Model-Based Diagnosis: An Overview*

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

Abstract

Diagnosis is an important application area of Artificial Intelligence. First generation expert diagnostic systems had exhibited difficulties which motivated the development of model-based reasoning techniques. Model-based diagnosis is the activity of locating malfunctioning components of a system solely on the basis of its structure and behavior. The paper gives a brief overview of the main concepts, problems, and research results in this area.

1 Introduction

Diagnosis is one of the earliest areas in which application of Artificial Intelligence techniques was attempted. The diagnosis of a system which behaves abnormally consists of locating those subsystems whose abnormal behavior accounts for the observed behavior. For example, a system being diagnosed might be a mechanical device exhibiting malfunction, or a human patient. There are two fundamentally different approaches to diagnostic reasoning.

In the first, heuristic approach, one attempts to codify diagnostic rules of thumb and past experience of human experts in a given domain. Representatives of this approach are diagnostic expert systems of the first generation, such as MYCIN [Shortliffe, 1976]. Here, diagnostic reasoning of human experts is being modeled, and diagnostic accuracy depends on the successful encoding of human experience. The structure of the real-world system being diagnosed is not explicitly represented, nor is its behavior being modeled. As a result, these diagnostic systems are mainly specialized and restricted to applications sufficiently covered by experience. This leads to high costs for developing, maintaining, and extending such systems and makes many potential areas of applications too expensive.

The second approach is often called diagnosis from the first principles, or model-based diagnosis, where one starts with a description (a model) of a real-world system. The earliest model-based diagnosis systems were developed by [de Kleer, 1976,

*Appears in *Advanced Topics in Artificial Intelligence* (V. Marik, O. Stepankova, R. Trappl, Eds.), pp. 419-430, Springer-Verlag (LNAI 617), 1992.

Genesereth, 1984, Davis, 1984]. A model explicitly represents the structure of the system, i.e., its constituent components and their connections. The diagnostic problem arises when an observation of the system's actual behavior conflicts with the system's expected behavior. The diagnostic task is to identify those system components which, when assumed to function abnormally, will account for the difference between the observed and expected system behavior. To solve the problem, model-based diagnosis relies solely on the system description and observations of its behavior. In particular, it does not use any heuristic information about the system failures.

This paper gives a brief overview of the main concepts and research results in the area of model-based diagnosis. Most relevant publications appear in the proceedings of the IJCAI, AAAI, and ECAI conferences, and in the AI Journal. Since the area received increasing attention in recent years, there is also a regular, specialized International Workshop on Principles of Diagnosis [1990, 1991]. Section 2 outlines different approaches to model-based diagnosis. In section 3 we illustrate how to model structure and behavior of a simple device. In section 4, computation of diagnoses and complexity issues are described. Section 5 addresses the complexity problems by abstracting a model and by model compilation.

2 Approaches to model-based diagnosis

There are two prevailing approaches to model-based diagnosis, consistency-based and abductive [Poole, 1989] which differ in the representation of knowledge about the normality and faults, and in how diagnoses are defined and computed.

Reiter [1987], exemplifying the *consistency-based* approach, defines a model as a pair $\langle \mathrm{SD}, \textsc{Comps} \rangle$. SD is the system description, and $\textsc{Comps}$, the system components, is a finite set of constants. A system description is a set of first-order sentences defining how the system components are connected and how they *normally* behave. A distinguished unary predicate $\textsc{Ab}$ whose intended meaning is 'abnormal' is used in a system description. An observation $\textsc{Obs}$ is a finite set of first-order sentences. A diagnosis Δ for (SD, $\textsc{Comps}$, $\textsc{Obs}$) is a minimal subset $\Delta \in \textsc{Comps}$ such that

$$\mathrm{SD} \cup \textsc{Obs} \cup \{\ \textsc{Ab}(\mathrm{c}) \mid c \in \Delta\ \} \cup \{\ \neg\textsc{Ab}(c) \mid c \in \textsc{Comps} - \Delta\ \}$$

is consistent. A direct generate-and-test mechanism which systematically generates subsets of $\textsc{Comps}$, with minimal cardinality first, is too inefficient for systems with large numbers of components. Instead, Reiter [1987] proposes a diagnostic method based on the concept of a conflict set, originally due to de Kleer [1976]. De Kleer and Williams [1987] have independently implemented General Diagnostic Engine (GDE) which effectively realizes the above ideas.

