

Computing Minimal Diagnoses with Critical Set Algorithms*

Igor Mozetič

Department of Information Systems

Technical University of Vienna

and (mailing address)

Austrian Research Institute for Artificial Intelligence

Schottengasse 3, A-1010 Vienna

Austria

Email: igor@ai.univie.ac.at

Fax: (+43 1) 532 0652

Phone: (+43 1) 533 6112

Abstract

The paper is concerned with the time complexity of model-based diagnosis. Our experiments indicate that the time to compute minimal diagnoses is dominated by the calls to the model of the device being diagnosed. In the paper we describe an attempt to reduce the number of model calls by incorporating two critical set algorithms [Loveland, 1987] into IDA [Mozetič, 1992]. A critical set algorithm computes a minimal diagnosis with $O(\log n)$ model calls as opposed to $O(n)$ model calls made by a straightforward algorithm. We performed experiments on two non-trivial domains: (a) a 1000-bit adder which has simple structure and behaviour, but large number of components (5000) and minimal diagnoses, and (b) the KARDIO model of the heart with complicated structure and behaviour, but relatively small search space and few minimal diagnoses. The reported results are negative: the straightforward algorithm outperforms more sophisticated critical set algorithms. We analyse the results and show that both critical set algorithms are suboptimal in the number of failed model calls which dominate the total number of model calls and consequently the overall diagnostic time.

*Appears in *Proc. 11th European Conf. on Artificial Intelligence, ECAI-94*, pp. 657–661, Amsterdam, August 8–12, John Wiley & Sons, 1994.

1 Introduction

Model-based diagnosis is the activity of locating malfunctioning components of a system on the basis of the system’s model and observations of its behaviour. In the paper we consider just the task of computing minimal sets of broken components (i.e., diagnoses) which account for the observations different then expected. We ignore the tasks of specifying appropriate test vectors, proposing additional, discriminating measurements, or interleaving the diagnostic process with repair.

When searching for minimal diagnoses it is convenient to represent the search space of potential diagnoses as a subset-superset lattice. The top element of the lattice corresponds to all components being broken, and the bottom element to all components being ok. The diagnostic process consists of the lattice search, interleaved with calls to the model of the system (consistency checks). We can identify two sources of exponential worst-case complexity in the number of components: the set operations performed during the lattice search, and individual model calls which depend on the type of the fault model used. A ‘weak’ fault model (where only normal behaviour is considered) can be specified by Horn clauses with bounded term depth and arity. Such a model can be checked for consistency with observations in polynomial time. [Friedrich *et al.*, 1990] give an algorithm to compute a single minimal diagnosis with a ‘weak’ fault model in polynomial time. However, they show that, due to the set operations performed during the lattice search, finding the next minimal diagnosis is an NP-complete problem. [Mozetič, 1992] gives an improved incremental algorithm IDA which computes k minimal diagnoses in polynomial time. With a ‘strong’ fault model (when abnormal behaviour is also specified) even computing the first (non-minimal) diagnosis is NP-complete [Bylander *et al.*, 1989].

In practice, the exponential growth of the lattice search is kept under control by focusing on probable diagnoses [de Kleer, 1991], or by keeping k low as in IDA. The cost of an individual model call can be reduced by model abstractions [Mozetič, 1991], by efficient constraint propagation or by model compilation [Mozetič and Pfahringer, 1992].

A factor which is never critical in the worst-case time analysis is the *number* of model calls. In practice, however, model calls dominate the overall diagnostic time. Assuming that individual model calls are tractable, we can reduce the overall diagnostic time by reducing the total number of model calls. In the paper we describe such an attempt by incorporating two critical set algorithms, proposed by [Loveland, 1987] into IDA. The fact that a critical set algorithm can be used to compute a minimal diagnosis was first observed by [Childress and Valtorta, 1993]. However, they used an unmodified critical set algorithm which works just with a weak fault model. We modified both algorithms so that they can be used with any model, and did an extensive evaluation on two non-trivial domains.

