PROCEEDINGS OF THE INTERNATIONAL CONFERENCE
ON FIFTH GENERATION COMPUTER SYSTEMS 1992,
edited by ICOT.

CONSISTENCY-BASED AND ABDUCTIVE DIAGNOSES AS GENERALISED STABLE MODELS

Chris Preist, Kave Eshghi

Hewlett Packard Laboratories, Filton Road,

Bristol, BS12 6QZ, Great Britain

cwp@hplb.hpl.hp.com
ke@hplb.hpl.hp.com

Abstract

If realistic systems are to be successfully modelled and diagnosed using model-based techniques, a more expressive language than classical logic is required. In this paper, we present a definition of diagnosis which allows the use of a nonmonotonic construct, negation as failure, in the modelling language. This definition is based on the generalised stable model semantics of abduction.

Furthermore, we argue that, if negation as failure is permitted in the modelling language, the distinction between abductive and consistency-based diagnosis is no longer clear. Our definition allows both forms of diagnosis to be expressed in a single framework. It also allows a single inference procedure to perform abductive or consistency-based diagnoses, as appropriate.

1 Introduction

Many different definitions of diagnosis have been used in an attempt to formalise and automate the diagnosis process. In the so-called 'logical' approach, two frameworks, namely the *consistency-based* [Reiter 1987] and *abductive* [Cox and Pietrzykowski 1986], have attracted a lot of attention. Typically, the modelling language used in these frameworks is first order logic (or some subset of it). In this paper we present a unified framework for diagnosis which brings together these two styles of diagnosis, as well as providing a non-monotonic modelling language.

We were primarily motivated by the need to incorporate *negation as failure*, the non-monotonic construct in logic programming, into the modelling language. We first show the need for this construct through some examples, and then argue that the incorporation of negation as failure in the modelling language necessitates the inclusion of both consistency-based and abductive diagnosis within the same framework. We then present our unified framework, which allows negation as failure in the modelling language and naturally incorporates both abductive and consistency-based diagnosis. We then show that in the special cases, our approach reduces to pure consistency and pure abductive diagnosis, i.e. it is a generalisation of both styles.

Our work is similar in spirit to the work of Console and Torasso, [1990],[1991], but goes beyond it in many ways. We will compare our approach to that of Console and Torasso in a later section. Our proposed framework is based on the Generalised Stable Model semantics [Kakas and Mancarella 1990a] of generalised logic programs with abduction, strengthening the link between logic programming and diagnosis first explored in [Eshghi 1990].

2 Consistency-based and abductive approaches to diagnosis

In both consistency-based and abductive approaches, a set of axioms SD (called the *system description*) models the system under investigation, and a set of abnormality assumptions $Ab=\{ab_1,ab_2,...ab_n\}$ represents the possible underlying causes of failure. A set of statements, Obs, represents observations of the behaviour of the system which are to be explained.

In the consistency-based approach, a *diagnosis* is a set of abnormality assumptions, Δ, such that

(1) $SD \cup OBS \cup \Delta \cup \{\neg ab_k \mid ab_k \in Ab-\Delta\}$ is consistent.

The consistency-based approach focuses primarily on a model of the system's correct behaviour. When the abnormality assumptions relate to the failure of the components of the system, it attempts to find a set of normality and abnormality assumptions which can be assigned to the system's components to give a theory consistent with the observations.

In the abductive approach, a diagnosis is a set of abnormality assumptions, Δ, such that

(2) $SD \cup \Delta \vdash OBS$

$SD \cup \Delta$ is consistent.

The abductive approach primarily models the behaviour of a failing system, by using fault models in the system description, SD. The diagnosis process consists of look-

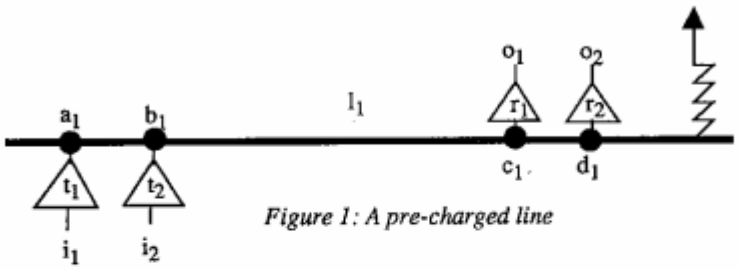

Figure 1: A pre-charged line

ing for a set of abnormality assumptions which, when adopted, will logically predict the observed faulty behaviour given the system description and the context of the observation.

