Qualifying the Expressivity/Efficiency Tradeoff: Reformation-Based Diagnosis

Helmut Prendinger and **Mitsuru Ishizuka**
Department of Information and Communication Engineering
School of Engineering, University of Tokyo
7-3-1, Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
E-mail: {helmut,ishizuka}@miv.t.u-tokyo.ac.jp

Abstract

This paper presents an approach to model-based diagnosis that first compiles a first-order system description to a propositional representation, and then solves the diagnostic problem as a linear programming instance. Relevance reasoning is employed to isolate parts of the system that are related to certain observation types and to economically instantiate the theory, while methods from operations research offer promising results to generate near-optimal diagnoses efficiently.

Introduction and Motivation

A central problem of model-based diagnosis is the computational complexity of the underlying diagnostic reasoning task (see, e.g. Eiter and Gottlob (1995)). Therefore, several researchers have proposed to preprocess a given system description, mostly a propositional theory, such that the 'compiled' form can be processed more efficiently (Williams & Nayak 1996; Darwiche 1998). In many cases, however, it is more natural to describe systems in a more expressive language, such as *first-order logic*. In this paper, we will consider the case where the system description is given as a first-order Horn theory (without function symbols), and compile this description to a propositional one. Techniques from relevance reasoning (Levy, Fikes, & Sagiv 1997; Schurz 1999) will be employed to keep the resulting propositional theory within manageable size. More specifically, relevance reasoning is used to filter out the part of the system that is 'relevant' for certain observation types, and to economically instantiate variables by appropriate constants (usually denoting values of system attributes). In this way, we may preserve the compactness of first-order descriptions, and allow for processing an efficient propositional theory. Thus we qualify the notorious expressivity/efficiency tradeoff.

Given a propositional system description, we will use the Networked Bubble Propagation (NBP) mechanism (Ohsawa & Ishizuka 1997), a high-speed hypothetical ('abductive') reasoner, as the diagnostic engine. Focusing on most-probable (least-expensive) diagnoses is realized by assigning a numerical weight to each hypothesis (possible system fault). The NBP mechanism tries to find an optimal solution, i.e., a diagnosis with a minimal sum of individual faults' weights, and actually generates a near-optimal solution in polynomial time of approximately $\mathcal{O}(n^{1.4})$, where n the number of hypotheses in the problem formulation. The efficiency of the NBP mechanism relies on methods from the area of 0-1 integer linear programming.

Following a comprehensive step-by-step approach, we show how a general definition of model-based diagnosis can be translated to a integer linear programming problem. Although the possibility of such a translation is not completely surprising, we are not aware of any previous attempts in the literature. In the planning field, however, Bylander (1997) shows how to translate propositional STRIPS planning instances to linear programming instances, and Williams and Nayak (1996) recast their model-based configuration manager as a combinatorial optimization problem.

The main contribution of this paper can be seen as bringing together methods from the fields of deductive databases and operations research. In particular, relevance reasoning is employed to preprocess a given behavioral model encoded in first-order Horn logic, and operations research methods allow to efficiently solve diagnosis problems.

The rest of the paper is organized as follows. In the following section, we explain some notions related to model-based diagnosis. Then we introduce a general framework for extracting information relevant to finding diagnoses, relative to certain observation types and the structure of the system. Moreover, we discuss a sophisticated instantiation method based on relevance reasoning. Next, we explain how hypothetical reasoning problems (corresponding to diagnostic problems) can be recast as problems of integer linear programming. We also report on some preliminary experimental results obtained in testing our approach. Finally, we discuss related work and draw some conclusions.

Model-based Diagnosis

A *diagnostic problem* is characterized by a set of observations to be explained, given a behavioral model of some system (device). The behavioral model of a system describes its normal and/or faulty behavior. A solution to a diagnostic problem (a *diagnosis* for short) is a set of hypotheses which, if assumed, would 'explain' the observations. We adopt the general definition of a diagnostic problem proposed by Console and Torasso (1991), which subsumes both abductive and consistency-based definitions of diagnosis.

