1 Theorem Proving with Equality

In this chapter we discuss rules for theorem proving with equality. Equality can be efficiently handled in the context of resolution by using the *paramodulation rule*. Another option would be axiomatize the equality predicate and to use the standard resolution rule, but this is less efficient.
It is possible to axiomatize equality in first order logic. In order to do this, the following axioms have to be added:

EQREFL $\forall x(x \approx x)$.

EQTRANS $\forall xyz(x \approx y \wedge y \approx z \rightarrow x \approx z)$.

EQSYMM $\forall xy(x \approx y \rightarrow y \approx x)$.

EQREPL For each predicate symbol p with arity n, the following axiom has to be added:

$$\forall x_1 y_1 \cdots x_n y_n(\ x_1 \approx y_1 \wedge \cdots \wedge x_n \approx y_n \wedge p(x_1, \ldots, x_n) \rightarrow p(y_1, \ldots, y_n)\ ).$$

EQFUNC For each function symbol f with arity n, the following axiom has to be added:

$$\forall x_1 y_1 \cdots x_n y_n(\ x_1 \approx y_1 \wedge \cdots \wedge x_n \approx y_n \rightarrow f(x_1, \ldots, x_n) \approx f(y_1, \ldots, y_n)\ ).$$

It is in general not possible that the user enters EQFUNC-axioms because they include the Skolem functions. If the user would enter EQFUNC-axioms for the Skolem functions the theorem prover would choose different functions. *** Wat is hier aan de hand? Kan het ook zonder?

There are several reasons not to take this approach.

- The EQREFPL and EQFUNC axioms can resolve with each other, causing enumeration of the Herbrand base.
- The equality predicate has a special logical status, because it is supposed to mean identity.
- The equality predicate is very frequent. In most mathematical theorems equality occurs.

For this reason we introduce a dedicated equality rule, the *paramodulation rule*. Using the paramodulation rule it is possible to delete the EQREPL axioms, and the EQFUNC axioms.

PARAMOD We define the *paramodulation rule* as follows: If $c_1 = \{t_1 \approx t_2\} \cup R_1$ and $c_2 = \{A[t_1']\ \} \cup R_2$ are clauses, such that t_1' and t_1 are unifiable, and Θ is the mgu, then the clause $\{A\Theta[t_2\Theta]\ \} \cup R_1\Theta \cup R_2\Theta$ is a *paramodulant* of c_1 and c_2. The clause c_1 is called the *from*-clause, and clause c_2 is called the *into*-clause.

We give some examples:

Example 1.1 • The clauses $c_1 = \{1 + 1 \approx 2\}$ and $c_2 = \{\sin(X)^2 + \cos(X)^2 \approx 1\}$ have the following paramodulants.

c_1 into c_2 :

$$\{\sin(X)^{1+1} + \cos(X)^2 \approx 1\},$$
$$\{\sin(X)^2 + \cos(X)^{1+1} \approx 1\},$$

c_1 into itself:

$$\{2 \approx 2\}, \{1 + 1 \approx 1 + 1\},$$

c_2 into c_1 :

$$\{\sin(X)^2 + \cos(X)^2 + 1 \approx 2\},$$
$$\{1 + \sin(X)^2 + \cos(X)^2 \approx 2\},$$

c_2 into itself:

$$\sin(X)^2 + \cos(X)^2 \approx \sin(Y)^2 + \cos(Y)^2\},$$
$$\{1 \approx 1\}.$$

- Consider clauses $c_1 = \{\neg nat(X), X + 0 \approx X\}$ and $c_2 = \{\neg nat(Y), 0 + Y \approx Y\}$. When using c_1 as from clause the variable X is unifiable with every subterm of c_2 :

$$\{\neg nat(X), \neg nat(X + 0),) + X \approx X\},$$
$$\{\neg nat(0), \neg nat(X), (0 + 0) + X \approx X\},$$
$$\{\neg nat(X), 0 + (X + 0) \approx X\},$$

