Finding Faults with Uncertain Assumptions

Jeroen Keppens* and **Qiang Shen**
Department of Artificial Intelligence
University of Edinburgh
Edinburgh EH1 1HN, UK
jeroen.keppens@econ.kuleuven.ac.be; qiangs@dai.ed.ac.uk

Abstract

The General Diagnostic Engine (GDE) postulates plausible faults from given measurements by making extensive use of an Assumption based Truth Maintenance System (ATMS). However, it relies upon an entropy-based uncertainty calculus, separate from the ATMS, to compute optimal locations for measurements gathering. The practical usability of GDE is thus restricted by the statistical information required and by the strong assumptions necessary for making subsequent simplifications. In this paper, a simple numeric certainty calculus is integrated with the ATMS. The integrated ATMS, capable of handling uncertain assumptions, is employed as the unified basis upon which to guide the measurements gathering process of the dependent diagnostic engine. This enables finding faults in uncertain environments, while reducing restrictive assumptions on component failures.

Introduction

The General Diagnostic Engine (GDE) (de Kleer, J. & Williams, B.C. 1987) is a typical system for model-based multiple fault diagnosis of real-world artefacts. It makes extensive use of an Assumption-based Truth Maintenance System (ATMS) (de Kleer 1986) to generate a minimal set of fault hypotheses. Additionally, GDE employs a measurement proposer to choose locations in the artefact being diagnosed to acquire further information, in order to discriminate between the generated fault hypotheses. As the original version of the ATMS is not able to cope with varying degrees of certainty of the assumptions, a separate numeric uncertainty calculus based on the entropy theory is used to construct this measurement proposer. However, this uncertainty calculus limits the practical applicability of the GDE because, in its general version, it requires a good amount of statistical information and its simplifications are relied upon strong assumptions such as faults occurring independently and with an equal small probability.

An ATMS enables a problem solver, such as a diagnostic engine to keep track of the assumptions underlying the inferences it makes and to explore the consequences of changes in these assumptions. In the standard ATMS, assumptions are deemed to be either true or false. Yet, integrating a numeric uncertainty handling mechanism into the ATMS can significantly extend its capabilities. For example, sets of assumptions supporting inferences can then be ranked, thereby providing a means of grading the fault hypotheses in a diagnostic system. Several alternative versions of such an extended ATMS have been suggested, but some of them are unsuitable for GDE-style diagnosis, whilst the others involve computationally costly calculations.

In this paper, a basic certainty factor calculus (Buchanan, B. & Shortliffe, E.H. 1984) is incorporated into the ATMS, resulting in a simple certainty factor based ATMS, named CF-ATMS. An alternative measurement proposer is then suggested that directly uses the numeric information provided by the CF-ATMS for measurements gathering. The paper is arranged as follows. After briefly describing the notations of the standard ATMS upon which the present extension is based, the next section presents the formulisation of the CF-ATMS. In the third section, the role of the ATMS and that of the measurement proposer in the original GDE are given and, then, a novel measurement proposer that utilises the numeric certainty information provided by the CF-ATMS is presented. The fourth section illustrates how this measurement proposer can help a GDE-style system in finding faults without involving complex calculations. Conclusions of the work are provided in the final section.

A Certainty Factor Based ATMS

Assumption-based TMS

The task of the ATMS is to maintain a database of justifications for a problem solver, say a constraint propagator in a GDE-style diagnostic engine. To this end, it stores each datum x in which the problem solver is interested as a node n_x. The actual problem-solver datum may be utilised to identify the node, but is of no interest to the ATMS. Certain nodes are registered as assumptions, representing the primitive data, upon which the problem solver makes inferences.

*J. Keppens is now with the Katholieke Universiteit Leuven, Naamsestraat 69, 3000 Leuven, Belgium

The problem solver incrementally presents the ATMS with justifications. A justification is a material implication in which the antecedent is a conjunction of non-negated nodes and the consequent is a non-negated node: $n_1 \wedge n_2 \wedge ... \wedge n_m \rightarrow n_c$

The problem solver also informs the ATMS of the sets of nodes which cannot be true at the same time. Such sets are called no-goods. A no-good represents an inconsistent conjunction of nodes, a conjunction of nodes from which false can be derived. For this reason, it is stored as a justification of the specific node $n_\perp$ which represents false.