Definition 1 *A* diagnostic problem DP *is a quadruple* $DP = \langle T, I, CXT, OBS \rangle$, *such that*
(1) T *is a* behavioral model, *i.e., a set of Horn clauses of the form "*$p(\bar{X}_{n+1}) \leftarrow q_1(\bar{X}_1 \wedge ... \wedge q_n(\bar{X}_n)$*" where* $p(\bar{X}_{n+1}), q_1(\bar{X}_1), ..., q_n(\bar{X}_n)$ *are atomic formulas, and* $\bar{X}_i$ *denotes the sequence of variables* $X_{i,1}, ..., X_{i,m_i}$. *Each* q_i *in the body (the RHS of the clause) denotes either (i) a state that can be assumed as a hypothesis (an* abducible*), or (ii) context conditions on which the behavior of the system depends, such as 'inputs' to a device, or (iii) a state that can neither be observed nor assumed (internal state). The atom* p *in the head (the LHS of the clause) denotes either a state that can be observed or measured and for which we want to find an explanation, or an internal state.*
(2) I *is a set of* inconsistency constraints, *i.e., a set of Horn clauses of the form "*$\perp \leftarrow q_1(\bar{X}_1) \wedge ... \wedge q_n(\bar{X}_n)$*" where each* q_i *denotes a context condition or a hypothesis, and the symbol* $\perp$ *denotes the logical constant* falsum.
(3) CXT *is a set of variable-free (ground) atoms that denote context conditions (inputs).*
(4) OBS *is a set atoms that denote observations.*

We require that T contains no cycles, i.e., a state must not be a direct or indirect cause of itself. The class of acyclic theories properly contains the class of so-called 'tree-structured' theories (Stumptner & Wotawa 1997) where each state can cause at most one other state.

The notion of 'diagnosis' will be explained by giving a definition of a *solution* to a diagnostic problem. A *diagnostic procedure* allows to generate a solution to a diagnostic problem automatically. Since the diagnostic procedure will be implemented as a hypothetical reasoning mechanism, we first reformulate a diagnostic problem as a problem of hypothetical reasoning. We borrow the reformulation suggested by Console and Torasso (1991).

Definition 2 *Let* $DP = \langle T, I, CXT, OBS \rangle$ *be a diagnostic problem. A* hypothetical reasoning problem HRP *corresponding to* DP *is a quadruple* $HRP = \langle T, I, CXT, \langle \mathcal{O}^+, \mathcal{O}^- \rangle \rangle$, *such that*

- $\mathcal{O}^+ \subseteq OBS$;
- $\mathcal{O}^- = \{\neg m(v_i) : m(v_j) \in OBS$, *for each value* v_i *of* m *different from* $v_j\}$.

Here, $\mathcal{O}^+$ denotes the set of observations that have to be covered by the solution; $\mathcal{O}^-$ is the set of observations that 'contradict' (or 'conflict with') the observations.

The following definition of a *solution* for a HRP problem is an extension of the definition given by Console and Torasso (1991).

Definition 3 *Let* $HRP = \langle T, I, CXT, \langle \mathcal{O}^+, \mathcal{O}^- \rangle \rangle$ *be a hypothetical reasoning problem, and* $\mathcal{H}$ *a set of abducibles. A set* $H \subseteq \mathcal{H}$ *is a* solution hypotheses set *for HRP if and only if*

- *for each* $m \in \mathcal{O}^+$: $T \cup CXT \cup H \vdash m$;
- $T \cup CXT \cup H \cup \mathcal{O}^- \nvdash \perp$;
- $I \cup CXT \cup H \nvdash \perp$.

For convenience, we will sometimes call a solution hypotheses set simply a *solution* or *explanation*. The second condition in the definition is called *consistency constraint* in (Console & Torasso 1991). We also account for the case where certain solution sets are not admissible, by means of so-called *inconsistency constraints* (third condition). This condition is not present in (Console & Torasso 1991).

Reformation by Relevance Reasoning

In the off-line reformation (or compilation) phase, the first-order system description is first partitioned into subtheories, possibly indexed with an observation type. Next, clauses that cannot contribute to the solution of any query, also called *strongly irrelevant* clauses, are removed from the subtheory (Schurz 1999). Finally, the *query-tree* idea is employed to obtain exactly the set of ground clauses relevant to a query type (Levy, Fikes, & Sagiv 1997).

