Prioritized Defaults: Implementation by TMS and Application to Diagnosis

Ulrich Junker
GMD
P 0 Box 1240
5205 St. Augustin 1
Fed. Rep, of Germany
++49 - 2241/142671
junker®gmdzi.gmd.de

Abstract

We demonstrate the technological value of non-monotonic logics by an example: We use *prioritized defaults* for candidate generation in *diagnosis from first principles.* We implement this non-monotonic logic by *TMS* similar to default logic. Prioritized defaults allow an easy formulation of a diagnosis problem including statements such as *'eletrical parts are more reliable than mechanical ones'* or *'prefer correct models to fault models'* since defaults are put into different levels of reliability. These preferences prune some counterarguments in TMS and thus lead to a reduced network. Moreover, the labelings of this network are exactly the preferred subtheories of its prioritized default theory.

1 Introduction

Although the development of non-monotonic formalisms has been triggered by practical problems most work in this area has been devoted to the study of formal properties of non-monotonic inference. The technological value of non-monotonic formalisms is often ignored or questioned. We argue that non-monotonic logics can serve as *intermediate formalisms* in the development of applications for problems that require the generation and retraction of hypotheses. Good examples are diagnosis and configuration. Thus, we would decompose the design of an application into two steps:

1. Representation: First concentrate on the relevant representational decisions (e.g. what statements are encoded by hypotheses, defaults, priorities, counterarguments etc.) without being concerned with technical details. The result of this step should be a comprehensive and implementation-independent solution in terms of a non-monotonic formalism. Like a data-base scheme, it can be kept if the implementation is changed. This step also clarifies whether some properties of non-monotonic logics (e.g. multiple extensions) are really problems or can be useful for applications.
2. Implementation: Different techniques can be used for implementing a non-monotonic logic. If they are already available the main task is to select techniques which are efficient for the special case under consideration. Thus, we can benefit from a lot of techniques (e.g. truth maintenance, dependency-directed backtracking) without being in trouble with technical details. Non-monotonic logics serve as a specification for these techniques and show exactly how and in what circumstances they can be applied.

We illustrate this approach by developing a TMS-based system for generating candidates for diagnosis. There are already several (A)TMS-based diagnosis systems (e.g. [de Kleer and Williams, 1987], [Struss and Dressier, 1989], [Dressier and Struss, 1990]). These works contain a lot of innovations how to do diagnosis. Unfortunately, parts of their results are hidden in the code and expressed in system-specific terms like justifications and labels.

Therefore, we consider an intermediate logic to do diagnosis (similar as [Poole, 1989]). We choose prioritized default logic [Brewka, 1989] where defaults are put into different levels of reliability. This allows to express preferences such as *'adders are more reliable than multipliers'ox 'prefer correct models to fault models'.* We discuss this topic in sections 2.2 and 2.3.

In a second step, we provide a general implementation of level-based default theories using TMS. Since they are equivalent to the special case of prioritized circumscription implemented in [Baker and Ginsberg, 1989] we get an alternative method for handling prioritized circumscription. It does not need such notions as rebut and refute. We just map every level of the default theory into a subnetwork of a TMS using the translation in [Junker and Konolige, 1990].

To verify our translation, we need some results on components of TMS-networks. For this purpose, we split networks into independent components which can be labeled in isolation. Extensions of the complete network can be composed of extensions of the subnetworks. The precise results are presented in section 3,2. They are of their own interest because they enable a kind of divide-and-conquer strategy for computing extensions. Thus we have extended the work of Goodwin [Goodwin, 1987 who uses strongly connected components to guide the search in a particular labeling algorithm.

