Exploiting abstractions in cost-sensitive abductive problem solving with observations and actions

Exploiting abstractions in cost-sensitive abductive problem solving with observations and actions

Gianluca Torta [a,*], Luca Anselma [a] and Daniele Theseider Dupré [b]
[a] *Dipartimento di Informatica, Università di Torino*
Corso Svizzera 185, 10149 Torino (Italy)
E-mail: {torta,anselma}@di.unito.it
[b] *DiSIT, Università del Piemonte Orientale*
Viale Teresa Michel 11, 15121 Alessandria (Italy)
E-mail: dtd@di.unipmn.it

Several explanation and interpretation tasks, such as diagnosis, plan recognition and image interpretation, can be formalized as abductive and consistency reasoning. The interpretation task is usually executed for the purpose of performing actions, e.g., in diagnosis, repair actions or therapy. Some proposals address the problem based on a task-independent representation of a domain which includes an ontology or taxonomy of hypotheses and observations. In this paper we rely on the same type of representation, and we point out the role of abstractions in an iterative abduction process. At each iteration, as in model-based diagnosis and troubleshooting, our algorithm chooses to perform further observations or actions taking into account their costs and the likelihood of candidate hypotheses. The main goal of the algorithm is to ensure discrimination among hypotheses and, more importantly, to perform the appropriate actions for the case at hand. We discuss an implementation of the proposed method and report experimental results that support the conclusion that abstractions are indeed useful for the considered task.

Keywords: Abduction, Abstraction, Actions, Costs

1. Introduction

Several explanation and interpretation tasks, such as diagnosis, plan recognition and image interpretation, can be formalized as abductive reasoning or related forms of nonmonotonic reasoning. A number of approaches [4,14,7,17,2] address the problem based on a representation of a domain which includes an ontology or taxonomy of hypotheses.

However, explanation or interpretation is usually an intermediate step to a final goal, which is performing actions, such as repair or therapy in diagnosis, or reacting to the recognized plan, in plan recognition. In some cases, such actions are also needed, or, at least, useful, for discriminating among alternative explanations *during* the explanation/interpretation process itself (e.g., trying a repair action would either solve the problem or at least provide the information that the corresponding hypothesis is not the correct one).

Ontologies have been proposed as the basis for large knowledge bases to be used also for other problem solving tasks (including planning, see [21, 11]), but, as noted in [6], they should be shared among different problem solvers for related tasks; therefore, they should be developed independently of the reasoning task[1]: i.e., their structure should reflect a natural representation of the domain, but it might not directly provide the best structure for diagnosis, interpretation, or planning and acting.

In this paper we propose a novel approach where a similar representation is adopted in the context of an iterative abduction process where:

- further observations or actions (e.g. substituting a suspect component in the system), as in model-based diagnosis [12] and *troubleshooting* [13], can be proposed with the intermediate goal of discriminating among candidate explanations and the ultimate goal of per-

*Corresponding author: Gianluca Torta, Corso Svizzera 185, 10149 Torino (Italy).

[1] In perspective, a shared ontology for different reasoning tasks may be available on the Web.

AI Communications
ISSN 0921-7126, IOS Press.

forming actions that are appropriate for the case at hand. Actions may be interleaved with observations [10].
- The costs of observations are balanced with reduced costs of the actions performed for solving the problem.

The costs associated with the results of abduction, in a diagnostic setting, correspond to the cost of repair actions or therapy, and are expected to decrease as long as more information is available on the hypotheses; similarly, in a plan recognition or in an interpretation task, the human or software agent using the results should achieve an advantage from a better discrimination of hypotheses or from more specific hypotheses, leading to a more focused action, possibly with reduced costs — e.g., if hypotheses are threats to the agent with costly defense actions. In all settings, we intend that some actions have to be taken based, in general, on the remaining candidate hypotheses. If the set of candidates is too broad or too abstract, the agent is expected to incur into higher action costs due to (a combination of) the following reasons:

- an action which is stronger than necessary is taken, in order to account for all current possibilities;
- unnecessary actions are taken, e.g., repairing the wrong part, taking the wrong therapy, defending from the wrong threat.

The different issues are related: discrimination may be performed among hypotheses at the same level of abstraction, but it could also involve refining hypotheses. In any case, discrimination requires more observations, whose cost should be balanced with the benefits, in terms of more suitable actions, of better discrimination.

The presence of a domain representation with IS-A abstractions has a significant impact on this trade-off. The cost of observing the same phenomenon at different levels of abstraction is expected to vary significantly; in fact, it may range from subjective information from a human (patient or user) to more or less costly medical or technical tests, or, in an image interpretation task, it may involve computationally complex image processing, to be performed interactively with the reasoning task, as suggested in [15]. Note that, in any case, the presence of abstractions should not prevent in general the ability to exploit detailed observations and knowledge when convenient [24].

In several settings, an observation which is itself expensive, because it consumes resources and time to be performed, implies additional costs due to the delay before taking an action: breakdown costs in diagnosing a physical system, risk of death of the patient in medical diagnosis, taking defensive actions too late, missing the opportunity of earning money. Note that similar drawbacks result, at least in some scenarios, from time spent in computing an optimal or near-optimal solution, with respect to performing a suboptimal action earlier.

Moreover, if the knowledge base has been designed independently of the explanation/action task (e.g., diagnosis and repair), it could include a detailed description of the domain which is not necessary for the task; more generally, the usefulness of a detailed discrimination may depend on the specific case at hand.

Finally, human problem solvers have knowledge and are able to reason on abstract actions, such as "taking an antibiotic therapy" if the leading hypothesis is "bacterial infection", and evaluating their costs in a broad sense, for example including side effects, without necessarily reasoning on specific instances. Of course, an abstract action cannot be executed directly, but abstract knowledge may be used to consider it as a candidate "next step" before committing to a specific instance.

The main expected benefit in explicitly considering abstractions in the iterative abduction process is a significant reduction of the computational cost of deciding what to do next (observe or perform an action? which observation or action?), without significantly increasing the total cost of the observations and actions performed to solve the problem.

In the following, we first describe the knowledge we expect to be available and define the concept of explanation of a set of observations in general terms. Then, we describe a specific syntax for expressing the knowledge (based on causal graphs) and associate a precise semantics with such syntax in terms of propositional logic. The syntax and the associated semantics described in the paper are by no means the only possible choice; however they make the generic notion of explanation more concrete for illustrative purposes, and they are used for building an implementation of the method.

In the subsequent sections we describe a basic iterative abductive problem solving loop and we concentrate on the execution of actions and their

estimated costs, and on the criterion for selecting the next step in the loop: either performing a further observation at some level of detail, or an abstract or concrete action. Then, we describe the key points of our implementation of the method and present experimental results which confirm the expectations about the advantages of using abstract hypotheses and actions.

For the same purpose of making the framework concrete, in the problem solving loop we assume that actions are "repair" actions, in the sense that, as in most forms of diagnostic problem solving, they make a corresponding hypothesis false, i.e., they remove the cause of the problem (as far as causes are modeled in the domain), while observations are evidence of the problem. In other settings, e.g., plan recognition, the purpose of the action, to be chosen appropriately for the situation (which can only be assessed through hypothetical reasoning), may be different from making the hypothesized situation false. However, also in these settings performing an action is at least useful (like performing further observations) to confirm or disconfirm the correctness of the hypothesis, even though computing the predicted effects of performing the action may be different from "removing symptoms if the hypothesis was correct", as in the concrete framework described in the paper.

2. Domain Representation

2.1. Hierarchies

The basic elements of the domain model are a set of abducibles (i.e., assumptions, hypotheses) $\mathcal{A} = \{A_1, \ldots, A_n\}$ and a set of manifestations (i.e., observables) $\mathcal{M} = \{M_1, \ldots, M_m\}$.

Each abducible A_i is associated with an IS-A hierarchy $\Lambda(A_i)$ containing abstract values of A_i as well as their refinements at multiple levels; similarly, each manifestation M_j is associated with an IS-A hierarchy $\Lambda(M_j)$.

We assume that the direct refinements $v_1, \ldots, v_q$ of a value V in a hierarchy (either $\Lambda(A_i)$ or $\Lambda(M_j)$) are mutually exclusive, i.e., the Λ hierarchies are trees; moreover, in a given situation, exactly one ground instance (i.e., leaf) of each manifestation M_j is *true* while, for each abducible A_i, either one ground instance is *true* (i.e., the abducible is present) or none of them is *true* (i.e., the abducible is not present). The assumption that there is always a *true* leaf for each manifestation M_j is made just for convenience, so that increasing our knowledge about manifestations can always be viewed as a refinement of the previous knowledge; clearly, knowing that the root of a manifestation hierarchy $\Lambda(M_j)$ is *true* represents complete lack of knowledge about M_j (see section 3).