Theory Factorizing

For the case of *tree-structured* systems, the idea of theory factorizing is to split a theory T into disjoint subtheories $T_1, ..., T_n$ such that no clause C in a given subtheory T_i resolves with some clause D from a different subtheory T_j. This means that the search space for a given atomic query type $p(\bar{X})$ can be restricted to a single subtheory T_i. The theory factorizing algorithm is described in (Prendinger & Ishizuka 1999) and can be summarized as follows: (i) if a clause C does not resolve with any independent subset of the already generated partition, then $\{C\}$ is added as a new element of the partition; (ii) if C resolves with independent subsets $\mathcal{D}_1, ..., \mathcal{D}_k$ in the partition, then those subsets and C form a new element of the partition while the old elements $\mathcal{D}_1, ..., \mathcal{D}_k$ get cancelled. Theory factorizing has to be applied only once, and can be done in polynomial time. Note that factorizing is an application of the formal notion of *independence* (Lang & Marquis 1998).

In the more general case of *acyclic* theories, factorizing is performed by means of an algorithm that computes all clauses that are 'reachable' from a query type. Informally, a clause C is *reachable* from a query type $p(\bar{X})$ if there exists some path from $p(\bar{X})$ to the head of C. The set of clauses reachable from $p(\bar{X})$ is denoted by T_p. It is important to note that in either case, factorizing can be done by only considering the query *types* such as $p(\bar{X})$, i.e., independent of particular instantiations such as $p(a)$ or $p(b)$.

Theory Simplification

A given (independent) theory may still contain strongly irrelevant clauses. This is the case when a clause C contains an atom in $bd(C)$ that does not resolve with the head of any other clause. C is called a *failing* clause. Theory simplification removes failing clauses from the initial theory T, obtaining the simplified theory T'. The simplification process is repeated until no failing clauses are detected. Note that clauses having hypotheses $h \in \mathcal{H}$ in their body are *not* deleted, since hypotheses *may* contribute to a proof (if they are assumed). In addition to remove strongly irrelevant clauses, theory simplification can be utilized to detect unspecified context conditions in a behavioral model. A situation where a subquery does not resolve with a fact indicates that some input to a device has not been declared in the model.

Theory Instantiation

In the last phase of the reformation process, the theory is actually instantiated. By employing the query-tree idea (Levy, Fikes, & Sagiv 1997), we obtain exactly the set of ground clauses relevant to a query type, together with all instantiations of the query type that have a solution w.r.t. the theory. A *query-tree* is a compact representation of a search tree for first-order Horn theories and has the form of an AND-OR tree with goal-nodes and rule-nodes (Levy, Fikes, & Sagiv 1997). Since we do not allow for recursion in clauses, our construction of the query tree is simpler than the original one in (Levy, Fikes, & Sagiv 1997). On the other hand, we allow that some leaves of the query-tree are uninstantiated. Those typically denote hypotheses (abducibles).

Example 1 *Consider the following theory T where the atom with predicate h denotes a hypothesis.*

(r_1) $p(X,Y) \leftarrow q1(X,Y) \wedge q2(X,Y)$.
(r_2) $q1(X,Y) \leftarrow r1(X,Y) \wedge h(X,Y)$.
(r_3) $q2(X,Y) \leftarrow r2(X,Y)$.
(r_4) $q2(X,Y) \leftarrow r3(X,Y)$.
(f_1) $r1(a,b)$. (f_2) $r1(a,c)$.
(f_3) $r2(a,b)$. (f_4) $r2(c,d)$. (f_5) $r3(b,d)$.

The query tree algorithm consists of two phases (a more detailed description is given in (Levy, Fikes, & Sagiv 1997)). In the *bottom-up* phase, a set of *adorned* predicates and rules is generated. An adorned predicate p^c is a predicate p with constraint c on its arguments. The adorned rules are the rules of the theory with predicates replaced by adorned predicates. We start with the base predicates of the theory, i.e., the predicates of facts and hypotheses. For instance, the adorned predicate $r1^c$ is obtained by completion: $r1(X,Y) \leftrightarrow (X = a \wedge Y = b) \vee (X = a \wedge Y = c)$. For convenience, the adornment of $r1$ is written as $c(X,Y) = \{\langle a,b\rangle, \langle a,c\rangle\}$. Let U be the set of all constants appearing in the theory. Then the adornment of the (uninstantiated) hypothesis h is $\{\langle X,Y\rangle : \langle X,Y\rangle \in U^2\}$. The adornments of predicates in head atoms of rules are generated by projecting the adornments of predicates in body atoms onto the