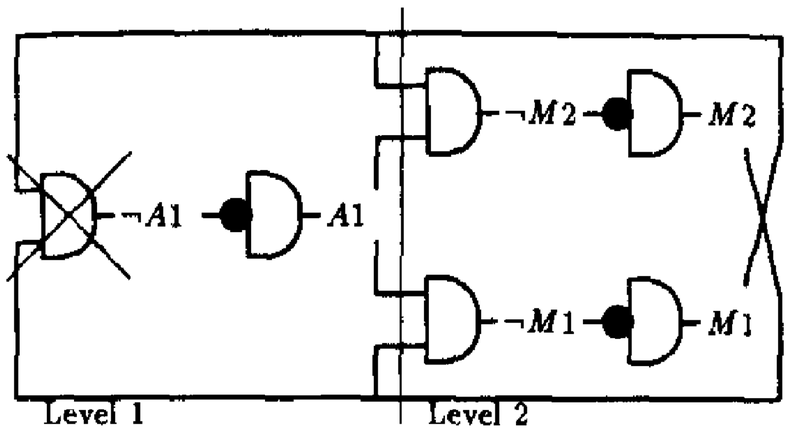

Figure 1: Removing Backpointing Justifications

By linking the subnetworks of every level we obtain a TMS-network that reflects the level-structure of the default theory. Its extensions can be split in the same way as the preferred subtheories of the level-based default theory. This network is also obtained if we first ignore the levels and translate the defaults according to [Junker and Konolige, 1990]. Then we remove all justifications leading from a higher level to a lower level (as illustrated in figure 1). It is difficult to imagine how a direct, ad-hoc approach to diagnosis could yield such a regular TMS-network.

2 From Diagnosis to Defaults

2.1 Prioritized Defaults

In this section, we introduce the non-monotonic formalism which will be used *to* solve the diagnosis problem. We consider prioritized defaults [Brewka, 1989] which have been introduced by Gerd Brewka to extend the simple formalism of David Poole [Poole, 1988]- Unlike to Poole's Theorist, defaults can there be ordered into different *levels* L_i of reliability. Defaults in L_i have higher priority than defaults in L_{i+k}.

To formulate this precisely, consider a first-order language $\mathcal{L}$ including a special unsatisfiable constant $\perp$. The first-order consequences of a subset X of $\mathcal{L}$ are defined as $Th(X) := \{q \in \mathcal{L} \mid X \models q\}$.

Defaults are represented by a name q (q is an atomic ground formula from $\mathcal{L}$) and a premise $q \supset \phi$ (where ϕ is an arbitrary formula of $\mathcal{L}$):

Definition 2.1 *A* level-based default theory $\Delta := (W, L)$ *consists of a set* $W \subseteq \mathcal{L}$ *of classical premises and a tupel* $L := (L_1, \ldots, L_k)$ *of disjoint levels* L_i *containing atomic ground formulas from* $\mathcal{L}$.

Preferred subtheories of level-based default theories are defined recursively. A preferred subtheory of level $i+1$ is obtained from a preferred subtheory of level i by adding elements of L_{i+1} as long as consistency with W is ensured:

Definition 2.2 *Let* $\Delta = (W, (L_1, \ldots, L_k))$ *be a level-based default theory.* T *is a preferred subtheory of level* i *iff*

1. $T = \emptyset$ *if* $i = 0$
2. T *is the union of a preferred subtheory* T' *of level* $i-1$ *and a maximal subset of* L_i *s.t.* $T \cup W \not\models \perp$ *if* $i \in \{1, \ldots, k\}$.

The preferred subtheories of Δ *are the preferred subtheories of level k.*

We obtain Poole's Theorist if we consider only a single level. Furthermore, if we already know a preferred subtheory T' of level i we could determine a subtheory *of* the next level by supplying Theorist with T' as a set of premises and L_{i+1} as a set of hypotheses. This view helps us to get a link to Reiter's default logic and from this to Doyle's TMS using the existing translation. Poole has shown that his simple hypotheses q correspond to prerequisite-free normal defaults $\frac{:q}{q}$ in Reiter's default logic. We define

$$D_L := \{\frac{:q}{q} \mid q \in L\} \qquad (1)$$

Then T is a preferred subtheory of level i+1 iff T consists of elements of levels and $Th(T \cup W)$ is an extension of a default theory (D^.^.IV U T') a la Reiter which is supplied with a preferred subtheory T^* of level i.

2.2 Diagnosis by Prioritized Defaults

In this section, we sketch how diagnosis problems can be formulated by level-based default theories. Tins includes statements for ranking different kinds of models (e.g. *'adders are more reliable than multipliers'or 'prefer correct models to fault models).*

In the sequel, let O be a set of objects or components. Their normal and faulty behaviour is described by several models[1] that exclude each other. The models of $o \in O$ are named by atomic formulas *rm{o).* For the sake of simplicity, we assume that every component has exactly k models. A first-order theory W is used to describe the following facts:

- definition of the models (i.e. the hehaviour of the single components)
- connections of pins or relationship between attributes of different components
- values of the pins/attributes (including supplied input values, as well as observed output values)
- exclusion of models ($\forall x. \neg m_i(x) \vee \neg m_j(x)$ for $i \neq j$)
- completeness of models ($\forall x. m_1(x) \vee \ldots \vee m_k(x)$)

