses and the problem arises of how to choose among them. Peirce gives several criteria that a good explanation should fulfill (see [Gou50]). First and foremost, a hypothesis should account for the facts. But it should also follow Occam's Razor, i.e., it should be the "simplest" hypothesis available. In general, "simplicity" is interpreted as logical simplicity, which means that those hypotheses are preferred that contain fewer different predicates. These are the hypotheses that require the fewest additional assumptions to what has been observed. Peirce stresses the importance of logical simplicity, but what he estimates higher is "psychological simplicity" in the sense that an adequate hypothesis should be the most natural, i.e., one that would be intuitively preferred.

Current research favours logical simplicity as selection criterion and one reason for this is certainly the fact that psychological simplicity is hard to define. However, Mooney and Ng (cf. [NM90]) recently have suggested a new measure for the quality of explanations which they call *explanatory coherence* (see also section 4.2). This metric controls the choice of hypotheses so that those are selected that best "tie together" the various observations that are to be explained, i.e., that yield the most coherent explanation. As the authors claim, these are usually also the intuitively preferred ones. Especially in the field of text

understanding and plan recognition or planning, where reasoning processes of human agents have to be simulated, such a measure that prefers the most natural hypothesis seems inevitable.

2 Formal Models

It is possible to determine different models that are used to specify the abductive process:

- set-cover-based approaches (e.g., [ATBJ87]),
- logic-based approaches (e.g., [EK88b]), and
- the knowledge-level approach (cf. [Lev89]).

In the following, we will describe each of them.

2.1 A Set-cover-based Approach

In set-cover-based approaches a set of explanations is found by selecting a suitable subset from a given set of hypotheses. This subset should best account for the observations and is determined by coverings, parsimony, plausibility or another suitable selection criterion. Since hypotheses are constructed using a set of previously known candidates, this approach is also called *hypothesis assembly*.

To be able to treat this form of abductive reasoning more formally, we will first introduce the terminology used in [ATBJ87].

Definition 1: [ATBJ87] (hypothesis assembly)
A *domain for hypothesis assembly* is defined by the triple (Φ, Ω, e), where Φ is a finite set of hypotheses, Ω is a set of observations and e is a mapping from subsets of Φ to subsets of Ω. $e(\Phi)$ is called the *explanatory power* of the set of hypotheses Φ and determines the set of observations Φ accounts for. An assembly problem is given by a set $\Omega' \subseteq \Omega$ of observations that have to be explained.

Example: To illustrate the ideas presented in this and the following sections, we will always use slight variations of the scenario outlined below.
Let $\mathcal{T}$ be a theory consisting of the following propositions:

$$\forall x\ (bird(x) \wedge \neg ab(x) \supset flies(x)) \quad (1)$$
$$\forall x\ (ufo(x) \supset flies(x)) \quad (2)$$
$$\forall x\ (penguin(x) \vee ostrich(x) \supset ab(x)) \quad (3)$$
$$\forall x\ (songbird(x) \supset bird(x)) \quad (4)$$
$$\forall x\ (songbird(x) \supset eats\_insects(x)) \quad (5)$$
$$\forall x\ (frog(x) \supset eats\_insects(x)) \quad (6)$$
$$\forall x\ (frog(x) \supset green(x) \wedge croaks(x)) \quad (7)$$
$$\forall x\ (frog(x) \supset ab(x)) \quad (8)$$

In addition, we assume axioms stating the disjointness of the extensions of predicates denoting different animal types, i.e., frogs are not birds, etc.

Taking this example scenario, but with explicit knowledge about things that do not fly, a domain for hypothesis assembly can be defined as follows:

$$\begin{aligned}
\Phi &= \{frog(x), songbird(x), bird(x), ufo(x), no_bird(x)\} \\
\Omega &= \{flies(x), green(x), croaks(x), \neg flies(x), eats_insects(x)\}, \\
e(\{frog(x)\}) &= \{eats_insects(x), \neg flies(x), green(x), croaks(x)\}, \\
e(\{songbird(x)\}) &= \{eats_insects(x), flies(x)\}, \\
e(\{ufo(x), bird(x)\}) &= \{flies(x)\}, \\
e(\{penguin(x)\}) &= \{\neg flies(x)\}, \\
e(\{ostrich(x)\}) &= \{\neg flies(x)\}.
\end{aligned}$$

However, in order to ensure the practical use of this formal model some additional assumptions must be made which a feasible domain should fulfil. The *computability assumption* is essential for all set-cover-based models.

Computability Assumption

For any subset Φ' of Φ, $e(\Phi')$ can be computed.

This means that it is always exactly known which observations are explained by which set of hypotheses. Thus, following our example, e.g., $e(\{bird(x)\}$ must also be computable. How this can be done is determined by the *independence assumption.*

This states that the union of two hypotheses sets Φ_1 and Φ_2 accounts for the observations explained by Φ_1 as well as for those explained by Φ_2. Thus, it is suffient to give for each single hypothesis the observations accounted for.

Independence Assumption

Let $\Phi_1, \Phi_2 \subseteq \Phi$. Then $e(\Phi_1 \cup \Phi_2) = e(\Phi_1) \cup e(\Phi_2)$.

The independence assumption is necessary for the stepwise computation of the mapping e. If $e(\Phi_1)$ is known, then $e(\Phi_2) = e(\Phi_1 \cup \{\varphi\})$ can easily be generated by conjoining $e(\{\varphi\})$ to the already known set $e(\Phi_1)$. Thus,

$$e(\{bird(x), songbird(x)\}) = \{flies(x), eats\_insects(x)\}.$$

Allemang et al. point out that this assumption is satisfied by most diagnostic domains that allow decomposition in independent parts. The independence assumption is also used in the set-covering model defined by Reggia (cf. [Reg88]). The algorithm of Allemang et al. allows one to weaken it by the following two assumptions:

Monotonicity Assumption

Let $\Phi_1, \Phi_2 \subseteq \Phi$ and $\Phi_1 \subseteq \Phi_2$. Then $e(\Phi_1) \subseteq e(\Phi_2)$.

The observations accounted for by a subset of Φ_2 are a subset of those accounted for by Φ_2. This assumption is weaker than the previous one. We only have $e(\Phi_1) \subseteq e(\Phi_1 \cup \Phi_2)$ and

$e(\Phi_2) \subseteq e(\Phi_1 \cup \Phi_2)$, i.e., the union of both hypotheses sets accounts for all data explained by Φ_1 and Φ_2, but possibly also for more. With the following assumption, those data can be determined for which a hypothesis is essential.

Accountability Assumption

The function $\alpha : \Phi \rightarrow \Omega^*$ *with*

$$\alpha(\varphi) = \{\ \omega \in \Omega |\ \exists \Phi_1 \subseteq \Phi \text{ with } \varphi \in \Phi_1 \text{ such that } \omega \in e(\Phi_1) \text{ and } \omega \notin e(\Phi_1 \backslash \{\varphi\})\ \}$$

is computable.

By $\alpha(\varphi)$ we get the observations that cannot be explained without φ. If the monotonicity assumption holds, we have $\alpha(\varphi) \subseteq e(\varphi)$. Our example domain yields $\alpha(\varphi) = e(\varphi)$, as it obeys the stronger independence assumption that states that each hypothesis φ accounts precisely for $e(\varphi)$.

Set-cover-based approaches rely heavily on the previously known mapping e that already determines a superset of the explanations searched for. As a consequence, there exists no actual hypothesis generation mechanism, but rather a collection procedure that selects the relevant hypotheses out of this superset. Therefore, we will now describe the complete abductive algorithm.

Let (Φ, Ω, e) be an assembly domain and $\Omega' \subseteq \Omega$ an assembly problem. The abductive algorithm of Allemang et al. consists of four parts:

- Screening phase,
- Collection phase,
- Parsimony phase, and
- Critique phase.

Screening phase:

In the first phase, the plausibility of the hypotheses in Φ is determined,e.g., by some kind of heuristics or by predefined probability values that can be evaluated with a hierarchical classification system. In the following phases, only those hypotheses with a sufficiently high plausibility are taken into account. Considering our example we can, e.g., classify owls as birds and assign a higher priority to the more general explanation "bird".

Collection phase:

In the collection phase, the hypotheses accounting for each $\omega \in \Omega'$ are collected iteratively. Note that Φ is now the result of the screening phase. In detail, we have:

$Hyp \leftarrow \emptyset$

until $\Omega' = \emptyset$

do let ω be the "most salient" element of Ω';

find the most plausible hypothesis $\varphi \in \Phi$ with $\omega \in \alpha(\varphi)$;

$Hyp \leftarrow Hyp \cup \{\varphi\}$;
$\Omega' \leftarrow \Omega' \setminus e(Hyp)$;
od

Remarks:

- The meaning of "salient" still has to be interpreted for a given application domain.
- A hypothesis can account for several observations. All of these are then taken out of the set Ω' of observations that still have to be explained.
- A conflict arises if incompatible hypotheses are collected in Hyp. In an actual implementation of Allemang et al. this is resolved by maintaining only the last hypothesis found. As a consequence, the data accounted for by the rejected hypothesis must be explained again and strong heuristics may have to be used in order to avoid infinite loops.

Example: Assume the previously defined example domain of hypothesis assembly is given.

Let $\Omega' = \{\neg flies(F), croax(F)\}$. First, consider the observation $\neg flies(F)$. We have to find the most plausible $\varphi \in \{penguin(F), frog(F), ostrich(F)\}$. Assume that $\varphi = penguin(F)$ is chosen, i.e., $Hyp^1 = \{penguin(F)\}$. Thus,

$$\Omega' = \Omega' \setminus e(\{penguin(F)\} = \{croaks(F)\}$$

In the second loop iteration we find $\varphi = frog(F)$ with $croaks(F) \in \alpha(frog(F))$. Hence,

$$Hyp^2 = \{penguin(F), frog(F)\}.$$

Those hypotheses are incompatible (as we assumed disjointness of incompatible predicates denoting animals), so $penguin(F)$ is rejected and $\neg flies(F)$ has to be explained again. If $no\_bird(F)$ is taken as hypothesis, we get

$$Hyp = \{no\_bird(F), frog(F)\}$$

as a solution.

Parsimony phase:

The parsimony phase guarantees that there is no proper subset of Hyp that also explains Ω'. This case can occur because the elements of Ω' are considered successively. A hypothesis φ_1, added to Hyp as an explanation of the observation ω_1, can later become superfluous, because hypothesis φ_2 explains ω_2 and ω_1 as well. We get the following procedure:

for each $\varphi \in Hyp$
do if $\Omega' \subseteq e(Hyp \backslash \{\varphi\})$
then $Hyp \leftarrow Hyp \setminus \{\varphi\}$;
fi
od

Example (continuation): We had the solution $Hyp = \{no\_bird(F), frog(F)\}$. As

$$\begin{aligned} \{\neg flies(F), croaks(F)\} &\subseteq e(\{frog(F)\}) \\ &= \{eats\_insects(F), croaks(F), \neg flies(F)\} \end{aligned}$$

we can find a subset of Hyp that also accounts for the observations, i.e.,

$$Hyp \leftarrow \{frog(F)\}.$$

Remark: This procedure does not ensure that the smallest solution with respect to set cardinality is found. An algorithm solving this problem in the context of a set-cover-based model was designed by Reggia (cf. [Reg88]).

Critique phase:

Essential hypotheses, i.e., hypotheses without which no explanation can be found at all, are marked as such. This supplies the user with additional information about important parts of *every possible* explanation. The collection phase is repeated $|Hyp|$ times with each hypothesis in Hyp marked as unuseable in turn.

for each $\varphi \in Hyp$
do $G \leftarrow Hyp \backslash \{\varphi\}$;
repeat collection phase with $Hyp \leftarrow G$
and $H \leftarrow H \setminus \{\varphi\}$
if no explanation can be found
then mark φ as essential
od

Remarks:

- The collection phase is executed starting with the solution set diminished by exactly one hypothesis φ in each loop iteration. This hypothesis is marked as unuseable. If no explanation can be found, then φ is essential.
- Obviously, $frog(F)$ is essential in our example.

This set-cover-based algorithm has been implemented in the system RED (cf. [ATBJ87]) for a blood bank antibody analysis. It yields one possible plausible explanation that is subset-minimal.

Nevertheless, there are some drawbacks that cannot be disregarded. First and foremost, the computability of the mapping e is crucial for the choice of possible explanations. All causal relationships that might be relevant must be encoded in form of relations before starting the abductive procedure. This seems practicable only in restricted areas, e.g., repair problems. Apart from that, the domain must satisfy further assumptions. The independence assumption is quite strong and even if it is relaxed by the monotonicity and accountability assumptions the application seems to be restricted to domains that are easy to manage, i.e., no domains where databases of commonsense knowledge are involved. As a last point, one should mention that, as Levesque notes (see [Lev89]), small changes in the underlying theory most probably lead to difficulties, as the corresponding changes in the explanations are hard to express, i.e., the addition of new facts can enforce a quite extensive respecification of the funtion e.

So finally we could state that the set-cover-based model appears to be adequate only for diagnostic tasks or repair problems where all causal relationships are well known and can easily be represented by a function. In addition, the underlying theory should not undergo any changes.

2.2 Logic-based Approaches

The majority of research in abduction that is also the most widely accepted is based on a logical model. It allows that knowledge represented in some logical language for the purpose of deductive inferences can also be used for abduction.

Before going into the detail of several logic-based approaches, we will introduce the general idea. An abductive system consists of

- a logical theory $\mathcal{T}$ defined over the language $\mathcal{L}$, and
- a set of sentences A of $\mathcal{L}$ that are called *abducible.*

If a sentence φ is found as the result of an abductive process in searching for an explanation of ω, it must satisfy the following conditions:

- $\mathcal{T} \cup \varphi \vdash \omega$,
- $\mathcal{T} \cup \varphi$ is consistent,
- φ is abducible, i.e., $\varphi \in A$.

Remark: Sometimes predicates instead of sentences are declared as abducible. Then, sentences φ are called abducible if they contain only abducible predicates.