In section 2 we briefly describe the IDA algorithm, its relation to consistency-based diagnosis, and an extended function which implements calls to the underlying model. In section 3 we describe two critical set algorithms which compute a minimal diagnosis with $O(\log n)$ model calls as opposed to $O(n)$ model calls made by a straightforward algorithm. Section 4 gives experimental results on two very different testing domains: (a) a 1000-bit adder which has simple structure and behaviour, but large number of components (5000)

and minimal diagnoses, and (b) the KARDIO model of the heart [Bratko *et al.*, 1989] with complicated structure and behaviour, but relatively small search space and few minimal diagnoses. The reported results are negative: the straightforward algorithm outperforms more sophisticated critical set algorithms. In section 5 we analyse the results and show that both critical set algorithms are suboptimal in the number of failed model calls which dominate the total number of model calls and consequently the overall diagnostic time.

2 Computing minimal diagnoses

In consistency-based approach to model-based diagnosis [Reiter, 1987], a triple (SD, COMPS, OBS) is given. SD is the system description, COMPS is the set of the system components, and OBS are observations. SD defines connections between the system components and their normal behaviour by using a distinguished unary predicate AB (meaning ‘abnormal’). A *diagnosis* D for (SD, COMPS, OBS) is a subset $D \subseteq \text{COMPS}$ such that

$$\text{SD} \cup \text{OBS} \cup \{ \text{AB}(c) \mid c \in D \} \cup \{ \neg \text{AB}(c) \mid c \in \text{COMPS} - D \}$$

is consistent. A diagnosis is *minimal* iff no proper subset of it is also a diagnosis.

The above consistency test effectively defines a binary function f from a powerset of COMPS to $\{0,1\}$. If the above expression is consistent (i.e., D is a diagnosis) then $f(D) = 1$, otherwise $f(D) = 0$. For a binary function which satisfies certain conditions, Loveland [1987] defined several algorithms which can be used to compute a minimal diagnosis [Childress and Valtorta, 1993].

According to [Loveland, 1987], a binary function f defined on the power set of U is *monotone* if

$$f(S) = 1 \Rightarrow f(S_1) = 1, \quad \forall S_1 : S_1 \supseteq S.$$

A *critical set* of a binary monotone function f is a set S such that

$$f(S) = 1 \wedge f(S_2) = 0, \quad \forall S_2 : S_2 \subset S.$$

When $U = \text{COMPS}$ and the consistency test is used for f then a critical set corresponds exactly to a minimal diagnosis. However, f is monotone only when each superset of a diagnosis is also a diagnosis. This requirement is in general fulfilled only by SD which describes just *normal* behaviour of components, i.e., by a weak fault model [de Kleer *et al.*, 1992]. With exoneration or strong fault models which characterize also *abnormal* behavior the function f might not be monotone.

IDA [Mozetič, 1992] implements a deductive approach to model-based diagnosis. It works with weak, exoneration, or strong fault models. SD is a logic program containing a distinguished binary predicate $m(\text{COMPS}, \text{OBS})$ which represents a model and relates states of components to observations. A diagnosis D for (SD, COMPS, OBS) is defined as an instance of COMPS such that $\text{SD} \models m(D, \text{OBS})$. A logic programming system is used to call the model and to find an assignment of states to D such that this assignment is ‘consistent’ with SD and OBS. The main difference to the consistency-based approach is

that the model call is not just a binary test, but a function from a powerset of COMPS to a diagnosis D or to 0. Given a partial assignment of states to components (a subset S of the universe U), the function TP returns a diagnosis D if there exists a complete assignment, or 0 if S is inconsistent with SD and OBS, and the model call fails:

$$\begin{aligned} TP(S) &= D \text{ if } \exists D : (D \subseteq S) \wedge \text{SD} \models m(D, \text{OBS}) \\ TP(S) &= 0 \text{ otherwise.} \end{aligned}$$

Function TP thus effectively implements the *theorem prover* function, originally proposed by Reiter [1987]. The only difference is that the role of conflicts is replaced by diagnoses. IDA thus computes minimal diagnoses directly from diagnoses, and not via conflicts as in most consistency-based approaches.

