Chapter 5

SEMANTIC TREE BASED RESOLUTION VARIANTS

5.1 PRELIMINARIES

In this chapter we shall consider an approach to the decision problem which is basically the one worked out by William H. Joyner in his dissertation [Joy 73] (see also [Joy 76]). Most of the results of this chapter already appeared in [Fer 91a] and [Fer 91]. We shall use A-ordering strategies and saturation mechanisms to guarantee that there are only finitely many resolvents for classes of clause sets. A kind of uniform frame for the results of this chapter is provided by the fact that all employed variants of resolution are demonstrated to be complete (at least for certain classes of clause sets) via semantic trees. Unlike in the cases investigated by Joyner, the depth of a resolvent will generally not be bounded by the maximum of the term depth of its parents. Anyway, it suffices to show that a "global" term depth limit exists, i.e. the term depth of resolvents cannot grow arbitrarily often. In order to achieve a decision procedure we also have to guarantee a length limit for the generated clauses. This, in most of the cases described below, will be achieved by employing the splitting rule.

Our reinvestigation of A-orderings and saturation should make clear that the method is by no means restricted to the prefix classes that have been considered by Joyner. On the contrary, to rely on the flexibility of the clause syntax terminology also for the definition of the decision classes, allows for both: generalisations of Joyner's results and more transparent procedures and proofs.

A-orderings can be described as a special subcase of π-orderings as treated in chapter 4. Nevertheless, from a methodological point of view, it seems worthwhile to study A-ordering strategies indepedently of this more general context of π-ordering refinements: A-orderings are purely syntactically defined π-orderings on atoms (not literals!), which e.g. means that no reference to the location of clauses within a "proof tree" is needed. Therefore they are somewhat easier to describe and handle than general π-orderings. (See chapter 4 for more details). More importantly, we may employ the well known device of semantic trees for the completeness proofs, which directly implies the admissibility of all types of subsumption and tautology elimination rules for the resolution variants of this chapter.

5.1.1 A-orderings

The concept of A-orderings was introduced by Kowalski and Hayes [KH69]. We present a definition, that can easily be demonstrated to be equivalent with that of Kowalski and Hayes.

<u>DEFINITION</u> 5.1.: An <u>A-ordering</u> $<_A$ is a binary relation on atoms s.t.

(i) $<_A$ is irreflexive,

(ii) $<_A$ is transitive,

(iii) for all atoms A, B and all substitutions ϑ:

$$A <_A B \text{ implies } A\vartheta <_A B\vartheta.$$

Joyner remarks that this definition should be augmented by the following condition:

(iv) for every clause set S, $<_A$ is compatible with some enumeration of the corresponding Herbrand base (see def. 2.31). I.e. there is some enumeration $A_0, A_1 \ldots$ of all ground atoms that are Herbrand instances of atoms occurring in S s.t. $A_i <_A A_j$ implies $i < j$.

According to Joyner, Kowalski and Hayes make implicit use of such an ordering compatible enumeration in their completeness proof. Thus, in Joyner's version of A-orderings, it is e.g. not allowed to order all atoms starting with predicate letter "P" in front of all atoms starting with "Q", although this ordering trivially fulfills conditions (i) - (iii). But anyway, we shall see in section 5.1.2 that condition (iv) is not necessary to ensure the completeness of the resulting resolution strategy.

A more subtle point concerns the definition of the resolvents that obey the A-ordering restriction. Whereas Kowalski and Hayes demand the atoms to be resolved upon to be maximal w.r.t. the ordering, we shall use the following definition which is due to Joyner.

DEFINITION 5.2.: For any clause set S and any A - ordering $<_A$:
$E \in R_{<_A}(S)$ iff E is a Robinson - resolvent of clauses in S s.t. for no atom B in E: $A <_A B$, where A is the resolved atom.

According to the terminology developed in chapter 4 this means that we are employing an a posteriori criterion as opposed to the weaker (i.e. less restrictive) priori criterion as given by Kowalski and Hayes.

It is interesting to mention that in order to arrive at decision procedures for certain classes of clause sets we have to insist on the stronger a posteriori condition.

5.1.2 Completeness via semantic trees

In the sections to come we shall not only refer to the completeness of A - ordering strategies, but also have to present completeness proofs for some more special resolution refinements. For this purpose we rely on the the important device of semantic trees. To remain fairly selfcontained and to easify the argumentation in later sections we present a proof of the (refutational) completeness of A - ordering strategies. We stress that the definition and theorems in this section are not new; versions of them rather can be found e.g. in [KH69] or [Joy73].

Semantic trees have been introduced by Robinson [Rob68] and remained a standard tool in the field of automated theorem proving ever since. Our version of semantic trees is basically the one of Kowalski and Hayes [KH68]. We repeat the relevant definitions, but assume the reader to be familiar with the basic vocabulary of graph theory.

DEFINITION 5.3.: A semantic tree based on a set of atoms K is a binary tree with elements of K and their negations labelling the edges in the following way:
Let A_0, A_1,... be an enumeration of K. The two edges leaving the root of the corresponding semantic tree T are labelled A_0 and $\neg A_0$, respectively; and if either A_i or $\neg A_i$ labels the edge entering a node, then A_{i+1} and $\neg A_{i+1}$ label the edges leaving that node. With each node of T we associate a refutation set, consisting of the literals labelling the edges of the path from the root down to this node.

We shall only use semantic trees that are based on finite subsets of some Herbrand base (for the relevant definitions we refer to chapter 2).

DEFINITION 5.4.: A clause C fails at a node of a semantic tree T iff the complement of every literal in some ground instance $C\gamma$ of C is in the refutation set of that node. A node n is a failure node for a clause set S iff some clause of S fails at n but no clause of S fails at a node above n (i.e. a node with a shorter path to the root). A node is called inference node iff both of its sons are failure nodes. T is closed for S iff there is a failure node for S on every branch of T.

The completeness proofs of this chapter essentially rest on the following well known corollary to Herbrand's Theorem:

THEOREM 5.1.: If a clause set S is unsatisfiable then there is a finite subset K of $\hat{H}_S$ s.t. every semantic tree T based on K is closed for S.

A proof of theorem 5.1 can e.g. be found in [KH69].

Resolution strategies R are usually defined such that, for any clause set S, the set of resolvents R(S) is a proper subset of the set of Robinson-resolvents. In contrast, we shall also investigate strategies where R(S) contains certain instances of Robinson-resolvents.

DEFINITION 5.5.: Let C be a class of clause sets. We call a resolution strategy R complete via semantic trees on C iff

(i) for any unsatisfiable S in C there is some finite subset K of $\hat{H}_S$ s.t. there exists a semantic tree T based on K closed for S,

(ii) for any two clauses C, D which fail immediately below an inference node of T there is an R - resolvent of C and D which fails at the inference node.

Given theorem 5.1 it is an easy task to prove

LEMMA 5.2.: Any resolution strategy R which is complete via semantic trees on some class C is a (refutational) complete deduction procedure for class C.

Proof: Let S be an unsatisfiable clause set in C. Observe that, theorem 5.1 guarantees the existence of a semantic tree T as specified in condition (i) of definition 5.5. It remains to show that condition (ii) implies Box in $R^*(S)$.

Q.E.D.

Because $S \subseteq R(S)$ T is also closed for R(S) and any failure node for S is either also a failure node for R(S) or some predecessor of this node is a failure node for R(S). Since T is closed for S there must be at least one inference node for S in T. But then, by condition (ii), some failure node for R(S) is a predecessor of at least two failure nodes for S. This means that the number of failure nodes shrinks strictly monotonically as we iteratively apply R. Therefore there exists some n s.t. some clause in $R^n(S)$ fails at the root of T. But the only clause failing at the root of a semantic tree is $\square$.

Q.E.D.

We are now prepared to present a completeness proof for A - ordering strategies.

THEOREM 5.3: For any A - ordering $<_A$ $R_{<_A}$ is a resolution strategy which is complete on all classes of clause sets.

Proof:

Let S be any unsatisfiable clause set. By theorem 5.1 there is a finite $K \subseteq \hat{H}_S$ s.t. any semantic tree T for K is closed for S. Because of lemma 5.2 it suffices to show that we can choose T s.t. if clauses C and D define an inference node

in T then some $R_{<_A}$ -resolvent E of C and D fails above this node.
To this aim we order the atoms of K in a list A_0, A_1, ..., A_n compatible with $<_A$; i.e. $A_i <_A A_j$ implies $i < j$. Let T be the semantic tree labelled according to this enumeration of K (compare definition S.3). Let C and D be clauses in S failing immediately below an inference node n of T. Let A_k and $\neg A_k$ be the two ground literals labeling the edges leaving the inference node n. Clearly, there is a Robinson- resolvent E of C and D s.t. A_k is ground instance of the corresponding resolved atom A. We show that E also is an $R_{<_A}$ - resolvent.

Let γ be the substitution s.t. $E\gamma$ is in the refutation set of n. By definition of T we have $A\gamma \not<_A B\gamma$ for all atoms B in E. But since $<_A$ is an A- ordering $A\vartheta \not<_A B\vartheta$ implies $A \not<_A B$ for all substitutions ϑ. Therefore E fulfills the defining conditions of an $R_{<_A}$- resolvent.

Q.E.D.

A look at the terminology used in the above proof reveals why, contrary to the comments of Joyner, condition (iv) as refinement of the definition of A - orderings is empty: As we based the semantic trees on finite subsets of the Herbrand base there is always an enumeration of the relevant atoms which is compatible with the ordering.

