Diagnostic Efficiency of Deep and Surface Knowledge in KARDIO

Igor Mozetič
Austrian Research Institute for Artificial Intelligence
Schottengasse 3, 1010 Vienna
Austria
igor@ai-vie.uucp

Abstract

The KARDIO system deals with the problem of diagnosing cardiac arrhythmias from symbolic descriptions of electrocardiograms. The system incorporates a qualitative model which simulates the electrical activity of the heart. In the paper we outline two methods for an efficient application of a simulation model to diagnosis. First, through abstractions and refinements, the model is represented at several levels of detail. Second, the model is 'compiled' into surface diagnostic rules. Through simulation, a relational table is generated and subsequently compressed into efficient diagnostic rules by inductive learning. A novel contribution to the KARDIO, presented here, includes a comparison of diagnostic efficiency and space complexity of four types of knowledge: a simulation model of the heart, a hierarchical four-level model, a relational table, and compressed diagnostic rules.

Keywords: Qualitative modelling, Abstractions, Machine learning

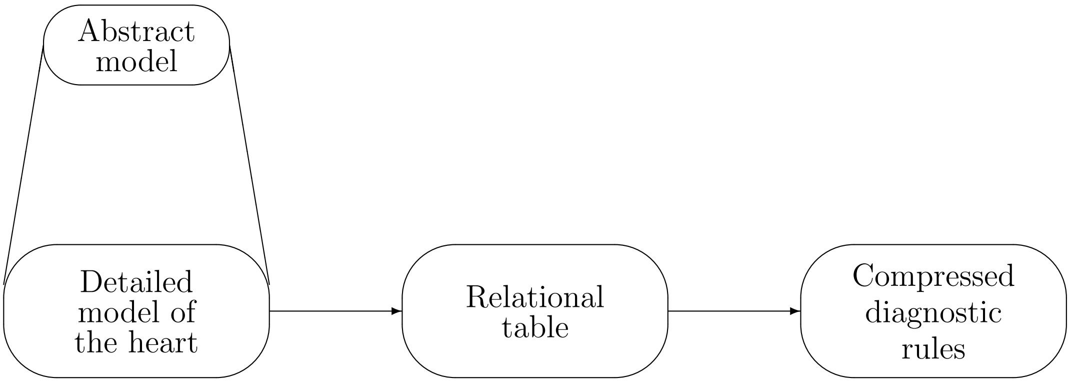

Figure 1: Deep and surface representations of the electrocardiographic knowledge.

1 Introduction

The heart can be viewed as a device with an electrical control system consisting of interconnected components. This electrical system works autonomously within the heart and is responsible for generating the rhythmical stimulation impulses that cause the contraction of the heart muscle, and consequently changes in the electrical potentials in the body. The changes of these potentials in time can be recorded as an electrocardiogram (ECG). Disorders which can occur in the electrical control system of the heart are reflected in the ECG curves. For example, an impulse generator may become silent or overactive, or some electrical conductance may become partially or totally blocked. These disorders are called *cardiac arrhythmias* and cause some characteristic changes in the ECG. The diagnostic problem is to decide which cardiac arrhythmias could have caused an abnormal ECG.

In KARDIO (Bratko *et al.* 1989), the ECG interpretation problem is formulated as follows: given a *symbolic* description of the ECG data, find *all* possible cardiac arrhythmias. There are both single and multiple disorders in the electrical system of the heart. In the medical literature (e.g. Goldman 1976), however, there is no systematic description of ECG features which correspond to complicated multiple disorders. Further, there is no simple rule yielding ECG features of multiple disorders, given ECG features of the constituent single disorders. These were the two main problems we encountered when attempting to construct the diagnostic knowledge base. Instead of directly constructing diagnostic rules which deal with multiple disorders we took an indirect approach. We first developed a simulation model of the electrical system of the heart. The model is qualitative in the sense that it does not deal with electrical signals represented numerically as functions of time, but rather by symbolic descriptions. The model can be efficiently used for simulation, but not for diagnosis.

In the paper we present and compare two alternative approaches to efficient application of a simulation model to diagnosis. First, by representing a deep model at several levels of abstraction, and second, by 'compiling' a model into a set of surface diagnostic rules (Figure 1). In section 2 we describe the deep model of the heart, its representation at several levels of abstraction, and the hierarchical diagnostic algorithm. The underlying idea is 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.

In section 3 we show the automatic compilation of the model into a set of efficient diagnostic rules. First, by exhaustive simulation, the model is transformed into a relational table. Entries in the table are then used as examples by an inductive learning program, and the table is compressed into a set of simple if-then rules.

We compare the complexity and efficiency of different diagnostic representations in section 4. Complexity is measured by the space required to store a knowledge base, and efficiency is the average time needed to find all diagnoses. In our experiments, a nontrivial subset of the original KARDIO problem domain was used. The model described here comprises 943 cardiac arrhythmias (both single and multiple) which arise by combining 29 single heart disorders. The original KARDIO domain can be reconstructed by a set of rules specified in (Bratko *et al.* 1989).

There are two novel contributions of the paper with respect to the KARDIO project. First, the heart model is represented by a logic program which does not require any special purpose interpreter. And second, four different representations (a one-level model, a four-level hierarchical model, a relational table, and compressed diagnostic rules) are compared on the same problem domain. In all cases, knowledge bases and diagnostic algorithms are implemented as logic programs and compiled by Quitus Prolog.

2 Deep model of the heart

2.1 Detailed level model

There are two fundamentaly different approaches to diagnostic reasoning. In the first, heuristic approach, one codifies diagnostic rules of thumb and experience of human experts in a given domain (e.g., MYCIN, Shortliffe 1976). 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.