Thus, the observation ω must be derivable from the logical theory $\mathcal{T}$ augmented with the explanation φ under the additional condition that φ is consistent with $\mathcal{T}$. This ensures what Peirce called soundness of the abductive procedure. Furthermore, the explanation

must be an element of the set of abducible predicates or sentences. This last condition is no restriction, as A can be chosen to cover all predicate symbols of $\mathcal{L}$.

Thus, abduction is defined over global logical properties such as consistency and derivability and, as Levesque points out in [Lev89], this seems to be a drawback of this approach. In addition, the knowledge about causal relationships that is used for finding abductive explanations is represented implicitly in theory $\mathcal{T}$. The implication in $\mathcal{T}$ has to be interpreted in two different ways corresponding to actual use, i.e., in sentences containing abducible predicates implication is seen as a kind of cause-effect relationship, whereas otherwise the material implication is meant. Abduction and deduction within the same logical theory require different notions of implication (cf. [Lev89]).

In the following we will sketch two different abductive frameworks based on the logical model.

2.2.1 Simple Causal Theories

Konolige analyses in [Kon90] the relationship between abduction and Reiter's consistency-based approach to explanation (cf. [Rei87]). The formal model for abduction that he defines corresponds in general to the standard model already outlined.

Definition 2: [Kon90] (explanation)
Let $(C, E, \mathcal{T})$ be a simple causal theory defined over the first-order language $\mathcal{L}$, i.e., C is a set of causes, E a set of effects and $\mathcal{T}$ is a logical theory defined over $\mathcal{L}$. An *explanation* of a set of observations $\Omega \subseteq E$ is a finite set of sentences Φ such that

- Φ is consistent with $\mathcal{T}$,
- $\mathcal{T} \cup \Phi \vdash \Omega$, where Ω denotes the conjunction of all $\omega \in \Omega$,
- Φ is subset-minimal.

example: Let $(C, E, \mathcal{T})$ be defined as follows: Let $\mathcal{T}$ be given by our example specification in section 2.

$$C = \{frog(x), songbird(x), bird(x), ostrich(x)\}$$
$$E = \{flies(x), green(x), croaks(x), eats\_insects(x)\}$$

If we have the set of observations $\Omega = \{\neg flies(F), croaks(F)\}$ then $\Phi = \{frog(F)\}$ is an explanation, because

- $frog(F)$ is consistent with $\mathcal{T}$,
- $\mathcal{T} \cup frog(F) \vdash \neg flies(F) \wedge croaks(F)$, and
- $frog(F)$ is subset-minimal.

The last point of the definition for explanation embodies a selection criterion for "good" explanations that Allemang called parsimony (see section 2.1). It prevents the choice of a set of sentences as explanations that contains a proper subset which itself constitutes a valid explanation.

Another similarity to the set-cover-based approach can also be noted: the abductive process relates a set of observations that have to be elements in a predefined set E to a set of causes restricted by the set C. However, there is not an equivalent to the mapping e which determines for every feasible hypothesis the data it accounts for. The corresponding relationships are expressed in the logical theory by material implication going from cause to effects.

The results of Konolige show that this form of abductive reasoning can also be carried out with the consistency-based method of Reiter. In fact, a transformation is described by which abductive explanations can be generated or deduced respectively. For details see [Kon90].

2.2.2 Abduction instead of Negation as Failure

Eshgi and Kowalski (cf. [EK88b]) consider abduction in the context of logic programming with integrity constraints. They show that "negation by failure can be simulated by making negative conditions abducible and by imposing appropriate denials and disjunctions as integrity constraints." With this means they get a semantics for negation as failure that generalizes the stable model semantics. An application of this method in the field of planning is discussed in section 6.1.

In this approach, logic programs working with negation as failure (NAF) are transformed into an abductive framework, where more general integrity constraints than denials can be defined. For this purpose, abduction is introduced as follows:

Definition 3: [EK88b] (abductive framework)
$(\mathcal{T}, I, A)$ is an *abductive framework* iff

- $\mathcal{T}$ is a Horn clause theory without denials,
- I is a set of integrity constraints, and
- A is a set of predicate symbols defined as abducible.

Definition 4: [EK88b] (abductive solution)
Let $(\mathcal{T}, I, A)$ be an abductive framework. A hypotheses set Φ is an *abductive solution* for the query q iff

- Φ consists of a set of variable free abducible atoms,
- $\mathcal{T} \cup \Phi \vdash q$,
- $\mathcal{T} \cup \Phi \cup I$ is satisfiable.

Remarks:

- Integrity constraints are used as selection criterion for explanations. Hypotheses not satisfying them are ruled out by the last condition.
- Φ is restricted to being variable free in order to avoid Skolemization. This limitation is not necessary and if more general formulae are to be allowed, the authors refer to the combination of Skolemization and reverse Skolemization as outlined in [CP86].

Programs working with negation as failure are now converted into ones using abduction by the following algorithm:

Conversion from NAF-formulation into an abductive framework

(1) Negative conditions g are replaced by a new symbol g^*.

(2) Add the integrity constraint "$\leftarrow g^*(x) \wedge g(x)$".

(3) Declare g^* abducible.

Remarks:

- By (1) all negative conditions are eliminated by introducing an unambiguous new symbol for them.
- (2) and (3) ensure that if g is not valid then g^*, the new symbol introduced for $\neg g$, is abducible.
- The search space of the conversion is almost equal to the original one, but instead of testing the provability of negated conditions by negation as failure, the consistency of abducible predicates is checked.

Example: Let the Horn clause theory $\mathcal{T}$ be given by

$$\begin{aligned}\mathcal{T} = \{ flies(x) &\leftarrow bird(x) \wedge \neg ab(x),\\ bird(x) &\leftarrow songbird(x),\\ eats\_insects(x) &\leftarrow songbird(x),\\ frog(x) &\leftarrow green(x) \wedge croaks(x),\\ ab(x) &\leftarrow frog(x)\}\end{aligned}$$

A conversion yields the framework $(\mathcal{T}', I, A)$ with

$$\begin{aligned}\mathcal{T}' &= \{flies(x) \leftarrow bird(x) \wedge ab(x)^*,\\ &\quad bird(x) \leftarrow songbird(x),\\ &\quad eats\_insects(x) \leftarrow songbird(x),\\ &\quad frog(x) \leftarrow green(x) \wedge croaks(x),\\ &\quad ab(x) \leftarrow frog(x)\}\\ I &= \{\leftarrow ab^*(x) \wedge ab(x)\}\\ A &= \{ab^*\}\end{aligned}$$

Now, consider the query: $flies(Sam) \wedge eats\_insects(Sam)$. $\mathcal{T}$ with NAF yields the answer *true* as $\neg ab(Sam)$ can be proved by NAF.

The proof in the abductive framework is essentially equivalent till we reach the clause $\leftarrow ab^*(Sam)$. $ab^*(Sam)$ is abducible, if

$$\mathcal{T} \cup \{ab^*(Sam)\} \vdash flies(Sam) \wedge eats\_insects(Sam) \text{ and,}$$
$$\mathcal{T} \cup \{ab^*(Sam)\} \cup \{\leftarrow ab^*(Sam) \wedge ab(Sam)\} \text{ is satisfiable.}$$

By showing the consistency of $ab^*(Sam)$ this is proven.

Note: With the aid of this transformation, it is possible to also introduce metalogical statements as integrity constraints. For example, in [EK88b] the formula $Demo(\mathcal{T}, Q)$ is introduced that holds iff $\mathcal{T} \vdash Q$. The value is determined by reflection as described in [Wey80].

The authors have shown that abduction can be seen as an extension of logic programming by transforming programs working with negation as failure into an abductive framework that is at least as powerful as the original formulation. In order to drive the generation of abductive explanations Φ deduction is used, which ensures the soundness of the process, i.e., $\mathcal{T} \cup \Phi \vdash \Omega$, where Ω describes the facts to be explained.

2.3 The Knowledge-level Approach

The knowledge-level approach was proposed by Levesque (cf. [Lev89]) and is based on a model of belief. Considering implicit belief, this approach coincides with the general ideas underlying the logical models as presented in the preceding section. But furthermore, it allows the exchange of the model of belief, thus yielding a very general definition of abduction that does not depend on the respective knowledge representation.

We suppose that a propositional language $\mathcal{L}$ is given. Beliefs are formulated in the language $\mathcal{L}^*$, whose atomic sentences are of the form $B\alpha$ with $\alpha \in \mathcal{L}$. A subscript is used to distinguish different types of belief.

Definition 5: [Lev89] (epistemic state)
An *epistemic state* e determines which sentences of $\mathcal{L}$ are believed, i.e., $e \models B_\lambda \alpha$ for beliefs of type λ.

Explanation is defined as follows:

Definition 6: [Lev89] (explanation)
φ expl_λ ω with respect to e iff $e \models (B_\lambda(\varphi \supset \omega) \wedge \neg B_\lambda \neg \varphi)$.

Note: This means that φ is an explanation for ω if it is believed that φ implies ω, regarding material implication, and if the negation of φ is not believed. The last conjunct incorporates some kind of consistency check w.r.t. the belief set under consideration.

However, this definition allows multiple explanations. Therefore, a syntactic selection criterion is defined that will restrict the feasible explanations, thereby influencing the knowledge operator $EXPLAIN$ that is defined below. Following Occam's Razor, explanations with fewer propositional letters are preferred, where p and $\neg p$ are considered as different. More formally we have:

Definition 7: [Lev89] (literals of a formula)
The *set of literals* $LITS(\alpha)$ of a formula α is defined by
$LITS(false) = \emptyset$;
$LITS(p) = \{p\}$ for an atom p;
$LITS(\neg\alpha) = \{\ \overline{m} \mid m \in LITS(\alpha)\ \}$;
$LITS(\alpha \wedge \beta) = LITS(\alpha \vee \beta) = LITS(\alpha) \cup LITS(\beta)$.

Definition 8: ([Lev89]) (simplicity)
A formula α is *simpler* than a formula β (written $\alpha \prec \beta$) iff $LITS(\alpha) \subset LITS(\beta)$. $\preceq$ is defined straightforwardly.

With this in mind we are able to clarify what the statement "φ explains ω minimally" means.

Definition 9: [Lev89] (minimal explanation)
φ *minimally explains* ω with respect to an epistemic state e and a belief type λ, i.e., $\varphi\ min\_expl_\lambda\ \omega$ with respect to e iff

$$\begin{cases} \varphi\ expl_\lambda\ \omega \text{ with respect to } e \text{ and} \\ \text{there is no } \varphi^* \prec \varphi \text{ with } \varphi^* expl_\lambda\ \omega \text{ with respect to } e. \end{cases}$$

An abductive process should return the disjunction of all minimal explanations, i.e, as a semantic characterization we get

Definition 10: [Lev89] (explanation operator)

$$EXPLAIN_\lambda[\![e, \omega]\!] = \|\{\ \varphi \mid \varphi\ min\_expl_\lambda\ \omega \text{ with respect to } e\ \}\|,$$

where $\|\varphi\|$ is the set of all models in which φ is true.
An algorithm is called *correct* if it returns exactly all minimal explanations with respect to an epistemic state. Assume that the results of the abductive procedure are described by the function $explain(\mathcal{T}, \omega)$, i.e., $explain(\mathcal{T}, \omega)$ yields the explanations of ω with respect to the theory $\mathcal{T}$. Then, correctness can be characterized by the equation:

$$EXPLAIN_\lambda[\![\mathcal{R}_\lambda(\mathcal{T}), \omega]\!] = \|explain(\mathcal{T}, \omega)\|,$$

where the function $\mathcal{R}_\lambda$ maps sets of sentences into epistemic states with respect to the belief type λ.

The model described gives rise to different abductive procedures corresponding to the underlying notion of belief. In section 3.2, we will discuss explicit and implicit belief and the resulting connection between abduction and an Assumption-based Truth Maintenance System.

3 Hypotheses Generation

The most important part of an abductive procedure is the formation of a new theory that explains the observed data. In section 2.1 one such method was described in connection with the set-cover-based model. This, however, was not hypotheses generation but hypotheses assembly. Out of a set of predefined hypotheses, the feasible ones are selected

with the aid of a computable, predefined mapping e. The mapping e goes from hypotheses to data sets and determines accountability.

In this section we will investigate algorithms that do not rely on such a mapping, i.e., all relevant information is encoded in the underlying theory. We distinguish between the following methods for hypotheses generation that strongly depend on the respective formal model:

- generation by a form of linear resolution and
- generation with an Assumption-based Truth Maintenance System.

3.1 Generation by Resolution

Many authors have shown that forms of linear resolution can be used for the generation of hypotheses in logic-based models, e.g. [Pop73], [EK88a], [CP86]. In this context, we will outline the basic algorithm of Pople (cf. [Pop73]), the algorithm of Cox and Pietrzykowski (cf. [CP86]) and the abductive reasoning method of Shanahan (cf. [Sha89]) that incorporates a generalization of SLD-resolution.

3.1.1 Linear Resolution and Skolemization

The procedure of Cox and Pietrzykowski is based on linear resolution and reverse skolemization. This allows the explanation of facts that are described by arbitrary formulae, not just variable-free clauses (cf. [Esh88]).

Let $\mathcal{T}$ be a logical theory and let ω be a formula (the fact to be explained). $\mathcal{T}$ also denotes the conjunction of all formulae in $\mathcal{T}$. Furthermore, we assume $\mathcal{T} \nvdash \omega$, as otherwise ω is already explained by $\mathcal{T}$. A cause φ of ω in $\mathcal{T}$ is a formula determined by the following two conditions

- $\mathcal{T} \wedge \varphi \vdash \omega$ and
- φ is consistent with $\mathcal{T}$.

To compute the cause φ, do the following:

(1) Convert $\mathcal{T}$ in clausal form.

(2) Negate ω and perform linear resolution with input clauses from $\mathcal{T}$ and some element of $\neg\omega$ as top clause.

(3) We have assumed $\mathcal{T} \nvdash \omega$. Thus, we distinguish two cases:

(4) the deduction does not terminate,

(5) the deduction terminates with some dead-end, the clause d.