In both approaches, a diagnosis Δ is defined to be *minimal* if there is no other diagnosis, Δ', which is a proper subset of Δ.

3 The Diagnosis Problem

The system description used in model-based diagnosis takes one of two forms. It is either a causal model, or a model consisting of the system's structure and the behaviour of individual components. In general, work on abductive diagnosis has focused on the former, while work on consistency-based diagnosis has focused on the latter.

For the purposes of this paper, we adopt a specification of a diagnosis problem based on those used in [deKleer and Williams 1987] and [Reiter 1987], which uses a component-based approach. However, the results hold equally for a causal model-based approach, and for this reason, we adopt slightly more general language in the definition.

Definition:

A diagnosis problem consists of a triple, <SD, OBS, C> where;

(i) The system description, SD, specifies the behaviour of the system.

(ii) The observation set, OBS, specifies a set of observations of the system as unit clauses.

(iii) C consists of constants, c_i, which represent *causal clusters* within the system.

Causal clusters are groups of causes of abnormal system behaviour which it makes sense to consider together. Each cause, n, within the cluster, c_i, is modelled in SD with two clauses;

$$effects\_of\_cause\_n \leftarrow ab(c_i, n).$$

$$ab(c_i) \leftarrow ab(c_i, n).$$

Furthermore, if so desired, we can define emergent properties of the system which occur when none of the causes in cluster c_i are present, the 'good behaviour model' of this cluster;

$$good\_behaviour\_model \leftarrow \text{not } ab(c_i).$$

In the component-based approach, c_i represents a component, and each cause in cluster c_i represents a possible fault model of the component. Note that the effects of a cause need not be defined deterministically. For example, the 'arbitrary behaviour' mode of a component, proposed in [deKleer and Williams 1989], is consistent with any behaviour of the component, but predicts nothing.

The logical language adopted to represent SD can vary with the definition of diagnosis adopted. In this paper, we focus on two possible languages; classical logic, as adopted by Reiter [1987], and horn clauses with negation as failure, as used in the logic programming community.

4 The need for negation as failure in the system description

The desire to integrate consistency-based and abductive diagnosis was motivated primarily by the need to include negation as failure in our models. The following two examples illustrate this need:

RAM modelling

In order to model the behaviour of a random access memory cell, we needed an axiom that says: the content of a cell at time T is X if X was written to this cell at time T', and no other write operation has been performed between T and T'. The most straightforward way of writing this is as the clause

```
contents(Cell, X, T)  ←  written(Cell, X, T'),
                         T'<T,
                         not over-written(Cell,T',T).

over-written(Cell,T',T) ←  written(Cell,X,T''),
                           T'<T''<T.
```

This is an instance of the 'frame-problem' being solved through negation-as-failure, as explored in [Shanahan 1989]. If we don't use negation as failure, or some other non-monotonic device, we need to have axioms which allow us to derive –over-written(Cell,T',T) for all cells and all time instants, which is very inefficient both in terms of speed of inference and storage required.

Pre-Charged Lines

A common technique used in the computer industry to implement data buses is the pre-charged line. Devices communicate with one another using transmitters and receivers, all connected to a common line whose value floats to 1 when no transmitter is transmitting. (There are n lines for an n-bit wide data bus. Here we concentrate on one line).

Physically, a value of 1 corresponds to high voltage, and a value of 0 to low voltage. In order to give the line its pre-charged value, it is connected to the positive power line by means of a pull-up resistor. Figure 1 gives a schematic of a typical pre-charged line.

To transmit a 0, a transmitter on a line pulls the line to low. Since lines are pre-charged, transmitting a 1 does not involve any action by the transmitter. (Obviously, there is a bus protocol to determine which transmitter, if any, is transmitting at any given time. Here we ignore protocol issues.)