IDA consists of two main procedures which clearly separate the search through the lattice from the model calls. The top-level, *All_diags* procedure starts with an initial set of k minimal diagnoses and computes the next, $k + 1$ -s diagnosis. A candidate S for the next diagnosis is computed as a complement of a hitting set of previous diagnoses. If $TP(S) = 0$ then another candidate is computed, and if $TP(S) = D$ then the *Min_diag* procedure is invoked. The *Min_diag* procedure searches the sub-lattice under a diagnosis D and returns a minimal diagnosis. For a fixed k it was shown that IDA computes the next $k + 1$ -st minimal diagnosis in polynomial time, i.e., in $O(n^{2k})$ set operations and in $O(n)$ model calls [Mozetič, 1992]. Since for a reasonable k model calls dominate the overall diagnostic time we tried to improve the *Min_diag* procedure which uses a straightforward algorithm by more sophisticated critical set algorithms as proposed by Loveland [1987].

3 Critical set algorithms

Loveland [1987] proposed five algorithms to compute a critical set, called *Algorithm I, II, II', III, IV*. As already noted in [Childress and Valtorta, 1993], *Algorithm II* is incorrect. From the remaining four, we implemented and tested *Algorithms II'* and *IV* which were the most promising according to Loveland. The algorithms had to be modified due to the use of a more powerful function TP instead of a binary monotone function f .

We use the following notation: U denotes the universe (a non-minimal diagnosis with which the *Min_diag* procedure is invoked) with cardinality n , C retains known members of the critical set being isolated (a minimal diagnosis with cardinality r), A includes at least one element of the critical set, and D is a (non-minimal) diagnosis. Whenever A is non-empty then A must contain members of the critical set being isolated. It is assumed that at least one critical set exists.

First we give a naive algorithm using function f . The algorithm tests $f(U - \{c\})$ for each element of the universe U , performing $U := U - \{c\}$ whenever the function evaluation is 1. The resulting U is a critical set, and the algorithm takes n function calls. This algorithm was used by [Friedrich *et al.*, 1990].

Algorithm 0 is a straightforward modification when TP is used instead of f , and has a structure similar to Loveland's *Algorithms II'* and *IV*. It was used in IDA in the original

Min_diag procedure.

Algorithm 0

- (1) $C := \{\}$, $A := U$
- (2) Remove an element c from A , $A_1 := A - \{c\}$
- (3) If $TP(C \cup A_1) = D \neq 0$ then $A := D - C$ ($A \subseteq A_1$)
 else $A := \{c\}$
- (4) If $|A| > 1$ then return to step (2)
 else $C := A \cup C$, $A := A_1$
 if $A = \{\}$ then C is a critical set
 else return to step (2).

Algorithm 0 takes at most n and at least r model calls. Note that U might already be a minimal diagnosis, and that r model calls are then needed to prove that it is a minimal diagnosis indeed. This is optimal.

The main idea of the two Loveland's critical set algorithms is to use a binary search instead of the linear. *Algorithm II'* repeatedly finds one element of the critical set at a time, by continual binary splitting of the space under consideration. The variable R contains the remaining unknown members of the critical set being isolated.

Algorithm II'

- (1) $C := \{\}$, $A := U$, $R := \{\}$
- (2) Split A into roughly equal disjoint sets A_1 and A_2 such that $|A_1| \leq |A_2|$
- (3) If $TP(C \cup R \cup A_1) = D \neq 0$ then $A := D - C - R$ ($A \subseteq A_1$)
 else $R := A_1 \cup R$, $A := A_2$
- (4) If $|A| > 1$ then return to step (2)
 else $C := A \cup C$, $A := R$, $R := \{\}$
 if $A = \{\}$ then C is a critical set
 else if $TP(C) \neq 0$ then C is a critical set
 else return to step (2).

Algorithm II' has an upper bound of $r(1 + \log n)$ model calls, which follows directly from the depth-first nature of the algorithm.

Algorithm IV is similar to *Algorithm II'*, but checks both halves of the split, and introduces a stack to retain sets known to certain points (R is defined as the union of sets on the stack).

Algorithm IV

- (1) $C := \{\}$, $A := U$, *empty_stack_R* := *true*
- (2) Split A into roughly equal disjoint sets A_1 and A_2
- (3) If $TP(C \cup R \cup A_1) = D \neq 0$ then $A := D - C - R$ ($A \subseteq A_1$)
 else if $TP(C \cup R \cup A_2) = D \neq 0$ then $A := D - C - R$ ($A \subseteq A_2$)
 else *push_R*(A_1), $A := A_2$
- (4) If $|A| > 1$ then return to step (2)

else $C := A \cup C$
 if *empty_stack_R* then C is a critical set
 else *pop_R(A)* and return to step (4).