We are interested in sets of hypotheses which explain the observed behaviour (i.e. the values of the output pins). A usual approach selects as many model as possible without violating consistency (models arc selected by adding their names to the theory above). This method treats every model in the same manner. However, we at least want to prefer normal models to fault models as in [Dressier and Struss, *1990].* A fault model should only be selected if a normal model is inconsistent with the current selection. Furthermore, it is useful to focus the diagnosis process to components that are more likely to fail. For example, cable connections often fail whereas TTL-circuits are reliable. We can easily express these preferences in a level-based default theory consisting of the first-order theory W and different levels ($L_1, \ldots, Z, i, £, -+!, \ldots, L_i+k-1$) containing the names of models:

[1] Models in the sense of model-based diagnosis.

- rank normal models of the components in O according to their reliability and distribute them among the levels $L_1, \ldots, L_i$.
- include the j-th fault model m, (o) of every component into L_{i+j-i}.

Thus, we have also ranked the different fault models. Other rankings could be used for other approaches to diagnosis. Each preferred subtheory of this level-based default theory corresponds to a diagnosis candidate as defined in [Reiter, 1987] and [de Kleer and Williams, 1987], To be more precisly, the negation of the correct models of the failing components are derived from the preferred subtheory. It is a maximal consistent subset of the set of models. Due to the preferences we do not obtain every candidate. Thus, diagnosis is focused to luss robust components. A similar effect is achieved in [de Kleer, 1990] by preferring diagnoses with less faults.

2.3 Formulating Davis' Familiar Example

We illustrate our approach by the well-known adder-multiplier circuit of Davis. It consists of two adders *A1*, *A2* and three multipliers M1, M2, M3:

adder(Al) multiplier(M1) multiplier(M3) (2), adder(A2) multiplied (M2) W

These components are connected as follows:

$$\begin{array}{lll} in1(M1) = A & in2(M1) = C & out(M1) = X \\ in1(M2) = B & in2(M2) = D & out(M2) = Y \\ in1(M3) = C & in2(M3) = E & out(M3) = Z \\ in1(A1) = X & in2(A1) = Y & out(A1) = F \\ in1(A2) = Y & in2(A2) = Z & out(A2) = G \end{array} \quad (3)$$

The correct models are described by a predicate ok and:

$$\begin{array}{l} \forall x.adder(x) \wedge ok(x) \supset in1(x) + in2(x) = out(x) \\ \forall x.multiplier(x) \wedge ok(x) \supset in1(x) * in2(x) = out(x) \end{array} \quad (4)$$

The supplied input and observed output values are:

$$\begin{array}{lllll} A = 2 & B = 2 & C = 3 & D = 3 & E = 2 \\ F = 10 & G = 12 & & & \end{array} \quad (5)$$

In this example, we don't consider fault models. Let W be a first-order theory including the formulas above and further axioms for reasoning with equality and arithmetics[2]. Now, we built up a level-based default theory $\Delta_{circ} = (W, L)$ where $L := (L_1, L_2)$ consists of two levels. As a heuristc, we assume that adders are more reliable than multipliers:

$$\begin{array}{l} L_1 := \{ok(A1), ok(A2)\} \\ L_2 := \{ok(M1), ok(M2), ok(M3)\} \end{array} \quad (6)$$

Δ_{circ} has two preferred subtheories, namely:

$$\begin{array}{l} T_1 := \{ok(A1), ok(A2), ok(M2), ok(M3)\} \\ T_1 := \{ok(A1), ok(A2), ok(M1)\} \end{array} \quad (7)$$

Thus, only two of the four diagnosis candidates in [de Kleer and Williams, 1987] are obtained, namely $\{\neg ok(M1)\}$ and $\{\neg ok(M2), \neg ok(M3)\}$. Thus, preferences have a similar effect as the fault probabilities in [de Kleer and Williams, 1987]: They focus the diagnosis process to candidates which are more likely to fail. Since we use a non-monotonic logic other candidates (e.g. $\{\neg ok(A1)\}$) are obtained if additional observations (e.g. $X = 6$ and $Y = 6$) are added. Other approaches to diagnosis including fault models [Struss and Dressler, 1989] could be realized by adding these models to higher levels which have a lower priority.

[2] In practical systems, we realize equality and arithmetics using attached calls of lisp functions and simple propagation rules as in Steele's constraint-system.

