On the Semantics of Conflict Resolution in Truth Maintenance Systems

Catholijn Jonker

Logic Group
Preprint Series
No. 65
March 1991

Department of Philosophy
University of Utrecht
Heidelberglaan 2
3584 CS Utrecht

On the Semantics of Conflict Resolution in Truth Maintenance Systems

Catholijn Jonker

University of Utrecht, Department of Pilosophy,
Heidelberglaan 2, 3584CS Utrecht
The Netherlands

Key words and phrases: Provability,
Conflict resolution, TMS

On the Semantics of Conflict Resolution in Truth Maintenance Systems

C. M. Jonker
Utrecht University*

June 7, 1991

Abstract

A Truth Maintenance System (TMS) maintains a consistent state of belief given a set J of justifications, i.e. arguments for belief. To resolve contradictions dependency-directed backtracking is performed.

In this paper we introduce a method that can be used to track all dependency-directed backtracking methods simultaneously. This has been previously done by adding the contrapositions of justifications to J. We will show that contrapositions are not mandatory, namely that the same result can be reached by adding one disjunctive justification (the consequent is not one atom, but can be a disjunction of atoms).

Simplifying the added disjunctive justification leads to more informative models [Ross 89b].

Furthermore, we will give an alternative proof for the correctness of Witteveen's method [Witteveen 90a] for computing the Well-Founded model.

1 Introduction

Informally, a TMS is a program that manages a database of *nodes* and *justifications*. Each node stands for a piece of information. Each justification says that one node should be believed if some others are respectively believed and disbelieved. There are two kinds of truth maintenance systems. One kind is the justification-based TMS (JTMS), the other is the assumption-based TMS (ATMS).

A JTMS maintains a single context of belief (a model) at a time and supports nonmonotonic justifications. In everyday life, conclusions have to be drawn in the absence of complete information. This type of commonsense reasoning unavoidably depends on default assumptions, so it is intrinsically nonmonotonic: new information can cause old conclusions to loose their validity.

An ATMS maintains multiple models simultaneously, but does not support nonmonotonicity. This makes it suitable for applications where several partial solutions are to be considered and compared in the process of developing complete solutions. Contrary to JTMSs in which the switching of context to compare solutions is possible but inefficient.

In the sequel we focus on the JTMS, the nonmonotonicity supporting system.

A TMS can be seen as a system working together with a problem solver. Whenever the problem solver performs an inference step, it sends a justification to the TMS. The task of the TMS is to maintain a consistent state of belief. The TMS determines this state from the statements of belief, the reasons for belief, and the known inconsistent states of belief (constraints) which have been given to the TMS by the problem solver.

Among the normal justifications, constraints may exist. Constraints say that a conflict or contradiction arises, if the nodes of some set are believed and the nodes of another set are disbelieved.

If a conflict occurs, some sort of backtracking must be performed to reach a consistent state of belief. In the process of backtracking new beliefs are formed by reasoning from the conflict: if

*Department of Computer Science, and Department of Philosophy, University of Utrecht, Utrecht, the Netherlands. Email: cmjonker@cs.ruu.nl and cmjonker@phil.ruu.nl.

belief in n nodes causes a conflict, then we can reason in the contrapositive direction: Belief in $n-1$ of the nodes indicates that we should disbelieve the last node. This implies that there are n such contrapositive reasonings possible for this one conflict.

Giving a semantics for belief revision one can produce a consistent state of belief for every possible contrapositive or one can give one consistent state of belief which satisfies every contraposition.

The problem is that one has to produce every contraposition, and their number is linear in the number of beliefs causing the conflict. Furthermore, two kinds of negation get mixed. In TMSs negation as failure is used, meaning that we only disbelieve a node if there is no reason to believe it. However, reasoning in the contrapositive direction can tell us to believe in the falsehood of a node (so to disbelieve it) if others are respectively believed of disbelieved. This belief in falsehood is by definition not a negation as failure. As only negation as failure is allowed in justifications, new nodes are created to stand for the negated node. Then extra justifications must be added to ensure that the new node and the original node get opposite truth-values.

In this paper we propose a way to reason from the contradiction in which no contrapositions are needed. Instead, only one, possibly disjunctive, justification is added. Furthermore, we do not have the problem of new negations.

This new approach to conflict resolution is called Dependency-directed Minimal Backtracking (DDMB), which makes no choices during backtracking. We show that a consistent model of belief exists for this approach, which corresponds to the proposal of Witteveen ([Witteveen 91]). This proves that contraposistions are not necessary to perform dependency-directed backtracking. Simplifying the justification added in the DDMB approach creates intrinsic extensions of Witteveen's proposal.

In the next section preliminaries for the rest of the paper and the terminology used in this paper will be described.

In section 3 several of the semantics for logic programs will be discribed in short. The Stable and Well-Founded semantics for logic programs will be discussed in more detail, because these semantics are directly applicable to TMSs and will be referred to frequently. Here the alternative proof for the correctness of Witteveen's method for computing the Well-Founded model will be given.

Section 4 explains when conflicts arise and how these conflicts depend on the truth-values of the nodes. Then the use of contrapositions is explained and dependency-directed backtracking is described in short. The last part of this section describes two semantics for conflict resolution based on contrapositions.

In section 5 we introduce a new approach to depencency-directed backtracking based on reasoning from a conflict, but not on contrapositions. Two variations of this approach will be explained in detail and compared with the semantics of the previous section.

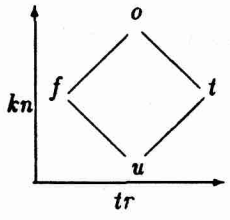
In section 6 we give a detailed example of the different approaches and semantics.

2 Preliminaries

A TMS is a pair $D = (N, J)$, where N is a finite set of propositional atoms, called nodes, and J is a finite set of directed propositional formulas j, called justifications, of the form $\alpha \wedge \beta \rightarrow c$, where c is an atom, called the conclusion or head of j ($hd(j) = c$), α is the conjunction of positive literals (i.e. atoms) and β the conjunction of negative literals (i.e. negated atoms). $\beta(j)$ denotes the nonmonotonic part of justification j. The body of a justification $j = \alpha \wedge \beta \rightarrow c$ is $body(j) = \alpha \wedge \beta$. Justifications are called directed because c is distinguished from β in the sense that $\sim b \rightarrow c$ is not the same as $\sim c \rightarrow b$. $At(\alpha)$ ($At(\beta)$) denotes the set of atoms in α (β). $Lit(\phi)$ is the set of literals used in wff ϕ. The intended meaning of such a justification is the following: "Belief in the elements of $At(\alpha)$ and disbelief in the elements of $At(\beta)$ justify the belief in c". A premise can now be seen as a justification with an empty antecedent. A constraint is a justification $\alpha \wedge \beta \rightarrow \bot$ where $\bot$ is seen as the node indicating a contradiction if it has truth-value *true*. If $\alpha \wedge \beta$ is believed, $\bot$ gets truth-value *true* as well.

In this paper we will use 3-valued interpretations. In 3-valued interpretations a third value called *undefined*, written u, is used for the nodes with no reason for belief or disbelief. An interpretation is a tuple $I = (I_t, I_f)$. I_t (I_f) is the set of atoms considered true (false) in the interpretation.

In this paper two orderings are used. One to compare truth-values in an interpretation and one to compare interpretations. These orderings are the orderings of Belnap's four valued logic, called $\mathcal{FOUR}$. In $\mathcal{FOUR}$ one can think of truth-values as sets of truth values in the ordinary sense, namely $\{true\}$, which we will write as t, $\{false\}$, written as f, $\{\}$, which we will write as u and read as underdetermined or as undefined; and $\{true, false\}$, written as o and read as overdetermined. The four truth-values can be given a simple natural mathematical structure. We give $\mathcal{FOUR}$ two partial orderings namely the *knowledge ordering*, denoted $\leq_{kn}$, and the *truth ordering*, denoted $\leq_{tr}$ as depicted adjacent.



Thus $a \leq_{kn} b$ if there is a way up from a to b, where the intuition is "b gives more information than a" so $\leq_{kn}$ represents an increase in knowledge.
And $a \leq_{tr} b$ if there is a path from left to right from a to b. The intuition is that $\leq_{tr}$ represents an increase in "truth" (or decrease in "falseness").
Given an interpretation I propositions are evaluated as follows:

- $val_I(\sim a) = \sim val_I(a)$, where $\sim t = f$ and $\sim f = t$ and $\sim u = u$
- $val_I(\phi \wedge \psi) = min\{val_I(\phi), val_I(\psi)\}$
- $val_I(\phi \rightarrow \psi) = t$ if $val_I(\phi) \leq val_I(\psi)$ otherwise f.

Instead of val_I we will write I. Note that $\phi \rightarrow \psi \neq \sim \phi \vee \psi$ as it is classically.

Definition 2.1 *Let M and M' be interpretations of a* JTMS *D.*

- *M is a* model *of D if it satisfies every justification in D.*
- *$M \sqcup M' = (M_t \cup M'_t, M_f \cup M'_f)$ is called the* union *of M and M'.*
- *M and M' are* consistent *if $M \sqcup M'(\bot) \neq t$.*
- *M' is an* extension *of M if M and M' are consistent and $M_f \subseteq M'_f$ and $M_t \subseteq M'_t$.*

Notice that with this definition a 3-valued interpretation can be a 3-valued model while not having a 2-valued extension which is a 2-valued model. An example can be found in example 4.1.

When considering a semantics for a *TMS* a choice is to be made between a minimal, unique, model and a maximal, not unique, model. A minimal will in general be 3-valued, some choices have not been made. A maximal model is 2-valued, but several choices have been made, for which there is no grounded proof. Which of these models to prefer, depends on the application. But some nodes can only have one truth-value (t or f), because the other choice would lead to a contradiction ($\bot$). Such a node has an *intrinsic* truth-value.

Definition 2.2 *A model M of J is* intrinsic *if M is consistent with every other model of J.*

Definition 2.3 *A model M of J is* maximal intrinsic *if M is a intrinsic model of J, which extends every other intrinsic model of J.*

Let M be an intrinsic model of J, then $a \in M_t$ ($a \in M_f$) implies that $M' = (M_t - \{a\}, M_f \cup \{a\})$ ($M' = (M_t \cup \{a\}, M_f - \{a\})$) is not a model of J. For example, let $J = \{\sim a \rightarrow b, \sim b \rightarrow a, a \rightarrow c, b \rightarrow c\}$, then M is a model of J if and only if $M = (\emptyset, \emptyset)$ or $c \in M_t$. But only $(\emptyset, \emptyset)$ and $(\{c\}, \emptyset)$ are intrinsic models of J.

Definition 2.4 *[vGRS 88]*

- *For a set of literals S we denote the set formed by taking the complement of each literal in S by $\circ S$.*
- *For an interpretation I define $\circ I$ to be the set of literals $\circ I_t \cup \circ I_f$.*
- *Literal q is* inconsistent *with S if $q \in \circ S$.*
- *Sets of literals R and S are* inconsistent *if some literal in R is inconsistent with S (or vice versa), i.e., if $R \cap \circ S \neq \emptyset$*
- *A set of literals is* inconsistent *if it is inconsistent with itself; otherwise it is* consistent.
- *Sets of literals R and S are* disjoint *if no literal in R has the same atom as a literal in S (or vice versa), i.e., if $R \cap S = R \cap \circ S = \emptyset$*

Definition 2.5 *Let $D = (N, J), j \in J$, we define*

- $C(j, I) := \{hd(j) \mid I(body(j)) = t\}$
- $C(D, I) := \bigcup\{C(j, I) \mid j \in J\} \cup I$
- $Cl(D, I) := (\lambda I'.C(D, I'))^{\omega}(I)$

3 Semantical issues of Logic Programming

Whenever there are several models for a set of justifications of a JTMS it is not clear what the semantics should be, as a JTMS maintains only one model at a time. Given the similarity of a set of justifications and a logic program, it makes sense to look to the semantics of logic programs, when searching for a semantics for sets of justifications.

Definition 3.1 *A* General Logic Program *is a set of general rules, which may have both positive and negative subgoals. A general rule is written with its* head *(conclusion) to the right and its subgoals (body) to the left of the symbol "$\rightarrow$", which may be read "if".*

An example of a general rule is $a(X), \sim b(X) \rightarrow p(X)$. For more information and further details I refer to [Lloyd 87].