The overall goal of our problem solving process is to perform actions that remove all of the abducibles which are present in a given situation at an (approximately) minimum cost.

The abducibles set $vals(A_i)$ of an abducible A_i is the set of all of the elements of the hierarchy $\Lambda(A_i)$, while $gndvals(A_i)$ is the subset of $vals(A_i)$ containing only ground abducibles, i.e., the leaves of hierarchy $\Lambda(A_i)$. The definition of set *vals* (resp. *gndvals*) can be extended to a set of abducibles by taking the union of the *vals* (resp. *gndvals*) of each abducible in the set; we also define set *vals* (resp. *gndvals*) for an abducible value α belonging to the hierarchy $\Lambda(A_i)$ by considering only the values (resp. ground values) belonging to the subhierarchy $\Lambda(\alpha)$ of $\Lambda(A_i)$ rooted at α.

The sets *vals* and *gndvals* are defined for manifestations M_j in the same way as for abducibles.

We assume that an a-priori probability $p(a)$ is given for each leaf value a of an abducible A: in fact, we assume that different abducibles are independent. Instead, the probability of an inner node is defined as the sum of the probabilities of its direct refinements (which, as said before, are mutually exclusive).

We also associate costs with the (ground) values of abducibles and the (abstract) values of manifestations.

The cost of a ground abducible value represents the cost of the action needed to remove (e.g., repair) it; in general, such a cost may depend on the current status of the world, however, in this paper we assume that for each leaf value a of an abducible, a cost $rc(a)$ is assigned, independently of the current hypotheses.

As for the manifestations, let ω be an internal value belonging to the IS-A hierarchy of M_j (i.e., $\omega \in vals(M_j) \backslash gndvals(M_j)$); its cost $oc(\omega)$ is the cost of making the observation which refines the value ω into one of its children $\omega_1, \ldots, \omega_q$ in $\Lambda(M_j)$.

2.2. Explanatory Knowledge

The hypotheses space $\mathcal{S}(\mathcal{A})$ is the set of all of the combinations $\gamma = \{\alpha_1, \ldots, \alpha_r\}$ of values drawn from zero or more distinct hierarchies $\Lambda(A_i)$ (i.e., we allow the presence of multiple abducibles at the same time) and, similarly, the observations space $\mathcal{S}(\mathcal{M})$ is the set of all of the combinations $\mu = \{\omega_1, \ldots, \omega_m\}$ of values drawn from each of the distinct hierarchies $\Lambda(M_j)$. In the paper, γ will be referred to as a *candidate explanation* (*candidate* for short) and μ as an *observation*.

If γ and γ_A are two candidates with the same number r of abducible values, and each value $\alpha_i \in \gamma$ is a (possibly improper) refinement of a value $\alpha_{A,i} \in \gamma_A$ according to the IS-A hierarchies of abducibles, then we say that γ_A is an abstraction of γ and, conversely, that γ is a refinement of γ_A. A similar relationship can be defined between two observations μ and μ_A, by taking into account the IS-A hierarchies of manifestations.

The relationship between abducibles and manifestations is defined by the explanatory domain knowledge $\mathcal{K}_E \subseteq \mathcal{S}(\mathcal{A}) \times \mathcal{S}(\mathcal{M})$. Given an observation $\mu \in \mathcal{S}(\mathcal{M})$ and a candidate $\gamma \in \mathcal{S}(\mathcal{A})$, the fact that $(\gamma, \mu) \in \mathcal{K}_E$ means that μ is a possible observation corresponding to candidate γ (and, conversely, that γ is a possible *explanation* for μ).

Our definition of $\mathcal{K}_E$ as a relation between sets $\mathcal{S}(\mathcal{A})$ and $\mathcal{S}(\mathcal{M})$ does not imply that such a relation should be represented extensionally and that the reasoning algorithms should directly manipulate such an extensional representation. In general, $\mathcal{K}_E$ will be specified intensionally with a multi-valued propositional or causal model whose semantics corresponds to the extensional enumeration of the tuples in $\mathcal{K}_E$ (in the next section we discuss the intensional representation to which we will refer in this paper). Moreover, also the reasoning involving $\mathcal{K}_E$ may take place at the syntactic level, e.g., as propositional or causal inference.

Given the above definition of $\mathcal{K}_E$, the set Γ of candidate explanations (*candidate set*) for an observation $\mu \in \mathcal{S}(\mathcal{M})$ is:

$$\Gamma = \{\gamma \in \mathcal{S}(\mathcal{A}) : (\gamma, \mu) \in \mathcal{K}_E\}$$

An important issue is that there may be too many ground explanations of the given observations. This problem may be solved much more efficiently thanks to the presence of abstractions in the model and, in particular, to the fact that abstract as well as ground abducibles may take part in explanations.

A general criterion which is suitable in this setting is the preference for *least presumptive* explanations [18], which generalize minimal (wrt set inclusion) explanations, in order to avoid both unnecessary assumptions, when a subset of assumptions is sufficient to explain the observations, and assumptions that are unnecessarily specific, when a less specific assumption is sufficient. An explanation γ is more presumptive than another explanation γ' if (also based on the IS-A hierarchies $\Lambda(A_i)$) γ implies γ'.

Guaranteeing that a set of explanations is the set of *least presumptive* explanations is, in general, computationally complex; in the following, we just require that the sets of candidate explanations Γ computed during the problem solving process do not contain explanations that are more presumptive than other members of Γ.

3. A Causal Graph Representation Formalism

In Figure 1 we show a fragment of a fictitious medical domain model, where we have adopted a causal graph formalism inspired by [7]. In this section we describe the formalism and relate it to the explanation knowledge $\mathcal{K}_E$ through its semantics.

3.1. Representation Formalism

We describe the formalism (which should be fairly intuitive) through the example of Figure 1. On the left, there is the nosological description of some diseases, represented as three IS-A hierarchies of abducibles (with roots *D1*, *D2*, and *D3*). For example, *D1.1* and *D1.2* are two refinements of *D1*. The a-priori probabilities of the leaves of abducibles (not shown in the figure) are assumed to be $\frac{1}{2^8}$, except for $p(D1.1) = \frac{1}{2^7}$. The costs *rc* of the actions that remove the ground abducibles are shown in the figure.

On the right, there are possible symptoms and possible medical examinations (lab tests) to be performed, represented as three IS-A hierarchies of manifestations (with roots *Sym1*, *LT1*, and *LT2*). Observation costs *oc* associated with each internal node of manifestation hierarchies are the costs of performing the related laboratory test (we assume that the cost of observing the presence of symptoms such as *Sym1* is 0).

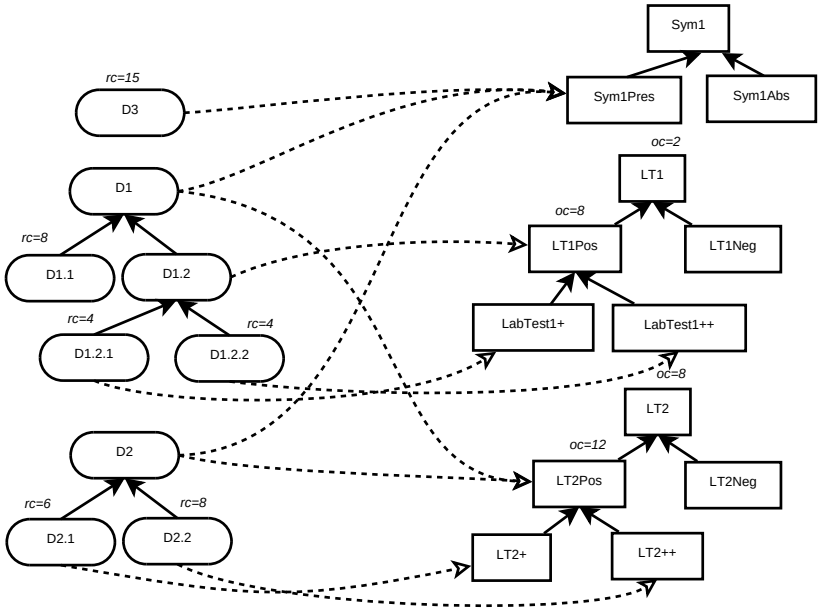

Fig. 1. A (fictitious) medical domain model.

The relationships between abducibles and manifestations are represented by rightwards dashed arrows. For example, *D1* causes *LT2* to be positive; *D1.2* causes *LT1* to be positive, and its refinements *D1.2.1* and *D1.2.2* cause more specific positive values of *LT1*.

3.2. Propositional Semantics

In order to map the graph-based formalism adopted in our example to the explanatory knowledge $\mathcal{K}_E$, we interpret the graph as a propositional theory $\mathcal{T}_E$; the tuples of $\mathcal{K}_E$ will then be straightforwardly obtained from the logical models that satisfy such a theory (see section 3.3).

