Reduction of Diagnostic Complexity through Model Abstractions*

Igor Mozetič
Austrian Research Institute for Artificial Intelligence
Schottengasse 3, A-1010 Vienna
Austria
igor@ai.univie.ac.at

Abstract

The paper presents an application of abstractions to model-based reasoning with the goal to improve the efficiency of diagnosis. We define three abstraction operators which map a detailed, complex model to an abstract, simpler one. Our approach is not restricted to discrete, qualitative models. We define a provably correct and complete diagnostic algorithm and analyse its complexity. Abstractions and the algorithm were applied to a complex medical problem: ECG interpretation based on the model of the heart's electrical activity. Results show that hierarchical diagnosis has logarithmic time complexity when compared to a one-level diagnosis — in particular, a speedup of a factor of 20 was achieved.

1 Introduction

There are two fundamentaly different approaches to diagnostic reasoning. In the first, heuristic approach, one encodes diagnostic rules of thumb and experience of human experts in a given domain. In the second, model-based approach, one starts with a model of a real-world system which explicitly represents the structure and components of the system (e.g., de Kleer 1976, Genesereth 1984, Reiter 1987). When the system's actual behavior is different from the expected behavior, the diagnostic problem arises. The model is then used to identify components and their internal states which account for the observed behavior.

From a formal viewpoint, a model-based diagnosis falls between the extremes of abductive and consistency-based approaches (Poole 1989). The main difference is, in abductive approach the diagnoses imply the observations, while in the consistency-based approach the observations imply the diagnoses. Our approach can be characterized as *deductive*

*Appears in *Working notes First Intl. Workshop on Principles of Diagnosis*, pp. 102-111, Stanford University, Palo Alto, CA, July 23-25, 1990.

diagnosis since a model implies pairs state-observation. There is no distinction between normal and abnormal states of components, and a model defines a mapping from any internal state to external observations. The diagnostic problem is then to find the inverse mapping, i.e., all internal states for the given observation. However, due to the 'forward' directionality bias of the model (from states to observations), its 'backward' application might be inefficient, and abstractions are introduced to improve the efficiency.

In section 2 we define a model representation formalism and illustrate it on a numeric OR gate model. In section 3 we define three abstraction operators which map a detailed, complex model to an abstract, simpler one. Our approach is related to abstractions in theorem proving (Giunchiglia & Welsh 1989, Plaisted 1981) and subsumes ABSTRIPS abstractions in planning (Sacerdoti 1974). Abstraction operators are applied to a detailed, quantitative model of an OR gate from which three successively more abstract qualitative models are automatically derived.

In section 4, a hierarchical diagnostic algorithm is defined. The algorithm uses an abstract model to generate potential diagnoses, and a more detailed model to verify them. Since the models are always used in the 'forward' direction, for simulation, the algorithm is suitable for integrating numerical and qualitative models (e.g., Gallanti et al. 1989). With appropriate multi-level abstractions, the complexity of diagnosis may be reduced from $O(S)$ to $O(\log S)$, where S is the number of states to be verified at the detailed level. A similar complexity reduction using abstractions is reported by Genesereth (1984) and Korf (1987, in planning), but without any experimental evidence to support the claim.

An application of abstractions to a complex medical problem, described in section 5, confirms the expected complexity reduction. The problem—originating from the KARDIO project (Bratko, Mozetic & Lavrac 1989) is to diagnose heart disorders, given an ECG and a simulation model of the heart's electrical activity. Until now, all attempts to *directly* use the model for efficient diagnosis failed—in an average case more than $50sec.$ is needed to find all diagnoses. By abstractions and refinements, the model was represented at four levels of detail, and the average diagnostic time was reduced to $2.7sec.$

2 Model representation

Model-based reasoning about a system requires an explicit representation (a model) of the system's components and their connections. Reasoning is typically based on theorem proving when a model is represented by first-order logic (Genesereth 1984, Reiter 1987), or on constraint propagation (Davis 1984), possibly coupled with an ATMS (de Kleer & Williams 1987). We found *typed logic programs* (Lloyd 1987) useful to represent models since they can be naturally extended to efficiently solve constraints over finite domains (e.g., by forward checking, Van Hentenryck 1989) and to solve systems of linear equations and inequalities over real arithmetic terms (e.g., by a Constraint Logic Programming language CLP(R), Jaffar & Michaylov 1987).

In our approach it is essential that a model explicitly relates an internal state of components to external observations. A model M defines a mapping m from *any* state (normal and abnormal) to external observations. We denote the domain of m (states) by τ_x, the range (observations) by τ_y, and a typed version of m by m_τ, where

$$\forall x \in \tau_x \, \forall y \in \tau_y \; m_\tau(x,y) \leftarrow m(x,y).$$

Definition(model description)
A model description M consists of a type-free definition of $m_\tau(1)$, a type theory (2 and 3, Lloyd 1987), and a formal system which defines the mapping m (4):

1. $m_\tau(x,y) \leftarrow \tau_x(x), \tau_y(y), m(x,y)$.
2. $\tau(a)$. for each constant a of type τ.
3. $\tau(f(x_1,\ldots,x_n)) \leftarrow \tau_1(x_1),\ldots,\tau_n(x_n)$. for each functor f of type $\tau_1 \times \ldots \times \tau_n \rightarrow \tau$.
4. P, a formal system (e.g., a logic program, a system of constraints, a system of equations) which defines m.