Logic Programming has the same problem, for there the semantics should be the one *intended* by the programmer. As Shepherdson [Sheperdson 86] said:

> "Even if the practicing logic programmer does not regard the written text of the program as its declarative meaning, we feel that in order to be true to the basic aims of logic programming two fundamental principles should be observed.
>
> 1. The semantics of negation should be clear and easily intelligible. That is: the naive programmer should be able to understand the full meaning of what he writes.
> 2. The syntax should be computable. That is, at least in theory, an automatic proof procedure should exist. If the only reason for abandoning the clear and well-known concept of classical negation is the inefficiency of its implementations, we might not unreasonably ask for a complete proof procedure to be feasibly implementable."

Przymusinski [Przymusinski5] agrees with the first point, but unlike Shepherdson he sees other reasons for abandoning classical semantics:

> "We really do not want classical semantics for logic programs. We want logic programs to be as easy to write and comprehend as possible, free from overwhelming amounts of explicit negative information and as close to natural discourse as possible. In other words we want the semantics of a logic program to be determined more by its *commonsense meaning* than by its purely logical contents."

As we do not want negative information explicitly in our programs, several ideas to avoid this have come forward like Circumscription, the Closed World Assumption, default logic, Autoepistemic Logic, Completion, Stable Models, Stratification, Perfect and Well-Founded models. One of the most important notions in logic programming is the notion of *Negation as Failure*. By *negation as failure* we mean the result of adding negation as failure to SLD-resolution (see [Lloyd 87]), i.e., the query evaluation procedure described by Clark in [Clark 78], where a negative literal is selected only if it is a ground literal $\neg p$. The next step is then to query p; if p succeeds, then the evaluation of $\neg p$ fails; if p fails on every evaluation path then, $\neg p$ succeeds. Since by König's lemma the evaluation tree is then finite, this is usually called *finite failure* to distinguish it from the situation where the evaluation tree has no succesful path but has infinite paths.

The different evaluation paths correspond to the different program clauses whose head matches the chosen literal. Different query evaluation procedures result from different rules for selecting the literal to be resolved on.

Suppose $\neg p$ succeeds, and there is a clause $\neg p \rightarrow q$ in program, then in a subsequent step, q succeeds. For JTMSs this means that disbelief in some nodes can lead to belief in others not believed earlier. So in fact when negation as failure is used, we use 4-valued logic instead of 3-valued logic (or similarly we use 3-valued logic when we say we use 2-valued logic). Because we do not only have true as "provable true", but also true as "true as failure" (consequence of nodes false).

For more information see [Sheperdson 86] and [Lloyd 87].
Several of the proposed semantics for Logic Programming will be discussed here.

3.1 Circumscription (CIRC)

Circumscription has been introduced by McCarthy [McCarthy 80] but later new formulations have been made by McCarty himself [McCarthy 84] and by V. Lifschitz [Lifschitz 85] and [Lifschitz 86]. But the principle idea to all versions of circumscription is the existence of a circumscription axiom. It is essentially a second-order axiom that is assumed to implicitly accompany any first-order theory. For a formal definitions see [McCarthy 80], [McCarthy 84] or [Lifschitz 86].

Informally, circumscribing a wff P, with free variables $\vec{x}$, in a theory T is to limit the instances of $\vec{x}$ which satisfy P to only those that are necessary in the light of T.

Example 3.2

Let P be *CanFly(x)* and let T be

$\{\forall b(Bird(b) \rightarrow CanFly(b)),$
$Bird(Tweety)\},$

Then circumscribing P has the effect that the only object that can fly is *Tweety*.
Note that this is a way of preferring the models in which only *Tweety* can fly to other models.

3.2 Closed World Assumption (CWA)

The central idea of Reiter's Closed World Assumption [Reiter 78] is that the program contains all the positive information about the objects in the domain and any positive ground literal that is not implied by the program is false. Since this is defined in terms of non-provability it is not surprising that principle 2 above is violated.

In other words the closed world assumption says that for any relation R, and any individuals $x_1, \ldots, x_n$ one can assume $\neg R(x_1, \ldots, x_n)$ whenever it is consistent to do so.

Example 3.3 *See [Reiter 78].*

Consider the following database, which consists of ground literals only.

$$
\begin{array}{lcl}
|Teacher| & = & \{a, b, c, d\} \\
|Student| & = & \{A, B, C\} \\
Teach(a,A) & & \\
Teach(b,B) & & \\
Teach(c,C) & & \\
Teach(a,B) & &
\end{array}
$$

Now consider the query: Who does not teach B?

Intuitively, we want $\{c, d\}$, but first order logic interprets the database literally, all the logic knows for certain is what is explicitly represented in the database. Just because $\neg Teach(c,B)$ is not present in the database is no reason to conclude that $\neg Teach(c,B)$ is true. Rather, as far as first order logic is concerned, the truth of $Teach(c,B)$ is unknown. Thus, we would also have to include all required negative facts about *Teach*, like $\neg Teach(c,B)$.

Reasoning about a world under the closed world assumption considerably simplifies the representation of that world. Only positive information need be explicitly represented, negative information is inferred by default. Since in general the amount of negative information about a world vastly exceeds the positive things known, there is a considerable computational and representational advantage to reasoning under the closed world assumption.

Different intuitions appear to underlie the closed world assumption, leading to different formalizations, see [EMR 87]:

1. Negation as failure to derive; If a ground atomic formula cannot be proved using the given information as premise, then assume the formula's negation.
2. Predicate completion; When the given information about a predicate consists of a set of sufficient conditions on that predicate, assume that these conditions are also necessary.
3. Domain circumscription; Assume that the individuals required by the given information are all there are.
4. Predicate circumscription; Assume those facts that are true in all models of the given theory having minimal extensions for certain predicates.

The relations among these different formalisms are not completely understood.

3.3 Default logic

One of the nonmonotonic logics cited often is Reiter's default logic [Reiter 80]. Default logic represents a form of plausible inference, which is typically required whenever conclusions must be drawn despite the absence of total knowledge about a world.

A good deal of what we know about a world is 'almost always' true, with a few exceptions. The fact that most birds fly is represented in first order logic by explicitly listing the exceptions to flying. But with this representation one cannot conclude of a 'general' bird that it can fly. For example trying to prove that Tweety flies when all we know is that Tweety is a bird, one must establish the subgoal $\neg$Penguin(Tweety) $\wedge\neg$Ostrich(Tweety)$\wedge\ldots$, which is impossible given that there is no further information on Tweety. However intuitively we want to deduce that Tweety flies. So we allow that Tweety flies by default. The interpretation is as follows: "If x is a bird and it is consistent to assume that x can fly, then infer that x can fly".

More formally the default rule is:

$$\frac{Bird(x) : Fly(x)}{Fly(x)}$$

The exceptions to flight are then given a standard first order representation. Notice that if *Fly*(Tweety) is inferred by default then the assertion *Fly*(Tweety) has the status of a belief; it is subject to change, when additional information contradicting this belief becomes available.

For a formal definition of the consistency requirement see [Reiter 80]. A good intuitive interpretation is to view this consistency requirement with respect to all of the first order facts about the world, together with all of the other beliefs sanctioned by all of the other default rules in force.

3.4 Autoepistemic logic (AEL)

Autoepistemic logic, developed by Robert Moore [Moore 83], is a model of an ideally rational agent's reasoning about his own beliefs. The best way to explain about autoepistemic logic is probably to point out the differences between default logic and autoepistemis logic and then to give an example. For more information see also [Konolidge 87].

By default reasoning we mean the drawing of plausible inferences from less-than-conclusive evidence in the absence of information to the contrary. The examples about birds being able to fly are of this type. Autoepistemic reasoning, while different from default reasoning, is an important form of commonsense reasoning in its own right. Moore's example is the following:

Example 3.4

> Consider my reason for believing that I do not have an older brother. It is surely not that one of my parents once casually remarked, "You know, you don't have any older brothers", nor have I pieced it together by carefully sifting other evidence. I simply believe that if I did have an older brother I would know about it; therefore, since I don't know of any older brothers, I must not have any.

This is quite different from the tentative conclusion that Tweety can fly, given that we know that Tweety is a bird and that we have no information to the contrary.

Default reasoning and autoepistemic reasoning are both nonmonotonic, but for different reasons. Default logic is nonmonotonic because it is defeasible, i.e. its conclusions are tentative, so, given better information, they may be withdrawn. Purely autoepistemic logic, however, is not defeasible. If you really believe that you already know all the instances of birds that cannot fly, you cannot consistently hold to that belief and at the same time accept new instances of birds that cannot fly.

These two forms of reasoning can be combined, we might believe that we know about *most* of the birds that cannot fly. This would lead to defeasible autoepistemic inferences, but theit defeasibility would be the result of their also being default inferences.

Autoepistemic logic is intended to model the beliefs of an agent reflecting upon his own beliefs. The primary objects of interest are sets of autoepistemis logic formulas that are interpreted as the total beliefs of such agents. Such a set of formulas will be called an autoepistemic theory. The truth of an agent's beliefs, expressed by which propositional constants are true in the external world and which formulas the agent beliefs. The operator L is used to differentiate between truth in the external world and truth with respect to an agents belief. A formula LP will be true with respect to an agent if and only if P is in his set of beliefs; P need not be true in the external world and other agents possibly do not belief P.

Considering a semantics for autoepistemic logic, the following question rises: given an initial set of beliefs A about the world, what final set T should an ideal introspective agent settle on? If you forget about the self-belief operator L for the moment, then an ideal agent should believe all of the logical consequences of his initial beliefs.

Let $\Gamma \models \phi$ mean that the sentence ϕ is logically implied by the set of sentences Γ. Then if A is the initial set of beliefs, the belief set T of an ideal agent is given by:

$$T = \{\phi \mid A \models \phi\}$$

The presence of a self-belief operator complicates matters. Because the intended meaning of $L\phi$ depends on the belief set of the agent, the definition of the belief set itself becomes circular, which

necessitates the use of a fixed-point equation to define T. So an appropriate choice for logical implication must be made. Given that the intended meaning of L is self-belief, it becomes obvious that we should consider all models in which the interpretation of $L\phi$ is the belief set of the agent itself.

3.5 Completion

The idea behind the Clark completion of a program is to collect all rules for a proposition into a single rule whose body is disjunction of conjunctions, then replace the "if" symbol, "$\rightarrow$", by "$\leftrightarrow$". This states in effect that the proposition is completely defined by the given rules.

The original "logical consequence" approach declares that only conclusions that are logical consequences of the program completion should be drawn. This yields an implicit 3-valued interpretation, by assigning atom p an "undefined" truth-value, u, when neither p nor $\neg p$ is a logical consequence. The "$\leftrightarrow$" produced by the program completion process is interpreted as Łukasiewicz's operator of "having the same truth value", so that $u \leftrightarrow u$ evaluates to true.
Consider the program P with only the rule $\neg p \rightarrow p$. The only two-valued model of P is that in which p is true. But the program's completion is $\neg p \leftrightarrow p$, which has no 2-valued model.

3.6 Stable model

The Stable model semantics has been introduced by Gelfond en Lifschitz in [GL 88]. The semantics is 2-valued and is based on Moore's Autoepistemic Logic. In this semantics every choice made is grounded, i.e. does not depend on a circular chain of beliefs. Informally a model is a stable model if it reproduces itself under a certain operation. This operation, is defined as follows:

Definition 3.5 *(2-valued stable) Let $D = (N, J)$ be a* TMS. *Then M is a stable model of D iff M is the unique minimal model of $D' = (N, J(M))$, where $J(M) = \{\alpha \rightarrow c \mid \alpha \wedge \beta \rightarrow c \in J, M \models \beta\}$.*

The stable model semantics has a few drawbacks as it is not universal and not unique. However its reproductive property is attractive as is the easy method of checking whether or not a model is stable. In order to overcome the problem of non-universality, several people proposed (3-valued) generalizations of stable models.

Giordano and Martelli [GiMar1] proposed generalized stable models to handle conflict resolution in Truth Maintenance. More details of their proposal are given in section 4.2.

Przymusinska and Przymusinski gave a definition of a 3-valued stable model of Logic Programs in [PaPi 90b], which Witteveen [Witteveen 90] translated to a definition for JTMSs.