3 Results for Doyle's TMS

Before we are able to map level-based default theories to TMS, we need some properties of TMS.

3.1 Relevant Properties of Doyle's TMS

In [Junker and Konolige, 1990], we developed a compact description of Doyle's TMS that is based on closures of operators. For a detailed discussion, we refer the reader to this work. We cite only the prerequisites for the theorems below: An operator $apply : 2^{\mathcal{L}} \rightarrow 2^{\mathcal{L}}$ mapping subsets of $\mathcal{L}$ to subsets of $\mathcal{L}$ is monotonic and compact iff

$$\begin{array}{c} \text{if } X \subseteq Y \text{ then } apply(X) \subseteq apply(Y) \\ \text{if } q \in apply(Y) \text{ then } \exists X \subseteq Y : X \text{ is finite} \\ \text{and } q \in apply(X) \end{array} \quad (8)$$

From this, we obtain a *closure* operation $apply^*$ as follows: $apply^*(X)$ is the minimal superset of X that is closed w.r.t. $apply$. Sometimes, $apply$ depends only on a (relevant) part R of its input. In this case, it is sufficient to iterate it on the relevant part to get (the relevant part of) the closure. Hence, we take a restricted operator, which is expressed by the lambda-expression $(\lambda(X).apply(X) \cap R)$:

Lemma 3.1 *Let* $apply : 2^{\mathcal{L}} \mapsto 2^{\mathcal{L}}$ *be a monotonic and compact operator and* $R \subseteq \mathcal{L}$ *s.t.* $apply(X) = apply(X \cap R)$ *for all* $X \subseteq \mathcal{L}$. *Then* $apply^*(\emptyset) \cap R = (\lambda(X).apply(X) \cap R)^*(\emptyset)$.

We obtain particular instances of such operators if we consider justification networks (N, J) of Doyle's TMS. N is a set of (arbitrary) nodes q from $\mathcal{L}$ that are justified by non-monotonic justifications in J having the form $\langle in(I), out(O) \rightarrow q \rangle$. Their inlists I and outlists O are both finite subsets of N. We obtain a monotonic and compact operator for applying justifications if we use an extra-index to check the outlist:

$$\begin{array}{rl} apply_{J,Y}(X) := & \{c \mid \langle in(I), out(O) \rightarrow c \rangle \in J, \\ & I \subseteq X \text{ and } O \cap Y = \emptyset\} \end{array} \quad (9)$$

In contrast to the TMS-translation of unprioritized defaults, we supply justification networks with an input $X \subseteq N$, which is included in every TMS-extension:

Definition 3.1 *Let* $\nu = (N, J)$ *be a justification network and* $X \subseteq N$. T *is an* extension of ν and X *iff* $T = apply^*_{J,T}(X)$. *If* ν *has no extension it is* incoherent.

If $X = \emptyset$ we just say that T is an extension of ν. The extra input X does not make the computation of extensions more difficult. Its elements just get a fixed IN-label.

Now, we consider a set Q of nodes and denote their justifications by

$$J^Q := \{\langle in(I), out(O) \rightarrow c\rangle \in J \mid c \in Q\} \quad (10)$$

If we are only interested in the part of the result of $apply_{J,Y}$ that is in Q we apply only justifications in J^Q:

$$apply_{J,Y}(X) \cap Q = apply_{J^Q,Y}(X) \quad (11)$$

3.2 Clusters in TMS

In the sequel, we explore the structure of TMS-networks. More precisely, we decompose a justification network into components which can be labeled in isolation. These components must not depend on the remaining network. A part of a justification network depends on another part if they are linked by a justification:

Definition 3.2 *Let $\nu = (N, J)$ be a justification network, and C_1, C_2 be two subsets of N. C_2* depends on *C_1 if J^{C_2} contains a justification $\langle in(I), out(O) \rightarrow c\rangle$ s.t. $(I \cup O) \cap C_1 \neq \emptyset$.*

An *independent component consists of nodes* that have no justification depending on the rest of the network.