An environment E is a set of conjunctively joint assumptions $\{a_1, a_2, ..., a_k\}$. A node n is said to hold in an environment E if its truth can be derived from E given the set of justification $\mathcal{J}$ stored in the ATMS. We write: $E, \mathcal{J} \vdash n$

An environment is said to be inconsistent if false can be derived from it. Consequently, within the ATMS, an environment E is said to be inconsistent if: $E, \mathcal{J} \vdash n_\perp$

For each node n, the ATMS computes a label $\mathcal{L}(n)$. A label $\mathcal{L}(n)$ is a set of (disjunctively joint) environments $\{E_1, E_2, ..., E_l\}$ which has the following properties:

- A label is consistent: $\forall E_i \in \mathcal{L}(n) : E_i, \mathcal{J} \nvdash n_\perp$
- A label is sound: $\forall E_i \in \mathcal{L}(n) : E_i, \mathcal{J} \vdash n$
- A label is complete: $\forall E', \exists E_i \in \mathcal{L}(n) : E', \mathcal{J} \vdash n \Rightarrow E_i \subseteq E'$
- A label is minimal: $\forall E_i, E_j \in \mathcal{L}(n) : E_i \neq E_j \Rightarrow E_i \nsubseteq E_j$

Requirements of an uncertainty calculus for fault diagnosis

The extended ATMS proposed in this paper is to be utilised in conjunction with a GDE-style problem solver. While this problem solver propagates the available data on an artefact through the modelled constraints, the ATMS maintains a number of plausible states of each connection in the artefact and the environments under which these are true. Given the involvement of uncertain assumptions, the extension must provide a method for numeric evaluation of the certainties of those environments.

Any certainty calculus that is employed for such a purpose should possess a number of desirable properties. First, it should not increase the complexity of what is already a computationally demanding algorithm. Second, an certainty factor of a node (label) should not only depend upon the certainties of the individual assumptions that support (are contained in) it, it should also depend upon the number of composite assumptions. This requirement is justified as long as the system is supposed to model a significant degree of independence between component failures, as is commonly assumed in GDE style systems. This latter property can be formalised as follows, under the assumption that a certainty factor expresses a degree of truth (an assumption from which later discussions will deviate):

- A numeric evaluation (γ) of the degree of truth of a conjunction should tend to decrease with the number of conjuncts involved, except when dealing with absolute certainties:

$$\begin{array}{l}\forall a_1, a_2, ..., a_n, a_{n+1}, \\ \gamma(a_1 \wedge a_2 \wedge ... \wedge a_n) > 0 \wedge \gamma(a_{n+1}) < 1 \\ \Downarrow \\ \gamma(a_1 \wedge a_2 \wedge ... \wedge a_n) > \\ \gamma(a_1 \wedge a_2 \wedge ... \wedge a_n \wedge a_{n+1})\end{array} \quad (1)$$

- Similar to the previous issue, a numeric evaluation (γ) of the degree of truth of a disjunction should tend to increase with the number of disjuncts involved, except when dealing with absolute certainties:

$$\begin{array}{l}\forall a_1, a_2, ..., a_n, a_{n+1}, \\ \gamma(a_1 \vee a_2 \vee ... \vee a_n) < 1 \wedge \gamma(a_{n+1}) > 0 \\ \Downarrow \\ \gamma(a_1 \vee a_2 \vee ... \vee a_n) < \\ \gamma(a_1 \vee a_2 \vee ... \vee a_n \vee a_{n+1})\end{array} \quad (2)$$

Several versions of an ATMS with an integrated numeric uncertainty calculus have already been devised. Dubois, Lang and Prade (Dubois, H., Lang, J., & Prade, H. 1990) have redefined the ATMS within the framework of their possibilistic logic. Bernasconi, Rivoira and Termini have outlined a similar definition (Bernasconi, C., Rivoira, S., & Termini, S. 1990). Both approaches use the conventional min/max operators in fuzzy logic for the interpretation of conjunctions and disjunctions. The problem with these operators is that they model a strong dependence between component failures because the certainty value of a conjunction (disjunction) is not decreased (increased) for the possibility that one of the conjuncts (disjuncts) is true while the others are not.

