A Graph-Theoretic Optimisation of Temporal Abductive Validation

Tim Menzies[1] and Robert F. Cohen[2]

[1] Department of Artificial Intelligence, School of Computer Science and Engineering, University of New South Wales, Sydney, Australia, 2052; timm@cse.unsw.edu.au

[2] Department of Computer Science and Software Engineering, University of Newcastle, Callaghan, Australia 2308; rfc@cs.newcastle.edu.au

Abstract. Abductive validation for a theory $\mathcal{T}$ expressed in language $\mathcal{L}_{\rm QCM}$ is known to be exponential. Despite this, abductive validation over $\mathcal{L}_{\rm QCM}$ using the HT4 abductive inference engine is a useful technique for a large range of real-world theories. However, doubts persist about $\mathcal{L}_{\rm TQCM}$: a time-based variant of $\mathcal{L}_{\rm QCM}$. In $\mathcal{L}_{\rm TQCM}$, abductive validation is executed for for theories used in long time-based simulations. Here we show that, in the special case where (i) the theory is only measured at a few time intervals and (ii) $\mathcal{L}_{\rm TQCM}$is restricted to $\mathcal{L}^{-}_{\rm TQCM}$ (which contains only bi-state objects connected by symmetrical relations) then temporal abductive validation is practical.

1 Introduction

We are used to assessing representations via their soundness, completeness and their tractability. Here we offer an assessment criteria for a language based on its "testability". The assessment criteria proceeds as follows. (i) Define a representation language $\mathcal{L}$. (ii) Define a validation engine for $\mathcal{L}$. (iii) Explore the computational limits of that validation engine. (iv) Modify $\mathcal{L}$ to $\mathcal{L}^{-}$ such that the computational limits of the validation engine are reduced. This approach echoes earlier research by Brachman & Levesque on the tractability of testing subsumption in frame-based languages [BL84]. Brachman & Levesque found that seemingly minor variations in a representational language can have an enormous impact of the computational complexity of inference over that representation.

In this paper we explore the testability of a temporal extension to the $\mathcal{L}_{\rm QCM}$ language (§3). Theories expressed in $\mathcal{L}_{\rm QCM}$ can be tested using our HT4 abductive validation engine [Men96b, MC97]. While abductive validation approach has proved useful for an interesting range of real world theories (§2), it cannot be extended to $\mathcal{L}_{\rm TQCM}$: i.e. temporal abductive validation which tests theories used in a time-based simulation (§4) (the differences between abductive validation and temporal abductive validation is discussed in §4.1 and §5 respectively). Such time-based simulations occur in at least two common KBS systems. (i) Rule-bases that are processed via a standard match-select-act loop may assert and retract facts; i.e. at different times within the execution of the rule-base, literals have different belief values. (ii) Domain knowledge may be expressed as qualitative theories which may contain loops. As the inference executes around the loop, literals may be assigned different belief values at different times.

In this paper we prove that we can reduce the cost of temporal validation by reducing the granularity of the time axis to the granularity of the measurements of that theory. More precisely, based on a graph-theoretic analysis (§6), a restricted form of $\mathcal{L}_{\mathrm{TQCM}}$ ($\mathcal{L}^{-}_{\mathrm{TQCM}}$) is suitable for temporal abductive validation using HT4, provided the theory is only measured at a few time intervals.

2 Is $\mathcal{L}_{\mathrm{QCM}}$ Appropriate for KBS?

This paper explores general computational issues relating to KBS validation via an an analysis of the computational properties of abductive validation over the $\mathcal{L}_{\mathrm{QCM}}$ language. This section justifies the use of $\mathcal{L}_{\mathrm{QCM}}$ as a general KBS validation device.

Informally, abductive validation answers the following *validation question*: "can a model of X reproduce the known or desired behaviour of X?". Abductive validation was initially defined as a method of analysing qualitative neuroendocrinological theories (§4.3). However, once developed, it was realised that abductive validation answers our validation question for any theory which can be reduced to a directed and-or graph of literals. Many common knowledge representations can be mapped into such graphs. For example, propositional rule bases can be viewed as graphs connecting literals from the rule left-hand-side to the rule right-hand-side. More generally, horn clauses can be viewed as a graph where the conjunction of sub-goals leads to the head goal. In the special (but common) case where the range of all variables is known, this graph can be partially evaluated into a ground form. Once in the ground form, the literals in the ground form can be viewed as vertices in an and-or graph.

Internally we implement this mapping via an intermediary translation into $\mathcal{L}_{\mathrm{QCM}}$. That is, abductive validation over the qualitative language $\mathcal{L}_{\mathrm{QCM}}$ can be viewed as a virtual machine for addressing our validation question for a variety of representations. Hence, we explore $\mathcal{L}_{\mathrm{QCM}}$ as a means of exploring computational issues in validating numerous KBS forms. This exploration implies two equivalences: (i) KBS = qualitative modeling; (ii) KBS = abduction.

These equivalences are supported by other research. Clancey argues convincingly [Cla92] that the runtime activity of an expert system is the construction and reflection of a specific qualitative model from a larger and more general qualitative model (knowledge base to situation-specific model). A wide range of knowledge-level tasks can be mapped into an abductive framework such as prediction, classification, explanation, tutoring, qualitative reasoning, planning, diagnosis, monitoring, case-based-reasoning, and verification [Men96b]. Hence, we are satisfied that there is some validity to the above two equivalences.

