2 Allgemeine Gleichheitsverfahren

(Karl-Hans Bläsius, Axel Präcklein)

Unter dem Gesichtspunkt einer möglichst breiten Anwendbarkeit sind in diesem Teilgebiet zahlreiche Kontrollstrategien und Kalküle zur besonderen Behandlung der Gleichheit entwickelt worden (z.B. [WRCS67], [RW69], [Sib69], [Mor69], [KB70], [Bra75], [Sho78], [HR78], [Dig79], [HO80], [Pet83], [LH85], [HR86], [Rus87], [ZK88], [BG90]), die im Rahmen dieses Buches nicht alle vorgestellt werden können. Stattdessen wird an einigen ausgewählten Methoden eine gewisse Entwicklungslinie aufgezeigt.

2.1 Untertermersetzung: Paramodulation

Wie im vorigen Abschnitt ausgeführt, beruht eine natürliche, in der Mathematik häufig verwendete Regel zur Behandlung der Gleichheit auf dem Prinzip der *Untertermersetzung*: In beliebigem Kontext kann Gleiches durch Gleiches ersetzt werden. Ähnlich wie beim Modus Ponens ist auch hierbei in vielen Fällen die Anwendung einer Instantiierungsregel nötig, um die Voraussetzungen für die Anwendbarkeit der Untertermersetzung herzustellen.

Die Einführung des Unifikationsprinzips durch J.A. Robinson [Rob65] macht willkürliche Instantiierungen überflüssig und ermöglicht stattdessen ein zielgerichtetes Instantiieren auf der allgemeinsten Ebene. So wie der Modus Ponens unter Verwendung des Unifikationsprinzips zur Resolution verallgemeinert wurde, haben G. Robinson und L. Wos auch die Gleichheitsersetzungsregel zur Paramodulationsregel verallgemeinert [RW69].

Formal wird die Paramodulationsregel wie folgt definiert:

Regel: Seien $C_1 := \{L_1, L_2, ..., L_n\}$ und

$C_2 := \{l = r, K_2, ..., K_m\}$

zwei Klauseln. Falls L_1 den Unterterm s enthält und s und l unifizierbar sind mit dem allgemeinsten Unifikator σ (d.h. σs und σl sind syntaktisch identische Terme), dann ist die Klausel

$\sigma\{L_1', L_2, ..., L_n, K_2, ..., K_m\}$

Paramodulant der Klauseln C_1 und C_2, wobei L_1' aus L_1 durch Ersetzen des Terms s durch den Term r entsteht.

Beispiel: Seien $\{P(c, h(f(a, y), b))\}$ und

$\{f(x, e) = g(x),\ Q(x)\}$ zwei Klauseln,

dann ist $\{P(c, h(g(a), b)),\ Q(a)\}$ Paramodulant der beiden Klauseln. Die Terme f(a, y) und f(x, e) sind unifizierbar mit $\{x \leftarrow a, y \leftarrow e\}$. Diese Substitution muß auf die neue Klausel angewendet werden.

Paramodulation ist in zweifacher Hinsicht allgemeiner, als die oben erwähnte Regel „Gleiches durch Gleiches zu ersetzen“:

1. Paramodulation ist nicht nur mit unbedingten, sondern auch mit bedingten Gleichungen anwendbar, das heißt, die Gleichungs-Klausel kann noch weitere Literale enthalten.

2. Die beiden Terme, welche eine Ersetzung ermöglichen sollen, müssen nicht gleich, sondern lediglich unifizierbar sein, was bedeutet, daß es Instanzen der beteiligten Klauseln geben muß, so daß die entsprechenden Terme gleich sind.

Die Paramodulationsregel ist korrekt: Falls S eine E-erfüllbare (bzw. E-unerfüllbare) Menge von Klauseln und C Paramodulant von zwei Klauseln aus S ist, dann ist auch $S \cup \{C\}$ E-erfüllbar (bzw. E-unerfüllbar).

Der Resolutionskalkül, erweitert um die Paramodulationsregel und das Axiom $\{x = x\}$, bilden einen widerlegungsvollständigen Kalkül (RP-Kalkül genannt) für die Prädikationlogik erster Stufe mit Gleichheit: Für jede E-unerfüllbare Menge von Klauseln existiert eine Ableitung der leeren Klausel unter Anwendung der Regeln und Axiome des Kalküls. Das Reflexivitätsaxiom ist nötig, da andernfalls die leere Klausel nicht aus der E-unerfüllbaren, einelementigen Klauselmenge $\{\{\neg a = a\}\}$ ableitbar wäre.