Definition 3.3 *Let $\nu = (N, J)$ be a justification network. Then $C \subseteq N$ is an* independent component *of ν iff C does not depend on $N - C$.*

In the sequel, we explore properties of components. Nodes of an independent component C can only be derived from nodes in this component.

$$\begin{aligned} apply_{J^C,Y}(X) &= apply_{J^C,Y}(X \cap C) \\ apply_{J^C,Y}(X) &= apply_{J^C,Y \cap C}(X) \end{aligned} \quad (12)$$

Due to lemma 3.1 and equation 11, these properties can be extended to the closure of $apply_{J,Y}$:

$$\begin{aligned} apply^*_{J,Y}(\emptyset) \cap C &= apply^*_{J^C,Y}(\emptyset) \\ apply^*_{J^C,Y}(\emptyset) &= apply^*_{J^C,Y \cap C}(\emptyset) \end{aligned} \quad (13)$$

This allows to state one important property of independent components: Every extension of the complete network can be split into an extension of the component and an extension of the remaining network and the first extension:

Theorem 3.2 *Let $\nu = (N, J)$ be a justification network, C be an independent component of ν. T is an extension of ν iff*

1. *$T \cap C$ is an extension of (C, J^C)*
2. *T is an extension of (N, J^{N-C}) and $T \cap C$.*

Hence, we get an extension of T if we first find an extension of an independent component and then extend it further. In general, there is no guarantee that we find such an extension of an extension because the remaining network may be incoherent (due to odd loops). For level-based default theories we will however obtain coherent subnetworks.

Networks which can be split into two independent components have an even nicer property: Their extensions can be split completely. Hence, we can ignore the extensions of (C, J^C) if we determine an extension of $(N - C, J^{N-C})$.

Theorem 3.3 *Let $\nu = (N, J)$ be a justification network that can be split into two independent components C and $N - C$. T is an extension of ν iff*

1. *$T \cap C$ is an extension of (C, J^C)*
2. *$T \cap (N - C)$ is an extension of $(N - C, J^{N-C})$*

Due to these results we can take a *divide-and-conquer*-approach for computing extensions. We can compute extensions of independent subnetworks first, record them and then proceed to get extensions of dependent components. Furthermore, we can combine extensions of different independent components without problems.

4 From Prioritized Defaults to TMS

In the next section, we consider the translation of a default theory à la Reiter to a TMS-network and modify it slightly to handle changing premise sets that are obtained as results of lower levels.

4.1 Translating Default Logic to TMS

In [Junker and Konolige, 1990] we extracted two sets of formulas from Reiter's defaults. Let $\Delta = (D, W')$ be a default theory. We consider the set of consequents of defaults and the set of relevant formulas consisting of prerequisites and negated consistency assumptions of defaults:

$$\begin{aligned} C_D = &\ \{c \mid (a : b_1, \cdots, b_k/c) \in D\} \\ \mathcal{L}_D = &\ \{q \mid (a : b_1, \cdots, b_k/c) \in D, \\ &\ q \in \{a, \neg b_1, \ldots, \neg b_k\}\} \end{aligned} \quad (14)$$

Defaults and classical proofs are translated separately into justifications. Every default yields a non-monotonic justification:

$$\begin{aligned} NM_D := &\ \{\langle in(a), out(\neg b_1, \ldots, \neg b_k) \rightarrow c\rangle \mid \\ &\ (a : b_1, \cdots, b_k/c) \in D\} \end{aligned} \quad (15)$$

Furthermore, we need all proofs of relevant formulas from premises W and consequents C_D. For this purpose, we define a set $M_W(U, R)$ of monotonic justifications for a domain $U \subseteq \mathcal{L}$, a range $R \subseteq \mathcal{L}$, and a theory $W \subseteq \mathcal{L}$:

$$\begin{aligned} M_W(U, R) := &\ \{\langle in(Q) \rightarrow q\rangle \mid Q \text{ is a minimal} \\ &\ \text{subset of } U \text{ s.t. } Q \cup W \models q\} \end{aligned} \quad (16)$$

As pointed out in [Junker and Konolige, 1990], these minimal arguments for a goal q can be determined by consequence-finding algorithms based on linear-ordered resolution. The justifications in $M_W(U, R)$ are sufficient to check for every element of R whether it can be derived *from any subset of U*.

