Abductive Problem Solving with Abstractions

Abductive Problem Solving with Abstractions

Gianluca Torta[1] and **Daniele Theseider Dupré**[2]
(1) Università di Torino, Italy, email: torta@di.unito.it
(2) Università del Piemonte Orientale, Italy, email: dtd@mfn.unipmn.it

Abstract

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. In some cases actions are the only or the main way for discriminating among alternative explanations. Some proposals address the problem based on a task-independent representation of a domain which includes an ontology or taxonomy of hypotheses and actions. In this paper we rely on the same type of representation, and we point out the role of abstractions in an iterative process where, like in model-based diagnosis and troubleshooting, further observations or actions are proposed to achieve the overall goal of discriminating among candidate hypotheses and, more importantly, performing the appropriate actions for the case at hand. Discrimination is performed up to an appropriate level which depends on the cost of actions (e.g. repair actions or therapy) to be taken based on the results of abduction, and on the cost of additional observations, which should be balanced with the benefits, in terms of more suitable actions, of better discrimination. Abstractions have a significant impact on this trade-off, given that the cost of observing the same phenomenon at different levels of abstraction may be quite different, and choosing a generic action, without information on which specific instance is most appropriate, is, in general, suboptimal.

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 (Chu and Reggia 1991; Console and Theseider Dupré 1994; Kautz 1991), including recent ones (Besnard, Cordier, and Moinard 2007; Neumann and Möller 2006), address the problem based on a representation of a domain which includes an ontology or taxonomy of hypotheses.

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. Ontologies have also been proposed as the basis for large knowledge bases to be used for problem solving that includes planning (Valente et al. 1999; Gil and Blythe 2000), but, as noted in (Cohen et al. 1998), they should be shared among different problem solvers for related tasks; therefore, they should be developed independently of the reasoning task[1]: i.e., the structure should reflect a natural representation of the domain, but it should not necessarily provide directly the best structure for diagnosis, interpretation, or planning and acting.

In this paper we adopt a similar representation, and concentrate on the following issues:

- Dealing with abduction as an iterative process where further observations, as in model-based diagnosis (Hamscher, Console, and de Kleer 1992), or actions (e.g. substituting a suspect component in the system), as in *troubleshooting* (Heckerman, Breese, and Rommelse 1995), can be proposed to the intermediate goal of discriminating among candidate explanations and the ultimate goal of performing actions that are appropriate for the case at hand.
- Balancing the costs of observations with (reduced) costs or (increased) benefits, in terms of actions, of the results of abduction.

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 hypotheses; in a plan recognition or in an interpretation task, the human or software agent using the results should similarly achieve some benefit from a better discrimination of hypotheses or from more specific hypotheses, leading to a more focused action: this could either imply a reduced cost — e.g. if hypotheses are threats to the agent with costly defense actions — or an increased benefit — e.g. if the agent might use the results to earn money. In all settings, we intend that some action has to be taken based, in general, on the remaining candidate hypotheses. If the set of candidates is too broad or too abstract, the agent may 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, e.g., in

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

component-oriented diagnosis, replacing an entire subsystem;

- more actions than necessary are taken, e.g. actions are performed in an order based on their cost and the probability of the related explanation, but they can be the wrong action (repairing the wrong part, taking the wrong therapy, defending from the wrong threat); and similarly, in the *benefit* case, the action may only *probably* (but not certainly) be the right one for achieving the benefit.

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 abstractions has a significant impact on this trade-off. The cost of observing the same phenomenon at different levels of abstraction may 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 (Lamma et al. 1999).

In several settings, an observation which is itself expensive, because it consumes resources and time to be performed, may imply 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.

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.

By explicitly considering abstractions in the iterative abduction process, we expect to reduce the observation and action costs significantly, yet maintaining the ability to exploit detailed observations and knowledge when convenient (similar advantages have been shown in inductive classification with abstractions, e.g. (Zhang, Silvescu, and Honavar 2002)).

In the following, we first describe the knowledge we expect to be available. We then describe a basic iterative abductive problem solving loop and we concentrate on the execution of abstract tasks 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.

Domain Representation

The basic elements of the domain model are a set of tasks $\mathcal{T} = \{T_1, \ldots, T_l\}$, a set of abducibles (atomic assumptions) $\mathcal{A} = \{A_1, \ldots, A_n\}$ and a set of manifestations $\mathcal{M} = \{M_1, \ldots, M_m\}$.