(6) Let D be the set of all dead-ends. Do the following steps for each dead-end d in D.

Negate d, thus obtaining a conjunction $p_1 \wedge \ldots \wedge p_n$ of literals.

Apply the reverse skolemization algorithm, i.e., replace all skolem terms in p_i by universally quantified variables. This yields the formula

$$\varphi :\equiv Q_1 x_1 \ldots Q_k x_k (q_1 \wedge \ldots q_n),$$

where the Q_i stand for the universal quantifiers. Then, φ is a cause for ω in $\mathcal{T}$.

Remarks:

- The algorithm incorporates a selection criterion. All computed causes φ are *basic*, which means that every cause φ' of φ itself is trivial, i.e., φ' implies φ directly. In most cases a minimality criterion is also satisfied. If φ is a computed cause then there is also no more specific cause x in the sense $x \supset \varphi$. This is a consequence of the resolution algorithm that works until no further resolution step can be made. An exception occurs only if a dead-end is subsumed by another.

- The algorithm is not complete, i.e., there exist cases where not all basic and minimal causes are found. For an example confer [CP86].

The criteria introduced for good explanations, in particular *basicness* and *minimality* will be presented in more detail in section 4.

Example:

(1) Take as given the example theory of section 2. Transforming the theory in clausal form yields:

$$\{\neg bird(x), ab(x), flies(x)\} \quad (9)$$
$$\{\neg ufo(x), flies(x)\} \quad (10)$$
$$\{\neg penguin(x), ab(x)\}\{\neg ostrich(x), ab(x)\} \quad (11)$$
$$\{\neg songbird(x), bird(x)\} \quad (12)$$
$$\{\neg songbird(x), eats\_insects(x)\} \quad (13)$$
$$\{\neg frog(x), eats\_insects(x)\} \quad (14)$$
$$\{\neg frog(x), green(x)\}\{\neg frog(x), croaks(x)\} \quad (15)$$
$$\{\neg frog(x), ab(x)\} \quad (16)$$

In addition, we assume as in section 2 that the different animal types are disjoint corresponding to our intuition, e.g., $\{\neg frog(x), \neg ostrich(x)\}$.

(2) Let the observation ω be given by

$$\omega \equiv \exists x (flies(x) \wedge eats\_insects(x))$$

We start resolution with some element of

$$\{\neg flies(F), \neg eats\_insects(F)\}$$

as top clause.

(3)-(5) Resolution with top clause $\neg flies(F)$ yields

$$\{\neg ufo(F), \neg eats\_insects(F)\} \qquad (*)$$
$$\{\neg bird(F), ab(F), \neg eats\_insects(F)\} \qquad (**)$$

Further resolving (*) does not lead to a dead-end, as the resolvents become inconsistent. A dead-end is reached only if (**) is resolved with (12) and (13). We get

$$\{\neg songbird(F), ab(F)\}$$

(6)

$$D = \{\{\neg songbird(F), ab(F)\}\}$$

Negating the elements of D and applying reverse skolemization gives the solution

$$\forall x(songbird(x) \wedge \neg ab(x)).$$

3.1.2 Problem Reduction by Linear Resolution

Instead of adding the negation of the observed data ω to the logical theory, Pople (cf. [Pop73]) tries to directly show $\mathcal{T} \supset \omega$. The observation is converted into disjunctive form. By regarding each disjunct separately the problem of proving ω is split into an equivalent set of subproblems. A clause in the logical theory is interpreted as a kind of rewrite rule and by backward chaining one arrives at nodes that will not create successor nodes, i.e., that cannot be proved from the axiom set. These are candidates for an explanatory hypothesis. If all different subproblems are considered, a set of possibly competing hypotheses is generated.

In detail Pople proceeds as follows:

(1) Convert the theory $\mathcal{T}$ to quantifier-free conjunctive normal form.

(2) Use skolemization of universally quantified variables to convert the observation ω in disjunctive normal form:

- Eliminate implication.
- Reduce the scope of negation.
- Replace each *universally* quantified variable by a skolem function that has as arguments the variables of any existential quantifiers ocuring before the universal quantifier.
- Drop existential quantifiers.
- Transform the resulting expression into disjunctive normal form.

(3) By considering all possible combinations for satisfying the disjunction ω, the problem can be split into n different but equivalent formulations, where n is the number of disjuncts. To see this, assume that

$$\begin{aligned} \omega &\equiv d_1 \vee \ldots \vee d_{n-1} \vee d_n \\ &\equiv (\neg d_1 \wedge \ldots \wedge \neg d_{n-1}) \supset d_n \end{aligned}$$

If the conjunction $\neg d_1 \wedge \ldots \wedge \neg d_{n-1}$ is added to $\mathcal{T}$ the problem is reduced to showing d_n. This process can be repeated with each d_i in turn as the right-hand side of the implication so that n different problem formulations are generated.

(4) Clauses in $\mathcal{T}$ are interpreted as productions with a single literal on the left-hand side. Every clause with l disjuncts allows the formation of l different rules in which each disjunct constitutes the left-hand side in turn.

(5) Try to show each subproblem by backward chaining. This can yield up to n different sets of dead-ends.

(6) Note that for each subproblem the conjecture d_i consists of a possibly unary conjunction. Now, those dead-ends are preferred that give the most coherent explanation of the observations, i.e., that account for the most conjuncts of those to be explained. This heuristics is incorporated in the reasoning mechanism by a selection process called *synthesis* or *factoring across partial trees.* It states that, whenever possible, literals occuring in the different proof trees built for each conjunct of the observation should be unified. The resulting unifier accounts for both top literals of the corresponding proof trees, i.e., both observations are explained. Among several feasible hypotheses those are always preferred that account for the most observations.

Example:

(1) Let $\mathcal{T}$ be given by

$$\{\neg bird(x), ab(x), flies(x)\} \tag{17}$$
$$\{\neg airplane(x), ab(x), flies(x)\} \tag{18}$$
$$\{\neg ufo(x), flies(x)\} \tag{19}$$
$$\{\neg penguin(x), ab(x)\}\{\neg ostrich(x), ab(x)\} \tag{20}$$
$$\{\neg songbird(x), bird(x)\} \tag{21}$$
$$\{\neg songbird(x), eats\_insects(x)\} \tag{22}$$
$$\{\neg frog(x), eats\_insects(x)\} \tag{23}$$
$$\{\neg frog(x), green(x)\}\{\neg frog(x), croaks(x)\} \tag{24}$$
$$\{\neg frog(x), ab(x)\} \tag{25}$$

(2) Let the observation ω be described by

$$\exists x((flies(x) \wedge eats\_insects(x)) \vee (eats\_insects(x) \wedge croaks(x)))$$

By skolemization we get

$$(flies(F) \wedge eats\_insects(F)) \vee (eats\_insects(F) \wedge croaks(F))$$

(3) The reformulation yields two subproblems, namely

(a) Add $\neg flies(F) \vee \neg eats\_insects(F)$ to the theory and show

$$eats\_insects(F) \wedge green(F) \wedge croaks(F).$$

(b) Add $\neg eats\_insects(F) \vee \neg green(F) \vee \neg croaks(F)$ to the theory and show

$$flies(F) \wedge eats\_insects(F).$$

(4) Assume $\mathcal{T}$ is reformulated as demanded, e.g., the first clause

$$\{\neg bird(x), ab(x), flies(x)\}$$

allows for three "production rules":

$$\begin{aligned} bird(x) &\supset ab(x) \vee flies(x) \\ \neg ab(x) &\supset \neg bird(x) \vee flies(x) \\ \neg flies(x) &\supset \neg bird(x) \vee ab(x) \end{aligned}$$

(5) Thus, the dead-ends for (a) are $\{frog(F)\}$ and for (b) $\{songbird(F) \wedge \neg ab(F)\}$

Remarks:

- With the synthesis process, Pople implements a selection procedure that obeys Occam's Razor in the sense that only those hypotheses are regarded that are strongly confirmed by several observable data.
- For diagnostic tasks, Pople suggests the choice of one abducible predicate, e.g., "$presence(X, Y)$," that explains the observed malfunction. In this case, synthesis is only regarded with respect to the occurrence of this predicate.
- Other heuristics for the controlled search for abductive explanations are estimated as useful, but are not yet implemented in the medical diagnosis system described by Pople.

3.1.3 Abduction and Default Persistence

The approach of Shanahan (cf. [Sha89]) will be briefly reviewed, as it incorporates a resolution-based abductive mechanism in a temporal reasoning system, in which prediction and explanation are realized by deduction respectively abduction within the same logical theory $\mathcal{T}$. This shows possible applications of abductive algorithms working with resolution.

The approach is based on Kowalski's and Sergot's Event Calculus (see [KS86]). The prediction problem consists of the search for a set of causal consequences ω of a set of events φ such that $\mathcal{T} \cup \varphi \models \omega$. If in turn ω describes the observed events, then abduction is used to find a feasible set φ of explanations satisfying $\mathcal{T} \cup \varphi \models \omega$.

As usual, the abductive algorithm requires that all causal relationships are expressed by formulae of the form "effect if cause." Furthermore, it is assumed that all properties known to be true have an explanation in terms of events. This embodies a kind of *default persistence*. To see this, assume that in order to explain $prop_1$ at time t_1 an event e_1 is introduced that must occur before t_1, i.e., at a time $t < t_1$. Properties can only be changed by another event, so $prop_1$ holds also at time $t_2 > t_1$, if no such event has been

observed. As Shanahan points out, this combination of abduction and persistence allows for the correct handling of some hard problems not solvable with other approaches to default persistence.

The abductive algorithm works with a kind of linear resolution. If ω_0 are the facts to be explained, then the problem can be formulated as follows: Find a set of unit clauses φ_n, called the *residue*, such that $\mathcal{T} \cup \varphi_n \models \omega_0$ and φ_n contains only abducible predicates. As usual, the set of abducible predicate symbols is predefined. As the procedure corresponds in general to the algorithm of Cox and Pietrzykowski described in section 3.1.1, it will not be outlined further.

In order to handle default persistence correctly, the mechanism must be extended. All negated assumptions derived by Negation as Failure must be recorded and checked in every later proof step, as abduction allows the addition of facts to the theory that may invalidate them. Shanahan also describes a method to cope with nested negation as failure. Essentially, it consists of a combination of normal SLD-Resolution that does not allow addition to the theory, and abduction.

3.2 An Assumption-based Truth Maintenance System as Abductive Procedure

In section 2.3 the basic ideas of the knowledge-level account for abduction were described. We have

$$EXPLAIN_\lambda[\![e, \omega]\!] = \|\{\ \varphi \mid \varphi\ min\_expl_\lambda\ \omega \text{ with respect to } e\ \}\|.$$

EXPLAIN gives a semantic characterization of explanation. So in order to develop an abductive algorithm, we need a syntactic counterpart that allows the actual computation of the simplest explanations.

In the following, we will give such a syntactic definition of explanation, and show how an abductive procedure can be developed, depending on a belief type λ. It will turn out that in the case of implicit belief, abductive reasoning can be modelled with an Assumption-based Truth Maintenance System (ATMS).

Levesque (cf. [Lev89]) defines the function ∇ for two sets of clauses Σ and Γ. If Γ is the set of clauses to be explained and Σ contains the believed clauses with respect to a belief of type λ, then $\nabla(\Sigma, \Gamma)$ determines the corresponding minimal explanations.

Definition 11: ([Lev89])
$\nabla(\Sigma, \Gamma) = \mu(\Phi)$, where Φ is a set defined by

$$\Phi = \left\{ \neg z \;\middle|\; \begin{array}{l} \forall y \in \Sigma, y \not\subseteq z \\ \forall x \in \Gamma, \exists y \in \Sigma, x \cap y \neq \emptyset \text{ and } (y - x) \subseteq z \end{array} \right\}$$

The function μ determines the smallest set with respect to subsets.

The function ∇ can be interpreted as follows: First, the clauses z searched for are not allowed to be expansions of some $y \in \Sigma$, the set of believed clauses. This is required in order to avoid inconsistencies, if $\neg z$ is later added to the belief set as an explanation. Additionally, z must satisfy another condition. If it is actually possible to explain all

observations in Γ, then there must exist for every observation $x \in \Gamma$ a belief $y \in \Sigma$ that, at least partially, accounts for it, i.e., x and y have common literals, $x \cap y \neq \emptyset$. The clauses z are chosen in such a way that they contain every literal of such a y that is not a part of the corresponding observation x. If, in the following, z is negated and added to the belief set Σ, this has the consequence that the observations of Γ are forced to be true. The formula $\neg z$ denies exactly those disjuncts of the clauses y that do not account for the observations. Thus, the remaining disjuncts of each y that are part of the observations have to be true. If different clauses z can be found, they constitute alternative explanations. Finally, the function μ that is applied to the set of clauses $\neg z$ determines the minimal set with respect to set inclusion. This obeys the principle that an explanation should not contain more literals than necessary.

Example: Let Σ be given by

$$\begin{aligned}
&\{\neg bird(x) \vee ab(x) \vee flies(x)\}\\
&\{\neg songbird(x) \vee bird(x)\}\\
&\{\neg songbird(x) \vee eats_insects(x)\}\\
&\{\neg frog(x) \vee eats_insects(x)\}\\
&\{\neg frog(x) \vee green(x)\}\\
&\{\neg frog(x) \vee ab(x)\}
\end{aligned}$$

As usual, we assume that corresponding to our intuition the extension of predicates that model animals are disjoint, i.e., we also have axioms of the form $\{\neg frog(x), \neg songbird(x)\}$. The proposition ω to be explained is given by

$$\omega \equiv \{eats\_insects(F), flies(F) \vee green(F)\}$$

The set $y - x$ of definition 11 is determined by

$$\begin{aligned}
y - x = \ \{\ &\{\neg songbird(F) \vee \neg bird(F) \vee ab(F)\},\\
&\{\neg frog(F) \vee \neg bird(F) \vee ab(F)\},\\
&\{\neg frog(F)\},\\
&\{\neg songbird(F) \vee \neg frog(F)\ \}
\end{aligned}$$

The second and fourth clause are already elements of Σ, but this contradicts $\forall y \in \Sigma, y \not\subseteq z$ as $(y - x) \subseteq z$. Thus, Φ is given by

$$\begin{aligned}
\Phi = \ \{\ &\{songbird(F) \wedge bird(F) \wedge \neg ab(F)\},\\
&\{frog(F)\ \}
\end{aligned}$$

This can be simplified to

$$\begin{aligned}
\Phi = \ \{\ &\{songbird(F) \wedge \neg ab(F)\},\\
&\{frog(F)\ \}
\end{aligned}$$

As Φ is already minimal with respect to subsets, we get $\nabla(\Sigma, \Gamma) = \Phi$.