A model of the electrical system of the heart comprises four types of components: impulse generators, conductors of impulses, impulse summators, and projectors of impulses to the ECG. In general, a component relates its qualitative state to the input and output. In the heart, the state of a component corresponds to an isolated disorder A, the input is an electrical impulse *Impulse*, and the output is either an electrical impulse or an individual ECG feature E. Specifically, the components have the following form:

- $generator(\ A_{STATE},\ Impulse_{OUT}\ )$
- $conductor(\ A_{STATE},\ Impulse_{IN},\ Impulse_{OUT}\ )$
- $summator(\ Impulse_{IN},\ Impulse_{IN},\ Impulse_{OUT}\ )$
- $projector(\ Impulse_{IN},\ E_{OUT}\ )$

An arrhythmia *Arr* is defined as a 7-tuple of isolated disorders A:

$$Arr = \langle A_1, \ldots, A_7 \rangle$$

Variables $A_1, \ldots, A_7$ denote states of the heart components (impulse generators or foci, and conductors):

$$Arr = \langle SA,\ AF,\ AV,\ JF,\ BB,\ VF,\ VEF\ \rangle$$

SA denotes the sino-atrial node, *AF* is an atrial focus, *AV* is the atrio-ventricular conduction, *JF* is a junctional focus, *BB* denotes conduction through the bundle branches, *VF* is a regular ventricular focus, and *VEF* is an ectopic ventricular focus. Each component may be in a normal or one of several abnormal states. For example, the normal state of the heart, sinus rhythm (*sr*) is defined by

$$Arr = \langle sr,\ quiet,\ normal,\ quiet,\ normal,\ quiet,\ quiet\ \rangle$$

where the SA node is in the state *sr*, other generators (*AF, JF, VF, VEF*) are *quiet* and both conductors (*AV, BB*) are *normal.* Sometimes we use the attribute-value notation instead of pure relational notation in order to improve the readability. Each element of a relational tuple is assigned to a variable which corresponds to the element position in the tuple. For example, a multiple arrhythmia, atrial tachycardia with the LGL syndrome and junctional ectopic beats (*at, lgl, jeb*) is described by the following 7-tuple:

$$\textit{Arr} = \langle \textit{SA=quiet, AF=at, AV=lgl, JF=jeb, BB=normal, VF=quiet, VEF=quiet} \rangle$$

An ECG pattern ECG is a 10-tuple of individual ECG features E:

$$ECG = \langle E_1, \ldots, E_{10} \rangle$$

The above arrhythmia (*at, lgl, jeb*) has three corresponding ECG patterns:

$$
\begin{array}{lll}
\textit{ECG} = \langle & \textit{Rhythm} = \textit{regular}, & \textit{regular}, \\
 & \textit{P\_wave} = \textit{abnormal}, & \textit{abnormal}, \\
 & \textit{Rate\_of\_P} = \textit{between\_100\_250}, & \textit{between\_100\_250}, \\
 & \textit{Relation\_P\_QRS} = \textit{after\_P\_always\_QRS}, & \textit{after\_P\_always\_QRS}, \\
 & \textit{PR\_interval} = \textit{shortened}, & \textit{shortened}, \\
 & \textit{QRS\_complex} = \textit{normal}, & \textit{normal}, \\
 & \textit{Rate\_of\_QRS} = \textit{between\_100\_250}, & \textit{between\_100\_250}, \\
 & \textit{Ectopic\_P} = \textit{abnormal}, & \textit{absent} \\
 & \textit{Ectopic\_PR} = \textit{after\_QRS\_is\_P} \vee \textit{shortened}, & \textit{meaningless} \\
 & \textit{Ectopic\_QRS} = \textit{normal} & \textit{normal} \rangle
\end{array}
$$

For shortness and better readability we allow for an internal disjunction to appear in a tuple. An expression $\langle E_1 = v_1, E_2 = v_2 \vee v_3 \rangle$ is equivalent to two pairs $\langle v_1, v_2 \rangle$ and $\langle v_1, v_3 \rangle$.

The heart model maps any arrhythmia (a single or a multiple disorder) to all corresponding ECG patterns. The mapping m from *Arr* to *ECG* is a relation (many-to-many) since each arrhythmia may have more than one corresponding ECG, and several arrhythmias may map to the same ECG pattern. For example, for the left-most ECG pattern above there are two possible arrhythmias:

$$
\begin{array}{c}
\langle \textit{SA=quiet, AF=at, AV=lgl, JF=jeb, BB=normal, VF=quiet, VEF=quiet} \rangle \\
\langle \textit{SA=quiet, AF=at, AV=normal, JF=jeb, BB=normal, VF=quiet, VEF=quiet} \rangle
\end{array}
$$

The mapping *m(Arr,ECG)* is defined by the *possible(Arr)*, and the simulation model *heart(Arr,ECG). Possible* eliminate physiologically impossible and medically uninteresting heart states, and *heart* simulates the heart activity for an arrhythmia *Arr*:

```
m( Arr, ECG )   ←
      possible( Arr ),
      heart( Arr, ECG ).
```

Throughout the paper, we define models, rules, and algorithms by logic programs. We use the standard Edinburgh Prolog syntax (e.g., Clocksin and Mellish 1984), where constants start with lowercase letters, variables start with capital letters, and all variables are implicitly universally quantified. However, we divert from the standard syntax by allowing for subscripts, and tuples instead of structured terms (e.g., $\langle X_1, X_2\rangle$ instead of $f(X_1, X_2)$).

The simulation model is defined by its structure (a set of components and their connections) and functions of the constituent components. The following clause defines the structure of the heart model:

```
heart( ⟨SA, AF, AV, JF, BB, VF, VEF⟩,
       ⟨Rhythm, P_wave, Rate_of_P, Relation_P_QRS, PR_interval,
       QRS_complex, Rate_of_QRS, Ectopic_P, Ectopic_PR, Ectopic_QRS⟩)  ←
    sa_node_generator( SA, ImpulseSA ),
    atrial_generator( AF, ImpulseAF ),
    summator( ImpulseSA, ImpulseAF, ImpulseATR ),
    anterograde_av_conductor( AV, ImpulseATR, ImpulseAV ),
    junctional_generator( JF, ImpulseJF ),
    regular_ventricular_generator( VF, ImpulseVF ),
    ectopic_ventricular_generator( VEF, ImpulseVF ),
    summator( ImpulseJF, ImpulseVF, ImpulseIV ),
    retrograde_av_conductor( AV, ImpulseIV, ImpulseRET ),
    summator( ImpulseAV, ImpulseIV, ImpulseINV ),
    summator( ImpulseRET, ImpulseATR, ImpulseSV ),
    summator( ImpulseAV, ImpulseJF, ImpulseHIS ),
    bundle_branches_conductor( BB, ImpulseHIS, ImpulseBB ),
    summator( ImpulseBB, ImpulseVF, ImpulseVENT ),
    atrial_projector( ImpulseSV, P_wave, Rate_of_P, Ectopic_P ),
    atrio_vent_projector( ImpulseSV, ImpulseINV, Relation_P_QRS,
                          PR_interval, Ectopic_PR ),
    ventricular_projector( ImpulseVENT, Rhythm, QRS_complex,
                          Rate_of_QRS, Ectopic_QRS ).
```

The head of the clause relates the state of the heart *Arr* to the output *ECG*. Atoms in the body represent heart components (generators, conductors, summators and projectors), and shared variables (impulses) denote connections between components.