First of all, each value in a hierarchy $\Lambda(A_i)$ or $\Lambda(M_j)$ is mapped to a propositional variable. If value V has children $v_1, \ldots, v_q$ in a hierarchy, a natural representation in $\mathcal{T}_E$ would be:

$$\begin{array}{l} V \Leftrightarrow v_1 \vee \ldots \vee v_q \\ \forall i \neq j \neg(v_i \wedge v_j) \end{array} \quad (1)$$

expressing the fact that an abstract value V can be refined in exactly one of its children $v_1, \ldots, v_q$, which is what we stated in our discussion in section 2.1.

However, our aim in defining $\mathcal{T}_E$ is to be able to map its (2-valued) logical models as directly as possible to candidate explanations; in this respect, the problem with the above translation is that in each logical model of formulas (1) where V is *true*, also one child v_i must be *true* while, for the purpose of computing explanations, we want to allow candidates where none of the children is *true*, i.e., where V alone is an (abstract) explanation.

Consider, e.g., the graph of Figure 1 and assume that we know that *LT1Pos* is *true*; we would like to have an explanation where *D1.2* is *true* but none of its children *D1.2.1* and *D1.2.2* is *true*, to avoid commitment.

In order to handle this issue, for each internal value V of each hierarchy, we add a variable uk_V to represent explicitly, in a 2-valued model, the fact that it is unknown which refinement of V is true. If value V has children $v_1, \ldots, v_q$, instead of formulas (1), the following formulas are added to $\mathcal{T}_E$:

$$\begin{array}{l} V \Leftrightarrow v_1 \vee \ldots \vee v_q \vee uk_V \\ \forall i \neq j \neg(v_i \wedge v_j); \quad \forall i \neg(v_i \wedge uk_V) \end{array} \quad (2)$$

Note that this translation is not intended as a general, logically satisfactory approach to the logic of knowledge; its purpose is to have abstract explanations as 2-valued propositional interpretations (see section 5.1).

The relationships between abducibles and manifestations are translated as follows, adapting the completion semantics of abduction described in [8] to take into account the fact that abstract assump-

tions may not be predictive enough to entail observations [14]. Let ω be a value in a manifestation hierarchy $\Lambda(M)$, such that in the causal graph the abducible values $\alpha_1, \ldots, \alpha_k$ point to ω through causal arrows (note that $\alpha_1, \ldots, \alpha_k$ are, in general, values belonging to different abducible hierarchies). Moreover, let $\beta_1, \ldots, \beta_l$ be abducible values that point to possibly not distinct ancestors $\omega_1, \ldots, \omega_l$ of ω, such that none of the descendants of each β_i points to ω or an ancestor of ω below ω_i. Then, we add the following formulas to $\mathcal{T}_E$:

$$\begin{array}{l} \alpha_1 \vee \ldots \vee \alpha_k \Rightarrow \omega \\ \omega \Rightarrow \alpha_1 \vee \ldots \vee \alpha_k \vee \beta_1 \vee \ldots \vee \beta_l \end{array} \quad (3)$$

Note that, if for a manifestation value ω there are no α_is and no β_js satisfying the above conditions, we do not add any formula. In other words, ω is interpreted as a value which is consistent with any abducible value α, as far as α does not explicitly predict a value ω' such that ω and ω' are mutually exclusive according to formulas (2). In Figure 1, this role is played by *Sym1Abs*, *LT1Neg* and *LT2Neg*, which do not have any incoming causal arc.

Let us consider some examples from the model of Figure 1. Two abducible values point to *LT2Pos*, namely *D1* and *D2*; since no abducible value points to the (only) ancestor *LT2* of *LT2Pos* (i.e., there are no β_is in formula (3)), we add the following formulas:

$$\begin{array}{l} D1 \vee D2 \Rightarrow LT2Pos \\ LT2Pos \Rightarrow D1 \vee D2 \end{array}$$

According to the first formula, if *LT2Pos* is *false*, then also *D1* and *D2* are *false*, i.e., neither *D1*, nor *D2*, nor any refinements of such diseases can be explanations. If, on the other hand, *LT2Pos* is observed to be *true*, then, according to the second formula, either *D1* or *D2* must be true; in turn, this may be due to, e.g., the fact that the refinement *D1.2* of *D1* is *true* but, as discussed above, it may also be the case that uk_{D1} is *true*, i.e., we do not need to commit to any particular refinement of *D1* in order to explain *LT2Pos*.

Let us now consider a more complex example, where the β_is of formula (3) are involved. The only abducible value pointing to *LT2+* is *D2.1*; however, *D1* points to the ancestor *LT2Pos* of *LT2+*; therefore we add the following formulas:

$$\begin{array}{l} D2.1 \Rightarrow LT2+ \\ LT2+ \Rightarrow D2.1 \vee D1 \end{array}$$

The second formula states that if *LT2+* is *true*, then *D2.1* or *D1* must be *true*: the first one, because it directly causes *LT2+*; the second one because it causes the ancestor *LT2Pos* of *LT2+* and its descendants do not predict more specific values.

In this way, although the abducible hierarchy of *D1* predicts the value of manifestation *LT2* at a coarser level of granularity than the abducible hierarchy of *D2*, we still allow *D1* to explain some fine-grained values of *LT2*, such as *LT2+*.

Adopting the hierarchical formulas (2) with uk_V variables has the benefit of allowing abstract explanations, as discussed above, but it also weakens the theory $\mathcal{T}_E$. In particular, let us assume that an observation value ω is *not* unknown (i.e., one of its children ω_j is known to be *true*), and that a child α_i of an abducible value α points to one of the other children ω_h of ω, $h \neq j$. According to formulas (2), it may still be possible that uk_α is *true*, i.e., that α is an abstract explanation of ω_j. We would like to avoid α being an explanation of ω_j when (at least) one of its children, namely α_i, is certainly not *true*. To this end, we add to $\mathcal{T}_E$ the formula:

$$\neg uk_\omega \Rightarrow (\neg\omega_h \Rightarrow \neg uk_\alpha) \quad (4)$$

The formula says that, unless the value of ω is unknown, if value ω_h is *false* then abducible value α is *not* unknown, i.e., α cannot be an (abstract) explanation.

To illustrate this point, let us consider value *LT1* in Figure 1, and let us assume that we have excluded its child *LT1Pos* by observing *LT1Neg*. Thanks to the presence of the formula:

$$\neg uk_{LT1} \Rightarrow (\neg LT1Pos \Rightarrow \neg uk_{D1})$$

the manifestation value *LT1Neg* cannot be explained by *D1* alone, although it can still be explained by one of its refinements, namely *D1.1*.

3.3. Mapping to the Explanatory Knowledge

Once the theory $\mathcal{T}_E$ has been generated from the causal graph, its logical models are easily mapped to an explanatory knowledge $\mathcal{K}_E \subseteq \mathcal{S}(\mathcal{A}) \times \mathcal{S}(\mathcal{M})$ as defined in section 2.2.

Let $\mu = \{\omega_1, \ldots, \omega_m\}$ be any observation. Starting from theory $\mathcal{T}_E$, we want to define the portion $\mathcal{K}_E(\mu)$ of $\mathcal{K}_E$ which contains the explanations of μ. To this end, we start by considering the propositional theory:

$$\mathcal{T}_E(\mu) = \mathcal{T}_E \cup \{\omega_1 \wedge \ldots \wedge \omega_m\}$$

which asserts observation $\mu = \{\omega_1, \ldots, \omega_m\}$ in $\mathcal{T}_E$.

Let us denote as $\mathcal{M}(\mu)$ a logical model of $\mathcal{T}_E(\mu)$, and restrict it to a partial model $\mathcal{M}(\mu)_{\mathcal{A}}$ which assigns truth values only to the variables associated with abducibles:

$$\mathcal{M}(\mu)_{\mathcal{A}} = \mathcal{M}_{A_1} \cup \ldots \cup \mathcal{M}_{A_n}$$

where $\mathcal{M}_{A_i}$ is a truth assignment to each variable in $vals(A_i)$. Note that $\mathcal{M}(\mu)_{\mathcal{A}}$ does *not* contain the truth assigments to the unknown variables uk_α associated with the abducible values.

From each model $\mathcal{M}(\mu)_{\mathcal{A}}$ we can derive exactly one candidate γ as follows:

- if $\mathcal{M}_{A_i}$ assigns *false* to each variable, then none of the values $vals(A_i)$ of abducible A_i belongs to γ;
- otherwise, let α_i be the most specific value of A_i such that $\mathcal{M}_{A_i}$ assigns *true* to α_i; then α_i belongs to γ.

The candidate derived from $\mathcal{M}(\mu)_{\mathcal{A}}$ as we have just explained is denoted as $\gamma(\mathcal{M}(\mu)_{\mathcal{A}})$.