∇ is only defined for clauses, so it can not be used in developing a general abductive procedure. But Levesque has shown that for a certain class of beliefs the simplest explanations are in fact the result of an application of ∇. We will first characterize the beliefs that can be handled.

Definition 12: ([Lev89]) (regular beliefs)
A type of belief λ is *regular*, iff for every epistemic state the following sentences of $\mathcal{L}^*$ are true:

- $B_\lambda \neg \Box$, where $\Box$ denotes the empty clause;
- $(B_\lambda \alpha \vee B_\lambda \beta) \supset B_\lambda(\alpha \vee \beta)$;
- $(B_\lambda \alpha \wedge B_\lambda \beta) \supset B_\lambda(\alpha \wedge \beta)$;
- $B_\lambda(\alpha \wedge \beta) \supset (B_\lambda \alpha \wedge B_\lambda \beta)$;
- $B_\lambda \alpha \equiv B_\lambda \alpha^*$, if α^* is α in conjunctive or disjunctive normal form, or if α is the result of replacing any subformula β in α by β^*, where (recursively) $B_\lambda \beta \equiv B_\lambda \beta^*$ is always true.

For regular beliefs, the simplest explanations can be computed by applying ∇ to the belief set and the transformation of the observations in conjunctive normal form (CNF).

Theorem 1 ([Lev89]) *For regular belief*

$$EXPLAIN_\lambda[\![e, \omega]\!] = \|\nabla(\{y \mid e \models B_\lambda y\}, CNF(\omega))\|.$$

We will now separately consider implicit and explicit belief and the resulting connection between the operation ∇ for the respective belief type and an Assumption-based Truth Maintenance System (ATMS).

3.2.1 Implicit Belief

Implicit belief will be denoted by the belief operator B_I. A corresponding epistemic state e is determined by a set of assignments r. Implicitly believed formulae are characterized by

$$e \models B_I \alpha \text{ iff for every assignment } r \in e,\ r \models \alpha,$$

i.e., if α is believed in an epistemic state e, then it must be true in all assignments r that charaterize e. This implies that exactly those formulae are believed that are logical consequences of the underlying theory:

$$e \models B_I \alpha \text{ iff } \mathcal{T} \models \alpha,$$

where $e = \mathcal{R}_I(\mathcal{T})$ (see section 2.3). In section 2.3 we defined explanation with respect to an arbitrary belief type as follows:

$$\varphi \text{ explains } \omega \text{ in } e \text{ iff } e \models B_\lambda(\varphi \supset \omega) \wedge \neg B_\lambda \neg \varphi.$$

If we apply this to the above defined implicit belief, we get

$$\varphi \; expl_I \; \omega \text{ with respect to } e \text{ iff } \mathcal{T} \models (\varphi \supset \omega) \text{ and } \mathcal{T} \not\models \neg\varphi.$$

This is equivalent to the definition of abduction in logic-based approaches:

$$\mathcal{T} \cup \{\varphi\} \models \omega \text{ and } \mathcal{T} \cup \{\varphi\} \text{ is consistent.}$$

With the function ∇ we already indicated a computational procedure for the generation of abductive explanations. Thus, the above shows that the further development that is based on the function ∇ will also be applicable to logic-based approaches.

The result of ∇ is reminiscient of the support sets used in an ATMS, and Levesque shows that in fact "an ATMS can be understood as computing all simplest explanations with respect to implicit belief" (cf. [Lev89]). To see this, we first show that an ATMS procedure with respect to a set of clauses Σ and a symbol p can be defined in terms of ∇.

Theorem 2 ([Lev89])

$$atms[\Sigma, p] = \nabla(Th(\Sigma), \{\{p\}\}).$$

$Th(\Sigma)$ is the deductive closure of theory Σ.

This result can be generalized in order to treat clauses as second argument.

Definition 13: ([Lev89]) (generalized ATMS)
A generalized ATMS procedure for a set of clauses Σ and a clause β is defined by

$$gatms[\Sigma, \beta] = \nabla(Th(\Sigma), CNF(\beta)).$$

Implicit belief is regular. Thus, if Σ is taken as the set of beliefs in an epistemic state e, theorem 1 yields the following for a correct abductive procedure:

Lemma 1 ([Lev89])

$$\|EXPLAIN_I[\![\mathcal{R}_I(\Sigma), \omega]\!]\| = \|gatms(\Sigma, \omega)\|.$$

Proof: For a correct abductive procedure we have

$$\|EXPLAIN_I[\![\mathcal{R}_I(\Sigma), \omega]\!]\| = \|EXPLAIN_I[\![e, \omega]\!]\|.$$

According to theorem 1, this is equivalent to

$$\|\nabla(\{y \mid e \models B_I y\}, CNF(\omega))\|.$$

Implicit belief is defined in such a way that $e \models B_I\alpha$ iff $\Sigma \models \alpha$, when $\mathcal{R}_I(\Sigma)$ determines the epistemic state e. So, the above is equivalent to

$$\|\nabla(\{Th(\Sigma)\}, CNF(\omega))\|.$$

The definition of a generalized ATMS completes the proof.

The lemma proves that an ATMS can be used to generate explanations in a model for implicit belief and, as the equivalence was shown, an ATMS can also be used in logic-based models. But in section 5.1 the exponential time of the ATMS procedure will be discussed. This suggests the consideration of another kind of belief that might be computationally more tractable.

3.2.2 Explicit Belief

If α is implicitly believed, then all logical consequences of α are believed as well. In contrast, explicit belief in α only sanctions the belief in all tautological consequences in the sense of *relevance logic*, which allows contradictory information in the knowledge base.

An epistemic state for explicit belief is defined with the aid of *situations*. Situations are total functions that assign a truth value to literals p, where at least p or $\neg p$ is assigned the value 1, but possibly both. Thus, a situation is in general not a valid assignment. A literal follows from a situation s, iff it is assigned the value 1: $s \models p$ iff $s(p) = 1$ and $s \models \neg p$ iff $s(\neg p) = 1$. Connectors and negation are defined as follows:

$$\begin{aligned} s \models (\alpha \wedge \beta) &\quad \text{iff} \quad s \models \alpha \text{ and } s \models \beta; \\ s \models \neg(\alpha \wedge \beta) &\quad \text{iff} \quad s \models \neg\alpha \text{ or } s \models \neg\beta; \\ s \models \neg\neg\alpha &\quad \text{iff} \quad s \models \alpha. \end{aligned}$$

An epistemic state is determined by a set of situations. This yields the following characterization for explicit belief:

$$e \models B_E\alpha \text{ iff for every } s \in e,\ s \models \alpha.$$

The function $\mathcal{R}_E$ is defined similar to $\mathcal{R}_I$ (see section 3.2.1). For a given knowledge base $\mathcal{T}$, it determines the set of all situations that satisfy $\mathcal{T}$, i.e., the corresponding epistemic state. For $e = \mathcal{R}_E(\mathcal{T})$ we get

$$e \models B_E\alpha \text{ iff } \mathcal{T} \cup \mathcal{S} \text{ tautologically entails } \alpha \text{ in the sense of relevance logic,}$$

where $\mathcal{S}$ is the set of all clauses of the form $\{p, \neg p\}$. This set is added to ensure that at least one of p or $\neg p$ is assigned the value 1.

The abductive explanations for explicit belief can also be described with the function ∇. But instead of using the logical closure of Σ as belief set, we now need the set $EXPS(\Sigma) = \{y \mid y \text{ is tautologous or } \exists y^* \in \Sigma, y^* \subseteq y\}$.

Definition 14: ([Lev89]) (explanation with respect to explicit belief)
Let Σ be a theory and β a clause.

$$abd[\Sigma, \beta] = \nabla(EXPS(\Sigma), CNF(\beta)).$$

The next theorem states that *abd* in fact yields the simplest explanations with respect to explicit belief.

Theorem 3 ([Lev89])

$$EXPLAIN_E[\![\mathcal{R}_E(\Sigma), \beta]\!] = \|abd[\Sigma, \beta]\|.$$

Levesque shows that generating the explanation of a single clause with respect to explicit belief is computationally easier than the corresponding ATMS procedure for implicit belief. A theorem that relates explicit to implicit beliefs allows the use of *abd* in those cases to compute the explanations for implicit beliefs by recursively computing explicit beliefs.

It has been shown that an ATMS is one possible tool to generate abductive explanations. If we take the results of Selman and Levesque into account, presented in section 5.1, then we will see that an ATMS is computationally also the best procedure that can be expected. This suggests lowering the demands in favour of higher efficiency, e.g., by finding *some* explanation instead of all possible ones. The knowledge-level account suggests further ideas that point in this direction. Considering explicit belief, a computationally simpler procedure can be found that also allows one to recursively compute implicit belief. In addition, this is a very general approach that permits the investigation of further belief types.

4 Selection of Hypotheses

The result of an abductive procedure is a *set* of possible explanations. As this set can be quite extensive, it seems reasonable to discard the "less interesting" hypotheses in order to gain efficiency in the following computational process. Most abductive procedures described in the previous sections already incorporate such a selection criterion (e.g., [CP86], [ATBJ87]), to prevent, for instance, the generation of the observations themselves as trivial explanations.

Selection principles are based on heuristics that try to determine good explanations. What seems promising depends partly on the application domain, e.g., fault diagnosis systems should yield the most specific explanation for a malfunction, whereas other tasks may require more abstract hypotheses. But syntactic criteria also exist that can be applied to both domains, e.g., in order to exclude trivial explanations.

The philosopher Peirce (see [Gou50]) claims that those hypotheses should be selected that correspond to Occam's Razor in the sense that they are the psychologically simplest, i.e., the most intuitive explanations. This seems hard to realize, for which reason Occam's Razor is normally interpreted as meaning logical and syntactical simplicity. Hypotheses containing superfluous literals or hypotheses that are subsumed by others are unwanted. The negative results of Levesque (cf. [Lev89]) confirm that no more can be expected, because he shows that it is impossible to formulate a selection criterion on purely semantic grounds.

In the following, we will investigate different approaches for finding the most promising explanations. The greatest part is formed by methods that realize in some way Occam's Razor. Only recently has an alternative been proposed that is based on a metric of coherence. This approach will also yield a method for determining an appropriate level of specificity of an explanation.

4.1 Simplicity Criteria

Independent of the respective criteria that determine a level of specificity, i.e., the level of abstraction of an explanation, in general, all accepted hypotheses have to satisfy some basic conditions that formalize syntactical simplicity. Representative are the ones that Cox (see also section 3.1) demands for the hypotheses generated by his algorithm.

Let $\mathcal{T}$ be a first-order knowledge base and Φ a set of hypotheses for the observation ω. If $\varphi \in \Phi$ is interesting, it should satisfy the following conditions:

Consistency: $\mathcal{T} \wedge \varphi$ is satisfiable. This is required in all logic-based models.

Non-Triviality: $\neg\varphi \supset \omega$, i.e., the observation is not a direct consequence of the hypotheses. In particular, this excludes that ω itself is synthesized as a feasible explanation.

Basicness: Every consistent explanation of φ itself is trivial. This favours the most specific explanation, in the sense that there is no non-trivial explanation for the explanation itself.

Minimality: For all hypotheses φ' of ω: $\varphi \supset \varphi'$ implies $\varphi' \equiv \varphi$, i.e., there exists no more general hypothesis for ω than φ, in the sense that all superfluous literals or unnecessary universal quantifications are omitted from explanations.

Given, for example, $croaks(x) \supset frog(x)$ as explanation, $croaks(x) \wedge green(x) \supset frog(x)$ is not minimal.

Note:

- If $\varphi \supset \psi$ is a part of the theory, then ψ will not be among the generated hypotheses. The reason for this lies in the resolution algorithm that chains backward as far as possible, thus reaching only φ as possible dead-end. This means that such a resolution-based method always guarantees basicness, and non-minimality can only occur in the cases mentioned.
- Basicness already realizes a strategy for determining an appropriate level of specificity of the explanation set that is called *most-specific abduction* (see also below).

Explanations satisfying consistency, minimality and non-triviality realize Occam's Razor, if it is interpreted syntactically. But the set of interesting hypotheses selected this way can still be quite extensive and usually an additional selection is made to prefer hypotheses of a certain level of specificity. Appelt and Pollack consider in [AP90] two different sorts of criteria:

- global criteria, and
- local criteria.

Global Criteria

Selection methods based on global criteria consider the assumption set as a whole. We distinguish (cf. [AP90]):

- cardinality comparisons,
- least presumptive or least specific abduction,
- most specific abduction, and
- minimal abnormality.

Cardinality comparisons: These are used only in diagnostic systems and ensure that those hypotheses are assumed that imply the failure of the smallest number of components. Thus, the application system has to be specified in terms of distinguishable components whose input and output behaviour is completely determined (see [AP90]). As Appelt and Pollack state, this is not the case for natural languange understanding or planning resp. plan recognition.

Least and most presumptive explanations: Less presumptive or less specific is defined as follows: Let h_1 and h_2 be hypotheses and $\mathcal{T}$ a logical theory, then h_1 is less presumptive or less specific than h_2 iff $\mathcal{T} \cup h_2 \models h_1$. Thus, the least presumptive or specific explanations are those that provide the most general explanation. An abductive process that realizes this form of selection was proposed by Stickel (cf. [Sti90]).