As for the completeness in presence of the splitting rule (see chapter 2), note that if a clause C fails at some node n then clearly all split components of C fail at n, too. Thus all resolution strategies that are complete via semantic trees remain complete if combined with the splitting rule. Similar facts hold for the usual deletion strategies: A tautology does not fail at any node of a semantic tree, because no refutation set can contain both, an atom and its negation. Moreover if C subsumes D, i.e. there is a substitution ϑ s.t. $C\vartheta \subseteq D$, then clearly C fails at any node at that D fails. Therefore tautologies and subsumed clauses may be removed from the sets of resolvents without effecting completeness (for the details we again refer to [KH69]).
We remark that actually it is not necessary to employ the - computationally complex - splitting rule in order to get decision procedures. However, the proofs are much simpler if the clauses are assumed to be decomposed.

5.1.3 Covering terms and atoms

Throughout the next sections we speak of terms, atoms and literals characterized by special properties which we shall call covering and weakly covering, respectively:

DEFINITION 5.6.: A functional term t is called covering iff for all functional subterms s occurring in t we have $V(s) = V(t)$. An atom or literal A is called covering iff each argument of A is either a constant or a variable or a covering term t s.t. $V(t) = V(A)$.

DEFINITION 5.7.: A functional term t is called weakly covering iff for all non ground, functional subterms s of t: $V(s) = V(t)$. Similary to definition 5.6, an atom or literal A is called weakly covering iff each argument of A is either a ground term or a variable or a weakly covering term t s.t. $V(t) = V(A)$.

EXAMPLES: $g(h(x), a, f(x, x))$ is a covering term $f(x, g(h(a), f(y,x), y))$ is weakly covering but not covering; $g(f(x,y), x, h(x))$ is not (weakly) covering. $P(h(x), x)$ is a covering and $Q(f(x, f(x,y)), f(y, x), h(a))$ a weakly covering atom; $P(f(a, h(b)), h(c))$, like any ground atom, is covering, too. $P(h(x),y)$ and $P(h(x), f(x,y))$ are not weakly covering.

Clearly, all covering atoms or terms are also weakly covering, but the converse does not hold. Covering atoms originate for instance by Skolemization of (pure) prenex formulas when the universal quantifiers precede all existential quantifiers. Also observe that any atom or term that contains at most one variable is weakly covering.

In section 5.4.1 below we shall prove that neither the term depth nor the number of variables of the resolvents in a certain class of clause sets, which is defined in terms of covering literals, can increase. For this purpose we need the following lemma:

<u>LEMMA</u> 5. 4: Let ϑ be a m.g.u. of two covering atoms A and B; then the following properties hold for $A\vartheta$ (= $B\vartheta$):

(i) $A\vartheta$ is covering,

(ii) $\tau(A\vartheta) = \max(\tau(A), \tau(B))$

(iii) $|V(A\vartheta)| \leq \max(|V(A)|, |V(B)|)$.

<u>Proof</u>:

Some additional terminology will result in a more concise formulation of the proof: Let us write $A \geq B$ if $A = B$ or if B can be obtained from A by substituting some variables of A by constants or other variables. Observe that,

(*) if $A \geq B$, then B is covering whenever A is covering.

Moreover we write $A < B$ if each functional subterm t of B contains all variables of A (i.e. $V(A) \subseteq V(t)$ for all functional subterms t of B).

We have to trace the process of unification. Let ρ_i denote the concatenation of the mesh substituents applied so far, i.e.

$\rho_0 = \varepsilon$,

$\rho_i = \{ t_1/x_1 \} \cdot \{ t_2/x_2 \} \cdot \ldots \{t_i/ x_i \}$, where

$\{t_i / x_i\}$ denotes the i-th mesh substituent. (See chapter 2. 5) .

We prove that $[A \geq A\vartheta$ or $B \geq B\vartheta]$ by induction in the number of substitution steps using the following induction hypothesis:

Either

(IH1)	$A \geq A\rho_i$	and	$B \geq B\rho_i$	or
(IH2)	$A \geq A\rho_i$	and	$A\rho_i < B\rho_i$	or
(IH3)	$B \geq B\rho_i$	and	$B\rho_i < A\rho_i$	

(IH1) trivially holds for $i = 0$. Let $\{ t_{i+1}/ x_{i+1} \}$ be the next mesh substituent. W.l.o.g. we may assume that x_{i+1} is found in $A\rho_i$ and t_{i+1} in $B\rho_i$ in the point of disagreement between $A\rho_i$ and $B\rho_i$ (otherwise exchange $A\rho_i$ and $B\rho_i$) .

We have to consider the following cases:

(1) (IH1) holds:

(1a) t_{i+1} is a variable or a constant:
This means that $A\rho_i \geq A\rho_{i+1}$ and $B\rho_i \geq B\rho_{i+1}$. Thus, by transitivity of the relation " $\geq$ ", (IH1) also holds for i+1.

(1b) t_{i+1} is a functional term:
By (IH1) and (*) $B\rho_i$ is covering. Therefore $V(t_{i+1}) = V(B\rho_i)$. This implies that $B\rho_i < A\rho_{i+1}$ (= $A\rho_i \cdot \{t_{i+1}/ x_{i+1}\}$). Now observe that x_{i+1} is not in t_{i+1} (otherwise A and B would not be unifiable). But as $B\rho_i$ is covering this amounts to $B\rho_i = B\rho_{i+1}$ Summarizing we have shown that, in this case, (IH3) must hold for i+1.

(2) (IH2) holds:

(2a) t_{i+1} is a variable or a constant: Clearly $A\rho_i \geq A\rho_{i+1}$ and $A\rho_{i+1} < B\rho_{i+1}$. Therefore, similiar to case (1a), (IH2) remains valid.

(2b) t_{i+1} is a functional term: This cannot happen because $A\rho_i < B\rho_i$ implies $V(A\rho_i) \subseteq V(t_{i+1})$. This, in turn, would mean that x_{i+1} in $V(t_{i+1})$ which contradicts the assumption that A and B are unifiable.

(3) (IH3) holds: The argument is completely analogous to case (2).

Summarizing we have shown that $A \geq A\vartheta$ or $B \geq B\vartheta$. As A and B are covering it follows that $A\vartheta$ (= $B\vartheta$) is covering, too. Moreover, by definition of $\geq$, $A \geq A\vartheta$ implies $\tau(A) = \tau(A\vartheta)$, therefore $\tau(A\vartheta) = \max(\tau(A), \tau(B))$. Finally, observe that $A\vartheta$ cannot contain more than $\max(|V(A)|, |V(B)|)$ variables.

Q.E.D.

To assist the formulation of a similiar lemma for weakly covering atoms we introduce the following notation:

DEFINITION 5. 8.: For any atom or literal A let $\Gamma(A)$ be the set of all ground terms that are subterms of A. Additionally, let $\tau_g(A)$ be the term depth of the deepest term in $\Gamma(A)$; i.e. $\tau_g(A) = \max\{\tau(t) \mid t \text{ in } \Gamma(A)\}$. The definitions of Γ and τ_g are extended to clauses C in the obvious way, i.e. $\Gamma(C) = \bigcup_{L \in C} \Gamma(L)$ and $\tau_g(C) = \max\{\tau_g(L) \mid L \text{ in } C\}$.

DEFINITION 5.9.: For any atom, literal or clause E let $\tau_V(E)$ be the maximum of $\tau_{max}(x, E)$ over all $x \in V(E)$. If E is ground we define $\tau_V(E) = 0$.

Observe that in any covering or weakly covering atom the maximal depth of occurrence is the same for all variables of the atom. Thus $\tau_V(A) = \tau_{max}(x, A)$ for every $x \in V(A)$.

We may now state:

LEMMA 5.5.: Let ϑ be a m.g.u. of two weakly covering atoms A and B then the following properties hold for $A\vartheta$ (= $B\vartheta$):

(i) $A\vartheta$ is weakly covering,

(ii) $\tau_V(A\vartheta) \leq \max(\tau_V(A), \tau_V(B))$,

(iii) $\tau(A\vartheta) \leq \max(\tau_V(A), \tau_V(B)) + \max(\tau_g(A), \tau_g(B))$,

(iv) either $\Gamma(A\vartheta) \subseteq \Gamma(A) \cup \Gamma(B)$ or $A\vartheta$ is ground,

(v) $|V(A\vartheta)| \leq \max(|V(A)|, |V(B)|)$.

Proof:

We proceed similarly to the proof of lemma 5.4. For atoms A, B and a set of ground terms G let us write $A \geq_G B$ if $A = B$ or if B can be obtained from A by substituting some variables of A by other variables or some member of G. It clearly holds that,

(*') if $A \geq_G B$, then B is weakly covering whenever A is weakly covering (for any set of ground terms G).

Moreover we write $A <' B$ iff each subterm t of B, that properly contains variables, contains all variables of A (i.e., $|V(t)| = 0$ or $V(A) \subseteq V(t)$ for all subterms t of B).

Again, let ρ_i denote the concatenation of the first i mesh substituents of the unifier.

In the following we use E as an abbreviation for $\Gamma(A) \cup \Gamma(B)$ and prove that $A \geq_E A\vartheta$ or $B \geq_E B\vartheta$ by induction on the number of mesh substitions. The induction hypothesis now reads:

Either

(IH1)	$A \geq_E A\rho_i$	and	$B \geq_E B\rho_i$	and	$\Gamma(A\rho_i) \cup \Gamma(B\rho_i) = E$, or
(IH2)	$A \geq_E A\rho_i$	and	$A\rho_i < B\rho_i$	and	$\Gamma(A\rho_i) \cup \Gamma(B\rho_i) = E$, or
(IH3)	$B \geq_E B\rho_i$	and	$B\rho_i < A\rho_i$	and	$\Gamma(A\rho_i) \cup \Gamma(B\rho_i) = E$, or
(IH4)	either $A\rho_i$	or	$B\rho_i$ is ground.		