Other approaches are those of Laskey and Lehner (Laskey, K.B. & Lehner, P.E. 1988), Pearl (Pearl 1988) and Provan (Provan 1988) who have independently integrated the ATMS with the Dempster-Shafer theory of belief functions, and that of Srinivas (Srinivas 1994) who has extended the ATMS with Bayesian probability theory. Srinivas' probabilistic ATMS can be extended for modelling correlated certainty values (probabilities) between assumptions whereas the approaches based on the Dempster-Shafer theory assume independent assumptions. However, in the probabilistic ATMS complex Bayesian networks must be constructed for expensive probability propagation, and the calculation of beliefs in the Dempster-Shafer theory also demands a high computational cost.

Integration of certainty factor calculus with ATMS

The requirements of an uncertainty calculus for fault diagnosis as discussed above can be satisfied, however. To deal with uncertain assumptions, similar to all other hybrid ATMSs mentioned earlier, every assumption a within the CF-ATMS proposed herein is assigned a numeric certainty value: a certainty factor $CF(a) \in [0, 1]$. These certainty factors are then propagated to all other nodes and their labels (or sets of environments). As argued below, different from the other hybrid ATMSs, the present method for propagating certainty factors does not increase the order of complexity of the overall extended ATMS.

A justification-based approach to certainty factor propagation traces the chains of inferences by means of the set of justifications $\{J_{n_1}, J_{n_2}, ..., J_{n_p}\}$ stored with each node n. This set of justifications represents the disjunction of the justifications $J_{n_1} \vee J_{n_2} \vee ... \vee J_{n_p}$ and, therefore, the certainty factor of the associated node $CF(n_x)$ equals $op_\vee(CF(J_{n_1}), CF(J_{n_2}), ..., CF(J_{n_p}))$[1]. Each justification J_{n_i} represents a conjunction of nodes $n_1 \wedge n_2 \wedge ... \wedge n_q$. Consequently, each justification's certainty factor $CF(J_{n_i})$ equals $op_\wedge(CF(n_1), CF(n_2), ..., CF(n_q))$[2].

However, this algorithm must be improved in order a) to avoid double-counting separate but overlapping disjuncts, and b) to account for inconsistent subsets of justifications. This is because most instances of the operators $op_\vee$ and $op_\wedge$ other than max and min do not automatically check for these.

Logically, the justification-based approach traces back the labels of the nodes by means of the justifications. A label-based approach to certainty factor propagation, on the other hand, computes a node's certainty factor relying solely on the label maintained by the ATMS. When given a node n with as label $\mathcal{L}(n) = \{\{a_{ij} \mid j \in \{1, 2, ..., l_k\}\} \mid i \in \{1, 2, ..., k\}\}$, the label-based approach calculates its certainty factor as $CF(n) = op_\vee{}_{i=1}^{k}(op_\wedge{}_{j=1}^{l_k}(CF(a_{ij})))$. Because the application of the laws of distributivity for the logical 'and' and 'or' connectives is equivalent to that for the algebraic 'min' and 'max' operators, the justification-based and label-based approaches produce the same certainty factors for nodes, if $op_\vee$ and $op_\wedge$ are interpreted using the operators 'max' and 'min' respectively.