Alternatively, the *most specific explanations* can also be chosen, i.e., those that themselves have no more non-trivial explanations (see also the definition of basicness above). Above all, this strategy seems to be useful for fault diagnosis, where very detailed knowledge about the origin of the failure is required.

Appelt suggests that for plan recognition and mental state ascription, a combination of most and least specific abduction seems adequate. Nevertheless, there is still the problem of deciding which strategy should be applied in which particular case. Furthermore, it remains possible that there exists no single least or most specific explanation which forces a further selection.

Minimal abnormality: In [Poo89], Poole defines an abductive framework that relies on the specification of abnormality and normality assumptions. As Appelt claims, an inherent problem of this approach is the fact that minimally abnormal assumptions may be inconsistent with the "best" explanation.

Local Criteria

To avoid the shortcomings of those approaches local criteria have been introduced that allow one to single out one "best" hypothesis. We have

- Bayesian statistical methods,
- weighted abduction, and
- cost-based abduction.

Bayesian statistical methods: Standard probability theory can be used to single out the "most probable" hypothesis. However, the set of possible hypotheses must be determined in advance with each hypothesis having a probability assigned. In general, other approaches are preferred, since this method is computationally too expensive.

Weighted abduction: Appelt and Pollack (cf. [AP90]) overcome this difficulty by adapting a method developed by Hobbs, Stickel et al. (see [HSME89]). To guide the application of rules, they assign weighting factors to all literals in the premise. Thus, rules have the form

$$\varphi_1^{\mu_1} \wedge \ldots \wedge \varphi_n^{\mu_n} \supset \psi.$$

The weights μ_i are used to compute the assumption cost of literals. This is done by multiplying the assumption cost of the consequent by the weighting factor of the considered

literal in the premise, e.g., if ψ has cost c then φ_1 can be assumed at cost $c * \mu_1$. In the abductive procedure the assumption set with the lowest cost is preferred.

Appelt snd Pollack also give a model-theoretic semantics of their approach based on model preference. By weighted abduction the models of a theory $\mathcal{T}$ are restricted in such a way that those models are filtered out that are inferior according to the model preference constraints. Thereby, an underlying partial preference order on the models of $\mathcal{T}$ is assumed. The weights in the rules are interpreted as additional constraints on this order. If the rule $\varphi^{\alpha} \supset \psi$ is given with weight $\alpha < 1$, this means that every model satisfying $\varphi \wedge \psi$ is preferred to some model satisfying $\neg\varphi \wedge \psi$.

Charniak and Shimony (cf. [CS90]) note that there is no semantics given for the case $\alpha > 1$, so only a kind of most-specific abduction can be modelled. Furthermore, the rule weightings must result in a consistent ordering without cycles, which makes it even more difficult to determine the weights of newly added rules.

Cost-based abduction: Charniak and Shimony (cf. [CS90]) present an alternative approach based on a probabilistic semantics for cost-based abduction. They try to find the best explanation of an observation by finding a minimal cost proof of it. This method corresponds in general to the one presented by Hobbs et al. (cf. [HSME89]) with the advantage that a suitable semantics is given in terms of a Boolean belief network.

The above described ideas show that if broader applications are intended, the assignment of cost or weighting factors seems a promising strategy. But, as also becomes obvious, the respective application domain must be formalized very carefully in order to avoid unintuitive results as a consequence of badly chosen weighting factors or cost.

In the following, we will investigate another approach that judges the quality of explanations with a *coherence metric*. This method also yields an alternative to determine an appropriate level of specificity.

4.2 Explanatory Coherence as Selection Criterion

Instead of implementing Occam's Razor as syntactical simplicity, Ng and Mooney (see [NM90]) have determined a metric of coherence that helps to find the most intuitive explanations. The search for explanations is guided by heuristics (beam search), in order to increase efficiency. The metric also determines the level of specificity.

As the authors claim, Occam's Razor is insufficient for text understanding and plan recognition. They suggest selecting those explanations that best "tie together" all observations. This means that they prefer "those with more connections between any pair of observations," where connections are interpreted as directed paths in the proof graph. More precisely, the metric is defined as follows:

Definition 15: ([NM90]) (coherence metric)
The *coherence metric* C is defined by

$$C = \frac{\sum_{1 \leq i < j \leq l} N_{i,j}}{N \binom{l}{2}},$$

where $\mathbf{l}$ is the number of observations, $\mathbf{N}$ the number of nodes in the proof graph, and $\mathbf{N_{i,j}}$ the number of distinct nodes in the proof graph such that there exists a sequence of directed edges from n_k to n_i and from n_i to n_j, where n_i and n_j are observations. The sequences may be empty.

Remarks:

- $N \binom{l}{2}$ is a scaling factor ensuring that the final value lies between 0 and 1.
- The numerator is the sum of the number of nodes in a graph that simultaneously support the interpretation represented by this graph. To determine a "good" explanation all possible connections of nodes, i.e., several different proof graphs, are regarded. The interpretation represented by the graph with the highest coherence is chosen.
- The metric can be computed in time $O(lN + e)$ by using depth-first search.

As the authors point out, this approach has several advantages. On the one hand, coherent explanations are usually also syntactically simpler explanations. This is the case because unification is favoured by preferring tight connections. Furthermore, this method does not allow "too many degrees of freedom," as is the case, e.g., with weighted abduction where the rule weights can be chosen arbitrarily. In addition, the metric helps to determine a level of specificity, since only the proof of a subgoal is attempted by backward chaining if this will increase the overall coherence.

An algorithm using this metric is implemented in the system ACCEL (Abductive Calculation of Causal Explanations for Language). To gain efficiency, the authors envisage the use of an ATMS.

Explanatory Coherence appears to be very promising for systems that incorporate in some way human reasoning capabilities, i.e., systems whose behaviour cannot be determined for all cases in advance. The authors show that it can be applied successfully in the field of natural language understanding. But, as they also suggest, the most promising approach is one that uses both coherence and likelihood information in order to be able to cope with incomplete or vague information.

5 The Relationship between Abduction and Default Reasoning

Abduction is the process of finding plausible explanations for some observed events. If the reasoner becomes more experienced, i.e., more facts become known, it is possible that previously assumed explanations have to be rejected. Thus, a non-monotonic mechanism is needed to keep track of the "plausible" hypotheses.

Default logic seems to be a non-monotonic formalism that can be quite easily adapted to the problem of explanation finding. It is possible to interpret defaults as predefined hypotheses and reasoning with them as a simple kind of theory formation. This view of default logic corresponds to the abductive approach of Poole (see [Poo88]) that is further investigated in the following. We have

Definition 16: [Poo88] (scenario)
Let $\mathcal{T}$ be a set of closed first-order formulae and Δ an arbitrary set of first-order formulae. A *scenario* of $\mathcal{T}$ and Δ is a set $D \cup \mathcal{T}$ where D is a set of ground instances of elements of Δ such that $D \cup \mathcal{T}$ is consistent.

Definition 17: [Poo88] (explanation)
A closed formula ω is *explainable* from a scenario of $\mathcal{T}$ and Δ if there is a set of ground instances D of elements from Δ such that

- $\mathcal{T} \cup D \models \omega$, and
- $\mathcal{T} \cup D$ is consistent.

The set $\mathcal{T} \cup D$ is called an *explanation* of ω.

The correspondence to the logic based model for abductive reasoning is obvious at first sight. A set of hypotheses out of Δ is accepted as an explanation if it can be consistently added to the theory $\mathcal{T}$ and the observation can then be proved. The sentences in $\mathcal{T}$ are used as a kind of constraint for the hypotheses in Δ. They determine which elements of Δ are feasible hypotheses.

Definition 18: [Poo88] (extension)
An *extension* of $\mathcal{T}$ and Δ is the set of logical consequences of a maximal scenario of $\mathcal{T}$ and Δ with respect to set inclusion.

This definition can be related to the notion of extension in default logic. Poole shows that if for the default

$$\frac{\varphi : \psi}{\psi}$$

the formula $\varphi \supset \psi$ is added to the set Δ, his notion of extension corresponds to the one defined by Reiter (cf. [Rei80]). For general defaults there is no exact translation, but Poole argues that in those cases where the approaches give different results, Reiter's method yields nonintuitive conclusions. This pathological case occurs if the underlying theory contains only disjunctive information about the prerequisites of defaults. Consider, e.g., the theory $\{p(a) \vee q(a)\}$ and the defaults

$$\frac{p(x) : r(x)}{r(x)}, \quad \frac{q(x) : r(x)}{r(x)}.$$

Then, no default is applicable as no single disjunct can be proved. However, Poole claims that the knowledge that the disjunction of the prerequisites is true is enough to sanction the belief that the common conclusion $r(a)$ of the defaults is also true. Thus, in his system the corresponding inference is allowed.

But, if abductive approaches are compared with default reasoning systems in more detail some differences are striking. In general, defaults yield several extensions and in order to be able to use them for further derivations, an attempt is made to rule some of them out by assigning priorities. Abductive systems use heuristics to constrain the set of feasible hypotheses but nevertheless multiple hypotheses are not always undesirable. The origin of this discrepancy lies in the different tasks of the two approaches. Hypotheses generated by abduction are used for explanation whereas defaults are used in connection

with incompletely specified knowledge bases, in order to allow at least non-monotonic conclusions. Thus, each alternative extension of a default theory determines a maximal extension of the knowledge base. In contrast, abductive hypotheses are in general not mutually exclusive and only partially determine the underlying theory (cf. [Esh88]). The fact that the approaches of Poole and Reiter coincide for normal defaults has its origin in Poole's special definition of an extension. An extension is a *maximally consistent* scenario and hence different sets of hypotheses are in fact orthogonal, i.e., inconsistent.

As a conclusion, one could point out that default reasoning can be classified within the framework of theory formation if defaults are viewed as hypotheses. But abduction constitutes the more general approach in the sense that it allows more flexibility by admitting partial extensions.

5.1 The Complexity of the Abductive Task

Some results concerning the complexity of abduction can be received by filtering out a common subtask of abductive procedures and methods for computing default extensions.

In section 3.2 an Assumption Based Truth Maintenance System (ATMS) was shown to constitute one possibility for generating abductive hypotheses in a logic based model. Selman and Levesque prove in [SL89] that the exponential time needed by the ATMS procedure is not a consequence of a possibly exponential number of hypotheses. Even when the explanations are restricted to those of a predefined set A of abducible sentences, the task is NP-hard. The crucial point is the so-called *support selection task*, where support of the observations has to be found. This task is shown to also be a subtask of every algorithm determining default extensions that contain a given set of propositions A, i.e., for so-called *goal-directed reasoning*.

Assume that a Horn clause theory $\mathcal{T}$, a set of abducible predicate symbols A, and a symbol ω are given. An ATMS computes all explanations φ for ω with respect to the theory $\mathcal{T}$ in such a way that the symbols of α are in A and φ is a minimal set of literals satisfying

$$\mathcal{T} \cup \varphi \models \omega \text{ and } \mathcal{T} \cup \varphi \text{ is consistent.}$$

It is shown (cf. [SL89]) that finding such an assumption-based explanation, as well as generating explanations at all, is NP-hard. This means that even for Horn-clause theories there is probably no improvement in efficiency. The responsible subproblem can be singled out:

Support Selection Task:

Let $\mathcal{T}$ be a set of Horn clauses, A a predefined set of predicates and ω a predicate symbol. Find a set of literals φ, the so-called *support set* such that

- $\mathcal{T} \cup \varphi \models \omega$;
- $\mathcal{T} \cup \varphi$ is consistent;
- φ contains only symbols from A.

An assumption based explanation for an observation q is a minimal support set of ω. But the support selection task can also be used to describe default extensions. Let a default theory Δ be defined by $(\mathcal{T}, D)$, where all defaults in D have the form

$$\frac{:p}{p}$$

with $p \in A$. Then $Th(\mathcal{T} \cup \varphi)$, the deductive closure of $\mathcal{T}$ augmented with φ, is an extension of Δ if and only if φ is a maximal support set of ω (see [SL89]).

Both minimality and maximality do not add to the computational difficulty. In both cases the support selection task is responsible for the intractibility. Concerning abductive explanations, an improved algorithm can be found if not all explanations are computed, but only one. For instance, for Horn theories *some* non-trivial explanation can be found in polynomial time.

Thus, to retain efficiency *credulous reasoning*, i.e., generating one extension and one set of compatible hypotheses instead of all possible ones, should be favoured over goal-directed reasoning. This result applies both to abduction and default reasoning, since they have the support selection task in common.

6 Applications to Planning and Plan Recognition

Planning or plan recognition is one possible application of abductive reasoning that requires a quite general and sophisticated abductive method because of the wide range of possible application domains. In this section, we will investigate two logic-based approaches to planning and plan recognition that use abduction as problem solving strategy. The model presented by Esghi (cf. [Esh88]) uses abduction with metalevel integrity constraints instead of negation as failure (see also section 2.2.2). The second approach to be described presents a theory for plan recognition as abduction and relevance (cf. [HK90]).

6.1 Abduction and the Event Calculus

Instead of following the usual approach and employing situation calculus and deduction, Eshghi (see [Esh88]) develops a planning algorithm with the event calculus and abduction. His approach solves the frame problem, allows the generation of non-linear plans, and is able to incorporate an ATMS and a least-commitment strategy for efficient search.

The context of abduction is the same as described in section 2.2.2. An abductive framework $(\mathcal{T}, I, A)$ consists of a Horn clause theory $\mathcal{T}$ without denials, a set I of integrity constraints, and a set A of abducible predicate symbols. A set of sentences Φ is an abductive solution for the framework $(\mathcal{T}, I, A)$ and the query ω, iff

- Φ consists of a set of variable free abducible atoms,
- $\mathcal{T} \cup \Phi \vdash \omega$,
- $\mathcal{T} \cup \Phi \cup I$ is satisfiable.

Backward chaining with resolution is used for the generation of assumptions. A dead-end R consisting exclusively of abducible predicates is called a *residue*. $\mathcal{T} \cup \neg R$ entails the formula to be explained.