The relation $\mathcal{K}_E(\mu)$ containing the explanations of μ is thus defined as:

$$\mathcal{K}_E(\mu) = \{(\gamma, \mu) \mid \exists \mathcal{M}(\mu)_{\mathcal{A}} : \gamma = \gamma(\mathcal{M}(\mu)_{\mathcal{A}})\}$$

i.e., the explanations of μ are the candidates derived from the (partial) models $\mathcal{M}(\mu)_{\mathcal{A}}$ of $\mathcal{T}_E(\mu)$.

Finally, $\mathcal{K}_E$ itself is defined as the union of $\mathcal{K}_E(\mu)$ for each possible observation μ. As stated before, our purpose is not that of explicitly enumerating all of the tuples of relation $\mathcal{K}_E$, which would be infeasible for all but the smallest domain models. Instead, the above discussion implies that we can compute the explanations of any observation μ by directly manipulating the propositional theory $\mathcal{T}_E$.

Let us consider again an example from the model of Figure 1. We may ask whether, according to the above definitions, candidate $\gamma = \{D1.2\}$ is an explanation of observation $\mu = \{Sym1Pres, LT1, LT2\}$, i.e., whether $(\{D1.2\}, \{Sym1Pres, LT1, LT2\}) \in \mathcal{K}_E$.

It is easy to see that the propositional semantics of the graph includes a logical model where the only abducible variables that are *true* are *D1*, *D1.2* and $uk_{D1.2}$, while the *true* manifestation variables are *Sym1Pres*, *LT1*, *LT1Pos*, *LT2*, *LT2Pos* plus other unknown variables and variables associated with their refinements.

input: a set of values $\hat{\varphi} = \{\hat{\omega}_1, \ldots, \hat{\omega}_m\}$ representing the initial observations

$\varphi := \hat{\varphi}$
$\Sigma := []$
generate a set Γ of candidates γ which explain $\hat{\varphi}$
loop
  if $\Gamma = \{\emptyset\}$ **then exit**
  $\rho := \{\alpha \mid \exists \gamma_i \in \Gamma : \alpha \in \gamma_i\}$
  $\sigma :=$ ChooseNextStep(Γ, φ, ρ)
  $\Sigma := \Sigma \cdot \sigma$
  if $\sigma = \omega \in \varphi$
    $(\varphi, \Gamma) :=$ Observe(φ, Γ, Σ, ω)
  elseif $\sigma = \alpha \in \rho$
    $(\varphi, \Gamma) :=$ Remove(φ, Γ, Σ, α)
  endif
end

Fig. 2. Main loop of the troubleshooting algorithm.

Clearly, if $\mu = \{Sym1Pres, LT1, LT2\}$, this is also a logical model $\mathcal{M}(\mu)$ of $\mathcal{T}_E(\mu) = \mathcal{T}_E \cup \{Sym1Pres, LT1, LT2\}$. Let us now consider the restriction $\mathcal{M}(\mu)_{\mathcal{A}}$ of $\mathcal{M}(\mu)$ to the $\mathcal{A}$ variables; by eliminating all the assignments to uk and manifestation variables, $\mathcal{M}(\mu)_{\mathcal{A}}$ assigns *true* just to abducible variables *D1* and *D1.2*, both belonging to the portion $\mathcal{M}_{D1}$ of $\mathcal{M}(\mu)_{\mathcal{A}}$.

From the rules for deriving a candidate γ from $\mathcal{M}(\mu)_{\mathcal{A}}$, it follows that $\gamma(\mathcal{M}(\mu)_{\mathcal{A}}) = \{D1.2\}$, so we finally conclude that $(\{D1.2\}, \{Sym1Pres, LT1, LT2\}) \in \mathcal{K}_E$.

From this example, we easily see that $\{D1.2\}$ is also an explanation for, e.g., $\{Sym1Pres, LT1Pos, LT2\}$ or $\{Sym1Pres, LT1Pos, LT2Pos\}$, where more refined observation values have been included into the observation μ that we want to explain.

4. Method Description

4.1. Troubleshooting Algorithm

The algorithm shown in Figure 2 illustrates the overall approach to troubleshooting with abstractions we propose in this paper.

We define $\varphi = \{\omega_1, \ldots, \omega_m\}$ as the current *fringe* over the manifestations, containing the most specific values ω_j known to be true so far for manifestations M_j, $j = 1, \ldots, m$.
Since we assume that at least one ground value of

each manifestation is true in each situation, if we do not have any information about the value of a manifestation M_j, its value in φ is the root of the hierarchy for M_j, i.e., $\omega_j = root(\Lambda(M_j))$; otherwise, ω_j may be a more specific value in $vals(M_j)$.

An initial fringe $\hat{\varphi}$ of observations is given and the fringe is updated as the problem solving process goes on. We also initialize the sequence Σ of observations/actions performed so far to the empty sequence []. The sequence Σ will be useful for ignoring actions that have already been performed [22].

Given the set of initial observations $\hat{\varphi}$, a set of candidate explanations Γ are generated.

At each iteration of the main loop, we first check whether Γ contains just an empty candidate $\emptyset$, meaning that the problem is solved and the algorithm can terminate. If this is not the case, the set ρ of abducible values that appear in Γ is computed.

Then, we have to choose whether to perform an observation, in order to refine or discriminate the candidates, or to remove an abducible, by implicitly performing the related action (for example a repair action in the troubleshooting context). We select what to do next based on the current candidate set Γ, the fringe φ and the set of abducibles ρ. Clearly, this choice is in general suboptimal, due to the prohibitive complexity of making an optimal choice.

If the choice is to perform an observation, the candidate set Γ and the fringe φ are updated according to its outcome (call to function *Observe*). In particular, if ω is a value in manifestation hierarchy $\Lambda(M)$ and the outcome of observing ω is ω_k, then ω is replaced by ω_k as the value of M in φ; the candidate set Γ is updated by generating the candidate explanations for the updated fringe.

Also when the choice is to remove an abducible value α, Γ and φ must be updated (call to function *Remove*). The details of such an update are given in the next section.

4.2. Removing Abducibles

In this section we describe how the fringe φ and the candidate set Γ are updated when an abducible value α is removed. We denote as φ' and Γ' the updated fringe and candidate set, respectively.

Let us start by considering the update of φ. In order to update φ, the first step is a transformation of the previous candidate set Γ. In particular, if α is a ground abducible value a, we compute a set $\Gamma_{\overline{a}}$ by updating each candidate $\gamma \in \Gamma$ as follows:

$$\gamma' = \gamma \backslash \{a\}$$

(clearly, if $a \notin \gamma$ then $\gamma' = \gamma$; note also that if $\gamma = \{a\}$, then $\gamma' = \emptyset$).

Each updated candidate γ' will make, in general, a set of possibly non-deterministic predictions $\{\mu_1, \ldots, \mu_q\}$ on the values of manifestations $\mathcal{M}$, where each prediction is a set $\mu_i = \{\omega_{i,1}, \ldots, \omega_{i,m}\}$. Each candidate γ' represents all of its possible extensions and refinements; therefore the predictions of γ' are all the observations μ_i that can be induced by any of its extensions/refinements. For example, the candidate $\gamma' = \emptyset$ will make essentially no prediction (except the obvious fact that the roots of each manifestation are present), since it can be extended to any other candidate.

Let us denote with $LUB(\gamma', M_j)$ the value in $\Lambda(M_j)$ that is the least upper bound of $\{\omega_{1,j}, \ldots, \omega_{q,j}\}$ (i.e., of the set of values for M_j predicted by γ'). The new fringe predicted by γ' will then be $\varphi(\gamma') = \{LUB(\gamma', M_1), \ldots, LUB(\gamma', M_m)\}$ (recall that a value in the fringe for M_j is the most specific value of M_j known to be certainly true). Similarly, we denote with $LUB(\Gamma_{\overline{a}}, M_j)$ the value in $\Lambda(M_j)$ that is the least upper bound of the set $\{LUB(\gamma', M_j) : \gamma' \in \Gamma_{\overline{a}}\}$ (i.e., of the set of values for M_j predicted by $\Gamma_{\overline{a}}$). The new fringe predicted by $\Gamma_{\overline{a}}$ will therefore be $\varphi(\Gamma_{\overline{a}}) = \{LUB(\Gamma_{\overline{a}}, M_1), \ldots, LUB(\Gamma_{\overline{a}}, M_m)\}$.

The updated fringe φ' should be set to $\varphi(\Gamma_{\overline{a}})$; note however that $\varphi(\Gamma_{\overline{a}})$ may contain very weak (i.e., abstract) values for manifestations, since they must be consistent with all of the possible predictions made by all of the (modified) candidates in $\Gamma_{\overline{a}}$. For this reason, it is useful to assume that, for a (possibly empty) subset $\mathcal{M}^*$ of manifestations, it is possible to perform at no cost an immediate check (at a given level of abstraction) after the removal of an abducible.