In the *abductive* approach [Poole, 1989], SD contains just different modes of behavior and does not distinguish between the normal and abnormal behavior. An abductive diagnosis is then a minimal set of assumptions which, together with SD entail $\textsc{Obs}$:

$$\mathrm{SD} \cup \{\ \textsc{Mode}(\mathrm{c}) \mid c \in \Delta\ \} \quad \models \quad \textsc{Obs}$$

An early abductive diagnostic algorithm, based on set covering, is given by [Reggia *et al.*, 1987]. Cox and Pietrzykowski [1987] extend the notion of diagnoses to causes, and define a cause as fundamental iff it is minimal, acceptable, nontrivial, and basic. They show that for extended diagnostic problems their causes contain more useful information than Reiter's diagnoses. Geffner and Pearl [1987] present an improved constraint-propagation algorithm for diagnosis, based on a probabilistic approach. Console and Torasso [1990] have shown how to incorporate normal behavior into the abductive framework. A comparison of five approaches to abductive diagnosis is in [Finin and Morris, 1989]. Poole [1989] relates the modeling and diagnostic assumptions of the consistency-based and abductive approach.

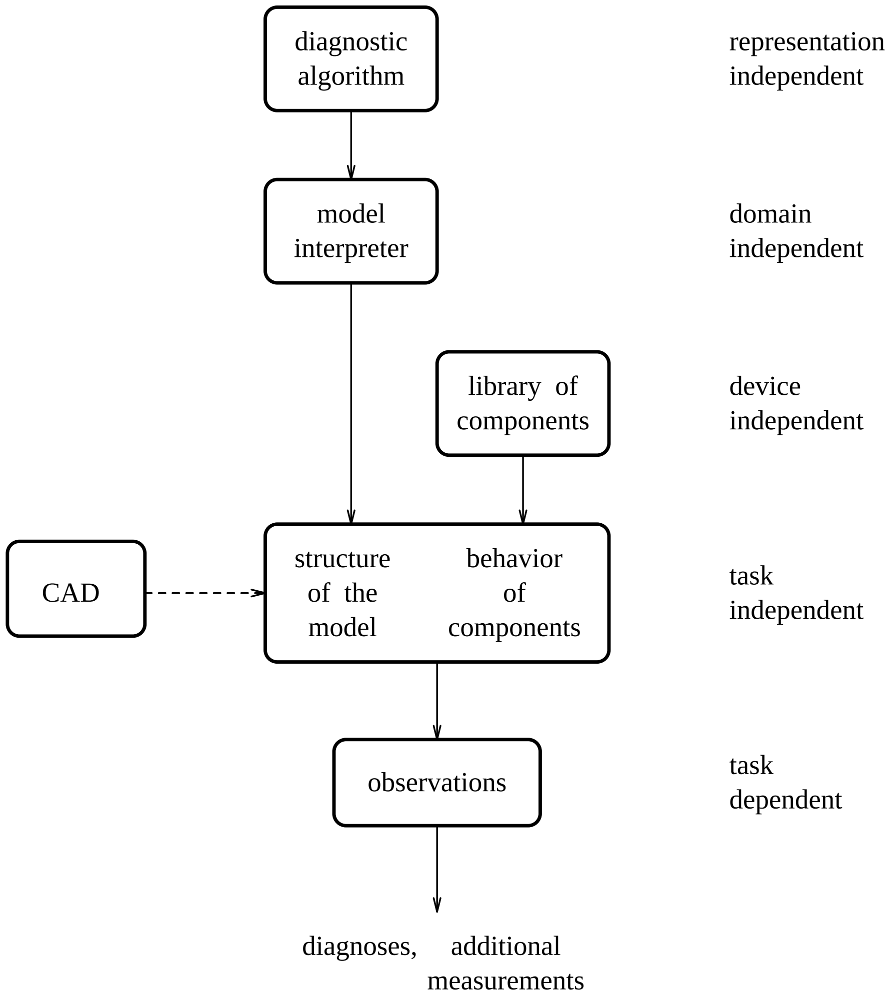

Figure 1: Architecture of a model-based diagnostic system.