The number of function calls used by *Algorithm IV* is bounded by $2r \log n$, since for each element of the critical set at most $2 \log n$ model calls are needed.

4 Two case studies

The goal of the experimental evaluation was twofold:

- (a) to confirm that the time spend by model calls represents the majority of the overall diagnostic time, and
- (b) to find an algorithm which minimizes the number of model calls and consequently results in an efficient diagnostic procedure.

We used two testing domains for the evaluation (Table 1): (a) a 1000-bit ripple carry adder [de Kleer, 1991], and (b) the KARDIO model of the heart [Bratko *et al.*, 1989].

	1000-bit adder	KARDIO model
No. of components	5000	17 (7 with state)
States per component	3	5.14 in average
Diag. search space size	5000^3	52,920
Testing observations	1	3,096
Computed min. diags.	first 60	all (1.33 in average)
Cardinality of min. diags.	1-20	3.83 in average

Table 1: Characteristics of both experimental domains.

The 1000-bit adder model consists of 1000 strong models of a binary adder. The binary adder consists of five Boolean gates which can be in states *ok*, *stuck-at-0*, or *stuck-at-1*. The output $Out2_{i-1}$ of the adder $i - 1$ is connected to the input $In3_i$ of the adder i ($i = 2, \dots, 1000$). All inputs and outputs were set to 0, except for the $Out1_{1000}$ of the last adder which was set to 1 (this is the only faulty output). Note that the inputs are propagated through almost every gate of this circuit. The model has a regular structure, but consists of 5000 components! For the given input-output observation, the first 60 minimal diagnoses were computed. Their cardinality ranges from one (single faults) to 20.

The KARDIO model of the heart relates disorders in the electrical activity of the heart to ECG descriptions. The model consists of seven components which can be in different states and correspond to individual disorders. The remaining 10 components model underlying physiology of the heart and are assumed to be normal. There are 3096 different ECG descriptions which indicate a possible (single or multiple) disorder in the heart. For each ECG we computed all minimal diagnoses. The model used in the experiments is

represented at four levels of abstraction [Mozetič and Pfahringer, 1992] in order to reduce the cost of individual model calls.

Both models, IDA and all the critical set algorithms are implemented in SICStus Prolog [Carlsson and Widen, 1991] and the experiments were run on a SUN IPX workstation.

4.1 A 1000-bit adder

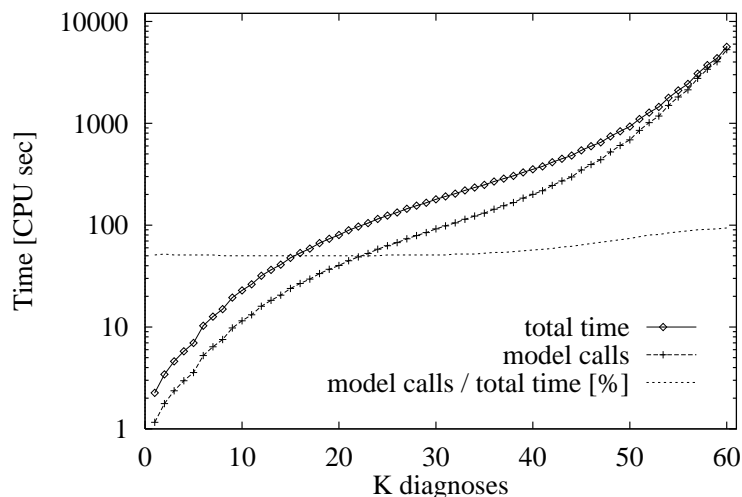


Figure 1: Total diagnostic time vs. time spent for model calls for the 1000-bit adder. Proportion of the time spent by the model calls increases from initial 50% to final 94%.

Figure 1 gives total time needed to compute the first k minimal diagnoses, and time spent by model calls for *Algorithm 0*. Proportion of the time spent by the model calls increases from 50% for small k to 94% for large k where time really matters. This confirms the thesis that the model calls dominate the overall time. Despite the huge search lattice, the set operations have negligible effect on the overall time.