The unrestricted paramodulation rule is a large improvement over the axiomatization approach, but it is still hopelessly inefficient, as is evident from the last example. Fortunately we can apply ordering refinements, like in Chapter **??**. It is possible to impose the condition that the from-literal, and the into-literal are maximal in their clauses. Moreover it is possible to use the order to direct paramodulation. In that case the replacement of t_1 by t_2 is only possible, if t_2 is not greater than t_1 under the given order. Some forms of tautology elimination can be used, but we have to be careful.
It is easy to see that the paramodulation rule can replace the EQREPL axioms, because it makes the same type of replacement. It is however not so easy to prove that the EQFUNC axioms can be deleted. The problems are caused by lifting. In Chapter ref4 we proved the completeness of refinements by first proving their completeness for ground clauses, and by secondly proving that ground proofs can be lifted to non-ground proofs. This does not work with paramodulation

because it is impossible to reproduce reproduce paramodulation steps that take place in substituted terms. Consider the ground paramodulation step:

$$\{p(s(a)), q(s(a))\}, \{a \approx b\} \Rightarrow \{p(s(b)), q(s(a))\}.$$

If $\{p(s(a)), q(s(a))\}$ is replaced by the non-ground clause $\{p(X), q(X)\}$, it is not possible to derive a clause that has $\{p(s(b)), q(s(a))\}$ as an instance using paramodulation alone. One way out would be to keep the EQFUNC axioms. Using $\{\neg a \approx b, s(a) \approx s(b)\}$, one could derive $\{s(a) \approx s(b)\}$. This clause can be paramodulated into $\{p(X), q(X)\}$ with result $\{p(s(b)), q(s(a))\}$. However we do not want the EQFUNC axioms because of their ability to resolve with themselves. Instead we show the completeness of another variant of paramodulation, which we call *paramodulation with solidarity.* This other variant of paramodulation is not practical, but it can be lifted into the paramodulation rule given above. Moreover it can be combined with orders, given that the order is of a certain type, called *reduction order.* Before we can do this, we must first study equality a bit better on ground clauses.

2 Ground Unit Clauses

In this section we study the satisfiability problem for ground atoms with ground equations. As is the case in logic without equality this problem is related to the implication problem. A set of axioms Γ implies a formula F if and only if $\Gamma, \neg F$ is unsatisfiable. The satisfiability problem for ground atoms with ground equations is harder than the satisfiability problem without equations, because one has to take the term equivalences induced by the equations.
Examples of non-satisfiable sets are:

$$a \approx b, b \approx c, c \approx d, \neg a \approx d,$$

$$x + y \approx y + x, x + s(y) \approx s(x + y), s(y) + x \approx s(y + x), \neg x + s(y) \approx s(y) + x,$$

$$a \approx s(s(a)), s(s(a)) \approx s(t(a)), t(a) \approx b, t(s(b)) \approx c, \neg s(c) \approx a.$$

The following rules give a calculus in which the unsatisfiability of ground atoms with equations can be proven. We assume that $t_1 \approx t_2$ and $t_2 \approx t_1$ are the same literal. We also assume that $\neg t_1 \approx t_2$ and $\neg t_2 \approx t_1$ are the same literal.

RES If Γ contains a pair $\{A\}, \{\neg\, A\}$ then Γ is unsatisfiable.

PARAMOD If Γ contains a pair $\{t_1 \approx t_2\}$, $\{A[t_1]\}$, then add $\{A[t_2]\}$ to Γ. Here $\{A[t_1]\}$ is a literal, positive or negative, equality or non-equality, that contains t_1.

EQREFL If Γ contains a literal of the form $\{\neg\, t \approx t\}$, then Γ is not satisfiable.

We apply the calculus on the first of the examples above.