In general, $\neg R$ is existentially quantified and thus needs skolemization. In order to be able to substitute for the skolem constants in the further reasoning process an equality theory is required. Eshghi argues that if all clauses in $\mathcal{T}$ are *homogenised* (see below), then a restricted equality theory that contains only equality axioms and no schemata for inequality is sufficient. The schemata should be avoided, because they lead to an explosion of the search space and render the procedure inefficient.

Definition 19: ([Esh88]) (homogeneous clauses)
A clause is homogeneous iff

- there are no constant symbols in its head atom, and
- no variable symbols occur more than once in the head.

For example, the homogenised form of the clause $P(a)$ is $\forall x((x = a) \supset P(x))$.

Remark: Eshghi gives an algorithm for how to transform arbitrary clauses into homogenised form.

The final algorithm for finding the set of explanations Δ for a goal G with integrity constraints I of the form $\mathcal{T} \cup \Delta \vdash R_1(x) \leftarrow \mathcal{T} \cup \Delta \vdash R_2(x)$ consists of five phases. We will first outline the complete procedure and in the following give some explanations and an example.

(1) Abductive phase:

$GS \longleftarrow \{G\}$; $\Delta \longleftarrow \{\}$;

repeat until $\Delta' = \{\}$

find Δ' such that $\mathcal{T} \cup \Delta \cup \Delta' \vdash GS$;

By this algorithm a residue is found.

(2) $\Delta \longleftarrow \Delta \cup \Delta'$

(3) Consistency checking phase:
Test with an arbitrary checking algorithm whether $\mathcal{T} \cup \Delta \vdash R_1(x) \leftarrow R_2(x)$ is consistent. If the answer is no, remove clauses from Δ until the clause is consistent.

(4) Precondition determination phase:
if no inconsistency was found in (3)

then determine the instantiated preconditions (bindings $B_1, B_2, \ldots$) for $R_2(x)$ in $\mathcal{T} \cup \Delta'$, i.e., find the bindings for x such that $\mathcal{T} \cup \Delta' \vdash R_2(x)$

else determine these bindings $B_1, B_2, \ldots$ in Δ. (Note that clauses from Δ have been removed in order to guarantee consistency.)

(5) **if** no inconsistency was found in 3.

then $GS \leftarrow \{R_1(B_1), R_1(B_2), \ldots\}$

else $GS \leftarrow \{G, R_1(B_1), \ldots\}$

This algorithm follows from a theorem about the unsatisfiability of the integrity constraints:

Theorem 4 ([Esh88]) *An integrity constraint* $\mathcal{T} \cup \Delta \vdash R_1(x) \leftarrow \mathcal{T} \cup \Delta \vdash R_2(x)$ *is unsatisfiable iff*

a) $\mathcal{T} \cup \Delta \cup \{R_1(x) \leftarrow R_2(x)\}$ *is inconsistent, or*

b) *for some* β: $\mathcal{T} \cup \Delta \vdash R_2(\beta) \wedge \mathcal{T} \cup \Delta \nvdash R_1(\beta)$.

Remarks:

- As a consequence, case a) is first tested in (3). If no inconsistency is found, the preconditions can be determined in $\mathcal{T}$, plus the assumptions generated by abduction. We proceed then to ensure case b) and try to find a hypothesis for $R_1(B_1), R_1(B_2), \ldots$ in the next iteration.
- If inconsistencies occur in step (3), then clauses have to be removed from Δ. (4) is executed with respect to this new theory that guarantees consistency. In the next iteration besides $R_1(B_1), R_1(B_2), \ldots$ the goal G still has to be explained.
- As the author points out, an ATMS will be of great use for recording the relationship between assumptions and goals and for determining the assumptions to be rejected in step (3).

For the purpose of planning, Eshghi combines this algorithm with a variant of Kowalski's and Sergot's event calculus (cf. [KS86]). Events are the basis of the ontology and represent points in time. Actions are associated with events with the predicate "*action*." The predicate "*holds*" is defined as usual, and "*postcon*" describes the postconditions of actions. The calculus will not be introduced in more detail; it is assumed that the meaning of the predicates is intuitively clear from their names. The relationship between facts in the domain are described with the predicates *holds*, $=$ and $\neq$. The theory has Horn clause format with denials. We have the following domain-independent axioms:

$$initiates(t, prop) \leftarrow action(com, t), postcon(com, effect) \quad (26)$$

$$holds(prop, c(t)) \leftarrow initiates(t, prop) \quad (27)$$

$$t \leq c(t) \quad (28)$$

$$c(t) \leq t_1 \leftarrow t < t_1 \quad (29)$$

where $c(t)$ denotes the earliest time point after the execution of an action at which the postcondition is true.

Preconditions are expressed as integrity constraints stressing the fact that they must be believed to be true by the planning agent. This is formalized by the axiom

$$\mathcal{T} \cup \Delta \vdash holds(prop, t) \leftarrow \mathcal{T} \cup \Delta \vdash action(com, t) \wedge precon(com, prop) \quad (30)$$

Furthermore, a partial order on events is defined where the initial state is characterized by a special time point $INITIAL$. The predicate $persists(prop, t_1, t_2)$ is introduced to formalize persistence. An axiom states that a proposition $prop$ is true at time t_2 if $prop$ is initiated at time t_1 lying before t_2 and if $prop$ persists through time until time t_3 lying after t_2 or being equal to t_2.

$$\begin{aligned} holds(prop, t_2) \leftarrow\ & initiates(t_1, prop) \\ & t_1 < t_2 \\ & persists(prop, t_1, t_3) \\ & t_2 \leq t_3 \end{aligned} \tag{31}$$

In this framework the abducible predicates are $action$, $<$, $=$ and $persists$. A least commitment strategy requests that objects are not determined until necessary. To avoid unconstrained search through looping during the consistency checking phase, the predicates $=$ and $<$ are treated specially (as in Constraint Logic Programming (cf. [Esh88])).

We will comment on the planning algorithm with a small example coming from the UNIX Mail System. A plan will be provided that one allows to read and then delete the mail from sender "kurt." As domain specific axioms we assume

$$precond(delete(x), \neg first(flag(x)) = 1) \tag{32}$$
$$precond(read(x), \neg first(flag(x)) = 1) \tag{33}$$

Delete and read flag are represented as pair (x, y), where x denotes the delete flag and y the read-flag. (32) and (33) state that a message can only be read or deleted if the delete-flag is not set.

$$postcon(read(x), second(flag(x) = 1)) \tag{34}$$
$$postcon(delete(x), first(flag(x) = 1)) \tag{35}$$

The initial state is described by

$$\begin{aligned} initiates(t, member(mbox(x))) &\leftarrow t = INITIAL, x = mail_1 \\ initiates(t, member(mbox(x))) &\leftarrow t = INITIAL, x = mail_2 \\ &\vdots \\ initiates(t, member(mbox(x))) &\leftarrow t = INITIAL, x = mail_6 \\ initiates(t, sender(x) = \text{“}kurt\text{”}) &\leftarrow t = INITIAL, x = mail_4 \\ initiates(t, sender(x) = \text{“}kurt\text{”}) &\leftarrow t = INITIAL, x = mail_5 \\ initiates(t, flag(x) = (0,0) &\leftarrow t = INITIAL, x = mail_5 \\ initiates(t, flag(x) = (1,1) &\leftarrow t = INITIAL, x = mail_4 \end{aligned}$$

The goal of the plan is specified by

$$\leftarrow holds(flag(mail_4) = (1,1), FINAL), holds(flag(mail_5) = (1,1), FINAL) \tag{36}$$

We will now briefly sketch how the algorithm works with this example.

(1) In the abductive phase, a residue is searched for. (36) is resolved with (31). By the ordering of the events that states $\forall t\ (t < FINAL)$ we get

$$\begin{aligned} \leftarrow\ & initiates(t_3, flag(mail_4) = (1,1)) \wedge persists(flag(mail_4) = (1,1), FINAL) \wedge \\ & initiates(t_3, flag(mail_5) = (1,1)) \wedge persists(flag(mail_5) = (1,1), FINAL) \end{aligned}$$

We apply our knowledge about the actions *read* and *delete*

$$\begin{aligned} \leftarrow\ & action(read(mail_4), t_3) \wedge persists(second(flag(mail_4) = 1), FINAL) \wedge \\ & action(delete(mail_4), t_4) \wedge persists(first(flag(mail_4) = 1), FINAL) \wedge t_3 < t_4 \wedge \\ & action(read(mail_5), t_3) \wedge persists(second(flag(mail_5) = 1), FINAL) \wedge \\ & action(delete(mail_5), t_4) \wedge persists(first(flag(mail_4) = 1), FINAL) \end{aligned}$$

We omit the residue where $t_3 > t_4$ as this will lead to inconsistencies.

(2) This residue is added to the theory in homogenised form.

(3) Our integrity constraints, i.e., the preconditions of our actions, were formulated in (30). We have to test whether

$$\mathcal{T} \cup \Delta \vdash holds(prop, t) \leftarrow action(com, t) \wedge precon(com, prop)$$

holds. This is inconsistent, and we have to reject some assumptions. We remove

$$\begin{aligned} & action(read(mail_4), t_3) \quad \text{and} \\ & action(delete(mail_4, t_4) \end{aligned}$$

from Δ.

(4) We have to find the bindings satisfying

$$\mathcal{T} \cup \Delta \vdash action(com, t) \wedge precon(com, prop)$$

We get the binding $< read(mail_5), t_3, \neg first(flag(mail_5) = 1) >$.

(5) The goal can then be reduced to

$$\begin{aligned} \leftarrow\ & precon(read(mail_5), \neg first(flag(mail_5) = 1)) \wedge \\ & precon(delete(mail_5), \neg first(flag(mail_5) = 1)) \end{aligned}$$

This is already satisfied by $\mathcal{T}$. Thus, the result of the algorithm is

$$\begin{aligned} \Delta\ = \ & action(read(mail_5), t_3), persists(second(flag(mail_5) = 1), FINAL) \wedge \\ & action(delete(mail_5), t_4), persists(first(flag(mail_5) = 1), FINAL) \end{aligned}$$

The result Δ of the abductive algorithm can be quite intuitively interpreted as a plan: the predicate "*action*" specifies the actions to be executed and "*persists*" in connection with $<$ the temporal ordering and persistence of the facts. The algorithm enforces an order only when necessary and as a consequence actions may also be carried out in parallel.

As the author claims, this is a great advantage over situation calculus. Nevertheless, one should not forget to take into account orderings that follow logically from given ones. One further advantage of the approach follows from the persistence assumption. This assumption makes it possible to handle certain kinds of unforeseen events, namely those that do not violate persistence.

This method also allows the circumvention of the frame problem. Eshghi distinguishes two aspects: the *epistemological frame problem* that consists in writing down all frame axioms, and the *computational frame problem* that concerns the repeated application of the frame axioms. Since there are no frame axioms, the first problem is irrelevant. Persistence of propositions through time is modelled by the abducible predicate "*persists*." Thus, the computational frame problem only consists of checking the consistency of persistence assumptions. Eshghi points out that this can be reduced to checking the consistency of clauses of the form $\leftarrow holds(prop, t_2), t_1 < t_2, t_2 \leq t_3$ with timepoints t_1, t_2, t_3. Then, consistency is determined by testing the satisfiability of the negative conjunction and for this problem special purpose algorithms exist.

The above shows that abduction through deduction is a very promising problem solving technique for logic-based planning. In particular, it provides an elegant and intuitive solution to the frame problem, in contrast to most deductive approaches. By handling persistence with the aid of an abducible predicate that is itself part of the plan, the approach can even cope with unforeseen events, as long as they do not violate those assumptions. By the use of a variant of the event calculus in connection with abduction it furthermore becomes possible to specify the ordering of actions only partially.

6.2 Abduction and Relevance

Helft and Konolige present in [HK90] a theory for plan recognition that accounts for both acting and planning. Plan recognition is interpreted as "the recovery of the hidden parts of the plan/act process."

This approach distinguishes between the world knowledge $\mathcal{T}_a$ of the agent and the theory $\mathcal{T}_\omega$ that describes the consequences of actions. To achieve a goal, the agent constructs a plan using solely the knowledge in $\mathcal{T}_a$. The plan P in its simplest form consists of a sequence of basic actions that may be observed as well as their effects. Observed events are denoted by Ω. For a goal g, this can be formally described by:

(1) $\mathcal{T}_a \cup \{P\} \vdash g$ and

(2) $\mathcal{T}_\omega \cup \{P_0\} \vdash \Omega$ with $P_0 \subseteq P$.

The combined planning and plan recognition process gets as input the observations Ω and the theories $\mathcal{T}_a$ and $\mathcal{T}_\omega$. It produces a plan P and a goal g such that (1) and (2) are satisfied. For finding P and g, as well as for explaining the observations, abduction is used. Relevance serves as a constraint mechanism in the sense that it is assumed that an agent only executes actions that help to achieve his goal.

This model has several advantages over deductive approaches. The "dual nature of causality" is recognized: the agent's intentions cause the basic actions of plan P in the sense that he wants to achieve his goals with the plan. The basic actions in turn cause the

observations that can be made. Furthermore, the approach incorporates a "rich model of intention." It is possible to distinguish between intended consequences (goals) and side-effects by considering two processes, namely planning and acting.

Helft and Konolige define a formal model that incorporates the ideas presented above.

Definition 20: ([HK90]) (planning theory)
A *planning theory* is a tuple $(A, G, \mathcal{T}_a, \mathcal{T}_\omega)$, where A is a set of basic actions, G is a set of predefined goals and $\mathcal{T}_a$ and $\mathcal{T}_\omega$ are first-order theories.

