has two explanations, *MakePastaDish* and *MakeMeat&SauceDish*. There is actually no need for discriminating among these two explanations; the abstraction causes a loss of information.

Due to problems 1 and 2, the generation of refinements in the forward chaining refinement approach has to be much less constrained than one would expect (even if it is not necessarily required to consider all non-minimal abstract explanations, and to refine each assumption with an unlimited number of its specializations [6]).

Even in cases where the complications due to problems 1 and 2 do not arise, the convenience of the two top-down approaches with respect to the flat approach can be questioned. Problem 3 is an extreme case; in general, detailed observations and abstract explanations provide two groups of constraints on low-level assumptions; the interaction of these two groups of constraints provides the desired solutions, but the best way for constraining search depends on which group of constraints is stronger.

Top-down abduction with forward-chaining refinement may be convenient with respect to the 'flat' approach if the constraints given by abstract explanations are stronger.

The advantage of top-down abduction with backward chaining refinement with respect to the 'flat' approach can be due to the fact that, once a set of assumptions explaining positive observations has been found, forward chaining is applied to it, in order to check consistency with respect to negative observations, as well as to find other observable consequences that can be used for discrimination. Then a single forward chaining at the higher level can save forward chaining from several low-level explanations; for this reason, the running time of this top-down approach increases as the number of explanations of positive low-level observations increases, but slower than the flat approach. When such a number is sufficiently large to balance the overhead due to the more complex architecture, this top-down approach becomes convenient (more details and some experimental results are given in [6]).

# References


[1] L. Console and D. Theseider Dupré. Abductive reasoning with abstraction axioms. In *Foundations of Knowledge Representation and Reasoning*. Lecture Notes in Computer Science, Springer Verlag (to appear).

[2] F. Giunchiglia and T. Walsh. A theory of abstraction. *Artificial Intelligence*, 57:323–389, 1992.

[3] H. Kautz. A formal theory of plan recognition and its implementation. In J. Allen, H. Kautz, R. Pelavin and J. Tenenberg, *Reasoning about Plans*, Morgan Kaufmann, 1991.

[4] I. Mozetic. Hierarchical model-based diagnosis. *Int. J. of Man-Machine Studies*, 35(3):329–362, 1991.

[5] D. Poole. Explanation and prediction: An architecture for default and abductive reasoning. *Computational Intelligence*, 5(2):97–110, 1989.

[6] D. Theseider Dupré. *Characterizing and Mechanizing Abductive Reasoning*. PhD Thesis, Dip. Informatica, Università di Torino, 1994.

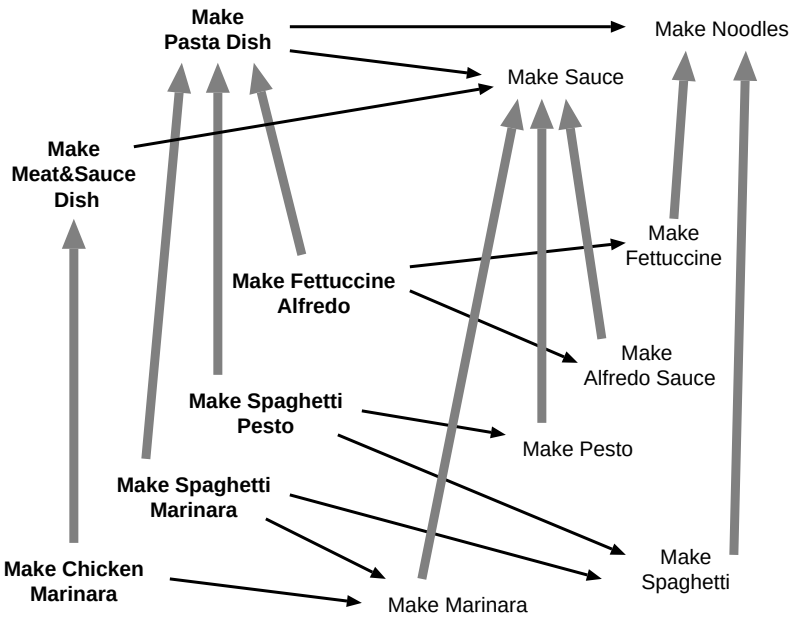


Figure 2: Another explanatory hierarchy in the cooking world.

explanations and testing such refinements (by forward chaining), in order to check whether they actually explain low-level observations (**forward chaining refinement**). This is similar to the use of abstractions in model-based diagnosis proposed in [4], which explicitly assumes to deal with models where there is too much non-determinism in backward reasoning.

In order to adopt one of these two approaches rather than the "flat" one which directly explains low-level observations, a number of subtleties have to be taken into account.