Unfortunately, as indicated earlier, the traditional min and max combination operators for logical expressions are not useful for the present purposes. The certainty factors computed with these operators discard a large amount of uncertainty information. In order to remedy this situation, other operators for $op_\wedge$ and $op_\vee$ should be used. The following alternative operators (which are commonly utilised in fuzzy logics (Smith, F.S. & Shen, Q. 1997)) have been experimented with:

- Conjunction operators:
  Algebraic product $op_\wedge(x, y) = xy$
  Bounded product $op_\wedge(x, y) = max(0, x + y - 1)$
  Drastic product $op_\wedge(x, y) = \begin{cases} x & y = 1 \\ y & x = 1 \\ 0 & x, y < 1 \end{cases}$
- Disjunction operators:
  Algebraic sum $op_\vee(x, y) = x + y - xy$
  Bounded sum $op_\vee(x, y) = min(1, x + y)$
  Drastic sum $op_\vee(x, y) = \begin{cases} x & y = 0 \\ y & x = 0 \\ 1 & x, y > 0 \end{cases}$

where $x, y \in [0, 1]$

It can easily be verified that the algebraic product and sum and the drastic product and sum always meet the requirements of 1 and 2. Bounded product meets these requirements for sufficiently high certainty values and bounded sum meets these requirements for sufficiently low values, because otherwise, the certainty values are always truncated to 0 and 1 respectively.

The disadvantage of these alternative certainty factor calculi, which employ a combination of the operators defined above, is that when they are utilised to compute the certainty factors of the ATMS nodes, label-based and justification-based computations are not guaranteed to produce the same results. Nevertheless, GDE requires a numeric uncertainty calculus for computing certainty factors attached to the candidates (straightforwardly derived from inconsistent environment sets) in order to discriminate between them. Consequently, the CF-ATMS implemented in the present work utilises the label-based computation method.

In contrast with the other numeric extensions of the ATMS, the CF-ATMS' label-based computation of certainty factors does not increase the computational complexity of the original ATMS implementation. An efficient ATMS implementation maintains a unique data structure for each environment (de Kleer 1986). This allows for the number of applications of the conjunction operator in computing certainty factors to be reduced to once per environment and per change of an assumption's certainty factor. Given that there are A assumptions in a CF-ATMS, there are 2^A environments and 2^A associated applications of the conjunction operator to a set of assumptions while initialising the ATMS. Only when the certainty factor of an assumption changes, must $2^{(A-1)}$ conjunctions be recalculated. However, in applications for supporting model-based diagnosis, the certainty factors of the assumptions given *a priori* do not change during the diagnostic process. Similarly, the corresponding disjunction operator needs to be applied only once per addition of a justification to a node. Each such addition of a justification implies that the disjunction operator is applied to a set of pre-calculated environment certainty factors in addition to the set-operations required for updating the label of that node.

Besides, the actual calculations necessary for certainty factor propagation are straightforward. This presents a main advantage of CF-ATMS in assisting the problem solver which makes use of it to minimise the

[1] $op_\vee$ is the operator utilised to compute the certainty factor of a disjunction. In the original certainty factor calculus $op_\vee = max$.

[2] $op_\wedge$ is the operator utilised to compute the certainty factor of a conjunction. In the original certainty factor calculus $op_\wedge = min$.

computational efforts involved. In the next section, it is shown how the certainty factors produced by this CF-ATMS are useful within the framework of a GDE-style diagnostic system (though the CF-ATMS may also be used in conjunction with other types of problem-solving tasks).

Fault Diagnosis with CF-ATMS

The General Diagnostic Engine

GDE is a system for model-based diagnosis of multiple faults. An artefact which is diagnosed by it is modelled by means of components and connections between these components. The description of a component formalises the relationship between the respective contents of the connections attached to the component. Within a diagnostic session the content or value of a connection must not change with respect to the same inputs to the artefact, as GDE is not designed to cope with time-varying behaviours.

In general, a diagnostic session in GDE consists of an iteration of steps. At each step, an additional value of a connection is measured. GDE's constraint propagator utilises this value to derive possible values for the unmeasured connections. The constraints are those formal relationships between connections stored in the description of the components, representing the model of the artefact. All possible candidate-explanations for observed faulty behaviour are then generated. Next, the measurement proposer calculates the likelihood of each candidate and computes which of all possible next measurements differentiates best between the available candidates.