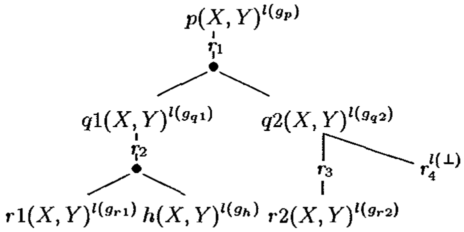

Figure 1: Query-tree for Example 1. The label for each goal-node g is $l(g) = \{\langle a,b\rangle\}$. For simplicity, labels of rule nodes are omitted. Note that expanding node $q2(X,Y)$ with rule r_4 would result in an inconsistent label.

head variables. For instance, the adornment of $q1$ is $\{\langle a,b\rangle, \langle a,c\rangle\}$. The bottom-up phase terminates when no new adornments are generated. In Example 1, the following further predicate adornments are generated: $\{\langle a,b\rangle, \langle c,d\rangle\}$ for $r2$, $\{\langle b,d\rangle\}$ for $r3$, $\{\langle a,b\rangle, \langle c,d\rangle, \langle b,d\rangle\}$ for $q2$, and $\{\langle a,b\rangle\}$ for p.

In the *top-down* phase, the predicate adornment of the query type is 'pushed down' to the base predicates. Starting with the node of the adorned query type q^c, we construct the query-tree such that each node g of a predicate p has a *label* $l(g)$. Initially the goal-node $l(g) = q^c$ is created (see Fig. 1). A goal-node g for a predicate q^{c_h} can be unified with adorned rules r of the form $q^{c_h}(\bar{X}_{n+1}) \leftarrow c \wedge p_1^{c_1}(\bar{X}_1) \wedge \ldots \wedge p_n^{c_n}(\bar{X}_n)$. If $l(g) \wedge c$ is satisfiable, a rule-node g_r is created as a child of g, with $l(g_r) = l(g) \wedge c$ as its label. For every body atom $p_i^{c_i}$ in r, the rule-node $l(g_r)$ has a child goal-node whose label is the projection of $l(g_r)$ onto $\bar{X}_i$. The top-down construction halts, since nodes of base predicates and nodes with unsatisfiable label, denoted by $l(\perp)$, are not expanded. As shown in (Levy, Fikes, & Sagiv 1997), the complexity of building the query-tree is linear in the number of rules and possibly exponential in the arity of predicates.

Interestingly, *instantiation* of the theory is simply a by-product of the top-down construction of the query-tree: if $l(g_r)$ is the label of a rule r in the query-tree, the propositional version of r is obtained by performing all unifications appearing $l(g_r)$. As a result, the following propositional KB is obtained for Example 1.

(r_1') $p(a,b) \leftarrow q1(a,b) \wedge q2(a,b)$.
(r_2') $q1(a,b) \leftarrow r1(a,b) \wedge h(a,b)$.
(r_3') $q2(a,b) \leftarrow r2(a,b)$.
(f_1') $r1(a,b)$. (f_3') $r2(a,b)$.

It is a consequence of the construction of the query-tree that $p(a,b)$ is the only instance of the query type $p(X,Y)$ with a solution (given that $h(a,b)$ is assumed).

As output of the reformation process, we obtain either independent propositional theories (tree-structured systems), or propositional theories indexed with query types (acyclic systems). It is important to note that rather than creating specialized theories for specific observations, we extract the relevant part for a set of observations (denoted by query types).

Diagnosis as Linear Programming

In this section, we show how hypothetical reasoning problems (HRPs) corresponding to diagnostic problems (DPs) can be solved by integer linear programming problems (LPPs). First, we describe the translation from HRP to a problem of integer linear programming. Next, the notion of 'best explanation' is explicated within the framework of linear programming. Finally, we briefly describe the mechanism of Networked Bubble Propagation (NBP), an efficient hypothetical reasoning method for computing near-optimal solutions.