The behaviour of pre-charged lines is best modelled by a default reasoning mechanism. The default value of a line is assumed to be 1 unless it can be proved to be 0. Using negation-as-failure, we could represent this as:

```
received_value(Line,0) ← driven_value(Line,0).
received_value(Line,1) ← not driven_value(Line,0).
driven_value(Line,0)   ← connected(Line,output(X)),
                          trasmits(X,0).
```

The alternative, avoiding the use of negation-as-failure, would be to have an axiom such as:

```
¬driven_value(Line,0) ←
  ∀X(connected(output(X),Line)→ ¬transmits(X,0)).
```

However, in order to prove ∀X(connected(output(X),-Line)→ ¬transmits(X,0)), we would need closure axioms exhaustively enumerating all the transmitters on the line, which would be both cumbersome to write and inefficient to reason with.

Full details of this modelling problem are given in [Eshghi and Preist 1992].

5 Negation As Failure blurs the distinction between abductive and consistency-based diagnosis

Conceptually, the processes behind abductive and consistency-based diagnoses are quite different. In consistency-based diagnosis, one removes normality assumptions until the theory regains consistency. In abductive diagnosis, one adds abnormality assumptions until the specified bad observations are provable in the theory.

However, by moving to a nonmonotonic theory, we can use the same process to perform both styles of diagnosis. We use negation as failure to represent the good behaviour of a cluster as its default behaviour;

```
behaviour ← not ab(c)
```

In a situation where the system is malfunctioning, and in the standard consistency-based approach we would derive an inconsistency by adding normality assumptions, we would get an inconsistency without adding any assumptions. This is because the negation as failure results in clusters defaulting to their 'good' behaviour model. Furthermore, the theory can be restored to consistency by adding abnormality assumptions, as in abduction, rather than by removing normality assumption as in the standard consistency-based approach.

It is exactly because of this effect that an abductive framework can be used to represent both consistency-based and abductive diagnoses. A similar approach to representing a component's good behaviour as its default behaviour was introduced in the context of the Nonmonotonic ATMS, in [Dressler 1990].

If we are to use negation as failure in the system description, as we argued we need to do in many instances, it is necessary to integrate abductive and consistency-based approaches. This is because, in a logic with negation as failure, consistency-based and abductive diagnoses are the dual of each other. By passing through a negation, you pass from a consistency-based problem to an abductive problem, or vice-versa. To see this, let us consider some simple examples;

a) Consistency-Based diagnosis

```
SD:  obs ← not g
     g ← ab(c)
OBS: ¬obs
```

In a consistency-based diagnosis, we attempt to restore consistency by making assumptions so as to *'not-prove'* a certain proposition which contradicts with the integrity constraints. In the case of the above example, we wish to not-prove obs. However, to do this, we must *prove* the negated goal, g. Hence we want an abductive diagnosis of the observation, g.

b) Abductive diagnosis

```
SD:  obs ← not g
     g ← ab(c)
OBS: obs
```

In an abductive diagnosis, we wish to make assumptions so as to *prove* a certain proposition which is required to be true by the integrity constraints. In the above example, we wish to prove obs. However, to do this, we must fail to prove the negated goal, g. Hence, we want a consistency-based diagnosis for the observation ¬g.

Thus a diagnostic problem of one sort may have a diagnostic problem of the other sort embedded in it. So, when the modelling language includes negation as failure, abductive and consistency-based diagnosis cannot

be considered in isolation from each other. It is this that led us to formulate this integration.

6 The Generalised Stable Model Semantics for Abduction

Various semantics have been proposed for abduction, both formally and informally. Originally, an abductive explanation for an observation was informally defined as a set of assumables which, when added to a theory, allowed proof of the observation. This was then formalised to give a metalevel definition of abduction in [Eshghi and Kowalski 1989].

Console *et al.* [1990] have used the completion semantics to give a semantics to abduction in horn clause theories. Recently, they have extended it to cover hierarchical logic programs [Console *et al.* 1991].

The semantics of abduction which we have chosen to use, however, is that provided by Kakas and Mancarella [1990a]. By extending the stable model semantics of logic programs [Gelfond and Lifschitz 1988], they give a semantics for abduction which holds for arbitrary general logic programs with integrity constraints.