At each iteration of a diagnostic session, candidate-explanations are easily computed with the help of an ATMS. The role of the ATMS in GDE consists of discriminating between the consistent and inconsistent sets of assumptions underlying the inferences made when propagating measurements through the modelled constraints. To this end, for each component in the model of the artefact, an ATMS-assumption is created which represents the presumption that the component behaves correctly with respect to its design intention. For each plausible value in a connection, an ATMS-node is maintained. Naturally, each couple of different values for the same connection corresponds with an ATMS-nogood. Additionally, every inference made through constraint propagation is passed on to the ATMS by means of justifications. This information is utilised by the ATMS to compute a consistent, sound, complete and minimal label for every node.

A label now represents which configurations of correctly functioning components suffice to explain the value of the associated node for the relevant connection. The label of the no-good node $n_\perp$ represents the sets of those components which cannot behave properly at the same time. These sets are called conflicts and are used to form candidates. A candidate is a set of components representing a hypothesis of each such component being faulty, which is sufficient to explain the observations in the artefact. Given that $\mathcal{L}(n_\perp) = \{E_{n_\perp 1}, E_{n_\perp 2}, ..., E_{n_\perp p}\}$ a complete set, CS, of candidates can be specified by:

$$CS = \{\{c_1, c_2, ..., c_p\} \mid c_1 \in E_{n_\perp 1}, c_2 \in E_{n_\perp 2}, ..., c_p \in E_{n_\perp p}\}$$

From this set only the minimal candidates are maintained as these are sufficient to completely specify all candidates. That is, $\forall C_i \in CS$, C_i is removed from CS if $\exists C_j \in CS, C_i \subseteq C_j$, where $i \neq j$.

GDE's measurement proposer suggests which connection to measure next in order to discriminate between fault candidates, while minimising the total number of measurements required. For this purpose, GDE utilises an entropy-based one-step look-ahead strategy. It chooses the connection x_i which will result in a minimised expected entropy $H_e(x_i)$. Given that the constraint propagator takes into account m_i values v_{i1}, v_{i2}, ..., v_{im_i} for a connection x_i, the expected entropy associated with that connection is computed by:

$$H_e(x_i) = \sum_{k=1}^{m_i} p(x_i = v_{ik}) H(x_i = v_{ik})$$

This method is critically dependent on the availability of failure probabilities for system components. Additionally, this method can be computationally very expensive if there are many connections with a substantial number of plausible values to consider. An alternative version of this method exists which simplifies the computation of $H_e(x_i)$, yet it assumes that all components fail independently with an equally small probability. Obviously, these assumptions are rather restrictive for many real applications.

The overall design of the following certainty factor based GDE-style diagnostic system is similar to that of the GDE described above. However, the two systems differ in terms of the numeric uncertainty calculus underlying the measurement proposer.

Additional information gained using CF-ATMS

Given that each assumption is assigned a certainty factor or degree of belief, the certainty factor computed for the label of a node expresses the certainty degree of which for at least one of the environments supporting that node all assumptions are true simultaneously.

For the purpose of diagnosis, certainty factors assigned to candidates must be computed. The degree of belief in a candidate equals that in which all components in the candidate are malfunctioning simultaneously. If each assumption A_i is assigned a degree of belief that the associated component is faulty $cf_i = CF(\sim a_i)$, then the certainty factor of a candidate can be calculated by means of the label-based computation as explained before: $CF(\text{candidate:}\{a_1, a_2, ..., a_m\}) =$

$CF(\sim a_1 \wedge \sim a_2 \wedge ... \wedge \sim a_m) = op_\wedge(cf_1, cf_2, ..., cf_m)$. This is why label-based calculus is used herein.

It is worth pointing out that a candidate's certainty factor would also be computable if degrees of belief in the truth of the assumptions were assigned. For example, if $op_\vee = max$ and $op_\wedge = min$ (however unsuitable these operators actually are), then $CF(\sim a_1 \wedge \sim a_2 \wedge ... \wedge \sim a_m) = CF(\sim (a_1 \vee a_2 \vee ... \vee a_m)) = -CF(a_1 \vee a_2 \vee ... \vee a_m)$. Nevertheless, as the present interest is in the determination of component failure, the certainty factor of a candidate should express belief in the falsehood of the associated assumptions of correctly functioning components.