Definition 3.6 *(3-valued stable) Let $M = (M_t, M_f)$ be an interpretation of $D = (N, J)$ and let $J(M, t) = \{\alpha \rightarrow c \mid \alpha \wedge \beta \rightarrow c \in J$ and $M(\beta) = t\}$ and $J(M, u) = \{\alpha \rightarrow c \mid \alpha \wedge \beta \rightarrow c \in J$ and $M(\beta) \geq_{tr} u\}$. M is a* 3-valued stable *model of D iff*

- *(a) M_t is the truth-minimal 2-valued model of $D(M, t) = (N, J(M, t))$ in short: $M_t = Cl((N, J(M, t)))$*
- *(b) $M_t \cup M_u$ is truth-minimal 2-valued model of $D(M, u) = (N, J(M, u))$ in short: $M_t \cup M_u = Cl((N, J(M, u)))$.*

3.7 Stratified semantics

Stratification was introduced into logic programming, based on the idea that relations must be completely defined before they can be used negatively. Though not all programs are stratifiable, and for those that are the stratification need not be unique, the stratified semantics assigns a two-valued meaning to a stratified program that is independent of its stratification.

Definition 3.7 *A* stratified program P *is one that can be partitioned*

$$P = P_1 \cup \ldots \cup P_n$$

so that if a proposition occurs positively in the body of a clause in P_i, all clauses where it occurs in the head are in P_j with $j \leq i$, and if a proposition occurs negatively in the body of a clause in P_i, then all clauses where it occurs in the head are in P_j with $j < i$. So the clauses of P_i define the i^{th} level propositions.

The stratified semantics proposed by Apt, Blair and Walker [ABW 86], called M_P is $M(P_n)$, where $M(P_1)$ is defined as the intersection of all Herbrand models of P_1 and $M(P_{i+1})$ as the intersection of all Herbrand models of P_{i+1} whose intersection with the Herbrand base of P_i is $M(P_i)$.

For more information see [Van Gelder 86], [FBJ 88], [ABW 86], [PaPi 88], [Przymusinski 89] and [Przymusinski5].

3.8 Perfect model

The perfect model semantics was introduced by Przymusinski [Przymusinski5] based on the class PERF(P) of all, not necessarily Herbrand, *perfect models* of a program P. Only two-valued models are considered. First the notion of a *dependency graph* G of the program P must be introduced. The vertices of G are proposition symbols occurring in P. If p and q are proposition symbols, then there is a directed edge in G from q to p if and only if there is a clause (rule, justification) in P such that p is its conclusion and q appears in its body. If q appears negatively in this body, then the edge is called *negative*. For any two proposition symbols in P we say that p *has lower priority than* q ($p < q$) if there is a directed path in G from p to q containing at least one negative edge. This relation is the *priority relation* (see also [Przymusinski 87]).

If M is a model of P and if a new model N is obtained from M by changing extensions of some propositions in M, then N is *preferable* to M if and only if addition of some new element(s) to the extension of a higher priority proposition p is always *justified* by the removal of some elements from the extension of a lower priority proposition q, i.e. such that $q < p$. The reason for this is that the goal is to minimize extensions of low priority propositions as much as possible, and we are willing to do that even at the cost of enlarging extensions of propositions of higher priority.

Now we can define a *perfect* model as: A model M is *perfect*, if there are *no* models preferable to it. And the perfect model semantics as:

Definition 3.8 *Let p be a logic program and let PERF(P) be the set of all perfect models of P. By the* perfect model semantics *of P we mean the semantics induced by the set PERF(P). Under this semantics a sentence ϕ is considered to be true if and only if ϕ is satisfied in all perfect models of P.*

Theorem 3.9 *[Przymusinski5] Every perfect model is minimal. (As only two-valued models are considered, minimal is here truth-minimal.)*

Example 3.10 *[Przymusinski5]*

Let P be

$$\begin{array}{ll} & \rightarrow \ edge(a,b) \\ & \rightarrow \ edge(c,d) \\ & \rightarrow \ reachable(a) \\ reachable(Y), edge(Y,X) & \rightarrow \ reachable(X) \\ not\ reachable(X) & \rightarrow \ unreachable(X) \end{array}$$

P's unique Herbrand perfect model M consists of:
$edge(a,b)$, $edge(c,d)$, $reachable(a)$, $reachable(b)$, $unreachable(c)$, $unreachable(d)$.

However not every logic program has perfect models. For example, the program $P = \{\neg q \rightarrow p, \neg p \rightarrow q\}$ does not have any perfect models as a consequence of the fact that both $p < q$ and $q < p$.

3.9 Well-Founded model

In 1988 Van Gelder, Ross and Schlipf [vGRS 88] proposed a new definition of canonical model, which they called the Well-Founded model. This semantics is unique and universal, in that it always produces a 3-valued model and exactly one. In this subsection we will only describe their original formulation for the propositional case and the formulation by Witteveen [Witteveen 90] for JTMSs. Other general formulations have been proposed in [Van Gelder 89] and [Przymusinski 89].

The key idea in the original formulation by Van Gelder, Ross and Schlipf is the concept of an "unfounded" set. This concept is used to capture negation by failure.

Definition 3.11 *Given a* JTMS $D = (N, J)$ *and a 3-valued interpretation* I. *We say that* $A \subseteq N$ *is an* unfounded set of D with respect to I *(denoted* $unf(A, D, I)$*) if*

$$\forall c \in A\ \forall(\alpha \wedge \beta \rightarrow c) \in J : \begin{cases} I(\alpha \wedge \beta) = f \\ or \\ At(\alpha) \cap A \neq \emptyset \end{cases}$$

Equivalent with this definition is

Definition 3.12 *Let* $D = (N, J)$ *be a* JTMS *and* I *a 3-valued interpretation.*

- $Poss(D, X) = \{C(J, Y) \mid X \text{ and } Y \text{ consistent}\}$
- $unf(A, D, I)$ *if* $\circ A$ *is consistent with* $I \cup Poss(D, I \cup \circ A)$

Informally, the Well-Founded semantics uses the condition above to draw negative conclusions. Justifications that satisfy $I(\alpha \wedge \beta) = f$ are not usable for further derivations. Condition $At(\alpha) \cap A \neq \emptyset$ is the unfoundedness condition: of all the justifications that still might be usable to derive something in A, each requires an atom in A to be true. Note that not necessarily $A \cap I = \emptyset$ (however in the following this gives no problems as one starts with $I = \emptyset$). It is immediately clear that, with respect to any interpretation I, the union of arbitrary unfounded sets is an unfounded set.

Definition 3.13 *The* greatest unfounded set with respect to I *(*$Gus(D, I)$*) is the union of all sets that are unfounded with respect to* I, $Gus(D, I) = \bigcup\{A \mid unf(A, D, I)\}$.

3.9.1 Well-Founded 3-valued Models

The Well-Founded semantics of a JTMS D is defined as the least fixpoint of an operator, called V, on interpretations.

Definition 3.14 *[vGRS 88]*

- *$U(D,I)$ is defined by: $U(D,I) = \circ Gus(D,I)$.*
- *$V(D,I) = C(D,I) \cup U(D,I)$.*

Lemma 3.15 *[vGRS 88] $C(D,I)$, $U(D,I)$ and $V(D,I)$ are monotonic transformations.*

Definition 3.16 *[vGRS 88] For all countable ordinals κ the set V^κ, whose elements are literals in $N \cup \circ(N)$, is defined recursively by:*

1. *For limit ordinal κ, $V^\kappa = \cup_{\mu<\kappa} V^\mu$*
2. *For successor ordinal $\kappa+1$, $V^{\kappa+1} = V(D, V^\kappa)$*

Note that 0 is a limit ordinal, and $V^0 = \emptyset$.

Lemma 3.17 *[vGRS 88] $(V^\kappa)_{\kappa \geq 0}$ as defined in definition 3.16 is a monotonic sequence of 3-valued interpretations (i.e, I_κ is consistent for all κ).*

The set N is countable, so the sequence of definition 3.16 reaches a fixed point V^* after some countable ordinal. Now the Well-Founded semantics can be defined:

Definition 3.18 *The Well-Founded semantics of a* JTMS *D is the "meaning" represented by the limit V^* described above.*

Examples can be found in section 4. The complexity of this semantics is obviously very high, so Van Gelder [Van Gelder 89] and (independently) Przymusinski [Przymusinski 89] have proposed faster construction methods for this semantics, but with these the complexity of computing the Well-Founded model is still very high. Fortunately Witteveen proved a faster method when the program is restricted to the propositional case (as in JTMSs) based on work by Goodwin ([Goodwin 82] and [Goodwin 87]).

3.9.2 A faster method for finding the Well-Founded model for a JTMS.

In [Witteveen 90] a faster method for computing the Well-Founded model is described. This method seems not use the notion of unfounded sets, as it is computed solely using transitive closure operations.

Definition 3.19 *Let I and I' be 3-valued interpretations for* JTMS *$D = (N, J)$.*

- *$Ker(I) = I_t \cup I_f$.*
- *$I' - I = (I'_t - I_t, I'_f - I_f)$.*
- *$Un(\alpha, I)$ ($Un(\beta, I)$) is the conjunction of atoms in α (β) evaluating to u under I.*
- *$N^{-I} = N - Ker(I)$*
- *$J^{-I} = \{Un(\alpha) \wedge Un(\beta) \rightarrow c \mid \alpha \wedge \beta \rightarrow c \in J \& c \notin Ker(I) \& I(\alpha \wedge \beta) >_{tr} f\}$.*
- *$D^{-I} = (N^{-I}, J^{-I})$.*
- *$Cl^+(D,I) := Cl((D^{-I})_+, \emptyset)$*
- *$J_+ = \{\alpha \rightarrow c \mid \alpha \wedge \beta \rightarrow c \in J\}$.*

- $D_+ = (N, J_+)$
- $W(D,I) = I \cup (Cl(D,I), N^{-I} - Cl^+(D,I))$.

Theorem 3.20 *[Witteveen 90a] The least fixpoint W^* of W is the Well-Founded model of D.*

A proof of this theorem is given in [Witteveen 90], but in the next section we give an alternative proof, which shows that this method does use the notion of the Greatest Unfounded Set.
The time-complexity of W^* is O(| N | × | D |), because Dowling and Gallier [DG 84] and Minoux [Minoux 88] gave linear-time algorithms to find a least truth-assignment for a set of Horn clauses. Notice that a set of justifications without negation is a set of Horn clauses.

3.9.3 Some new facts and new proofs of old facts about the Well-Founded semantics.

In this section we will prove that $N^{-I} - Cl^+(D,I)$ is equal to $Gus(D,I)$. In the following, we will think of an interpretation I as a set of literals $I_t \cup \circ I_f$.

Lemma 3.21 $Gus(D,I) \subseteq N - (Cl^+(D,I) \cup I)$.

Proof: $Cl^+(D,I) \cup (I \cap N) = Cl((D^{-I})_+, \emptyset) \cup (I \cap N) = (\lambda X.C((D^{-I})_+, X))^\omega(\emptyset) \cup (I \cap N) = \bigcup_{i \in \omega} I_i \cup (I \cap N)$ where $I_0 = \emptyset$ and

$$
\begin{array}{ll}
 & I_{n+1} = C((D^{-I})_+, I_n) = \\
 & \quad = \{c \mid \alpha' \rightarrow c \in (J^{-I})_+, I_n \models \alpha\} \\
(*) & \quad = \{c \mid (Lit(\alpha \wedge \beta) - I) \cap N \subseteq I_n, Lit(\alpha \wedge \beta) \rightarrow c \in J, p \notin I, consistent(I, Lit(\alpha \wedge \beta))\} \\
 & \quad \text{which implies that } \neg\exists A(p \in A \wedge unf(A,D,I)).
\end{array}
$$

Suppose $p \in A$ and $unf(A,D,I)$, then $\circ A$ is consistent with $I \cup Poss(D, I \cup \circ A)$. This can be reformulated to $A \cap I = \emptyset$ and $A \cap Poss(D, I \cup \circ A) = \emptyset$. $A \cap Poss(D, I \cup \circ A) = \emptyset$ is the same as $A \cap \bigcup\{C(D,X) \mid I \cup \circ A$ and X consistent$\} = \emptyset$. So $\neg\exists X(p \in C(D,X)$ and consistent $(I \cup \circ A, X))$ which is $\forall X(\text{consistent}(I \cup \circ A, X) \rightarrow \forall(Y \rightarrow p) \in J : Y \not\subseteq X)))$.