Figure 2 gives the number of model calls made by different algorithms. Somehow surprisingly *Algorithm 0* outperforms *Algorithms II* and *IV*. While all three have the same number of successful calls, *Algorithm 0* makes lower number of failed calls than *Algorithm IV*, which in turn is better than *Algorithm II*. For all the 60 diagnoses, the number of failed calls is 552, 936, and 2077, respectively. For larger k the number of failed calls greatly exceeds the number of successful calls and therefore dominates the overall diagnostic time.

4.2 The KARDIO heart model

Table 2 gives the number of model calls for different algorithms, and the total diagnostic time. Proportion of the time spent by the model calls is between 86–88% which confirms

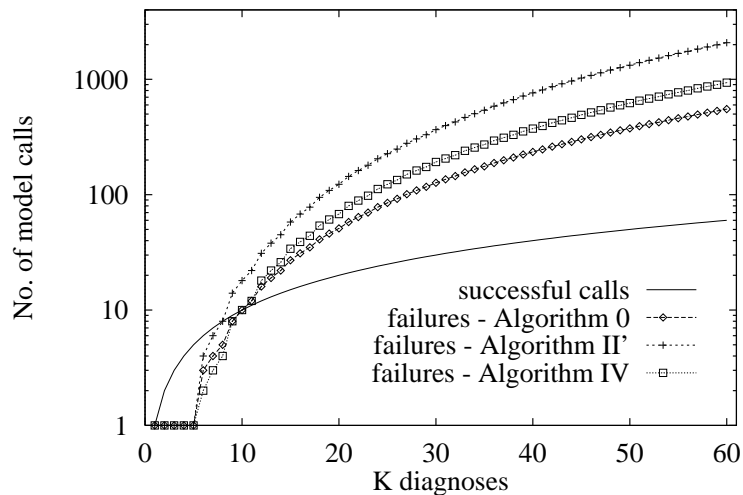


Figure 2: The number of successful and failed model calls for the 1000-bit adder. The number of successful calls is the same for all the three algorithms while the number of failed calls varies and dominates the diagnostic time.

	<i>Algorithm 0</i>	<i>Algorithm II'</i>	<i>Algorithm IV</i>
Successful model calls	4313	4314	4314
Failed model calls	23985	41648	29299
Total model calls	28298	45962	33613
Total diag. time [CPU sec]	554	909	669
Proportion of time spent by model calls	88 %	87 %	86 %

Table 2: Number of model calls and overall diagnostic times to compute all minimal diagnoses for the KARDIO model for all 3096 distinct observations.

the claim that the model calls dominate the overall time. All three algorithms make approximately the same number of successful model calls. Since there are altogether 4130 minimal diagnoses (for all 3096 observations), only 4% of successful model calls yield non-minimal diagnoses. In terms of failed model calls, results are similar to the 1000-bit adder example. *Algorithm 0* makes lower number of failed calls than *Algorithm IV*, which in turn is better than *Algorithm II'*. This is also reflected in the overall times.

5 Analysis of the algorithms

Both experiments indicate that *Algorithm 0* outperforms *Algorithms II'* and *IV*. Our explanation is the following. In most of the successful model calls, *TP* returns a minimal

diagnosis. Note, however, that TP does not solve the minimal diagnosis problem because it remains to prove that the diagnosis is minimal indeed. For this proof, *Algorithm 0* makes an optimal number of failed model calls, while *Algorithms II'* and *IV* are suboptimal. The reason for better performance of *Algorithm 0* is not the large cardinality of minimal diagnoses, but the use of a powerful TP function.

Take an example in Figure 3. Assume a sublattice with a root $\{1,2,3,4,5\}$. Assume that the root is also a minimal diagnosis. Let's ignore the number of successful model calls needed to find this minimal diagnosis (this is where *Algorithms II'* and *IV* are better than *Algorithm 0* since they make $O(\log n)$ calls instead of $O(n)$). Consider just the number of failed calls needed to *prove* that the node $\{1,2,3,4,5\}$ is a minimal diagnosis indeed. In this case *Algorithm 0* makes only 5 calls (one cannot do better), *Algorithm II'* makes 13 calls (node $\{2,3,4,5\}$ is visited twice!), and *Algorithm IV* makes 8 failed calls. This is consistent with the experimental results and explains why *Algorithm 0* outperforms *Algorithms II'* and *IV*.