Architecture of a typical model-based diagnostic system is illustrated in Figure 1. The idea is that a diagnostic algorithm is largely domain independent, and that it should be relatively easy to construct a model of a device. A model is defined by its structure, and behavior of the constituent components. The structure of a technical artifact is, at least in principle, available from a CAD system, while the behavior of components is drawn from a generic (domain dependent, but specific device independent) library. Such a model can then be used for prediction, control, monitoring, or diagnosis. As a result of diagnosis, alternative sets of faulty components are identified, or additional, discriminating measurements are proposed.

3 Modeling structure and behavior

A model interpreter is domain independent, but depends on the representation formalism chosen for modeling. Reasoning is typically based on theorem proving if a model is represented by first-order logic [Reiter, 1987], or on constraint propagation coupled with an ATMS [de Kleer and Williams, 1987]. Alternatively, one can represent and interpret models by *logic programs* [Lloyd, 1987], or by *constraint logic programs* [Jaffar *et al.*, 1986, Cohen, 1990]. The origin of this representation paradigm goes back to the KARDIO model [Bratko *et al.*, 1989]; similar representation was proposed by Saraswat *et al.* [1990]

Constraint Logic Programs are logic programs extended by interpreted functions. A proper implementation of the CLP scheme allows for an easy integration of specialized problem solvers into the logic programming framework. For example, in our implementation [Holzbaur, 1990] specialized solvers communicate with the standard SICStus Prolog [Carlsson and Widen, 1991] via extended semantic unification and are implemented in Prolog themselves. So far, three solvers have been implemented: constraint propagation over finite domains by forward checking [Van Hentenryck, 1989], CLP($\mathcal{B}$) — a solver over boolean expressions, and CLP($\Re$) — a solver for systems of linear equations and inequalities over $\Re$eals.

In the following we define the important diagnostic concepts in the logic programming framework, and illustrate them by a simple example.

Definition. A *model* of a system is a triple ⟨SD, COMPS, OBS⟩ where

1. SD, the system description, is a logic program with a distinguished top-level binary predicate m(COMPS, OBS).

2. COMPS, states of the system components, is an n-tuple $\langle S_1, \ldots, S_n \rangle$ where n is the number of components, and variables S_i denote states (e.g., normal or abnormal) of components.

3. OBS, observations, is an m-tuple $\langle In_1, \ldots, In_i, Out_{i+1}, \ldots, Out_m \rangle$ where In and Out denote inputs and outputs of the model, respectively.

Example (binary adder, [Genesereth, 1984], Figure 2).
The top-level binary predicate m is *adder*, COMPS is a five-tuple ⟨*X1,X2,A1,A2,O1*⟩, OBS is a five-tuple ⟨*A,B,C, D,E*⟩. SD consists of the following clause which specifies the structure of the adder, and of additional clauses which define behavior of the components:

adder(⟨*X1,X2,A1,A2,O1*⟩, ⟨*A,B,C, D,E*⟩) ←
  xorg(X1, A, B, X),
  xorg(X2, C, X, D),
  andg(A1, A, B, Y),
  andg(A2, C, X, Z),
  org(O1, Y, Z, E).

Connections between components are represented by shared variables. Specification of the behavior depends on the type of the fault model available: structural, weak, exoneration, or strong. For illustration we define just the behavior of an OR gate (*org*).