Each task T_h is associated with an IS-A hierarchy $\Lambda(T_h)$ containing abstract values of T_h as well as their refinements at multiple levels; similarly, each abducible A_i is associated with an IS-A hierarchy $\Lambda(A_i)$ and 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 $\Lambda(T_h)$, $\Lambda(A_i)$ or $\Lambda(M_j)$) are mutually exclusive, i.e. the Λ hierarchies are trees; moreover, in a given situation, exactly one ground instance 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 overall goal of our problem solving process is to remove all of the abducibles which are present in a given situation at an (approximately) minimum cost.

The actions set $vals(T_h)$ of a task T_h is the set of all of the elements θ drawn from the hierarchy $\Lambda(T_h)$, while $gndvals(T_h)$ is the subset of $vals(T_h)$ containing only ground actions t, i.e. the leaves of hierarchy $\Lambda(T_h)$. The definition of set $vals$ (resp. $gndvals$) can be extended to a set of tasks by taking the union of the $vals$ (resp. $gndvals$) of each task in the set; we also define set $vals$ (resp. $gndvals$) for an action θ belonging to the hierarchy $\Lambda(T_h)$ by considering only the values (resp. ground values) belonging to the sub-hierarchy of $\Lambda(T_h)$ rooted in θ.

Sets $vals$ and $gndvals$ are defined for abducibles A_i and manifestations M_j in the same way as for tasks.

The hypotheses space $\mathcal{S}(\mathcal{A})$ is the set of all of the combinations γ of values drawn from one 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 μ of values drawn from one or more distinct hierarchies $\Lambda(M_j)$.

The relationship between tasks and abducibles is defined by the remove/repair domain knowledge, which we assume to be a relation $\mathcal{K}_R \subseteq vals(\mathcal{T}) \times vals(\mathcal{A})$. Given an element $\theta \in vals(\mathcal{T})$ and an element $\alpha \in vals(\mathcal{A})$, the fact that $(\theta, \alpha) \in \mathcal{K}_R$ means that θ is an appropriate action to be taken to remove α, e.g. an action that repairs a fault α. Note that, according to our definition of $\mathcal{K}_R$, the relation between the status of the world (represented by the abducibles) and the actions we can take (represented by the tasks) is simpler than in general planning problems, where actions can have complex pre-conditions and post-conditions.

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 a combination of instances of manifestations $\mu \in \mathcal{S}(\mathcal{M})$ and a combination of instances of abducibles $\gamma \in \mathcal{S}(\mathcal{A})$, the fact that $(\gamma, \mu) \in \mathcal{K}_E$ means that μ is a possible observation set corresponding to hypotheses set γ (and, conversely, that γ is a possible explanation for μ).

Our definition of $\mathcal{K}_E$ and $\mathcal{K}_R$ as relations is not intended to imply that such relations should be represented extensionally and that the reasoning algorithms should directly manipulate such an extensional representation. $\mathcal{K}_E$ may be represented intensionally with a multi-valued propositional or causal model (as it is the case in our illustrative example of Figure 1) whose semantics correspond to the extensional enumeration of the tuples in $\mathcal{K}_E$. Reasoning involving $\mathcal{K}_E$ may take place at the syntactic level, e.g. as propositional or causal inference.

For each leaf value a of an abducible A, an a-priori probability $p(a)$ is given. Instead, we associate costs with the values of tasks and manifestations.

The cost of an action may, in general, depend on the current status of the world, however, in this paper we assume that for each leaf value t of a task, a cost $tc(t)$ is assigned, independently of the current hypotheses. The cost $tc(\theta)$ of an abstract task value θ is either associated with θ by the user (based, e.g. on experience) or estimated automatically as the cost of the approximately optimal *sequence* of executions of ground tasks $(t_1, \ldots, t_q)$ needed to execute θ, as explained in the following sections.

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

In Figure 1 we show a fragment of a fictitious medical domain model. On the left, there is the nosological description of some diseases, represented as three IS-A hierarchies of abducibles (with roots $Disease1$, $Disease2$, and $Disease3$). For example, $Disease1.1$ and $Disease1.2$ are two refinements of $Disease1$. On the right, there are possible symptoms and the possible medical examinations to be performed, represented as three IS-A hierarchies of manifestations (with roots $Symptom1$, $LabTest1$, and $LabTest2$). The a-priori probabilities of the leaves of abducibles is assumed to be $\frac{1}{2^8}$, except for $p(Disease1.1) = \frac{1}{2^7}$.