Due to the simulation nature of the model m, its application in the 'forward' direction

$$m(\langle A_1, \ldots, A_7\rangle) \;\mapsto\; \langle E_1, \ldots, E_{10}\rangle$$

can be carried out efficiently. For a given disorder *Arr*, the logic program interpreter can derive all *ECG* patterns resorting only to shallow backtracking. For diagnostic purposes, however, the 'backward' application is required—for a given *ECG* find all *Arr*:

$$m^{-1}(\langle E_1, \ldots, E_{10} \rangle) \;\mapsto\; \langle A_1, \ldots, A_7 \rangle$$

Since the model m is specified by a logic program there is no inherent obstacle to the 'backward' application. However, the reasoning from *ECG* to *Arr* involves deep backtracking where a large number of fruitless paths are explored, and therefore renders the 'backward' application inefficient. The main source of fruitless branching is the model component *summator(X,Y,Z)* which, when applied, requires that for a given impulse Z, a pair of impulses X and Y is to be found, such that their 'sum' yields Z. Usually, there is a number of possible decompositions of Z, only few of which are consistent with other constraints in the model, and further, those inconsistencies may be found only in late stages of the model application.

Until now, all attempts to *directly* use the model for efficient diagnosis failed. Using the naive generate-and-test method with chronological backtracking, the average diagnostic time is more than 50 sec. The application of more sophisticated constraint satisfaction techniques (different goal selection strategies and forward checking, e.g., Van Hentenryck 1989) provided no improvement. The computational complexity is due to the large number of syntactically possible states (52,920), and high arity of predicates in the model. Impulses are namely structured terms with 5 arguments, and the model components have therefore between 6 and 15 primitive arguments.

2.2 Model abstractions and refinements

One approach to improve the diagnostic efficiency of the deep model is to represent it at several levels of abstraction, and to first solve the diagnostic problem at an abstract level. The abstract diagnoses are then used to restrict the search for more detailed diagnoses.

First we define three abstraction/refinement operators which can be used in a multi-level model representation. The abstraction operators are applied when one simplifies a model in a *bottom-up* fashion (from detailed to abstract). Complementary refinement operators are used in a *top-down* model development (from abstract to detailed).

- Collapse/refinement of values.
  Indistinguishable values of a variable can be abstracted into a single value. For example, the values *wide_LBBB* and *wide_RBBB* of the ECG feature *QRS_complex* are abstracted to *wide*. We represent the abstraction by a binary predicate h:

$$h(\mathit{wide\_LBBB}, \mathit{wide}). \qquad\qquad h(\mathit{wide\_RBBB}, \mathit{wide}).$$

- Deletion/introduction of variables.
  Irrelevant variables can be deleted at the abstract level. For example, the last three ECG features *Ectopic_P, Ectopic_PR,* and *Ectopic_QRS* can be ignored. This is represented by the following clause:

  $$h(\langle E_1, \ldots, E_{10} \rangle, \langle E'_1, \ldots, E'_7 \rangle) \;\; \leftarrow \;\; h(E_1, E'_1), \ldots, h(E_7, E'_7).$$

  where E'_i denote the abstract ECG features.

- Simplification/elaboration of the mapping m.
  Detailed level mapping m can be simplified to m' by ignoring and/or simplifying some model components. In general, however, the mapping abstractions are defined by a formal consistency condition which must hold between m and m' (for details see Mozetic 1990a,b).

By an application of the abstraction and refinement operators, the heart model was represented at four levels of detail. All three abstraction/refinement operators were used. Apart to the introduction of new variables, values of the variables were refined at each level of detail. The hierarchical model defines different mappings $m_1, \ldots, m_4$ from *Arr* to *ECG* by introducing new components at each level. First, the three-level model was constructed in a top-down fashion, using QuMAS, a semiautomatic Qualitative Model Acquisition System (Mozetic 1987). The fourth, detailed level model, described in the previous subsection was then added manually. The heart model at the first, most abstract level is very simple:

```
heart( Arr, ECG )  ←
    generator( Arr, Impulse ),
    projector( Impulse, ECG ).

    generator( brady, form(under_60) ).
    generator( rhythm, form(between_60_100) ).
    generator( tachy, form(over_100) ).

    projector( form(Rate), Rate ).
```

It consists of only two components, an impulse generator and a projector. Instead of a 7-tuple, an arrhythmia description *Arr* is a singleton and only three abstract arrhythmias are considered: a bradycardia (*brady*), a normal rhythm (*rhythm*), and a tachycardia (*tachy*). An *ECG* pattern is also a singleton, covering one ECG feature, *Rate_of_QRS*. An impulse is a structured term, but at this level it has only one argument (rate) in contrast to five arguments at the detailed level (shape, rhythm and rate of the regular part, and type and shape of the ectopic part).

Suppose that given is a list of mappings $m_1, \ldots, m_n$, ordered from abstract to detailed. Hierarchical relations between detailed and abstract ECG patterns are defined by the predicate h_E, and relations between detailed and abstract arrhythmias by the predicate h_A. The hierarchical diagnostic algorithm is then defined by the following logic program which implements a depth-first, backtracking search through the space of possible diagnoses:

```
$diag_i$( ECG, Arr )  $\leftarrow$
    $h_E$( ECG, ECG' ),
    $diag_{i-1}$( ECG', Arr' ),
    $h_A$( Arr, Arr' ),
    $m_i$( Arr, ECG ).
$diag_i$( ECG, Arr )  $\leftarrow$
    $\neg\exists Arr'$ $h_A$( Arr, Arr' ),
    $m_i$( Arr, ECG ).
```

The algorithm first climbs the hierarchy of ECG patterns, and recursively finds an abstract diagnosis *Arr'* which maps to the abstract *ECG'*. The detailed model m_i then simulates refinements *Arr* of *Arr'* to verify which arrhythmias actually map to the given *ECG*. Notice that detailed arrhythmias which are not refinements of the abstract diagnosis *Arr'* are not considered at all. With appropriate abstractions, this results in a major reduction of the search space at the detailed level.

However, an abstract model may be incomplete with respect to the the detailed level model, i.e., not all phenomena are necessarily abstracted. For example, the abstract model of the heart above does not incorporate any conduction disorders—they are introduced only at more detailed levels. Unfortunately, such incompleteness prevents the search space reduction at an abstract level. The diagnostic algorithm has to resort to the inefficient generate-and-test method for the arrhythmias *Arr* without abstractions. This is covered by the second clause of the algorithm.

Formal conditions which have to be satisfied by the hierarchical model representation, and a relation to the relevant work on abstractions are in (Mozetic 1990a,b). If the conditions are satisfied, then the diagnostic algorithm is correct and complete. Further, with appropriate abstractions, the algorithm reduces the linear complexity of the generate-and-test method to logarithmic.