Here, we briefly recall their definitions;

Definition 1

An abductive framework is a triple <P,A,IC> where

1) P is a set of clauses of the form $H \leftarrow L_1,..,L_k$ k0 where H is an atom and L_i is a literal.

2) A is a set of predicate symbols, the abducible predicates. The abducibles, Ab, are then all ground atoms with predicate symbols in A.

3) IC, the integrity constraints, is a set of closed formulae.

Hence an abductive framework extends a logic program to include integrity constraints and abducibles. The semantics of this framework is based on the stable model semantics for logic programs;

Definition 2

Let P be a logic program, and M a set of atoms from the Herbrand base. Define P_M to be the set of ground horn clauses formed by taking ground(P), in clausal form, and deleting;

(i) each clause that has a negative literal $\neg l$ in its body, and $l \in M$.

(ii) all negative literals $\neg l$ in the body of clauses, where $l \notin M$.

M is a *stable model* for P if M is the minimal model of P_M.

This definition is extended to give a semantics to abductive frameworks.

Definition 3

Let <P,A,IC> be an abductive framework, and $\Delta \subseteq$ atoms(A) be a set of abducibles. Then the set $M(\Delta)$ of ground atoms is a *generalised stable model* (GSM) for <P,A,IC> iff it is a stable model for the logic program $P \cup \Delta$, it is a model for the integrity constraints IC, and $\Delta = A \cap M(\Delta)$.

The above definition is an extension of that in [Kakas and Mancarella 1990a] to allow abducibles to appear in the head of a clause. As a result of this, the set of abducibles chosen as generators can be smaller than Δ, the set of abducibles true in the generalised stable model.

A unit clause, q, representing an observation, has an abductive explanation with hypothesis set Δ if there exists a generalised stable model, $M(\Delta)$, in which q is true.

Equivalently, we can say that q has an abductive explanation, Δ, within the abductive framework <P,A,IC> if the abductive framework <P,A,IC+q> has a generalised stable model $M(\Delta)$. Having q in the integrity constraints imposes the condition that q must be true in the generalised stable model, and hence must follow from the logic program together with the set of abducibles chosen.

7 Generalised Stable Models and Diagnosis

The generalised stable model semantics for abduction can be applied to diagnosis by mapping a diagnosis problem, <SD, OBS, C>, with multiple observations, onto an abductive framework as follows;

- Represent the system description, SD, as a logic program with integrity constraints, <P,IC>. The integrity constraints will usually contain sentences stating that observation points cannot take multiple values at a given time.
- Let the abducibles represent the causes within the clusters, $\{ab(c_i.n_y) \mid c_i \in C\}$, hence A = {ab(X,N)}.

Intuitively, given an observation set OBS, represented by a set of unit clauses, we have a choice of how to use it. We either wish to predict it, giving an abductive diagnosis, or make assumptions to restore the theory to consistency, giving a consistency-based diagnosis. By adding OBS to the integrity constraints, only models in which the observations are true, and hence explained by the system description together with selected abducibles, are legal generalised stable models. Hence we get an abductive diagnosis. If, instead, we add OBS to the logic program representing the system description, then a set of assumptions can only be made if they are consistent with the observations; i.e. the observations, system description and assumptions cannot derive anything which violates the integrity constraints. This will give us consistency-based diagnoses. Furthermore,

we can partition OBS into two sets, and predict some observations, OBS_p, while maintaining consistency with others, OBS_c. We do this by placing OBS_p in the integrity constraints, and OBS_c in the logic program.

This allows us to give a definition of unified diagnosis as follows;

Definition 4

Let <SD,OBS_p,OBS_c,C> be a diagnosis problem, where;

SD is a logic program with integrity constraints, <P,IC>.

OBS_p is the set of observations to be predicted by diagnoses.

OBS_c is the set of observations which diagnoses need to be consistent with.

C is the set of causal clusters in the system.

Then;

Δ is a GSM-diagnosis of <SD,OBS_p,OBS_c,C> iff there is a generalised stable model, M(Δ), of the abductive framework <P∪OBS_c,A,IC∪OBS_p>.

where A = {ab(C,N)} represents the set of possible root causes of misbehaviour in SD.