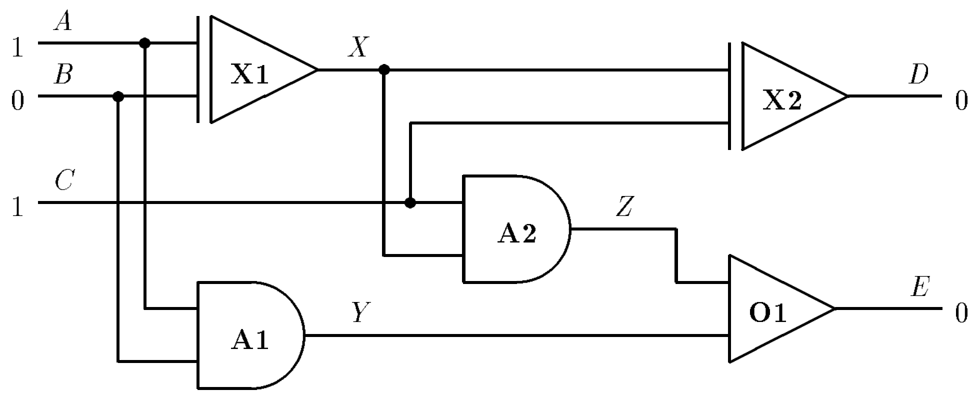

Figure 2: A binary adder consisting of two EXCLUSIVE-OR gates (X1, X2), two AND gates (A1, A2) and an OR gate (O1). The output $E = 0$ is faulty.

A **structural** model specifies the most general condition about the propagation of faults [Bakker *et al.*, 1989]. Only if all inputs to a component are correct (c) and the component is normal (ok) then the output is correct. In other words, given the observation that all inputs are correct (c) and the output is faulty (f) then the component is abnormal (ab). Otherwise, nothing can be concluded about the component state.

org(ok, c, c, c).
org(ok, c, f, _).
org(ok, f, _, _).
org(ab, _, _, _).

A **weak** fault model defines just normal behavior of components (state ok), abnormal behavior (state ab) is unconstrained:

$org(\ ok,\ X,\ Y,\ Z\ )\ \leftarrow\ or(\ X,\ Y,\ Z\ ).$
org(ab, _, _, _).

A **strong** fault model [Struss and Dressler, 1989] specifies all the possible ways in which a component can fail. In general, a component may have several failure states. An abnormal OR gate, for example, might have the output stuck-at-1 ($s1$) or stuck-at-0 ($s0$):

$org(\ ok,\ X,\ Y,\ Z\ )\ \leftarrow\ or(\ X,\ Y,\ Z\ ).$
org(s1, 0, 0, 1).
org(s0, 0, 1, 0).
org(s0, 1, 0, 0).
org(s0, 1, 1, 0).

The **exoneration** principle [Raiman, 1989] is a special case of a strong fault model. It specifies as abnormal any behavior different than normal:

$org(\ ab,\ X,\ Y,\ Z\ )\ \leftarrow\ \neg\ or(\ X,\ Y,\ Z\ ).$

In the case of combinatorial circuits, one can formulate different models in terms of boolean expressions in CLP($\mathcal{B}$) [Mozetič and Holzbaur, 1991b], instead of using extensional descriptions in pure Prolog. We can encode correct inputs and outputs and the normal state as 0 (zero) instead of *ok*, and the abnormal states and faulty inputs and outputs as 1. For brevity we only show the OR gate reformulations:

- structural model: *org(* $(\neg X \wedge \neg Y \wedge Z) \vee W, X, Y, Z$*).*
- weak model: *org(* $((X \vee Y) \oplus Z) \vee W, X, Y, Z$*).*
- exoneration model: *org(* $(X \vee Y) \oplus Z, X, Y, Z$*).*

There are domains where pure logic programs or ATMS-like systems have insufficient expressive power to reason about the system under consideration. In particular, modeling real-valued system parameters with tolerances requires some degree of numerical processing, and feedback loops in general cannot be resolved by local constraint propagation methods. Examples of such systems are analogue circuits, such as amplifiers or filters [Wakeling and McKeon, 1989]. Dague *et al.* [1990] use an ATMS-like system, augmented with the ability to compute with intervals, but unable to solve simultaneous equations, for the diagnosis of analogue circuits. The first application of CLP($\Re$) to the analysis of analogue circuits was reported by [Heintze *et al.*, 1987]. In [Mozetič *et al.*, 1991] we show how to apply CLP($\Re$) to diagnose analog circuits operating under the AC conditions. CLP($\Re$) enables modeling of soft faults — drifts from the nominal parameter values, and computation with parameter tolerances.

4 Computing diagnoses

In order to define the concepts of a diagnosis and a conflict, we assume that an observation, a ground instance of OBS, is given. In the following definitions $\forall F$ denotes universal closure, i.e., all free variables in the formula F are universally quantified.