In [Junker and Konolige, 1990], we have considered $M_{W'}(C_D, \mathcal{L}_D)$ for a default theory $\Delta = (D, W')$. Here, we take a slightly changed encoding since we want to handle default theories with different premises. For this purpose, we split W' into a a fixed part W and a changing part X which is a subset of an upper bound C. Thus, we obtain a set of default theories for D, W, and C:

$$\mathcal{S} := \{(D, W \cup X) \mid X \subseteq C\} \quad (17)$$

These default theories have a common TMS-network. Its non-monotonic justifications are NM_D and its monotonic justifications are $M_W(C_D \cup C, \mathcal{L}_D)$:

$$\nu_{D,W,C} := (C_D \cup \mathcal{L}_D \cup C, NM_D \cup M_W(C_D \cup C, \mathcal{L}_D)) \quad (18)$$

To regain the relevant part of an extension we supply the network with the additional premises $X \subseteq C$ of the selected default theory:

Theorem 4.1 *Let W and C be subsets of $\mathcal{L}$ and D be a set of defaults over $\mathcal{L}$. Let $X \subseteq C$. There exists a bijective mapping of the set of extensions of $(D, W \cup X)$ to the set of TMS-extensions of $\nu_{D,W,C}$ and X.*

Using this network, we can handle default theories which differ in some part of their premise sets.

4.2 Mapping Levels to Clusters

Now, we are ready to map a level-based default theory $(W, (L_1, \ldots, L_k))$ to TMS. We map every level to a TMS-network getting the nodes of lower levels as input. It is important that the different subnetworks are disjoint (i.e. don't share any node). The translation profits from the results in [Junker and Konolige, 1990]: We relate every level to a default theory and map it to a TMS-network using the translation in section 4.1. According to the results of section 3.2 we obtain the extensions of the complete TMS-network by a recursive characterization which is very similar to the definition of a preferred subtheory. Hence, it is not difficult to show that the extensions of this network are exactly the preferred subtheories of the level-based default theory.

Below, we elaborate this in detail. Let $\Delta = (W, (L_1, \ldots, L_k))$ be a level-based default theory. A preferred subtheory of level i is a subset of $L_i' := L_1 \cup \ldots \cup L_i$. As pointed out in section 2.1, we obtain for every level L_i different default theories $\Delta_i := (D_{L_i}, W_i)$ having a varying premise set W_i. Such a set W_i consists of W and a preferred subtheory of the next lower level which is a subset of L_{i-1}'. Then we get the following result:

Theorem 4.2 *Let $\Delta = (W, (L_1, \ldots, L_k))$ be a level-based default theory. Let $\mathcal{S}_i := \{(D_{L_i}, W \cup T') \mid T'$ is a preferred subtheory of level $i-1\}$. Then T is a preferred subtheory of level i iff T is an extension of a default theory in $\mathcal{S}_i$.*

In section 4.1, we extended the TMS-translation of default theories to capture changing premise sets. For level i, the changing part is $C_{D_{L_{i-1}'}}$ containing all consequents of defaults of lower levels. Note that $C_{D_{L_i}} = L_i$ and $C_{D_{L_i'}} = L_i'$ since the defaults in D_{L_i} have the simple form $\frac{:q}{q}$ for $q \in L_i$. Thus, we obtain the network $\nu_i := \nu_{D_{L_i}, W, L_{i-1}'}$. If we supply it with a preferred subtheory of level $i-1$ we get a preferred subtheory of level i as an extension of ν_i.

Thus, we could get the preferred subtheories of the level-based default theory if we link the networks of its levels. Let ν_i be equal to (N_i, J_i). First, we consider the networks for the levels 1 to i:

$$\nu_{\Delta,i} := (N_1 \cup \ldots \cup N_i, J_1 \cup \ldots \cup J_i) \qquad (19)$$

The complete network ν_Δ is equal to $\nu_{\Delta,k}$.

Now, we would like to apply the results of section 3.2 in order to relate the extensions of subnetworks to the preferred subtheories of a certain level. First, we have to identify independent components. Since every default is positive and occurs in exactly one level the sets N_i of