Definition(diagnostic problem)
Given a model description M and an observation $y \in \tau_y$ a diagnosis Δ is a state $\Delta \in \tau_x$ such that $M \models m_\tau(\Delta, y)$.

In contrast to the abductive and consistency-based diagnosis, our approach can be characterized as *deductive* diagnosis. At this point we do not appeal to the principle of parsimony, and concentrate on the task of finding all, minimal and non-minimal diagnoses. However, finding an individual diagnosis (a model state consistent with the given observation), or determining an inconsistency of a state to the observation, is just a subproblem where efficiency considerations and the proposed solution apply as well.

The diagnostic problem is to find the inverse mapping m^{-1} for given observations y. Suppose P is a simulation model and consists of a system of equations over reals. In general, it is not possible to interpret equations or run the simulation 'backwards' in order to find the inverse mapping m^{-1}. Even if the domain τ_x is finite and P is a system of constraints, there may be no efficient constraint propagation method for a system with a large number of components and large domain τ_x. One possible solution is to represent M at several levels of abstraction and to first solve the diagnostic problem at an abstract level where the model is simpler and the search space smaller. The abstract, coarse solutions are then used to guide the search at more detailed levels, where the model is more complex and the search space larger.

2.1 An OR gate example

A model of an OR gate is specified in CLP(R) where unification is extended to solving constraints over real arithmetic terms. The model description is structurally decomposed, basic components are transistors and resistors (Figure 1).

The model relates internal states of transistors to real valued voltages and currents. The description of an npn transistor is from Heinze, Michaylov & Stuckey (1987). The

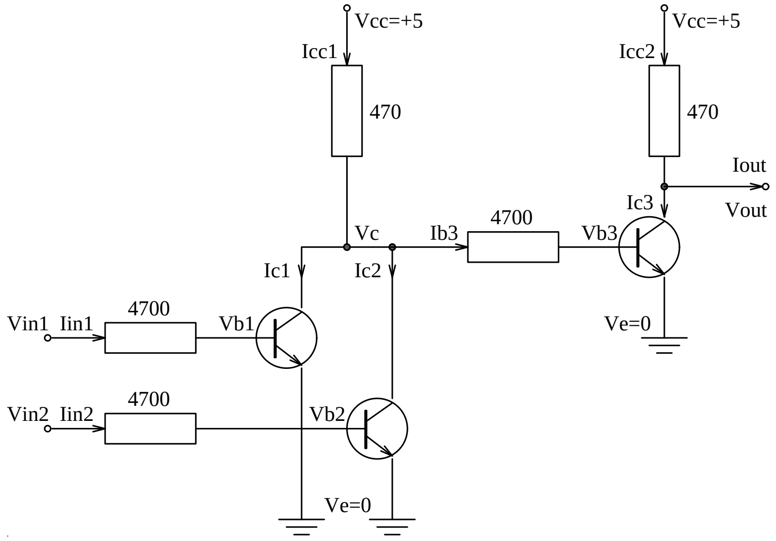

Figure 1: An OR gate realized with three npn transistors.

transistor operates in three states: *cutoff*, *saturated*, and *active*. In digital circuits only the cutoff and saturated states are of interest, while the active state, interesting in amplifier circuits, is included just for completeness. *Vx* and *Ix* denote the voltages and currents for the base, collector and emmiter, respectively. Constants *Beta*, *Vbe*, and *Vcesat* are device parameters.

```
org( s(S12,S3), X, Y, Z ) ←
    norg( S12, X, Y, W ),
    inv( S3, W, Z ).

norg( n(S1,S2), b(Vin1,Iin1), b(Vin2,Iin2), b(Vout,Iout) ) ←
    switch( S1, Vin1, Iin1, Vout, Ic1 ),
    switch( S2, Vin2, Iin2, Vout, Ic2 ),
    Ic1+Ic2=Ic,
    power( Ic, Vout, Iout ).

inv( S, b(Vin,Iin), b(Vout,Iout) ) ←
    switch( S, Vin, Iin, Vout, Ic ),
    power( Ic, Vout, Iout ).

switch( S, Vin, Iin, Vc, Ic ) ←
    Ve=0, Beta=100, Vbe=0.7, Vcesat=0.3,
    resistor( Vin, Vb, Iin, 4700 ),
    transistor( S, Beta, Vbe, Vcesat, Vb, Vc, Ve, Iin, Ic, Ie ).

power( Ic, Vout, Iout ) ←
    Vcc=5, Ic+Iout=Icc,
    resistor( Vcc, Vout, Icc, 470 ),
          0≤Iout, Iout≤0.006.
resistor( V1, V2, I, R )  ←  R>0, V1−V2=I*R.
transistor( cutoff, Beta, Vbe, Vcesat, Vb, Vc, Ve, Ib, Ic, Ie )  ←
          Vb<Ve+Vbe, Ib=0, Ic=0, Ie=0.
transistor( saturated, Beta, Vbe, Vcesat, Vb, Vc, Ve, Ib, Ic, Ie )  ←
          Vb=Ve+Vbe, Vc=Ve+Vcesat, Ib≥0, Ic≥0, Ie=Ic+Ib.
transistor( active, Beta, Vbe, Vcesat, Vb, Vc, Ve, Ib, Ic, Ie )  ←
          Vb=Ve+Vbe, Vc≥Vb, Ib≥0, Ic=Beta*Ib, Ie=Ic+Ib.
```

The following is a part of the type theory of the OR gate model, and a type-free definition of inv_τ.