(1)	$\{a \approx b\}$	initial
(2)	$\{b \approx c\}$	initial
(3)	$\{c \approx d\}$	initial
(4)	$\{\neg a \approx d\}$	initial
(5)	$\{a \approx c\}$	paramod(1,2)
(6)	$\{b \approx d\}$	paramod(2,3)
(7)	$\{\neg b \approx d\}$	paramod(1,4)
(8)	$\{\neg a \approx c\}$	paramod(3,4)

At this point both $(5, 8)$ and $(6, 7)$ resolve into a contradiction.
The rules do not provide a decision procedure, because in general an infinite number of literals can be derived. In the last example it is possible to derive an infinite number of literals from $a \approx s(s(a))$ alone, by self paramodulation. Also the rules are very redundant, this can be seen from the way in which the first example is solved. In the last example the following clauses can be derived, by using the a of the first rule $a \approx s(s(a))$:

$$a \approx s(s(s(s(a)))), s(s(s(s(a)))) \approx s(t(a)),$$

$$\{s(s(a)) \approx s(t(s(s(a))))\}, \{t(s(s(a))) \approx b\}, \{\neg s(c) \approx s(s(a))\}.$$

The other paramodulants of the first generation are

$$\{a \approx s(t(a))\}, \{s(s(a)) \approx s(b)\}, \{t(s(t(a))) \approx c\}, \{\neg s(t(s(b))) \approx a\}.$$

Definition 2.1 A *reduction order* is a relation on terms satisfying the following conditions:

O1 $\forall x(x \not\sqsubset x)$.

O2 $\forall xyz(x \sqsubset y \wedge y \sqsubset z \rightarrow x \sqsubset z)$.

WF In every set of terms T over a finite signature there is at least one minimal term t. This is a term $t \in T$, such that there is no $t' \in T$ with $t' \sqsubset t$.

CONT Relation $\sqsubset$ is preserved in contexts: If $x \sqsubset y$, then $A[x] \sqsubset A[y]$.

O1 and O2 state that $\sqsubset$ is an order. Property WF states that $\sqsubset$ is *well-founded*, and that it can be used in induction proofs. We will use this property later. The most important reduction order for theorem proving is the *Knuth-Bendix* order. [?] Other orders are *recursive path* orders, and *lexicographic path* orders. In this section we study only ground terms, and we don't need the LIFT property. Before the Knuth-Bendix order can be defined, we need the following recursively defined term order:

Definition 2.2 Let $\sqsubset$ be some order on the signature. (We assume that a function symbol does not occur twice with the same arity) Order $\sqsubset$ can be extended to an order on terms as follows:

- If $f \sqsubset g$, then $f(t_1, \ldots, t_n) \sqsubset g(u_1, \ldots, u_m)$.
- If $u_l \sqsubset v_l$, then $f(t_1, \ldots, t_{l-1}, u_l, \ldots, u_n) \sqsubset f(t_1, \ldots, t_{l-1}, v_l, \ldots, v_n)$.

The order $\sqsubset$ is total on ground terms. It satisfies the liftability property. It can be easily extended into a liftable A-order.

Lemma 2.3 Let $<$ be an order on some set S. Let S^* be the set of tuples of finite length, with elements in S. Let $<<$ be defined on S^* as follows:

- $s << t$ if $s_i < t_i$, on the smallest position i, where $s_i \neq t_i$.
- $s << t$ if s is an initial segment of t.

Then $<<$ is an order on S^*.

$<<$ is called the *lexicographic* extension of $<$. It is the order with which telephone books and dictionaries are ordered.

Theorem 2.4 $\sqsubset$ is a total order on ground terms, and it is preserved in contexts.

proof:

- It is easily seen that $\sqsubset$ is irreflexive, because none of the cases of Definition **??** does apply.
- Next we show that $\sqsubset$ is transitive. Assume that $x \sqsubset y$ and $y \sqsubset z$. We show by induction on $\#x + \#y + \#z$ that $x \sqsubset z$.