In particular, following [23], we define a *cut* $\mathcal{C}(M)$ on a hierarchy $\Lambda(M)$ to be a set of values $\omega \in vals(M)$ such that each ground value in $gndvals(M)$ is an instance of exactly one $\omega \in \mathcal{C}(M)$ (i.e., a cut can be seen as a curve line which makes an horizontal cut of the tree $\Lambda(M)$ in two parts by touching a set of values at possibly different levels of abstraction).
The immediate check on each manifestation $M_j \in$

$\mathcal{M}^*$ will result in exactly one (abstract) value ω_j belonging to the cut $\mathcal{C}^*(M_j)$ associated with $\Lambda(M_j)$.
For instance, in our running example, the manifestation *Sym1* is associated with the cut {*Sym1Pres*, *Sym1Abs*}, i.e., after performing an action we know for free whether the symptom persists (*Sym1Pres*) or it has disappeared (*Sym1Abs*). For the other manifestations *LT1* and *LT2*, we assume trivial cuts consisting just in the roots of the respective hierarchies.
Note that a cut may in general consist of both ground and abstract values: for instance, {*LT1Pos*, *LT1Neg*} would be a valid cut for *LT1*, although *LT1Pos* is an abstract value, while *LT1Neg* is a ground value.

In general, the observed value $\omega_j \in \mathcal{C}^*(M_j)$ of a manifestation $M_j \in \mathcal{M}^*$ may be more precise than the predicted value $LUB(\Gamma_{\overline{a}}, M_j)$ and it will therefore be included in the new fringe φ'.

The updated candidate set Γ' will be computed by generating the explanations for φ', taking into account that the abducibles in the sequence Σ (including a) will not be part of any explanation.

Let us now consider the execution of an action for removing an abstract abducible value α. Since α, by being abstract, is not associated with an action which could remove it, we need to iteratively remove the values $a \ : \ a \in gndvals(\alpha)$ until we ensure that α has indeed been removed (i.e., has become *false*) in the candidates Γ.

The algorithm starts by selecting an approximately best value a to remove (see below). After the removal of abducible value a, there are two possible cases: either some manifestations in $\mathcal{M}^*$ have changed, or not. In the first case, a was clearly the real refinement of α, so the removal of α is complete. Otherwise, a new approximately best value a' is selected, and so on until some manifestations in $\mathcal{M}^*$ change. It is possible (in particular when we have chosen to remove an adbucible α that was not present in the first place), that we need to remove all the ground values in $gndvals(\alpha)$, since we don't detect any change in the manifestations.

The selection of the best value to remove is based on a slight modification of the *efficiency* measure defined in [13]. In particular, let us denote with Γ_a the subset of Γ whose candidates contain the abducible value a (i.e., $\Gamma_a = \{\gamma \in \Gamma : a \in \gamma\}$); the efficiency of value a is defined as:

$$ef(a) = \frac{p(\Gamma_a|\Gamma)}{rc(a)}$$

Intuitively, the efficiency of a is increased by the probability that the abducible value a is in the candidate set, and it is decreased by the cost of removing it. The value a to be removed next is the one with the highest efficiency.

Independently of the sequence of values ($a_1, \ldots, a_q$) which is actually removed, once the removal of α is complete we can proceed to update φ and Γ as in the case of a ground abducible value described above.

4.3. Estimated Cost of Removing Abducibles

In the previous section we have considered the actual removal of an abducible value, and its effects on φ and Γ. In this section we consider the problem of *estimating* the cost of such a removal before actually executing any action. This estimate is needed in order to choose what should be done next, i.e., observe or remove, in the call to *ChooseNextStep* in Figure 2; we will explain such a choice in detail in the next section.

If α is a ground abducible value a, then the cost $rc(a)$ is defined directly in the model, so there is no need to estimate it. If, on the other hand, α is an abstract value, its cost can be estimated by adapting to our setting a simple technique from the troubleshooting literature, namely the *greedy* approach of [16]. Let $(a_1, \ldots, a_q)$ be the sequence of ground values $a_i \in gndvals(\alpha)$ in decreasing efficiency order according to the formula introduced in the previous section. The estimated cost of removing α from a candidate set Γ is computed as follows:

$$rc_\Gamma(\alpha) = \sum_{i=1}^{q} rc(a_i) \cdot \left[1 - p(\alpha|\Gamma) \cdot p(\alpha \not\in \Gamma^i|\alpha)\right] \quad (5)$$

where Γ^i is the candidate set *after* values $a_1, \ldots, a_{i-1}$ have been removed starting from candidate set Γ. When convenient, we will use the notation $rc_\Gamma(a)$ to denote the fixed cost $rc(a)$ defined in the model for a ground abducible value a.

To understand this definition, let us first note that the most efficient ground value a_1 will always be removed at the cost $rc(a_1)$: indeed, $\Gamma^1 = \Gamma$ and therefore $p(\alpha \not\in \Gamma^1|\alpha) = 0$, since α is considered for removal just because it appears in Γ. As for a_2, its cost $rc(a_2)$ will be paid, *except* in case, after

removing a_1, the resulting candidate set Γ^2 does no longer contain α (i.e., α has been removed by removing a_1, and this fact has been detected - see below). Similarly, the cost of $a_i, i > 2$, will be paid *except* in case, after removing $a_1, \ldots, a_{i-1}$, the resulting candidate set Γ^i does no longer contain α.

The exact way $p(\alpha \notin \Gamma^i|\alpha)$ is computed depends on the set of manifestations $\mathcal{M}^*$ that can be checked at no cost after the execution of each action, and on the cuts associated with such checks. One possibility is to make the strong assumption that the removal of any value a_i (when a_i is actually present) always makes the manifestations in $\mathcal{M}^*$ change; in such a case:

$$p(\alpha \notin \Gamma^i|\alpha) = \sum_{j=1}^{i-1} p(a_j|\alpha)$$

i.e., Γ^i will not contain α provided one of the values $a_1, \ldots, a_{i-1}$ was present.

If, on the other hand, we make the weaker assumption that the manifestations in $\mathcal{M}^*$ change only when the problem has been solved, then:

$$p(\alpha \notin \Gamma^i|\alpha) = \begin{cases} 0 & \text{if } \{\alpha\} \notin \Gamma \\ \sum_{j=1}^{i-1} p(a_j|\alpha) & \text{otherwise} \end{cases} \quad (6)$$

since, if there is no candidate containing just $\{\alpha\}$, the problem will certainly not be solved by removing α while, otherwise, detecting that the problem is solved is equal to detecting that α has been removed.

The assumption we choose to make (the strong or weak ones described above, as well as other ones) will not affect the correctness of the algorithm, since it is used just for estimating the cost of removing α. However, the assumption should reflect as far as possible the characteristics of the domain (in this case, the number and discrimination power of manifestations $\mathcal{M}^*$), in order to make the estimate as precise (and useful) as possible. For the example in section 4.5 and for the experimental evaluation in section 5, we will use the weaker assumption.

4.4. Choosing the Next Step

As discussed in section 4.1, at each iteration of the problem solving process we need to choose whether to observe a value ω in the fringe φ or to remove an abducible α in the set ρ.

For each $\omega \in \varphi$, we evaluate the estimated cost $c(\omega)$, which is the sum of the cost $oc(\omega)$ of refining ω and the expected cost of the candidate set after refining ω, i.e.:

$$c(\omega) = oc(\omega) + \sum_{k=1}^{q} p(\omega_k|\Gamma) \cdot c(\Gamma_k) \quad (7)$$

where $\Gamma_1, \ldots, \Gamma_q$ are the possible candidate sets that would result by observing ω and getting values $\omega_1, \ldots, \omega_q$ respectively; $p(\omega_k|\Gamma)$ is the probability of getting value ω_k (computed based on current candidates Γ); and $c(\Gamma_k)$ is the estimated cost of Γ_k as detailed below.

For each $\alpha \in \rho$, we evaluate the estimated cost $c(\alpha)$, which is the sum of the cost $rc_\Gamma(\alpha)$ of removing α and the expected cost of the candidate set after removing α, i.e.:

$$c(\alpha) = rc_\Gamma(\alpha) + \sum_{\Gamma_k \in PW(\Gamma,\alpha)} p(\Gamma_k|\Gamma) \cdot c(\Gamma_k) \quad (8)$$

where $PW(\Gamma, \alpha)$ (for *possible worlds*) is an *estimate* of the possible candidate sets resulting from the removal of α and $p(\Gamma_k|\Gamma)$ is the probability that the actual candidate set after removing α is Γ_k.