Clearly, (IH1) holds for $i = 0$. Let $\{x_{i+1} \leftarrow t_{i+1}\}$ be the next mesh substituent. W.l.o.g. we assume that x_{i+1} is found in $A\rho_i$ and t_{i+1} in $B\rho_i$ in the point of disagreement between $A\rho_i$ and $B\rho_i$.

We have to investigate the following cases:

(1) (IH1) holds:

(1a) t_{i+1} is a variable:

In this case $A\rho_i \geq_E A\rho_{i+1}$ and $B\rho_i \geq_E B\rho_{i+1}$. Thus, by transitivity of the relation "$\geq_E$", $A \geq_E A\rho_{i+1}$ and $B \geq_E B\rho_{i+1}$. Moreover, the set of ground subterms of $A\rho_{i+1}$ and $B\rho_{i+1}$ remains equal to E. Therefore (IH1) also holds for i+1.

(1b) t_{i+1} is a functional term containing variables:

(The argumentation parallels that of case (1b) in the proof of lemma 5. 4). By (IH1) and (*') $B\rho_i$ is weakly covering. Therefore $V(t_{i+1}) = V(B\rho_i)$. This implies that

$$B\rho_i <' A\rho_{i+1} \ (= A\rho_i \cdot \{x_{i+1} \leftarrow t_{i+1}\}).$$

Since A and B are unifiable we have $x_{i+1} \notin V(t_{i+1})$. Because $B\rho_i$ is weakly covering this implies

$$B\rho_i = B\rho_{i+1}.$$

Again, $\Gamma(A\rho_{i+1}) \cup \Gamma(B\rho_{i+1}) = \Gamma(A\rho_i) \cup \Gamma(B\rho_i) = E$.

Summarizing we have shown that, in this case, (IH3) must hold for i+1.

(1c) t_{i+1} is a ground term. There are two possibilities:

(A) Both, $A\rho_{i+1}$ and $B\rho_{i+1}$, still contain variables: In this case no new ground terms arise, i.e. $\Gamma(A\rho_{i+1}) \cup \Gamma(B\rho_{i+1}) = \Gamma(A\rho_i) \cup \Gamma(B\rho_i) = E$. Moreover, $A\rho_i \geq_E A\rho_{i+1}$ and $B\rho_i \geq_E B\rho_{i+1}$. Thus also $A \geq_E A\rho_{i+1}$ and $B \geq_E B\rho_{i+1}$. Therefore (IH1) still holds. (B) Either $A\rho_{i+1}$ or $B\rho_{i+1}$ is ground: Then clearly (IH4) holds.

(2) (IH2) holds:

(2a) t_{i+1} is a variable or a ground term:

Clearly, $A\rho_i \geq_E A\rho_{i+1}$ and $A\rho_{i+1}$ <' $B\rho_{i+1}$.

Similiar to case (1c), either

$\Gamma(A\rho_{i+1}) \cup \Gamma(B\rho_{i+1}) = \Gamma(A\rho_i) \cup \Gamma(B\rho_i) = E$ or at least one of $A\rho_{i+1}$, $B\rho_{i+1}$ is ground. In the first case (IH2) remains valid; in the second case (IH4) holds for i+1.

(2b) t_{i+1} is a functional term containing variables:

This cannot happen because $A\rho_i <'_E B\rho_i$ implies $V(A\rho_i) \subseteq V(t_{i+1})$. This, in turn, we would mean that x_{i+1} is in $V(t_{i+1})$ which contradicts the assumption that A and B are unifiable.

(3) (IH3) holds: The argument is completely analogous to case (2).

(4) (IH4) holds: Clearly, a ground atom remains unchanged by applications of substitutions. Therefore (IH4) also holds for i+1.

We have just proved that $A \geq_E A\vartheta$ or $B \geq_E B\vartheta$, thus statements (i), (ii), and (v) of the lemma follow directly by definiton of "$\geq_E$" and τ_V, respectively. An inspection of the induction hypothesis and the single steps of the argumentation above reveals that also (iii) and (iv) hold.

Q.E.D.

We also shall make use of the following lemma:

<u>LEMMA</u> 5.6.: Let A and B be two (weakly) covering atoms s.t. $V(A) = V(B)$. For any substitution ϑ it holds: If $A\vartheta$ is (weakly) covering then $B\vartheta$ is (weakly) covering, too.

<u>Proof</u>:

We have to show that for each functional subterm t of $B\vartheta$ either $V(t) = V(B\vartheta)$ (i.e. t contains all variables of $B\vartheta$) or, in the case of weakly covering atoms, $V(t) = \emptyset$ (i.e. t is ground).

Clearly, one of the following cases must apply:

(1) $t = s\vartheta$ for some functional subterm s of B:

If A and B are covering, then by definition, $V(s) = V(B)$.

This implies $V(t) = V(s\vartheta) = V(B\vartheta)$.
If A and B are only weakly covering then s may also be ground although B contains variables. But in this case $t = s\vartheta$ must be ground, too.

(2) t is a subterm of some term in $rg(\vartheta)$: Since $A\vartheta$ is (weakly) covering (either t is ground or) $V(t) = V(A\vartheta)$. But, since $V(B) = V(A)$, we have $V(B\vartheta) = V(A\vartheta)$. Therefore, also in this case, $V(t) = V(B\vartheta)$.

Q. E. D.

5.2 A SIMPLE A-ORDERING STRATEGY

The concept of (weakly) covering literals permits concise definitions of some decision classes. Consider, e.g., the following class of clause sets:

5.2.1 Class E_1

DEFINITION 5.10: (E_1): A clause set S belongs to E_1 iff the following holds for all clauses C in S:

(i) All literals in C are covering, and
(ii) for all literals L, M in C either $V(L) = V(M)$ or $V(L) \cap V(M) = \emptyset$.

EXAMPLES. Let $C_1 = \{P(f(x,y), a), Q(y,x,x)\}$, $C_2 = \{P(f(x,f(x,a)), Q(z,y,a)\}$, $C_3 = \{P(x,f(a))\}$ and $C_4 = \{Q(x,y,a), P(x,x)\}$. Then $\{C_1, C_2\}$ in E_1, but any clause set containing C_3 or C_4 is not in E_1.

E_1 may be regarded as an extension of the initially extended Ackermann class (characterized by the prefix type $\exists^* \forall \exists^*$). A closely related class, called E^+, will be considered in the section below (compare also chapter 4).

We demonstrate that a simple A- ordering strategy guarantees a term depth limit for resolvents of clauses in E_1. A limit for the clause length can be achieved by splitting; thus we arrive at a resolution variant which provides a decision procedure for E_1.

The following A - ordering will prove suitable to provide a term - depth limit for covering literals.

DEFINITION 5.11.: Let A and B be two atoms, then $A <_d B$ iff

(i) $\tau(A) < \tau(B)$, and

(ii) for all x in V(A): $\tau_{max}(x, A) < \tau_{max}(x, B)$
(implying $V(A) \subseteq V(B)$).

We first have to show:

LEMMA 5.7.: $<_d$ is an A-ordering.

Proof:

Irreflexivity directly follows from the irreflexivity of "<"; transitivity holds because both "<" and "$\subseteq$" are transitive.
Let A and B be atoms s.t. $A <_d B$ and let $\sigma = \{x \leftarrow t\}$ be a component of some substitution. If $x \notin V(A)$ then $A\sigma = A$; therefore $A\sigma <_d B\sigma$ holds. If x in V(A) then, by definition of "$<_d$" $\tau_{max}(x, A) < \tau_{max}(x, B)$. Let y be any variable of V(t). We have

$$\tau_{max}(y, A\sigma) = \tau_{max}(x, A) + \tau_{max}(y, t) \text{ and}$$

$$\tau_{max}(y, B\sigma) = \tau_{max}(x, B) + \tau_{max}(y, t) \text{ and therefore}$$

$$\tau_{max}(y, A\sigma) < \tau_{max}(y, B\sigma).$$

Applying some substitution ϑ to an atom has the same effect as applying all components of ϑ separately. Thus we have proved that $A <_d B$ implies $A\vartheta <_d B\vartheta$ for any ϑ.

Q.E.D.

We now show that, for every $S \in E_1$, every resolvent in $R_{<_d}(S)$ – as defined in definition 5.2 fufills the defining conditions of class E_1 and is smaller or equal w.r.t. term depth to its parent clauses. This is mainly a consequence of lemmata 5.4 and 5.6.

LEMMA 5.8.: If $E \in R_{<_d}(\{C,D\})$, where $\{C,D\}$ in E_1, then $\{E\} \in E_1$ and $\tau(E) \leq \tau(\{C,D\})$.

Proof:
Let M be the set of literals resolved upon in C and L be a literal in C - M; by ϑ we denote the m.g.u. used to generate E. By definition any two literals of C share either all or none of their variables. We have to consider the following cases:

(1) $V(L) \cap V(M) = \emptyset$: In this case $\vartheta(x) = x$ for all x in $V(C - M)$ which implies $L\vartheta = L$. Thus L occurs unchanged in E.