  If both $x \sqsubset y$ and $y \sqsubset z$ are due to the first case, then we can write

  $$f(t_1, \ldots, t_n) \sqsubset g(u_1, \ldots, u_m), \text{ and}$$

  $$g(u_1, \ldots, u_m) \sqsubset h(v_1, \ldots, v_p),$$

  with $f \sqsubset g$ and $g \sqsubset h$. Then $f \sqsubset h$, and $f(t_1, \ldots, t_n) \sqsubset h(v_1, \ldots, v_p)$ by the first case.

  If $x \sqsubset y$ by the first case, and $y \sqsubset z$ by the second case, then we can write

  $$f(t_1, \ldots, t_n) \sqsubset g(u_1, \ldots, u_{l-1}, v_l, \ldots, v_m), \text{ and}$$

  $$g(u_1, \ldots, u_{l-1}, v_l, \ldots, v_m) \sqsubset g(u_1, \ldots, u_{l-1}, w_l, \ldots, w_m).$$

  Since $f \sqsubset g$, we get

  $$f(t_1, \ldots, t_n) \sqsubset g(u_1, \ldots, u_{l-1}, w_l, \ldots, w_m).$$

The converse case, where $x \sqsubset y$ by the second case, and $y \sqsubset z$ by the first case, goes analogously.

In that case we apply Lemma 2.3.

- We show that $\sqsubset$ is total by induction on $\#x + \#y$. Assume that $x \neq y$, and that the first case does not apply. Write $x = f(t_1, \ldots, t_n)$, and $y = f(u_1, \ldots, u_n)$. Because $x \neq y$, there is a smallest i, such that $t_i \neq u_i$. By the induction hypotheses either $t_i \sqsubset u_i$, or $u_i \sqsubset t_i$. It follows by Case 2 that either $x \sqsubset y$, or $y \sqsubset x$.

$\sqsubset$ is in general not a reduction order, although in some cases it might be.

Definition 2.5 The Knuth-Bendix order $\sqsubset_{KBO}$ is defined from the $\sqsubset$-order as follows: Let t_1 and t_2 be terms.

1. If $\#t_1 < \#t_2$, then $t_1 \sqsubset_{KBO} t_2$.
2. If $\#t_1 = \#t_2$, and $t_1 \sqsubset t_2$, then $t_1 \sqsubset_{KBO} t_2$.

At the moment we have defined the KBO-order on ground terms only. In the next section we will extend the KBO-order to non-ground terms.

Theorem 2.6 The relation $\sqsubset_{KBO}$ is a reduction order.

proof: We first prove O1.

PARAMOD Let $\{t_1 \approx t_2\}$ be an equality atom in Γ, such that $t_2 \sqsubset t_1$. Let $\{A[t_1]\} \in \Gamma$ be a literal containing t_1 but not $t_1 \approx t_2$ itself. It is possible to delete $\{A[t_1]\}$ from Γ and replace it by $\{A[t_2]\}$. It is possible to impose a further restriction on A, in the case that A is a positive or negative equality atom of the form $(\neg)u_1 \approx u_2$. The condition is that $u_2 \sqsubset u_1$, and that the term t_1 occurs in u_1.

In the case that A is a non-equality atom one could also impose the condition that t_1 is the leftmost, innermost position of A into which paramodulation is possible.

EQREFL If there is a literal $\{\neg t \approx t\}$ in Γ, then Γ is unsatisfiable.

RES If Γ contains a complementary pair $\{A\}$ and $\{\neg A\}$, then Γ is unsatisfiable.

TAUT If there is an equation of the form $\{t \approx t\} \in \Gamma$, then it can be removed from Γ.

Theorem 2.7 The calculus always terminates.

Theorem 2.8 The calculus is sound and complete.

