Bridging the Gap between Theory and Practice: REVISE Progress Report

Carlos Viegas Damásio, Luís Moniz Pereira, Michael Schroeder
Department of Computer Science, F.C.T., Universidade Nova de Lisboa,
2825 Monte de Caparica, Portugal,
Phone: ++351-1-3.500.279,Email: {cd,lmp,msch}@fct.unl.pt

February 9, 1996

Introduction REVISE is an extended logic programming system for revising knowledge bases. It is based on logic programming with explicit negation and integrity constraints and provides a two- or three-valued assumption revision to remove contradictions from the knowledge base. It has been tested on a spate of examples with an emphasis in model-based diagnosis.

The first prototype of the system showed that extended logic programming is well suited to solve model-based diagnosis problems [DNP94]. Also all extensions of a state-of-the-art diagnoser such as preferences and strategies [FNS94, NFS95] can be incorporated into the existing approach [DNPS95]. The second implementation of REVISE is based on top-down evaluation of well founded semantics with explicit negation (WFSX) [ADP94a, ADP94b], which led to a dramatic speed increase. In this system description we give an overview over the system's architecture and we evaluate and compare the implemented prototypes.

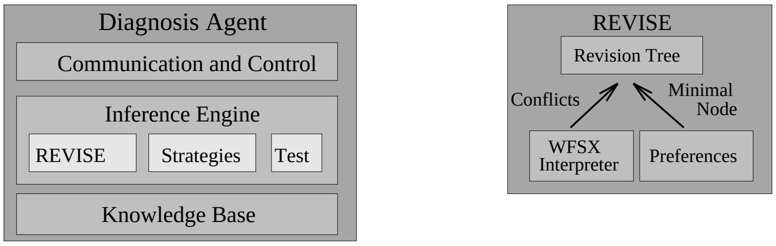


Figure 1: Components of a diagnosis agent and REVISE, its core.

Architecture The REVISE system is embedded into an architecture for a diagnosis agent consisting of three layers: a knowledge base, an inference layer and on top a component for communication and control (see Fig. 1). The core of the inference machine is the REVISE system, which computes diagnoses. Additionally strategies to guide the diagnostic process are supported. The test component realises probing and provides new observations.

REVISE has to undertake two main tasks: producing conflicting assumptions and managing diagnosis candidates through prefrences. The conflicts are computed by an abductive WFSX interpreter and are used in the revision tree to compute the candidates. The preference component provides the revision tree with an order for the diagnosis candidates.

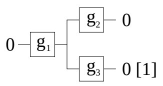
Example. Consider three not-gates g_1, g_2, g_3 as shown below. Though the output of g_3 should be 0 we observe 1. The system description states that a value can

n	5	10	11	12
Version 1.0	4.6	1100	4300	15000

n	5	10	50	100
Version 2.3	0.16	0.53	12	57
Version 2.4	0.222	0.71	20	110

Figure 2: Bulb example. Times in seconds. A DEC5000/240 was used.

either be observed or predicted based on the assumption that the component is not abnormal. A contradiction arises if an observation yields 1, while 0 is predicted.



$$\begin{array}{rcl} val(Node, V) & \leftarrow & obs(Node, V) \\ val(out(not, g_i), V) & \leftarrow & not\ ab(g_i), val(in(not, g_i), W), not\_table(V, W). \\ \bot & \leftarrow & obs(Node, 1), val(Node, 0). \end{array}$$

Diagnoses are computed by removing the contradiction. A call to the WFSX interpreter reveals that the contradiction is supported by the assumption that g_1 and g_3 are not abnormal. Thus the two diagnosis candidates $ab(g_1)$ and $ab(g_3)$ are kept in the revision tree ordered by an enforced preference relation. Assuming g_1 abnormal the WFSX interpreter detects that contradiction is still supported, namely by the assumption that g_2 is not abnormal. Assuming g_3 abnormal no more contradictions are detected. Finally the revision tree contains the diagnoses $\{ab(g_3)\}$ and $\{ab(g_1), ab(g_2)\}$.

Features of the REVISE System The use of extended logic programming in model-based diagnosis has turned out to be a very fruitful approach. Through several extensions the REVISE system contains now all features of a state-of-the-art diagnoser.

1. The system provides a logic programming background and therefore a formal semantics. In particular, explicit negation and negation by default, as well as integrity constraints form a powerful language to express diagnosis problems.