Let us consider the set $PW(\Gamma, \alpha)$ in more detail. As in the actual removal of an abducible value, we first compute the candidate set $\Gamma_{\overline{\alpha}}$ obtained by removing α from each candidate in Γ. In order to simplify our estimate, we consider that each candidate $\gamma' \in \Gamma_{\overline{\alpha}}$ represents just itself, instead of representing also all of its extensions and refinements, e.g., we do not interpret $\emptyset$ as the representation of any possible candidate as we do in the actual execution of actions, but just as the representation of the case where none of the abducible values is present. This approximation makes the predictions of candidates γ' much more precise, improving the efficiency of the estimate as we shall see shortly.

We then consider the predictions made by the candidates in $\Gamma_{\overline{\alpha}}$ on the $\mathcal{M}^*$ manifestations at the level of the cuts $\mathcal{C}^*(M_j)$, and group all of the candidates γ' which make the same predictions into the same possible world Γ_k. The set $PW(\Gamma, \alpha)$ will contain all of the candidate sets $\Gamma_k \subseteq \Gamma_{\overline{\alpha}}$ obtained in this way.

In general, the number of possible (non deterministic) predictions on manifestations $\mathcal{M}^*$ can be exponential in $|\mathcal{M}^*|$, and therefore $PW(\Gamma, \alpha)$ may be intractable to compute if $\mathcal{M}^*$ is large. However, even when $\mathcal{M}^*$ is large, it is sufficient that the predictions made by the candidates $\gamma' \in \Gamma_{\overline{\alpha}}$ on the values of manifestations $\mathcal{M}^*$ are deterministic

at the level of the cuts $\mathcal{C}^*(M_j)$; in such a case it is easy to see that $|PW(\Gamma, a)|$ is bounded by the number of candidates $|\Gamma|$.

In both equations 7 and 8, we need to be able to estimate the cost of the problem solving process for a candidate set Γ_k.

In order to compute such a cost, we adopt a technique similar to the one adopted for estimating $rc_\Gamma(\alpha)$ (section 4.3), inspired by [16]. In particular, we start by computing ρ_k, i.e., the set of abducible values that appear in Γ_k; then we order such abducibles in decreasing efficiency order using the following formula, which was introduced previously for ground abducible values but can be straightforwardly applied also to abstract abducible values:

$$ef_{\Gamma_k}(\alpha) = \frac{p(\Gamma_\alpha|\Gamma_k)}{rc_{\Gamma_k}(\alpha)}$$

Let $\hat{\rho}_k = (\alpha_1, \ldots, \alpha_q)$ be the sequence of abducible values ordered by decreasing order of efficiency. The cost of Γ_k is computed as follows:

$$c(\Gamma_k) = \sum_{i=1}^{q} rc_{\Gamma_k}(\alpha_i) \cdot p(\Gamma_k^i \neq \{\emptyset\}) \qquad (9)$$

where Γ_k^i is the candidate set *after* abducible values $\alpha_1, \ldots, \alpha_{i-1}$ have been removed starting from candidate set Γ_k.

Note that the cost of each abducible α_i is weighted with the probability that the action to remove it will actually be executed, i.e., that the candidate set Γ_k^i is not equal to $\{\emptyset\}$, which corresponds to the situation where the problem has already been solved and this has been detected, so that the only candidate left is $\emptyset$.

As in the case of the estimate of $rc_\Gamma(\alpha)$, the way $p(\Gamma_k^i \neq \{\emptyset\})$ is computed depends on the set of manifestations $\mathcal{M}^*$ and on the cuts $\mathcal{C}^*(M_j)$. If we assume that, after each action execution, it is possible to check at no cost whether the problem has been solved or not, then:

$$p(\Gamma_k^i \neq \emptyset) = \sum_{\gamma' \in \Gamma_k : \gamma' \not\supseteq \{\alpha_1, \ldots, \alpha_{i-1}\}} p(\gamma'|\Gamma_k)$$

since, if the real world status is a candidate $\gamma' \in \Gamma$ which is completely removed by removing $\alpha_1, \ldots, \alpha_{i-1}$, we must be aware (through our checks) that the problem is solved, and Γ^i must therefore be equal to $\{\emptyset\}$.

After we have computed the expected observation costs $c(\omega)$ and expected action costs $c(\alpha)$, we simply choose the observation or action σ such that:

$$\sigma = argmin_{\hat{\sigma} \in (\varphi \cup \rho)} [c(\hat{\sigma})]$$

i.e., the observation or action of minimum expected cost.

4.5. Example

In order to get a better understanding of the problem solving process, let us consider in detail the execution of the algorithm in Figure 2 on the medical example in Figure 1. A schematic view of the solution process is shown in Figure 3.

Initial observations. Let us suppose that an initial manifestation of *Sym1* is detected, i.e., $\hat{\varphi} = \{Sym1Pres, LT1, LT2\}$. The initial candidate set is $\Gamma = \{\{D1\}, \{D2\}, \{D3\}\}$, representing the possible alternative diagnoses (in fact, *D1*, *D2* and *D3* explain *Sym1Pres*).

First iteration. The abducibles to be considered for removal are $\rho = \{D1, D2, D3\}$, while the fringe φ is initially equal to $\hat{\varphi} = \{Sym1Pres, LT1, LT2\}$. Figure 3 shows the possible choices as dashed arcs leaving the root of the graph; note that we don't consider the observation of *Sym1Pres* since it is a ground value in the hierarchy of manifestation *Sym1*.

The costs are estimated as follows. Regarding the observation $\omega = LT1$, two outcomes are possible: the test is either negative (*LT1Neg*) or positive (*LT1Pos*). The candidate set Γ_N resulting from observing *LT1Neg* is:

$$\Gamma_N = \{\{D1.1\}, \{D2\}, \{D3\}\}$$

Indeed, according to section 3.2, *LT1Neg* is explained by any abducible value except those that predict *LT1Pos*, namely *D1.2* and its children, i.e., exactly by the (least presumptive) candidates contained in Γ_N. Note that these candidates also explain the manifestation values already in φ, in particular *Sym1Pres*.

On the other hand, if the outcome is *LT1Pos*, the candidate set is:

$$\Gamma_P = \{\{D1.2\}\}$$

since, according to equation (3) in section 3.2, the following holds:

$$LT1Pos \Rightarrow D1.2$$

The probability of Γ_N given Γ is:

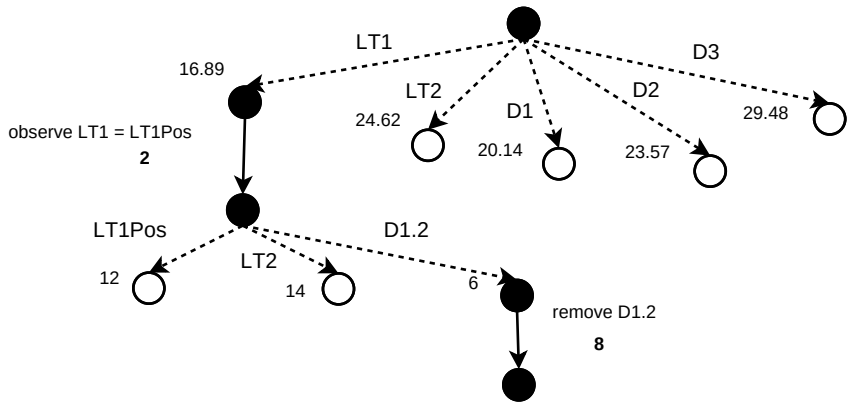

Fig. 3. Graph representing the solution of the example problem. Dashed arcs represent alternative choices to consider, while solid arcs represent actual observations and actions.

$$p(\Gamma_N|\Gamma) = \frac{p(\Gamma_N)}{p(\Gamma)} = \frac{\left(\frac{2}{2^8} + \frac{2}{2^8} + \frac{1}{2^8}\right)}{\frac{7}{2^8}} = \frac{5}{7}$$

Similarly, the probability of Γ_P given Γ is $\frac{2}{7}$.