To demonstrate this, we consider a simple example from the medical domain, that of *pericardial tamponade*. The heart consists of two parts, the *myocardium* is the muscle which beats, while the *pericardium* is the protective sac which surrounds this muscle. If this sac is pierced, instantaneous pain occurs, which can subside fairly quickly. However, blood slowly flows into the pericardium over a period of time, increasing the pressure on the myocardium. Later, the myocardium will become so compressed that blood does not flow round the arteries, even though the myocardium itself is functioning perfectly.

The model of this phenomenon is given below. For simplicity, we treat time discretely, in units of hours.

```
pulse_ok(T)    ←   normal_cardiac_contraction(T),
                   not heart_compressed(T).

no_pulse(T)    ←   heart_compressed(T).

heart_compressed(T) ← ab(pericardium,pierced(T')),
                      T' < T - 10.
normal_cardiac_contraction(T) ←
                   not ab(myocardium,failure(T')),
                   T'<T.

bad_ecg(T)     ←   ab(myocardium, failure(T)).
```

We give the pericardium the possible failure cause 'pierced' at a given time, while the myocardium simply suffers a 'failure' of some sort. The latter is consistent with any behaviour of the myocardium, but only predicts a bad ecg trace.

The above clauses form the logic program part of SD.In addition, we need the integrity constraints, IC. These simply state which observations conflict with each other;

```
¬(pulse_ok(T) & no_pulse(T)).
¬(ecg_bad(T) & ecg_good(T)).
```

Assume we have the observation, no_pulse(12). Let us consider the generalised stable models of <P,A,IC>.

If we place the observation in the logic program as a unit clause, any set of abducibles can be assumed as long as they do not violate the integrity constraints - i.e. they must not generate a stable model in which pulse_ok(12) is true. If we assume nothing, the resulting stable model contains pulse_ok(12) as true, resulting in a conflict. There are two possible (minimal) ways to restore consistency. We can assume ab(myocardium,failure(10))[1], and cease to contain normal_cardiac_contraction(12) in the stable model. Alternatively, we assume ab(pericardium,pierced(2))[1], which predicts heart compression at time 12. The resulting stable model will therefore not contain pulse_ok(12), and so be a legitimate generalised stable model of <P∪{no_pulse(12)},A,IC>.

If, instead, we place the observation in the integrity constraints, IC, we are restricted to stable models which contain no_pulse(12). In this case, only by assuming ab(pericardium,pierced(2)) do we generate a stable model which contains no_pulse(12). As this also satisfies IC, it is a legitimate GSM for <P,A,IC∪{no_pulse(12)}>.

Hence, by making a choice of where to place the observation, we can generate either consistency-based or abductive diagnoses. Furthermore, if we have a second observation, ecg_good(12), we can choose to treat it in a different way from the first. Let OBS_p = {no_pulse(12)} and OBS_c = {ecg_good(12)}. In this case, the only (minimal) GSM of <P∪OBS_c,A,IC∪OBS_p> is that generated by ab(pericardium, pierced(2)). However, if we swap OBS_p and OBS_c, the only (minimal) GSM is that generated by ab(myocardium, failure(10)).

Note how the model uses negation-as-failure to handle the frame problem. If we used classical negation instead, it would be necessary to have extra clauses to predict not_heart_compressed at all relevant times, resulting in a larger, less understandable, and less efficient model.

8 Abductive and consistency-based diagnosis as special cases

If we restrict our attention to the traditional definitions of diagnosis, we can show that our definition is equivalent to these under certain conditions.

[1] Or, of course, at any other appropriate time instant.

8.1 Abductive Diagnoses as Generalised Stable Models

If all the observations are to be predicted in the abductive sense, and the system description contains only horn clauses, our definition of diagnosis reduces to the standard definition of abduction given in section 1. This is achieved as follows:

Given an abductive diagnosis problem <SD,OBS_p,C>, where SD is a horn-clause theory, divide the system description into a set of definite clauses, P, and a set of denials, D. Let A be the set of abducibles.

It is easy to show that abductive diagnoses of SD according to formula (2) correspond to generalised stable models of the framework <P,A,IC∪OBS_p>.