2.3 Hierarchical diagnosis

In this section we give an example of diagnostic reasoning based on the heart model represented at four levels of detail. Suppose the following ECG pattern at the fourth level of detail is given:

$$
\begin{aligned}
ECG_4 = \langle & \mathit{Rhythm} = \mathit{regular}, \\
& \mathit{P\_wave} = \mathit{abnormal}, \\
& \mathit{Rate\_of\_P} = \mathit{between\_100\_250}, \\
& \mathit{Relation\_P\_QRS} = \mathit{after\_P\_always\_QRS}, \\
& \mathit{PR\_interval} = \mathit{shortened}, \\
& \mathit{QRS\_complex} = \mathit{normal}, \\
& \mathit{Rate\_of\_QRS} = \mathit{between\_100\_250}, \\
& \mathit{Ectopic\_P} = \mathit{abnormal}, \\
& \mathit{Ectopic\_PR} = \mathit{after\_QRS\_is\_P}, \\
& \mathit{Ectopic\_QRS} = \mathit{normal} \rangle
\end{aligned}
$$

The hierarchical diagnostic algorithm first uses hierarchies defined by h_E to find a more abstract ECG pattern. At the third level, the last three variables *Ectopic_P, Ectopic_PR,* and *Ectopic_QRS* are deleted:

$$
\begin{aligned}
ECG_3 = \langle & \mathit{Rhythm} = \mathit{regular}, \\
& \mathit{P\_wave} = \mathit{abnormal}, \\
& \mathit{Rate\_of\_P} = \mathit{between\_100\_250}, \\
& \mathit{Relation\_P\_QRS} = \mathit{after\_P\_always\_QRS}, \\
& \mathit{PR\_interval} = \mathit{shortened}, \\
& \mathit{QRS\_complex} = \mathit{normal}, \\
& \mathit{Rate\_of\_QRS} = \mathit{between\_100\_250} \rangle
\end{aligned}
$$

At the second level of abstraction, variables *Rhythm, Rate_of_P,* and *PR_interval* are deleted. Values of *P_wave = abnormal* and *QRS_complex = normal* are both abstracted to the value *present*:

$$
\begin{aligned}
ECG_2 = \langle & \mathit{P\_wave} = \mathit{present}, \\
& \mathit{Relation\_P\_QRS} = \mathit{after\_P\_always\_QRS}, \\
& \mathit{QRS\_complex} = \mathit{present}, \\
& \mathit{Rate\_of\_QRS} = \mathit{over\_100} \rangle
\end{aligned}
$$

At the most abstract level, all variables but *Rate_of_QRS* are deleted:

$$
ECG_1 = \langle \mathit{Rate\_of\_QRS}{=}\mathit{over\_100} \rangle
$$

The abstract model of the heart is then used to find a possible diagnosis at this extremely simple level. The only possibility is *tachy*—a tachycardia in medical terminology. Now the algorithm resorts to hierarchies of arrhythmias h_A to refine this abstract diagnosis, and uses more detailed heart models to verify which refinements can actually produce the given ECG pattern.

$$
\begin{array}{llllllll}
\mathrm{Arr}_1 = & & & & \langle \mathit{Arr}\, \rangle & & & \\
\mathrm{Arr}_2 = & & \langle \mathit{SV}, & \mathrm{AV}, & & & \mathit{IV}\, \rangle & \\
\mathrm{Arr}_3 = & \langle \mathit{SA}, & \mathit{AF}, & \mathit{AV}, & \mathit{JF}, & \mathit{BB}, & & \mathit{VF}\, \rangle \\
\mathrm{Arr}_4 = & \langle \mathit{SA}, & \mathit{AF}, & \mathit{AV}, & \mathit{JF}, & \mathit{BB}, & \mathit{VF}, & \mathit{VEF}\, \rangle
\end{array}
$$

Figure 2: Representation of arrhythmias at different levels of detail.

Hierarchies of arrhythmias are more complicated than hierarchies of ECG patterns. At each level new variables are introduced, and typically a value of an abstract level variable depends on values of tuples of detailed level variables and not only on individual detailed level variables (as is the case with ECG patterns). Recall that individual variables correspond to the states of the heart components and that their values denote isolated disorders. Figure 2 defines hierarchies of tuples and dependencies between individual variables. At the first level *Arr* denotes the state of the heart, regarded as an impulse generator. At the second level, SV corresponds to a supra-ventricular focus, AV is the atrio-ventricular conduction, and IV denotes an intra-ventricular focus. Variables at the third and the fourth level were defined in section 1.1.

Figure 3 gives some examples of hierarchical relations between values of individual variables and tuples of variables. A variable which has no value assignment in a tuple can take any value from its domain. Abbreviations for isolated arrhythmias used at the fourt level of detail correspond to the following medical terms: *st* is sinus tachycardia, *aeb* are atrial ectopic beats, *at* is atrial tachycardia, *mat* is multifocal atrial tachycardia, *lgl* is the LGL syndrom, *wpw* is the WPW syndrom, *avb1* is the AV block, first degree, *wen* is the AV block of type Wenckebach, *mob2* is the AV block, type Mobitz 2, *avb3* is the AV block, third degree, *jt* is junctional tachycardia, *jeb* are junctional ectopic beats, *vt* is ventricular tachycardia, *lbbb* is left bundle branch block, *rbbb* is right bundle branch block, and *veb* are ventricular ectopic beats.

In our example, hierarchies in Figure 3 are used by the diagnostic algorithm to refine the abstract level diagnosis *tachy*. The following dialog with the system illustrates the depth-first search for diagnoses through abstraction spaces. The user responses are in italics and each diagnosis is followed by the corresponding medical term.