$inv_\tau(S,\ X,\ Y) \ \leftarrow\ \tau_t(S),\ \tau_b(X),\ \tau_b(Y),\ inv(S,\ X,\ Y).$

$\tau_t(cutoff).$
$\tau_t(saturated).$
$\tau_t(active).$

$\tau_b(b(V,I)) \ \leftarrow\ \tau_v(V),\ \tau_i(I).$

$\tau_v(V) \ \leftarrow\ real(V).$
$\tau_i(I) \ \leftarrow\ real(I).$

The model can be used for simulation only, since it does not incorporate any fault model. Take, for example, the following query:

```
← org( S, b(5,I1), b(0,I2), b(V,I) ).
```

The CLP(R) interpreter returns the following answer substitution, with some unresolved constraints:

```
S=s(n(saturated, cutoff), cutoff),
I1=0.0009,
I2=0,
V=5−470*I,
0.006≥I, I≥0
```

3 Abstraction operators

A relation $M \mapsto M'$ denotes an abstraction from a detailed model M to an abstract model M'. In general, $\mapsto$ is a *partial* and not total mapping from M to M'. Below we define three abstraction operators which map M to M'.

Definition(abstraction operators)

1. Collapsing constants (in the domains of x and y).
   Different constants can be abstracted to a single constant. We represent constant abstractions $\mapsto$ by a binary predicate h. For example:

   $$\begin{array}{ll} \tau(a_1). \mapsto \tau'(a'). \qquad & h(a_1, a'). \\ \tau(a_2). \mapsto \tau'(a'). \qquad & h(a_2, a'). \end{array}$$

2. Deleting arguments of terms (x and y are terms in general).
   Irrelevant arguments at the detailed level can be deleted, for example

   $$\tau(f(x_1, x_2 \ldots, x_n)) \leftarrow \tau_1(x_1), \tau_2(x_2) \ldots, \tau_n(x_n). \quad \mapsto$$
   $$\tau'(f'(x_2, \ldots, x_n)) \leftarrow \tau_2'(x_2), \ldots, \tau_n'(x_n).$$

   where $\tau \mapsto \tau', \tau_i \mapsto \tau_i'(2 \leq i \leq n)$, and f' is f with the first argument deleted. If all arguments of f are eliminated we replace the functor f' by a constant a'. Term abstractions are also represented by the predicate h:

   $$h(f(x_1, x_2 \ldots, x_n), f'(x_2', \ldots, x_n')) \leftarrow h_2(x_2, x_2'), \ldots, h_n(x_n, x_n'). \qquad \text{or}$$
   $$h(f(x_1, x_2 \ldots, x_n), a').$$

3. Simplifying the mapping m and formal system P.
   Only some useful abstractions of $m \mapsto m'$ and the corresponding formal system $P \mapsto P'$ can be defined syntactically. If P is a set of wffs then the constituent atomic formulas can be abstracted in the following ways:

   (a) By uniformly renaming constant, function, and predicate symbols throughout P (the renaming is typically many-to-one, analogous to operator 1).

   (b) By uniformly deleting some arguments of functions and predicates throughout P (analogous to operator 2).

   In general, abstractions of m and P are defined implicitly by the consistency condition CC, in section 4.

Two examples of related work on abstractions:

1. Dropping conditions in ABSTRIPS (Sacerdoti 1974).
   A precondition $m(x, y)$ of an operator can be defined as a mapping m from a state of the world x to $y \in \{true, false\}$ depending on primitive conditions c_i being satisfied or not: $m(x, y) \leftarrow c_1, c_2, \ldots, c_n$. In the abstract space, some primitive conditions with low criticality (e.g., c_1) are deleted, and m is simplified to: $m'(x, y) \leftarrow c_2, \ldots, c_n$.

2. Abstracting a quantitative to a qualitative model (Gallanti *et al.* 1989).
   P is a system of linear equations $\Delta\mathrm{s} = \mathrm{C}\Delta\mathrm{p}$, where $\Delta\mathrm{p}$ are variations of the system parameters which caused variations of the observable state $\Delta\mathrm{s}$, and C is a sensitivity matrix. Real-valued variables and matrix coeficients are abstracted to 0 and 1: $\Delta\mathrm{s}, \Delta\mathrm{p}, \mathrm{c} = 0 \;\mapsto\; 0$ and $\Delta\mathrm{s}, \Delta\mathrm{p}, \mathrm{c} \neq 0 \;\mapsto\; 1$. Linear equations are simplified to

boolean expressions: $cx \pm y = z \mapsto cx \oplus y = z$, where

$$0 \oplus 0 = 0. \qquad 0 \oplus 1 = 1. \qquad 1 \oplus 0 = 1. \qquad 1 \oplus 1 = 0 \vee 1.$$

A qualitative model is used to generate candidate diagnoses which are then verified by conventional methods for solving linear systems.