We now make the first and the last of the examples in the beginning of this section, using the optimized calculus. For the first example we put

$$a \prec b \prec c \prec d.$$

For the third example put:

$$s \prec a \prec t \prec b \prec c.$$

The following literals are the only derivable literals. We have sorted all equalities, such that the greater term comes first:

(1)	$\{s(s(a)) \approx a\}$	initial
(2)	$\{s(t(a)) \approx s(s(a))\}$	initial
(3)	$\{t(a) \approx b\}$	initial
(4)	$\{t(s(b)) \approx c\}$	initial
(5)	$\{\neg s(c) \approx a\}$	initial
(6)	$\{s(s(a)) \approx s(b)\}$	paramod(3,2), 2 is deleted
(7)	$\{s(b) \approx a\}$	paramod(1,6), 6 is deleted
(8)	$\{t(a) \approx c\}$	paramod(7,4), 4 is deleted
(9)	$\{c \approx b\}$	paramod(3,8), 8 is deleted
(10)	$\{\neg s(b) \approx a\}$	paramod(9,5), 5 is deleted
(11)	$\{\neg a \approx a\}$	paramod(7,10), 10 is deleted

Literal 11 results in a contradiction because of the EQREFL-rule. It was possible to derive a contradiction earlier, by applying the RES-rule on literals 7 and 10. One can always obtain the contradiction in two ways if there is a pair of complementary equations.

3 Ground Non-Unit Clauses

In this section we extend the improved calculus of the previous section to the non-unit case. This could be done in a straightforward way, but then the resulting calculus cannot be lifted to the non-ground case, and one would lose most of the simplification rules. The straightforward way would be just to fix some order on the literals, (possibly unrelated to term order), and use this order to sort the clauses. This would work in the ground case.
We want to be able to apply the reduction order within clauses, s.t. we can use ordered resolution as in Chapter **??**. In order to do this we need to extend the reduction order to literals. There are many ways to do this. We present here the easiest.

Definition 3.1 Let $\sqsubset$ be a reduction order. Let $\prec$ be a arbitrary order on all objects of the form f or $\neg f$, where f is a symbol that can occur as predicate symbol. $\prec$ musch satisfy the condition that $\approx$ is minimal. In order to extend $\sqsubset$

to atoms/literals, we first sort all equalities as $t_1 \approx t_2$, with $t_2 \prec' t_1$. Then for literals/atoms, we define $(\neg)f(t_1, \ldots, t_n) \sqsubset (\neg)g(u_1, \ldots, u_m)$ if $(\neg)f \prec (\neg)g$, or $f = g$, and for the smallest i, where $t_i \neq u_i$, it is the case that $t_i \sqsubset u_i$.

The extension of $\sqsubset$ is defined in such a manner that every equality atom $t_1 \approx t_2$ (with $t_2 \sqsubset t_1$) is smaller than every atom $A[t_1]$ that contains t_1.

Example 3.2 Assume that $f \prec a \prec b \prec c \prec g$.

We define the deduction rules: Let $\sqsubset$ be a reduction order, extended to atoms. The rules are separated into *derivation* rules and *simplification* rules. We first describe the derivation rules:

EQREFL Let $c = \{\neg\ t \approx t\} \cup R$ be a clause, s.t. $\neg\ t \approx t$ is maximal in c. Then the clause R can be obtained by equality reflexivity.

RES Let $c_1 = \{A\} \cup R_1$ and $c_2 = \{\neg\ A\} \cup R_2$ be clauses s.t. that A and $\neg\ A$ are maximal in their clauses. Then the clause $R_1 \cup R_2$ is a resolvent of c_1 and c_2.

PARAMOD Let $c_1 = \{t_1 \approx t_2\} \cup R_1$ and $c_2 = \{A[t_1]\} \cup R_2[t2]$ be clauses, s.t. $t_1 \approx t_2$ and $A[t_1]$ are maximal in their clauses. Assume that $t_1 \approx t_2$ and $A[t_1]$ satisfy the conditions of the improved paramodulation rule in Section 2. Then the clause $R_1 \cup \{A[t_2]\} \cup R[t_2]$ is an ordered paramodulant. Here $A[t_2]$ is obtained from $A[t_1]$ by replacing *all* occurrences of t_1 by t_2. Similarly $R_2[t_2]$ is obtained by replacing all occurrences of t_1 in R_2 by t_2.