2. The system is open to all kinds of diagnosis paradigms. It supports minimal diagnosis by set inclusion, cardinality and probability. Due to the system's modular character other notions of minimality can easily be incorporated.

3. The system includes on top a strategy language to guide the diagnostic process. Concepts as multiple models, structural refinement and measurement strategies can be used.

4. The system is general. Beside the main application of diagnosis to digital circuits and computer networks, we have applied it to the domain of cars and to logic programming debugging.

Evolution of the System and Benchmarks Three prototypes of REVISE have been implemented. The main difference between them is in the evaluation of conflicts and candidates. Version 1.0 is based on bottom-up evaluation so that all conflicts are computed before constructing the revision tree. The timings of the bulb example in Fig. 2, where version 1.0 has to compute all 2^n conflicts to obtain the n diagnoses, clearly show the limits of this approach. For the versions 2.x the top-down-evaluation for WFSX allows to interleave the steps of computing conflict sets and diagnosis candidates (i.e. hitting sets). The revision tree is computed incrementally, i.e. after one conflict is computed the candidate space is updated. This avoids the explosion of version 1.0 in the bulb example (see Fig. 2).

Circuit	c17	gte	max	goel	alu4	voter	c432	c499
Version 2.3	0.30	7.0	280	290	360	710	25.000	82.000
Version 2.4	0.28	6.0	27	44	80	460	1.200	13.000

Figure 3: ISCAS85 benchmark circuits. Times in seconds. REVISE computes all single faults. A DEC5000/240 was used.

The difference between versions 2.3 and 2.4 is in the representation of conflicts. The former uses difference lists and the latter sets as ordered lists. The difference lists allow a constant append, but may contain duplicates, while the ordered sets are free of duplicates, but a union costs $O(min\{m,n\})$, where m,n is the size of the sets. Avoiding recomputation turns out to be crucial to solve small real world examples such as the ISCAS85 benchmark circuits [BPH85]. Version 2.4 copes with this problem by its set representation and memoizing in the WFSX interpreter. Depending on the fanout of the circuits' gates version 2.4 is up to 20 times faster than 2.3 (see Fig. 3).

Conclusion The REVISE system started as a prototype to show how to employ extended logic programming in model-based diagnosis. By now the system is integrated into a general architecture for a diagnosis agent, several applications exist and its performance was increased to solve small real world problems. REVISE cross-fertilized the theory of extended logic programming and the practical application of model-based diagnosis.

References

[ADP94a] J. J. Alferes, C. V. Damásio, and L. M. Pereira. Top–down query evaluation for well–founded semantics with explicit negation. In A. Cohn, editor, *Proc. of the European Conference on Artificial Intelligence'94*, pages 140–144. John Wiley & Sons, August 1994.

[ADP94b] J. J. Alferes, C. V. Damásio, and L. M. Pereira. A top-down derivation procedure for programs with explicit negation. In M. Bruynooghe, editor, *Proc. of the International Logic Programming Symposium'94*, pages 424–438. MIT Press, November 1994.

[BPH85] F. Brglez, P. Pownall, and R. Hum. Accelerated ATPG and fault grading via testability analysis. In *Proceedings of IEEE Int. Symposium on Circuits and Systems*, pages 695–698, 1985. The ISCAS85 benchmark netlists are available via ftp `mcnc.mcnc.org`.

[DNP94] Carlos Viegas Damásio, Wolfgang Nejdl, and Luís Moniz Pereira. REVISE: An extended logic programming system for revising knowledge bases. In J. Doyle, E. Sandewall, and P. Torasso, editors, *Knowledge Representation and Reasoning*, pages 607–618, Bonn, Germany, May 1994. Morgan Kaufmann.

[DNPS95] C. V. Damásio, W. Nejdl, L. Pereira, and M. Schroeder. Model-based diagnosis preferences and strategies representation with meta logic programming. In Krzysztof R. Apt and Franco Turini, editors, *Meta-logics and Logic Programming*, chapter 11, pages 269–311. The MIT Press, 1995.

[FNS94] Peter Fröhlich, Wolfgang Nejdl, and Michael Schroeder. A formal semantics for preferences and strategies in model-based diagnosis. In *5th International Workshop on Principles of Diagnosis (DX-94)*, pages 106–113, New Paltz, NY, October 1994.

[NFS95] Wolfgang Nejdl, Peter Fröhlich, and Michael Schroeder. A formal framework for representing diagnosis strategies in model-based diagnosis systems. In *International Joint Conference on Artificial Intelligence*, Montreal, August 1995.