From Diagnosis to Integer Linear Programming

An *integer linear programming problem (LPP)* is defined by a set of variables, a set of linear constraints, and an objective function. The set of linear constraints consists of linear inequalities and equalities, and the objective function is a linear function on the variables. A solution for LPP is called *feasible* if it satisfies the constraints, and it is called *optimal* if it is feasible and maximizes (or minimizes) the objective function.

Let V denote the union of the set of all propositional variables occurring in $T \cup I \cup CXT$ and $\{\bot\}$. Then $\mathcal{V}$ is the set of variables indexed by V, i.e., $\mathcal{V} = \{x_p : p \in V\}$. The set $\mathcal{V}$ has the following distinguished subsets: (i) $\mathcal{V}_{\mathcal{O}^+} \subset \mathcal{V}$ is the set of variables denoting symptoms; (ii) $\mathcal{V}_{\mathcal{O}^-} \subset \mathcal{V}$ is the set of variables denoting observations that conflict with symptoms; (iii) $\mathcal{V}_{\mathcal{H}} \subset \mathcal{V}$ is the set of hypothesis variables; (iv) $\mathcal{V}_{CXT} \subset \mathcal{V}$ is the set of variables denoting context conditions; and (v) $x_\bot \in \mathcal{V}$ is the variable associated with $\bot$.

Clauses in the behavioral model T are assumed to have one of the following forms, depending on whether the bodies of clauses in T are AND-related or OR-related. The heads of clauses with AND-related and OR-related bodies are called AND-nodes and OR-nodes, respectively: (AND) $p \leftarrow q_1 \wedge ... \wedge q_n$, or (OR) $p \leftarrow q_1, ..., p \leftarrow q_m$. If the body of an OR-node p is a conjunction of propositional variables $q_{k,1} \wedge ... \wedge q_{k,n_k}$, an auxiliary propositional variable $q_{k,0}$ is invented such that $p \leftarrow q_{k,0}$ and $q_{k,0} \leftarrow q_{k,1} \wedge ... \wedge q_{k,n_k}$. For convenience, we define a successor set S_p for each $p \in V$ as follows: $S_p = \{q : q$ occurs in an AND-related or OR-related body in clauses with head $p\}$. $|S_p|$ is the cardinality of S_p.

The following translation is similar to the one given by Santos (1994).

Definition 4 *Let* $HRP = \langle T, I, CXT, \langle \mathcal{O}^+, \mathcal{O}^- \rangle\rangle$ *be a hypothetical reasoning problem. An* integer linear programming problem (LPP) *corresponding to HRP is a pair* $LPP = \langle \mathcal{V}, \mathcal{I}\rangle$, *where* $\mathcal{V}$ *is a set of variables and* $\mathcal{I}$ *is a finite set of linear inequalities and equalities on* $\mathcal{V}$.
(1) Clauses with AND*-related and* OR*-related bodies are translated to linear inequalities as follows:*

- *Let p be an* AND*-node with successor set S_p.*

$$x_p \leq x_q \; (q \in S_p), \quad \sum_{q \in S_p} x_q - |S_p| + 1 \leq x_p$$

- *Let p be an* OR*-node with successor set S_p.*

$$\sum_{q \in S_p} x_q \geq x_p, \quad x_p \geq x_q \; (q \in S_p)$$

(2) Inconsistency constraints $ic \in I$ of the form "$\bot \leftarrow p_1 \wedge ... \wedge p_n$" are a special cases of clauses with AND*-related body because their head is the constant $\bot$, where $x_\bot = 0$. Therefore, we only need a single linear inequality*

$$\sum_{p \in S_\bot} x_p - |S_\bot| + 1 \leq 0$$

(3) In general, for each $x_p \in \mathcal{V}$, x_p is either 0 or 1. In particular, (i) for each $x_p \in \mathcal{V}_{\mathcal{O}^+}$, we add the equation $x_p = 1$, i.e., a given observation p must be assigned the value true; (ii) for each $x_p \in \mathcal{V}_{\mathcal{O}^-}$, we add the equation $x_p = 0$, thereby assigning observations that contradict symptoms the value false; (iii) for each $x_q \in \mathcal{V}_{CXT}$, the variable associated with a context atom q, we add the equation $x_q = 1$, saying that the conditions expressed by contextual data hold.