Da das Reflexivitätsaxiom $\{x = x\}$ E-allgemeingültig ist, ist der RP-Kalkül kein reiner negativer Testkalkül (siehe Kap. II). Im Sinne eines negativen Testkalküls müßte statt des Reflexivitätsaxioms das E-unerfüllbare Axiom $\{\neg x = x\}$ verwendet werden, welches genau wie die leere Klausel als Ziel (elementarer Widerspruch) angestrebt werden könnte. Es müßte dann aber auch einen Mechanismus geben, der erkennt, daß eine Klausel nur aus Instanzen von $\{\neg x = x\}$ besteht, die Klausel $\{\neg a = a, \neg b = b, \neg c = c\}$ wäre ja auch ein elementarer Widerspruch.

Aus praktischen Gründen wird stattdessen meist das Reflexivitätsaxiom verwendet, um E-unerfüllbare Literale (Instanzen von $\{\neg x = x\}$) durch einen Resolutionsschritt zu eliminieren.

Verglichen mit der expliziten Verwendung der Gleichheitsaxiome, ist der Suchraum für die Ableitung der leeren Klausel durch Einführung der Paramodulation deutlich kleiner. Viele der sinnlosen Resolutionen mit und zwischen den Gleichheitsaxiomen sind nicht mehr möglich. Trotzdem sind die entstehenden Suchräume ohne geschickte Steuerung der Paramodulation noch viel zu groß, da auch diese Regel fast überall in der Klauselmenge angewendet werden kann. Für das erste Beispiel in Abschnitt III.1 sind nach A. Bundy ([Bun83] S. 86) bei reiner Breitensuche etwa 10^{11} Paramodulationschritte nötig, um den Beweis des Satzes zu finden.

Eine beliebige willkürliche Auswahl von Paramodulationsschritten führt also zu riesigen Suchräumen. Daher stellt sich die Frage: Welche Rolle sollen die Gleichungen bei der Beweissuche übernehmen? Wann und wie sollen also die Gleichheitsliterale bei der Auswahl eines Schrittes berücksichtigt werden? Mit Hilfe des folgenden Beispiels werden zwei unterschiedliche Ansätze für globale Kontrollstrategien zur Steuerung von Gleichheitsoperationen vorgestellt.

Beispiel: Gegeben sei die E-unerfüllbare Klauselmenge

C1 := {P(f(a)), Q(y), R(b)} C4 := {¬R(a)}

C2 := {¬P(f(b))} C5 := {f(x) = x}

C3 := {¬Q(a)} C6 := {a = b}

Lösung a:

1. Paramodulation C1,1 mit C5 ⊢ P1 := {P(a), Q(y), R(b)}
2. Paramodulation P1,1 mit C6 ⊢ P2 := {P(b), Q(y), R(b)}
3. Paramodulation C2 mit C5 ⊢ P3 := {¬P(b)}
4. Paramodulation C3 mit C6 ⊢ P4 := {¬Q(b)}
5. Paramodulation C4 mit C6 ⊢ P5 := {¬R(b)}
6. Resolution P2,1 und P3 ⊢ R1 := {Q(y), R(b)}
7. Resolution R1,1 und P4 ⊢ R2 := {R(b)}
8. Resolution R2 und P5 ⊢ R3 := □

Bei dieser Lösung werden die in der Formelmenge vorkommenden Gleichungen f(x) = x und a = b verwendet, um das gegebene Problem zu vereinfachen. Die Gleichungen werden mit fest vorgegebener Richtung auf alle übrigen Literale der Formelmenge angewendet. Durch Anwendung von f(x) = x werden die Terme f(b) und f(a) vereinfacht zu b bzw. a und das Funktionssymbol f wird aus der Formelmenge eliminiert. Durch Anwendung von a = b wird das Symbol a aus der Formelmenge eliminiert, die so mit fünf Paramodulationsschritten reduziert wird zu: {P(b), Q(y), R(b)}, {¬P(b)}, {¬Q(b)}, {¬R(b)}. Diese Art von Steuerung wurde für die Demodulation [WRCS67] zuerst verwendet. Die Demodulation arbeitet ähnlich wie die Paramodulation und wird speziell verwendet, um Terme zu vereinfachen. Anders als bei der Paramodulation werden Gleichungen nur in einer Richtung angewendet, Unifikationen einseitig durchgeführt (matching) und Formeln destruktiv verändert. Aus der Demodulation hat sich das Gebiet der Termersetzungssysteme (Abschnitt III.4) entwickelt. Die Termersetzung zur Problemreduzierung läßt sich relativ einfach mit anderen Kontrollstrategien kombinieren (z.B. [LH85], [Blä87]).