6 Conclusion

Experiments indicate that the number and cost of model calls is a dominating factor in the time needed to compute minimal diagnoses. We modified two critical set algorithms, proposed by [Loveland, 1987], and incorporated them into the incremental diagnostic algorithm IDA [Mozetič, 1992]. In terms of the number of model calls, the worst-case complexity to compute k minimal diagnoses is reduced from $O(kn)$ to $O(k \log n)$. However, in testing experiments, the simple *Algorithm 0* with $O(n)$ complexity outperforms more sophisticated *Algorithms II'* and *IV* with $O(\log n)$ complexity. The reason is the use of a powerful TP function which enables 'dives' deep into the lattice. In most cases TP returns a near-minimal diagnosis. It only remains to show that the diagnosis is minimal by showing that no subset of it is also a diagnosis. For this proof, *Algorithm 0* is optimal in terms of failed model calls.

As a consequence of this empirical study, one can draw two conclusions. First, the worst-case complexity results have to be properly interpreted. Worst-case is not the average case, and the actual average case might be much closer to the best case. Due to the use of the more powerful TP instead of the f function, the overall proportion between successful and failed model calls changed. Essentially, there was a shift of burden from the *Min_diag* procedure to the TP function. The worst-case complexity analysis remains valid, but it does not properly grasp the relevant factors any more, and therefore loses its tightness and significance.

Second, the experimental results point to the direction in which further research should be focused. In order to improve the efficiency of diagnosis one should better try to reduce the cost of individual model calls. According to the results of this paper and despite of the low number of testing domains, we feel that this is more promising than trying to reduce the number of model calls. Our past experiments with the KARDIO model [Mozetič, 1991, Mozetič and Pfahringer, 1992] support the claim. It was shown that model abstraction,

efficient constraint propagation, and model compilation, each individually yielded at least a 10-fold improvement in diagnostic efficiency over a simple, one-level model of structure and behaviour.

Another example of the improved model efficiency is the use of structure-based abduction [El Fattah and Dechter, 1994]. First, a model has to be transformed into an acyclic constraint network. This might require a relatively expensive pre-processing of the initial constraints using tree clustering. However, as a consequence, an equivalent of the *Min_diag* procedure is replaced by a single model call which returns not only a minimal, but a minimal cardinality diagnosis in polynomial time.

Acknowledgements

This work was supported in part by the Austrian “Fonds zur Förderung der wissenschaftlichen Forschung” under grant P9426-PHY. Austrian Research Institute for Artificial Intelligence is supported by the Austrian Federal Ministry of Science and Research.

References

- [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.
- [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.
- [Childress and Valtorta, 1993] Childress, R.L., Valtorta, M. Polynomial-time model-based diagnosis with the critical set algorithm. *Proc. Fourth Intl. Workshop on Principles of Diagnosis, DX-93*, pp. 166-177, Aberystwyth, Wales, 1993.
- [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 *et al.*, 1992] de Kleer, J., Mackworth, A.K., Reiter, R. Characterizing diagnoses and systems. *Artificial Intelligence 56*, pp. 197-222, 1992. Also in *Readings in Model-based Diagnosis* (W.C. Hamscher, L. Console, J. de Kleer, Eds.), pp. 54-65, Morgan Kaufmann, San Mateo, CA, 1992.
- [El Fattah and Dechter, 1994] El Fattah, Y., Dechter, R. Diagnosing tree-decomposable circuits. Technical Report 94-18, Univ. of California at Irvine, CA, 1994.