By way of example, we show how the second condition guarantees that the inconsistency constraints are satisfied. Since $\bot$ is (always) assigned *false*, some p in $S_\bot$ must be false. Take the *ic* $\bot \leftarrow h_1 \wedge h_2 \wedge h_3$ and assume that each of the hypothesis is assigned the value *true* (each of the hypothesis is assumed). Then we have $1 + 1 + 1 - 3 + 1 \leq 0$, i.e., $1 \leq 0$, which violates the constraint.

Definition 5 *A* variable assignment *for* $LPP = \langle \mathcal{V}, \mathcal{I}\rangle$ *is a function ϕ from $\mathcal{V}$ to $\{0, 1\}$. ϕ is a* 0-1 solution *for LPP if ϕ satisfies all the constraints in $\mathcal{I}$. A* 0-1 solution hypotheses set $H_{0\text{-}1}$ *for LPP consists of all $x_p \in \mathcal{V}_{\mathcal{H}}$ which are assigned 1 in the 0-1 solution for LPP.*

Not surprisingly, the following theorem of Santos (1994) can be extended to problems including inconsistency constraints.

Theorem 1 *H is a solution hypotheses set for HRP if and only if $H_{0\text{-}1}$ is a 0-1 solution hypotheses set for LPP.*

Optimal 0-1 Solutions

In *cost-based* hypothetical reasoning, each hypothesis has an associated numerical weight, and the weight of an solution hypotheses set is simply the sum of the weights associated with hypotheses in the set (Santos 1994; Ohsawa & Ishizuka 1997). A solution hypotheses set is optimal if the sum is minimal.

In diagnostic reasoning, it is often desirable to obtain an optimal solution or best explanation; a fault can be said to have low weight if the fault is easy to repair, or even, a fault with low weight is more probable (de Kleer 1991). The idea to concentrate on the preferred (more probable, less expensive) diagnoses is also known as *focusing* (e.g. Freitag and Friedrich (1992)). To capture the notion of optimal solution (or best explanation) formally, we define a function w from $\mathcal{H}$ to the set of natural numbers. Given an integer linear programming problem, the *objective function* being minimized is as follows:

$$\Psi_{LPP} = \sum_{h \in \mathcal{V}_{\mathcal{H}}} x_h w(h)$$

Definition 6 *An* optimal *0-1 solution hypotheses set for LPP is a 0-1 solution hypotheses set (for LPP) that minimizes* Ψ_{LPP}.

The cost-based variant of hypothetical reasoning has great potential for incorporating notions of uncertain reasoning such as probability. In the case where all hypotheses are uniformly assigned a default weight, an optimal solution corresponds to a *minimal* diagnosis.

Networked Bubble Propagation (NBP)

In practice, it is more advantageous (in terms of efficiency) to search for a *near-optimal* solution rather than for the optimal one. NBP is a method for cost-based hypothetical reasoning that can be used to compute near-optimal diagnoses (Ohsawa & Ishizuka 1997). The search mechanism draws inspiration from 0-1 linear integer programming and improves the behavior of the so-called 'pivot and complement' method, which is known to find a near-optimal solution close to the optimal solution in polynomial time. NBP can find a near-optimal solution in polynomial time of approximately $\mathcal{O}(n^{1.4})$, where n is the number of hypotheses in the problem formulation (for details, see (Ohsawa & Ishizuka 1997)).

Preliminary Empirical Evaluation

The reformation methods are implemented in Sicstus Prolog, the NBP method is implemented in C. For the experiments we use a Sun Ultra 2 workstation with 320 MB memory. Running times exclude the time needed for reformation. For example, the relevant part of a theory consisting of 1000 clauses can be extracted in about 4 seconds. Our results should be seen as preliminary, more experiments are planned.