3.1 Abstracting the quantitative OR gate model

For the purpose of diagnosis, when one has to identify just faulty components, the OR gate model is unneccessarily detailed. It does not really matted if the voltage is 4.4 or 4.6, what is important is whether it is qualitatively *high* or *low*, and whether the transistors properly operate as switching devices. Here we assume that only transistors can fail, and we ignore the *active* state at the abstract level. Abstraction operators will be applied throughout the model description. Since we will subsequently derive two more abstract OR gate models, the following qualitative model is placed at the third level of detail.

Recall that in the binary predicate h the first and second argument denotes the detailed and abstract entities, respectively. In what follows, the first six lines specify constant abstractions, the next three clauses specify term abstractions, and the last six clauses specify predicate abstractions.

$h_t(cutoff,\ ok).$
$h_t(saturated,\ ok).$

$h_i(0,\ 0).$
$h_i(I,\ 1) \leftarrow I{>}0.$

$h_v(V,\ 0) \leftarrow 0{\leq}V,\ V{<}0.7. \quad \%\ low$
$h_v(V,\ 1) \leftarrow 2{\leq}V,\ V{\leq}5. \quad \%\ high$

$h_n(n(S1,S2),\ n(S1',S2')) \leftarrow h_t(S1,S1'),\ h_t(S2,S2').$

$h_s(s(S12,S3),\ s(S12',S3')) \leftarrow h_n(S12,S12'),\ h_t(S3,S3').$

$h_b(b(V,I),\ V') \leftarrow h_v(V,V'). \qquad \%\ I\ is\ deleted$

$h(org(S,X,Y,Z),\ org_3(S',X',Y',Z')) \leftarrow$
$\quad h_s(S,S'),\ h_b(X,X'),\ h_b(Y,Y'),\ h_b(Z,Z').$

$h(norg(S,X,Y,Z),\ norg_3(S',X',Y',Z')) \leftarrow$
$\quad h_n(S,S'),\ h_b(X,X'),\ h_b(Y,Y'),\ h_b(Z,Z').$

$h(inv(S,X,Y),\ inv_3(S',X',Y')) \leftarrow$
$\quad h_t(S,S'),\ h_b(X,X'),\ h_b(Y,Y').$

$h(switch(S,Vin,Iin,Vc,Ic),\ switch_3(S',Vin',Vc',Ic')) \leftarrow \qquad \%\ Iin\ is\ deleted$
$\quad h_t(S,S'),\ h_v(Vin,Vin'),\ h_v(Vc,Vc'),\ h_i(Ic,Ic').$

$h(power(Ic,Vout,Iout),\ power_3(Ic',Vout')) \leftarrow \qquad \%\ Iout\ is\ deleted$

```
        h_i(Ic,Ic'), h_v(Vout,Vout').
    h(X+Y=Z, sum3(X',Y',Z'))  ←
        h_i(X,X'), h_i(Y,Y'), h_i(Z,Z').
```

From the original OR gate model, and the above abstractions, an abstract OR gate model was automatically derived through term rewriting and partial evaluation. The derivation can be regarded as an enhancement of explanation-based generalization without any learning example (Van Harmelen & Bundy 1988). Predicates, for which no abstraction is specified are regarded as 'non-operational' and are evaluated. The remaining predicates and terms are rewritten according to the abstraction specifications.

```
org3( s(S12,S3), X, Y, Z )  ←
    norg3( S12, X, Y, W ),
    inv3( S3, W, Z ).

norg3( n(S1,S2), Vin1, Vin2, Vout )  ←
    switch3( S1, Vin1, Vout, Ic1 ),
    switch3( S2, Vin2, Vout, Ic2 ),
    sum3( Ic1, Ic2, Ic ),
    power3( Ic, Vout ).

inv3( S, Vin, Vout )  ←
    switch3( S, Vin, Vout, Ic ),
    power3( Ic, Vout ).

switch3( ok, 0,_, 0 ).     % cutoff
switch3( ok, 1, 0, 0 ).    % saturated
switch3( ok, 1, 0, 1 ).    % saturated
switch3( ab, 1, 1, 0 ).    % open

power3( 0, 1 ).
power3( 1, 0 ).
power3( 1, 1 ).

sum3( 0, 0, 0 ).
sum3( 0, 1, 1 ).
sum3( 1, 0, 1 ).
sum3( 1, 1, 1 ).
```

The original OR gate model does not entail any fault model, and neither can the abstracted model. Therefore we introduced a strong fault model here, by adding a clause *switch*$_3$*(ab,1,1,0)*. This specifies that a *switch*$_3$ is abnormal (open) if for a high control voltage $Vin=1$, the voltage drop across the switch is high, $Vout=1$, and there is no current through the switch, $Ic=0$.