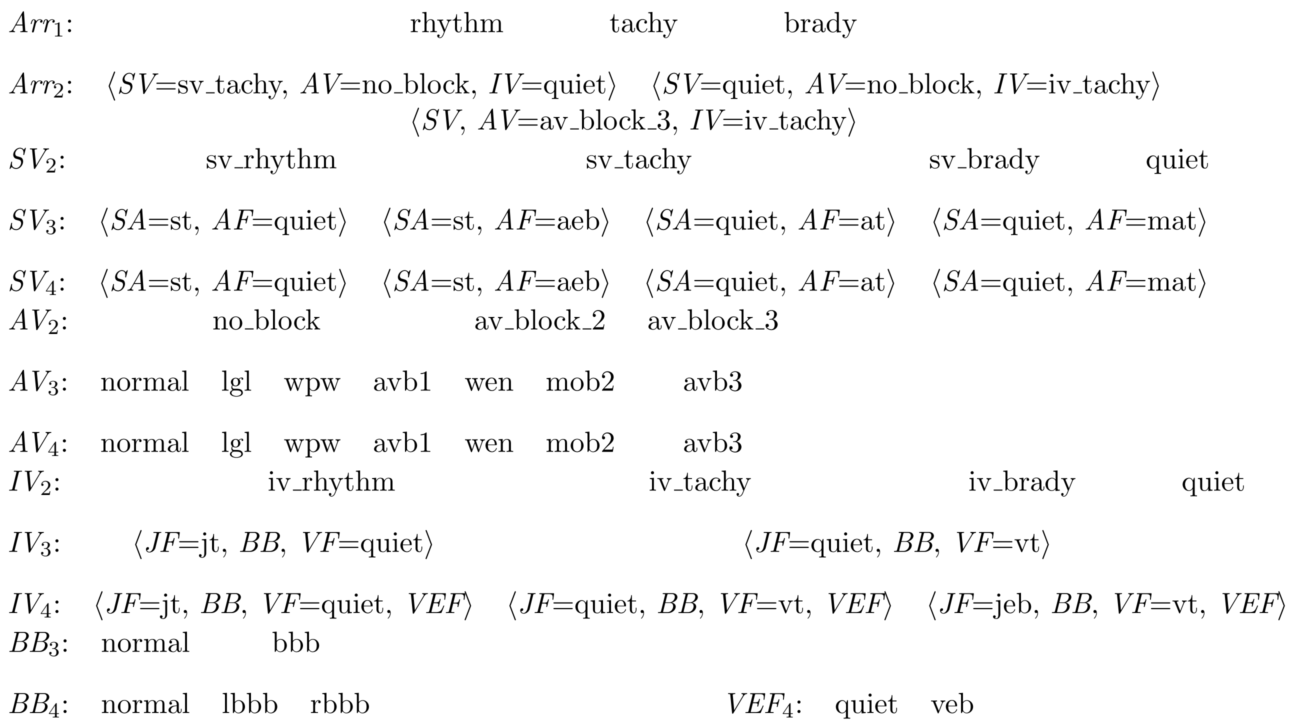
Arr_1: rhythm tachy brady
Arr_2: $\langle SV$=sv_tachy, AV=no_block, IV=quiet$\rangle$ $\langle SV$=quiet, AV=no_block, IV=iv_tachy$\rangle$
$\langle SV$, AV=av_block_3, IV=iv_tachy$\rangle$
SV_2: sv_rhythm sv_tachy sv_brady quiet
SV_3: $\langle SA$=st, AF=quiet$\rangle$ $\langle SA$=st, AF=aeb$\rangle$ $\langle SA$=quiet, AF=at$\rangle$ $\langle SA$=quiet, AF=mat$\rangle$
SV_4: $\langle SA$=st, AF=quiet$\rangle$ $\langle SA$=st, AF=aeb$\rangle$ $\langle SA$=quiet, AF=at$\rangle$ $\langle SA$=quiet, AF=mat$\rangle$
AV_2: no_block av_block_2 av_block_3
AV_3: normal lgl wpw avb1 wen mob2 avb3
AV_4: normal lgl wpw avb1 wen mob2 avb3
IV_2: iv_rhythm iv_tachy iv_brady quiet
IV_3: $\langle JF$=jt, BB, VF=quiet$\rangle$ $\langle JF$=quiet, BB, VF=vt$\rangle$
IV_4: $\langle JF$=jt, BB, VF=quiet, $VEF\rangle$ $\langle JF$=quiet, BB, VF=vt, $VEF\rangle$ $\langle JF$=jeb, BB, VF=vt, $VEF\rangle$
BB_3: normal bbb
BB_4: normal lbbb rbbb
VEF_4: quiet veb

Figure 3: Some examples of the hierarchical relation h_A between the abstract and detailed level arrhythmias.

A possible diagnosis:
$\rangle$ $Arr_1 = \mathit{tachy}$
Tachycardia

More detailed diagnosis? *yes*
$\rangle\rangle$ $Arr_2 = \langle \mathit{SV{=}sv\_tachy,\ AV{=}no\_block,\ IV{=}quiet} \rangle$
Supra-ventricular tachycardia

More detailed diagnosis? *yes*
$\rangle\rangle\rangle$ $Arr_3 = \langle \mathit{SA{=}quiet,\ AF{=}at,\ AV{=}normal,\ JF{=}quiet,\ BB{=}normal,\ VF{=}quiet} \rangle$
Atrial tachycardia

More detailed diagnosis? *yes*
$\rangle\rangle\rangle\rangle$ $Arr_4 = \langle \mathit{SA{=}quiet,\ AF{=}at,\ AV{=}normal,\ JF{=}jeb,\ BB{=}normal,\ VF{=}quiet,\ VEF{=}quiet} \rangle$
Atrial tachycardia with junctional ectopic beats

Alternative diagnosis? *yes*
$\rangle\rangle\rangle$ $Arr_3 = \langle \mathit{SA{=}quiet,\ AF{=}at,\ AV{=}lgl,\ JF{=}quiet,\ BB{=}normal,\ VF{=}quiet} \rangle$
Atrial tachycardia with the LGL syndrome

More detailed diagnosis? *yes*
$\rangle\rangle\rangle\rangle$ $Arr_4 = \langle \mathit{SA{=}quiet,\ AF{=}at,\ AV{=}lgl,\ JF{=}jeb,\ BB{=}normal,\ VF{=}quiet,\ VEF{=}quiet} \rangle$
Atrial tachycardia with the LGL syndrome and junctional ectopic beats

Alternative diagnosis? *yes*
$\rangle\rangle$ $Arr_2 = \langle \mathit{SV{=}quiet,\ AV{=}no\_block,\ IV{=}iv\_tachy} \rangle$
Intra-ventricular tachycardia

More detailed diagnosis? *yes*
$\rangle\rangle\rangle$ $Arr_3 = \langle \mathit{SA{=}quiet,\ AF{=}quiet,\ AV{=}normal,\ JF{=}jt,\ BB{=}normal,\ VF{=}quiet} \rangle$
Junctional tachycardia

More detailed diagnosis? *yes*
$\rangle\rangle\rangle\rangle$ No consistent refinement !

For the given detailed ECG pattern, there are two possible diagnoses: atrial tachycardia with junctional ectopic beats, and atrial tachycardia with the LGL syndrome and junctional ectopic beats. The first diagnosis appear to be more general than the second one, but for a physician it is important to be aware of both possibities, since the second diagnosis is potentially more dangerous and might require a different treatment. Note that a diagnosis possible at the third level, junctional tachycardia, has several refinements at the fourth level, but none of them actually maps to the given ECG pattern.