Definition 21: ([HK90]) (plan recognition problem)
A *solution to the plan recognition problem* is a tuple (A_0, P, g) such that
(1) A_0 is a minimal subset of A, consistent with $\mathcal{T}_\omega$ with

$$\mathcal{T}_\omega \cup A_0 \vdash \Omega$$

(2) P is a subset of A consistent with $\mathcal{T}_a$, $A_0 \subseteq P$, $g \subseteq G$ with

$$\begin{array}{ll} \text{(a)} & \mathcal{T} \cup \{P\} \vdash g \\ \text{(b)} & \mathcal{T} \cup \{P - e\} \nvdash g, \quad \forall e \in A_0 \end{array}$$

Remarks:

- (1) explains the observations with respect to $\mathcal{T}_\omega$, the "physical" world description.
- In (2) the intentions and beliefs of the agent are considered. With (a) a plan P containing the actions in A_0 is found that achieves the goal g, if the agent's theory $\mathcal{T}_a$ is taken into account. (b) ensures that every action out of A_0 is in fact relevant, i.e., essential for the plan in order to accomplish g.

Definition 22: ([HK90]) (side-effects)
Propositions that are entailed by $\mathcal{T}_\omega \cup P$ but not by $\mathcal{T}_\omega$ alone are called side-effects.

The abductive algorithm that solves the plan recognition problem relies on the function NEW defined by

Definition 23: ([HK90]) (function NEW)

$$NEW_P(\Sigma, \varphi) = (Th(\Sigma \cup \{\varphi\}) - Th(\Sigma)) \cap P,$$

i.e., NEW specifies what a formula φ adds to a theory Σ with respect to a set P, where the intersection with P means that only formulae containing symbols of P should be regarded.

The function is computed by a linear resolution algorithm. A solution (A_0, P, g) is found by several applications of NEW:

Suppose that the knowledge is in conjunctive normal form.

(1) Compute $\Sigma := NEW_B(\mathcal{T}_\omega, \neg\Omega)$, where B contains the symbols denoting the basic actions.

NEW_B computes the relevant consequent actions of $\mathcal{T}_\omega \cup \{\neg\Omega\}$. Thus, for every $\sigma \in \Sigma$, $\neg\sigma := \alpha_1 \wedge \ldots \wedge \alpha_n$ is a candidate for A_0. To see this, consider the definition of NEW. We have

$$\mathcal{T}_\omega \cup \{\neg\Omega\} \vdash \sigma \tag{37}$$

$$\mathcal{T}_\omega \nvdash \sigma \tag{38}$$

By (38), $\neg\sigma$ is consistent with $\mathcal{T}_\omega$. With the aid of the deduction theorem, (37) can be transformed into

$$\begin{aligned} \mathcal{T}_\omega &\vdash \neg\Omega \supset \sigma \\ \mathcal{T}_\omega &\vdash \neg\sigma \supset \Omega \\ \mathcal{T}_\omega \cup \{\neg\sigma\} &\vdash \Omega \end{aligned}$$

Thus, $\neg\sigma$ explains Ω in $\mathcal{T}_\omega$.

(2) Compute $\Gamma := NEW_{B \cup G}(\mathcal{T}_a, \alpha_1 \wedge \ldots \wedge \alpha_n)$. G is the set of instantiated goals. The elements of Γ have the form $\beta_1 \wedge \ldots \wedge \beta_m \supset \gamma$.

Verify as follows that the β_i are possible elements of $P \backslash A_0$ and that γ is a goal candidate:
The conjunctions $\alpha_1 \wedge \ldots \wedge \alpha_n$ are elements of A_0, therefore we abbreviate an arbitrary conjunction by a_0. The definition of NEW yields

$$\mathcal{T}_a \cup \{a_0\} \vdash \beta_1 \wedge \ldots \wedge \beta_m \supset \gamma$$

If the agent considers a_o as action, his theory tells him that a_0 together with the actions $\beta_1 \wedge \ldots \wedge \beta_m$ implies the goal γ. So a plan P to achieve the goal γ consists of the actions a_0 and $\beta_1 \wedge \ldots \wedge \beta_m$:

$$\mathcal{T}_a \cup \{a_0\} \cup \{\beta_1 \wedge \ldots \wedge \beta_m\} \vdash \gamma.$$

As a_0 is also considered as a part of the plan, the requirement $A_0 \subseteq P$ is fulfilled.

(3) Check if $\mathcal{T}_a \wedge \beta_1 \wedge \ldots \wedge \beta_m$ is consistent. If the answer is yes, a solution is found.

(4) Compute the side-effects in G by $NEW_G(\mathcal{T}_\omega, \alpha_1 \wedge \ldots \alpha_n \wedge \beta_1 \wedge \ldots \wedge \beta_m)$ for all elements $\alpha_1 \wedge \ldots \wedge \alpha_n \wedge \beta_1 \wedge \ldots \wedge \beta_m$ in P.

This function call computes exactly the side-effects, as NEW determines those elements of G for which holds

$$\mathcal{T}_\omega \cup p \vdash s \text{ but } \mathcal{T}_\omega \nvdash s,$$

for an arbitrary element p of P. But this is exactly the definition of a side-effect.

Thus, this framework can also account for observations that are merely side-effects of actions. This overcomes a drawback of many current approaches, that is, that they demand a direct causal link between the actions and all observable effects.

This feature also comments on the different demands on diagnosis and plan recognition systems. In diagnosis, all observed symptoms are caused by faults that have to be determined. In plan recognition this need not be the case, as the occurring side-effects prove. An effect need not be caused by the agent's intention.

The separation of the world knowledge and the agent's knowledge is the reason for a further nice property. Plans are constructed with respect to the knowledge $\mathcal{T}_a$ of the agent, thus it is also possible to determine plans that are correct with respect to the user knowledge but ill-formed with respect to the world knowledge.

The two described approaches to abductive planning and plan recognition have shown how the previously presented general methods can be successfully adapted to special applications. Hard problems for deductive approaches, e.g., the frame problem, can be solved in an intuitive and elegant manner if abduction is augmented with additional principles, for instance with default persistence, that are especially suited for problems in which commonsense reasoning is used.

7 Conclusion

In the previous sections an overview of existing abductive reasoning methods has been given, focusing in the last part on applications in plan recognition and planning systems.

Repair or diagnostic problems have shown to be one obvious possibility for applying abductive reasoning. They possess the great advantage that the underlying theory usually has a very simple structure concerning causality. Cause and effect relationships can be represented in a straightforward way, as implications. Furthermore, the search space for hypotheses can often be limited or even completely determined. This is used by the set-cover based model described in section 2.1.

It was shown that such strong assumptions are not suited for more general tasks. The logic based model (see section 2.2) is the most widely spread approach as it allows more flexibility and thus seems to be adequate for a greater range of applications. But, if the generation and selection methods in sections 3 and 4 are regarded in more detail, it becomes obvious that those approaches as well implicitly rely on a special knowledge representation. Causal relationships must be determinable, since they are responsible for choosing the right explanations.

Levesque (cf. section 2.3) goes one step further and defines a model for abduction in dependence of a belief type. He shows that his approach implies the logic based model for implicit belief. Hence, further generality has been gained. The explanation operator is defined semantically and thus ensures the independence of the respective knowledge representation.

Nevertheless, it remains necessary to represent causality on the computational level. So, in order to gain further insight into the foundations of abductive reasoning, the role of causality should be incorporated into the theoretical investigations. In addition, it seems profitable to uncover the relationship between abduction and induction, whose close connection has already been stressed by Peirce.

In section 3, different methods for hypotheses generation were described, in particular resolution based algorithms and an ATMS procedure.

Formal Model	Article	Context of Abduction	Generation Method described or used	Selection Method described or used	Comment
set-cover-based	[ATBJ87]	best explanation for observations	predefined function	plausibility, parsimony	restricted applicability: observation - explanation relationship is predefined
logic-based	[Kon90]	diagnosis		subset-minimality	transformation to the consistency-based method of Reiter (s. [Rei87])
	[Esh88] [EK88]	generation of conditional answers to queries; application to planning	linear resolution	integrity constraints	abduction as generalization of logic programming
	[CP84]	find explanations	linear resolution and reverse skolemization	basicness; minimality	algorithm is incomplete w.r.t. basicness and minimality
	[Pop73]	find explanations	linear resolution	factoring across partial proof trees	suggestion: combination of abduction, induction, and deduction
	[Sha89]	temporal reasoning system: abduction and default persistence	linear resolution	explanation postulating fewest events	based on Event Calculus (s. [KS86])
	[AP90]	recognition of plans and goals	theorem prover + control	weighting factors; lowest cost explanation	how are weighting factors determined? models only most-specific abduction
	[Poo88]	defaults as predefined hypotheses	PROLOG	allows different methods, e.g., definition of priorities	general tool for reasoning with defaults
	[HK90]	determination of plans and corresponding goals	linear resolution		distinguishes between side-effects and intended consequences
	[NM90]	deep causal explanations for natural language text	linear resolution	coherence	heuristic search (beam search) to improve efficiency
knowledge-level	[Lev89]	find explanations	function yielding disjunction of feasible explanations; ATMS for implicit belief	explanations with fewest propositional symbols	most general definition of abduction

Table 1: Survey of presented approaches

Article	Implementation
[ATBJ87]	RED: antibody identification in the domain of red blood cell typing (see also [Smi85])
[Pop73]	abductive procedure implemented in GOL (see also [Pop72])
[Sha89]	prototype in PROLOG
[AP90]	preliminary implemetation using Prolog Technology Prover (PTTP) (see [Sti88])
[NM90]	algorithm using metric in ACCEL (Abductive Calculation of Causal Explanations for Language)
[Poo88]	THEORIST: framework for default reasoning
[Esh88]	planning system ABPLAN

Table 2: Implemented approaches

In the actual realizations of abductive procedures, one suffers from the problem that in most cases one abductive explanation has to be singled out. As a consequence, selection criteria have to be defined that allow one to find, in some sense, the "best" hypothesis. Apart from syntactic criteria and different strategies to determine an appropriate level of specificity, a method based on a coherence metric was described (see section 4). The metric was shown to be a first step in the direction proposed by Peirce, who demands the selection of the psychologically simplest explanation, not the logically simplest. Nevertheless, in most cases a combination of some of the described approaches seems to be adequate. A unifying theory is a further research topic.

A comparison of default and abductive reasoning in section 5 has on the one hand clarified the close connection of both approaches by showing that defaults can be interpreted as a kind of predefined hypotheses. Furthermore, it became possible to derive some results concerning the complexity of abductive and default reasoning. Both approaches were shown to be NP-hard in the general case.

The first steps for wider applications of abduction have already been made. If the field of planning is considered, it becomes obvious that the general models for abduction are not sufficient. They have to be augmented with features that allow the handling of intentions, persistence, time, etc. The applications presented in section 6 show how the formal models have to be changed for special applications. In particular, Eshghi has shown that in the context of logic programming, integrity constraints are a further method for detecting appropriate explanations. A survey of presented approaches is given in table 2 (see page 42). Implementations are listed in table 2.

Thus, the presented ideas are valuable as a first sketch of the possibilities of abductive reasoning. It should be profitable to investigate abduction in the more general framework of theory formation and to develop a model that is able to cover applications that involve commonsense reasoning and theory formation with commonsense. This requires, e.g., the treatment of time, causality, self-reflection. Furthermore, it seems promising to compare in more detail the connection between abduction, induction, and deduction.

Acknowledgements

I would like to thank Susanne Biundo, Jana Köhler, Bernhard Nebel, Wolfgang Wahlster, and especially Mathias Bauer for helpful comments on earlier versions of this paper.

References

[AP90] D.E. Appelt and M. Pollack. Weighted abduction for plan ascription. Technical report, Artificial Intelligence Center and Center for the Study of Language and Information, SRI International, Menlo Park, California, 1990.

[ATBJ87] D. Allemang, M. Tanner, T. Bylander, and J.Josephson. Computational complexity of hypothesis assembly. In *Proceedings of the 10th International Joint Conference on Artificial Intelligence*, pages 1112–1117, 1987.

[CM85] E. Charniak and P. McDermott. *Introduction to Artificial Intelligence.* Addison Wesley, Menlo Park, California, 1985.

[CP86] P. Cox and T. Pietrzykowski. Causes for events: Their computation and application. In *Proceedings CADE 86*, pages 608–621, 1986.

[CS90] E. Charniak and S.E. Shimony. Probabilistic semantics for cost based abduction. In *Proceedings of the 8th National Conference on Artificial Intelligence*, pages 106–111, 1990.

[EK88a] K. Eshghi and R. Kowalski. Abduction compared with negation by failure. In *Proceedings of the 6th International Conference on Logic Programming*, 1988.

[EK88b] K. Eshghi and R. Kowalski. Abduction through deduction. Technical report, Imperial College of Science and Technology, Department of Computing, 1988.

[Esh88] K. Eshghi. Abductive planning with event calculus. In *Proceedings of the 5th International Conference on Logic Programming*, page 562, 1988.

[Gou50] Th.A. Goudge. *The Thought of C.S. Peirce.* Dover Publications Inc., New York, 1950.

[HK90] N. Helft and K. Konolige. Plan recognition as abduction and relevance. Draft version, Artificial Intelligence Center, SRI International, Menlo Park, California, 1990.

[HSME89] J. R. Hobbs, M. Stickel, P. Martin, and D. Edwards. Interpretation as abduction. Draft version, SRI International, Artificial Intelligence Center, Menlo Park, CA, 1989.

[Kon90] K. Konolige. Closure + minimization implies abduction. In *PRICAI-90, Nagoya, Japan*, 1990.

[KS86] R. Kowalski and M. Sergot. A logic based calculus of events. *New Generation Computing*, 4:67–95, 1986.

[Lev89] H. Levesque. A knowledge-level account of abduction. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, pages 1061–1067, 1989.

[NM90] Hwee Tou Ng and R.J. Mooney. On the role of coherence in abductive explanation. In *Proceedings of the 8th National Conference on Artificial Intelligence*, pages 337–342, 1990.

[Pei58] C.S. Peirce. *Collected Papers of Charles Sanders Peirce (eds. C. Hartshorne et al.)*. Harvard University Press, 1931-1958.

[Poo88] D. Poole. A logical framework for default reasoning. *Artificial Intelligence*, 36:27–47, 1988.

[Poo89] D. Poole. Normality and faults in logic-based diagnosis. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, pages 1304–1310, 1989.