3.2 Abstracting the qualitative OR gate model

The next step involves just a structural abstraction, by ignoring the internal structure of the NOR gate and the inverter. The NOR gate is considered abnormal if any constituent transistor is in abnormal state.

$$\begin{array}{l}
h_n(n(ok,ok),\ ok).\\
h_n(n(ok,ab),\ ab).\\
h_n(n(ab,\_),\ ab).\\
h_s(s(S12,S3),\ s(S12',S3)) \ \leftarrow\ h_n(S12,S12').
\end{array}$$

$$\begin{array}{l}
h(org_3(S,X,Y,Z),\ org_2(S',X,Y,Z)) \ \leftarrow\ h_s(S,S').\\
h(norg_3(S,X,Y,Z),\ norg_2(S',X,Y,Z)) \ \leftarrow\ h_n(S,S').\\
h(inv_3(S,X,Y),\ inv_2(S,X,Y)).
\end{array}$$

The following is the automatically derived model at the second level of detail.

$$\begin{array}{l}
org_2(\ s(S1,S2),\ X,\ Y,\ Z\ ) \ \leftarrow\\
\quad norg_2(\ S1,\ X,\ Y,\ W\ ),\\
\quad inv_2(\ S2,\ W,\ Z\ ).
\end{array}$$

$$\begin{array}{l}
norg_2(\ ok,\ 0,\ 0,\ 1\ ).\\
norg_2(\ ok,\ 0,\ 1,\ 0\ ).\\
norg_2(\ ok,\ 1,\ 0,\ 0\ ).\\
norg_2(\ ok,\ 1,\ 1,\ 0\ ).\\
norg_2(\ ab,\ 0,\ 1,\ 1\ ).\\
norg_2(\ ab,\ 1,\ 0,\ 1\ ).\\
norg_2(\ ab,\ 1,\ 1,\ 1\ ).
\end{array}$$

$$\begin{array}{l}
inv_2(\ ok,\ 0,\ 1\ ).\\
inv_2(\ ok,\ 1,\ 0\ ).\\
inv_2(\ ab,\ 1,\ 1\ ).
\end{array}$$

Finally, a structurless description of an OR gate at the most abstract, first level is derived. Note that the resulting fault model is not the weakest, since it does not entail the behavior $org_1(ab,0,0,0)$.

$$\begin{array}{l}
h_s(s(ok,ok),\ ok).\\
h_s(s(ok,ab),\ ab).\\
h_s(s(ab,\_),\ ab).
\end{array}$$

$$h(org_2(S,X,Y,Z),\ org_1(S',X,Y,Z)) \ \leftarrow\ h_s(S,S').$$

$$\begin{array}{l}
org_1(\ ok,\ 0,\ 0,\ 0\ ).\\
org_1(\ ok,\ 0,\ 1,\ 1\ ).\\
org_1(\ ok,\ 1,\ 0,\ 1\ ).\\
org_1(\ ok,\ 1,\ 1,\ 1\ ).\\
org_1(\ ab,\ 0,\ 0,\ 1\ ).\\
org_1(\ ab,\ 0,\ 1,\ 0\ ).\\
org_1(\ ab,\ 0,\ 1,\ 1\ ).\\
org_1(\ ab,\ 1,\ 0,\ 0\ ).\\
org_1(\ ab,\ 1,\ 0,\ 1\ ).\\
org_1(\ ab,\ 1,\ 1,\ 0\ ).\\
org_1(\ ab,\ 1,\ 1,\ 1\ ).
\end{array}$$

4 Hierarchical diagnosis

In order to exploit possible computational advantages of multi-level over one-level model representation, two conditions must be satisfied by any pair M and M'. Let us denote by τ^- a subset of τ which is not abstracted since it is irrelevant at the abstract level, and by τ^+ a subset of τ which is abstracted. Define $\tau^-(x) \leftarrow \neg\exists x'\, h(x,x')$ and $\tau^+(x) \leftarrow \exists x'\, h(x,x')$. The following condition restricts the relation between the mapping m_τ, and subsets of its range τ_x^- and domain τ_y^+ which are (not) abstracted.

Definition(restriction of incompleteness)
For any m_τ, if $M \models m_\tau(x,y)$ then $\tau_x^-(x)$ or $\tau_y^+(y)$. With respect to the model M this is equivalent to:

$$C1: \ \ \forall x, y\; m(x,y) \Rightarrow \neg\exists x'\; h_x(x,x') \vee \exists y'\; h_y(y,y')$$

Note that the OR gate model at the third level of detail is incomplete with respect to the original model since the *active* states have no abstraction. Further, the condition C1 is not satisfied either, e.g., *org(s(n(cutoff, saturated), cutoff), b(−2,_), b(10,_), b(2.65, 0.005))* is true, the state x has an abstraction *s(n(ok,ok),ok)*, but the voltages -2 and 10 have no abstraction. This does not matter, however, since for diagnosis we need just the three qualitative models.

If we denote a subset of the mapping m_τ which is abstracted by m_τ^+ and define $m_\tau^+(x,y) \leftarrow \tau_x^+(x),\, \tau_y^+(y),\, m(x,y)$ then the following condition defines the relation between the detailed and abstract mapping.