(**) $\forall \tilde{I}((I \cup \circ A) \cap \circ \tilde{I} \neq \emptyset \vee \forall(\tilde{I}' \rightarrow p) \in J : \tilde{I}' \not\subseteq \tilde{I})$

But from (*): $\exists(\alpha \wedge \beta \rightarrow p) \in J$, where I and $Lit(\alpha \wedge \beta)$ are consistent, i.e. $I \cap \circ Lit(\alpha \wedge \beta) = \emptyset$
$(Lit(\alpha \wedge \beta) - I) \cap N \subseteq I_n$ (Ind. hyp.: $I_n \cap Gus(D,I) = \emptyset$).
From (**) with $X := Lit(\alpha \wedge \beta)$ and $\tilde{I}' := Lit(\alpha \wedge \beta)$ we can derive
$(I \cup \circ A) \cap \circ Lit(\alpha \wedge \beta) \neq \emptyset$.
To summarize:

1. $I \cap \circ Lit(\alpha \wedge \beta) = \emptyset \Leftrightarrow \circ I \cap Lit(\alpha \wedge \beta) = \emptyset$
2. $(I \cup \circ A) \cap \circ Lit(\alpha \wedge \beta) \neq \emptyset \Leftrightarrow (\circ I \cup A) \cap Lit(\alpha \wedge \beta) \neq \emptyset$
3. $(Lit(\alpha \wedge \beta) - I) \cap N \cap A = \emptyset$

From (2) we get $(\circ I \cap Lit(\alpha \wedge \beta)) \cup (A \cap Lit(\alpha \wedge \beta)) \neq \emptyset$, we combine this with (1) to $A \cap Lit(\alpha \wedge \beta) \neq \emptyset$. Furthermore, from (3) combined with $A \subseteq N$ we can conclude $A \cap Lit(\alpha \wedge \beta) - I = \emptyset$. We already had $A \cap I = \emptyset$, so $A \cap Lit(\alpha \wedge \beta) - I = A \cap Lit(\alpha \wedge \beta) = \emptyset$, contradiction.

□

Lemma 3.22 $N - (Cl^+(D,I) \cup I) \subseteq Gus(D,I)$.

Proof: Let $X = N - (Cl^+(D,I) \cup I)$. We have to show that X is an unfounded set of D with respect to I, i.e. $unf(X,D,I)$. This is true if and only if $\circ X$ is consistent with $I \cup Poss(D, I \cup \circ X)$. So $unf(X,D,I)$ if and only if $(X \cap I = \emptyset$ and $X \cap Poss(D, I \cup \circ X) = \emptyset)$. If and only if $X \cap Poss(D, I \cup \circ X) = \emptyset$. If and only if

$\odot$ $Poss(D, I \cup \circ X) \subseteq N - X$

Now $p \in Poss(D, I \cup \circ X)$ iff $\exists Y \exists \alpha \wedge \beta \rightarrow p \in J$ such that Y is consistent with $I \cup \circ X$) and $Y(\alpha \wedge \beta) = t$. So $p \in Poss(D, I \cup \circ X)$ iff ($\exists \alpha \wedge \beta \rightarrow p \in J$ such that $Lit(\alpha \wedge \beta) \cap (\circ I \cup X) = \emptyset$)
iff ($\exists (\alpha \wedge \beta \rightarrow p) \in J$ such that $Lit(\alpha \wedge \beta) \cup \circ I = \emptyset$ and $Lit(\alpha \wedge \beta) \cap X = \emptyset$)
iff

$\oplus$ $(\alpha \wedge \beta \rightarrow p) \in J$ and $Lit(\alpha \wedge \beta) \cap \circ I = \emptyset$ and $Lit(\alpha \wedge \beta) \subseteq N - X = (N \cap I) \cup Cl^+(D, I)$

We have to prove that $p \in (I \cap N) \cup Cl^+(D, I)$ in order to prove $\odot$. So we have to show $p \in I$ or $p \in Cl^+(D, I)$. If $p \in I$, we are done, so assume $p \notin I$. $p \in Cl^+(D, I)$ if $\exists j \in J$ with $hd(j) = p$ and $p \notin I$ and I consistent with $Lit(body(j))$, such that $(Lit(body(j)) - I) \cap N \subseteq Cl^+(D, I)$. Take $j = \alpha \wedge \beta \rightarrow p$, then $(Lit(\alpha \wedge \beta) - I) \cap N \subseteq Cl^+(D, I)$ because of $\oplus$.

□

Corollary 3.23 $N^{-I} - Cl^+(D, I) = Gus(D, I)$.

Theorem 3.24 *Let K and L be monotonic operators on sets. Let $A(X) = K(X) \cup L(X)$ and $B(X) = K^\omega(X) \cup L(X)$ for any set X. If K is increasing and continuous, then $A^\omega(X) = B^\omega(X)$.*

Proof: Note that

1. $K^\omega(B^\omega(X)) = B^\omega(X)$
2. $L(B^\omega(X)) \subseteq B^\omega(X)$
3. $K(A^\omega(X)) = A^\omega(X)$
4. $L(A^\omega(X)) \subseteq A^\omega(X)$
5. $K^\omega(A^\omega(X)) = A^\omega(X)$.
6. $K(B^\omega(X)) = B^\omega(X)$

Properties (1), (3), (5) and (6) hold because $A^\omega(X)$ and $B^\omega(X)$ are fixpoints of K (and therefore of K^ω). Properties (2) and (4) hold because of $B(B^\omega(X)) = K^\omega(B^\omega(X)) \cup L(B^\omega(X)) = B^\omega(X)$ and $A(A^\omega(X)) = K(A^\omega(X)) \cup L(A^\omega(X)) = A^\omega(X)$.
First we prove that $A^\omega(X) \subseteq B^\omega(X)$, then we prove that $B^\omega(X) \subseteq A^\omega(X)$.

• $A^\omega(X) \subseteq B^\omega(X)$:
We prove this by induction. The basis $A^0(X) = X = B^0(X) \subseteq B^\omega(X)$ follows from the fact that $Y \subseteq K(Y)$ for any set Y. For the induction step suppose that $Y \subseteq B^\omega(X)$ for a set Y. Then $K(Y) \subseteq K(B^\omega(X))$ because of the monotonicity of K. And $K(B^\omega(X)) \subseteq B^\omega(X)$ because of property (6). Property (2) and the monotonicity of L prove that $L(Y) \subseteq B^\omega(X)$. So we can conlude that $A(Y) \subseteq B^\omega(X)$, which concludes this proof by induction.

• $B^\omega(X) \subseteq A^\omega(X)$:
We prove this by induction. The basis $B^0(X) = X = A^0(X) \subseteq A^\omega(X)$ follows from the fact that $Y \subseteq K(Y)$ for any set Y. For the induction step suppose that $Y \subseteq A^\omega(X)$ for a set Y. Then $K^\omega(Y) \subseteq K^\omega(A^\omega(X))$ because of the monotonicity of K, which implies that K^ω is monotonic. And $K^\omega(A^\omega(X)) \subseteq A^\omega(X)$ because of property (5). Property (4) and the monotonicity of L prove that $L(Y) \subseteq A^\omega(X)$. So we can conlude that $B(Y) \subseteq A^\omega(X)$, which concludes this proof by induction.

□

Corollary 3.25 *Witteveen's method for constructing the Well-Founded model is correct; i.e. $W^* = V^*$, conform theorem 3.20.*

Proof: Recall $W^* = (\lambda I.Cl(D,I) \cup \circ Gus(D,I))^\omega(\emptyset)$. And $V^* = (\lambda I.C(J,I) \cup \circ Gus(D,I))^\omega(\emptyset)$. Theorem 3.24 is applicable, when taking $K := C$ (and $K^\omega = Cl$) and $L := Gus$, $A^\omega := V^*$ and $B^\omega := W^*$. C and Gus are monotonic, continuous operators, and C is increasing.

□

Comparing Well-Founded models with intrinsic models, the following holds:
The Well-Founded model need not be intrinsic, for example, let $D = (\{a,b,c,d\}, J)$, where $J = \{\sim a \rightarrow b, \sim c \rightarrow d, \sim c \rightarrow a, b \wedge d \rightarrow \bot\}$, then the Well-Founded model is $(\{a,d\},\{b,c\})$, but $(\{b,c\},\{a,d\})$ is also a model of J and these two models are not consistent.

Note that $Cl(D,\emptyset)$ is always an intrinsic model of J, and it is the least (i.e. knowledge-minimal) intrisic model of D.

4 Conflict resolution using contrapositions

In this section, we will discuss several methods to resolve conflicts. All of these methods are based on the use of justifications in their contrapositive directions. To solve a conflict, some choices need to be revised, i.e. a form of belief revision must be performed. When creating a model, a path of choices will be followed, leading to a contradiction if one or more choices are wrong.

The method attempts to resolve this contradiction by backtracking along our path of choices and revising some. As we want to revise our model as little as possible, we don't want to revise choices that didn't cause the contradiction; the choices to be revised are defined by the dependencies of the contradiction on these choices. Therefore, the backtracking performed is dependency-directed. Also, we only want to perform backtracking when it is mandatory, i.e. if there is no (3-valued) stable model for D. As the Well-Founded model is the knowledge-minimal 3-valued stable model, backtracking is necessary only if the Well-Founded model is inconsistent. That is why we concentrate on the Well-Founded semantics.

In section 4.1, we will first show how conflicts can arise. Subsequently, we will show the relation between belief revision and reasoning from the contradiction using justifications in their contrapositive direction. The idea of using contrapositions can be found in Dependency-directed backtracking, which will be described in short.

In sections 4.2 and 4.3, we will discuss the semantics for DDB in Justification-based Truth Maintenance Systems, which were presented by [GiMar1] and [Witteveen 91]. These different semantics depend strongly on the use of contrapositions.

4.1 Introduction

Now remember that the task of a JTMS is to maintain a model. Given a set of justifications J, we let the JTMS calculate the Well-Founded model. One of the problems arising in the presence of constraints is that it is possible that no 2-valued extension of the Well-Founded model is a 2-valued model. For example:

Example 4.1

Let $D = (N, J)$ with $N = \{a\}$ and let J be:

$$\begin{array}{ll} \sim a & \rightarrow a \\ \sim a & \rightarrow \bot \\ a & \rightarrow \bot \end{array}$$

For D the Well-Founded model is $(\emptyset,\emptyset)$, but D is inconsistent and therefore has no 2-valued model at all.

Then one can question the JTMS, for example "$WF \models p$?", or "does a model exist for D such that p (or $\sim p$) evaluates to true?". In most of these cases an alternative model must be found for D to satisfy the user, or a negative answer was provided. In most cases the question will not

be trivial; the Well-Founded model will not be the required model. This implies that the set of justifications does not give enough information to render the model sought after by the user (or problem solver). On the other hand, if the Well-Founded model contains a contradiction, then it could be the case that the set of justifications gives too much information. This is shown in the following.

Example 4.2

Let $D_1 = (N_1, J_1)$ with $N_1 = \{a, b\}$ and let J_1 be:

$$\begin{array}{ll} & \rightarrow a \\ & \rightarrow b \\ a \wedge b & \rightarrow \bot \end{array}$$

For D_1 the Well-Founded model is $(\{a, b, \bot\}, \emptyset)$, which contains a contradiction. In this case J_1 contains too much information. The problem solver (or user) will have to retract one of the justifications before a 3-valued model can be found.

Example 4.3

Let $D_2 = (N_2, J_2)$, with $N_2 = \{a\}$ and $J_2 = \{\sim a \rightarrow \bot\}$.

The Well-Founded model is $(\{\bot\}, \{a\})$ in which disbelief in a leads to belief in $\bot$. The JTMS has to perform backtracking to find a model for D_2, adding justification $\rightarrow a$ and calculating the Well-Founded model now gives $(\{a\}, \{\bot\})$. So a model does exist for J_2, but J_2 contains not enough information to find it without backtracking.

In the next example we will try to illustrate the dependencies between several 3-valued interpretations when constructing the Well-Founded model.

Example 4.4

Consider $D = (N, J)$ with $N = \{a, b, c, d\}$ and let J be:

$$\begin{array}{ll} \sim a & \rightarrow b \\ \sim c & \rightarrow d \\ b \wedge d & \rightarrow \bot \\ \sim c & \rightarrow a \end{array}$$

$WF = (\emptyset, \emptyset) \sqcup (\emptyset, \{c\}) \sqcup (\{a, d\}, \emptyset) \sqcup (\emptyset, \{b, \bot\}) \sqcup (\emptyset, \emptyset)$. Here the truth of a and d depends only on the falseness of c, and the falseness of b and $\bot$ depends only on the truth of a.

The idea to use the contrapositives of the justifications requires some remarks. The justification $a \wedge \sim b \rightarrow c$ allows c to be concluded, given that a is believed (that is, provable) and b is disbelieved (that is, not provable). Contrapositioning gives two new clauses: $a \wedge \neg c \rightarrow b$ and $\sim b \wedge \neg c \rightarrow \neg a$. where $\neg$ is the classical negation and $\sim$ a negation by default. The first new clause says that b can be concluded when a is provable and $\neg c$ is provable, while the second new clause says that it is possible to conclude $\neg a$ when b is not provable (disbelieved) and $\neg c$ is provable. Accordingly, when one wants to use a justification in its contrapositive direction, it is not sufficient to have its consequent disbelieved, instead the negation of the consequent must be provable (or the consequent cannot be consistently assumed). As this is only possible in the presence of constraints, contrapositives can only be used if constraints occur in the set of justifications.