**Problem 1: refining minimal abstract explanations does not suffice.** More precisely, search cannot be limited to explanations that can be obtained from *minimal* high-level explanations substituting every assumption $P$ with a $Q$ such that $Q \rightarrow P \in T_A$. Consider in fact figure 2 and suppose that $OBS = \{MakeMarinara, MakeSpaghetti\}$. The abstract observations are $OBS' = \{MakeSauce, MakeNoodles\}$, with minimal explanation $E_1 = \{MakePastaDish\}$. However, the minimal explanations of $OBS$ are $E_2 = \{MakeSpaghettiMarinara\}$ and $E_3 = \{MakeChickenMarinara, MakeSpaghettiPesto\}$; the latter is not a refinement of $E_1$ in the sense defined above.

**Problem 2: singleton refinements do not suffice**, i.e. it is not sufficient to limit to refining an abstract assumption $P$ into *one* $Q$ such that $Q \rightarrow P \in T_A$. Consider again figure 2 and suppose that $OBS = \{MakeFettuccine, MakeSpaghetti\}$. The corresponding abstract set of observations is $OBS' = \{MakeNoodles\}$ with explanation $E = \{MakePastaDish\}$; however, there is no singleton refinement of $E$ which explains $OBS$: two specific pasta dishes have to be assumed.

**Problem 3: abstraction may unfocus.** This problem is even more fundamental than problems 1 and 2. Suppose that in figure 2 $MakePesto$ is observed. There is only one explanation at the detailed level, $MakeSpaghettiPesto$; at the abstract level, $MakeSauce$

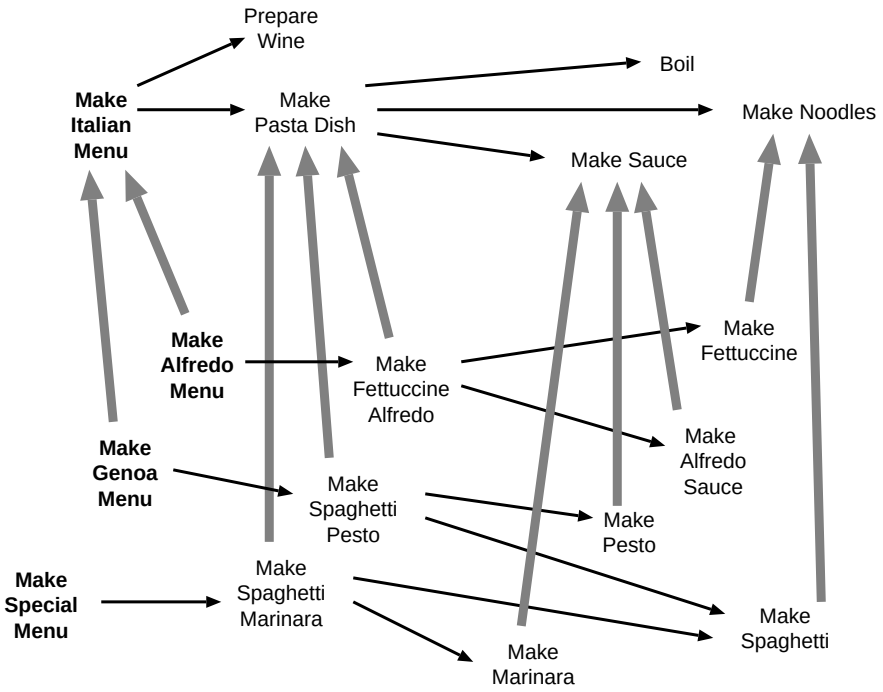


Figure 1: A cooking hierarchy.

An alternative is to identify (in a preprocessing phase) the incomplete abstractions; when explaining observations at some level, only those parts of lower levels that have no correspondent in the current level should be taken into account. In the example this means identifying in a preprocessing phase that $MakeSpecialMenu$ is an explanation of $MakePastaDish$ that is not represented in $MakeItalianMenu$ (the only explanation of $MakePastaDish$ that uses explanatory axioms only). There are several ways to implement this form of preprocessing; details are given in [6].

# 4 Top-down abduction

A second main goal, which is relevant if detailed observations are available from the beginning, is the *use of abstractions to focus the explanation process*. This a purely computational goal: speeding up the computation of explanations satisfying definition 2. The idea is to explain an abstraction of the detailed observations, possibly discriminating among different abstract explanations, and to use the high-level explanations in order to (hopefully) focus the explanation of detailed observations.

Notice that there are essentially two ways of exploiting the fact that high-level explanations have already been found:

- backward chaining from detailed observations and accepting only those solutions that are refinements (under some appropriate definition) of high-level explanations (**backward chaining refinement**);
- generating all the refinements (according to an appropriate definition) of high-level

**Definition 2** *Given an explanatory hierarchy* $\langle T_E, T_A \rangle$, *an* **explanation** *of a set of literals* $OBS$ *is a set* $E$ *of abducible symbols such that* $T_E \cup T_A \cup E \vdash O$ *for each positive literal* $O$ *in* $OBS$ *and* $T_E \cup T_A \cup E \not\vdash O$ *for each negative literal* $\neg O$ *in* $OBS$.