[Pop72] H.E. Pople, Jr. A goal oriented language for the computer. In *Representation and meaning - Experiments with information processing systems.* Prentice Hall (eds. H. Simon and L. Siklossy), 1972.

[Pop73] H.E. Pople, Jr. On the mechanization of abductive logic. In *Proceedings of the 3rd International Joint Conference on Artificial Intelligence*, pages 147–151, 1973.

[Reg88] J. Reggia. Diagnostic expert systems based on a set covering model. *International Journal of Man-Machine Studies*, November 83:437–460, 1988.

[Rei80] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(2):81–132, 1980.

[Rei87] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32:57–95, 1987.

[Sha89] M. Shanahan. Prediction is deduction but explanation is abduction. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, pages 1055–1060, 1989.

[SL89] B. Selman and H.L. Levesque. Abductive and default reasoning: A computational core. In *Proceedings of the 8th National Conference on Artificial Intelligence*, pages 343–348, 1989.

[Smi85] J.W. Smith. *RED: A Classificatory and Abductive Expert System.* PhD thesis, Ohio State University, Laboratory for Artificial Intelligence Research, Department for Computer Science, 1985.

[Sti88] M.E. Stickel. A prolog technology theorem prover: implementation by an extended prolog compiler. *Journal of Automated Reasoning*, 4:353–380, 1988.

[Sti90] M. Stickel. Rationale and methods for abductive reasoning in natural-language interpretation. In R. Studer, editor, *Natural Language and Logic*, pages 331–352. Springer-Verlag, Berlin, Heidelberg, New York, 1990. (forthcoming).

[Wey80] R. Weyhrauch. Prolegomena to a theory of mechanized formal reasoning. *Artificial Intelligence*, 13:133–170, 1980.

Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

DFKI
-Bibliothek-
PF 2080
D-6750 Kaiserslautern
FRG

DFKI Publikationen

Die folgenden DFKI Veröffentlichungen sowie die aktuelle Liste von allen bisher erschienenen Publikationen können von der oben angegebenen Adresse bezogen werden.
Die Berichte werden, wenn nicht anders gekennzeichnet, kostenlos abgegeben.

DFKI Publications

The following DFKI publications or the list of all publisched papers so far can be ordered from the above address.
The reports are distributed free of charge except if otherwise indicated.

DFKI Research Reports

RR-90-15
Harald Trost: The Application of Two-level Morphology to Non-concatenative German Morphology
13 pages

RR-90-16
Franz Baader, Werner Nutt: Adding Homomorphisms to Commutative/Monoidal Theories, or: How Algebra Can Help in Equational Unification
25 pages

RR-90-17
Stephan Busemann:
Generalisierte Phasenstrukturgrammatiken und ihre Verwendung zur maschinellen Sprachverarbeitung
114 Seiten

RR-91-01
Franz Baader, Hans-Jürgen Bürckert, Bernhard Nebel, Werner Nutt, Gert Smolka: On the Expressivity of Feature Logics with Negation, Functional Uncertainty, and Sort Equations
20 pages

RR-91-02
Francesco Donini, Bernhard Hollunder, Maurizio Lenzerini, Alberto Marchetti Spaccamela, Daniele Nardi, Werner Nutt: The Complexity of Existential Quantification in Concept Languages
22 pages

RR-91-03
B.Hollunder, Franz Baader: Qualifying Number Restrictions in Concept Languages
34 pages

RR-91-04
Harald Trost: X2MORF: A Morphological Component Based on Augmented Two-Level Morphology
19 pages

RR-91-05
Wolfgang Wahlster, Elisabeth André, Winfried Graf, Thomas Rist: Designing Illustrated Texts: How Language Production is Influenced by Graphics Generation.
17 pages

RR-91-06
Elisabeth André, Thomas Rist: Synthesizing Illustrated Documents: A Plan-Based Approach
11 pages

RR-91-07
Günter Neumann, Wolfgang Finkler: A Head-Driven Approach to Incremental and Parallel Generation of Syntactic Structures
13 pages

RR-91-08
Wolfgang Wahlster, Elisabeth André, Som Bandyopadhyay, Winfried Graf, Thomas Rist: WIP: The Coordinated Generation of Multimodal Presentations from a Common Representation
23 pages

RR-91-09
Hans-Jürgen Bürckert, Jürgen Müller, Achim Schupeta: RATMAN and its Relation to Other Multi-Agent Testbeds
31 pages

RR-91-10
Franz Baader, Philipp Hanschke: A Scheme for Integrating Concrete Domains into Concept Languages
31 pages

RR-91-11
Bernhard Nebel: Belief Revision and Default Reasoning: Syntax-Based Approaches
37 pages

RR-91-12
J.Mark Gawron, John Nerbonne, Stanley Peters:
The Absorption Principle and E-Type Anaphora
33 pages

RR-91-13
Gert Smolka: Residuation and Guarded Rules for Constraint Logic Programming
17 pages

RR-91-14
Peter Breuer, Jürgen Müller: A Two Level Representation for Spatial Relations, Part I
27 pages

RR-91-15
Bernhard Nebel, Gert Smolka:
Attributive Description Formalisms ... and the Rest of the World
20 pages

RR-91-16
Stephan Busemann: Using Pattern-Action Rules for the Generation of GPSG Structures from Separate Semantic Representations
18 pages

RR-91-17
Andreas Dengel, Nelson M. Mattos:
The Use of Abstraction Concepts for Representing and Structuring Documents
17 pages

RR-91-18
John Nerbonne, Klaus Netter, Abdel Kader Diagne, Ludwig Dickmann, Judith Klein:
A Diagnostic Tool for German Syntax
20 pages

RR-91-19
Munindar P. Singh: On the Commitments and Precommitments of Limited Agents
15 pages

RR-91-20
Christoph Klauck, Ansgar Bernardi, Ralf Legleitner
FEAT-Rep: Representing Features in CAD/CAM
48 pages

RR-91-21
Klaus Netter: Clause Union and Verb Raising Phenomena in German
38 pages

RR-91-22
Andreas Dengel: Self-Adapting Structuring and Representation of Space
27 pages

RR-91-23
Michael Richter, Ansgar Bernardi, Christoph Klauck, Ralf Legleitner: Akquisition und Repräsentation von technischem Wissen für Planungsaufgaben im Bereich der Fertigungstechnik
24 Seiten

RR-91-24
Jochen Heinsohn: A Hybrid Approach for Modeling Uncertainty in Terminological Logics
22 pages

RR-91-25
Karin Harbusch, Wolfgang Finkler, Anne Schauder:
Incremental Syntax Generation with Tree Adjoining Grammars
16 pages

RR-91-26
M. Bauer, S. Biundo, D. Dengler, M. Hecking, J. Koehler, G. Merziger:
Integrated Plan Generation and Recognition
- A Logic-Based Approach -
17 pages

RR-91-27
A. Bernardi, H. Boley, Ph. Hanschke, K. Hinkelmann, Ch. Klauck, O. Kühn, R. Legleitner, M. Meyer, M. M. Richter, F. Schmalhofer, G. Schmidt, W. Sommer:
ARC-TEC: Acquisition, Representation and Compilation of Technical Knowledge
18 pages

RR-91-28
Rolf Backofen, Harald Trost, Hans Uszkoreit:
Linking Typed Feature Formalisms and Terminological Knowledge Representation Languages in Natural Language Front-Ends
11 pages

RR-91-29
Hans Uszkoreit: Strategies for Adding Control Information to Declarative Grammars
17 pages

RR-91-30
Dan Flickinger, John Nerbonne:
Inheritance and Complementation: A Case Study of *Easy* Adjectives and Related Nouns
39 pages

RR-91-31
H.-U. Krieger, J. Nerbonne:
Feature-Based Inheritance Networks for Computational Lexicons
11 pages

RR-91-32
Rolf Backofen, Lutz Euler, Günther Görz:
Towards the Integration of Functions, Relations and Types in an AI Programming Language
14 pages

RR-91-33
Franz Baader, Klaus Schulz:
Unification in the Union of Disjoint Equational Theories: Combining Decision Procedures
33 pages

RR-91-34
Bernhard Nebel, Christer Bäckström:
On the Computational Complexity of Temporal Projection and some related Problems
35 pages

RR-91-35
Winfried Graf, Wolfgang Maaß: Constraint-basierte Verarbeitung graphischen Wissens
14 Seiten

RR-92-03
Harold Boley:
Extended Logic-plus-Functional Programming
28 pages

RR-92-04
John Nerbonne: Feature-Based Lexicons: An Example and a Comparison to DATR
15 pages

RR-92-05
Ansgar Bernardi, Christoph Klauck, Ralf Legleitner, Michael Schulte, Rainer Stark:
Feature based Integration of CAD and CAPP
19 pages

RR-92-08
Gabriele Merziger: Approaches to Abductive Reasoning - An Overview -
46 pages

DFKI Technical Memos

TM-91-01
Jana Köhler: Approaches to the Reuse of Plan Schemata in Planning Formalisms
52 pages

TM-91-02
Knut Hinkelmann: Bidirectional Reasoning of Horn Clause Programs: Transformation and Compilation
20 pages

TM-91-03
Otto Kühn, Marc Linster, Gabriele Schmidt:
Clamping, COKAM, KADS, and OMOS: The Construction and Operationalization of a KADS Conceptual Model
20 pages

TM-91-04
Harold Boley (Ed.):
A sampler of Relational/Functional Definitions
12 pages

TM-91-05
Jay C. Weber, Andreas Dengel , Rainer Bleisinger:
Theoretical Consideration of Goal Recognition Aspects for Understanding Information in Business Letters
10 pages

TM-91-06
Johannes Stein: Aspects of Cooperating Agents
22 pages

TM-91-08
Munindar P. Singh: Social and Psychological Commitments in Multiagent Systems
11 pages

TM-91-09
Munindar P. Singh: On the Semantics of Protocols Among Distributed Intelligent Agents
18 pages

TM-91-10
Béla Buschauer, Peter Poller, Anne Schauder, Karin Harbusch: Tree Adjoining Grammars mit Unifikation
149 pages

TM-91-11
Peter Wazinski: Generating Spatial Descriptions for Cross-modal References
21 pages

TM-91-12
Klaus Becker, Christoph Klauck, Johannes Schwagereit: FEAT-PATR: Eine Erweiterung des D-PATR zur Feature-Erkennung in CAD/CAM
33 Seiten

TM-91-13
Knut Hinkelmann:
Forward Logic Evaluation: Developing a Compiler from a Partially Evaluated Meta Interpreter
16 pages

TM-91-14
Rainer Bleisinger, Rainer Hoch, Andreas Dengel:
ODA-based modeling for document analysis
14 pages

TM-91-15
Stefan Bussmann: Prototypical Concept Formation An Alternative Approach to Knowledge Representation
28 pages

TM-92-01
Lijuan Zhang:
Entwurf und Implementierung eines Compilers zur Transformation von Werkstückrepräsentationen
34 Seiten

DFKI Documents

D-91-01
Werner Stein , Michael Sintek: Relfun/X - An Experimental Prolog Implementation of Relfun
48 pages

D-91-02
Jörg P. Müller: Design and Implementation of a Finite Domain Constraint Logic Programming System based on PROLOG with Coroutining
127 pages

D-91-03
Harold Boley, Klaus Elsbernd, Hans-Günther Hein, Thomas Krause: RFM Manual: Compiling RELFUN into the Relational/Functional Machine
43 pages

D-91-04
DFKI Wissenschaftlich-Technischer Jahresbericht 1990
93 Seiten

D-91-06
Gerd Kamp: Entwurf, vergleichende Beschreibung und Integration eines Arbeitsplanerstellungssystems für Drehteile
130 Seiten

D-91-07
Ansgar Bernardi, Christoph Klauck, Ralf Legleitner
TEC-REP: Repräsentation von Geometrie- und Technologieinformationen
70 Seiten

D-91-08
Thomas Krause: Globale Datenflußanalyse und horizontale Compilation der relational-funktionalen Sprache RELFUN
137 Seiten

D-91-09
David Powers, Lary Reeker (Eds.):
Proceedings MLNLO'91 - Machine Learning of Natural Language and Ontology
211 pages
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

D-91-10
Donald R. Steiner, Jürgen Müller (Eds.):
MAAMAW'91: Pre-Proceedings of the 3rd European Workshop on „Modeling Autonomous Agents and Multi-Agent Worlds"
246 pages
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

D-91-11
*Thilo C. Horstmann:*Distributed Truth Maintenance
61 pages

D-91-12
Bernd Bachmann:
$H^{iera}C_{on}$ - a Knowledge Representation System with Typed Hierarchies and Constraints
75 pages

D-91-13
International Workshop on Terminological Logics
Organizers: Bernhard Nebel, Christof Peltason, Kai von Luck
131 pages

D-91-14
Erich Achilles, Bernhard Hollunder, Armin Laux, Jörg-Peter Mohren: KRIS : Knowledge Representation and Inference System
- Benutzerhandbuch -
28 Seiten

D-91-15
Harold Boley, Philipp Hanschke, Martin Harm, Knut Hinkelmann, Thomas Labisch, Manfred Meyer, Jörg Müller, Thomas Oltzen, Michael Sintek, Werner Stein, Frank Steinle:
µCAD2NC: A Declarative Lathe-Worplanning Model Transforming CAD-like Geometries into Abstract NC Programs
100 pages

D-91-16
Jörg Thoben, Franz Schmalhofer, Thomas Reinartz:
Wiederholungs-, Varianten- und Neuplanung bei der Fertigung rotationssymmetrischer Drehteile
134 Seiten

D-91-17
Andreas Becker:
Analyse der Planungsverfahren der KI im Hinblick auf ihre Eignung für die Abeitsplanung
86 Seiten

D-91-18
Thomas Reinartz: Definition von Problemklassen im Maschinenbau als eine Begriffsbildungsaufgabe
107 Seiten

D-91-19
Peter Wazinski: Objektlokalisation in graphischen Darstellungen
110 Seiten

Approaches to Abductive Reasoning
- An Overview -

Gabriele Merziger

RR-92-08
Research Report