EQFACT Let $c = \{t_1 \approx t_2, t_1 \approx t_3\} \cup R$ be a clause. Assume that $t_1 \approx t_2$ is maximal in c, that both $t_2, t_3 \sqsubset t_1$, and that $t_1 \approx t_3$ is larger than all literals in R. Then the clause $\{\neg\ t_2 \approx t_3, t_1 \approx t_3\} \cup R$ is an equality factor of c.

In the paramodulation rule there must be one occurrence of t_1 occurring in a maximal literal and at a position into which paramodulation is allowed. There are no further conditions on the other occurrences. The other replacements are called *solidarity replacements*.
The correctness of the EQFACT rule is a consequence of the transitivity of equality: If a literal from R or $t_1 \approx t_3$ is true, then the result is trivially true. Otherwise $t_1 \approx t_2$ is true. If $\neg\ t_2 \approx t_3$ is true, then the result is true. Otherwise $t_2 \approx t_3$ is true. But in that case $t_1 \approx t_3$ follows from the transitivity of $\approx$.
The EQFACT-rule is provable using the other rules, but this does not imply that it is redundant.

Exercise 3.3 Show that the EQFACT-rule is correct by showing that $\Gamma = \{\{R, t_1 \approx t_2, t_1 \approx t_3\}, \{\neg\ R\}, \{t_2 \approx t_3\}, \{\neg\ t_1 \approx t_3\}\}$ can be refuted by the other rules.

Example 3.4 Let $c_1 = \{a \approx b\}$, with $b \sqsubset a$. Let $c_2 = \{p(a,b,a), q(a,b), r(b,a)\}$. Assume that $p(a,b,a)$ is the maximal literal in c_2. The following clauses are paramodulants of c_1 into c_2 :

$$\{p(b,b,b), q(a,b), r(b,a)\},$$

$$\{p(b,b,b), q(a,a), r(b,a)\},$$

$$\{p(b,b,b), q(a,b), r(a,a)\}.$$

By iterating paramodulation it is possible to eventually derive

$$\{p(b,b,b), q(b,b), r(b,b)\}.$$

Next we define the simplification rules, they are special cases of the previous rules, but the result implies one of the parents.

EQREFL Let $c = \{t \approx t\} \cup R$ be a clause. The clause c can be deleted and replaced by R.

PARAMOD Let $c_1 = \{t_1 \approx t_2\} \cup R_1$ and $c_2 = \{A[t_1]\} \cup R_2$ be clauses, s.t. $c_1 \neq c_2$, $t_2 \sqsubset t_1$, and $R_1 \subseteq R_2$, where $A[t_2]$ is the result of replacing all occurrences of t_1 in $A[t_1]$ by t_2. Then c_2 can be deleted and replaced by $\{A[t_2]\} \cup R_2$.

RES Let $c_1 = \{A\} \cup R_1$ and $c_2 = \{\neg A\} \cup R_2$ be clauses, s.t. $R_1 \subseteq R_2$. Then c_2 can be deleted and replaced by R_2. Similarly if $R_2 \subseteq R_1$, then c_1 can be deleted and replaced by R_1.

SUBS If c_1 and c_2 are clauses, s.t. $c_1 \subseteq c_2$, then clause c_2 can be deleted.

Most theorem provers do not implement the simplification rules in their full strength. Usually in the case of resolution, there is the restriction that one of the parents should be a unit clause. In that case the RES-simplification rule is called the *unit resolution* rule. Similarly most provers demand in the case of demodulation that the equality clause is a unit clause. The resulting simplification rule is called *demodulation*. There is no simplification rule based on equality factoring, since $\{\neg\, t_2 \approx t_3, t_1 \approx t_3\} \cup R$ does not imply $\{t_1 \approx t_2, t_1 \approx t_3\} \cup R$.