# 3 Avoiding unnecessarily detailed explanations

A first main goal in defining abduction in the presence of abstractions is the introduction of *preference criteria based on abstraction.* In particular, explanations that are as abstract as possible could be preferred, according to a form of parsimony that generalizes the preference for minimal explanations: unnecessarily detailed assumptions are avoided, in the same way as unnecessary assumptions.

A good declarative characterization for this preference is the notion of least presumptive explanation introduced in [5]. However, there is an important computational counterpart: avoiding the computation of non-preferred explanations would be much better than discarding them a-posteriori; this is especially relevant in explanatory hierarchies where there is significant redundancy (due to the expression of the same knowledge at different levels).

This task is relevant if the observations can be explained by "abstract" assumptions that can be further refined, but the refinement is not necessarily required; in other words, if the goal of the user, at least in a first step, is *just* to have explanations of the observations, no more and no less. In this task abstraction is relevant at least for accommodating natural and economic representations where inheritance is used to avoid repeating common consequences of a class of hypotheses, and for providing flexibility, in the sense of being able to deal (at different times or at the same time) with observations and assumptions at different levels of abstraction.

Figure 1 is an explanatory hierarchy adapted from [3] (the task consists in recognizing the dish - or menu - an agent is preparing by observing his/her actions). Thick grey arrows correspond to abstraction axioms while thin black arrows correspond to explanatory axioms; abducible symbols are in boldface.

Given the observation $MakeNoodles$, the explanation $E_1 = \{MakeItalianMenu\}$ should be preferred to $E_2 = \{MakeGenoaMenu\}$ and to $E_3 = \{MakeAlfredoMenu\}$ since they are unnecessarily specific. This can be formalized as in the definition of least presumptive explanation introduced in [5]: since each of $E_2$ and $E_3$ implies $E_1$, given $T_A$, they are considered "more presumptive" than $E_1$, which is a "least presumptive" explanation.

Notice however that $E_4 = \{MakeSpecialMenu\}$, is not more presumptive than other explanations (it does not imply other explanations). In this case the abstraction is *incomplete* since there is no counterpart, at the abstract level, for knowledge about $MakeSpecialMenu$; therefore, not all of the least presumptive explanations of $MakeNoodles$ can be found using just explanatory axioms.

The problem is how to constrain the search for explanations in such a way that explanations such as $E_2$ and $E_3$ are not found, rather than found and discarded; however, explanations such as $E_4$, if any, are not disregarded.

There are several possible answers to this problem. One is to require abstractions to be complete, i.e. disallowing hierarchies like the given one. This would require duplicating at abstract levels knowledge that cannot be significantly abstracted.

# The Different Roles of Abstraction in Abductive Reasoning

Daniele Theseider Dupré  Mauro Rossotto

Dipartimento di Informatica — Università di Torino
Corso Svizzera 185 – 10149 Torino, Italy
E-mail: dtd@di.unito.it

## 1 Introduction

There are several different ways of exploiting abstractions in abduction, both from a declarative and a computational point of view, depending on the scenario in which the abductive system is expected to operate; in particular, depending on the availability of observations and on the ultimate goal of the user of the abductive system. This abstract briefly analyzes two of such ways: the definition and implementation of preference criteria for explanations, and the use of abstractions to focus the explanation process.

## 2 Explanatory hierarchies

The following is a simple propositional language (derived from [3] and already introduced in [1]) that is sufficient for discussing several issues about the use of abstractions in abduction, even if it does not necessarily accommodate for all of the ways abstraction can be used or has been used in AI [2, 4].

**Definition 1** *An* **explanatory hierarchy** *is a pair* $\langle T_E, T_A \rangle$ *where:*

- $T_E$*, the set of* **explanatory axioms***, is a set of definite clauses* $P_1 \wedge ... \wedge P_n \rightarrow Q$ *where* $P_1 \wedge ... \wedge P_n$ *is intended as a direct explanation of* $Q$ *(e.g. a cause of* $Q$*).*
- $T_A$*, the set of* **abstraction axioms***, is a set of definite clauses* $P \rightarrow Q$ *where* $P$ *is intended as an entity which is more specific than* $Q$ *(*$Q$ *is an abstraction of* $P$*).*
- *There is a distinguished set of symbols, the* **abducible symbols** *or* **assumptions***, that can be accepted in explanations and are the ones that do not appear in the consequent of any explanatory axiom.*
- *There are no cyclic dependencies in* $T_E \cup T_A$*.*

The following definition of abductive explanation (a variation on definitions used in the literature) can be adopted if the two dimensions of explanatory hierarchies are disregarded, i.e. if the set $T_E \cup T_A$ is considered in the same way as a "flat" domain theory. The same definition can be used as a reference for analyzing the role of abstractions in abduction.