Definition(preservation of mapping)
For any m_τ^+, if $M \models m_\tau^+$ then there exists m'_τ such that $M' \models m'_\tau$. With respect to M and M' this is equivalent to:

$$C2: \ \ \forall x, y\; (\exists x'', y''\; m(x,y) \wedge h_x(x,x'') \wedge h_y(y,y'')) \Rightarrow \exists x', y'\; m'(x',y') \wedge h_x(x,x') \wedge h_y(y,y')$$

In the case of the OR gate model, for example, the mapping *org(s(n(saturated, cutoff), cutoff), b(5, 0.0009), b(0, 0), b(2.18, 0.006))* has an abstraction *org$_3$(s(n(ok,ok),ok), 1,0,1)*. In general, if the abstraction operators are applied globally (as was the case with the OR gate example), and not only locally, to terms denoting model states and observations, the condition C2 is always satisfied. The condition C2 seems a well known characterization of a certain type of abstractions, e.g., Giunchiglia and Welsh (1989) call such abstractions *truthful*. As far as we know, the condition C1 is unique to our approach. A comparison of our approach to the related research on abstractions is in (Mozetic 1990a, 1991).

Conditions C1 and C2 which must hold for any model abstraction $M \mapsto M'$ can be conjoined into a *consistency condition*.

Definition(consistency condition)

$$CC: \ \ \forall x, y\; m(x,y) \wedge (\exists x''\; h_x(x,x'')) \Rightarrow \exists x', y'\; m'(x',y') \wedge h_x(x,x') \wedge h_y(y,y')$$

The following logic program tests whether CC holds between models M and M'.

Algorithm(consistency test)

$$\begin{aligned}
consistent &\leftarrow \neg inconsistent. \\
inconsistent &\leftarrow h_x(x, x'), m(x, y), \neg exist\_x'y'(x, y). \\
exist\_x'y'(x, y) &\leftarrow h_x(x, x'), h_y(y, y'), m'(x', y').
\end{aligned}$$

Theorem. The consistency test algorithm is correct and complete. The proof is a corollary to Lemmas 18.3 and 18.4 of (Lloyd 1987, p. 115).

Suppose that given is a list of models $M_1, \ldots, M_n$, ordered from abstract to detailed where corresponding state-observation mappings are defined by predicates $m_1, \ldots, m_n$. The hierarchical diagnostic algorithm is then defined by the following logic program which implements a depth-first, backtracking search through the space of possible states.

Algorithm(hierarchical diagnosis)

$$\begin{aligned}
diag_i(y, x) &\leftarrow h_y(y, y'), diag_{i-1}(y', x'), h_x(x, x'), m_i(x, y). \\
diag_i(y, x) &\leftarrow \neg exists\_x'(x), m_i(x, y). \\
exists\_x'(x) &\leftarrow h_x(x, x').
\end{aligned}$$

Theorem. If the consistency condition CC is satisfied by any two adjacent models then the hierarchical diagnostic algorithm is correct and complete with respect to the mapping m_i. The proof is in (Mozetic 1990a).

4.1 Diagnosing the 3-level OR gate

Take the three qualitative models of an OR gate derived automatically from the quantitative model. State-observation mappings are defined by $m_i(x, y) \leftrightarrow org_i(S, In1, In2, Out)$ for $i = 1, 2, 3$, where $x = S$, and $y = \langle In1, In2, Out \rangle$. Suppose that an observation $y = \langle 1, 0, 0 \rangle$ is given which indicates that the OR gate is faulty (the correct output would be 1). After submitting the query, the hierarchical diagnostic algorithm returns the following answer substitution:

$\leftarrow\ diag_3(\ \langle 1, 0, 0 \rangle,\ S\ ).$

$S = s(n(ab,\ ok),\ ok)$

The answer indicates that the first transistor in the NOR gate is faulty. The search space exploited by the algorithm is in Figure 2. Note that only three out of eight states at the third level are verified as candidate diagnoses, since neither *s(ok,ab)* nor *s(ab,ab)* are diagnoses at the second level.

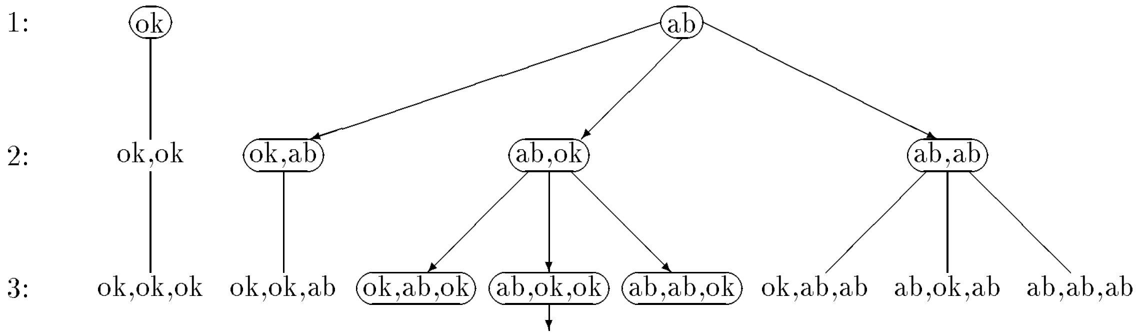

Figure 2: A search space for the 3-level OR gate example. States verified by the hierarchical diagnostic algorithm are in ovals, and states which actually map to the given observation 1,0,0 have outgoing arrows.