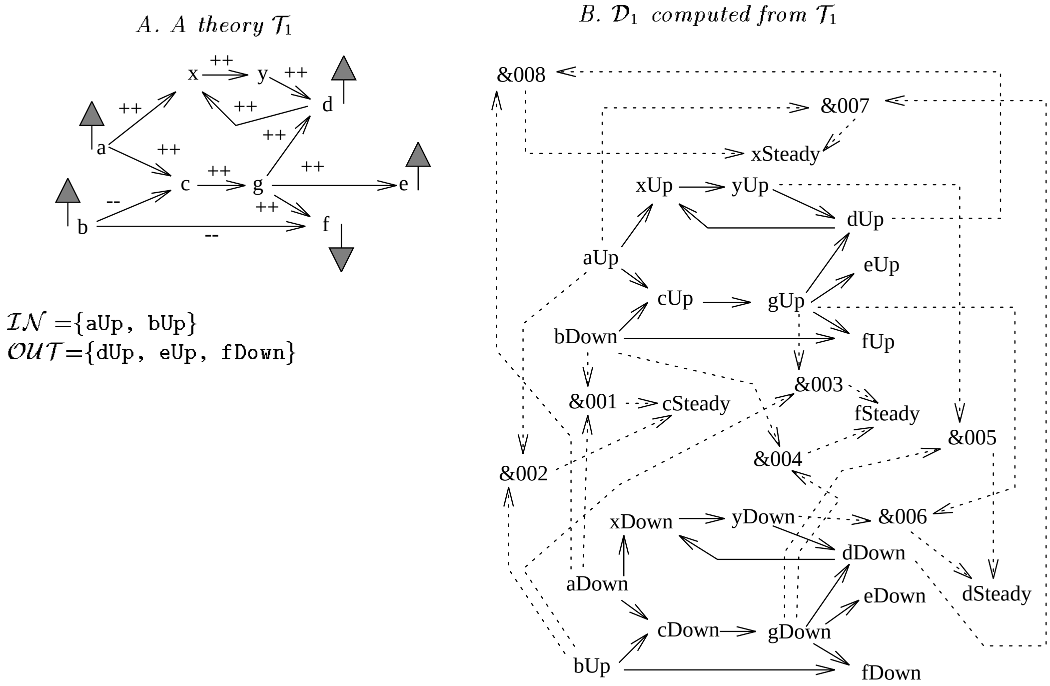

Fig. 1. Theories to dependency graphs

3 $\mathcal{L}_{\mathrm{QCM}}$

$\mathcal{L}_{\mathrm{QCM}}$ is a qualitative compartmental modeling language [MC97]. Compartmental models include tubs with in-flows and out-flows. In quantitative compartmental modeling, the current level of liquid in a tub equals its initial value plus its in-flows, minus its out-flows. In qualitative compartmental modeling, the sign of the the rate of change (first derivative) is used to control the qualitative simulation. Qualitative theories are indeterminate; i.e. multiple incompatible belief sets can be generated. These belief sets must be maintained in one world for each indeterminate possibility.

In $\mathcal{L}_{\mathrm{QCM}}$, experts can quickly sketch out their intuitions in a *theory digraph* $\mathcal{T}$ representing *objects* which can be in one of 3 mutually exclusive states: up, down, or steady denoted $\uparrow$,$\downarrow$, or $\oslash$ respectively (or, more generally, we say that an object `x` can have the state $\mathtt{x}\sigma$). Edges between objects denote inferences between object states which would be acceptable to the KB author; for example: (i) the *direct* link $\mathtt{x} \stackrel{++}{\rightarrow} \mathtt{y}$ denotes that `y` being $\uparrow$ or $\downarrow$ can be explained by `x` being $\uparrow$ or $\downarrow$ respectively; (ii) The *inverse* link $\mathtt{x} \stackrel{--}{\rightarrow} \mathtt{y}$ denotes that `y` being $\uparrow$ or $\downarrow$ could be explained by `x` being $\downarrow$ or $\uparrow$ respectively. It is important to stress that these edges represent *possible* but not *mandatory* inferences. That is, we give these edges an abductive (not deductive) semantics (§4). An edge is used if it leads, eventually, to the satisfaction of some global satisfaction criteria; e.g. in abductive validation, maximum coverage.

Direct and inverse links are *symmetric edges*; i.e a inference from one state of x to one state of y tells us something about the opposite inference. For example, $x \overset{++}{\rightarrow} y$ tells us something about y if x goes $\uparrow$ *or* $\downarrow$. *Asymmetric edges* tell us nothing about the opposite inference; e.g. the *creator* link $x \overset{+-+}{\rightarrow} y$ denotes that y being $\uparrow$ can be explained by x being $\uparrow$ but not visa versa and the *destroyer* link $x \overset{+--}{\rightarrow} y$ denotes that y being $\downarrow$ could be explained by x being $\uparrow$ but not visa versa. Creator and destroyer links denote in-flows and out-flows to tubs. Only creators can increase the amount of fluid in a tub and only destroyers can decrease the amount of tub fluid.

If we assume that (i) the conjunction of an $\uparrow$ and a $\downarrow$ can explain a $\oslash$ (steady) and that (ii) no change can be explained in terms of a steady (i.e. a steady vertex has no children), then we can convert Figure 1.A into the dependency graph $\mathcal{D}$ of literals shown in Figure 1.B. This graph contains one vertex for each possible state of the objects of Figure 1.A as well as *and* vertices which models combinations of influences (for example, g$\downarrow$ and b$\downarrow$ can lead to f being steady (denoted f$\oslash$). The dotted lines in Figure 1.A denote edges around *and* vertices.

The following lemma shows the relationship between paths in T and D. The proof are immediate from the definitions (see, for example, Figure 1.A).

Lemma 1. *If there is a path from vertex* $x\sigma$ *to vertex* $y\tau$ *in dependency digraph* D *then there is a path from vertex* x *to vertex* y *in the associated theory digraph* T.

4 Abductive Validation over $\mathcal{L}_{\mathrm{QCM}}$

Abduction is the search for assumptions $\mathcal{A}$ which, when combined with some theory $\mathcal{T}$ written in some language $\mathcal{L}$ explains some subset $\mathcal{OUT}'$ of our $\mathcal{OUT}$put goals without causing some contradiction [Esh93]. That is: $EQ_1{:}\mathcal{T} \cup \mathcal{A} \vdash \mathcal{OUT}'$; $EQ_2{:}\mathcal{T} \cup \mathcal{A} \not\vdash \bot$.

Mutually exclusive, generated assumptions must be managed separately. Each maximal consistent subset of $\mathcal{A}$ defines a world $\mathcal{W}_i$ which *cover* (explain) $\mathcal{OUT}'_i$. Abductive validation [Men96a] is the search for worlds that cover the greatest percentage of the $\mathcal{OUT}$puts. The maximum cover of the worlds of a theory is the cover of that theory. If $\mathcal{T}_i^{cover}$<100% then we can report that a theory $\mathcal{T}_i$ of X cannot reproduce all of the known/desired behaviour of X. Further, $\mathcal{T}_i$ is said to be a better theory than theory $\mathcal{T}_j$ if $\mathcal{T}_i^{cover} \gg \mathcal{T}_j^{cover}$.

4.1 An Example

Abductive validation executes over theories with invariants of the form "certain literals cannot be believed together". In $\mathcal{L}_{\mathrm{QCM}}$, the invariant rule is that no two states of an object can be believed together. This section discusses abductive validation and is adapted from [Men96b]. Temporal abductive validation executes over theories with invariants of the form "certain literals cannot be believed *at the same time interval*". Temporal abductive validation is discussed below (§5).