(2) $V(L) = V(L')$ for some $L' \in M$:

(2a) $L <_d L'$: As "$<_d$" is an A- ordering we have $L\vartheta <_d L'\vartheta$ which implies $\tau(L\vartheta) < \tau(L'\vartheta)$. Thus by lemma 5.4 also $\tau(L\vartheta) < \tau(\{C,D\})$. Moreover it follows from lemma 5.6 that $L\vartheta$ is covering.

(2b) $L \not<_d L'$: By definiton of $R_{<_d}$ we also know that $L' \not<_d L$. Now observe that in a covering atom A all variables occur somewhere in maximal depth; i.e. for all x in V(A): $\tau_{max}(x, A) = \tau(A)$. This implies $\tau(L) = \tau(L')$ and $\tau(L\sigma) = \tau(L'\sigma)$ for any substitution σ. Moreover lemmata 5.4 and 5.6 again guarantee that $L\vartheta$ is covering.

By analogy the same holds for the literals in D. Thus we have proved that E fullfills the relevant conditions.

Q.E.D.

If we use $R_{<_d}$ in combination with the splitting rule (c.f. definition 2.29) we arrive at the following theorem:

THEOREM 5.9.: Class E_1 is decidable; $R_{<_d}$ combined with the splitting rule provides a decision procedure.

Proof:
Lemma 5.8 provides a term depth limit and guarantees that the resolvents are in E_1 again. What is still missing in order to obtain the decidability of E_1 is a length limit for the resolvents. To this aim we employ the splitting rule.

In order to decide whether a clause set $S \in R_{<_d}$ is satisfiable we first construct SPLIT(S) and then apply $R_{<_d}$ to each S' in SPLIT(S).

Because of condition (ii) in the definition of E_1 the set of variables V(L) is the same for all literals L of a split component of a clause. This fact together with part (iii) of lemma 5.4 implies

$$|V(E)| \leq \max(|V(C)|, |V(D)|)$$

for resolvents E of clauses C and D that are split components of clauses in class E_1. Considering the term depth limit for the resolvents as expressed by lemma 5.8 we arrive at a length limit because, under equivalence w.r.t. renaming of variables, there are only finitely many literals L if $|V(L)|$ as well as $\tau(L)$ are bounded. (Remember that we only consider finite clause sets and thus may assume that there are only finitely many predicate letters and function symbols).

Q.E.D.

5.3 CLASS E^+

The following class was investigated first by T.Tammet in [Tam90] (compare also chapter 4):

<u>DEFINITION</u> 5.12.: (E^+): A clause set S belongs to E^+ iff the following holds for all clauses C in S:

(i) All literals in C are weakly covering, and

(ii) for all literals L, M in C either $V(L) = V(M)$ or $V(L) \cap V(M) = \emptyset$.

<u>EXAMPLES</u>. Let $C_1 = \{P(f(x,y), a), Q(y,x,x)\}$, $C_2 = \{P(f(x, f(x,a)), Q(z,y,a)\}$, $C_3 = \{P(x, f(a))\}$ and $C_4 = \{Q(x,y,a), P(x,x)\}$. Then $\{C_1, C_2, C_3\}$ in E^+, but any clause set containing C_4 is not in E^+.

E^+ clearly is a superset of E_1, because it additionally allows arbitrary ground terms to occur everywhere in the clauses. E^+ also contains the class of clause sets only consisting of clauses C s.t. $|V(C)| \leq 1$. This subclass seems to have been proved decidable first by Y. Gurevich [Gur73] and is often called class E, which motivated the name E^+ for the class defined above.

T. Tammet [Tam90] showed that a special ordering strategy terminates on all sets in E^+ (i.e. there are only finitely many resolvents). But, as already mentioned in chapter 4, Tammet's refinement is neither an A- ordering nor even a π - ordering strategy and no completeness proof for this refinement could be achieved so far. Thus the decidability of E^+ remained an open problem until now. Using a semantic tree argument we shall show that a simple strengthening of the A-ordering strategy of the last section combined with a saturation rule is both, terminating and complete, on clause sets of E^+. (To mention an open problem we conjecture that already the $R_{<_d}$ -refinement itself is sufficient to guarantee the terminiation of the resolution procedure; however this seems hard to prove). We start by defining explicitly the ordering that arises if condition (i) of the definition of $<_d$ (definition 5.11) is dropped for non ground atoms:

<u>DEFINITION</u> 5.13.: For any two atoms A and B $A <_{vd} B$ iff

(i) $\tau(A) < \tau(B)$ whenever A and B are ground, and

(ii) for all x in V(A): $\tau_{max}(x, A) < \tau_{max}(x, B)$
(implying $V(A) \subseteq V(B)$).

<u>Remark</u>: Condition (i) is neither necessary for the termination proof nor for the completeness proof below. But since it makes the ordering condition more restrictive and thus results in a more efficient resolution procedure we prefered to include it. Observe that for non ground, weakly covering atoms A and B: $A <_{vd} B$ iff $\tau_v(A) < \tau_v(B)$.

We define $R_{<_{vd}}$ analogously to the A- ordering refinements:

<u>DEFINITION</u> 5.14.: For any clause set S: E is in $R_{<_d}(S)$ iff E is a Robinson-resolvent of clauses in S s.t. for no atom B in E: $A <_{vd} B$, where A is the resolved atom.

Unlike in the case of E_1, the $R_{<_{vd}}$ - resolvents of clauses of E^+ may be deeper w.r.t. term depth than their parent clauses.

<u>EXAMPLE</u>: Let $C = \{\neg P(x, g(x, y)), P(g(x, y), f(a))\}$ and $D = \{P(f(a), g(u, v))\}$. Then $E = \{P(g(f(a), y), f(a))\}$ is an $R_{<_{vd}}$ - resolvent of C and D; but $\tau(E) = 2$ whereas $\tau(C) = \tau(D) = 1$.

However, we shall show that, for $R_{<vd}$ -resolvents, the maximum depth of occurrences of variables (τ_V) cannot increase (compared to the corresponding parent clauses) and that this fact suffices to guarantee the termination of our resolution procedure.
We remark that the investigation of E^+ as carried out in chapter 4 is quite similar to the one presented here.

<u>THEOREM</u> 5.10.: $R_{<vd}$ terminates if applied to decomposed clause sets S in E^+, i.e. $R^*_{<vd}(S)$ is finite for all S in E^+.

<u>Proof</u>:
We start by proving that for any decomposed S in E^+ the set $R_{<vd}(S)$ is in E^+, again. Let E be some clause of $R_{<vd}(S)$ and A be the resolved atom that was used to form E. By the definition of E^+ and part (i) of lemma 5.5, A is weakly covering. Therefore all literals in E are weakly covering by lemma 5.6. The fact that $V(A) = V(B)$ implies $V(A\sigma) = V(B\sigma)$ for any atoms A, B and any substitution σ guarantees that {E} in E^+. Clearly, if E is not ground it is again decomposed. Otherwise, by the definition of $R_{<vd}$ and E^+, only new ground clauses, not deeper than their parent clauses, may be derived using E. Since there are only finitely many such clauses we may, w.l.o.g., in the following assume that the clauses are not ground.

Observe that, by part (ii) of lemma 5.5, $\tau_{V(A)} \leq \tau_V(\{C,D\})$. By the ordering condition of the definition of $R_{<vd}$ this implies that $\tau_V(E) \leq \tau_V(\{C,D\})$. We have thus proved that $\tau_V(R^*_{<vd}(S)) \leq \tau_V(S)$. This means that there is a (global) limit for the maximum depth of variable occurrences in $R_{<vd}$ - resolvents of any clause set in E^+.

Now observe that there also is *only a finite number* of different ground terms that may occur in $R_{<vd}$ - resolvents: Indeed, it follows from part (iv) of lemma 5.5 that (non ground) resolvents contain only ground terms that already *occur somewhere in the respective parent clauses*.

Combining these two results we arrive at a (global) term depth limit for the resolvents.

Since, by part (v) of lemma 5.5, the number of variables in resolvents of decomposed clauses can not surpass that of its parent clauses we have shown that – up to renaming of variables - only finitely many resolvents can be derived by $R_{<_{vd}}$.

Although we do not know whether $R_{<_{vd}}(S)$ is a complete resolution refinement – at least if restricted to E^+ – we shall show below that the combination of a special saturation rule and $R_{<_{vd}}(S)$ is complete on E^+.

<u>DEFINITION</u> 5.15.: Let L_1 and L_2 be weakly covering literals s.t. $V(L_1) = V(L_2)$. (L_1, L_2) is called a <u>critical pair</u> iff $\tau_V(L_1) < \tau_V(L_2)$ and $\tau(L_1) \geq \tau(L_2)$.

To motivate this definition and that of our saturation rule below, observe that for all clauses C and D fulfilling the defining conditions of E^+ and any $R_{<_d}$ - resolvent E of C and D: $\tau_V(E) \leq \tau_V(\{C,D\})$ unless either C or D contains a critical pair of literals. But, in contrast to $<_d$, we have $A <_{vd} B$ whenever (A, B) is a critical pair. It is this fact that prevents the generation of an $R_{<_{vd}}$ - resolvent E of C and D s.t. $\tau_V(E) > \tau_V(\{C,D\})$.

<u>EXAMPLES</u>: Let $L_1 = P(x,y,f(f(a)))$, $L_2 = Q(x, g(x,y), f(f(a)))$, and $L_3 = P(f(f(f(a))),x,y)$. Then (L_1, L_2) and (L_3, L_2) are critical pairs, but no other pair of literals in $\{L_1, L_2, L_3\}$ is critical.