Tasks associated with abducibles are not drawn explicitly; in this example, exactly one action is associated with each abducible, and its cost tc (the cost of treating the disease) is given in Figure 1. Observation costs oc associated with each internal node of manifestation hierarchies are the costs of performing the related laboratory exam. The relationships between abducibles and manifestations are represented by rightwards dashed arrows. For example, $Disease1$ (and all its more specific diseases) implies $LabTest2$ to be positive; $Disease1.2$ implies $LabTest1$ to be positive, and its refinements $Disease1.2.1$ and $Disease1.2.2$ imply more specific positive values of $LabTest1$.
The relation $\mathcal{K}_E$ completes such explicit knowledge with negative values of tests as default values, e.g. the hypotheses set $\{Disease3\}$ predicts $\{Symptom1, LabTest1Neg, LabTest2Neg\}$, and when refinements of hypotheses do not predict refinements of observations, they are intended as compatible with all refinements.

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

φ := $\hat{\varphi}$
generate a set Γ of candidates γ which explain $\hat{\varphi}$
generate a set Θ of tasks θ relevant for Γ
loop
  σ := ChooseNextStep(Γ, Θ);
  if σ = STOP **then exit**
  if $\sigma = \omega \in vals(\mathcal{M})$
    φ := Observe(ω)
    Γ := UpdateWithObs(Γ, φ, ω)
  elseif $\sigma = \theta \in vals(\mathcal{T})$
    φ := Execute(θ)
    Γ := UpdateWithAct(Γ, φ, θ)
  endif
  generate a set Θ of tasks θ relevant for Γ
end

Figure 2: Main loop of the abductive problem solving algorithm.

The Abductive Problem Solving Process

The algorithm shown in Figure 2 illustrates the overall approach to abductive problem solving we take into 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. In particular, if the current value in φ of a manifestation M is ω, and we make an observation which refines ω into one of its children ω_k, ω is replaced by ω_k as the value of M in φ.
Also when we perform an action t, we may need to substitute the value ω of M with another value, as detailed in the next section.

Given the set of initial observations $\hat{\varphi}$, a set of candidate explanations Γ and a set of tasks Θ relevant for solving the candidates in Γ are generated.
In particular, a task θ is relevant for Γ if there exists an atomic hypothesis $\alpha_{i,j}$ of some candidate $\gamma_i = \{\alpha_{i,1}, \ldots, \alpha_{i,r_i}\} \in \Gamma$ s.t. task θ is related to $\alpha_{i,j}$ by the domain knowledge $\mathcal{K}_R$ (i.e. $(\theta, \alpha_{i,j}) \in \mathcal{K}_R$).

At each iteration of the main loop, we select what to do next based on the current candidate set Γ and the set of relevant tasks Θ[2]. Clearly, this choice is in general suboptimal, due to the prohibitive complexity of making an optimal

[2]The choice may also need to take into account the sequence Σ of observations/actions performed so far, e.g. in order to ignore actions that have already been performed (Warnquist and Nyberg 2008). For the sake of clarity, in the following we do not explicitly discuss the role of Σ.