Definition. An *ok-instance* of a term is an instance where some variables are replaced by the constant *ok*. A *ground instance* is an instance where all the variables are replaced by constants.

Definition. A *diagnosis* D for $\langle$SD, COMPS, OBS$\rangle$ is an instance of COMPS such that SD $\models \forall \mathrm{m}(D, \mathrm{OBS})$.

Definition. A *conflict* C for $\langle$SD, COMPS, OBS$\rangle$ is an *ok-instance* of COMPS such that SD $\models \forall \neg \mathrm{m}(C, \mathrm{OBS})$.

This characterization of a diagnosis subsumes most of the previous ones. In the consistency-based approach [Reiter, 1987] a diagnosis is a set of abnormal ($\neq$*ok*) components such that SD and OBS are consistent with all other components being *ok*. De Kleer and Williams [1989] extended the definition to include a behavioral mode (state) for each component. In both cases a diagnosis is essentially a ground instance of COMPS. Poole [1989] observed that a diagnosis need not commit a state to each component when the state is 'don't care'. This led to the definition of a partial diagnosis [de Kleer *et al.*, 1990] which corresponds to a non-ground instance of COMPS but, on the other hand, does not include states of components. The definition of a conflict is standard, i.e., a set of components which cannot be simultaneously *ok*, and can be easily extended to a minimal conflict. A *minimal* diagnosis is a diagnosis which is subsumed by no other diagnosis.

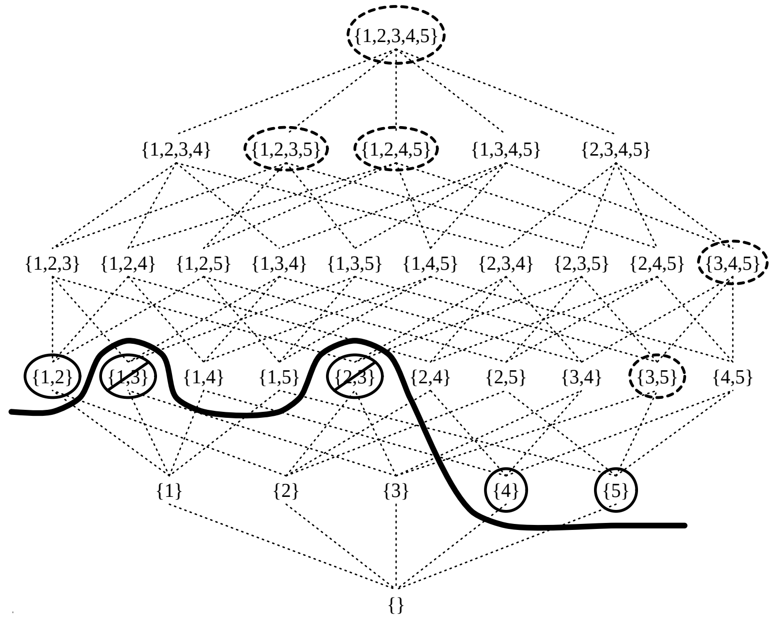

Figure 3: Search lattice for the adder model, given the observation ⟨1,0,1, 0,0⟩. ⊘ denote minimal conflicts, and ◯ minimal diagnoses. For the weak fault model all supersets of a minimal diagnosis are also diagnoses. For the strong fault model only nodes in dashed ovals denote non-minimal diagnoses.

In order to compute minimal diagnoses it is convenient to represent the search space of diagnoses and conflicts as a subset-superset lattice [de Kleer and Williams, 1987]. The top element of the lattice corresponds to a tuple where all components are $\neq ok$, and the bottom element to the tuple where all components are *ok*. Reiter's algorithm [1987], for example, searches the lattice bottom-up (from conflicts to diagnoses), in a breadth-first fashion (diagnoses of smaller cardinality are found first) and relies on the underlying ATMS-like theorem prover. In contrast, our algorithm [Mozetič, 1992] better fits into the logic programming environment, and implements a top-down, depth-first search through the lattice. In diagnosing real systems, the search lattice is large and minimal diagnoses are usually near the bottom of the lattice. Depth-first search, coupled with non-ground model calls, allows for deep 'dives' into the lattice and in the average case at least a few diagnoses are found quickly. Further, by computing minimal diagnoses incrementally, we can ensure that the worst case complexity of the algorithm remains polynomial.