We define a saturation operator that will help us to turn $R_{<_{vd}}$ into a provable complete resolution variant:

<u>DEFINITION</u> 5.16.: For any clause C in some clause set S of E^+ we define:

(i) If C contains a critical pair (L_1, L_2) then $FILL_S(C) = \{C\sigma \mid \sigma$ is a ground substitution, based on S (c.f definition 2.17) s.t. $\tau(L_1\sigma) \geq \tau(L_2\sigma)$ for all critical pairs $(L_1, L_2)\}$.

(ii) Otherwise $FILL_S(C) = \emptyset$.

For a clause set S we define $FILL(S) = S \cup \bigcup_{C \in S} FILL_S(C)$.

EXAMPLES: Let L_1 = P(x, f(f(a))), L_2 = Q(x, f(x), f(f(a))), and L_3 = P(f(f(f(a))), x). Let C = {L_1, L_2, L_3} be a clause in S, where CS(S) = {a} and FS(S) = {f}. Then

$FILL_S(C)$ = { {P(a, f(f(a))), Q(a, f(a), f(f(a))), P(f(f(f(a))), a)},
{P(f(a), f(f(a))), Q(f(a), f(f(a)), f(f(a))) P(f(f(f(a))), f(a)) },
{P(f(f(a)), f(f(a))), Q(f(f(a)), f(f(f(a))), f(f(a))), P(f(f(f(a))), f(f(a)) } }.

Observe that FILL(S) is always finite for finite clause sets S.

We define a resolution variant that combines $R_{<_{vd}}$ with the fill operator:

DEFINITION 5.17.: For any clause set S:

$$R_{FILL}(S) = FILL(R_{<_{vd}}(S)).$$

The members of $R_{FILL}(S)$ are called R_{FILL} - resolvents of S.

THEOREM 5.11.: Resolution procedure R_{FILL}, combined with the splitting rule, is complete for all clause sets S in E^+.

Proof:

Considering the remarks in section 5.1.2 about completeness via semantic trees (see especially lemma 5.2) we know that it is sufficient to prove the following:

(*) For any unsatisfiable clause set S there is a semantic tree T based on some $K \subseteq \hat{H}_S$ and closed for S s.t. any two clauses which fail immediately below an inference node of T have an R_{FILL} - resolvent which fails at that inference node.

Let T be a semantic tree closed for S, based on a subset K of $\hat{H}_S$ s.t. the atoms in any path from the root down to some other node of T are ordered according to their term depth. With other words, whenever $\tau(A) < \tau(B)$ for A, B in K then the edges of T labeled with A are situated above, i.e. closer to the root than, the edges labeled with B.

Let C and D be clauses in S failing immediately below, but not at, a node n of T. Let γ_C and γ_D be the two ground substitutions s.t. $C\gamma_C$ and $D\gamma_D$, respectively, are subsets of the refutation sets of the corresponding nodes. Finally, let A_k and $\neg A_k$ be the two ground literals labeling the edges leaving the inference node n. (W.l.o.g. we may assume that $A_k \in C\gamma_C$ and $\neg A_k \in D\gamma_D$). By theorem 5.2 we know that there exists a Robinson-resolvent E of C and D that fails at n. Let A' be the corresponding resolved atom. We have to show that there also exists an R_{FILL} - resolvent E' failing at n.

Consider the following cases:

(1) Neither in C nor in D there is a critical pair (L_1, L_2) s.t. L_1 is among the literals resolved upon: In this case $\tau(A') < \tau(B)$ iff $A' <_{vd} B$ for all atoms B in E. By the definiton of the semantic tree, A_k is of maximal term depth within the literals of the refutation sets of the sons of n. A' is an instance of A_k Therefore, by the defining conditions for clauses in sets of E^+, also $\tau(A) \geq \tau(E)$. This implies that E itself is the R_{FILL} - resolvent we are looking for.

(2) There is a critical pair (L_1, L_2) in C, s.t. L_1 is one of the literals resolved upon in C in the derivation of E:

(2a) $\tau(L_1\gamma_C) \geq \tau(L_2\gamma_C)$: This means that $C\gamma_C \in FILL_S(C)$. In this case, consider the $R_{<d}$-resolvents of $C\gamma_C$ and D: They are all in $R_{FILL}(S)$ and, by definition of the semantic tree, at least one of them fails at n.

(2b) $\tau(L_1\gamma_C) < \tau(L_2\gamma_C)$: We show that this cannot happen. Because we assumed that L_1 is one of the literals resolved upon in C we have $L_1\gamma_C = A_k$. The semantic tree is defined s.t. $\tau(A_k) \geq \tau(\{C\gamma_C, D\gamma_D\})$. But this contradicts $\tau(L_1\gamma_C) < \tau(L_2\gamma_C)$.

Q.E.D.

Given theorems 5.10 and 5.11 it is an easy task to prove the following:

<u>THEOREM</u> 5.12.: E^+ is decidable; R_{FILL} combined with the splitting rule provides a decision procedure.

<u>Proof</u>:

The only thing we need to observe is that -- on clause sets in E^+ -- R_{FILL} behaves exactly like $R_{<vd}$ (S) except for the additional generation of ground clauses that are always limited in term depth by the depth of their parent clauses. Thus we conclude from theorem 5.10 that also $R^*_{FILL}(S)$ is finite for all decomposed clause sets S in E^+. Now it follows directly from theorem 5.11 that $R_{<vd}$ yields a decision procedure.

Q.E.D.

5.4 A-ORDERINGS COMBINED WITH SATURATION RULES

5.4.1 An extension of the Skolem class

The initially extended Skolem class is the class of prenex formulas with a prefix of the form $\exists z_1 \dots \exists z_l \forall y_1 \dots \forall y_m \exists x_1 \dots \exists x_n$ s.t. each atom of the matrix has among its arguments either

(i) at least one of the x_i, or

(ii) at most one of the y_i, or

(iii) all of $y_1, \dots, y_m$.

In this section we consider a class which strongly generalizes the initially extended Skolem class. Consider the following definition:

<u>DEFINITION</u> 5.18.: ($\mathcal{S}^+$) A clause set S belongs to $\mathcal{S}^+$ iff for all clauses C in S and all literals L in C:

(i) If t is a functional term occurring in C then $V(t) = V(C)$, and

(ii) $|V(L)| \leq 1$ or $V(L) = V(C)$.

Observe that condition (i) is equivalent to:

(i') If L is functional, then L is covering and $V(L) = V(C)$.

Class $\mathcal{S}^+$ not only contains the initially extended Skolem class but also the initially extended Gödel class (i.e. the prefix class with quantifier prefix type $\exists^* \forall\forall \exists^*$).

$\mathcal{S}^+$ is related to E_1. In fact we have (analogously to lemma 5.8 in section 5.2.1:

$$E \in R_{<_d}(\{C,D\}) \text{ implies } \tau(E) \leq \tau(\{C,D\})$$

if $\{C,D\} \in \mathcal{S}^+$. The only problem is that in general $R_{<_d}(S) \notin \mathcal{S}^+$ for clause sets $S \in \mathcal{S}^+$. Atoms may be generated which, besides covering terms, have arbitrary variables as arguments.

To be able to argue more accurately we define:

DEFINITION 5.19.: An atom or literal A is called essentially monadic on a term t iff t is an argument of A and each other argument is either equal to t or a constant. A is called almost monadic on t iff t is functional and - besides t and constants - also some variables that are not in V(t) occur among the arguments of A. More precisely, A is almost monadic on some functional term t iff $t \in args(A)$, $\forall_{s \neq t \in args(a)}$: s is a constant or a variable $\notin V(t)$.

EXAMPLES: P(g(x), b, g(x)) is essentially monadic on g(x). P(f(f(z)), x, a, f(f(z))) is almost monadic on f(f(z)). P(x, y, a, x) and Q(f(x), f(y)) are neither essentially monadic nor almost monadic (on some subterm).

As mentioned above $R_{<d}$ provides a term depth limit for the resolvents of $\mathcal{S}^+$. But by resolving such clauses, almost monadic atoms may be generated besides covering ones.

EXAMPLE: Let C = {P(x), Q(x, y)} and D = {¬P(f(z)), Q(g(z), z)}. The only $R_{<d}$-resolvent of C and D is E = {Q(f(z), y), Q(g(z), z)}. The first literal of E is almost monadic on f(z); the second literal is covering and essentially monadic.

It is interesting to mention that (for class $\mathcal{S}^+$) almost covering atoms are generated by $R_{<d}$ only if one of the parent clauses is function free, and the atom(s) resolved upon in this clause contain(s) one variable only, whereas other atoms must have also additional variables as arguments. In all other cases all atoms of a resolvent are covering. It would be an easy task to refine the ordering $<_d$ in a way such that it becomes sufficiently restrictive to decide $\mathcal{S}^+$ (and many other classes). We would just have to add

(iv) $V(A) \subset V(B)$ implies $A <_d B$ for function free atoms A and B

to the defining conditions for $<_d$ (definition 5.11). Unfortunately the resulting resolution variant is neither an A-ordering nor an π - ordering strategy.

Special orderings, similiar to this, are treated in chapter 4. Although no ground projection of such orderings exists in general, the resulting strategies may nevertheless be complete on certain classes of clause sets. But this cannot be demonstrated in the realm of the semantic trees. Since, in this chapter, we are interested in strategies that are complete via semantic trees we have to go a different way: We shall combine $R_{<_d}$ with a special saturation rule. For any almost monadic atom we define a corresponding set of essentially monadic atoms.