4.2 Complexity of hierarchical diagnosis

Let us assume that the cost δ of finding *all* diagnoses for the detailed level model M_n (without using abstractions) is a function v_n of the number of states S to be verified:

$$\delta(M_n) = v_n(S) = O(S)$$

In the worst case, when faults of the model components are independent, all states have to be verified and $S = \mid \tau_x \mid$. Sometimes it is possible to reduce the number of states S apriori, by domain specific knowledge. With multi-level models $M_1, \ldots, M_n$, the cost of hierarchical diagnosis is the sum of costs of verifying refinements of abstract diagnoses D_{i-1} at each level i. B_{i-1} denotes a branching factor from the level $i-1$ to i. In addition, we also have to take into account the number of newly introduced states N_i (without abstraction) which must be verified. The overall cost is:

$$\delta(M_1, \ldots, M_n) = \sum_{i=1}^{n} v_i(D_{i-1} \times B_{i-1} + N_i)$$

where $D_0 = 0$ and $N_1 = S_1$ (the number of states at the top level). When tree-structured hierarchies h_x are used, and there is no incompleteness ($N_{i>1} = 0$), the number of states at each level is $S_i = S_{i-1} \times B_{i-1}$. If we take $B_0 = S_1$ then the number of detailed level states can be expressed as a product of branching factors $S = \prod_{i=1}^{n} B_{i-1}$. With a constant branching factor B, the number of abstraction levels needed is $n = \log_B S$. If we make a simplifying assumption that v_i, D_i and B_i are constant across levels ($v(D \times B) = C$) then the linear complexity of finding all diagnoses is reduced to logarithmic:

$$\delta(M_1, \ldots, M_n) = v(D \times B) \times n = C \times \log_B S = O(\log S)$$

The reduction comes from the fact that, while the total number of states grows exponentially, the number of states to be verified is kept constant across levels. In our experiments this actually turned out to be the case.

5 Experiments and results

The experimental evaluation involves a realistic medical problem. In KARDIO (Bratko, Mozetic & Lavrac 1989), the ECG interpretation problem is formulated as follows: given a symbolic ECG description *ECG*, find all possible—single and multiple—heart disorders (cardiac arrhythmias *Arr*). In the medical literature there is no systematic description of ECG features which correspond to complicated multiple disorders. Instead of constructing diagnostic rules directly we developed a *simulation* model of the electrical activity of the heart. The model of the heart in KARDIO can simulate over 2400 heart disorders, but in the experiments described here we used a subset of the original model which comprises 943 heart failures (single and multiple).

The mapping m is defined by *m(Arr,ECG)* $\leftarrow$ *possible(Arr), heart(Arr,ECG). Possible(Arr)* eliminates physiologically impossible and medically uninteresting heart states and thus reduces their number from $\mid \tau_x \mid = 52920$ to $S = 943$. *Heart(Arr,ECG)* simulates the electrical activity of the heart for an arrhythmia *Arr*.

Due to the simulation nature of the model m its application in the 'forward' direction (deriving ECGs for a given disorder) can be carried out efficiently. In contrast, diagnostic reasoning (finding disorders for a given ECG) involves deep backtracking and renders the 'backward' application inefficient. Using the naive generate-and-test method with chronological backtracking, the average diagnostic time is more than $50sec.$. The application of more sophisticated constraint satisfaction techniques (reordering of constraints in *heart*, forward checking) provided no improvement. The computational complexity is due to the large domain size (52920), and high arity of predicates in the simulation model (components have between 6 and 15 arguments).

In order to improve diagnostic efficiency, we represented the heart model at four levels of abstraction. First, the three-level model was constructed in a top-down way, using QuMAS, a semiautomatic Qualitative Model Acquisition System (Mozetic 1987). The fourth, most detailed level was then added manually, by rewriting the original KARDIO heart model (which required a special interpreter) into a logic program which can be interpreted directly. All three abstraction/refinement operators were used in the hierarchical model representation (see Mozetic 1990b, 1991).

We compared diagnostic efficiency and the number of states to be verified by the hierarchical diagnostic algorithm and the one-level generate-and-test method (Table 1). Diagnostic efficiency is the time needed to find *all* possible diagnoses for a given ECG, and was measured and averaged over all 3096 distinct ECG descriptions at the detailed level. The heart model and the diagnostic algorithm were compiled by Quintus Prolog and run on SUN 3.

The experimental results are consistent with the complexity analysis. In one-level diagnosis all possible states have to be verified for each ECG. Thus $v_4(943) \propto 50.4sec.$ and the average time to verify a state is $53msec$. In hierarchical diagnosis the cost is $\sum_{i=1}^{4} v_i(D_{i-1} \times B_{i-1} + N_i)$. We can ignore the cost of verifying states without abstraction (N_i) since corresponding pairs state-observation were cached in a table. We further simplify the matter by assuming that costs of verifying states at different levels were equal, thus

	Domain	Possible	States without	Refinements of		Branching
Level	size	states	abstraction	abstract diagnoses	Diagnoses	factor
i	$\mid \tau_x \mid_i$	S_i	N_i	$D_{i-1} \times B_{i-1}$	D_i	B_i
1	3	3	3	0	1.0	5
2	48	18	3	5	1.9	10
3	10080	175	26	19	1.3	9
4	52920	943	0	12	2.1	/

Four-level, hierarchical diagnosis	36	2.7 sec.
One-level, generate-and-test	943	50.4 sec.