3 Surface diagnostic rules

3.1 Derivation of a relational table

Another, *indirect* approach to use a deep model for efficient diagnosis is to 'compile' it. The 'compilation' proceeds in two steps. First, by exhaustive simulation, the model is transformed into a relational table. Entries in the table are then used as examples by an inductive learning program, and the table is compressed into a set of simple if-then rules.

The heart model m relates all arrhythmias Arr to all corresponding ECG patterns ECG, and through simulation one can generate a complete set of relations ⟨Arr, ECG⟩:

$$m(Arr, ECG) \mapsto \langle A_1, \ldots, A_7, E_1, \ldots, E_{10} \rangle$$

Such a relational table can be used for efficient diagnosis, if not excessively large. In KARDIO, for example, a table generated from the original model of the heart consists of over 140,000 entries. When properly organized into a set of rules it still occupies over 5 Mb, stored as a text file.

In many practical applications it might not even be feasible to generate all pairs disorder-observation, but only a small subset. Some *inductive learning* techniques must then be applied to the subset in order to extend the coverage to the whole diagnostic space (or at least most of it). The same approach of constructing a qualitative model, exhaustive simulation, and induction of compressed diagnostic rules was taken by Pearce (1988) to automatically construct a fault diagnosis system of a satellite power supply. Similarly, Buchanan *et al.* (1988) show the advantage of using a classical simulation model to generate a (non-exhaustive) set of learning and testing examples, which are then used to induce rules for location of errors in particle beam lines used in high energy physics.

3.2 Compression by inductive learning

In inductive learning (e.g., Michalski 1983), one is given a set of *learning examples* and some background knowledge, and the goal is to find a *concept description* which is consistent and complete with respect to the examples. A learning example is a pair

$$\langle object, class_i \rangle$$

where *object* is described by a tuple of attribute values $\langle v_1, \ldots, v_n \rangle$, and $class_i$ denotes an instance of the concept. The induced concept description is usually in the form of if-then rules:

$$\textbf{if}\ \ \lambda(v_1, \ldots, v_n)\ \ \textbf{then}\ \ \ class_i \qquad\qquad \text{or} \qquad\qquad \textbf{if}\ \ class_i\ \ \textbf{then}\ \ \lambda(v_1, \ldots, v_n)$$

where $\lambda(v_1, \ldots, v_n)$ is a boolean expression. The goal of learning is to find a logical expression λ for each $class_i$ which is as simple as possible, but sufficient to discriminate between the $class_i$ and all other classes $class_j,\ i \neq j$. It is worth emphasizing that in general, an if-then rule is not a logical implication, but rather a relation *if_then(*$class_i, \langle v_1, \ldots, v_n \rangle$*)*. The antecedent and the consequent of an if-then rule can be interchanged depending on the problem solving strategy since they merely indicate the direction of inference.

The inductive learning techniques were applied to the generated relational table. First, 10 sets of learning examples were prepared. For each 17-tuple relation in the table, 10 new 8-tuple relations were formed by projection:

$$\langle A_1, \ldots, A_7, E_k \rangle \ \leftarrow\ \langle A_1, \ldots, A_7, E_1, \ldots, E_{10} \rangle \qquad (1 \leq k \leq 10)$$

Then an inductive learning program NEWGEM, an ancestor of AQ15 (Michalski *et al.* 1986) was used. The result of learning were 10 sets of compressed diagnostic rules:

$$\begin{array}{c} \textbf{if}\ \ E_1\ \ \textbf{then}\ \ \lambda(A_1, \ldots, A_7) \\ \vdots \\ \textbf{if}\ \ E_{10}\ \ \textbf{then}\ \ \lambda(A_1, \ldots, A_7) \end{array}$$

A rule set k $(1 \leq k \leq 10)$ relates an individual ECG feature E_k to corresponding arrhythmias, described by $\lambda(A_1, \ldots, A_7)$. Each rule in a set k relates a value v_i of E_k to a minimal description of corresponding arrhythmias $\lambda(A_1, \ldots, A_7)$ which is still sufficient to discriminate between v_i and other values $v_j,\ i \neq j$ of E_k. For example, the following two rules belong to the ECG feature *QRS_complex* and discriminate between the values *wide_non_specific, normal*, and the remaining possible values.

$$\begin{array}{ll}
\textbf{if} & \mathit{QRS\_complex} = \mathit{wide\_non\_specific} \\
\textbf{then} & \mathit{VF} = \mathit{vr} \lor \mathit{avr} \lor \mathit{vt} \lor \mathit{vfl}
\end{array}$$

$$\begin{array}{ll}
\textbf{if} & \mathit{QRS\_complex} = \mathit{normal} \\
\textbf{then} & \mathit{AV} \neq \mathit{wpw} \ \land \\
 & \mathit{BB} = \mathit{normal} \ \land \\
 & \mathit{VF} = \mathit{quiet}
\end{array}$$

The first rule states that if a *QRS_complex* = *wide_non_specific* (i.e., *wide*, but not of type *LBBB* or *RBBB*) then there is either a ventricular rhythm (*vr*), an accelerated ventricular rhythm (*avr*), a ventricular tachycardia (*vt*), or a ventricular flutter (*vfl*) originating in a ventricular focus (*VF*). States of the remaining heart components are unspecified. On the other hand, if *QRS_complex* = *normal* then the ventricular focus must be *quiet*, the conduction through bundle branches (*BB*) is *normal*, and the atrio-ventricular conduction (*AV*) can be anything but the WPW syndrome (*wpw*).

The following two rules are slightly more complicated:

$$\begin{array}{ll}
\textbf{if} & \mathit{P\_wave} = \mathit{abnormal} \\
\textbf{then} & \mathit{SA} = \mathit{quiet} \ \land \\
 & \mathit{AF} \neq \mathit{wp} \lor \mathit{mat} \ \land \\
 & \mathit{VF} \neq \mathit{vfl} \lor \mathit{vf}
\end{array}$$

$$\begin{array}{ll}
\textbf{if} & \mathit{PR\_interval} = \mathit{shortened} \\
\textbf{then} & \mathit{AF} \neq \mathit{afl} \lor \mathit{af} \ \land \\
 & \mathit{AV} = \mathit{wpw} \lor \mathit{lgl} \\
 & \lor \\
 & \mathit{SA} = \mathit{quiet} \ \land \\
 & \mathit{AF} = \mathit{at} \lor \mathit{aeb} \lor \mathit{quiet} \ \land \\
 & \mathit{AV} = \mathit{normal} \ \land \\
 & \mathit{VF} = \mathit{quiet}
\end{array}$$