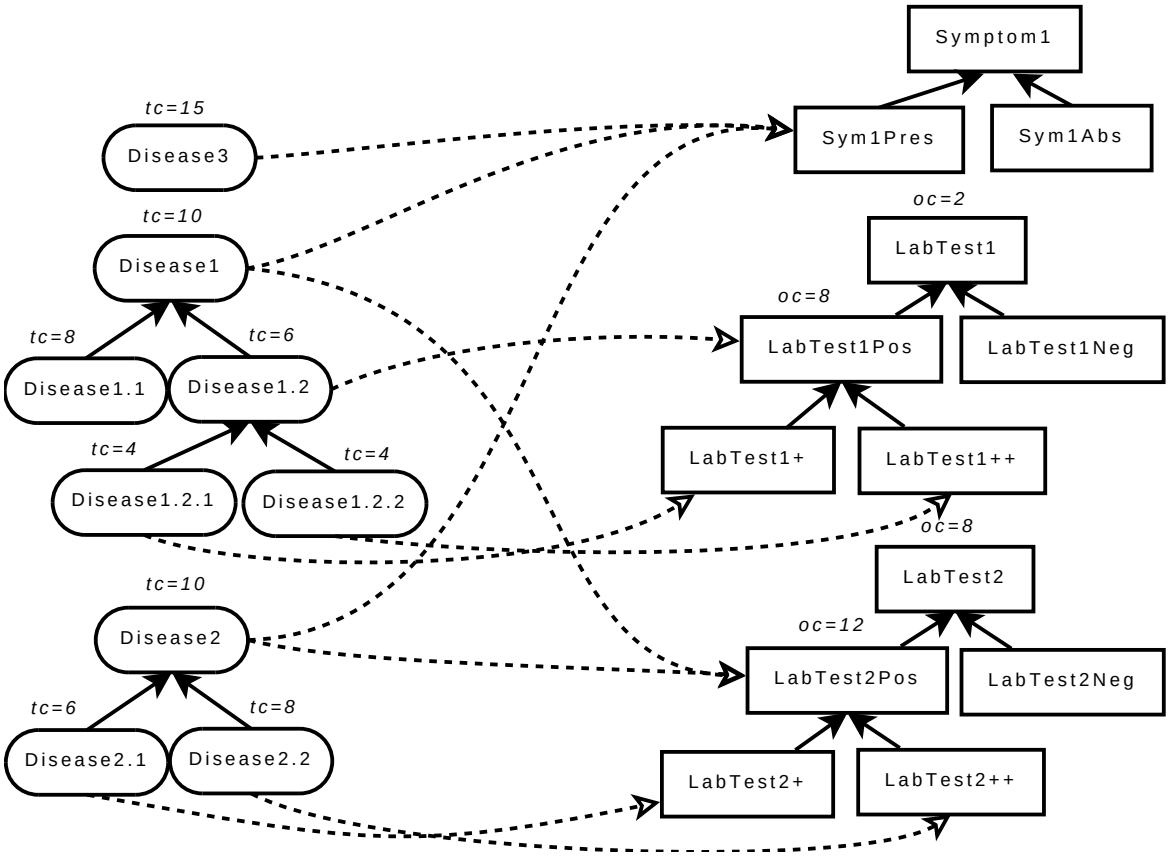

Figure 1: A (fictitious) medical domain model. The ellipses on the left represent the abducibles, i.e., the possible diseases, arranged in IS-A hierarchies; the rectangles on the right represent the manifestations, i.e., the possible symptoms and laboratory exams, arranged in IS-A hierarchies. The rightwards dashed arrows represent the relationships between abducibles and manifestations. The tasks (therapies), not drawn in the figure, have hierarchies isomorphic to the hierarchies of abducibles, and costs *tc* shown above the diseases they repair.

choice.

When the problem is solved (or there are no useful observations/actions to make), the algorithm stops. Otherwise, if the choice is to perform an observation, the candidate set (and manifestations fringe) must be updated according to its outcome, while if the choice is to perform a task, the candidate set must be updated according to the effects of t on the abducibles and, possibly, on the manifestations fringe (as explained in the next section).

The goal of this paper is to provide a general approach to the selection of the next step.

We do not embrace a specific semantics of abduction and implementation of the candidate generation and update steps in the loop. Their concrete definition would depend on several issues; in particular, it could involve a mix of abductive and consistency reasoning depending on the completeness and predictiveness of knowledge (Console and Torasso 1991; Theseider Dupré 2000). Its formulation would also depend on the way $\mathcal{K}_E$ and $\mathcal{K}_R$ are represented — in particular, what is explicitly represented and what is implicitly intended.

Predictiveness of knowledge, and, in particular, whether a set of abducibles implies a single value for an observation or not (so that requiring abductive explanations which imply the observed value either makes sense or is too demanding) is particularly relevant with hierarchies of abducibles and observations. In general, we should accept an explanation that implies some abstraction of the observation, rather than the observation itself (Kautz 1991; Besnard, Cordier, and Moinard 2007), but we do not explicitly address this issue in this paper.

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

Independent of the way explanations are defined and computed, a general criterion which is suitable in this setting is the preference for *least presumptive* explanations (Poole 1989), which generalize minimal (wrt set inclusion) explanations: an explanation γ that is a superset of another explanation γ' or that, based on the IS-A hierarchies $\Lambda(A_i)$, is an instance of another explanation γ'', is not least presumptive. In the following we assume that the candidates Γ computed

at each iteration represent the least presumptive explanations of the observations made so far, i.e. we do not commit to a detailed explanation when a more abstract one can explain the observations.