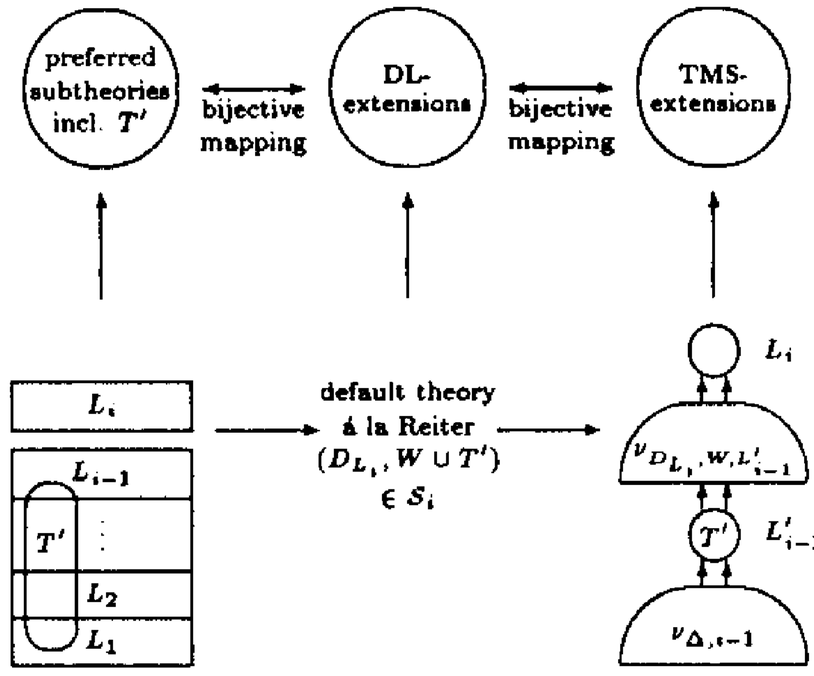

Figure 2: Illustration for theorem 4.4

nodes are mutually disjoint. Hence, there is no justification leading from $N_i - (N_1 \cup \ldots \cup N_{i-1})$ to $N_1 \cup \ldots \cup N_{i-1}$:

Lemma 4.3 *Let $\Delta = (W, (L_1, \ldots, L_k))$ be a level-based default theory. Then the set $N_1 \cup \ldots \cup N_{i-1}$ of the nodes of $\nu_{\Delta,i-1}$ is an* independent component *of $\nu_{\Delta,i}$.*

After getting all technical prerequisites, we are able to apply theorem 3.2 to the subnetworks of the level-based default theory. Thus, we can prove the following theorem by induction on i using the theorems 4.2, 4.1, and 3.2 (cf. figure 2):

Theorem 4.4 *Let $\Delta = (W, (L_1, \ldots, L_k))$ be a level-based default theory. Then there exists a bijective mapping of the set of extensions of the subnetwork $\nu_{\Delta,i}$ to the set of preferred subtheories of level i.*

The mapping is indeed very simple: Just take the nodes of an extension that are in L_i' to obtain a preferred subtheory. In turn, we get the main result of this paper:

Theorem 4.5 *(Main) Let $\Delta = (W, (L_1, \ldots, L_k))$ be a level-based default theory. Then there exists a bijective mapping of the set of extensions of the network ν_Δ to the set of preferred subtheories of Δ.*

Thus, we have translated a level-based default theory into a TMS-network. It reflects the level-structure of the default theory. It could also be obtained from the network of a level-free default theory by removing justifications leading from a subnetwork of a higher level to a subnetwork of a lower level.

Our approach can also be applied to a restricted kind of *prioritized circumscription* which is obtained by adding unique names assumption and domain closure. Due to [Baker and Ginsberg, 1989], such a circumscription theory can be mapped to a level-based default theory, which is in turn translated to a TMS-network. For circumscription, *goal-directed reasoning* is required. To prove a goal $q \in \mathcal{L}$ we need further justifications for proofs of q. After that, we can get rid of a large part of the network. According to theorem 3.2, we can focus our attention to the minimal independent component con-

taining the goal q because the remaining subnetwork is coherent in this case.

4.3 Network for Davis' Familiar Example

Consider again the circuit example of section 2.3. In this section, we translate the default theory Δ_{circ} into a TMS-network. The relevant first-order proofs can be extracted from the *conflict sets*, i.e. minimal sets of hypotheses that are inconsistent in conjunction with the premises. According to [de Kleer and Williams, 1987], we get two conflict sets:

$$\begin{array}{l} \{ok(M1), ok(M2), ok(A1)\} \cup W \models \bot \\ \{ok(M1), ok(M3), ok(A1), ok(A2)\} \cup W \models \bot \end{array} \quad (20)$$

We get the arguments for the relevant formulas $\neg ok(o)$ by applying the deduction theorem to the conflict sets.