The task of finding all minimal diagnoses is NP-complete, i.e., attempting to compute *all* minimal diagnosis is asking for more information than one could ever hope to use. Friedrich *et al.* [1990a] show that even the *next diagnosis* problem is intractable. This holds for a weak fault model, and with a strong fault model things get worse since then even deciding whether an *arbitrary diagnosis* exists is NP-complete. This has been shown in the context of abductive diagnosis [Bylander *et al.*, 1989], and in the context of consistency-based diagnosis [Friedrich *et al.*, 1990a]. On the other hand, computing minimal conflicts from the structural model can be done in polynomial time [Bakker *et al.*, 1989].

One solution to the complexity problem is to compute just some minimal diagnoses. De Kleer [1991] presents a focusing mechanism which enables efficient computation of a few most probable diagnoses. A polynomial algorithm for computing the first k diagnoses is in [Mozetič, 1992]. Another solution is to compute just one minimal diagnosis, and to interleave diagnosis and treatment [Friedrich *et al.*, 1990b]. Finally, complexity can be reduced by introducing abstractions [Gallanti *et al.*, 1989, Mozetič, 1990, Mozetič, 1991].

5 Abstractions and model compilation

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. Abstractions turned out to be useful in reducing the search space in theorem proving [Plaisted, 1981, Giunchiglia and Walsh, 1989], planning [Sacerdoti, 1974], and in model-based diagnosis [Gallanti *et al.*, 1989, Mozetič, 1990, Mozetič, 1991].

The following are three abstraction operators which can be used in a multi-level model representation.

- Collapse of values — indistinguishable values of a variable can be abstracted into a single value.
- Deletion of variables — irrelevant variables can be deleted at the abstract level.
- Simplification of the mapping m — detailed level mapping m can be simplified to m' by ignoring and/or simplifying some model components.

Given a detailed (possibly numerical) model, abstraction operators can be used to automatically derive an abstract, qualitative model of the system [Mozetič and Holzbaur, 1991c]. The abstract level model is then used to guide the diagnostic process at the detailed level.

Another, *indirect* approach to use a deep model for efficient diagnosis is to 'compile' it into surface diagnostic rules. This was first applied in the context of the KARDIO model for ECG diagnosis of cardiac arrhythmias [Mozetič, 1986, Bratko *et al.*, 1989]. 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 [Michalski, 1983], and the table is compressed into a set of simple if-then rules.

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.

6 Conclusion

The paper gives a brief overview of the area of model-based diagnosis. Current research trends aim at focusing diagnosis for a specific purpose, e.g., to reconstruct the functionality for which the system was originally designed [Friedrich *et al.*, 1990b]. Another interesting issue is the combination of multiple models, like physical and functional organization, and the common ways the components fail [Hamscher, 1991]. A really hard problem of diagnosis of transient failures and dynamical systems was addressed only recently [Friedrich and Lackinger, 1991]. The emerging model-based reasoning technology rises hopes that an increasing number of the techniques will find their application in practice.

Acknowledgements

This work was supported by the Austrian Federal Ministry of Science and Research. Thanks to Robert Trappl for making it possible.

References

[Bakker *et al.*, 1989] Bakker, R.R., Hogenhuis, P.A., Mars, N.J.I., van Soest, D.C. Diagnosis of technical systems using design descriptions. *Proc. Second Generation Expert Systems*, pp. 107-116, Avignon, France, 1989.

[Bratko *et al.*, 1989] Bratko, I., Mozetič, I., Lavrač, N. *KARDIO: A Study in Deep and Qualitative Knowledge for Expert Systems.* MIT Press, Cambridge, MA, 1989.

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

[Bylander *et al.*, 1989] Bylander, T., Allemang, D., Tanner, M.C., Josephson, J.R. Some results concerning the computational complexity of abduction. *Proc. Intl. Conf. on Principles of Knowledge Representation and Reasoning*, pp. 44-54, Toronto, Morgan Kaufmann, 1989.