Formally, an internal disjunction in the consequent of a rule is equivalent to a set membership, i.e., $A = v_1 \lor v_2 \ \Leftrightarrow \ A \in \{v_1, v_2\}$. The inequality is defined by $A \neq v_1 \lor v_2 \ \Leftrightarrow \ A \in \mathit{Domain}(A) - \{v_1, v_2\}$. Notice that the last rule, for the *PR_interval*, has a disjunctive consequent. In a logic program, such a rule is represented by two clauses:

$$\begin{array}{l}
\mathit{if\_then(\ shortened,\ \langle SA,\ AF,\ AV,\ JF,\ BB,\ VF,\ VEF\ \rangle)} \quad \leftarrow \\
\qquad \mathit{AF} \in \{\mathit{wp,\ at,\ mat,\ aeb,\ quiet}\}, \\
\qquad \mathit{AV} \in \{\mathit{wpw,\ lgl}\}.
\end{array}$$

$$
\begin{aligned}
&\textit{if\_then}(\ \textit{shortened},\ \langle SA,\ AF,\ AV,\ JF,\ BB,\ VF,\ VEF\ \rangle) \quad \leftarrow \\
&\qquad SA\ = \textit{quiet}, \\
&\qquad AF \in \{\textit{at},\ \textit{aeb},\ \textit{quiet}\}, \\
&\qquad AV = \textit{normal}, \\
&\qquad VF\ = \textit{quiet}.
\end{aligned}
$$

Let us illustrate diagnostic reasoning with the compressed rules by an example. Suppose that given are values of three ECG features, *P_wave = abnormal, PR_interval = shortened*, and *QRS_complex = normal.* A combination of the three corresponding rules defined above yields:

$$
\begin{array}{ll}
\textbf{if} & \mathit{P\_wave} = \mathit{abnormal} \ \wedge \ \mathit{PR\_interval} = \mathit{shortened} \ \wedge \ \mathit{QRS\_complex} = \mathit{normal} \\
\textbf{then} & \mathit{SA} = \mathit{quiet} \ \wedge \\
 & \mathit{AF} = \mathit{at} \vee \mathit{aeb} \vee \mathit{quiet} \ \wedge \\
 & \mathit{AV} = \mathit{lgl} \ \wedge \\
 & \mathit{BB} = \mathit{normal} \ \wedge \\
 & \mathit{VF} = \mathit{quiet} \\
 & \vee \\
 & \mathit{SA} = \mathit{quiet} \ \wedge \\
 & \mathit{AF} = \mathit{at} \vee \mathit{aeb} \vee \mathit{quiet} \ \wedge \\
 & \mathit{AV} = \mathit{normal} \ \wedge \\
 & \mathit{BB} = \mathit{normal} \ \wedge \\
 & \mathit{VF} = \mathit{quiet}
\end{array}
$$

The set of possible diagnoses is now restricted and only two heart components, *JF* and *VEF* are still unconstrained. The following logic program specifies the diagnostic algorithm:

```
diag( ⟨E_1, ..., E_10⟩, Arr )   ←
    if_then_1( E_1, Arr ),
        ⋮
    if_then_10( E_10, Arr ),
    possible( Arr ).
```

A diagnosis is an intersection of the consequents of if-then rules for individual ECG features, filtered through the *possible* which eliminate physiologically impossible and medically uninteresting arrhythmias. In the actual implementation, however, the algorithm computes with domains of variables (and not just ground values), and intersects disjunctive consequents through the breath-first search (unlike the standard Prolog interpreter which implements the depth-first search).

The application of learning to the 10 sets of examples required 40 hours of CPU time on SUN 2 (Mozetic 1986). The compressed rules occupy 30 times less space than the relational table, and can be used for efficient diagnosis. The reduction is due to the generalization of arrhythmia descriptions in the process of learning. The equivalency to the relational table is regained by the application of *possible* at the end of the diagnostic algorithm. In general, however, a relational table and the corresponding set of compressed rules are not equivalent. The difference is due to the projection of relational table entries to learning examples. The conditions under which both, a relational table and compressed rules produce equivalent diagnostic results are stated in (Bratko *et al.* 1989).

4 Comparison of complexity and efficiency

Tables 1 and 2 outline the complexity of the hierarchical model of the heart at each level of detail. For comparison, the complexity of the KARDIO model is given when possible. Table 1 comprises the number of components and the complexity of the arrhythmia descriptions. n denotes the arity of Arr tuples, $\mid A_1 \times \ldots \times A_n \mid$ is the number of syntactically possible descriptions, $\mid Arr \mid$ is the number of arrhythmias which satisfy the model constraints, and $\mid \neg Arr' \mid$ is the number of arrhythmias with no abstraction.

Level of	Model	Arrhythmias				
detail	components	Single	n	$\mid A_1 \times \ldots \times A_n \mid$	$\mid Arr \mid$	$\mid \neg Arr' \mid$
1	2	3	1	3	3	3
2	9	8	3	48	18	3
3	16	24	6	10,080	175	26
4	17	29	7	52,920	943	0
KARDIO	/	30	7	79,380	2,419	/

Table 1: The complexity of the heart model and the arrhythmia descriptions at different levels of detail, and in KARDIO.

Level of	ECG patterns				Relational table entries	
detail	m	$\mid E_1 \times \ldots \times E_m \mid$	$\mid ECG \mid$	$\mid \neg ECG' \mid$	$\mid \langle Arr, ECG \rangle \mid$	$\mid \neg \langle Arr, ECG \rangle' \mid$
1	1	3	3	3	3	3
2	4	64	12	0	23	5
3	7	41,472	263	6	333	79
4	10	3,386,880	3,096	0	5,240	0
KARDIO	7-19	/	/	/	140,966	/

Table 2: The complexity of ECG descriptions and the relational table at different levels of detail, and in KARDIO.

In Table 2 the complexity of ECG descriptions and the relational table are given. m denotes the arity of ECG tuples, $\mid E_1 \times \ldots \times E_m \mid$ is the number of syntactically possible ECG patterns, $\mid ECG \mid$ is the number of distinct ECG patterns derived from the model, and $\mid \neg ECG' \mid$ is the number of ECG patterns with no abstraction. $\mid \langle Arr, ECG \rangle \mid$ is the number of entries in the relational table, and $\mid \neg \langle Arr, ECG \rangle' \mid$ is the number of entries with no abstraction.

Tables 1 and 2 indicate that the heart model at the levels 1 and 2 is incomplete with respect to the levels 2 and 3, respectively. The level 3 model is complete with respect to the level

Type of diagnostic knowledge		Space (Kb)	Time (sec)
(1)	One-level model	15	50.35
(2)	Hierarchical four-level model	45	2.67
(3)	Relational table	750	0.22
(4)	Compressed diagnostic rules	25	0.55

Table 3: Space requirements for different representations and the average time needed to find all possible diagnoses for a given ECG pattern at the detailed level.