For example, suppose HT4 wants to validate that certain $\mathcal{OUT}$put goals can be reached from using the $\mathcal{IN}$puts shown in the theory $\mathcal{T}_1$ of Figure 1.A. If we assume that In the case where $\mathcal{OUT}$ = $\{\texttt{d}\uparrow,\texttt{e}\uparrow,\texttt{f}\downarrow\}$ and $\mathcal{IN}=\{\texttt{a}\uparrow,\texttt{b}\uparrow\}$, then HT4 can find the following paths $\mathcal{P}$ connecting $\mathcal{OUT}$s to $\mathcal{IN}$s: $\mathcal{P}_1= \texttt{a}\uparrow\rightarrow \texttt{x}\uparrow\rightarrow \texttt{y}\uparrow\rightarrow \texttt{d}\uparrow$, $\mathcal{P}_2= \texttt{a}\uparrow\rightarrow \texttt{c}\uparrow\rightarrow \texttt{g}\uparrow\rightarrow \texttt{d}\uparrow$, $\mathcal{P}_3= \texttt{a}\uparrow\rightarrow \texttt{c}\uparrow\rightarrow \texttt{g}\uparrow\rightarrow \texttt{e}\uparrow$, $\mathcal{P}_4= \texttt{b}\uparrow\rightarrow \texttt{c}\downarrow\rightarrow \texttt{g}\downarrow\rightarrow \texttt{f}\downarrow$, $\mathcal{P}_5= \texttt{b}\uparrow\rightarrow \texttt{f}\downarrow$. These paths may contain assumptions, i.e. literals that are not known $\mathcal{FACTS}$.

Continuing our example when $\mathcal{FACTS}=\mathcal{IN} \cup \mathcal{OUT}$ (the usual case), then $\{\texttt{x}\uparrow,\texttt{y}\uparrow,\texttt{c}\uparrow, \texttt{c}\downarrow, \texttt{g}\uparrow, \texttt{g}\downarrow\}$ are assumptions. If we apply the invariants of $\mathcal{L}_{\mathrm{QCM}}$, then we can declare $\{\texttt{c}\uparrow, \texttt{c}\downarrow, \texttt{g}\uparrow, \texttt{g}\downarrow\}$ to be conflicting (denoted $\mathcal{A}_c$). Figure 1.A shows us that g is fully dependent on c. Hence the key conflicting assumptions are $\{\texttt{c}\uparrow,\texttt{c}\downarrow\}$ (denoted *base controversial assumptions* or $\mathcal{A}_b$). We can used $\mathcal{A}_b$ to find consistent belief sets called worlds $\mathcal{W}$. A path $\mathcal{P}_i$ is in $\mathcal{W}_j$ if that path does not conflict with the environment $\mathcal{ENV}_j$ (an environment is a maximal consistent subset of $\mathcal{A}_b$). In our example, $\mathcal{ENV}_1=\{\texttt{c}\uparrow\}$ and $\mathcal{ENV}_2=\{\texttt{c}\downarrow\}$. Hence, $\mathcal{W}_1=\{\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, \mathcal{P}_5\}$ and $\mathcal{W}_2=\{\mathcal{P}_1\ \mathcal{P}_4\ \mathcal{P}_5\}$ (see Figure 2[3]). The overlap of $\mathcal{W}_1$ and $\mathcal{OUT}$ is the cover of $\mathcal{W}_1$ and contains $\{\texttt{d}\uparrow,\texttt{e}\uparrow,\texttt{f}\downarrow\}$. The overlap of $\mathcal{W}_2$ and $\mathcal{OUT}$ is $\{\texttt{d}\uparrow,\texttt{f}\downarrow\}$ That is, $\mathcal{W}_1^{cover}$=3=100% and $\mathcal{W}_2^{cover}$=2=67%. The maximum cover is 100%; i.e. their exist a set of assumptions ($\{\texttt{c}\uparrow\}$) which let us explain all of $\mathcal{OUT}$ and this theory has passed HT4-style abductive validation.

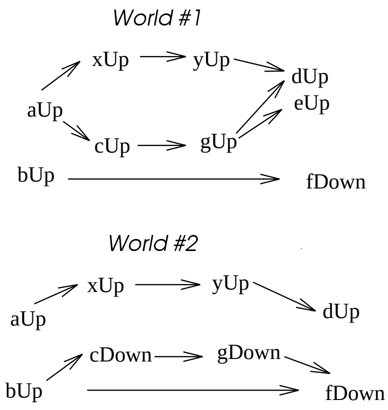

Fig. 2. Two consistent worlds generated from Figure 1.B.

4.2 Complexity

The core problem in HT4 is finding the base controversial assumption set $\mathcal{A}_B$. In the *forward sweep*, HT4 finds the conflicting assumption set $\mathcal{A}_C$ as a side-effect of computing the transitive closure of $\mathcal{IN}$. In the *backwards sweep*, HT4 constrains path generation to the transitive closure of $\mathcal{IN}$. As a path is grown from a member of $\mathcal{OUT}$ back to $\mathcal{IN}$, five invariants are maintained. (i) Paths maintain a *forbids* set; i.e. a set of literals that are incompatible with the literals used in the path. For example, the literals used in $\mathcal{P}_1$ forbid the literals $\{\texttt{a}\downarrow, \texttt{a}\oslash, \texttt{x}\downarrow, \texttt{x}\oslash, \texttt{y}\downarrow, \texttt{y}\oslash, \texttt{d}\downarrow, \texttt{d}\oslash\}$. (ii) A path must not contain loops or items

[3] The connection of HT4 to DeKleer's ATMS system [DeK86] is explored elsewhere [Men96b]

that contradict other items in the path (i.e. a path's members must not intersect with its *forbids* set). (iii) If a path crosses an *and* node, then all the parents of that node must be found in the path (iv) A literal in a path must not contradict the known $\mathcal{FACTS}$. (v) The upper-most $\mathcal{A}_C$ found along the way is recorded as that path's *guess*. The union of all the guesses of all the paths is $\mathcal{A}_B$.

Once $\mathcal{A}_B$ is known then the paths can be sorted into worlds via the *worlds sweep*. HT4 extracts all the objects $\mathcal{O}$ referenced in $\mathcal{A}_B$. A world-defining environment $\mathcal{ENV}_i$ is created for each combination of objects and their values. In our example, $\mathcal{ENV}_1 = \{\mathsf{c}\uparrow\}$ and $\mathcal{ENV}_2 = \{\mathsf{c}\downarrow\}$. The worlds sweep is simply two nested loops over each $\mathcal{ENV}_i$ and each $\mathcal{P}_j$. A path $\mathcal{P}_j$ belongs in world $\mathcal{W}_i$ if its *forbids* set does not intersect the assumptions $\mathcal{ENV}_i$ that define that world. For more details on the internals of HT4, see [MC97].