DEFINITION 5.20.: Let A be almost monadic on some functional term t and Con be some set of constants. The monadization MON(A, Con) consists of atoms that are like A except for replacing each variable that occurs as argument of A (but not of t) by t or some constant in Con. More exactly:

Let $\Sigma_{t,Con}$ be the set of all substitutions of the form

$\{ t_i / x_i \mid x_i \in V(A) - V(t)\}$ where $t_i = t$ or $t_i \in Con$ then

$MON(A,Const) = \{A\sigma \mid \sigma \in \Sigma_{t,Con}\}$.

We extend the definition of MON to clauses and clause sets: If all almost monadic atoms A of a clause C are almost monadic on the same functional term t then $MON(C, Con) = \{C\sigma' \mid \sigma'$ in $\Sigma'_{t,Con}\}$ where $\Sigma'_{t,con}$ is the set of substitutions $\{t_i/x_i \mid x_i \in V(C) - V(t)\}$, s.t. $t_i = t$ or $t_i \in Con$.

If C is function free and there is one and only one variable x, s.t. all literals $L \in C$ with $|V(L)| \geq 2$ contain x then $MON(C, Con) = \{C\sigma \mid \sigma$ in $\Sigma_{x,Con}\}$ where $\Sigma_{x,Con}$ is the set of substitutions $\{t_i/x_i \mid x_i \in V(C) - V(t)\}$, s.t. $t_i = x$ or $t_i \in Con$. In all other cases $MON(C, Con) = \{C\}$.

For any clause set S: $MON(S) = \bigcup_{C \in S} MON(C, CS(S))$
(where CS(S) is the set of all constants occurring in clauses of S).

Remark: As we only use the monadisation operator in the context of finite clause sets S we will implicitly assume that the set of constants occurring in S is always used for the monadisation of atoms or clauses. For sake of readability we therefore suppress the second argument and write MON(A) and MON(C) in stead of MON(A, Con) and MON(C, Con) , respectively. Observe that, in our context, MON(A) and MON(C) are always finite.

EXAMPLES: For all examples we assume that there is just one constant a.
Let $A = P(f(x,y), z)$ then $MON(A) = \{P(f(x,y), f(x,y)), P(f(x,y), a)\}$.
Let $C = \{Q(u,x,f(x,a)), P(u, f(x,a))\}$ then
$MON(C) = \{\{Q(f(x,a), x, f(x,a)), P(f(x,a), f(x,a))\}, \{Q(a,x,f(x,a)), P(a,f(x,a))\}$.
For $D = \{P(x,x), Q(u,a,x), P(x,v)\}$ we have $MON(D) = \{\{P(x,x), Q(x,a,x)\},$
$\{P(x,x), Q(x,a,x), P(x,a)\}, \{P(x,x), Q(a,a,x)\}, \{P(x,x), Q(a,a,x), P(x,a)\}\}$.

We may now define a new resolution variant R_m which is based on $R_{<_d}$.

DEFINITION 5.21: For any clause set S:

$$R_m(S) = MON(R_{<_d}(S)).$$

The members of $R_m(S)$ are called R_m- resolvents.

We state:

LEMMA 5.13.: If $\{C,D\} \in \mathcal{S}^+$ and $E \in R_m(\{C,D\})$ then $\tau(E) \leq \tau(\{C,D\})$.

Proof:
As mentioned above, each atom of an $R_{<_d}$-resolvent of any clause set in $\mathcal{S}^+$ is either almost monadic or covering. There are two cases:

(1) If E is an $R_{<_d}$ - resolvent of $\{C,D\}$ and does not contain almost monadic atoms: Then E is an R_m- resolvent, too, and $\tau(E) = \tau(\{C,D\})$ follows directly from lemma 5.4.
(2) Let E be an $R_{<_d}$-resolvent generated using the resolved atom A and let E contain some almost monadic atom B. In this case one of the parent clauses must be function free and the literals resolved upon in that clause may only contain one variable. Moreover B is the instance of an atom in that clause, containing additional variables. It follows that A is essentially monadic on some term t and B is almost monadic on t. Therefore $\tau(B) = \tau(A)$. By definition we have $\tau(B') = \tau(B)$ for all $B' \in MON(B)$. (The same holds for any almost monadic atom of E). By lemma 5.4 $\tau(A) \leq \tau(\{C,D\})$. Therefore we conclude that $\tau(E') \leq \tau(\{C,D\})$ for all $E' \in MON(E)$.

Q.E.D.

In order to prove that the resolvents of clause sets in $\mathcal{S}^+$ are in $\mathcal{S}^+$ again, we make use of the well known splitting mechanism (see section 2.6). We want to remark that even without splitting R_m decides class $\mathcal{S}^+$. We employ the splitting rule to make the proof somewhat simpler.

LEMMA 5.14.: If S in $\mathcal{S}^+$, then all members of SPLIT(R_m(S)) are in class $\mathcal{S}^+$, too.

Proof:

Let C, D in S and let A be an atom resolved upon to generate some resolvent E in $R_{<d}\{C,D\}$. Let ϑ be the m.g.u. used to get E. (Thus $A\vartheta$ is the resolved atom). We have to consider the following cases:

(1) $A\vartheta$ is function free: By definition of $R_{<d}$ and $\mathcal{S}^+$, this only occurs if both, C and D, are function free. Therefore E is function free, too.

(1a) $|V(A\vartheta)| = 0$:

In this case $V(C\vartheta) \cap V(D\vartheta) = \emptyset$. But this implies that for any two atoms B_1, B_2 in E s.t. $|V(B_i)| > 1$ but $V(B_i) \neq V(E)$ $(i = 1, 2)$ either $V(B_1) \cap V(B_2) = \emptyset$ or, by definition of $\mathcal{S}^+$, $V(B_1) = V(B_2)$. Thus the split components of E fulfill the defining conditions of class $\mathcal{S}^+$.

(1b) $|V(A\vartheta)| = 1$:

Let x be the only element of $V(A\vartheta)$. Then for any literal $L \in E$ s.t. $|V(L)| \geq 2$ we have $x \in V(L)$. Thus, by definition of monadisation, $|V(E')| \leq 1$ for all $E' \in$ MON(E). Moreover, all clauses in MON(E) are function-free, too. This implies that the R_m-resolvents and accordingly also their split components are in class $\mathcal{S}^+$.

(1c) $|V(A\vartheta)| > 1$:

In this case E itself is an R_m - resolvent and fulfills the relevant conditions.

(2) $\tau(A\vartheta) > 0$:

W.l.o.g. we may assume that A is in C and just investigate what happens with literals of C when ϑ is applied.

(2a) $|V(A)| = 1$: If all literals of C contain at most one variable the defining conditions of class $\mathcal{S}^+$ clearly remain satisfied after applying ϑ and splitting. The only interesting case arises when there is some $B \in C$, s.t. $|V(B)| > 1$. As A only contains one variable and - by definition of $\mathcal{S}^+$ - no functional ground term, it follows that $A\vartheta$ is essentially monadic on some functional term t. Therefore $B\vartheta$ is almost monadic on t. By the definition of the monadisation operator $V(B') = V(t) = V(A\vartheta)$ for all $B' \in MON(B\vartheta)$. This holds for all such literals. Therefore $V(A\vartheta) = V(E')$ for all $E' \in MON(E)$, which was to show.

(2b) $|V(A)| > 1$:

It follows from the definition of $\mathcal{S}^+$ that A is covering and that $V(A) = V(C)$. For each $B \in C$ we have either

(i) $V(B) = V(A)$, which implies that $V(B\vartheta) = V(A\vartheta)$, or

(ii) B is not functional and contains just one variable. Call this variable x.

By the proof of lemma 5.6 $\vartheta(x)$ is either a variable, a constant or a functional term t, s.t. $V(t) = V(A\vartheta)$.

Since $A\vartheta$ contains all variables that occur in any functional term of $C\vartheta$ or $D\vartheta$ we have $V(A\vartheta) = V(E')$ for all $E' \in MON(E)$. Therefore the defining conditions of class $\mathcal{S}^+$ remain satisfied for the respective resolvents.

Q.E.D.

We may now state that R_m decides $\mathcal{S}^+$:

<u>THEOREM</u> 5.15.: $\mathcal{S}^+$ is decidable; R_m combined with the splitting rule provides a decision procedure.

<u>Proof</u>:

Lemma 5.13 guarantees a term depth limit for R_m-resolvents of clause sets in $\mathcal{S}^+$. Lemma 5.14 shows that the resolvents (at least after splitting) are in $\mathcal{S}^+$ again. It remains to establish a length limit for the resolvents: By lemma 5.4 and the definition of $\mathcal{S}^+$ it follows that the number of variables of an $R_{<_d}$-resolvent is bounded by the number of variables of its parent clauses. Clearly, this number can never increase through splitting or monadisation. Taking into account the term depth limit, this guarantees a length limit for the R_m-resolvents. This in turn implies that $R_m^*(S)$ is finite for any finite clause set S. Thus the theorem follows from the completeness of R_m on clause sets of $\mathcal{S}^+$ (cf. section 5.4.2 below).

Q.E.D.

5.4.2 Completeness of R_m

To justify the theorem 5.15 it remains to prove the following.

THEOREM 5.16.: The resolution procedure R_m, combined with the splitting rule, is complete for all clause sets S in $\mathcal{S}^+$.

Proof:
By lemma 5.2 our task is to show:
(1) For any unsatisfiable clause set S there is a semantic tree T based on some $K \subseteq \hat{H}_S$ and closed for S s.t. any two clauses which fail immediately below an inference node of T form an R_m-resolvent which fails at the inference node.