We also assume that the relevant tasks in Θ are least presumptive, in the following sense: if two tasks θ and θ' are relevant for the same set of atomic hypotheses $\alpha_1, \ldots, \alpha_k$ which appear in candidates belonging to Γ, and θ *ISA* θ', then θ is not least presumptive. Therefore, we do not commit to a specific action when a more abstract one has the same effects, so that we can delay the choice of how the abstract action (which cannot be directly executed) will actually be instantiated.

Task Execution

Let us start by considering the execution of a ground task t when the candidate set is Γ. How does Γ look like, after executing t? Given a (ground or abstract) task θ, we denote with $trg(\theta)$ the set $\{\alpha : (\theta, \alpha) \in \mathcal{K}_R\}$. Executing t makes all of the abducible values in $trg(t)$ false, therefore each candidate $\gamma \in \Gamma$ must be updated as follows:

$$\gamma' = \gamma \backslash trg(t)$$

resulting in an updated candidate set Γ'. Note that, if a candidate γ'_i obtained from $\gamma_i \in \Gamma$ through the execution of t makes another candidate $\gamma'_j \in \Gamma'$ a superset or an instance of γ'_i, the candidate γ_j must *not* be removed from Γ'. This is because γ'_j is still least presumptive in explaining the evidence we have collected in the previous steps, while we have no evidence yet on the actual effects of t on the values of the manifestations; if, however, we can immediately obtain evidence about the values of manifestations *after* the execution of t s.t. γ'_j is inconsistent with this new evidence, then γ'_j must be removed (see below).

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 sequence $\mu_i = (\omega_{i,1}, \ldots, \omega_{i,m})$.

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 $LUB(\Gamma', M_j)$ the value in $\Lambda(M_j)$ that is the least upper bound of $\{LUB(\gamma', M_j) : \gamma' \in \Gamma'\}$ (i.e. of the set of values for M_j predicted by Γ'). The new fringe predicted by Γ' will therefore be $\varphi(\Gamma') = (LUB(\Gamma', M_1), \ldots, LUB(\Gamma', M_m))$.

From the discussion made so far, the execution of t would make the candidate set become Γ' and the fringe become $\varphi(\Gamma')$; note that $\varphi(\Gamma')$ 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 Γ'.

For this reason, (as anticipated above) 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 execution of a task t.

In particular, following (Zhang, Caragea, and Honavar 2005), we define a *cut* $\mathcal{C}(M)$ on a hierarchy $\Lambda(M)$ to be a set of values $\omega \in vals(M)$ s.t. 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 illustrative example, manifestation $Symptom1$ is associated with the cut $\{Sym1Pres, Sym1Abs\}$, i.e. after performing a task we know for free whether the symptom persists ($\{Sym1Pres\}$) or it has disappeared ($\{Sym1Abs\}$). For the other manifestations $LabTest1$ and $LabTest2$, 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, $\{LabTest1Pos, LabTest1Neg\}$ would be a valid cut for $LabTest1$, although $LabTest1Pos$ is an abstract value, while $LabTest1Neg$ 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 $LUB(\Gamma', M_j)$ and it will therefore be included in the new fringe. As a consequence, some candidates γ' may become inconsistent with the refined fringe.

We denote with Γ^* the candidate set resulting from removing such inconsistent candidates, and with $\varphi(\Gamma^*)$ the refined fringe.

The contents of Γ^* depend on the outcome of the checks made at no cost on manifestations $\mathcal{M}^*$ after executing t. We denote with $PW(\Gamma, t)$ (for *possible worlds*) the set of all of the possible candidate sets Γ^* that may result by executing t and then checking manifestations $\mathcal{M}^*$. The choice of the next step will be based on $PW(\Gamma, t)$, as described in the next section. In general, the number of possible outcomes of checking manifestations $\mathcal{M}^*$ can be exponential in $|\mathcal{M}^*|$, and therefore $PW(\Gamma, t)$ 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 $\gamma' = \gamma \backslash trg(t)$ 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 $|PW(\Gamma, t)|$ is bounded by the number of candidates $|\Gamma|$.