HT4's runtimes are clearly exponential on the number of edges in the theory. In a theory comprising a directed and-or graph with vertices $\mathcal{V}$, edges $\mathcal{E}$, and fanout $\mathcal{F} = \frac{|\mathcal{E}|}{|\mathcal{V}|}$, the worst-case complexity of the forwards sweep is acceptable at $O(|\mathcal{V}|^3)$. However, if the average size of a path is X, then worse case backwards sweep is $O(X^F)$). Further, the worlds sweep is proportional to the number of paths and the number of world-defining assumptions; i.e. EQ_3:$O(|\mathcal{P}| * |\mathcal{ENV}|)$= $O(X^F * |\mathcal{ENV}|)$.

4.3 In Practice

The exponential complexity of abductive validation is not surprising. Abductive validation is a variant on abduction and abduction is known to be NP-hard [BAMTJ91]. Nevertheless, it has been shown that abductive validation is practical for many real-world theories such as certain fielded expert systems and theories from neuroendocrinology [Men96b]. Feldman & Compton [FCS89], followed by Menzies [MC97], have shown that abductive validation engines could detect previously unseen errors in theories in neuroendocrinology (the study of nerves and glands) published in international refereed journals. Surprisingly, these faults were found using the data published to support those theories

Experiments with HT4, an abductive validation engine, showed that for a dependency graph $\mathcal{D}_i$ with vertices $\mathcal{V}$ generated from a language $\mathcal{L}_{\mathrm{QCM}}$, this algorithm fails after $|\mathcal{V}| > 850$. The claim that abductive validation is practical for real-world theories rests on the empirical observation that many interesting theories have $|\mathcal{V}| < 850$ (e.g. the neuroendocrinological systems studied by Feldman & Compton [FCS89] and Menzies [MC97]; the sample of expert systems studied by the verification community [PS92]). The claim that temporal abductive validation is impractical rests on the observation that theories used for simulation runs will have $|\mathcal{V}| > 850$ (§5).

5 $\mathcal{L}_{\mathrm{TQCM}}$

For theories used in simulation runs, we wish to reason about the time-dependent behaviour of an object; e.g. in Figure 3, the values of x are {$x_1 \uparrow$, $x_2 \uparrow$, $x_3 \downarrow$, $x_4 \uparrow$, $x_5 \downarrow$} where $\uparrow$ and $\downarrow$ are computed by comparing $x_{\mathcal{K}}$ with $x_{\mathcal{K}-1}$. One approach to implementing temporal abductive validation would be to create one copy of $\mathcal{D}$ for each time interval in the simulation. Literals in adjacent time intervals would be connected; i.e. belief that an object in a certain state at time $\mathcal{K}$ could justify an explanation of that object being in the same state in time $\mathcal{K}+1$. The invariants of $\mathcal{L}_{\mathrm{QCM}}$ would be extended to say that we can't believe that an object is in two states in the same time interval. For example, for a simulation of $\mathcal{D}_2$ (Figure 4.A) over 4 time intervals, we could execute HT4 over $\mathcal{D}_3$ (Figure 4.B).

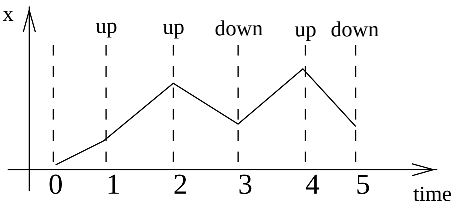

Fig. 3. Values for x generated over a simulation run.

The advantage of this copy-based approach is that we could use HT4 without modification. The disadvantage of this approach is that the number of edges would increase linearly with the number of copies. Recalling EQ_3, this means that the worst-case complexity increases exponentially with the number of copies. Further, recalling the experimental $\mathcal{L}_{\mathrm{QCM}}$ limit of $|\mathcal{V}| > 850$, if $\mathcal{D}_i$ is copied $\mathcal{K}$ times, then $|\mathcal{V}| > 850$ shrinks to $|\mathcal{V}| > \frac{850}{\mathcal{K}}$. This implies, for example, that using HT4 for temporal abductive validation over 100 time intervals becomes impractical after $|\mathcal{V}| > 8$ (which is clearly a significant restriction to modeling and the practicality of this technique).

Note that the copies repeat some structure for the known time intervals in the simulation. A compelling intuition is that no explanation path can be found through $X+1$ copies that can't be found in X copies. That is, we can ignore the copies representing the intermediary time intervals between measurements. If this was true then (i) explanations of events at time $\mathcal{K}=4$ in terms of inputs at time $\mathcal{K}=1$ over $\mathcal{D}_3$ does not need the copies at $\mathcal{K}=2$ and $\mathcal{K}=3$; (ii) $\mathcal{D}_3$ could be re-written as the smaller $\mathcal{D}_4$ (Figure 5.A).

This intuition is compelling since, if true, if we simulate for $\mathcal{K}$ time intervals but only measure values at $\mathcal{K}'$ intervals ($\mathcal{K}' < \mathcal{K}$), then we need not search the copies for the unmeasured intervals. So, for a simulation run of 1000 time intervals where we only measure values at the start and at the finish, abductive validation would only need two copies of $\mathcal{D}$.