In order to estimate the cost $c(\Gamma_N)$ we first need to estimate $rc_{\Gamma_N}(D2)$ (the other abducible values in Γ_N are ground, therefore their cost needs not be estimated). The ground values *D2.1*, *D2.2* of *D2* have the same probability, and *D2.1* costs less than *D2.2*, thus we consider them in the order of efficiency (*D2.1*, *D2.2*). We also note that, as soon as we solve the problem, we will immediately detect it at no cost from the symptom *Sym1* (equation (6)); the estimated cost is then computed as follows according to equation (5):

$rc(D2.1) = 6; rc(D2.2) = 8$
$p(D2|\Gamma_N) = \frac{2}{5}$
$p(D2 \notin \Gamma^1|D2) = 0$
$p(D2 \notin \Gamma^2|D2) = p(D2.1|D2) = \frac{1}{2}$
$rc_{\Gamma_N}(D2) = rc(D2.1) + rc(D2.2) \cdot \left[1 - \frac{2}{5} \cdot \frac{1}{2}\right] = 12.4$

The estimated cost $c(\Gamma_N)$ is computed based on the fact that the relative probabilities of the candidates $\{D1.1\}$, $\{D2\}$, $\{D3\}$ are 2, 2, 1, their costs are 8, 12.4, 15 and therefore the sequence in order of decreasing efficiency is $\{D1.1\}$, $\{D2\}$, $\{D3\}$. Then, according to equation (9):

$$c(\Gamma_N) = 8 + \frac{3}{5} \cdot 12.4 + \frac{1}{5} \cdot 15 = 18.44$$

Let us consider the estimated cost of $c(\Gamma_P)$. First of all, we compute $rc_{\Gamma_P}(D1.2)$ as follows:

$rc(D1.2.1) = 4; rc(D1.2.2) = 4$
$p(D1.2|\Gamma_P) = 1$
$p(D1.2.1|D1.2) = \frac{1}{2}$
$p(D1.2 \notin \Gamma^1|D1.2) = 0$
$p(D1.2 \notin \Gamma^2|D1.2) = p(D1.2.1|D1.2) = \frac{1}{2}$
$rc_{\Gamma_P}(D1.2) = rc(D1.2.1) + rc(D1.2.2) \cdot \left[1 - 1 \cdot \frac{1}{2}\right)]$
$= 6$

Since, according to equation (9), $c(\Gamma_P) = rc_{\Gamma_P}(D1.2)$, it follows that $c(\Gamma_P) = 6$. The total expected cost associated with observation *LT1* is therefore:

$$c(LT1) = 2 + \frac{5}{7} \cdot c(\Gamma_N) + \frac{2}{7} \cdot c(\Gamma_P) = 16.89$$

according to equation (7) and recalling that the immediate cost $oc(LT1)$ of performing *LT1* is 2.

Regarding the observation $\omega = LT2$, if the outcome of this observation is negative (*LT2Neg*), then the resulting candidate set is $\Gamma'_N = \{\{D3\}\}$. On the other hand, if the outcome is positive (*LT2Pos*), the candidate set is $\Gamma'_P = \{\{D1\}, \{D2\}\}$. The expected costs are:

$$\begin{aligned} c(\Gamma'_N) &= rc(D3) = 15 \\ c(\Gamma'_P) &= rc_{\Gamma'_P}(D1) + \tfrac{1}{3} \cdot rc_{\Gamma'_P}(D2) \\ &= 12.67 + \tfrac{1}{3} \cdot 12.67 = 16.89 \end{aligned}$$

and then:

$$c(LT2) = 8 + \frac{1}{7} \cdot 15 + \frac{6}{7} \cdot 16.89 = 24.62$$

since the immediate cost of performing *LT2* is 8 and the probabilities of Γ'_N, Γ'_P are, respectively, $\frac{1}{7}$ and $\frac{6}{7}$.

The expected cost associated with removing *D1* is as follows, according to equation (8):

$$c(D1) = rc_\Gamma(D1) + \frac{3}{7} \cdot c(\{\{D2\}, \{D3\}\})$$

since $\{D1\}$ has probability $\frac{4}{7}$ in Γ, and, if removing $\{D1\}$ does not solve the problem, the only remaining possible world is $\Gamma_{\overline{D1}} = \{\{D2\}, \{D3\}\}$.

The computation of $rc_\Gamma(D1)$ gives 13.14. As for the cost of $\Gamma_{\overline{D1}}$, its value is $11.33 + \frac{1}{3} \cdot 15 = 16.33$

given that *D2* has higher efficiency than *D3*, and that $rc_{\Gamma_{\overline{D1}}}(D2) = 11.33$ and $rc(D3) = 15$.

The resulting expected cost if we choose to repair *D1* is then:

$$c(D1) = 13.14 + \frac{3}{7} \cdot 16.33 = 20.14$$

Similarly, the expected costs for removing $D2$ and $D3$ are:

$$c(D2) = 23.57$$

$$c(D3) = 29.48$$

The estimated costs of the alternative choices are reported in Figure 3. We choose to observe *LT1*, whose expected cost of 16.89 is the lowest one. Let us now suppose that the outcome of $LT1$ is positive (i.e., $LT1Pos$); the candidate set Γ is updated to $\Gamma_P = \{\{D1.2\}\}$ and the fringe φ is updated to {*Sym1Pres*, *LT1Pos*, *LT2*}.

Second iteration. We need to estimate the costs of refining observations *LT1Pos* and *LT2*, and the cost of removing abducible $D1.2$. As shown in Figure 3, the best choice is to remove $D1.2$, with an expected cost of 6. In order to remove *D1.2*, we start removing its ground values *D1.2.1*, *D1.2.2* in that order (since they have the same cost and probability, they have the same efficiency, and thus the order is chosen arbitrarily). After removing *D1.2.1*, symptom *Sym1* is still present (i.e., *Sym1Pres* is still true), so we also remove *D1.2.2*. Now, *Sym1* disappears, and we conclude that the problem has been solved. Overall, the cost paid for this solution is 10: 2 for observing *LT1* and 8 for removing $D1.2$.

5. Experimental Evaluation

5.1. Implementation of the Method

We have implemented the proposed approach as a Perl program. The models consist in causal graphs $\mathcal{G}$ as specified in section 3; such graphs are stored in the file system in YAML format, and are loaded into appropriate memory data structures when needed.

A key part of the program, starting from the causal graph $\mathcal{G}$ of a model, generates the propositional theory $\mathcal{T}_E$ corresponding with the explanatory knowledge $\mathcal{K}_E$, as described in section 3.2. Such a propositional theory is further compiled into an OBDD (Ordered Binary Decision Diagram), denoted as $\mathcal{O}(\mathcal{T}_E)$. OBDDs are a special, canonical form for representing Boolean functions [3] that makes some important reasoning tasks[2] tractable, with a linear or even constant complexity. Due to these features, OBDDs have been successfully employed for knowledge compilation in several AI reasoning tasks, including planning [1] and diagnosis [5,19].

The implementation of the problem-solving algorithm shown in Figure 2 depends on the availability of an explanation function that, given a fringe φ, computes a set Γ of candidates that explain the observations in φ. Such a function is needed both to bootstrap the computation, and to update the current candidate set Γ after a new observation is made.

Our implementation of the function is based on suitable manipulations of OBDD $\mathcal{O}(\mathcal{T}_E)$. In particular:

1. we assert the truth of the fringe φ in $\mathcal{O}(\mathcal{T}_E)$; this operation can be done in linear time w.r.t. to the size of $\mathcal{O}(\mathcal{T}_E)$;
2. we assert the (negation of the) removed abducibles in $\mathcal{O}(\mathcal{T}_E)$ (also in linear time);
3. we extract explanations from the resulting OBDD by employing a well-known algorithm for extracting minimal models from an OBDD [9] whose complexity is exponential in the worst case, but usually tractable in practice; such an algorithm is slightly modified in order to enumerate the models that contain a minimal set (w.r.t. set inclusion) of true abducible variables $vals(A_1) \cup \ldots \cup vals(A_n)$, excluding uk_V variables (in order to eliminate non-least presumptive candidates).

Given this function, the implementation of the rest of the algorithm of Figure 2 was straightforwardly based on the contents of section 4.

5.2. Results of the Experiments

In order to empirically evaluate our approach, we ran a set of experiments. A main goal was comparing abductive problem solving performed by exploiting abstractions and abductive problem solving not relying on abstractions, i.e., the case

[2] Including consistency check, equivalence check, and most importantly enumeration of logical models.

Problem set	n_a	n_t	n_s	h	b_a	b_t	eb	N_a
SMALL	5	3	3	2	3	2	3	49.4
MEDIUM	8	3	3	2	4	2	6	104.8
LARGE	8	3	3	2-3	4	2	7	166.8

Table 1

Parameters for problem sets

Problem set	$to_{NOABSvsABS}$	$co_{ABSvsNOABS}$	$to_{ABSvsRND}$	$co_{RNDvsABS}$
SMALL	3.829	1.117	4.417	2.166
MEDIUM	8.265	1.121	1.789	3.354
LARGE	18.622	1.101	1.919	5.663

Table 2

Comparison with no abstractions and with random choices

where the reasoning process is restricted to formulate only ground hypotheses. Moreover, we evaluated the case where further lookahead in cost estimates is used, to get closer to the optimal choice. The different approaches were compared in terms of computation time and in terms of observation and action costs to solve the problem.

To this purpose, we implemented a generator of random models. Models are generated based on a number of parameters, including:

- n_a, n_t, n_s number of hierarchies of abducibles, tests and symptoms;
- h height of the hierarchies of abducibles and tests;
- b_a, b_t branching of the hierarchies of abducibles and tests;
- eb (explanation branching), number of abducibles explaining a symptom: a larger eb provides more candidate explanations.