A form of contraposition is used in dependency-directed backtracking (DDB).

Dependency-directed backtracking has been explained more elaborately by Doyle in [Doyle 79], here we only give a glossary of his work.

We need some extra definitions. Informally, an *assumption* is a node which truth-value relies on one or several nodes, which are *false* as a result of negation as failure.

Definition 4.5 *Let D be a* JTMS.

- *Atom a is a* fact *if $a \in Cl(D, \emptyset)$.*
- *Literal a is an* assumption *if a is not a fact.*
- *The* maximal foundations *of a node c are the literals of the bodies of any justification having c as its conclusion.*

When the TMS makes $\perp$ *in* (truth-value: *true*), it invokes DDB (dependency-directed backtracking) to find and remove *at least one* of the current assumptions in order to make the contradiction node *out* (truth-value: *false*). Suppose $\perp$ is *in* because of justification j, we show this relation by writing $\perp_j$.

DDB consists of several steps:

1. *Find the maximal assumptions*

   Let S be the set of maximal assumptions underlying $\perp_j$. Node $a \in S$ iff $a \in Lit(body(j))$ and a is an assumption and no node b exists such that $b \in Lit(body(j))$ with a an element of the maximal foundations of b. Let $S = \{a_1, \ldots, a_n\}$.

2. *Summarize the cause of the inconsistency.*

   If no previous backtracking attempt on j discovered S to be the set of maximal assumptions, then create a new justification: $\bigwedge_{A \in S} \rightarrow \perp$, otherwise this justification would already have been created.

3. *Select and reject a culprit.*

   Select some a_i, the culprit, from S. Let $\alpha \wedge \beta \rightarrow a_i$ be a_i's supporting justification. Select $d \in At(\beta)$, d is called the *denial*, and justify it with $a_1 \wedge \ldots \wedge a_{i-1} \wedge a_{i+1} \wedge \ldots \wedge a_n \wedge \beta^{-\{\sim d\}} \rightarrow d$. If the backtracker erred in choosing the culprit or denial, presumably a future contradiction will involve d and the remaining assumptions in β. However, if $\beta^{-\{\sim d\}} \neq \emptyset$, d will be an assumption, of higher level than the remaining assumptions, and so will be the first to be denied.

4. *Repeat if necessary.*

   If the TMS finds other arguments so that $\perp_j$ remains *in* after the addition of the new justification for d, repeat DDB. (Presumably a_i will no longer be an assumption.)

   Finally, if $\perp_j$ becomes *out* then halt, or if no assumptions can be found in $\perp_j$'s maximal foundations, notify the problem solver of an unanalyzable contradiction, then halt.

Note that in step (3) no selection criteria have been given. Several criteria are possible and give rise to the different DDB strategies.

4.2 Multiple states of belief for constraint satisfaction

Giordano and Martelli present a logical semantics for justification-based truth maintenance systems which is able to capture the idea of dependency-directed backtracking. This semantics is based on a generalization of the Stable model semantics and an active use of constraints. Of overall importance is the use of contrapositions of justifications.

In the presence of constraints, conflicts can arise. The conflict resolution process is shown mainly to rely on the intuitive idea of a contrapositive use of justifications to resolve inconsistencies. To provide a logic characterization of this contradiction resolution process, they propose a generalization of stable models.

They claim that it is reasonable to require a unique canonical model when dealing with the problem of defining a semantics for negation as failure in logic programs. But they say: "On the

contrary, when reasoning on the possible states of beliefs supported by a given set of justifications, multiple states of beliefs make perfectly sense and, thus, also sets of justifications with multiple stable models."

We do agree with their opinion in part, but not when dealing with *justification*-based truth maintenance, which has to deal with frequent non-monotonic reasoning. Then as said before it is too costly to maintain several possible states of beliefs (models). Giordano and Martelli note that it is possible to adapt the use of stable models to the situation when constraints are present in the set of justifications, by simply computing the stable models of the set of general clauses in the set of justifications, those that are no constraint, and then, later on, eliminating the stable models that do not satisfy the constraints (that is, the inconsistent stable models). In this way, constaints are simply used to cut out some inconsistent labellings and no belief revision is performed. Problems arise when for a set of justifications J no stable model exists which satisfies all constraints, then backtracking must be performed. However, since the Well-Founded model of the set of justifications J is the least 3-valued stable model of J, computing the Well-Founded model, directly tells whether or not backtracking is required.

In order to give a logical characterization for the TMS performing dependency-directed backtracking, Giordano and Martelli define a generalization of the notion of *stable model*, with this generalization they want to use constraints actively to perform belief revision. In this definition they take care of the fact that they need to use justifications also in their contrapositive form.

Let M be an interpretation in the classical propositional calculus. M can be regarded as a subset of the propositions occurring in J, those true in the interpretation. Let $\mathcal{J}'_M$ be the set of justifications obtained from J by deleting from the body of each justification all the literals $\sim b$ for which $b \notin M$, i.e. $\mathcal{J}'_M = \{\alpha \wedge \beta_1 \rightarrow c \mid \alpha \wedge \beta \rightarrow c \in J \text{ and } \beta = \beta_1 \wedge \beta_2 \text{ and } M \models \beta_2\}$. Now write each justification $a_1 \wedge \ldots \wedge a_n \wedge \sim b_1 \wedge \ldots \wedge \sim b_h \rightarrow c$ in $\mathcal{J}'_M$ as the disjunction of literals $\neg a_1 \vee \ldots \neg a_n \vee b_1 \vee \ldots \vee b_h \vee c$. Let $\mathcal{J}_M$ be the set justifications in $\mathcal{J}'_M$ having one and only one literal true in M.

Definition 4.6 *[GiMar1] Let M be a set of propositions occurring in J. M is a* generalized stable model *of J if M is a model of J and $M = \{a \mid a \text{ proposition in } J \text{ and } \mathcal{J}_M \models a\}$.*

Notice that, while negation in the initial set J, $\sim$, is a default negation, negation occurring in $\mathcal{J}_M$ is the classical one and $\models$ is the logical consequence in classical logic.

As the clauses in $\mathcal{J}_M$ have only one literal true in M, each of these clauses can be regarded as an implication having as its consequent that single literal which is true in model M. Accordingly, all the literals occurring in its body will also be true in M. This implies that $\mathcal{J}_M$ will not contain constraints, since each constraint containing a unique literal true in M occurs in $\mathcal{J}_M$ as an implication with that literal as its head, while other constraints are deleted. Thus negation in the head or body of the clauses of $\mathcal{J}_M$ is always interpreted as *classical negation.*

Notice that a set J of justifications can have several generalized stable models. On the other hand in the case that J is inconsistent there should not be a generalized stable model. J can be inconsistent if J contains constraints, however, since $\mathcal{J}_M$ is always consistent, it can happen that there is an interpretation M satisfying $M = \{a \mid a \text{ is a proposition in } J \text{ and } \mathcal{J}_M \models a\}$, even if J is inconsistent. This is the reason that M is also required to be a model of J.

If there is a unique generalized stable model of J, then it precisely corresponds to the labelling computed by the TMS on backtracking. However sets of justifications can be found for which several generalized stable models exist, examples can be found in [GiMar1], so using this method, the TMS seems to lose its merit of finding only the preferred solution. However Doyle, in his original paper [Doyle 79], desribes the behavior of the TMS *incrementally*, this incremental process can obtain solutions different from those expected in the static approach. This is due to the different methods for backtracking, the generalized stable models correspond to all situations which can be obtained by giving to the TMS the justifications in any order.

4.3 Single state of Belief for all possible belief revisions

Using contrapositions, classical negation appears in the head and in the body of justifications. Both occurences are troublesome, as classical negation must be proved and negation is not allowed in the head of justifications. Therefore, the classical negation must be circumvented by adding a new atom a^- to stand for $\neg a$. To preserve the meaning of such negated atoms, some justifications and constraints must be added:

$$a \wedge a^- \rightarrow \bot$$

This ensures that not both a and a^- can be true. But we also have to ensure that not both a and a^- can be false. This can be done by adding:

$$\sim a \rightarrow a^- \text{ and } \sim a^- \rightarrow a$$

Now define the set $Back(J)$ of backward justifications as follows:

Definition 4.7 *Let D be a* JTMS.

- $E(a^-) = \{a \wedge a^- \rightarrow \bot, \sim a \rightarrow a^-, \sim a^- \rightarrow a\}$.
- $Back(j,\alpha) = \{\alpha' \wedge c^- \wedge \beta \wedge \sim a \rightarrow a^- \mid \alpha' = \alpha^{-\{a\}}, a \in \alpha\}$.
- $Back(j,\beta) = \{\alpha \wedge c^- \wedge \beta' \wedge \sim b^- \rightarrow b \mid \beta' = \beta^{-\{\sim b\}}, b \in At(\beta)\}$.
- $N^-(j) = \{a^- \mid a^- \in At(Back(j,\alpha) \cup Back(j,\beta))\}$.
- *Let $Anc(c)$ be the set of* direct ancestors *of $c \in N$: $Anc(c) = \{a \mid \exists j \in J : a \in body(j)$ and $hd(j) = c\}$.*
- *Let $Anc^*(c)$ be the transitive closure of $Anc(c)$.*
- $N^- = \bigcup\{N^-(j) \mid j \in J$ *and* $hd(j) \in Anc^*(\bot)\}$
- $Back^-(j) = \bigcup\{E(a^-) \mid a^- \in N^-\}$.
- $Back(j) = Back(j,\alpha) \cup Back(j,\beta) \cup Back^-(j)$.

It is not necessary to create backward justifications for all $j \in J$. It suffices to take those justifications whose (in)direct consequence is $\bot$.

Definition 4.8 *Let D be a* JTMS. $B(J) = \bigcup\{Back(j) \mid j \in J$ *and* $hd(j) \in Anc^*(\bot)\}$. *Every subset of $B(J)$ is called a* DDB-*strategy.*

Theorem 4.9 *[Witteveen 91] $WF(B(J))$ is the knowledge-minimal 3-valued stable model for the class of* DDB-*strategies.*

5 Conflict resolution using one disjunctive justification

We present two semantics for DDB which avoid the use of contraposition. The first gives the same model as proposed by [Witteveen 91] in a more efficient and natural way. The second is an extension of the first in which more information is obtained from the contradiction.

The techniques used are variations of a method that will be called Dependency-directed Minimal Backtracking (DDMB). In the following we will use WF_D to denote the Well-Founded model for JTMS D.

5.1 Introduction

There is another way to make use of the dependencies among beliefs. Given a model, some of the justifications denote the actual dependencies between truth values of propositions. These are the justifications whose body is evaluated true under the given model. In the Well-Founded model, these dependencies can be seen in the layered structure of the model.

Consider a JTMS D with constraints, for D we calculate the Well-Founded model according to Witteveen. But now we are not only interested in the resulting 3-valued model, but also in the several 3-valued interpretations from which this model has been comprised. These interpretations contain some of the information regarding the dependencies between the elements of N.

Definition 5.1 *Let* $I_D^0 = (\emptyset, \emptyset)$, $I_D^{i+1} = W(D, I_D^i)$ *and let* I_D^* *be the least fixpoint of* W.

So the Well-founded model is given by:

$$WF_D = (\bigcup_{0 \leq i \leq *} I_{D,t}^i, \bigcup_{0 \leq i \leq *} I_{D,f}^i)$$

From now on we will discard the subscript D, unless there can be doubt about which TMS is considered.

Note that I_t^1 is the least fixpoint of the immediate consequence operator applied on the positive part of D, i.e. the justifications without negations. I_f^1 contains the atoms from the greatest unfounded set (Gus) with respect to I^0. Analogously, I_t^{i+1} is the least fixpoint of the immediate consequence operator applied on the positive part of $D^{-\tilde{I}^i}$ and I_f^{i+1} contains the atoms from the Gus with respect to I^i. Given the nature of the immediate consequence operator, each of the elements of I_t^{i+1} were justified by a justification $\alpha \wedge \beta \rightarrow c$ which had this element as its head and for which at least one literal in β was justified. In the same way the elements of I_f^{i+1} depend directly on I_t^i and maybe indirectly on several other I^k ($k < i$). And the elements of I_t^{i+1} depend directly on I_f^i, maybe indirectly on several other I^k ($k < i$).