Acyclic first-order Horn theories. In the first experiment we show the efficiency gain of relevance reasoning (factorizing and simplification[1]). It involves first-order Horn theories from 40 up to 1000 rules with a fixed percentage of hypotheses (about 30%) and integrity constraints (about 20%) and few facts. The theory of Ex. # 1 is systematically expanded to theories of larger size. Fig. 2 shows the inference time as a function of the number of hypotheses before and after reformation by relevance reasoning. In both cases, the same solution sets H are generated, each corresponding to a minimal explanation. Results are obtained by averaging over three different query types. The results show that the reformed theories can be processed more efficiently, usually in excess of a factor of 3. The best speedup factor was 196, for a query type where 98% of the hypotheses are irrelevant.

Ex. #	# hypotheses		time (sec)	
	before ref.	after ref.	before ref.	after ref.
1	13	6	0.02	0.01
2	39	17	0.08	0.04
3	78	33	0.21	0.07
4	156	67	0.63	0.16
5	312	129	2.14	0.46

Figure 2: Comparision for first-order Horn theories.

Medical diagnosis. The second experiment is intended to show the efficiency of the NBP method as a diagnostic procedure. It involves a propositional theory from the medical domain and real-world patient cases, first used by Ng and Mooney (1992) to validate their abductive system ACCEL. The behavioral model consists of 648 rules of the form "$e \leftarrow c$", where e a symptom type and c refers to one of 25 damaged brain areas. Ng and Mooney (1992) consider 50 patient cases of an average of 8.56 symptoms, where ACCEL computed all minimal diagnoses in an average of 2.4 seconds per case (with an average of 4.6 diagnoses per case). We consider the first 25 cases (average of 8.64 symptoms). The NBP method finds a near-optimal diagnosis in an average of 0.06 seconds per case, where the near-optimal diagnosis corresponds to one of the minimal diagnoses. When using the factorized model, we could not measure any speedup effect, although on average, 83% of the rules were strongly irrelevant to the query types. This can be explained by the fact that the number of hypotheses is only slightly decreased. Note that simplification does not apply here since all leaves are hypotheses.

Discussion and Conclusion

In this paper, we describe a new approach for generating diagnoses for observations. The diagnostic task is performed in two successive phases: in the off-line phase a first-order behavioral model is compiled to a propositional representation, in the on-line phase integer linear programming is employed to generate a near-optimal solution to the diagnostic problem. Note that our reformation procedures are equivalence-preserving w.r.t. query types. The procedures are guaranteed not to slow down inference and preserve *all* solutions.

Our notion of 'reformation' differs from other compilation methods found in the diagnosis literature. For instance, Williams and Nayak (1996) generate all prime implicants of (propositional) transition models; Darwiche (1998) compiles a propositional system description into a sentence called *consequence* that has good computational properties. By contrast, we introduce an effective way to produce a compact propositional theory from a given *first-order* model. In this respect, our ap-

[1] The speedup effect of the instantiation technique has not been tested in our experiments (see Levy *et al.* (1997) for a related empirical evaluation).

proach shares intuitions with the 'first-order planning as (propositional) satisfiability' framework of Kautz and Selman (1996).

Similar to the work of de Kleer (1991), *focusing* is integrated to the diagnostic engine (the NBP mechanism), by associating costs to individual hypotheses, and not part of the compilation (Darwiche 1998). Freitag and Friedrich (1992) discuss an approach to focusing that confines search to the smallest submodel which is independent of the remaining behavioral model. For the class of tree-structured systems, their notion of smallest independent submodel corresponds to the subtheories listed in the partition of a factorized model.

Recently, there is growing interest in developing diagnosing systems that deal with behavioral models exhibiting a particular *structure*, such as tree-structured systems (Stumptner & Wotawa 1997; Darwiche 1998). It is shown both theoretically (Darwiche 1998) and experimentally (Stumptner & Wotawa 1997) that tree-structured systems allow for significantly faster inference than acyclic systems. The experiments in (Ohsawa & Ishizuka 1997) re-enforce those findings with the NBP method. We have shown that structure is also important from a reformation point of view. Factorizing tree-structured systems has to be done only once, whereas for acyclic systems, the relevant part has to be determined for each (combination of) query type(s).

This paper is part of ongoing work on reformation-based diagnosis. In the near future, we want to test our approach on more diverse first-order Horn theories, and compare our results to competing approaches (e.g., de Kleer (1991), Darwiche (1998)). We try to provide formal guarantees on performance.