Let us now consider the execution of an abstract task θ. Since θ, by being abstract, is not directly executable, we need to perform a *sequence* of ground tasks $t \in gndval(\theta)$ so that we ensure that all of the abducible values $\alpha \in trg(\theta)$ have been removed (i.e. have become *false*) in the candidates Γ.

The algorithm starts by selecting an approximately best ground task $t \in gndvals(\theta)$ to execute. After the execution of task t, we end up in one of the possible worlds $\Gamma_i \in PW(\Gamma, t)$.

If Γ_i does not contain any $\alpha \in trg(\theta)$, the execution of θ is complete, and the algorithm continues with the next iteration of the main loop of observations/actions, or it stops. Otherwise, a new approximately best ground task t' is selected,

based on the candidate set Γ_i, and so on.

The selection of the best task to execute is based on a slight modification of the *efficiency* measure defined in (Heckerman, Breese, and Rommelse 1995). In particular, let us denote with Γ_α the subset of Γ whose candidates contain the abducible value α (i.e. $\Gamma_\alpha = \{\gamma \in \Gamma : \gamma = \{\alpha\} \cup \rho\}$) and let Γ_t be the subset of Γ whose candidates contain at least one abducible value affected by t (i.e. $\Gamma_t = \bigcup_{\alpha \in trg(t)} \Gamma_\alpha$); the efficiency of a ground task t is defined as:

$$ef(t) = \frac{p(\Gamma_t|\Gamma)}{tc(t)}$$

Intuitively, the task efficiency is increased by the probability that it will actually remove some abducible value in the candidate set and it is decreased by its cost. The task t to be executed next is the one with the highest efficiency.

Independently of which sequence of ground tasks $(t_1, \ldots, t_q)$ is actually executed, we end up with one of the candidate sets $PW(\Gamma, \theta)$, as in the case of the execution of a single ground task t.

Choosing the Next Step

We assume that the problem solving process stops only when there are no useful actions or observations to be made. In particular, let ω be an observation in φ and $\Gamma_1, \ldots, \Gamma_q$ be the possible candidate sets that would result by observing ω and getting values $\omega_1, \ldots, \omega_q$ respectively. Then, ω is *useful* if $\Gamma_k \neq \Gamma$ for at least one $k \in \{1, \ldots, q\}$, i.e. if at least one possible outcome of observing ω makes the candidate set change. We denote with $\Phi \subseteq \varphi$ the set of useful observations.

The stop criterion for our algorithm can therefore be formulated more precisely as: $\Phi = \emptyset$ (no useful observations) and $\Theta = \emptyset$ (no relevant tasks). Note that, in case our model does not specify an action for some of the abducible values, we may need to stop even if not all of the causes of manifestations have been removed.

Let us now assume that the stop criterion has not been met yet. For each $\omega \in \Phi$ (i.e. useful observation), 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.:

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

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 $\theta \in \Theta$, we evaluate the estimated cost $c(\theta)$, which is the sum of the cost $tc(\theta)$ of executing the task θ and the expected cost of the candidate set after executing θ, i.e.:

$$c(\theta) = tc(\theta) + \sum_{\Gamma_k \in PW(\Gamma,\theta)} p(\Gamma_k|\Gamma) \cdot c(\Gamma_k) \qquad (2)$$

where $PW(\Gamma, \theta)$ is the set of possible candidate sets resulting from the execution of θ (as explained in the previous section) and $p(\Gamma_k|\Gamma)$ is the probability that the actual candidate set after executing θ is Γ_k.

In both equations 1 and 2, 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 adapt to our setting a simple technique from the troubleshooting literature, namely the *greedy* approach of (Langseth and Jensen 2003). In particular, we start by computing Θ_k, i.e. the set of tasks relevant for Γ_k; then we order the tasks $\theta \in \Theta_k$ in decreasing efficiency order using the following formula, which was introduced in the previous section for ground tasks but can be straightforwardly applied also to abstract tasks:

$$ef(\theta) = \frac{p(\Gamma_\theta|\Gamma_k)}{tc(\theta)}$$

where $\Gamma_\theta = \bigcup_{\alpha \in trg(\theta)} \Gamma_\alpha$.

Let $\hat{\Theta}_k = (\theta_1, \ldots, \theta_q)$ be the sequence of tasks obtained in this way. The cost of Γ_k is computed as follows:

$$c(\Gamma_k) = \sum_{i=1}^{q} tc(\theta_i) \cdot p(\Gamma_k^i \neq \{\emptyset\})$$

where Γ_k^i is the candidate set *after* tasks $\theta_1, \ldots, \theta_{i-1}$ have been executed starting from candidate set Γ_k.