Lösung b:

1. Auswahl der Literale C1,1 und C2, also P(f(a)) und ¬P(f(b)) mit dem Ziel einer Resolution
2. Unterschiede zwischen den Termen f(a) und f(b) mit Hilfe der Gleichung C6 beseitigen:

   Paramodulation C1,1 mit C6 ⊢ P1 := {P(f(b)), Q(y), R(b)}
3. Resolution P1,1 und C2 ⊢ R1 := {Q(y), R(b)}
4. Resolution R1,1 und C3 ⊢ R2 := {R(b)}
5. Auswahl der Literale R2 und C4 mit dem Ziel einer Resolution
6. Unterschiede zwischen den Termen a und b mit Hilfe der Gleichung C6 beseitigen:

   Paramodulation R2 mit C6 ⊢ P2 := {R(a)}
7. Resolution C4 und P2 ⊢ R3 := □

Die Gleichung a = b wird gezielt angewendet, um die Unterschiede zwischen den Literalen

P(f(a)) und ¬P(f(b)), sowie zwischen R(b) und ¬R(a) zu reduzieren und so entsprechende Resolutionsschritte zu ermöglichen. Diese Art von Kontrollstrategie beruht auf einer Idee, die man als „paramodulation if needed“ bezeichnen kann: Paramodulationen sollen nur dann ausgeführt werden, wenn sie im Hinblick auf ein übergeordnetes Ziel nötig sind, d.h. insbesondere den Unterschied zwischen zwei potentiell resolvierbaren Literalen (Literale mit gleichem Prädikat und entgegengesetzten Vorzeichen) so reduzieren, daß schließlich ein Ableitungsschritt durch Resolution möglich wird. Eine solche Zielvorgabe und -verfolgung würde die Kontrolle zwischen Resolutions- und Paramodulationsfolgen regeln.

Der RP-Kalkül unterstützt aber weder die Vorgabe von Zielen für Gleichheitsoperationen (Auswahl potentiell resolvierbarer Literale) noch die Steuerung von Gleichheitsoperationen im Hinblick auf ein Ziel und ist daher denkbar ungeeignet zur Realisierung der angestrebten zielgerichteten Kontrollstrategie. Die Idee des „paramodulation if needed“ führte daher zur Entwicklung neuer Methoden, wobei die Behandlung der Gleichheit meist in eine verallgemeinerte Resolutionsregel eingefügt wurde (z.B. [Mor69], [HR78], [Dig79]). Einige dieser Verfahren werden im folgenden vorgestellt. Im Abschnitt über Termersetzungssysteme wird dann ein Verfahren vorgestellt, das die Demodulationsidee in einen vollständigen, aber bezüglich des Suchraumes drastisch eingeschränkten RP-Kalkül integriert.

2.2 Kontrolle Resolution - Gleichheit: E-Resolution

Explizit realisiert wurde die „if needed“-Idee mit dem Vorschlag von J.B. Morris, die Gleichheit in eine verallgemeinerte Resolutions-Regel, die E-Resolution (E = Equality), einzubeziehen [Mor69]. Durch diese Regel wird die Rolle der Gleichheit festgelegt, als Mittel zum Zweck, Resolutionsmöglichkeiten zu erzeugen. Zwei Literale werden ausgewählt, zwischen denen eine Resolution wünschenswert ist, und mit Hilfe von Gleichungen sollen die Voraussetzungen zur Anwendbarkeit der Resolution geschaffen werden.

Die E-Resolution ist ein Spezialfall der von M. Stickel eingeführten Theorieresolution [Sti85]. Während bei der Theorieresolution beliebige Theorien zugelassen sind, ist die E-Resolution auf Gleichheitstheorien beschränkt.