Certainty factor based measurement proposer

As with GDE's measurement proposer, for each unmeasured connection (or variable), x_i, the diagnostic system's constraint propagator computes a set of feasible values $V_i = \{v_{i1}, v_{i2}, ..., v_{im_i}\}$ and the conditions (labels) under which they are true. V_i is made exhaustive by including an allowed value in it, representing all values which were not predicted by the constraint propagator. If one of these values v_{ij} is true, the union of those labels supporting all other values are conflicts:

$$x_i = v_{ij} \Leftrightarrow (\bigcup_{k=1,k\neq j}^{m_i} \mathcal{L}(x_i = v_{ik}) \vdash n_\perp)$$

Each of these potential conflict sets (one for each anticipated value of the variable) is then transformed into a set of minimal candidates. For each resulting candidate C_{ik} a certainty factor $CF(C_{ik})$ is calculated using the method previously described.

The question is now how to make an informed proposal for the location of the next measurement, such that the largest part of the search space of potential faults can be eliminated. If working with probabilities, entropy theory would solve the problem as GDE's measurement proposer does. Yet, it is imperative to recognise that probabilities are not used and that different decision rules apply.

A heuristic underlying the present approach is that the average of beliefs in found faults tends to increase if the certainty factors attached to the candidate sets that are associated with a given connection are all high. Also, the number of candidate sets eliminated by a single measurement should be in proportion to the number of possible measurement outcomes to consider. In this way, the measurement chosen will, by finding one value, eliminate many other values and their corresponding plausible candidates, thus reducing the search space to a substantial extent. The idea behind the entropy theory based method that is used by the GDE is in fact very similar.

The implementation of these heuristics consists of the use of a metric in proportion to the expected total belief of all the candidate sets which will be eliminated by taking a measurement of the associated connection or variable:

$$\frac{m_i - 1}{m_i} \sum_{k=1}^{m_i} CF(C_{ik}) \qquad (3)$$

That is, the best location for getting the next measurement is at the connection which maximises the above metric. However, occasionally, all plausible candidates may be associated with the outcome of one measurement at a certain connection, and it is possible that this outcome is the actual value contained in the connection. Obviously, measuring such a connection is not likely to refine the diagnosis effectively. Yet, the single high $CF(C_{ik})$ may be sufficient to cause this connection's total belief to get the highest rank. Therefore, the following weighted average should be used as the metric instead:

$$\frac{m_i - 1}{m_i} \sum_{k=1}^{m_i} w_{ik}.CF(C_{ik}) \qquad (4)$$

where w_{ik} is the weight attached to possible value k of connection or variable i.

Each weight appearing in expression (4) should be in proportion to the degree of certainty that the corresponding value is not to be measured. Since the method should work by estimating the total certainty factor mass that will be eliminated by measuring a variable, values that have a higher chance of being eliminated from consideration merit a higher weight.

Under the closed world assumption, a node is false if its label is false. Given that a node's label equals $\{\{a_{ij} \mid j \in \{1, 2, ..., t_i\}\} \mid i \in \{1, 2, ..., s\}\}$, then it is false if:

$$\sim (\vee_{i=1}^{s} \wedge_{j=1}^{t_i} a_{ij}) \vdash \top$$

or

$$\wedge_{i=1}^{s} \vee_{j=1}^{t_i} (\sim a_{ij}) \vdash \top$$

Therefore, the certainty factor for falsehood of a label can be computed by:

$$op_{\wedge}{}_{i=1}^{s} op_{\vee}{}_{j=1}^{t_i} CF(\sim a_{ij})$$

where $CF(\sim a_{ij})$ equals the certainty factor assigned to the negation of the assumption a_{ij}, i.e. the degree of certainty of the associated component being faulty.

Sample Runs

In order to test the CF-ATMS and a GDE-style system that incorporates the CF-ATMS in it, both have been implemented and applied to a number of representative problems. Here, the familiar example of Figure 1 is used to illustrate the functioning of the diagnostic system.