Note that the cost of each task θ_i is weighted with the probability that the task 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 $\emptyset$.

The exact way $p(\Gamma_k^i \neq \{\emptyset\})$ is computed depends on the set of manifestations $\mathcal{M}^*$ that can be checked at no cost after the execution of each task, and on the cuts associated with such checks. If we assume that, after each task execution, it is possible to check at no cost whether the problem has been solved or not, and let $trg^i(\hat{\Theta}) = \bigcup_{j=1,\ldots,i} trg(\theta_j)$, then:

$$p(\Gamma_k^i \neq \emptyset) = p(\{\gamma \in \Gamma_k : \gamma \not\subseteq trg^{i-1}(\hat{\Theta})\})$$

Indeed, if the real world status is a candidate $\gamma' \in \Gamma_k$ which is completely removed by executing $\theta_1, \ldots, \theta_{i-1}$, we must be aware (through our checks) that the problem is solved, and Γ_k^i must therefore be equal to $\{\emptyset\}$; so, $p(\Gamma_k^i \neq \{\emptyset\})$ is the probability that γ' is one of the candidates in Γ_k which are *not* completely removed by $\theta_1, \ldots, \theta_{i-1}$, i.e. γ' is not a subset of $trg^{i-1}(\hat{\Theta})$.

After we have computed the expected observation costs $c(\omega)$ and expected action costs $c(\theta)$, we perform the observation or action σ s.t.:

$$\sigma = argmin_{\hat{\sigma} \in (\Phi \cup \Theta)} [c(\hat{\sigma})]$$

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

Example

Let us consider the execution of the algorithm on the example in Figure 1.

Initial observations. Let us suppose that an initial manifestation of $Symptom1$ is detected, i.e., $\varphi_{in} = \{Sym1Pres, LabTest1, LabTest2\}$. The initial candidate set is $\Gamma = \{\{Disease1\}, \{Disease2\}, \{Disease3\}\}$, representing the possible alternative diagnoses (in fact, $Disease1$, $Disease2$ and $Disease3$ explain $Sym1Pres$).

First iteration. The relevant tasks are, of course, $\{T1, T2, T3\}$ (i.e. the abstract tasks associated, respectively, with $Disease1$, $Disease2$ and $Disease3$), and the useful observations are $LabTest1$ and $LabTest2$.

The costs are computed as follows.

Regarding the observation $\omega = LabTest1$, two outcomes are possible: the test is either negative ($LabTest1Neg$) or positive ($LabTest1Pos$). For evaluating the candidate sets Γ_1 and Γ_2 resulting from the observation of $LabTest1$, we adopt an approach similar to (Console, Theseider Dupré, and Torasso 1990). In particular, if the outcome is $LabTest1Neg$, it will be possible to exclude $Disease1.2$, because this abducible explains $LabTest1Pos$, which is incompatible with $LabTest1Neg$; therefore:

$$\Gamma_1 = \{\{Disease1.1\}, \{Disease2\}, \{Disease3\}\}$$

On the other hand, if the outcome is $LabTest1Pos$, the minimal candidate set is:

$$\Gamma_2 = \{\{Disease1.2\}\}$$

only, which, since it explains $LabTest1Pos$, will be part of every candidate set (see (Console, Theseider Dupré, and Torasso 1990)). The probabilities of Γ_1 and Γ_2 are $\frac{5}{7}$ and $\frac{2}{7}$ respectively. The total expected cost associated with $LabTest1$ is therefore:

$$c(LabTest1) = 2 + \frac{5}{7} \cdot c(\Gamma_1) + \frac{2}{7} \cdot c(\Gamma_2) = 15.86$$

(the immediate cost of performing $LabTest1$ is 2).
Estimated cost $c(\Gamma_1)$ is computed based on the fact that the relative probabilities of the candidates are 2,2,1 and the corresponding actions are $T1.1, T2, T3$, in order of decreasing efficiency, and therefore:

$$c(\Gamma_1) = 8 + \frac{3}{5} \cdot 10 + \frac{1}{5} \cdot 15 = 17$$

while $c(\Gamma_2) = 6$ (the cost of performing action $T1.2$ is 6).