8.2 Consistency-Based Diagnoses as Generalised Stable Models

For a certain class of theories, namely *almost-horn* theories, we show that our definition of diagnosis is equivalent to the traditional definition of consistency-based diagnosis given in [Reiter 1987]. An almost-horn theory is a theory in which negation is used only to represent the negation of certain predicates. In the context of our theorem, these correspond to the abnormality assumptions.

Definition 5

A clause is said to be *almost-Horn with respect to A*, if , when in disjunctive normal form, it contains at most one positive literal with a predicate symbol not in A.

Theorem

Let <SD, OBS_c,C> be a consistency-based diagnosis problem, with SD a theory which is almost-horn with respect to A={ab}.

Then define the logic program with integrity constraints, SD'=<P,IC>, as follows;

Let $a_i \in$ atoms(A), and p, $q_i \notin$ atoms(A).

1. For every clause of the form
$p \leftarrow \neg a_1, \neg a_2 ... \neg a_k, a_{k+1}, .., a_m, q_1, q_2, .., q_n$ in SD, there is a program clause
$p \leftarrow$ not a_1,not a_2...not $a_k, a_{k+1}, .., a_m, q1, q_2, .., q_n$ in P.

2. For every clause of the form
$a_1 \vee a_2 ... \vee a_k \vee \neg a_{k+1} \vee ... \vee \neg a_m \neg q1 \vee \neg q_2 \vee .. \vee \neg q_n$ in SD there is an identical clause in IC.

Then;
D is a consistency-based diagnosis of <SD, OBS_c,C> according to formula (1)
⇔ D is a GSM-diagnosis of <SD', ∅, OBS_c,C>

The proof of this theorem is available in an extended version of this paper, available from the authors.

This theorem shows that, if negation is used only to represent the normality assumptions in the system, ¬ab, then the nonmonotonic definition of diagnosis given by us is equivalent to the monotonic definition given in [Reiter 1987]. However, if negation is used elsewhere in the theory, the two definitions diverge. The classical consistency-based definition requires explicit representation of all negative information. The GSM-diagnosis, however, will make the closed-world assumption, and assume information is false unless it can be proved otherwise.

9 Comparison with Console & Torasso [2]

Console & Torasso have defined a framework for a general abduction problem. This framework allows a spectrum of diagnosis styles to be represented within it, including the pure consistency-based and abductive styles described above.

They divide the observations into two sets. One set, OBS_a, is to be explained by the assumptions, while the other set, OBS_c, must be consistent with the assumptions. They then define two sets;

$\Psi^+ = OBS_a$.

$\Psi^- = \{ \neg f(x) \mid f(y) \in OBS_c, x \neq y \}$

A diagnosis is then a set of abducibles which, when added to the theory, allows prediction of all observations in Ψ^+, and is consistent with the negative literals in Ψ^-.

Our definition is more powerful in several ways.

- It extends the definition of Console and Torasso from horn-clause theories to general logic programs with integrity constraints. This gives a sophisticated and expressive language for modelling, which includes negation as failure.
- The inclusion of the consistency-based observations in the object level, rather than their negations in the integrity constraints, means that these can be used easily during inference. This can reduce the time to find a conflict, by using 'backwards simulation' of components. In some cases, such as the example documented in [van Soest *et al.* 1990] , certain diagnoses cannot be found without access to the observations in this way.
- Within this framework, it is possible to define minimal diagnoses model-theoretically. We will expand on this in section 10.

Placing the consistency-based observations at the object level potentially gives us more efficient inference. However, to do this in the context of joint diagnoses can lead to problems.

It may be possible to conclude that an abductive obser-

vation is true, based on the adding of a consistency-based observation to the theory alone;

SD: obs1 → obs2

OBS_a: obs2

OBS_c: obs1

By adding obs1 to the system description, we can conclude that obs2 is true. Whether this is legitimate depends on how we interpret the consistency-based observations. If we consider them true, but not necessarily explainable, then this is legitimate. This is the case in Reiter's formalisation of diagnosis, and also in the case of the setting factors of Reggia *et al.* [1983]. However, if we consider them not necessarily true, merely not false, then this is unacceptable. In such circumstances, it is necessary to restrict the model so that consistency-based observations do not appear in the body of clauses, or use the approach proposed by Console and Torasso.