In this example, it is assumed that the certainty factor attached to the malfunctioning of each component is 0.1 (though different certainty factors can be assigned to different components). To illustrate the basic ideas,

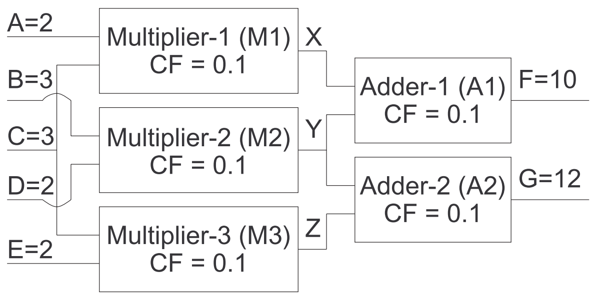

Figure 1: A familiar example

suppose that the algebraic product conjunction operator, the traditional 'maximum' disjunction operator, and the heuristic metric given in expression (3) are used to perform the diagnosis.

Given the available measurements for A, B, C, D, E, F and G, the constraint propagator computes all possible values for X, Y and Z and the ATMS maintains under which sets of assumptions each propagated value is true. At this stage, the content of the ATMS for the unmeasured variables looks as follows:

<$X = 4$, label: {{M2, A1}, {M3, A1, A2}}>
<$X = 6$, label: {{M1}}>
<$Y = 4$, label: {{M1, A1}}>
<$Y = 6$, label: {{M2}, {M3, A2}}>
<$Z = 6$, label: {{M3}, {M2, A2}}>
<$Z = 8$, label: {{M1, A1, A2}}>

For each possible value of each variable, the system generates the conflict set associated with that value and translates it into a set of minimal candidates. For variable X, for example, $X = 4$, $X = 6$ and $(X \neq 4) \wedge (X \neq 6)$ are the plausible values considered. The conflict sets for these potential outcomes are {{M1}} for $X = 4$, {{M2, A1}, {M3, A1, A2}} for $X = 6$ and {{M1}, {M2, A1}, {M3, A1, A2}} for $(X \neq 4) \wedge (X \neq 6)$. The minimal candidate sets for X's potential measurement outcomes are {{M1}} for $X = 4$, {{A1}, {M2, A2}, {M2, M3}} for $X = 6$ and {{M1, A1}, {M1, A2, A2}, {M1, M2, M3}} for $(X \neq 4) \wedge (X \neq 6)$. These minimal candidate sets have 0.1, 0.1 and 0.01 respectively as their corresponding certainty factors. These certainty factors are then combined into a single heuristic metric value as follows: $\frac{3-1}{3}(CF\{\{\mathtt{M1}\}\} + CF\{\{\mathtt{A1}\}, \{\mathtt{M2,A2}\}, \{\mathtt{M2,M3}\}\} + CF\{\{\mathtt{M1,A1}\}, \{\mathtt{M1,A2,A2}\}, \{\mathtt{M1,M2,M3}\}\}) = \frac{2}{3}(0.1 + 0.1 + 0.01) = 0.14$, where for instance $CF\{\{\mathtt{A1}\},\{\mathtt{M2,A2}\},\{\mathtt{M2,M3}\}\} = 0.1$ is calculated such that $CF\{\{\mathtt{A1}\},\{\mathtt{M2,A2}\},\{\mathtt{M2,M3}\}\} = max\{0.1, 0.1 \times 0.1, 0.1 \times 0.1\} = 0.1$ The metric values for Y and Z can be computed in the same way and these are 0.074 and 0.08 respectively. From this, the next measurement should be made at position X.

This result is interesting because the only single component faults that may fully explain the behaviour exhibited in the device of Figure 1 are Multiplier-1 or Adder-1 being faulty. These two components are connected at X, however. If one of these really fails, measuring X is the only way to distinguish between them. Measuring Y or Z would only enable us to differentiate between less certain multiple fault candidate explanations. Since variable X differentiates between the two single faults, the result has a very intuitive appeal in that single faults are usually more likely to occur than multiple faults. It is also important to notice that this result is the same as a sample run of the original GDE on the same device (de Kleer, J. & Williams, B.C. 1987) (where the assumption of components failing independently has to be made). Additionally, the computation effort involved in the present approach is very limited.