Next, we consider the subnetworks for both levels. The first level contains only adders leading to two non-monotonic justifications. ν_1 consists of

$$\begin{array}{l} \langle out(\neg ok(A1)) \rightarrow ok(A1)\rangle \\ \langle out(\neg ok(A2)) \rightarrow ok(A2)\rangle \end{array} \quad (21)$$

We don't get monotonic justifications in the first level because every conflict set contains some multiplier which belongs to a higher level. For the second level, we obtain a non-monotonic justification for each multiplier:

$$\begin{array}{l} \langle out(\neg ok(M1)) \rightarrow ok(M1)\rangle \\ \langle out(\neg ok(M2)) \rightarrow ok(M2)\rangle \\ \langle out(\neg ok(M3)) \rightarrow ok(M3)\rangle \end{array} \quad (22)$$

Additionally, ν_2 contains justifications for the negations of correct models of multipliers that are obtained from the conflicts of equation 20:

$$\begin{array}{l} \langle in(ok(A1), ok(M2)) \rightarrow \neg ok(M1)\rangle \\ \langle in(ok(A1), ok(A2), ok(M3)) \rightarrow \neg ok(M1)\rangle \\ \langle in(ok(A1), ok(M1)) \rightarrow \neg ok(M2)\rangle \\ \langle in(ok(A1), ok(A2), ok(M1)) \rightarrow \neg ok(M3)\rangle \end{array} \quad (23)$$

This network has two non-monotonic loops sharing *-ok(M1).* To find labelings, we choose a label for this node and proceed with propagation. In general, we first consider networks for lower levels. Since the network lacks rnonotonic loops no groundedness check is necessary as in [Junker and Konolige, 1990].

5 Conclusion

We pointed out how prioritized default theories [Brewka, 1989] can be used to obtain a TMS-based system for generating *preferred diagnoses.* To achieve this goal, we showed three results, which are of their own interest:

- Prioritized defaults allow to express preferences between models of behaviour in diagnosis from first principles. Thus, the diagnosis process can be focused to candidates that should be investigated first.
- Extensions of TMS-networks having independent components can be split into extensions of subnetworks. Hence, divide-and-conquer methods may be used to compute extensions. An example for this is Goodwin's TMS-algorithm [Goodwin, 1987].
- Levels of defaults are mapped to TMS according to [Junker and Konolige, 1990] and then linked together using the result above. Thus, we *get* a TMS-based prover for prioritized defaults which is an alternative to Baker's and Ginsberg's prioritized circumscription prover [Baker and Ginsberg, 1989].

Priorities on general defaults have also been considered in [Brewka, 1989], Our TMS-translation carries over to these defaults provided they don't share components. This condition *ensures* disjoint subnetworks.

An issue for future work is to see whether prioritized defaults can substitute fault probabilities [de Kleer and Williams, 1987], [de Kleer, 1990] in practical problems.

Acknowledgements

The different ideas presented in this paper arose from discussions with different persons. I would like to thank Prof. M. Richter, Kurt Konolige, Gerd Brewka, and Hans Voss. Peter Struss and Gerhard Friedrich gave helpful comments. This work has been supported by the Federal Ministry for Research and Technology within the joint project TASSO (Grant No. ITW8900A7).

References

[Baker and Ginsberg, 1989] A.B. Baker and M L. Ginsberg. A theorem prover for prioritized circumscription. In *IJCAI-89,* pages 463-467, 1989.

[Brewka, 1989] G. Brewka. Preferred subtheories: An extended logical framework for default reasoning. In *IJCAI-89,* pages 1043-1048, 1989.

[de Kleer and Williams, 1987] 3. de Kleer and B.C. Williams. Diagnosing multiple faults. *Artificial Intelligence,* 32:97-130, 1987.

[de Kleer, 1990] J. de Kleer. Using crude probability estimates to guide diagnosis. *Artificial Intelligence,* 45:381-391,1990.

[Dressier and Struss, 1990] 0, Dressier and P. Struss. Back to defaults: Computing diagnoses as coherent assumption sets. Technical report, Siemens AG, 1990,

[Goodwin, 1987] J W. Goodwin. *A Theory and System for Non-Monotonic Reasoning.* PhD thesis, University of Linkoping, Linkoping, Sweden, 1987.

[Junker and Konolige, 1990] U. Junker and K. Konolige. Computing the extensions *of autoepistemic* and default logics with a truth maintenance system. In *AAAI-90,* pages 278-283, 1990.

[Poole, 1988] D. Poole. A logical framework for default reasoning. *Artificial Intelligence,* 36:27-47, 1988.

[Poole, 1989] D. Poole. Normality and faults in logic-based diagnosis, In *IJCAI-89,* pages 1304-1310,1989.

[Reiter, 1987] R. Reiter. A theory of diagnosis from first principles. *Arttfictal Intelligence,* 32:57-952, 1987.

[Struss and Dressier, 1989] P. Struss and 0. Dressier. Physical negation - integrating fault models into the general diagnostic engine. In IJ*CA1-89,* pages 1318-1323, 1989.