Bei der formalen Definition der E-Resolutions-Regel werden zwei Formen unterschieden:

1. Form: Seien $C_1 := \{P(s_1, ..., s_m), A_1\}$

$C_2 := \{\neg P(t_1, ..., t_m), A_2\}$ und

$C_i := \{l_i = r_i, A_i\}$ für $i := 3 ... k$ Klauseln,

wobei die Symbole A_i $(i = 1 ... k)$ jeweils für mehrere Literale stehen sollen,

weiter sei $E := \{l_3 = r_3, ..., l_k = r_k\}$.

Falls eine Substitution σ existiert, so daß $\sigma s_i = \sigma t_i$ aus E folgt, d.h.

$E \models \sigma s_i = \sigma t_i$ für $i := 1 ... m$,

dann ist $\sigma(A_1 \cup ... \cup A_k)$ E-Resolvente der Klauseln $C_1, ..., C_k$.

Die erste Form der E-Resolutions-Regel sollte nicht zwischen Gleichungen und negierten Gleichungen angewendet werden, denn hierfür gibt es eine spezielle Form der E-Resolution:

2. Form: Seien $C_1 := \{\neg\, s = t\,,\ A_1\}$ und

$C_i := \{l_i = r_i\,,\ A_i\}$ für $i := 2 ... k$ Klauseln,

weiter sei $E := \{l_2 = r_2, ..., l_k = r_k\}$.

Falls eine Substitution σ existiert, so daß $\sigma s = \sigma t$ aus E folgt, d.h.

$E \models \sigma s = \sigma t$

dann ist $\sigma(A_1 \cup ... \cup A_k)$ E-Resolvente der Klauseln $C_1, ..., C_k$.

Mit dieser zweiten Form der E-Resolutions-Regel wird sehr viel deutlicher als bei der Paramodulation ausgedrückt, daß es bei negierten Gleichungen darauf ankommt, beide Seiten gleich zu machen, um so den gewünschten Widerspruch herbeizuführen. Damit wird durch die E-Resolution auch im Hinblick auf Ungleichungen ein Ziel explizit vorgegeben.

Beispiel: Seien $\{P(c, h(f(a, y), b)) , Q(g(y))\}$,

$\{\neg P(a, h(g(c), d))\}$,

$\{Q(x) , f(x, e) = g(x)\}$,

$\{a = c , R(a)\}$ und

$\{b = d\}$ Klauseln,

dann ist $\{Q(g(e)), Q(a), R(a)\}$ eine E-Resolvente dieser Klauseln.

Der E-Resolutions-Kalkül ist ein negativer Testkalkül bestehend aus einer Regel (in zwei Varianten) und der leeren Klausel als unerfüllbares Axiom. Der Kalkül erfüllt die gewünschten Eigenschaften der Korrektheit und der Vollständigkeit [And70].

Ein E-Resolutionsschritt kann als eine Folge von Paramodulationsschritten aufgefaßt werden, welche zwei potentiell resolvierbare Literale resolvierbar macht, gefolgt von dem entsprechenden Resolutionsschritt. Die hierbei entstehenden Zwischenklauseln werden aber bei der E-Resolution nicht erzeugt, wodurch der Suchraum deutlich kleiner wird als im RP-Kalkül.

Die Anwendung der E-Resolution könnte somit eine optimale Realisierung der „if needed“-Idee und potentiell eine der besten Möglichkeiten zur Behandlung der Gleichheit sein, denn Gleichungen werden nur benutzt, um die Unterschiede zwischen Termen zu beseitigen, die eine gewünschte Resolution verhindern. Darüberhinaus werden die Gleichungen nur dann angewendet, wenn es möglich ist, diese Unterschiede vollständig zu beseitigen.

Allerdings tritt gerade bei der Gleichheit das Problem auf, daß häufig ähnliche oder gleiche Unterprobleme auftreten, daß also eine Art „bottom up“-Strategie wesentlich erfolgreicher sein könnte. Aus einfachen Fakten bauen sich unter Umständen leichter die komplexen Sachverhalte

auf, als daß es möglich wäre sie durch Zerlegung herzuleiten. Der Effekt verstärkt sich noch, wenn nicht nur ein Ziel, sondern viele zu erreichen sind. Dieser Fall macht eine Art Lemmagenerierung fast zwingend.