[Carlsson and Widen, 1991] Carlsson, M., Widen, J. SICStus Prolog User's Manual. Swedish Institute of Computer Science, Kista, Sweden, 1991.

[Cohen, 1990] Cohen, J. Constraint logic programming languages. *Communications of the ACM 33 (7)*, pp. 52-68, 1990.

[Console and Torasso, 1990] Console, L., Torasso, P. Integrating models of the correct behavior into abductive diagnosis. *Proc. 9th European Conf. on Artificial Intelligence, ECAI-90*, pp. 160-166, Stockholm, Sweden, 1990.

[Cox and Pietrzykowski, 1987] Cox, P.T., Pietrzykowski, T. General diagnosis by abductive inference. *Proc. 1987 Symposium on Logic Programming*, pp. 183-189, San Francisco, CA, IEEE, 1987.

[Dague *et al.*, 1990] Dague, P., Deves, P., Luciani, P., Taillibert, P. Analog systems diagnosis. *Proc. 9th European Conf. on Artificial Intelligence, ECAI-90*, pp. 173-178, Stockholm, Sweden, 1990.

[Davis, 1984] Davis, R. Diagnostic reasoning based on structure and behavior. *Artificial Intelligence 24*, pp. 347-410, 1984.

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

[de Kleer, 1991] de Kleer, J. Focusing on probable diagnoses. *Proc. 9th Natl. Conf. on Artificial Intelligence, AAAI-91*, pp. 842-848, Anaheim, CA, MIT Press, 1991.

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

[de Kleer and Williams, 1989] de Kleer, J., Williams, B.C. Diagnosis with behavioral modes. *Proc. 11th Intl. Joint Conf. on Artificial Intelligence, IJCAI-89*, pp. 1324-1330, Detroit, Morgan Kaufmann, 1989.

[de Kleer *et al.*, 1990] de Kleer, J., Mackworth, A.K., Reiter, R. Characterizing diagnoses. *Proc. 8th Natl. Conf. on Artificial Intelligence, AAAI-90*, pp. 324-330, Boston, MIT Press, 1990.

[Finin and Morris, 1989] Finin, T., Morris, G. Abductive reasoning in multiple fault diagnosis. *Artificial Intelligence Review 3*, pp. 129-158, 1989.

[Friedrich *et al.*, 1990a] Friedrich, G., Gottlob, G., Nejdl, W. Physical impossibility instead of fault models. *Proc. 8th Natl. Conf. on Artificial Intelligence, AAAI-90*, pp. 331-336, Boston, MIT Press, 1990.

[Friedrich *et al.*, 1990b] Friedrich, G., Gottlob, G., Nejdl, W. Hypothesis classification, abductive diagnosis and therapy. *Proc. First Intl. Workshop on Principles of Diagnosis*, pp. 124-128, Stanford University, Palo Alto, 1990.

[Friedrich and Lackinger, 1991] Friedrich, G., Lackinger, F. Diagnosing temporal misbehavior. *Proc. 12th Intl. Joint Conf. on Artificial Intelligence, IJCAI-91*, pp. 1116-1122, Sydney, Australia, Morgan Kaufmann, 1991.

[Gallanti *et al.*, 1989] Gallanti, M., Roncato, M., Stefanini, A., Tornielli, G. A diagnostic algorithm based on models at different level of abstraction. *Proc. 11th Intl. Joint Conf. on Artificial Intelligence, IJCAI-89*, pp. 1350-1355, Detroit, Morgan Kaufmann, 1989.

[Geffner and Pearl, 1987] Geffner, H., Pearl, J. An improved constraint-propagation algorithm for diagnosis. *Proc. 10th Intl. Joint Conf. on Artificial Intelligence, IJCAI-87*, pp. 1105-1111, Milan, Italy, Morgan Kaufmann, 1987.

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

[Giunchiglia and Walsh, 1989] Giunchiglia, F., Walsh, T. Abstract theorem proving. *Proc. 11th Intl. Joint Conf. on Artificial Intelligence, IJCAI-89*, pp. 372-377, Detroit, MI, Morgan Kaufmann, 1989.