The PARAMOD-simplification rule would become unsound if there would be solidarity. For example $\{a \approx b, p(a)\}$ and $\{p(a), q(a)\}$ do not imply $\{p(b), q(b)\}$. When there are non-unit equality clauses, the notion of tautology becomes more complicated, since it is possible to have propositional tautologies of the form $\{A, \neg\, A\} \cup R$. Moreover there are new, non-propositional tautologies that arise from the semantics of $\approx$. For example the clauses $\{\neg\, a \approx b, f(b) \approx f(a)\}$ and $\{\neg\, a \approx b, \neg\, c \approx b, f(s(a)) \approx f(s(c))\}$ are tautologies. In general one can define a tautology as a clause for which the (negations of the) negative literals imply one of the positive literals.

Definition 3.5 Let $c = \{A_1, \ldots, A_p\}$ be a clause. Let $\{B_1, \ldots, B_p\}$ be the set of literals defined from $B_i = \neg A_i$ if A_i is positive, and $A_i = \neg B_i$ if A_i is negative. Then clause c is a *tautology* if the unit clause set

$$\Gamma = \{\{B_1\}, \ldots, \{B_p\}\}$$

is unsatisfiable.

This definition covers all the examples, since the unit clause sets

$$\{\neg A\}, \{A\},$$

$$\{a \approx b\}, \{\neg f(b) \approx f(a)\},$$

$$\{a \approx b\}, \{c \approx b\}, \{\neg f(s(a)) \approx f(s(c))\}$$

are unsatisfiable. It also covers tautologies of the form $\{t \approx t\} \cup R$, since all unit clause sets containing $\{\neg t \approx t\}$ are unsatisfiable.
It is easy in general to check that a clause is a tautology using the unit equality calculus of Section 2.
In all the cases (except SUBS) where a simplification rule can be applied, it is also possible to apply a derivation rule. It is obvious that the simplification rule should be preferred, since then one of the parents will be deleted. So an implementation should in principle first try to apply the simplification rules, and only after that the derivation rules.
The simplification rules can be combined with tautology elimination, and with subsumption. If a clause can be simplified into a tautology or into a clause that is subsumed, then it can be eliminated completely.
As was the case in Chapter **??** with subsumption, there are many ways in which the simplification rules can be applied. When a clause, which has just been derived, is simplified, this is called *forward* simplification, (or forward unit-resolution, forward demodulation, etc.) When a kept clause is used to simplify other clauses, this is called *backward* simplification.

1. Always keep the clauses fully simplified. When a clause is created it is forward simplified. When a clause is kept, it is used to simplify the other clauses.

2. Keep only the clauses in **usable** fully simplified. When a clause is generated it is forward simplified. Kept clauses are not used for backward simplification.

3. Don't do any backward simplification at all. This is an acceptable option in many cases, and it is certainly easy to implement. However there is a risk of missing important simplifications.

4 Non-Ground, Non-Unit Clauses

We are now ready for the full first order case. Think about how to deal with the order.

We need to extend the Knuth-Bendix order to the non-ground case:

Definition 4.1 Let t_1 and t_2 be terms. It is the case that $t_1 \sqsubset_{KBO} t_2$ iff one of the following holds:

1. $\#t_1 < \#t_2$, and each variable that occurs in t_1 occurs at least as often in t_2.
2. $\#t_1 = \#t_2$, each variable has the same number of occurrences in t_1 as it has in t_2, and $t_1 \sqsubset t_2$.

Theorem 4.2 $t_1 \sqsubset t_2$ iff for all ground instances $t_1\Theta,\ t_2\Theta$,

$$t_1\Theta \sqsubset_{KBO} t_2\Theta.$$

Exercise 4.3 One can eliminate equality in two ways. One supports subsumption, the other does not.