10 Minimality

We now focus attention on component-based diagnosis, and consider the problem of minimal diagnoses. We wish to restrict our attention to those diagnoses which contain a minimal number of failing components.

To do this, we introduce minimal generalised stable models;

Definition:

A general stable model, $M(\Delta)$, for an abductive framework,<P,A,IC>, is *minimal* if there is no other GSM, $M(\Delta')$, such that $\Delta' \subset \Delta$.

Hence, a minimal general stable model contains a minimal set of assumptions which allow the consequences of the logic program P to satisfy the integrity constraints, IC. Note that, because abductive frameworks are non-monotonic, this does not imply that any superset of Δ, Φ, will have a GSM, $M(\Phi)$.

If, in our diagnosis framework, we have a 1-1 correspondence between a hypothesised failed component and an abducible being assumed in the abductive framework, then minimal general stable models will correspond to minimal diagnoses. To do this, we must impose two restrictions on the relationship between the frameworks;

(i) There must be no abducible representing the correct behaviour of a component. This must instead be a default behaviour which is used in the absence of abducibles referring to the faulty behaviour of a component.

(ii) It must be illegal to make more than one assumption about a component's behaviour at a time.

Note that the second condition does not force fault modes to be mutually exclusive in real-life, merely that they must be mutually exclusive logically. This can easily be achieved by adding an integrity constraint forbidding a component to have two modes;

false ← $ab(c_i,m_{j1})$, $ab(c_i,m_{j2})$, $m_{j1} \neq m_{j2}$.

The framework provided by Console and Torasso satisfies the second of these conditions, but not the first. Because they work in a monotonic framework, it is not possible to represent the correct behaviour of a component as the default behaviour; instead, it must be explicitly assumed that a component behaves correctly.

As a result of this, they must specify a semantic minimisation criterion; a diagnosis is minimal if it contains a minimal set of abducibles corresponding to faulty behaviour. We, however, can specify a model theoretic criterion;

A diagnosis, Δ, is *minimal* if its corresponding GSM, $M(\Delta)$, is a minimal GSM.

11 Calculating Diagnoses

By providing a uniform model-theoretic framework for consistency-based, abductive and joint diagnoses, we have also provided a method for a uniform implementation. We simply need an algorithm for generating the minimal generalised stable models of an abductive framework, and we can use this for performing a variety of diagnosis tasks.

Much work has been carried out on the generation of stable models, and several efficient algorithms exist. However, as general stable models are a newer innovation, these results have yet to be fully exploited and extended to the GSM case. Currently, the state of the art in GSM generation is provided by Satoh and Iwayama [1991]. This work, however, has the drawback that it does not produce minimal GSMs.

Traditionally, in the abductive community, top-down algorithms have been used which tend to generate minimal solutions, as they avoid making irrelevant assumptions. (e.g. [Cox and Pietrzykowski 1986] [Kakas and Mancarella 1990b]) However, non-minimal abductive diagnoses are still acceptable in the model-theoretic semantics, and can be generated by the algorithms. Similarly, in the diagnosis community, generation of minimal diagnoses has tended to be a consequence of the algorithm selected (e.g. the ATMS in [deKleer and Williams 1987]) rather than a model-theoretic restriction.

However, Eshghi [1990] proposes an alternative approach. He generates a theory in which minimal diagnoses correspond exactly to the stable models of the theory. This means that non-minimal diagnoses are excluded by the semantics, rather than the algorithm. By extending these results beyond the almost-horn case, we are able to transform an abductive framework into a

logic program. The stable models of this logic program correspond exactly to the minimal generalised stable models of the abductive framework. This means that minimality is brought into the theory as a necessary property of each solution, rather than being a selection criterion between solutions. This work is currently in progress.

As a result of this, a wider variety of literature can be used to select appropriate and efficient algorithms, rather than being restricted to algorithms which have been developed specifically for the task of diagnosis.

12 Conclusions

By moving to a nonmonotonic logical framework, it is possible to bring abductive and consistency-based diagnosis together, and use the same inference method to perform both. We have done this by using generalised stable models to provide the semantics, which provides us with a rich and expressive modelling language. It also gives a link between diagnosis and logic programming, allowing application of theoretical and practical logic programming results to the domain of diagnosis.