Sadly, we can quickly find dependency graphs where the intermediary copies are required to make explanations. Consider Figure 4.A and the case where $\mathcal{OUT}$ = {a$\uparrow$} and $\mathcal{IN}$={b$\uparrow$}. The only path which connects these two literals is {b$\uparrow \rightarrow$ a$\downarrow \rightarrow$ b$\downarrow \rightarrow$ a$\uparrow$}. This path takes several time intervals since the temporal abductive validation invariant is that no object can be in different states

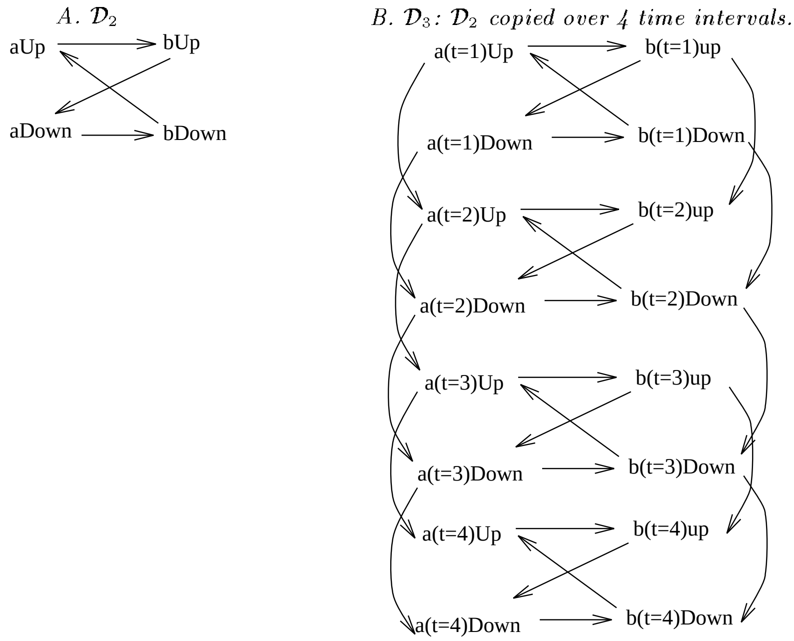

Fig. 4. Copying dependency graphs.

in the same time interval. Our path takes a minimum of 3 time intervals: one for $\mathtt{b}_1 \uparrow\rightarrow \mathtt{a}_1 \downarrow$; one for $\mathtt{a}_2 \downarrow\rightarrow \mathtt{b}_2 \downarrow$; and one for $\mathtt{b}_3 \downarrow\rightarrow \mathtt{a}_3 \uparrow$. That is, if we took measurements of this simulation at time intervals 1 and 3, we would need one intermediary copy at time interval 2 to complete the explanation. This example suggests that between each measured time interval, may need at least one intermediary copy (as in Figure 5.B). To make matters worse, note that Figure 6 duplicates the topology of Figure 4.A in the regions A, B, C with an extra link from the top-left vertex of one region to the top-right vertex of the next region. A path from $\mathtt{b}\uparrow$ to $\mathtt{e}\uparrow$ will take at 3 time intervals to cross each of A, B, C. By repeating A, B, C more times, we can generate dependency graphs which would require any number of intermediaries to traverse. This example suggests that between each measured time interval for theories written in $\mathcal{L}_{\mathrm{TQCM}}$, we may need more than one intermediary copy.

6 $\mathcal{L}^{-}_{\mathrm{TQCM}}$

Whilst exploring examples like Figure 6, we observed that we could only generate graphs requiring more than one intermediary copy if we used asymmetrical edges. For example, the edge (e.g.) $\mathtt{a}\uparrow\rightarrow\mathtt{d}\uparrow$ in Figure 6 is asymmetric since there is no second edge from $\mathtt{a}\downarrow\rightarrow\mathtt{d}\downarrow$. This prompted the exploration of $\mathcal{L}^{-}_{\mathrm{TQCM}}$, a restrictive form of $\mathcal{L}_{\mathrm{TQCM}}$in which asymmetric edges were illegal. In the special case where objects were restricted to the two states $\uparrow$ and $\downarrow$, we could find no examples in

A. $\mathcal{D}_4$: $\mathcal{D}_3$ without the copies at time $\mathcal{K} = 2$ and $\mathcal{K} = 3$.

B. $\mathcal{D}_4$: $\mathcal{D}_3$ with one intermediary copy at time 1+ representing the space between time between $\mathcal{K} = 1$ and $\mathcal{K} = 4$.

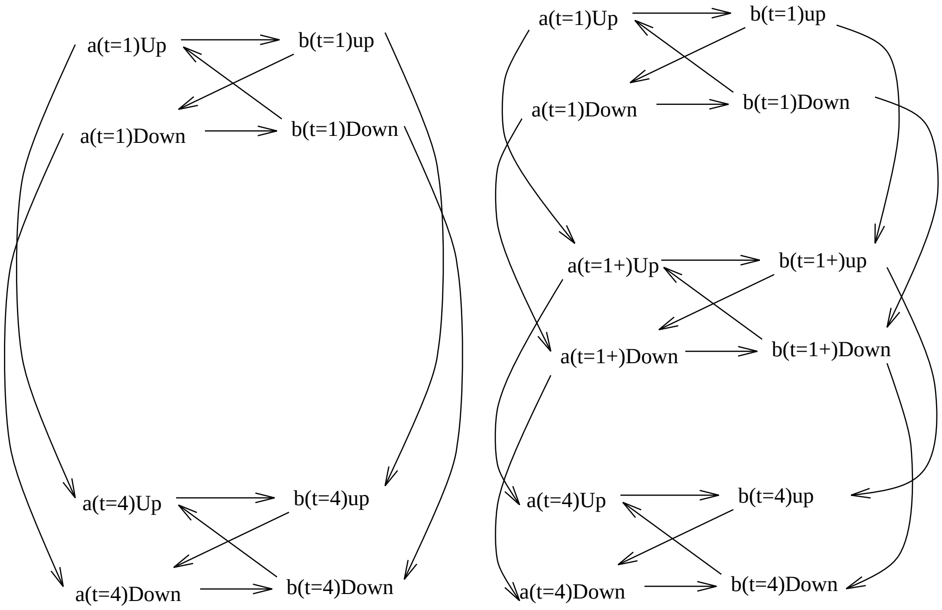

Fig. 5. Restrictive copying.

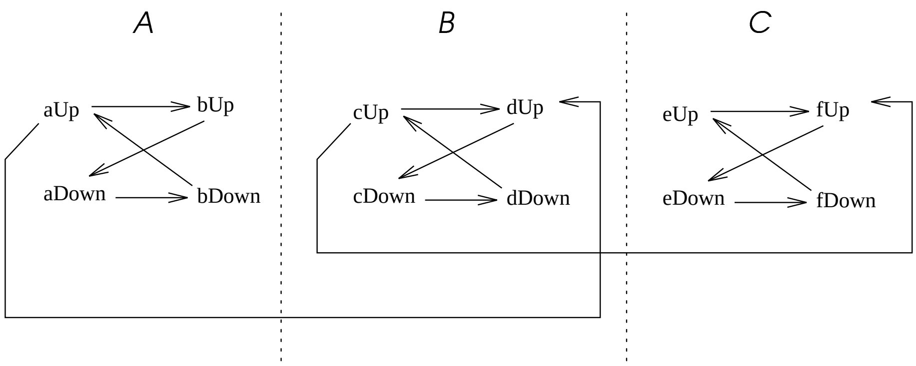

Fig. 6. A dependency graph which requires intermediaries to explain e↑ in terms of b↑.