Regarding the observation $\omega = LabTest2$, if the outcome of this observation is negative ($LabTest2Neg$), then the resulting candidate set is $\Gamma'_1 = \{\{Disease3\}\}$. On the other hand, if the outcome is positive ($LabTest2Pos$), the candidate set is $\Gamma'_2 = \{\{Disease1\}, \{Disease2\}\}$, in this case expected costs are:

$$c(\Gamma'_1) = 15$$
$$c(\Gamma'_2) = 13.33$$

and then:

$$c(LabTest2) = 8 + \frac{1}{7} \cdot 15 + \frac{6}{7} \cdot 13.33 = 21.57$$

since the immediate cost of performing $LabTest2$ is 8 and the probabilities of Γ'_1, Γ'_2 are, respectively, $\frac{1}{7}$ and $\frac{6}{7}$.

The expected cost associated with action $\theta = T1$ is as follows. If $\{Disease1\}$ (having probability $\frac{4}{7}$) is the correct candidate, $T1$ solves the problem (which is detected because $Symptom1$ becomes absent), otherwise the remaining candidates are $\{\{Disease2\}, \{Disease3\}\}$, and in this case the expected cost is $10 + \frac{1}{3} \cdot 15 = 15$ given that $T2$ has higher efficiency than $T3$. The resulting expected cost if $T1$ is chosen is then:

$$c(T1) = 10 + \frac{3}{7} \cdot 15 = 16.43$$

since the cost of performing $T1$ is 10, and the probability that this does not solve the problem is $1 - \frac{4}{7} = \frac{3}{7}$

Similarly, the expected costs for $T2$ and $T3$ are:

$$c(T2) = 19.29$$

$$c(T3) = 26.43$$

The result of the selection step is then $LabTest1$, which has an expected cost of 15.86.

Second iteration.

Let us suppose that the outcome of $LabTest1$ is negative (i.e., $LabTest1Neg$). Then, the algorithm performs a further iteration with

$$\Gamma := \Gamma_1 = \{\{Disease1.1\}, \{Disease2\}, \{Disease3\}\}$$

The options, now, are the tasks $\{T1.1, T2, T3\}$ and the observation $LabTest2$. The expected costs are as follows:

$$c(LabTest2) = 21.4$$
$$c(T1.1) = 17$$
$$c(T2) = 17.8$$
$$c(T3) = 25.4$$

Note that the smallest expected cost does not decrease wrt the previous iteration, because the observation eliminated $Disease1.2$ which was relatively cheap to treat - had the result of the observation been $LabTest1Pos$, which had probability $\frac{2}{7}$, the remaining expected cost would have been 6. This time $T1.1$ is selected; we suppose that it solves the problem, i.e. it makes $Symptom1$ become absent and the process stops.

Conclusions

In this paper, which generalizes (Torta, Theseider Dupré, and Anselma 2008), we proposed an approach to selecting the next step in an abductive problem solving loop which extends previous work on measurement selection in Model-Based Reasoning and on decision-theoretic troubleshooting. In fact, our work is based on a representation with abstractions where both abstract hypotheses and abstract tasks 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). Costs of observations and actions may be very different at different levels of abstraction: there is a trade-off between paying the cost of more 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. We expect that abstractions contribute to making the approach feasible, with a small number of abstract alternatives to be evaluated at each step.

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.

References

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

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

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

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

Console, L., and Torasso, P. 1991. A spectrum of logical definitions of model-based diagnosis. *Computational Intelligence* 7(3):133–141.

Console, L.; Theseider Dupré, D.; and Torasso, P. 1990. Introducing test theory into abductive diagnosis. In *Proc. 10th Int. Work. on Expert Systems and Their Applications (Conf. on 2nd Generation Expert Systems)*, 111–124.

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

Hamscher, W.; Console, L.; and de Kleer, J., eds. 1992. *Readings in Model-Based Diagnosis*. Morgan Kaufmann.

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

Kautz, H. 1991. 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.

Lamma, E.; Mello, P.; Milano, M.; Cucchiara, R.; Gavanelli, M.; and Piccardi, M. 1999. Constraint propagation and value acquisition: Why we should do it interactively. In *IJCAI*, 468–477.

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

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

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

Theseider Dupré, D. 2000. Abductive and consistency-based diagnosis revisited: a modeling perspective. In *Proc. 8th Int. Non-Monotonic Reasoning Workshop*.

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

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

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

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

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