Die E-Resolution regelt zwar das Zusammenwirken der Resolution und der Behandlung der Gleichheit und liefert Ziele für Gleichheitsoperationen, jedoch werden keine Hinweise gegeben, wie solche Ziele erreicht werden könnten. Es bleibt also das Problem: Wie können geeignete Instanzen und ein Beweis für die Gleichheit zweier Terme gefunden werden? Im allgemeinen (d.h. für beliebige Gleichungsmengen E) ist die Gleichheit zweier Terme nicht entscheidbar, also ist auch nicht entscheidbar, ob ein E-Resolutionsschritt überhaupt anwendbar ist.

Werden also zwei potentiell resolvierbare Literale für eine E-Resolution ausgewählt, dann muß der Versuch, die entsprechenden Terme unter der gegebenen Gleichheitstheorie zu unifizieren, nach einer bestimmten endlichen Zeit gestoppt werden, denn es könnte ja der falsche Weg sein, um die leere Klausel herzuleiten. Andererseits ist bei einem erfolglosen Abbruch der Suche nicht sicher, ob die Terme nicht doch noch unifizierbar sind. In diesem Fall könnte es zwingend nötig sein, gerade für diese beiden Terme die Suche nach einer Lösung irgendwann wieder fortzusetzen.

Ein automatisches Deduktionssystem basierend auf dem E-Resolutions-Kalkül kann also nur dann erfolgreich sein, wenn es möglich ist, den Versuch, zwei Literale resolvierbar zu machen, nach einer gewissen Zeit zu unterbrechen und irgendwann später wieder fortzusetzen. Praktisch bedeutet dies, daß gleichzeitig immer mehrere Ziele (Kandidaten für eine E-Resolution) gebildet sein und verfolgt werden müssen. Leider sind praktisch alle Literale mit gleichem Prädikat und verschiedenem Vorzeichen Kandidaten für E-Resolution. Die meisten Gleichheitstheorien lassen kaum eine Möglichkeit hier merklich zu filtern.

Vor Anwendung der E-Resolution müssen zwei wesentliche Probleme gelöst werden:

1. Für verschiedene potentielle E-Resolutionen muß die Erzeugung und Lösungssuche für die entsprechenden Gleichheitsprobleme sinnvoll organisiert werden (z.B. heuristische Vergabe von Resourcen).
2. Für eine ausgewählte potentielle E-Resolution muß eine Folge von Gleichheitsoperationen gefunden werden, die die Unterschiede in den Termen beseitigt, d.h. ein Unifikator und ein Beweis für die Gleichheit der entsprechenden Instanzen sind zu finden.

2.3 Abstandsverringerung: RUE-Resolution

Für ein praktisches Deduktionsystem ist ein E-Resolutionsschritt ein zu großer Schritt bei der Suche nach einem Beweis. Die Idee einer schrittweisen E-Resolution verfolgt V.J. Digricoli, der als Weiterführung der Methode von Morris einen neuen Kalkül für die Prädikatenlogik erster Stufe mit Gleichheit entwickelte [Dig79]. Dieser Kalkül bietet Lösungsansätze für die beiden wesentlichen Probleme der E-Resolution.

Ebenso wie bei der E-Resolution liefern die Regeln des Kalküls die Ziele für nötige

Gleichheitsoperationen in Form von potentiell resolvierbaren Literalen. Während aber eine E-Resolution nur dann ausgeführt werden kann, wenn die Unterschiede zwischen den korrespondierenden Termen der entsprechenden Literale vollständig beseitigt werden können, kann die RUE-Resolution (Resolution by Unification and Equality) jederzeit durchgeführt werden, wobei eine neue Klausel gebildet wird, welche die noch zu lösenden Teilprobleme repräsentiert. Zwei Literale mit dem gleichen Prädikatssymbol und unterschiedlichem Vorzeichen können auch dann wegresolviert werden, wenn nicht alle korrespondierenden Untertermpaare unifizierbar sind. Die nicht unifizierbaren Untertermpaare („disagreement pairs“) werden als negierte Gleichungen zur neuen Klausel hinzugefügt und bilden somit Unterprobleme, die irgendwann noch gelöst werden müssen. Diese Vorgehensweise hat zwar den Nachteil, daß wieder Zwischenklauseln entstehen (wie auch bei der Paramodulation), aber der Vorteil, daß eine Kontrolle für die Erzeugung und Lösungssuche für Gleichheitsprobleme vom Kalkül unterstützt wird, dürfte überwiegen.