Acnowledgements

Thanks to Bruno Bertolino and Enrico Coiera for their assistance.

References

[Console *et al.* 1990] L.Console, D. Theseider Dupre & P.Torasso. *A Completion Semantics for Object-level Abduction.* Proc. AAAI Symposium in Automated Abduction, 1990.

[Console *et al.* 1991] L.Console, D. Theseider Dupre & P.Torasso. *On the relationship between abduction and deduction.* Journal of Logic and Computation, 2(5), Sept. 1991.

[Console and Torasso 1990] L.Console & P.Torasso. *Integrating Models of the Correct Behaviour into Abductive Diagnosis.* Proceedings of the 9th European Conference on Artificial Intelligence, 1990.

[Console and Torasso 1991] L.Console & P.Torasso. *A Spectrum of Logical Definitions of Model-Based Diagnosis.* University of Torino Technical Report, 1991.

[Cox and Pietrzykowski 1986] P.T. Cox & T. Pietrzykowski. *Causes for Events: their Computation and Application.* Proc. 8th conference on Computer Aided Design and Engineering, 1986.

[Davis 1984] R. Davis. *Diagnostic Reasoning based on Structure and Behaviour.* Artificial Intelligence 24:347-410, 1984.

[deKleer *et al.*1990] J. deKleer, A. Mackworth & R Reiter. *Characterizing Diagnoses.* Proceedings of the Eighth National US Conference on Artificial Intelligence, Boston 1990.

[deKleer and Williams 1987] J. deKleer & B. Williams. *Diagnosing Multiple Faults.* Artificial Intelligence 32:97-130, 1987.

[deKleer and Williams 1989] J. deKleer & B. Williams. *Diagnosis with Behavioural Modes.* Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, Detroit 1989.

[Dressler 1990] O.Dressler. *Computing Diagnoses as Coherent Assumption Sets.* Proceedings of the First International Workshop on Principles of Diagnosis, Menlo Park 1990

[Eshghi 1990] K. Eshghi. *Diagnoses as Stable Models.* Proceedings of the First International Workshop on Principles of Diagnosis, Menlo Park 1990

[Eshghi and Kowalski 1989] K. Eshghi & R. Kowalski. *Abduction compared with Negation as Failure.* Proceedings of the 6th Int. Conf. on Logic Programming, Lisbon 1989, pp234-254.

[Eshghi and Preist 1992] K. Eshghi and C. Preist. *The Cachebus Experiment: Model Based Diagnosis applied to a Real Problem* in Industrial Applications of Knowledge-Based Diagnosis, ed Guida and Stefanini, Elsevier 1992.

[Gelfond and Lifshitz 1988] M. Gelfond & V. Lifshitz. *The Stable Model Semantics for Logic Programming.* Proceedings of the Fifth International Conference on Logic Programming, 1988.

[Kakas and Mancarella 1990a] A. Kakas & P. Mancerella. *Generalised Stable Models: A Semantics for Abduction.* Proceedings of the 9th European Conference on Artificial Intelligence, 1990.

[Kakas and Mancarella 1990b] A. Kakas & P. Mancarella. *On the relation between Truth Maintenance and Abduction.* Proceedings of PRICAI, 1990.

[Reiter 1987] R. Reiter. *A theory of diagnosis from first principles,* Artificial Intelligence Journal 32, 1987

[Reggia *et al.* 1983] J.A. Reggia, D.S. Nau & P.Y. Wang. *Diagnostic Expert Systems based on a Set Covering Model.* Int. J. of Man-Machine Studies 19, p437-460. (1983)

[Satoh and Iwayama 1991] K. Satoh & N. Iwayama. *Computing Abduction by using the TMS.* Proceedings of the Eighth International Conference on Logic Programming, 1991.

[Shanahan 1989] M. Shanahan. *Prediction is Deduction but Explanation is Abduction.* Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, Detroit 1989.

[vanSoest *et al.* 1990] D.C. van Soest, R.R. Bakker, F. van Raalte & N.J.I. Mars. *Improving effectiveness of model-based diagnosis,* Proc. 10th international workshop on expert systems and their applications, Avignon 1990.