Definition 5.2 *Let* $\tilde{I}_i = \bigcup_{l \leq i} I^l$ *for all* $i \geq 0$.

Lemma 5.3 *If* $I_f^i = \emptyset$ *then* $I_t^{i+1} = \emptyset$.

Proof: Let $J' = J^{-\tilde{I}_i}$. Suppose $I_f^i = \emptyset$ but $c \in I_t^{i+1}$. Without loss of generality assume that c is a fact in J'. Known is that $c \notin I_t^i$, which implies that the body of all justifications with conclusion c in J' (of which there is at least one) are evaluated to u under interpretation I^i ($I^i(\alpha \wedge \beta) = u$) while the positive part of this body (α) is evaluated to t under this interpretation ($I^i(\alpha) = t$). (Remember that $At(\beta) \neq \emptyset$ otherwise $c \in I_t^i$ and that $I_f^i = \emptyset$.) But $\rightarrow c \in J'$ implies that $At(\beta) \in I_f^i$.

□

How can the knowledge of these dependencies be used to efficiently compute a new Well-Founded model for an altered J? Suppose that an atom $p \in N$ is in WF_f for a certain set of justifications J and the problem solver would rather have $p \in WF_t$, it can add a justification to J like $\rightarrow p$, then the JTMS will have to adjust the Well-Founded model to encompass this new justification. The only way p can be made false is by negation by failure, unless a constraint is present in J with p in its body. So in most cases adding $\rightarrow p$ will help. Suppose the JTMS knows (by keeping track of the list of interpretations I^i) that $p \in I_f^i$, then the only nodes that could be affected by p's change of truth-value are those not in $\tilde{I}_{i-1}$. So we can recompute the Well-Founded model starting at I^i. The only problem that could occur is that one of the constraints could be violated by this Well-Founded model, then the JTMS must perform backtracking. This will be possible unless the body of this constraint consists entirely of facts (positive literals all of which are element of I_t^1). More will be said about backtracking later on.

On the other hand suppose that $p \in N$ is in WF_t and the problem solver would rather have $p \in WF_f$, then it is not allowed to add $\rightarrow \neg p$ to J, but it can add a constraint $p \rightarrow \perp$. However

there are other possibilities, by looking at the interdependencies of the atoms, it is sometimes possible to recalculate a part of the Well-Founded model. First the JTMS has to check whether or not $p \in WF_f$ would contradict $p \in I_t^1$, if this is not the case, then (if no constraints are present) it is possible to recalculate the Well-Founded model from I^i, where $p \in I_t^{i+1}$. Because p directly depends on the atoms in I_f^i. The JTMS acts as if one of the elements of I_f^i occurring in a justification with head p where added as fact to J and continues calculating the Well-Founded model at I^i. Picking only on of the elements of I_f^i as a fact to be added, corresponds with a specific dependency-directed backtracking method.

Making no selection corresponds to the DDMB methods.
In the following we will frequently refer to DDB-*strategy*, which will be defined as:

Definition 5.4 *A* DDB-strategy R *is a belief revision strategy, which adds to* J *a collection of contrapositions of justifications from* J.

5.2 Dependency-Directed Minimal Backtracking

We can avoid using contrapositions by noting that only *assumptions* can be responsible for the contradiction (otherwise D would not have any model). Assumptions are created by the use of negation as failure and apparently a number of atoms was injustly considered *false*, leading to the contradiction. This set of atoms can be characterized as follows:

Definition 5.5 *Atom* b *belongs to the foundations of* c *in interpretation* M $(b \in Gr(c, M))$ *if* $\exists \alpha \wedge \beta \rightarrow c \in J$ *such that* $M(\alpha \wedge \beta) = t$ *and at least one of the following holds:*

1. $\sim b \in Lit(\beta)$.
2. $a \in Lit(\alpha)$ *and* $b \in Gr(a, M)$.

Instead of reasoning in the contrapositive direction, one can also search for those negative atoms, that belong to the foundations of $\bot$, i.e. $Gr(\bot.M)$.

The approach described below actually comprises two different methods, DDMB_1 and DDMB_2. For DDMB_1 we prove that it corresponds with $B(J)$ as defined in 4.3. In DDMB_2 the added justification can be a fact, leading to a more intrinsic model. In every step of the method described below we will denote how the proposed methods differ.

MDDB-method

Suppose WF is an inconsistent interpretation for D and suppose $\bot$ is not a fact (there exists a model for D). Let constraint δ (consequence of δ is $\bot$) be the justification which caused WF to be inconsistent.

1. *Determine* $Gr(\bot_\delta, WF)$.
   This set can easily be found in WF, by looking at the dependencies.
2. For DDMB_1 add to J the following (disjunctive) justification j:

$$\bigwedge \circ Gr(\bot_\delta, WF) \rightarrow \bigvee Gr(\bot_\delta, WF)$$

3. For DDMB_2 add to J the following (disjunctive) justification j:

$$\rightarrow \bigvee Gr(\bot_\delta, WF)$$

The more intrinsic method (DDMB_2) renders a set of justifications for which the Disjunctive Well-Founded model can be computed, which extends the skeptical model for DDMB_1.

5.2.1 Skeptical model of disjunctive logic programs

Here we define a model for disjunctive logic programs which contains less information than the Well-Founded models (Weak and Strong) for disjunctive logic programs as defined by Ross in [Ross 89b]. This skeptical model is related to the Well-Founded model for normal logic programs. The reason we define such a skeptical model (Sk) is that we can prove that for all elements $a \in N$ $Sk(\mathsf{DDMB}_1)(a) = WF(B(J))(a)$, where $B(J)$ is the set of justifications defined in 4.3.

Definition 5.6

- *A disjunctive justification has the format: $\alpha \wedge \beta \rightarrow \delta$, where $\alpha \wedge \beta$ is as before and δ is a disjunction of atoms.*
- *Let* DJTMS *stand for a* JTMS *containing disjunctive justifications.*

Definition 5.7 *Let I be an interpretation and let D be a* DJTMS.

- *$Normal(D)$ is the part of D not containing disjunctive justifications.*
- *The Transitive Closure DCl of D is the transitive closure of $Normal(D)$, i.e. $DCl(D, I) = Cl(Normal(D), I)$.*

Note that if δ contains more than one atom, then $I(\delta) = t$ will not be used to derive extra information, since δ does not occur in the body of any justification. In the Well-Founded semantics as defined by Ross extra information can sometimes be derived. The definition of an *unfounded set* is only slightly adjusted:

Definition 5.8 *Given a* JTMS *D with disjunctive justifications and a 3-valued interpretation I. Then we say that $A \subseteq N$ is an* unfounded set of D with respect to I *(denoted $unf(A, D, I)$) if for each atom $c \in A$ the following holds: For each justification $j \in J$ in whose head p occurs ($p \in At(\delta)$), at least one of the following holds: $I(\alpha \wedge \beta) = f$ or $At(\alpha) \cap A \neq \emptyset$.*

Definition 5.9 *The Skeptical model $Sk(D)$ of a* DJTMS *D is the least fixpoint of $\lambda I.(DCl(D, I) \cup \circ DGus(D, I))$.*

We extend the Skeptical model to an interpretation model for all atoms in $WF(B(J))$ and prove the extension to be a stable model for $B(J)$.

Definition 5.10 *Let M be the interpretation defined by:*

$$M(x) = \begin{cases} Sk(\mathsf{DDMB}_1)(x) \text{ if } x \in N \\ \sim Sk(\mathsf{DDMB}_1)(y) \text{ if } x = y^- \in N^- \end{cases}$$

Lemma 5.11 *M is a model of $B(J)$.*

Proof: Let $x \in N \cup N^-$ and suppose there is a justification $j \in B(J)$ with head x such that $M(x) <_{tr} M(body(j))$. As M is a model of J, $j \notin J$. Hence $At(j) \cap N^- \neq \emptyset$. For x we have two possibilities: $x \in N$ or $x \in N^-$.
• $x \in N^-$ then $x = y^-$, $y \in N$, this implies that $\sim y \in Lit(body(j))$. So $M(body(j)) \leq M(\sim y) = \sim M(y) = M(y^-) = M(x)$. And M satisfies j.
• $x \in N$. Remember $j \notin J$ so there is a $y^- \in At(j)$. Apparently $j \in Back(\alpha \wedge \beta \wedge \rightarrow y, \beta)$. This implies that $\sim x^- \in Lit(body(j))$. We have to check each of the three possible truth-values for x in M.
If $M(x) = t$, then always $M(body(j)) \leq_{tr} M(x)$, so M satisfies j.
If $M(x) = u$, then $M(\sim x^-) = u$. As $M(body(j)) = min_{tr}\{M(l) \mid l \in Lit(body(j))\}$, $M(body(j)) \leq_{tr} M(\sim x^-) = u$. So m satisfies j.
If $M(x) = f$, then $M(\sim x^-) = f$ as well. Therefore, $M(body(j)) = f$ and M satisfies j.

□

Lemma 5.12 *M is a 3-valued stable model of $B(J)$.*

Proof: Let Min_a be the truth-minimal 2-valued model of $(N \cup N^-, B(J))(M, t)$. Let Min_b be the truth-minimal 2-valued model of $(N \cup N^-, B(J))(M, u)$. We have to prove that (see definition 3.6)

(a) $M_t = Min_a$ and

(b) $M_t \cup M_u = Min_b$.

a: Suppose $M_t \neq Min_a$, then there is a $x \in N \cup N^-$ such that

i. $x \in M_t$ but $x \notin Min_a$ or

ii. $x \in Min_a$ but $x \notin M_t$.

i:

• If $x \in N$, then $M(x) = t$ implies that $x \notin Anc^*(\bot)$, therefore $B(J) - J$ contains no justification with conclusion x. So $Min_a(x) = t$ as well.

• If $x = y^-$, then $M(x) = t$ implies that $M(y) = f$. Therefore no subset δ of N exists such that $y \in \delta$ and such that $\delta = hd(j)$ with $j \in \mathsf{DDMB}_1(J)(J) - J$. Apparently $y \notin Anc^*(\bot)$, but this contradicts $x = y^-$.

ii:

• Suppose $x \in N$, then $x \in Min_a$ implies that there exists a justification j' in $(N \cup N^-, B(J))(M, t)$ such that $Min_a(body(j')) = t$ and such that $Hd(j') = x$. Associated with j' must be a justification $j \in B(J)$. This justification j is a backward justification or $j \in E(x^-)$ or $j \in J$. In the first two cases $\sim x^-$ occurs in the body of j. As $j' \in (N \cup N^-, B(J))(M, T)$, we know that $M(\sim x^-) = t$. This implies that $M(x^-) = f$, but then $M(x)$ must be t.

So $j \in J$ and $At(\alpha(j)) \subseteq Min_a$. So $\forall a \in At(\alpha(j))$ a monotonic proof for a exists in $(N \cup N^-, B(J))(M, T)$, which begins with facts $\rightarrow c$. This implies that $\forall a \in At(\alpha(j))$ a non-monotonic proof exists in J, which begins with justifications $\beta \rightarrow c$, for which $M(\beta) = t$. This implies that $M(body(j)) = t$ and $M(x) = t$.

• Suppose $x \in N^-$, then there must be a justification $j = \alpha \wedge \beta \rightarrow x \in B(J)$ such that $M(\beta) = t$ and $Min_a(\alpha) = t$. Now $j \notin J$, so $\sim y \in Lit(\beta)$. Therefore $M(\sim y) = t$ and $M(y^-) = t$. Conclusion $M(x) = t$.