- [Friedrich *et al.*, 1990] Friedrich, G., Gottlob, G., Nejd, W. Physical impossibility instead of fault models. *Proc. 8th Natl. Conf. on Artificial Intelligence, AAAI-90*, pp. 331-336, Boston, MIT Press, 1990.
- [Loveland, 1987] Loveland, D.W. Finding critical sets. *Journal of Algorithms* 8, pp. 362-371, 1987.
- [Mozetič, 1991] Mozetič, I. Hierarchical model-based diagnosis. *Intl. Journal of Man-Machine Studies* 35 (3), pp. 329-362, 1991. Also in *Readings in Model-based Diagnosis* (W.C. Hamscher, L. Console, J. de Kleer, Eds.), pp. 354-372, Morgan Kaufmann, San Mateo, CA, 1992.
- [Mozetič, 1992] Mozetič, I. A polynomial-time algorithm for model-based diagnosis. *Proc. 10th European Conf. on Artificial Intelligence, ECAI-92*, pp. 729-733, Vienna, Austria, John Wiley & Sons, 1992.
- [Mozetič and Pfahringer, 1992] Mozetič, I., Pfahringer, B. Improving diagnostic efficiency in KARDIO: abstractions, constraint propagation, and model compilation. In *Deep Models for Medical Knowledge Engineering* (E. Keravnou Ed.), Elsevier, 1992.
- [Reiter, 1987] Reiter, R. A theory of diagnosis from first principles. *Artificial Intelligence* 32, pp. 57-95, 1987.

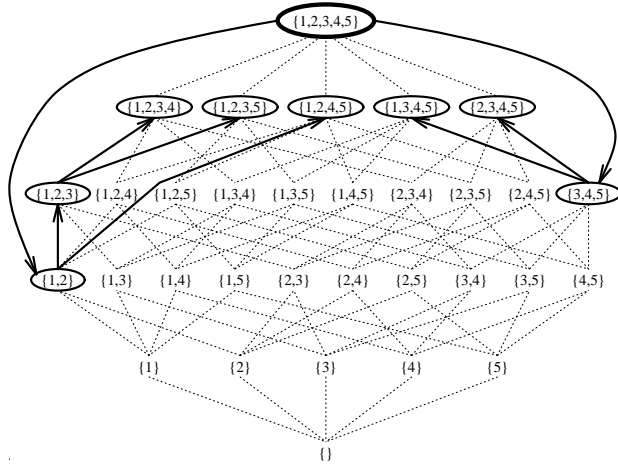
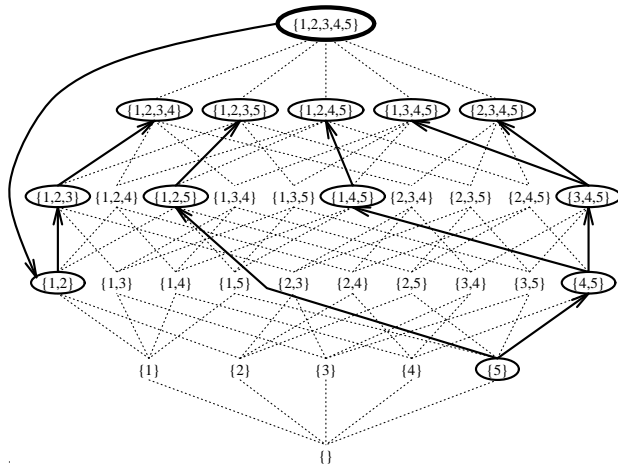
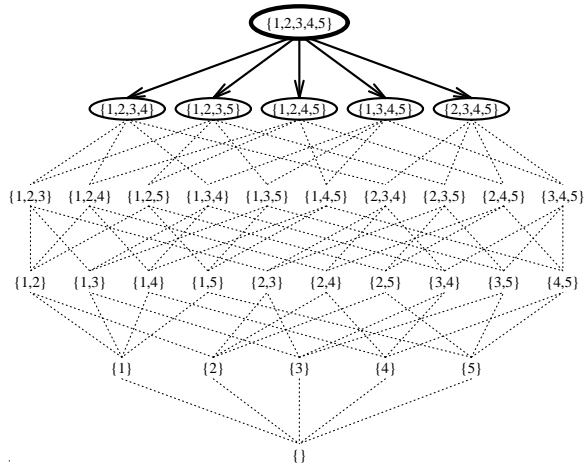


Figure 3: Lattice nodes visited by *Algorithms 0, II', IV* from top to bottom, respectively, to prove that $\{1,2,3,4,5\}$ is a minimal diagnosis. Visited nodes correspond to failed model calls, arrows indicate a partial order in which the nodes are visited.