4. Recall that in the case of incompleteness, the hierarchical diagnostic algorithm has to resort to the naive generate-and-test method, thus potentially decreasing the efficiency of diagnosis. First experiments with the three-level model of the heart (Bratko *et al.* 1989) showed no considerable advantage of hierarchical diagnosis over the generate-and-test method, due exactly to the high level of incompleteness in the model. Consequently, the heart model at the level 2 was modified to decrease its incompleteness. Further, for all arrhythmias *Arr* without abstraction $(\neg Arr')$ the hierarchical diagnostic algorithm resorted to the corresponding relational table entries $\langle Arr, ECG \rangle$ in order to avoid the repetitive generate-and-test.

We compared the space complexity and diagnostic efficiency of the four types of diagnostic knowledge: (1) one-level model of the heart, (2) hierarchical four-level model, (3) relational table, and (4) compressed diagnostic rules. In all cases, knowledge bases and diagnostic algorithms are implemented as logic programs and compiled by Quintus Prolog. Complexity is measured by the space requirements of each representation together with the corresponding algorithm, when both stored as text files. Diagnostic efficiency is the time needed to find all possible diagnoses for a given ECG, and was measured on all 3096 distinct ECG patterns at the detailed level. Results in Table 3 are the average times over 3096 ECGs.

With the one-level model of the heart, three constraint propagation strategies were applied: inverting the order of constraints, forward checking, and naive generate-and-test with chronological backtracking. Somehow surprisingly, the generate-and-test method turned out to be the most efficient. This is due to the simulation nature and high directionality bias of the model. When the model is used in the 'forward' direction, the average time to derive an *ECG* for a given *Arr* is only 0.063 seconds. This is consistent with the 50.35 seconds from Table 3 where the model is applied 943 times in the 'forward' direction, once for each distinct *Arr*. In contrast, the model application in the 'backward' direction (from a given *ECG* to *Arr*) with inverted order of constraints, requires as much as 66.30 seconds on the average. The application of the forward checking technique was completely unsuccessful, probably due to the high arity of constraints (between 6 and 15 arguments) in the model definition.

Figure 4: A tradeoff between space complexity and diagnostic efficiency for different representations: (1) one-level heart model, (2) hierarchical four-level model, (3) relational table, and (4) compressed diagnostic rules.

The relational table representation is the most time efficient since only a simple retrieval is required, but, on the other hand, it is more space demanding. Compressed diagnostic rules are optimal in terms of space and time efficiency and currently appear to be the best representation for the ECG interpretation. Finally, the four-level model is obviously outperformed by the compressed diagnostic rules, but still achieves satisfactory performance from the practical point of view. More importantly, it is 20 times more efficient than the one-level model, and requires only three times as much space (out of 45 Kb, 11 Kb are for the relational table entries without abstractions).

The relation between different representations of diagnostic knowledge is better illustared on a time/space tradeoff scale in Figure 4. Recall that (3) and (4) were automatically derived from (1) while (2) was constructed semiautomatically on top of (1).

5 Conclusion

We presented two approaches to an efficient application of a deep simulation model to diagnosis. First, by abstracting the model, and second, by compressing it into a set of surface diagnostic rules. We compared the deep and surface electrocardiographic knowledge representation in terms of space complexity and diagnostic efficiency. Compressed diagnostic rules are both space and time efficient, and can be derived automatically. Some of them are simple and have clear medical interpretation, but many are too complex to be easy to understand. In contrast to dedicated diagnostic rules, model-based reasoning offers better explanation facilities which can be even tuned to the desired level of detail. Further, the hierarchical diagnostic algorithm can be easily modified to accommodate diagnostic reasoning under time constraints, and to offer a tradeoff between diagnostic specificity and certainty.

Acknowledgments

Thanks to Ivan Bratko and Nada Lavrač for their collaboration in the KARDIO project, to Mario Veitl, M.D. for his comments on the paper, and to Robert Trappl for making some of this work possible. Experiments described in the paper were carried out during the author's stay at the Center for Artificial Intelligence, Department of Computer Science, George Mason University, USA.

References

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

Buchanan, B.G., Sullivan, J., Cheng, T. and Clearwater, S.H.: 1988, Simulation-assisted inductive learning. *Proc. 7th Natl. Conference on Artificial Intelligence, AAAI-88*, Saint Paul, MN, Morgan Kaufmann, pp. 552-557.

Clocksin, W.F. and Mellish, C.S.: 1984, *Programming in Prolog* (Second edition). Springer-Verlag, New York.

de Kleer, J.: 1976, Local methods for localizing faults in electronic circuits. MIT AI Memo 394, Cambridge, MA.

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

Goldman, M.J.: 1976, *Principles of Clinical Electrocardiography.* Lange Medical Publications, Los Altos, CA.

Michalski, R.S.: 1983, A theory and methodology of inductive learning. In: R. S. Michalski, J. G. Carbonell, T. M. Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach*, Tioga, Palo Alto, CA, pp. 83-134.

Michalski, R.S., Mozetic, I., Hong, J. and Lavrac, N.: 1986, The multi-purpose incremental learning system AQ15 and its testing application to three medical domains. *Proc. 6th Natl. Conference on Artificial Intelligence, AAAI-86*, Philadelphia, PA, Morgan Kaufmann, pp. 1041-1045.

Mozetic, I.: 1986, Knowledge extraction through learning from examples. In: T. M. Mitchell, J. G. Carbonell, R. S. Michalski (eds.), *Machine Learning: A Guide to Current Research*, Kluwer Academic Publishers, Boston, MA, pp. 227-231.

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

Mozetic, I.: 1990a, Abstractions in model-based diagnosis. Report TR-90-4, Austrian Research Institute for Artificial Intelligence, Vienna, Austria. *Proc. AAAI-90 Workshop on Automatic Generation of Abstractions and Approximations*, Boston, MA.

Mozetic, I.: 1990b, Reduction of diagnostic complexity through model abstractions. Report TR-90-10, Austrian Research Institute for Artificial Intelligence, Vienna, Austria. *Proc. 1990 Intl. Workshop on Principles of Diagnosis*, Stanford University, CA.

Pearce, D.A.: 1988, The induction of fault diagnosis systems from qualitative models. *Proc. 7th Natl. Conference on Artificial Intelligence, AAAI-88*, Saint Paul, MN, Morgan Kaufmann, pp. 353-357.

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

Shortliffe, E.H.: 1976, *Computer-Based Medical Consultation: MYCIN.* American Elsevier, New York.

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