For any clause set S in $\mathcal{S}^+$ we define enumerations A_0, $A_1, \ldots$ of arbitrary subsets of the corresponding Herbrand base s.t. less deep atoms precede deeper ones. Within atoms of equal depth essentially monadic atoms precede those atoms that are not essentially monadic.

More formally we have:

(i) if $\tau(A_i) < \tau(A_j)$ then $i < j$, and
(ii) if $\tau(A_i) = \tau(A_j)$ and $i < j$ then A_i is essentially monadic whenever A_j is essentially monadic.

Note that "$<_d$" is compatible with all such enumerations. Let $S \in \mathcal{S}^+$ be unsatisfiable and let T be a semantic tree for some subset K of $\hat{H}_S$ closed for S s.t. T is based on such an enumeration. (Theorem 5.1 guarantees the existence of T). Let C and D be clauses in S failing immediately below, but not at, a node n of T. Let A_k and $\neg A_k$ be the two ground literals labeling the edges leaving the inference node n. Because "$<_d$" is an A-ordering there exists an $R_{<_d}$-resolvent E of C and D, s.t. A_k is a ground instance of the resolved atom A, which fails at n. By definition of $\mathcal{S}^+$ and lemma 5.4 each atom of E either is covering or almost monadic on some term. If for all atoms B of E, B is covering and either $|V(B)| \leq 1$ or $V(B) = V(E)$ then E is an R_m-resolvent, too, and the theorem clearly holds.

There are two crucial cases:

(1) The $R_{<_d}$ - resolvent E contains literals that are almost monadic on some term: This may only be the case if A is essentially monadic on some term t, and all atoms of E are either almost monadic on t or covering. We have to show that some clause in MON(E), fails at node n.

Let γ be the substitution s.t. $E\gamma$ is in the refutation set of n. Clearly $A\gamma$ is essentially monadic on $t\gamma$. For any almost monadic atom B of E, $B\gamma$ is of the same depth as $A\gamma$ because all functional arguments of A as well as of B are equal to t. Therefore, by definition of the enumeration of the Herbrand base, also $B\gamma$ is essentially monadic. This implies that $B'\gamma = B\gamma$ for some $B' \in MON(B)$. Thus some $E' \in MON(E)$ fails at node n, too.

(2) E is function free and for some literal L in E we have $|V(L)| > 1$ but $V(L) \neq V(E)$ (i.e. condition (ii) of the definition of class $\mathcal{S}^+$ does not hold). There are two subcases:

(2a) The resolved atom A is ground: In this case we can split E into components that fulfill the relevant conditions.

(2b) A contains just one variable x:
Then all atoms of E with more than one variable contain x, too. By definition of MON all variables of E are replaced by x or by constants. As A is function free and $|V(A)| = 1$ we know that $A\gamma$ is essentially monadic on some term t. Clearly $\gamma(x) = t$. Therefore, by the construction of T, also $L\gamma$ is essentially monadic on t. Observe that, by definition of $\mathcal{S}^+$, for every literal $M \in E$, s.t. $|V(M)| = 1$ there is some literal L, s.t. $V(M) \subset V(L)$ and $x \in V(L)$. Therefore all literals of $E\gamma$ are essentially monadic on t or some constant. It follows, like in case (1), that $E'\gamma = E\gamma$ for some $E' \in MON(E)$.

Q.E.D.

5.5 PURE SATURATION STRATEGIES

5.5.1 The Bernays – Schönfinkel class

As already mentioned in section 3.2 the Bernays-Schönfinkel class is the class of prenex formulas with quantifier prefixes of type $\exists^*\forall^*$ (cf. [BS28]). This corresponds to the class of function free clause sets, which we shall call BS* in the following. Since the Herbrand universe is always finite for such clause sets the decidability of BS* is a trivial consequence of Herbrand's theorem. Nevertheless there seems to be no direct possibility to use a resolution strategy as a decision procedure. We have already seen in section 3.2 that no semantic resolution refinement (in the sense of chapter 3) can decide BS*.

The following example will make clear that also A-ordering refinements combined with the usual deletion strategy cannot always limit the length of the resolvents on class BS*.

<u>EXAMPLE</u>: Let $E_2 = \{P(x_0, x_1), P(x_1, x_2), \neg P(x_2, x_0)\}$ and $D = \{P(y_2, y_0), P(y_2, y_3), \neg P(y_3, y_0)\}$. For all $n > 2$:

$$E_n = \{P(x_0, x_1), P(x_1, x_2), \dots, P(x_{n-1}, x_n), \neg P(x_n, x_0)\}$$

is a resolvent of E_{n_1} and D. Observe that no atom ordering restriction that has a ground projection can prevent the generation of this sequence of resolvents. Moreover, for no $i \neq j$ E_i subsumes E_j nor is any clause E_i a tautology. This demonstrates that we cannot use A-ordering strategies to decide BS.

For sake of completeness we mention that for any S we might of course generate explicitly the finite set S_g of all Herbrand instances of the clauses in S and afterwards apply resolution to this set of ground clauses to check whether S is satisfiable or not. Replacing clauses by certain instances of it, is what we call a saturation rule. But if there are many different constants and some predicate symbols of higher arity in S, this saturation strategy clearly is impracticable. We just mention that more feasible strategies could be achieved by combining semantic resolution or A-ordering refinements with (partial) saturation.

Such a strategy is presented in chapter 3.2, where BS* is translated to PVD (via partial saturation) and afterwards decided by semantic clash resolution.

5.5.2 Uniform atoms and clauses

To support a concise formulation of the decision class to be stated in the next subsection we introduce some additional terminology:

DEFINITION 5.22.: Two terms t_1, t_2 are called congruent iff both terms are functional and the sequence of arguments of t_1 is a permutation of the sequence of arguments of t_2.

DEFINITION 5.23.: An atom or literal A is called uniform iff

(i) A is function free, or

(ii) there is a functional argument t of A, s.t. each argument of A is either congruent to t, an argument of t or a constant.

EXAMPLES. $P(a, x, f(x,y), g(y,x))$ and $P(u, u, a, v)$ are uniform literals, but $Q(z,f(x,y))$ is not uniform.

Remark. The definition of uniform literals generalizes the concept of argumental literals as introduced by Joyner [Joy76]: It is easy to show that all argumental literals are uniform.

DEFININITION 5.24.: A substitution σ s.t. $rg(\sigma)$ consists of variables and constants only is called flat.

We state some simple properties of uniform atoms:

LEMMA 5.17.:

(1) If ϑ is an m.g.u. of two uniform atoms A, B then $A\vartheta$ ($= B\vartheta$) is uniform, too.

(1) If ϑ is an m.g.u. of two uniform and functional atoms A, B then ϑ is flat.

Proof.
Follows immediately from definition 5.23.

Resolving two uniform clauses does not always result in a uniform clause again. The following example will help to unterstand the situation.

EXAMPLE. Let C = { P(x,y), Q(x,z), Q(z,w) } and D = { ¬P(f(u,v),v), P(g(v,u),u)}. C and D are uniform but their resolvent E = {Q(f(u,v),z), Q(z,w), P(g(v,u),u)} is not uniform, because there are variables among the arguments of the literals that do not occur in the *functional terms* f(u,v) and g(u,v), respectively.

5.5.3 The classes U and M^+

Consider the following class:

DEFININITION 5.25.: A clause set S belongs to U iff for all clauses C in S:

(i) Each L in C is uniform,

(ii) $\tau(C) \leq 1$ (i.e. there is *no nesting* of function symbols).

Clearly U is an undecidable class: It, e.g., contains the clause form of the Skolem class, i.e., the class of prenex formulas (without function symbols) with prefix of type $\exists^*\forall^*\exists^*$. However we shall demonstrate the following subclass of U to be decidable.

DEFININITION 5.26.: A clause set S belongs to M^+ iff S in U and all clauses C in S are Krom, i.e. $|C| \leq 2$.

M^+ is an extension of the Maslov class, i.e. the class of prenex formulas that are Krom and have a Skolem-type prefix (i.e., $\exists^*\forall^*\exists^*$). Obviously, we arrive at clause sets in M^+ if we Skolemize formulas of the Maslov class. Maslov formulas and hence also clause sets in M^+ can be used to encode quite naturally computations for different machine models (see e.g. [DL83]).

To decide M^+ we define a kind of resolution variant R_U which replaces each ordinary resolvent by all of its instances that fulfill the defining conditions of U.

DEFININITION 5.27.: For any clause set S the set of R_U- resolvents of S, $R_U(S)$, is the set of all clauses $E\sigma$, where E is an R-resolvent of clauses in S and σ is a substitution based on S s.t. $\{E\sigma\} \in$ U.

Resolution strategies R' are usually defined such that, for any clause set S, the set of resolvents R'(S) is a proper subset of R(S) (i.e. the set of Robinson resolvents). In contrast, $R_U(S)$ contains certain instances of R-resolvents. As any clause subsumes each of its instances this provides a correct deduction mechanism. But it is quite obvious that R_U is not complete for arbitrary clause sets; i.e. the empty clause may not be derivable from S although S is unsatisfiable. However we shall show that R_U is complete as long as we consider clause sets in U only. This has the interesting consequence that although U is an undecidable class we only have to take into account resolvents of a limited term complexity (depth 1). Of course this implies that there is no recursive bound on the length of resolvents needed to refute a clause set in U using R_U.