$\mathcal{L}^{-}_{\mathrm{TQCM}}$ where we needed more than one intermediary copy. Roughly speaking, we will show that if every edge offers a comment on all the states of its downstream vertices, then the state space rapidly saturates and explanations can be quickly generated. However, if we permit asymmetric edges, then the state space has unknown zones within it and explanations may take longer.

6.1 A Graph Theoretic Proof that 3 Copies of $\mathcal{D}$ is Enough

This section proves that paths either succeeded in 3 copies or never at all (§6.1). The implications of this is that between any two simulation times where measurements of the theory are taken (labeled 1 and 3), we only need one intermediary copy labeled 2. Hence, Figure 5.B is the general form: between any two copies with measurements, we only need one extra intermediary copy to prove a connection. So, for a simulation measured objects at times 1, 500, and 1000, we only need two intermediary copies representing the search space from times 1 to 500 and from 500 to 1000. That is, this simulation for 1000 time intervals measured at only three time ticks requires a mere 5 copies.

This section assumes that $\mathcal{L}$=$\mathcal{L}^{-}_{\mathrm{TQCM}}$.

The following lemma shows a relationship between paths in $\mathcal{T}$ and $\mathcal{D}$ for theories expressed in $\mathcal{L}^{-}_{\mathrm{TQCM}}$. The proof of this lemma is obvious from inspection of Figure 1.A and discarding all the steady and *and* vertices of Figure 1.B. Note that lemma 1 still holds for $\mathcal{L}^{-}_{\mathrm{TQCM}}$.

Lemma 2. *If there is a path from vertex* x *to vertex* y *in theory digraph* $\mathcal{T}$*, then, in the associated dependency digraph* $\mathcal{D}$*, there are either paths: (i) from* x↑ *to* y↑ *and from* x↓ *to* y↓*, or ...; (ii) from* x↑ *to* y↓ *and from* x↓ *to* y↑*.*

We will use $\bar{\sigma}$ to refer to the opposite direction of σ, that is, if $\sigma =\uparrow$ then $\bar{\sigma} =\downarrow$ and if $\sigma =\downarrow$ then $\bar{\sigma} =\uparrow$. $\mathcal{L}^{-}_{\mathrm{TQCM}}$ restricts itself to only ↑ and ↓ and symmetric edges since our proof depends on being able to say something about $\bar{\sigma}$.

Consider vertices $\mathtt{x}\sigma$ and $\mathtt{y}\tau$ in dependency digraph $\mathcal{D}$. A *simple proof* of $\mathtt{y}\tau$ from $\mathtt{x}\sigma$, $\Pi^s(\mathtt{x}\sigma,\mathtt{y}\tau)$, is a directed path from $\mathtt{x}\sigma$ to $\mathtt{y}\tau$ such that for any vertex z at most one of z↑ and z↓ are on $\Pi^s(\mathtt{x}\sigma,\mathtt{y}\tau)$. Simple proofs are those generated within one time interval of a simulation.

A *proof* of vertex $\mathtt{z}\upsilon$ from vertex $\mathtt{w}\rho$ $\Pi(\mathtt{w}\rho,\mathtt{z}\upsilon)$ is an ordered collection of simple proofs such that $\mathtt{w}\rho$ is the start of the first path, $\mathtt{z}\upsilon$ is the end of the final path, and the final vertex of each simple proof is the start vertex of its successor (see Fig 7). We think of each simple proof of a proof as being generated in the same time interval. The requirement that the final vertex of a simple proof being identical as the start vertex of its successor represents the connection between time intervals. Note that for a vertex z, both z↑ and z↓ can be contained in proof $\Pi(\mathtt{w}\rho,\mathtt{z}\upsilon)$, but cannot be contained in the same simple proof of $\Pi(\mathtt{w}\rho,\mathtt{z}\upsilon)$. That is, it cannot be found in the same time interval of the simulation.

The *time* of a proof, $time(\Pi(w\rho, z\omega))$ is the number of simple proofs in $\Pi(w\rho, z\omega)$. For vertices $\mathtt{x}\sigma$ and $\mathtt{y}\tau$, the minimum proof time, $minTime(\mathtt{x}\sigma,\mathtt{y}\tau)$, is the minimum time of any proof of $\mathtt{y}\tau$ from $\mathtt{x}\sigma$. If there is no proof of $\mathtt{y}\tau$ from $\mathtt{x}\sigma$, then we say $minTime(\mathtt{x}\sigma,\mathtt{y}\tau)= \infty$.

Lemma 3. *There is a path from* $\mathtt{x}\sigma$ *to* $\mathtt{y}\tau$ *in dependency digraph* $\mathcal{D}$ *if and only if* $minTime(\mathtt{x}\sigma,\mathtt{y}\tau)$ *is finite.*

Proof. The "if" direction is an immediate consequence of the definition of a proof: we form a path from $\mathtt{x}\sigma$ to $\mathtt{y}\tau$ by concatenating the simple proofs of a proof of $\mathtt{y}\tau$ from $\mathtt{x}\sigma$.

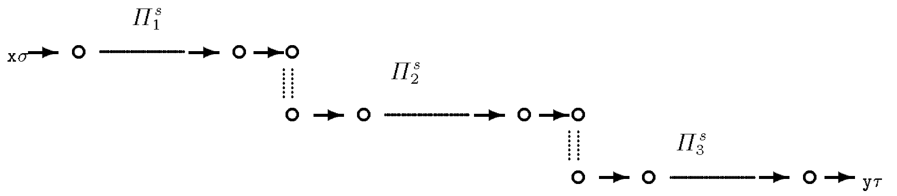

Fig. 7. Simple proofs and proofs. (a) A path $\mathcal{P}$ from xσ to yτ. (b) Dividing $\mathcal{P}$ into simple proofs. Note that for any z both z$\uparrow$ and z$\downarrow$ cannot be on the same simple proof.

For the "only if" direction, let $\mathcal{P}$ be a path from xσ to yτ. We can create a proof of yτ from xσ by making each edge of $\mathcal{P}$ a simple proof. (This is valid since the definition of a theory digraph does not allow self-loops.) Therefore, *minTime*(xσ,yτ) is less than or equal the length of $\mathcal{P}$.