A major problem of current NBP is that it does not scale well if the width of OR-related nodes is high. Hence we plan to employ another hypothetical reasoner which scales better than the NBP method. Ishizuka and Matsuo (1998) developed the so-called 'slide-down and lift-up' (SL) method that employs both linear and non-linear programming techniques to solve hypothetical reasoning problems. Although the SL method is slightly slower than the NBP method, it is attractive in terms of memory requirements and scalability.

In any case, reformation methods seem to be prime candidates for improving scalability. In (Prendinger & Ishizuka 1999) we describe further reformation methods that allow to 'shrink' the propositional theory by orders-of-magnitude. So-called *variable elimination* procedures allow to reduce the number of different variables in a clause, which becomes crucial when the clause is to be instantiated by constants. The methods developed in that paper also allow to handle (certain forms of) *recursive* Horn theories.

Acknowledgments

We would like to thank the anonymous referees for their very valuable comments. The first author was supported by a fellowship from the Japan Society for the Promotion of Science (JSPS).

References

Bylander, T. 1997. A linear programming heuristic for optimal planning. In *Proceedings 14th National Conference on Artificial Intelligence (AAAI-97)*, 694–699.

Console, L., and Torasso, P. 1991. A spectrum of logical definitions of model-based diagnosis. *Computational Intelligence* 7(3):133–141.

Darwiche, A. 1998. Compiling devices: a structure-based approach. In *Proceedings 6th International Conference on Knowledge Representation and Reasoning (KR-98)*, 156–166.

de Kleer, J. 1991. Focusing on probable diagnosis. In *Proceedings 9th National Conference on Artificial Intelligence (AAAI-91)*, 842–848.

Eiter, T., and Gottlob, G. 1995. The complexity of logic-based abduction. *Journal of the ACM* 42(1–2):3–42.

Freitag, H., and Friedrich, G. 1992. Focusing on independent diagnosis problems. In *Proceedings 3rd International Conference on Knowledge Representation and Reasoning (KR-92)*, 521–531.

Ishizuka, M., and Matsuo, Y. 1998. SL method for computing a near-optimal solution using linear and non-linear programming in cost-based hypothetical reasoning. In *Proceedings 5th Pacific Rim Conference on Artificial Intelligence (PRICAI-98)*, 611–625.

Kautz, H., and Selman, B. 1996. Pushing the envelope: Planning, propositional logic, and stochastic search. In *Proceedings 13th National Conference on Artificial Intelligence (AAAI-96)*.

Lang, J., and Marquis, P. 1998. Complexity results for independence and definability in propositional logic. In *Proceedings 6th International Conference on Knowledge Representation and Reasoning (KR-98)*, 356–367.

Levy, A. Y.; Fikes, R. E.; and Sagiv, Y. 1997. Speeding up inferences using relevance reasoning: a formalism and algorithms. *Artificial Intelligence* 97:83–136.

Ng, H. T., and Mooney, R. J. 1992. Abductive plan recognition and diagnosis: A comprehensive empirical evaluation. In *Proceedings 3rd International Conference on Knowledge Representation and Reasoning (KR-92)*, 499–508.

Ohsawa, Y., and Ishizuka, M. 1997. Networked bubble propagation: a polynomial-time hypothetical reasoning method for computing near-optimal solutions. *Artificial Intelligence* 91:131–154.

Prendinger, H., and Ishizuka, M. 1999. Preparing a first-order knowledge base for fast inference. In *Proceedings 12th International FLAIRS Conference (FLAIRS-99)*. To appear.

Santos, E. 1994. A linear constraint satisfaction approach to cost-based abduction. *Artificial Intelligence* 65:1–27.

Schurz, G. 1999. Relevance in deductive reasoning: A critical overview. In Schurz, G., and Ursic, M., eds., *Beyond Classical Logic*. St. Augustin: Academia Press.

Stumptner, M., and Wotawa, F. 1997. Diagnosing tree structured systems. In *Proceedings 15th International Conference on Artificial Intelligence (IJCAI-97)*, 440–445.

Williams, B. C., and Nayak, P. P. 1996. A model-based approach to reactive self-configuring systems. In *Proceedings 13th National Conference on Artificial Intelligence (AAAI-96)*, 971–978.