Three sets, SMALL, MEDIUM and LARGE, of five models each, were generated, with different values for parameters, as from table 1, where N_a is the resulting average total number of nodes in the abducible hierarchies. Observation costs increase when going deeper in the hierarchies, with an average 50% increase from one level to the next one.

For each model, a set of 50 cases was generated randomly, based on the a-priori probabilities of abducibles; i.e., for each case, a set γ of ground abducibles is generated — and, since their probabilities are used, in a large fraction of cases, γ is a singleton (i.e., a single fault in diagnosis/troubleshooting). The observations μ to be explained for solving the case are the consequences of γ.

Table 2 compares the results of three methods:

- ABS is the method described in the paper;
- NOABS only uses ground hypotheses and actions;
- RND performs a random choice of the next observation or action (among the sets ρ and φ of relevant actions and observations).

The comparison is provided in terms of the average relative overhead of a method with respect to one another, in terms of computation time, and in terms of observation and action cost paid to actually solve the problem. For example, $to_{NOABSvsABS}$ provides the average relative time overhead of NOABS with respect to ABS, and we see that for the SMALL problems, the computation time of NOABS is almost 4 times with respect to ABS, while the overhead of ABS in terms of observation and action cost ($co_{ABSvsNOABS}$) is 11.7%.

We see that the additional cost of ABS with respect to NOABS is around 10% and does not increase with the size of models, while the running time of NOABS diverges with respect to the one for ABS. We also see that ABS has an acceptable additional running time (less than double, for MEDIUM and LARGE) with respect to choosing the next observation or action at random, while RND has, as it can be expected, unacceptable and diverging additional costs.

Table 3 reports results related to using, for the SMALL problem set, additional lookahead for estimating the best choice, i.e., trying to get closer to the optimal choice.

Column *to* provides the average relative overhead in time with respect to the ABS methods for variants, with lookahead 2, 3 and 4, of the basic ABS method (which uses lookahead 1). Column

	to	*co*
2	10.156	1.046
3	27.355	1.048
4	71.784	1.053

Table 3
Results for additional lookahead

co provides the average relative overhead in cost (for observations and actions) of the ABS method with respect to the additional lookahead methods. As we can see, running times increase significantly and only provide minor cost savings.

The experiments confirm that the approach in the paper provides acceptable additional observation/action costs, with respect to not using abstract hypotheses, with major savings on computation time. The experiments also illustrate that using further lookahead provides small savings while adding significant computational costs. As observed in the introduction, small or at least feasible computation time may mean that an action, even though possibly suboptimal, is taken before it is too late; in a specific setting, the cost of delaying actions might be measured in the same unit as observation and action costs.

6. Conclusions

In this paper, we proposed a novel abductive problem solving method which extends previous work on measurement selection in Model-Based Reasoning and on decision-theoretic troubleshooting. Unlike previous approaches to troubleshooting which do not exploit structured representations of the domain (e.g., [13,16]), our work is based on a representation with abstractions where both abstract observations and abstract hypotheses are taken into account.

We present a general abductive problem solving loop where, depending on the costs of observations and the costs of actions to be taken, a further observation may be chosen for discriminating or refining current candidates, or an action can be taken based on the current candidate(s). In this respect, the paper is also a significant generalization of previous works which use ontologies or taxonomies of hypotheses for explanation/interpretation purposes, but assume that all of the observations are given in advance [4,14,7,17,2] or confine actions to a second phase [20]. Interleaving observations and actions requires more sophisticated reasoning, but the increased flexibility in the way the problem is solved allows for better solutions to be found; in the diagnosis domain, this corresponds to the difference between (sequential) diagnosis and troubleshooting.

Costs of observations and actions may be very different at different levels of abstraction: there is a trade-off between paying the cost of further observations (or more precise observations) and the one of performing unnecessary actions, or unnecessarily general actions. Given that in practical cases computing an optimal choice is not feasible, we adopt a greedy, approximate approach from model-based diagnosis and decision-theoretic troubleshooting, basing the choice on expected costs.

The approach is aimed at being general, because its motivations can be found in several tasks and domains including technical and medical diagnosis as well as interpretation tasks such as plan recognition. Different instances may be derived with specific approaches for representing domain knowledge and for generating and updating candidate explanations based on observations.

Nevertheless, in the paper we have also defined the syntax and semantics of a specific knowledge representation formalism based on causal graphs. We have focused on such a representation for deriving an algorithm for the computation of explanations and for implementing the whole abductive problem solving loop. The experiments performed with the implemented system suggest that the use of abstraction results in a very limited overhead on observation/action costs, while the savings on computation time are major.

References

[1] P. Bertoli, A. Cimatti, M. Roveri, and P. Traverso. Planning in nondeterministic domains under partial observability via symbolic model checking. In *Proc. IJCAI*, pages 473–478, 2001.

[2] Ph. Besnard, M.-O. Cordier, and Y. Moinard. Ontology-based inference for causal explanation. In *Knowledge Science, Engineering and Management, 2nd Int. Conf., LNCS 4798*, pages 153–164, 2007.

[3] R. Bryant. Symbolic boolean manipulation with ordered binary-decision diagrams. *ACM Computing Surveys*, 24:293–318, 1992.

[4] B. Chu and J. Reggia. Modeling diagnosis at multiple levels of abstraction I and II. *International Journal of Intelligent Systems*, 6(6):617–671, 1991.

[5] A. Cimatti, C. Pecheur, and R. Cavada. Formal verification of diagnosability via symbolic model checking. In *Proc. IJCAI*, pages 363–369, 2003.

[6] P. Cohen, R. Schrag, E. Jones, A. Pease, B. Starr, and D. Gunning. The DARPA high-performance knowledge bases project. *AI Magazine*, 19(4), 1998.

[7] L. Console and D. Theseider Dupré. Abductive reasoning with abstraction axioms. In G. Lakemeyer and B. Nebel, editors, *Foundations of Knowledge Representation and Reasoning*, pages 98–112. Lecture Notes in Computer Science 810, Springer Verlag, 1994.

[8] L. Console, D. Theseider Dupré, and P. Torasso. On the relationship between abduction and deduction. *Journal of Logic and Computation*, 1(5):661–690, 1991.

[9] O. Coudert and J.-C. Madre. Implicit and incremental computation of primes and essential primes of boolean functions. In *Proc. 29^{th} ACM/IEEE Design Automation Conf.*, 1992.

[10] Gerhard Friedrich and Wolfgang Nejdl. Choosing observations and actions in model-based diagnosis/repair systems. In *Proc. KR 92*, pages 489–498, 1992.

[11] Y. Gil and J. Blythe. Planet: A shareable and reusable ontology for representing plans. In *AAAI 2000 Workshop on Representational Issues for Real-world Planning Systems*, 2000.

[12] W. Hamscher, L. Console, and J. de Kleer, editors. *Readings in Model-Based Diagnosis*. Morgan Kaufmann, 1992.

[13] D. Heckerman, J.S. Breese, and K. Rommelse. Decision-theoretic troubleshooting. *Communications of the ACM*, 38(3):49–56, 1995.

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

[15] Evelina Lamma, Paola Mello, Michela Milano, Rita Cucchiara, Marco Gavanelli, and Massimo Piccardi. Constraint propagation and value acquisition: Why we should do it interactively. In *IJCAI*, pages 468–477, 1999.

[16] H. Langseth and F. Jensen. Decision theoretic troubleshooting of coherent systems. *Reliability Engineering & System Safety*, 80(1):49–62, 2003.

[17] B. Neumann and R. Möller. On scene interpretation with description logics. In H.-H. Nagel and H. Christensen, editors, *Cognitive Vision Systems*, pages 247–275. Springer, 2006.

[18] D. Poole. Explanation, prediction: an architecture for default, abductive reasoning. *Computational Intelligence*, 5:97–110, 1989.

[19] P. Torasso and G. Torta. Model-based diagnosis through obdd compilation: a complexity analysis. *Lecture Notes in Computer Science*, 4155:287–305, 2006.

[20] G. Torta, D. Theseider Dupré, and L. Anselma. Hypothesis discrimination with abstractions based on observation and action costs. In *Proc. Int. Work. on Principles of Diagnosis*, pages 189–196, 2008.

[21] A. Valente, T. Russ, R. MacGregor, and W. Swartout. Building and (re)using an ontology of air campaign planning. *IEEE Intelligent Systems*, 14(1):27–36, 1999.

[22] H. Warnquist and M. Nyberg. A heuristic for near-optimal troubleshooting using AO*. In *Proc. Int. Work. on Principles of Diagnosis*, pages 197–204, 2008.

[23] J. Zhang, D. Caragea, and V. Honavar. Learning ontology-aware classifiers. *LNCS*, 3735:308–321, 2005.

[24] J. Zhang, A. Silvescu, and V. Honavar. Ontology-driven induction of decision trees at multiple levels of abstraction. *LNCS*, 2371:316–323, 2002.