Lemma 4. *If minTime(xσ,yτ) is finite, then either minTime(xσ,y$\uparrow$)= 1 or minTime(xσ,y$\downarrow$)= 1.*

Proof. If x = y then *minTime*(xσ,xσ)= 1.

Otherwise, by lemmas 1 and 3 there is a path from x to y in theory digraph $\mathcal{T}$. Let $\mathcal{P}$ be a loop-free path from x to y in $\mathcal{T}$. (Such a path is guaranteed to exist since loops can always be removed.) Then, by lemma 2 there is a path from xσ to yτ such that for any vertex z of $\mathcal{T}$ at most one of z$\uparrow$ and z$\downarrow$ are on $\mathcal{P}$. Consequently, $\mathcal{P}$ is a simple proof and *minTime*(xσ,xσ)= 1.

Theorem 5 is the key result of this paper.

Theorem 5. *If xσ and yτ are vertices of dependency graph $\mathcal{D}$, then minTime(xσ,yτ) can only have values 1, 2, 3, or ∞.*

Proof. If there is no path from xσ to yτ then *minTime*(xσ,yτ)= ∞.

Suppose there is a path $\mathcal{P}_{\mathcal{D}}$ from xσ to yτ. By lemma 1, there is a path from x to y in $\mathcal{T}$, and by lemma 4, either *minTime*(xσ,y$\uparrow$)= 1 or *minTime*(xσ,y$\downarrow$)= 1. If *minTime*(xσ,yτ)= 1, then the theorem is proved.

Suppose *minTime*(xσ,yτ)> 1. We will show that *minTime*(xσ,yτ)$\leq$ 3 by exploring all paths $\mathcal{D}$ in dependency digraph $\mathcal{D}$ starting at vertex xσ, finding a proof Π to the end of $\mathcal{D}$ that has *time*(Π) ≤ 3. In this way, we are ensured that *minTime*(xσ,yτ)$\leq$ 3.

We proceed by induction on the length of shortest path from x$\sigma$to a vertex of $\mathcal{D}$:

- *Base Case:* Distance is 0. Then the destination is $\mathtt{x}\sigma$ itself and $minTime(\mathtt{x}\sigma,\mathtt{x}\sigma)=$ 1.
- *Inductive Step:* Suppose that $minTime(\mathtt{x}\sigma,\mathtt{v}\omega) \leq 3$ for all $\mathtt{v}\omega$ with shortest path from $\mathtt{x}\sigma$ less than k for some $k > 0$. Let $\mathbf{z}\upsilon$ be a vertex whose shortest path from $\mathtt{x}\sigma$ has length k. If $minTime(\mathtt{x}\sigma,\mathbf{z}\upsilon)= 1$ then we are done.
  Otherwise, $minTime(\mathtt{x}\sigma,\mathbf{z}\upsilon)> 1$. By lemma 4, we have $minTime(\mathtt{x}\sigma,\mathbf{z}\bar{\upsilon})=$ 1. Let $\mathtt{w}\rho$ be the predecessor of $\mathbf{z}\upsilon$ on some shortest path $\mathcal{P}$ from $\mathtt{x}\sigma$. If $minTime(\mathtt{x}\sigma,\mathtt{w}\rho)< 3$ then we can create a simple proof of $\mathbf{z}\upsilon$ from $\mathtt{x}\sigma$ by adding the simple proof consisting of the single edge $\mathtt{w}\rho \rightarrow \mathbf{z}\upsilon$ to a minimum time proof of $\mathtt{w}\rho$. Therefore, $minTime(\mathtt{x}\sigma,\mathbf{z}\upsilon)\leq 3$
  Suppose $minTime(\mathtt{x}\sigma,\mathtt{w}\rho)= 3$. Then there is a proof Π of $\mathtt{w}\rho$ from $\mathtt{x}\sigma$ consisting of 3 simple proofs, Π_1^s, Π_2^s, Π_3^s. If $\mathbf{z}\bar{\upsilon}$ is not in Π_3^s then we can create a proof Π' of $\mathbf{z}\upsilon$ from $\mathtt{x}\sigma$ by extending Π_3^s by adding the edge $\mathtt{w}\rho \rightarrow \mathbf{z}\upsilon$(see Fig. 8) and we have $time(\Pi') = 3$ so $minTime(\mathtt{x}\sigma,\mathbf{z}\upsilon)\leq 3$.
  Finally, suppose $\mathbf{z}\bar{\upsilon}$ is contained in simple proof Π_3^s. Notice that $\mathbf{z}\bar{\upsilon} \neq \mathtt{w}\rho$ since this would imply a self-loop in theory digraph $\mathcal{T}$. Let $\bar{\Pi}^s$ be a simple proof of $\mathbf{z}\bar{\upsilon}$ from $\mathtt{x}\sigma$ ($\bar{\Pi}^s$ exists since $minTime(\mathtt{x}\sigma,\mathbf{z}\bar{\upsilon})= 1$). Let $\mathtt{u}\xi$ be the successor of the final occurrence of $\mathbf{z}\bar{\upsilon}$ on Π_3^s. We can construct a proof of $\mathbf{z}\upsilon$ from $\mathtt{x}\sigma$ consisting of the following 3 paths (see Fig. 8): (i) simple proof $\bar{\Pi}^s$; (ii) the edge $\mathbf{z}\bar{\upsilon} \rightarrow \mathtt{u}\xi$; (iii) the remainder of Π_3^s starting at $\mathtt{u}\xi$ and the edge $\mathtt{w}\rho \rightarrow \mathbf{z}\upsilon$. Since all three are simple proofs, and they form a proof of $\mathbf{z}\upsilon$ from $\mathtt{x}\sigma$, we get $minTime(\mathtt{x}\sigma,\mathbf{z}\upsilon)\leq 3$.

7 Related Work

We are unaware of other research into the complexity of temporal validation. However, for the non-temporal case, our approach is similar in spirit to the non-monotonic KBS validation research. For example, automatic test suite generation procedures based on dependency-network are offered by of Ginsberg [Gin87] and Zlatereva [Zla93]. The dependencies between rules/conclusions are computed and divided into mutually consistent subsets. The root dependencies of these subsets represent the space of all reasonable tests. If these root dependencies are not represented as inputs within a test suite, then the test suite is incomplete. Test cases can then be automatically proposed to fill any gaps.