So $M_t = Min_a$ cannot be contradicted.

b: Suppose $M_t \cup M_u \neq Min_b$, then there is a $x \in N \cup N^-$ such that

i. $x \in M_f$ & $x \in Min_b$ & $x \in N$, or

ii. $x \in M_f$ & $x \in Min_b$ & $x \in N^-$, or

iii. $x \notin M_f$ & $x \notin Min_b$ & $x \in N$, or

iv. $x \notin M_f$ & $x \notin Min_b$ & $x \in N^-$.

i: $x \in Min_b$ implies that there is a justification $j = \alpha \wedge \beta \rightarrow x \in B(J)$ such that $M(\beta) \neq f$ and $Min_b(\alpha) = t$. If $j \notin J$, then $\sim x^- \in Lit(\beta)$, so $M(\sim x^-) \neq f$. Therefore $M(x^-) \neq t$ and $M(x) \neq f$. Contradiction, so $j \in J$. Then $M(\alpha) = f$ and $M(\beta) >_{tr} f$. $Min_b(\alpha) = t$ if and only if $\forall a \in At(\alpha)$ a monotonic proof for a exists, which begins with facts $\rightarrow c$. If and only if $\forall a \in At(\alpha)$ a non-monotonic proof for a exists, beginning with justifications $\beta' \rightarrow c$, such that $M(\beta') >_{tr} f$ (and so $M(c) >_{tr} f$. Therefore $M(\alpha \wedge \beta) >_{tr} f$ and $M(x) >_{tr} f$.

ii:

As in i a justification $j = \alpha \wedge \beta \rightarrow x \in B(J)$ such that $M(\beta) \neq f$ and $Min_b(\alpha) = t$. But now $j \notin J$. Let $x = y^-$, then $\sim y \in Lit(\beta)$. So $M(\sim y) \neq f$, implying that $M(y^-) \neq f$.

iii: The only case not properly treated earlier is: $x \in M_u$ & $x \in Anc^*$. This means that $\forall j = \alpha \wedge \beta \rightarrow x \in B(J)$ holds that if $M(\beta) \neq f$ then $Min_b(\alpha) = f$. Consider $\sim x^- \rightarrow x$, this justification

is in $B(J)$ and $M(\sim x^-) = \sim M(x^-) = M(x) = u$. So $\rightarrow x \in (N \cup N^-, B(J))(M, u)$. Therefore $Min_b(x) = t$, contradiction.
iv: Let $x = y^-$. In this case $y \in Anc^*(\bot)$. Consider $\sim y \rightarrow y^-$ which is in $B(J)$. As $M(y^-) \neq f$, $M(\sim y) \neq f$ as well. Therefore $\rightarrow y^- \in (N \cup N^-, B(J))(M, U)$. Conclusion: $Min_b(y^-) = Min_b(x) = t$.

Conclusion: $M_t \cup M_u = Min_b$ cannot be contradicted. Therefore M is a stable model of $B(J)$.

□

Corollary 5.13 $\forall a \in N\ Sk(\mathsf{DDMB}_1)(a) \geq_{kn} WF(B(J))(a)$.

Lemma 5.14 *$WF(B(J))$ is a model for* $\mathsf{DDMB}_1(J)$.

Proof:

Suppose not, then there is a $\delta \subseteq N$ and a justification $j \in \mathsf{DDMB}_1(J)$ such that $hd(j) = \delta$ and $WF(B(J)(\delta) \leq_{tr} WF(B(J))(body(j))$. As $WF(B(J))$ is a model of J, $j \notin J$.

If δ consists of one atom c, then $c \in Anc^*(\bot)$, so $WF(B(J))(c) \neq f$. Therefore $WF(B(J))(\sim c) \neq t$, contradiction.

So δ is a disjunction of atoms and $j = \delta$. $WF(B(J))$ is not a model of $\mathsf{DDMB}_1(J)$ implies that $WF(B(J))(\delta) <_{tr} u$. $WF(B(J))(\delta) <_{tr} u$ if and only if $\forall x \in At(\delta) : WF(B(J))(x) = f$.

If and only if $WF(B(J))(\bot) = t$, contradiction.

□

Lemma 5.15 $\forall a \in N\ Sk(\mathsf{DDMB}_1)(a) \leq_{kn} WF(B(J))(a)$.

Proof:

Here it is essential that $\rightarrow d$ was replaced by $\sim d \rightarrow d$. Suppose the lemma is not true, then an $a \in N$ exists such that $WF(B(J))(a) <_{kn} Sk(\mathsf{DDMB}_1(J)(a)$. The only cases to be considered are, $Sk(\mathsf{DDMB}_1(J)(a) = t$ and $Sk(\mathsf{DDMB}_1(J)(a) = f$, while in both cases $WF(B(J))(a) = u$.

- $WF(B(J))(a) = u$ and $SK(a) = t$.

$WF(B(J))(a) = u$ iff $\exists j \in B(J)$ with $hd(j) = a$ and $WF(B(J))(body(j)) = u$, while $\forall j \in B(J)$ holds that $hd(j) = a$ implies $WF(B(J))(body(j)) <_{tr} t$. Consider $Sk(\mathsf{DDMB}_1(J)$: $Sk(\mathsf{DDMB}_1(J)(a) = t$ iff $\exists j \in \mathsf{DDMB}_1(J) : hd(j) = a$ and $Sk(\mathsf{DDMB}_1(J)(body(j)) = t$. For this j there is only one possibility: $j \in J$, for $\mathsf{DDMB}_1(J) - J$ is not fact. Furthermore a must be an assumption, otherwise $WF(B(J))(a) = t$ as well.

Apparently $\exists \sim b \in Anc^*(a)$ such that $Sk(\mathsf{DDMB}_1(J)(b) = f$ and $WF(B(J))(b) >_{tr} f$. Iff $\exists j' \in B(J) - J$ with $hd(j') = b$ such that $WF(B(J))(body(j')) >_{tr} f$. Iff $\sim b \in Anc^*(\bot)$. This implies that $b \in Gr(\bot, M)$, therefore $\exists j =\!\!\rightarrow \delta \in \mathsf{DDMB}_1(J) - J$ with $b \in At(\delta)$. This implies that $SK(b) = u$, contradiction.

- $WF(B(J))(a) = u$ and $SK(a) = f$.

$SK(a) = f$ implies there is no $j \in \mathsf{DDMB}_1(J) - J$ with $a \in At(hd(j))$, but on the other hand $a \in Anc^*(\bot)$ (otherwise $WF(B(J))(a) = Sk(\mathsf{DDMB}_1(J)(a))$. Furthermore $\forall j \in J$ holds that if $hd(j) = a$ then $Sk(\mathsf{DDMB}_1(J)(body(j)) = f$. So there must exist at least one $c \in At(hd(j))$, with $j \in \mathsf{DDMB}_1(J) - J$, such that $\sim c \in Anc^*(a)$ and such that $Sk(\mathsf{DDMB}_1(J)(c) = t$. But $\forall c \in At(hd(j))$ with $j \in \mathsf{DDMB}_1(J) - J$ holds that $Sk(\mathsf{DDMB}_1(J)(c) = u$, contradiction.

□

Theorem 5.16 *$Sk(\mathsf{DDMB}_1(D))$ is the least 3-valued Stable model for the class of all* DDB*-strategies.*

Proof: As can be seen from theorem 4.9, it is sufficient to prove: For all $a \in N\ Sk(\mathsf{DDMB}_1)(a) = WF(B(J))(a)$. Then the proof is direct from corollary 5.13 and lemma 5.15.

□

5.2.2 On the Semantics of DDMB_2

In the DDMB approach, we had to add $\sim c \rightarrow c$ instead of $\rightarrow c$ to the set of justifications, otherwise the Skeptical model would have contained more information than $WF(B(J))$.

In DDMB_2 we add $\rightarrow c$ and not $\sim c \rightarrow c$, because disbelief in c was the only cause of the contradiction. Apparently the intrinsic value of c is t. This implies that the Skeptical model for DDMB_2 extends the Skeptical model for DDMB_1.

Furthermore we propose to use the Well-Founded model for disjunctive logic programs ([Ross 89b]) as the semantics of DDMB_2. The Well-Founded model for disjunctive logic programs is capable of making additional intrinsic choices.

A detailed example of the different approaches and semantics will be given in section 6. But here we will give an example of the difference between DDMB_1 and DDMB_2, which concerns the inconsistency of D.

Example 5.17

Let $D = (N, J)$ with $N = \{a, b, c\}$ and let J be:

$$\begin{array}{ll} \sim c & \rightarrow b \\ \sim c & \rightarrow a \\ \sim b & \rightarrow a \\ a & \rightarrow \bot \end{array}$$

For D the Well-Founded model is $(\emptyset, \{c\}) \sqcup (\{b, a, \bot\}, \emptyset)$, the contradiction seems to depend on the falsity of c. So in the DDMB_2 method $\rightarrow c$ is add to J. Recomputing the Well-Founded model gives $(\{c\}, \emptyset) \sqcup (\emptyset, \{b\}) \sqcup (\{a, \bot\}, \emptyset)$. Again the contradiction can be traced to c, so J is inconsistent. The DDMB_1 method would add $\sim c \rightarrow c$ to J, for which the Well-Founded model is $(\emptyset, \emptyset)$. This is a 3-valued model, so the inconsistency of J will not be noticed.

6 An Example

In this example we give a set of justifications on which the different approaches can and will be applied. The corresponding semantics will be given as well.

References

[ABW 86] K.R. APT, H. Blair, A. Walker, *Towards a Theory of Declarative Knowledge.* Workshop on Foundations of Deductive Databases and Logic Programming, J. Minker (ed), Washington, DC, August 1986, pp. 546-628.

[Belnap 77] N. D. BELNAP, *A useful four-valued logic.* Modern Uses of Multiple-Valued Logic, J. M. Dunn and G. Epstein (eds.), D. Reidel, 1977, pp. 8-37.

[Clark 78] K. L. CLARK, *Negation as Failure.* Logic and Data Bases, H. Gallaire and J. Minker (eds.), Plenum Press, New York, 1978, pp. 293-322.

[DG 84] W. DOWLING, J. Gallier, *Linear-Time Algorithms for Testing the Satisfiability of Propositional Horn Formulae.* Journal of Logic Programming 3, 1984, pp. 267-284.

[Doyle 79] JON DOYLE, *A truth maintenance system.* Artificial Intelligence, 12, 1979, pp.231-272.

[EMR 87] DAVID W. ETHERINGTON, Robert E. Mercer, Raymond Reiter, *On the adequacy of predicate circumscription for closed world reasoning.* Readings in Nonmonotonic Reasoning, Morgan Kaufmann Publ., Los Altos, 1987, pp. 174-178.

[Fitting 89] MELVIN FITTING, *Bilattices and the Theory of Truth.* Journal of Philosophical Logic 18, Kluwer Academic Publishers, 1989, pp. 225-256.

[FBJ 88] MELVIN FITTING, Marion Ben-Jacob, *Stratified and Three-valued Logic Programming Semantics.* Fifth Int'l Conf. Symposium on Logic Programming, R. Kowalski and K. Bowen (eds.), Seattle, 1988, pp. 1054-1069.

[Van Gelder 86] ALLEN VAN GELDER, *Negation as Failure Using Tight Derivations for General Logic Programs.* Proc. Third IEEE Symposium on Logic Programming, Springer-Verlag, Salt Lake City, Utah, September 1986. Also in: Foundations of Deductive Databases and Logic Programming, J. Minker (ed.), Morgan Kaufmann, Los Altos, CA, 1987, pp. 177-192.

[Van Gelder 89] ALLEN VAN GELDER, *The Alternating Fixpoint of Logic Programs with Negation.* Proceedings of the Symposium on Principles of Database Systems, ACM SIGACT-SIGMOD, 1989, pp. 1-10.

[vGRS 88] ALLEN VAN GELDER, Kenneth Ross, John Schlipf, *Unfounded Sets and Well-Founded Semantics for General Logic Programs.* Proceedings of the Seventh ACM Symposium on Principles of Database Systems, 1988, pp. 221-230.

[GL 88] MICHAEL GELFOND, Vladimir Lifschitz, *The Stable Model Semantics For Logic Programming.* Fifth Int'l Conf. Symposium on Logic Programming, R. Kowalski and K. Bowen (eds.), Seattle, 1988, pp. 1070-1080.

[Ginsberg 87] MATTHEW L. GINSBERG, *Multi-valued logics.* Readings in Nonmonotonic Reasoning, Morgan Kaufmann Publ., Los Altos, 1987, pp. 251-255.

[GiMar1] LAURA GIORDANO, Alberto Martelli, *Generalized Stable Models, Truth Maintenance and Conflict Resolution.* Proc. of the 7th International Conference on Logic Programming, D. Warren and P. Szeredi (eds.), 1990, pp. 427-441.

[Goodwin 82] J. GOODWIN, *An Improved Algorithm for Non-Monotonic Dependency Net Update.* LITH-MAT-R-82-83, Linkoeping University 1982.

[Goodwin 87] J. GOODWIN, *A Theory and System for Non-Monotonic Reasoning.* Linkoeping Studies in Science and Technology, Dissertations no 165, 1987.

[KM] A. KAKAS, P. Mancarella, *Generalized Stable Models: A Semantics for Abduction.*

[KM 90] A. KAKAS, P. Mancarella, *Abductive Logic Programming.* Proc. of the Workshop Logic Programming and Non-Monotonic Logic, 1990, pp. 49-61.