Finally, in order to show that the present proposal works even if device components fail with different rates, let us examine another diagnostic problem. The device under this consideration consists of a sequential construction of components as illustrated in Figure 2. Given the values in the first and last connections and different certainty factors attached to different components in the sequence, the diagnostic system biases its selection of measuring positions towards the components which are more likely to fail. This permits the system to reduce the total number of measurements required to locate a faulty component.

However, when the components which are more likely to fail appear to function correctly, the diagnostic system recovers from its bias and focusses its search towards the faulty sub-sequence of less error-prone components. Indeed, such an observation causes the candidates containing the more error-prone components to be eliminated from consideration. Consequently, the candidates causing the bias are no longer involved in the computation of a variable's heuristic metric, and this effectively removes the bias from the diagnostic system's future decisions.

Should the certainty factors become equal for all components, the system will equivalently perform a binary search for a faulty component. If the measurement in the middle of the (sub-)sequence under investigation differs from both ends, then the system continues its search in both halves of that (sub-)sequence separately in order to locate multiple faults.

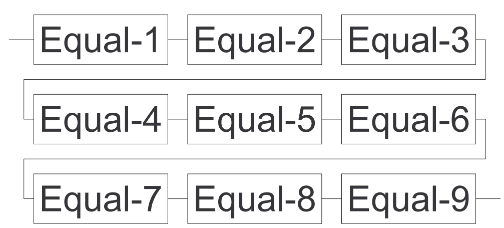

Figure 2: A sequencial device

Conclusion

This paper has presented an approach to incorporating a simple certainty factor calculus into the Assumption-based Truth Maintenance System for handling uncertain assumptions. Different operators useful for the interpretation of the required logic (conjunction and disjunction) connectives are provided. The resulting CF-ATMS has been employed as the dependency recording tool of a General Diagnostic Engine style system, and a measurement proposer has been proposed which utilises the certainty factor information produced by the CF-ATMS. The main advantage of this system is that it requires only weak assumptions on component failures. As demonstrated by experimental results, the new measurement proposer is able to effectively guide the diagnostic process in gathering evidence with simple computations, thereby offering the potential for complex artefacts to be diagnosed.

Acknowledgement: The authors are grateful to Elias Biris for his contribution, whilst taking full responsibility for the views expressed in this paper.

References

Bernasconi, C.; Rivoira, S.; and Termini, S. 1990. On the notion of uncertain belief revision systems. In *Extended abstracts of the 3rd International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems*, 458–460.

Buchanan, B., and Shortliffe, E.H., eds. 1984. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.* Reading, MA: Addison-Wesley.

de Kleer, J., and Williams, B.C. 1987. Diagnosing multiple faults. *Artificial Intelligence* 32:97–130.

de Kleer, J. 1986. An assumption-based TMS. *Artificial Intelligence* 28:127–162.

Dubois, H.; Lang, J.; and Prade, H. 1990. A possibilistic assumption-based truth maintenance system with uncertain justifications, and its application to belief revision. In *Proceedings of the ECAI Workshop on Truth Maintenance System*, 87–105.

Laskey, K.B., and Lehner, P.E. 1988. Belief maintenance; an integrated approach to uncertainty management. In *Proceedings of the National Conference on Artificial Intelligence*, 210–214.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan-Kaufmann.

Provan, G. 1988. Solving diagnostic problems using extended truth maintenance systems. In *Proceedings of the European Conference on Artificial Intelligence*, 547–552.

Smith, F.S., and Shen, Q. 1997. Choosing the right fuzzy logic controller. In *Proceedings of the 7th International Fuzzy Systems Association World Congress*, volume 3, 342–347.

Srinivas, S. 1994. A probabilistic atms. Technical Report KSL-94-13, Knowledge Systems Laboratory, Stanford University, Stanford, CA, USA.