Observe that each clause set of M^+ is finite - modulo renaming of variables - as long as the number of different constants, predicate and function symbols is finite. Therefore, by iteratively generating all R_U -resolvents starting with a clause set S in M^+, we arrive at a finite set $R_U^*(S)$ of clauses s.t. no new R_U -resolvents can be derived from $R_U^*(S)$. If $R_U^*(S)$ contains the empty clause then S is unsatisfiable, otherwise S is satisfiable, provided that R_U is complete for clause set S. In other words, the following theorem is a corollary to theorem 5.21 which we shall prove in section 5.5.4 below:

<u>THEOREM</u> 5.18.: R_U - resolution provides a decision procedure for M^+.

From a practical point of view the resolution variant R_U, although definitely simpler than Joyner's correponding R_3 (c.f. [Joy76]), is not very satisfying since it involves the finding of certain substitutions that are not related to the m.g.u.'s. Fortunately, it follows from our results that there is an even simpler way to decide class M^+.

<u>DEFININITION</u> 5.28.: For any clause set S the set of depth-1-resolvents of S is defined as $R_{\leq 1}(S) = \{E \mid E \in R(S) \ \& \ \tau(E) \leq 1\}$.

In contrast to R_U, R_1 is an ordinary resolution refinement in the sense that $R_{\leq 1}(S) \subseteq R(S)$. Clearly, R_U-resolvents are instances of $R_{\leq 1}$ -resolvents. The following proposition expresses the fact that resolvents of instances are instances of resolvents (of the original clauses):

PROPOSITION 5.19.: Let $E \in R(\{C\sigma, D\vartheta\})$ then there exists an $E' \in R(\{C,D\})$ s.t. E is an instance of E'.

Thus we conclude that also $R_{\leq 1}$-resolution is complete on U if R_U is so. It then follows

THEOREM 5.20.: $R_{\leq 1}$ provides a decision procedure for M^+.

5.5.4 Completeness of R_U

In order to prove the (refutational) completeness of R_U for class U we make use of the concept of semantic trees. We assume the reader to be familiar with this important device (see e.g. [KH69], but briefly review the terminology.

For any clause set S the Herbrand base of S is the set of all ground atoms consisting of predicate, function symbols and constants occurring in S (If there are no constants in S we just introduce one). A semantic tree of S is a binary tree with elements of the Herbrand base and their negations labeling the edges. Let A_0, A_1, A_1, ... be an enumeration of the Herbrand base of S. The two edges leaving the root of the corresponding semantic tree T are labeled A_0 and $\neg A_0$, respectively; and if either A_i or $\neg A_i$ labels the edge entering a node, then A_{i+1} and $\neg A_{i+1}$ label the edges leaving that node. With each node of T we associate a refutation set, consisting of the literals labeling the edges of the path from the root down to this node.

A clause C in S fails at a node of T if the complement of every literal in some ground instance $C\gamma$ of C is in the refutation set of that node. A node k is a failure node for S if some clause of S fails at k but no clause of S fails on a node above k (i.e. a node with a shorter path to the root). A node is called inference node if both of its sons are failure nodes. T is closed for S if there is a failure node for S on every branch of T.

A completeness proof for a resolution strategy R' applied to clause sets S essentially rests on the following well known fact:

(1) If S is unsatisfiable, every semantic tree T for S is closed.

Our task is to show:

(2) There is a semantic tree T for S s.t. any two clauses which fail immediately below an inference node of T form an R'—resolvent which fails at the inference node.

Given (1) and (2) the completeness of R' follows by well known arguments (see e.g. [KH69]). Observe that T can be based on an arbitrary enumeration of the Herbrand base. Whereas any enumeration may be used to prove the completeness of Robinsons original resolution strategy, we will have to make a more judicious choice of the enumeration.

THEOREM 5.21.: Resolution procedure R_U is (refutationally) complete for all clause sets S in U.

Proof.
For any clause set S we define an enumeration A_0, A_1, ... of the corresponding Herbrand base s.t. deeper atoms succeed less deep ones. Within atoms of equal depth uniform atoms preceed those atoms that are not uniform. More formally we have:

(i) if $\tau(A_i) < \tau(A_j)$ then $i < j$, and
(ii) if $\tau(A_i) = \tau(A_j)$ and $i < j$ then A_j is uniform implies that A_i is uniform, too.

By the completeness of (Robinson's) resolution with respect to semantic trees it holds that, for any clauses C and D failing immediately below, but not at, a node k of T, there exists an R-resolvent E of C and D which fails at k. In the following ϑ denotes the m.g.u. of atoms in C and D used to resolve E. We show that, for C, D in U, there also exists an R_U-resolvent E' failing at node k.

We have to investigate the following cases:

(1) E is function free ($\tau(E) = 0$): Since any function free clause satisfies the defining conditions of U, E itself is the R_U-resolvent we are looking for.

(2) $\tau(E) > \max(\tau(C), \tau(D))$: We show that this cannot happen. By lemma 5.17 the m.g.u. of two uniform, functional atoms of depth 1 is flat. Thus, the greater term depth of the resolvent can only arise if there is some function

free atom A among the atoms resolved upon and if $\vartheta(x)$ is functional for some $x \in V(A)$. Moreover there has to exist a functional atom B, not to be resolved upon but in the same clause as A s.t. x occurs in a functional term of B (i.e., $\tau_{max}(x,B) = 1$). Since $A\vartheta$ is uniform we have $V(A\vartheta) = V(\vartheta(x))$ and consequently $V(A\vartheta) \subseteq V(B\vartheta)$. Again by the uniformity of $A\vartheta$, it follows that

$$\tau(A\vartheta) < \tau(B\vartheta)$$

and thus also $\tau(A\vartheta\gamma) < \tau(E\gamma)$, where γ is the ground substitution s.t. $E\gamma$ is in the refutation set corresponding to node k. ($A\vartheta$ is the resolved atom). On the other hand the semantic tree is defined s.t. the ground instance of the resolved atom is at least as deep as all atoms in the refutation set at node k. This means that

$$\tau(A\vartheta\gamma) \geq \tau(E\gamma).$$

The contradiction implies that the resolution procedure remains complete if we discard resolvents that are deeper than their parent clauses.

By definition, $\max(\tau(C),\tau(D)) \leq 1$. It remains to investigate the following case:

(3) $\tau(E) = 1$: There are two subcases:

(3a) All literals in E are uniform: Then, like in case (1), E itself is an R_U-resolvent.

(3b) There are non-uniform literals in E: By lemma 5.17 ϑ is flat if in each parent clause there are functional literals among the literals resolved upon. If A is uniform then $A\vartheta$ is uniform, too, whenever ϑ is flat. Since not all literals of E are *uniform* it follows that all literals resolved upon in one of the parent clauses - w.l.o.g. say in C - are function free. Let M denote this subset of C. There has to exist some x in V(M) s.t. $\vartheta(x) = s$ is functional and x in V(B) for some function free B in C - M but $V(B\vartheta) \not\subseteq V(M\vartheta)$. (Since $V(s) = V(M\vartheta)$ we then know that $B\vartheta$ is a non - uniform literal of E.) Again, let γ be the ground substitution s.t. $E\gamma$ is in the refutation set corresponding to node k. By the definition of the enumeration of the Herbrand universe not only the single literal in $M\vartheta\gamma$, but all literals in $C\vartheta\gamma$ that are of depth $\tau(M\vartheta\gamma)$ are uniform; in particular also $B\vartheta\gamma$ is uniform. We shall specify a substitution σ s.t. $E\sigma\gamma = E\gamma$ and $B\vartheta\sigma$ is uniform and of depth ≤ 1.

Let $y \in V(B\vartheta) - V(s)$. The crucial fact is that (the only literal in) $M\vartheta\gamma$ and consequently also $B\vartheta\gamma$ is uniform on $s\gamma$. Therefore we know that either $\gamma(y) = c$ (for some constant c) or $\gamma(y) = t_i\gamma$, where t_i is an argument of s, or $\gamma(y) = s\gamma$. Corresponding to these cases we define $\sigma(y) = c$ or $\sigma(y) = t_i$ or $\sigma(y) = s$, respectively. Analogously we define $\sigma(z)$ for all variables z of $B\vartheta$ that do not occur in s. For all other variables let $\sigma(x) = x$. Clearly, $B\vartheta\sigma$ is uniform and of depth ≤ 1. We have to show that all other literals of $E\sigma$ remain of depth ≤ 1, too. Suppose $\tau(L\sigma) > 1$ for some L in E; then $\sigma(x) = s$ for some x that occurs as argument of a functional term in L. Since $\tau(s\gamma) = \tau(M\vartheta\gamma)$ we would then have $\tau(M\vartheta\gamma) < \tau(L\sigma\gamma)$. But by definition of the semantic tree we know that $\tau(M\vartheta\gamma) = \tau(E\gamma) = \tau(L\sigma\gamma)$. Therefore $\tau(E\sigma) = 1$.

There might still be literals in $E\sigma$ which are not yet uniform. For each such literal we proceed in the same way as for $B\vartheta$. Combining the corresponding substitutions we arrive at an instance $E\sigma'$ of E s.t. $E\sigma'\gamma = E\gamma$ but $\{E\sigma'\}$ in U.

Q.E.D.

It is interesting to observe that in the proof above we did not make use of the fact that the clause sets in M^+ are Krom. This implies that R_{M^+} remains complete for the class of clause sets defined by conditions (ii) and (iii) of definition 5.25 only. Of course this class is undecidable; nevertheless R_{M^+} provides a term depth limit for the resolvents. This emphasizes the significance of a length limit for the clauses.