Table 1: The number of states verified and diagnostic times needed to find all diagnoses from the heart model, measured and averaged over 3096 distinct ECG descriptions.

yielding the overall cost $\sum_{i=1}^{4} v(D_{i-1} \times B_{i-1}) = v(36) \propto 2.7sec$. The approximate time to verify one state in hierarchical diagnosis is therefore $75msec$. This is close to one-level diagnosis and confirms that the number of states to be verified is an indicative measure of complexity.

6 Conclusion

We applied abstractions to model-based diagnosis, and showed a considerable improvement of diagnostic efficiency on a non-trivial medical problem. We defined three abstraction operators, formal conditions they have to satisfy, and a provably correct and complete diagnostic algorithm. With appropriate abstractions, the linear complexity of diagnosis can be reduced to logarithmic. The complexity reduction is due to simpler models and smaller search space at the abstract levels. The search space size depends on the branching factor of hierarchical relations and on the number of newly introduced states without abstraction (due to incompleteness). Therefore, reducing incompleteness and introducing intermediate levels improves the efficiency of hierarchical diagnosis. Despite the fact that our approach is geared towards the problem of finding all (including non-minimal) diagnoses, there are strong indications that abstraction hierarchies can also be used to find minimal diagnoses more efficiently. The questions how to find appropriate abstractions and when constructing abstract models is cost-effective, remain open. Our current research indicates that partial evaluation is a powerful technique to automatically construct abstract models on top of an existing detailed model, provided that abstractions of states and observations are given. The main limitation of our approach is that we ignore probabilities, and do not address the question of suggesting additional measurements.

Acknowledgements

Austrian Research Institute for Artificial Intelligence is supported by the Austrian Federal Ministery of Science and Research. Thanks to Paolo Petta for valuable comments, and to Robert Trappl for making some of this work possible.

References

Bratko, I., Mozetic, I., Lavrac, N. (1989). *KARDIO: A Study in Deep and Qualitative Knowledge for Expert Systems.* The MIT Press, Cambridge.

Davis, R. (1984). Diagnostic reasoning based on structure and behaviour. *Artificial Intelligence 24*, pp. 347-410.

de Kleer, J., Williams, B.C. (1987). Diagnosing multiple faults. *Artificial Intelligence 32*, pp. 97-130.

Gallanti, M., Roncato, M., Stefanini, A., Tornielli, G. (1989). A diagnostic algorithm based on models at different level of abstraction. *Proc. 11th IJCAI*, pp. 1350-1355, Detroit, Morgan Kaufmann.

Genesereth, M.R. (1984). The use of design descriptions in automated diagnosis. *Artificial Intelligence 24*, pp. 411-436.

Giunchiglia, F., Walsh, T. (1989). Abstract theorem proving. *Proc. 11th IJCAI*, pp. 372-377, Detroit, Morgan Kaufmann.

Heintze, N., Michaylov, S., Stuckey, P. (1987). CLP(R) and some electrical engineering problems. *Proc. 4th Intl. Conference on Logic Programming*, pp. 675-703, Melbourne, Australia, The MIT Press.

Jaffar, J., Michaylov, S. (1987). Methodology and implementation of a CLP system. *Proc. 4th Intl. Conference on Logic Programming*, pp. 196-218, Melbourne, Australia, The MIT Press.

Korf, R.E. (1987). Planning as search: a quantitative approach. *Artificial Intelligence 33*, pp. 65-88.

Lloyd, J.W. (1987). *Foundations of Logic Programming* (Second edition). Springer-Verlag, Berlin.

Mozetic, I. (1987). The role of abstractions in learning qualitative models. *Proc. 4th Intl. Workshop on Machine Learning*, pp. 242-255, Irvine, CA, Morgan Kaufmann.

Mozetic, I. (1990a). Abstractions in model-based diagnosis. Report TR-90-4, Austrian Research Institute for Artificial Intelligence, Vienna, Austria. *Working notes Automatic Generation of Approximations and Abstractions, AAAI-90 Workshop*, pp. 64-75, Boston, Boston, July 30, 1990.

Mozetic, I. (1990c). Diagnostic efficiency of deep and surface knowledge in KARDIO. *Artificial Intelligence in Medicine 2 (2)*, pp. 67-83, 1990.

Mozetic, I. (1991). Hierarchical model-based diagnosis. *International Journal of Man-Machine Studies 35 (3)*, pp. 329-362, 1991.

Plaisted, D.A. (1981). Theorem proving with abstractions. *Artificial Intelligence 16*, pp. 47-108.

Poole, D. (1989). Normality and faults in logic-based diagnosis. *Proc. 11th IJCAI*, pp. 1304-1310, Detroit, Morgan Kaufmann.

Reiter, R. (1987). A theory of diagnosis from first principles. *Artificial Intelligence 32*, pp. 57-95.

Sacerdoti, E.D. (1974). Planning in a hierarchy of abstraction spaces. *Artificial Intelligence 5*, pp. 115-135.

Van Harmelen, F., Bundy, A. (1988). Explanation-based generalisation = partial evaluation. *Artificial Intelligence 36*, pp. 401-412.

Van Hentenryck, P. (1989). *Constraint Satisfaction in Logic Programming.* The MIT Press, Cambridge.