[Hamscher, 1991] Hamscher, W.C. Modeling digital circuits for troubleshooting. *Artificial Intelligence 51*, pp. 223-271, 1991.

[Heintze *et al.*, 1987] Heintze, N., Michaylov, S., Stuckey, P. CLP($\Re$) and some electrical engineering problems. *Proc. 4th Intl. Conf. on Logic Programming*, pp. 675-703, Melbourne, Australia, MIT Press, 1987.

[Holzbaur, 1990] Holzbaur, C. Specification of constraint based inference mechanisms through extended unification. Ph.D. thesis, Dept. of Medical Cybernetics and AI, University of Vienna, Austria, 1990.

[IWPD, 1990] First Intl. Workshop on Principles of Diagnosis, Working notes, Stanford University, Palo Alto, July 23-25, 1990.

[IWPD, 1991] Second Intl. Workshop on Principles of Diagnosis, Working notes, Milano, Italy, October 14-16, 1991.

[Jaffar *et al.*, 1986] Jaffar, J., Lassez, J.-L., Mahler, J. A logic programming language scheme. In: D. de Groot, G. Linstrom (eds.), *Logic Programming: Functions, Relations, and Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1986.

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

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

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

[Mozetič, 1990] Mozetič, I. Reduction of diagnostic complexity through model abstractions. Report TR-90-10, Austrian Research Institute for Artificial Intelligence, Vienna. *Proc. First Intl. Workshop on Principles of Diagnosis*, pp. 102-111, Stanford University, Palo Alto, 1990.

[Mozetič, 1991] Mozetič, I. Hierarchical model-based diagnosis. *Intl. Journal of Man-Machine Studies 35 (3)*, pp. 329-362, 1991.

[Mozetič, 1992] Mozetič, I. A polynomial-time algorithm for model-based diagnosis. Proc. *European Conf. on Artificial Intelligence, ECAI-92*, Vienna, Austria, 1992.

[Mozetič and Holzbaur, 1991b] Mozetič, I., Holzbaur, C. Model-based diagnosis with constraint logic programs. *Proc. 7th Austrian Conf. on Artificial Intelligence, ÖGAI-91*, pp. 168-180, Wien, Springer-Verlag, 1991.

[Mozetič and Holzbaur, 1991c] Mozetič, I., Holzbaur, C. Integrating numerical and qualitative models within constraint logic programming. *Proc. 1991 Intl. Logic Programming Symposium, ILPS-91*, pp. 678-693, San Diego, CA, MIT Press, 1991.

[Mozetič *et al.*, 1991] Mozetič, I., Holzbaur, C., Novak, F., Santo-Zarnik, M. Model-based analogue circuit diagnosis with CLP($\Re$). *Proc. 4th Intl. GI Congress*, pp. 343-353, München, Springer-Verlag, 1991.

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

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

[Poole, 1989] Poole, D. Normality and faults in logic-based diagnosis. *Proc. 11th Intl. Joint Conf. on Artificial Intelligence, IJCAI-89*, pp. 1304-1310, Detroit, Morgan Kaufmann, 1989.

[Raiman, 1989] Raiman, O. Diagnosis as a trial: the alibi principle. Report, IBM Scientific Center, Paris, 1989.

[Reggia *et al.*, 1987] Reggia, J.A., Nau, D.S., Wang, P.Y. Diagnostic expert systems based on a set covering model. *Intl. Journal of Man-Machine Studies 32*, pp. 57-96, 1987.

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

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

[Saraswat *et al.*, 1990] Saraswat, V.A., de Kleer, J., Raiman, O. Contributions to a theory of diagnosis. *Proc. First Intl. Workshop on Principles of Diagnosis*, pp. 33-38, Stanford University, Palo Alto, 1990.

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

[Struss and Dressler, 1989] Struss, P., Dressler, O. "Physical negation" — integrating fault models into the general diagnostic engine. *Proc. 11th Intl. Joint Conf. on Artificial Intelligence, IJCAI-89*, pp. 1318-1323, Detroit, Morgan Kaufmann, 1989.

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

[Wakeling and McKeon, 1989] Wakeling, A., McKeon A. On automatic fault finding in analogue circuits. *Electronic Engineering*, pp. 95-101, Nov. 1989.