The advantage of this technique is that it can be guaranteed that test cases can be generated to exercise all branches of a knowledge base. The disadvantage of this technique is that, for each proposed new input, an expert must still decide what constitutes a valid output. This decision requires knowledge external to the model, least we introduce a circularity in the test procedure (i.e. we test the structure of $\mathcal{T}_i$ using test cases derived from the structure of $\mathcal{T}_i$). Further, auto-test-generation focuses on incorrect features in the current model. We prefer to use test cases from a totally external source since such test cases can highlight what is absent from the current model. For these reasons, we caution against automatic test suite generation.

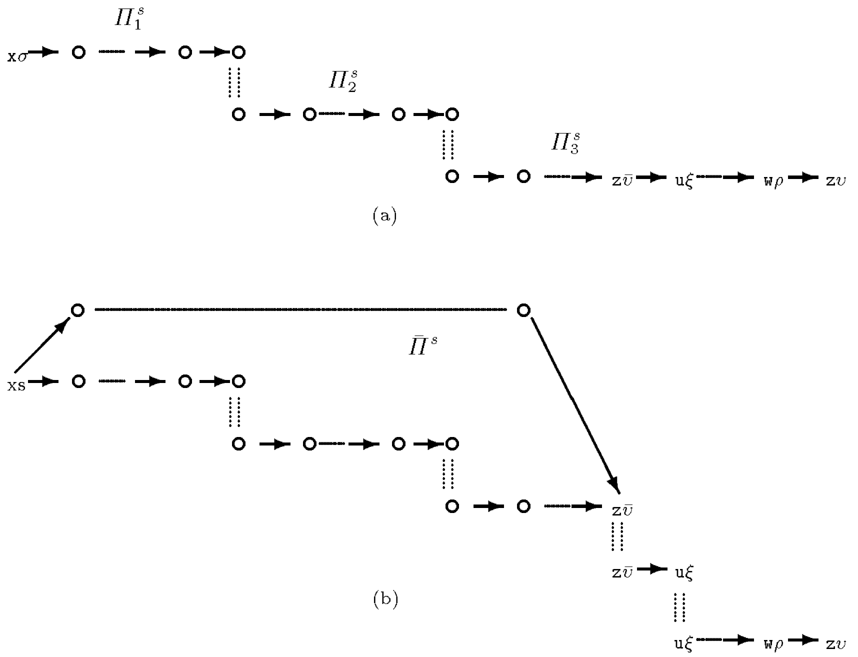

Fig. 8. Illustration of two of the cases in the proof of Theorem 5 . (a) Adding an edge when z$\bar{v}$ is not in Π_3^s. (b) Using a shortcut when z$\bar{v}$ is in Π_3^s.

Our preferred framework uses a graph-theoretic approach; i.e. inference is the selection of some subset of edges from the network of possible proof trees. We find that underneath efficient theorem provers is some sort of graph representation. Hence, we have elected to work directly at the graph-level rather than at a general logical level (e.g. Zlatereva). The simplicity of the graph-theoretic framework enables detailed complexity analyses such as this one.

8 Conclusion

We have proposed a restriction to $\mathcal{L}_{\text{QCM}}$ in which vertices are restricted to bi-state devices (removing steady and *and* vertices) and edges are restricted to symmetrical links (direct and inverse only).

The restricted language, $\mathcal{L}^{-}_{\text{TQCM}}$ has three advantages. (i) $\mathcal{D}$ will be smaller since there is no need for steady and *and* vertices. This is a significant saving since experience with abductive validation in neuroendocrinological models shows that at least half the vertices of $\mathcal{D}$ are *and* vertices. (ii) For a simulation runs over $\mathcal{K}$ time intervals where measurements of the objects are only taken at $\mathcal{K}'$ intervals, then abductive validation need only be executed over $2 * \mathcal{K}' - 1$ copies of $\mathcal{D}$. In the case where $\mathcal{K}' \ll \mathcal{K}$ this represents a significant optimisation of the processing. Roughly speaking, if every edge offers a comment on all the states of its downstream vertices (i.e. they are symmetric), then the state space rapidly saturates and the we can reduce the granularity of the time axis to just under twice the granularity of the measurements of that theory. (iii) In the case where $\mathcal{K}^{'} \ll \mathcal{K}$, we can use the HT4 abductive inference algorithm, without modification, to validate theories written in $\mathcal{L}^{-}_{\mathrm{TQCM}}$ and used for simulations.

9 Acknowledgments

Thanks to Paul Compton whose conversations where the initial spark for this work. The comments of the anonymous referees prompted §2 and a better expression of the basic intuition of this paper.

References

[BAMTJ91] T. Bylander, D. Allemang, M.C. M.C. Tanner, and J.R. Josephson. The Computational Complexity of Abduction. *Artificial Intelligence*, 49:25–60, 1991.

[BL84] R.J. Brachman and H.J. Levesque. The tractability of subsumption in frame-based description languages. In *AAAI '84*, pages 34–37, 1984.

[Cla92] W.J. Clancey. Model Construction Operators. *Artificial Intelligence*, 53:1–115, 1992.

[DeK86] J. DeKleer. An Assumption-Based TMS. *Artificial Intelligence*, 28:163–196, 1986.

[Esh93] K. Eshghi. A Tractable Class of Abductive Problems. In *IJCAI '93*, volume 1, pages 3–8, 1993.

[FCS89] B. Feldman, P. Compton, and G. Smythe. Hypothesis Testing: an Appropriate Task for Knowledge-Based Systems. In *4th AAAI-Sponsored Knowledge Acquisition for Knowledge-based Systems Workshop Banff, Canada*, 1989.

[Gin87] A. Ginsberg. A new Approach to Checking Knowledge Bases for Inconsistentcy and Redundancy. In *Proc. 3rd Annual Expert Systems in Government Conference*, pages 102–111, 1987.

[MC97] T.J. Menzies and P. Compton. Applications of abduction: Hypothesis testing of neuroendocrinological qualitative compartmental models. *Artificial Intelligence in Medicine*, 1997. To appear.

[Men96a] T.J. Menzies. On the practicality of abductive validation. In *ECAI '96*, 1996.

[Men96b] T.J. Menzies. Applications of abduction: Knowledge level modeling. *International Journal of Human Computer Studies*, 45:305–355, September, 1996.

[PS92] A.D. Preece and R. Shinghal. Verifying knowledge bases by anomaly detection: An experience report. In *ECAI '92*, 1992.

[Zla93] N. Zlatareva. Distributed verification and automated generation of test cases. In *IJCAI '93 workshop on Validation, Verification and Test of KBs Chambery, France*, pages 67–77, 1993.

Some of the Menzies papers can be found at *http:// www.cse.unsw.edu.au/~timm/pub/docs/papersonly.html.*