[Konolidge 87] KURT KONOLIDGE, *On the relation between default and autoepistemic logic.* Readings in Nonmonotonic Reasoning, M. L. Ginsberg (ed.), Morgan Kaufmann Publ., Los Altos, 1987, pp. 195-226.

[Lifschitz 85] V. LIFSCHITZ, *Computing circumscription.* Proceedings 9*th* IJCAI, 1985, pp. 121-127.

[Lifschitz 86] V. LIFSCHITZ, *Pointwise circumscription.* Proceedings of AAAI, Philadelphia, PA, 1986, pp. 406-410. ALso in: Readings in Nonmonotonic Reasoning, M. L. Ginsberg (ed.), Morgan Kaufmann Publ., Los Altos, 1987, pp. 195-226.

[Lloyd 87] J. W. LLOYD, *Foundations of Logic Programming.* Springer-Verlag, New York, 2nd edition, 1987.

[McCarthy 80] J. M. MCCARTHY, *Circumscription - a form of non-monotonic reasoning.* Artificial Intelligence, 13, 1980, pp. 27-39. Also in: Readings in Nonmonotonic Reasoning, M. L. Ginsberg (ed.), Morgan Kaufmann Publ., Los Altos, 1987, pp. 145-152.

[McCarthy 84] J. M. MCCARTHY, *Applications of circumscription to formalizing commonsense knowledge.* Proceedings of the Nonmonotonic Reasoning Workshop, 1984, pp. 295-324. Also in: Artif. Intell. 28, 1986, pp. 89-116. And in: Readings in Nonmonotonic Reasoning, M. L. Ginsberg (ed.), Morgan Kaufmann Publ., Los Altos, 1987, pp. 153-166.

[Minoux 88] M. MINOUX, *LTUR: A Simplified Linear-Time Unit Resolution Algorithm for Horn Formulae and Computer Implementation.* Information Processing Letters 29, 1988, pp.1-12.

[Moore 83] ROBERT C. MOORE, *Semantical Considerations on Nonmonotonic Logic.* Proceedings of the 8th IJCAI, 1983, pp. 272-279. Also in: Artif. Intell. 25, 1985, pp. 75-94. And in: Readings in Nonmonotonic Reasoning, M. L. Ginsberg (ed.), Morgan Kaufmann Publ., Los Altos, 1987, pp. 127-136.

[PaPi 88] HALINA PRZYMUSINSKA, Teodor Przymusinski, *Weakly Perfect Model Semantics For Logic Programs.* Fifth Int'l Conf. Symposium on Logic Programming, R. Kowalski and K. Bowen (eds.), Seattle, 1988, pp. 1106-1120.

[PaPi 90b] HALINA PRZYMUSINSKA, Teodor Przymusinski, *Semantic Issues in Deductive Databases and Logic Programs.* Formal Techniques in Artificial Intelligence, A Sourcebook, R. B. Banerji (ed.), Elsevier, Amsterdam, 1990, pp. 321-367.

[Przymusinski 89] TEODOR PRZYMUSINSKI, *Every Logic Program Has a Natural Stratification And an Iterated Least Fixed Point Model.* Eighth ACM Symposium on Principles of Database Systems, 1989, pp. 11-21.

[Przymusinski5] TEODOR PRZYMUSINSKI, *Perfect Model Semantics.*

[Przymusinski 87] TEODOR PRZYMUSINSKI, *On the Declarative Semantics of Deductive Databases and Logic Programs.* Foundations of Deductive Databases and Logic Programming, J. Minker (ed.), Morgan Kaufmann, Los Altos, CA, 1987, pp. 193-216.

[Reiter 78] R REITER, *On closed world data bases.* Logic and Data Bases, H. Gaillaire and J. Minker (Eds.), Plenum Press, New York, 1978, pp. 55-76. Also in: Readings in Nonmonotonic Reasoning, M. L. Ginsberg (ed.), Morgan Kaufmann Publ., Los Altos, 1987, pp. 300-310.

[Reiter 80] R REITER, *A Logic for Default Reasoning.* Artificial Intelligence, 13 (1,2), 1980, pp. 81-132. Also in: Readings in Nonmonotonic Reasoning, M. L. Ginsberg (ed.), Morgan Kaufmann Publ., Los Altos, 1987, pp. 68-93.

[Ross 89b] KENNETH ROSS, *The Well Founded Semantics for Disjunctive Logic Programs.* Proceedings of the First International Conference on Deductive and Object Oriented Databases, 1989, pp. 352-369.

Possible Models Semantics for Disjunctive Databases II.

[Sheperdson 86] J. SHEPHERDSON, *Negation in Logic Programming.* Proceedings of the 1986 Workshop on Foundations of Deductive Databases and Logic Programming, 1986. Also in: Foundations of Deductive Databases and Logic Programming, J. Minker (ed.), Morgan Kaufmann, Los Altos, CA, 1987, pp. 19-88.

[Witteveen 90] CEES WITTEVEEN, *Partial Semantics for Truth Maintenance: a compositional approach.*, TU Delft, september 1990.

[Witteveen 90a] CEES WITTEVEEN, *Fixpoint Semantics for Truth Maintenance*, TWI-report 90-85, Faculty of Technical Mathematics and Computer Science, Delft University of Technology, Delft, 1990.

[Witteveen 90a] CEES WITTEVEEN, *Partial Semantics for Truth Maintenance*, in : J.W. van Eyk(ed.), Proceedings of JELIA90, LNCAI, Springer Heidelberg, 1990, (to appear).

[Witteveen 91] CEES WITTEVEEN, *Skeptical Belief Revision is Tractable.*, TU Delft, ferbruari 1991.

Logic Group Preprint Series
Department of Philosophy, University of Utrecht
Heidelberglaan 2, 3584 CS Utrecht
The Netherlands

1 C.P.J. Koymans, J.L.M. Vrancken, *Extending Process Algebra with the empty process*, September 1985
2 J.A. Bergstra, *A process creation mechanism in Process Algebra*, September 1985
3 J.A. Bergstra, *Put and get, primitives for synchronous unreliable message passing*, October 1985
4 A. Visser, *Evaluation, provably deductive equivalence in Heyting's arithmetic of substitution instances of propositional formulas*, November 1985
5 G.R. Renardel de Lavalette, *Interpolation in a fragment of intuitionistic propositional logic*, January 1986
6 C.P.J. Koymans, J.C. Mulder, *A modular approach to protocol verification using Process Algebra*, April 1986
7 D. van Dalen, F.J. de Vries, *Intuitionistic free abelian groups*, April 1986
8 F. Voorbraak, *A simplification of the completeness proofs for Guaspari and Solovay's R*, May 1986
9 H.B.M. Jonkers, C.P.J. Koymans & G.R. Renardel de Lavalette, *A semantic framework for the COLD-family of languages*, May 1986
10 G.R. Renardel de Lavalette, *Strictheidsanalyse*, May 1986
11 A. Visser, *Kunnen wij elke machine verslaan? Beschouwingen rondom Lucas' argument*, July 1986
12 E.C.W. Krabbe, *Naess's dichotomy of tenability and relevance*, June 1986
13 H. van Ditmarsch, *Abstractie in wiskunde, expertsystemen en argumentatie*, Augustus 1986
14 A. Visser, *Peano's Smart Children, a provability logical study of systems with built-in consistency*, October 1986
15 G.R. Renardel de Lavalette, *Interpolation in natural fragments of intuitionistic propositional logic*, October 1986
16 J.A. Bergstra, *Module Algebra for relational specifications*, November 1986
17 F.P.J.M. Voorbraak, *Tensed Intuitionistic Logic*, January 1987
18 J.A. Bergstra, J. Tiuryn, *Process Algebra semantics for queues*, January 1987
19 F.J. de Vries, *A functional program for the fast Fourier transform*, March 1987
20 A. Visser, *A course in bimodal provability logic*, May 1987
21 F.P.J.M. Voorbraak, *The logic of actual obligation, an alternative approach to deontic logic*, May 1987
22 E.C.W. Krabbe, *Creative reasoning in formal discussion*, June 1987
23 F.J. de Vries, *A functional program for Gaussian elimination*, September 1987
24 G.R. Renardel de Lavalette, *Interpolation in fragments of intuitionistic propositional logic*, October 1987 (revised version of no. 15)
25 F.J. de Vries, *Applications of constructive logic to sheaf constructions in toposes*, October 1987
26 F.P.J.M. Voorbraak, *Redeneren met onzekerheid in expertsystemen*, November 1987
27 P.H. Rodenburg, D.J. Hoekzema, *Specification of the fast Fourier transform algorithm as a term rewriting system*, December 1987
28 D. van Dalen, *The war of the frogs and the mice, or the crisis of the Mathematische Annalen*, December 1987
29 A. Visser, *Preliminary Notes on Interpretability Logic*, January 1988
30 D.J. Hoekzema, P.H. Rodenburg, *Gauß elimination as a term rewriting system*, January 1988
31 C. Smoryński, *Hilbert's Programme*, January 1988
32 G.R. Renardel de Lavalette, *Modularisation, Parameterisation, Interpolation*, January 1988
33 G.R. Renardel de Lavalette, *Strictness analysis for POLYREC, a language with polymorphic and recursive types*, March 1988
34 A. Visser, *A Descending Hierarchy of Reflection Principles*, April 1988
35 F.P.J.M. Voorbraak, *A computationally efficient approximation of Dempster-Shafer theory*, April 1988
36 C. Smoryński, *Arithmetic Analogues of McAloon's Unique Rosser Sentences*, April 1988

37 P.H. Rodenburg, F.J. van der Linden, *Manufacturing a cartesian closed category with exactly two objects*, May 1988

38 P.H. Rodenburg, J.L.M.Vrancken, *Parallel object-oriented term rewriting : The Booleans*, July 1988

39 D. de Jongh, L. Hendriks, G.R. Renardel de Lavalette, *Computations in fragments of intuitionistic propositional logic*, July 1988

40 A. Visser, *Interpretability Logic*, September 1988

41 M. Doorman, *The existence property in the presence of function symbols*, October 1988

42 F. Voorbraak, *On the justification of Dempster's rule of combination*, December 1988

43 A. Visser, *An inside view of EXP, or: The closed fragment of the provability logic of* $I\Delta_0+\Omega_1$, February 1989

44 D.H.J. de Jongh & A. Visser, *Explicit Fixed Points in Interpretability Logic*, March 1989

45 S. van Denneheuvel & G.R. Renardel de Lavalette, *Normalisation of database expressions involving calculations*, March 1989

46 M.F.J. Drossaers, *A Perceptron Network Theorem Prover for the Propositional Calculus*, July 1989

47 A. Visser, *The Formalization of Interpretability*, August 1989

48 J.L.M. Vrancken, *Parallel Object Oriented Term Rewriting : a first implementation in Pool2*, September 1989

49 G.R. Renardel de Lavalette, *Choice in applicative theories*, September 1989

50 C.P.J. Koymans & G.R. Renardel de Lavalette, *Inductive definitions in COLD-K*, September 1989

51 F. Voorbraak, *Conditionals, probability, and belief revision (preliminary version)*, October 1989

52 A. Visser, *On the* Σ^0_1*-Conservativity of* Σ^0_1*-Completeness*, October 1989

53 G.R. Renardel de Lavalette, *Counterexamples in applicative theories with choice*, January 1990

54 D. van Dalen, *L.E.J. Brouwer. Wiskundige en Mysticus*, June 1990

55 F. Voorbraak, *The logic of objective knowledge and rational belief*, September 1990

56 J.L.M. Vrancken, *Reflections on Parallel and Functional Languages*, September 1990

57 A. Visser, *An inside view of EXP, or: The closed fragment of the provability logic of* $I\Delta_0+\Omega_1$, revised version with new appendices, October 1990

58 S. van Denneheuvel, K. Kwast, G.R. Renardel de Lavalette, E. Spaan, *Query optimization using rewrite rules*, October 1990

59 G.R. Renardel de Lavalette, *Strictness analysis via abstract interpretation for recursively defined types*, October 1990

60 C.F.M. Vermeulen, *Sequence Semantics for Dynamic Predicate Logic*, January 1991

61 M.B. Kalsbeek, *Towards the Interpretability Logic of* $I\Delta_0+EXP$, January 1991.

62 D. van Dalen, *I<R, Some Intuitionistic Elementary Equivalences*, February 1991.

63 M. Bezem, *Strong termination of Logic Programs*, March 1991.

64A. Visser, *The Unprovability of Small Inconsistency*, March 1991.

65 C.M. Jonker, *On the Semantics of Conflict Resolution in Truth Maintenance Systems*, March 1991.