Zur Lösung des zweiten Problems – Unifikation zweier Terme unter einer Gleichheitstheorie – stützt sich Digricoli nicht mehr auf das bis dahin dominierende Prinzip der Untertermersetzung (z.B. Paramodulation), sondern er verwendet das Prinzip der *Abstandsverringerung*, wobei der Unterschied zwischen zwei Termen reduziert und falls möglich beseitigt wird. Der Typ des Unterschieds zwischen zwei Termen bestimmt den nächsten Schritt, ähnlich wie dies beim General Problem Solver [NSS59] der Fall war. Verfahren, die auf diesem Prinzip beruhen, vergleichen meist die Symbole auf der obersten Termebene (das Symbol auf oberster Termebene eines Terms t ist t selbst, falls t eine Variable oder Konstante ist, und es ist f falls $t = f(t_1,..., t_n)$). Falls diese verschieden sind, wird versucht, eine Gleichungskette zu finden, die genau diesen Unterschied beseitigt, anderenfalls bilden die korrespondierenden Unterterme neue Unterprobleme, die nun zu lösen sind.

Genau wie Morris vermeidet auch Digricoli das Reflexivitätsaxiom durch die Verwendung von zwei Regeln. Diese Regeln basieren auf der Definition von „disagreement sets“. Ein disagreement set für s und t ist eine Menge von Termpaaren, deren Gleichheit die Gleichheit von s und t impliziert. Im allgemeinen gibt es mehrere disagreement sets für s und t.

Disagreement sets für zwei Terme s und t:

a) Falls die beiden Terme s und t identisch sind, dann ist die leere Menge das einzige disagreement set.

b) Falls für s und t die Symbole auf oberster Termebene verschieden sind (z.B. falls s := a und t := b, oder s := f(a, b, c) und t := g(x)), dann gibt es auch nur ein disagreement set, nämlich {[s, t]}.

c) Falls s und t die Form $f(s_1 ,..., s_n)$ bzw. $f(t_1 ,..., t_n)$ haben und verschieden sind, dann gibt es mehrere disagreement sets für s und t, nämlich

$$\{[s, t]\} \quad \text{(origin disagreement set)},$$

außerdem die Menge der nicht identischen korrespondierenden Argumentpaare der Terme s und t : $\{[s_i, t_i]: 1 \le i \le n\}$ (topmost disagreement set)

und, falls D ein disagreement set für die Terme s und t ist, dann ist auch D' gebildet aus D durch Ersetzen eines Elementes durch die Elemente eines seiner disagreement sets ein disagreement set für s und t.

Disagreement sets für zwei Literale $P(s_1, ..., s_n)$ und $\neg P(t_1, ..., t_n)$:

falls für i := 1 ... n D_i ein disagreement set für s_i und t_i ist, dann ist

$$D := \bigcup_{i:=1..n} D_i$$

ein disagreement set für die beiden Literale.

Beispiel: Die beiden Terme f(a, h(b, g(c))) und f(b, h(c, g(d))) haben die disagreement sets:

{ [f(a, h(b, g(c))), f(b, h(c, g(d)))] } (origin disagreement set)

{ [a, b], [h(b, g(c)), h(c, g(d))] } (topmost disagreement set)

{ [a, b], [b, c], [g(c), g(d)] } (intermediate disagreement set)

{ [a, b], [b, c], [c, d] } (bottommost disagreement set)

Während Morris bei der E-Resolution von einer Regel (in zwei Varianten) spricht, definiert Digricoli zwei Regeln, die „Resolution by Unification and Equality“ (RUE) Regel, welche der ersten Form der E-Resolutions Regel entspricht, und die „Negative Reflexive Function“ (NRF) Regel, die der zweiten Form der E-Resolution entspricht.

RUE Regel: Seien $C_1 := \{P(s_1, ..., s_m), A\}$

$C_2 := \{\neg P(t_1, ..., t_m), B\}$ zwei Klauseln,

wobei A und B wieder jeweils für mehrere Literale stehen sollen.
Weiter sei σ eine Substitution und D ein (beliebiges) disagreement set der beiden Literale $\sigma(P(s_1, ..., s_m))$ und $\sigma(\neg P(t_1, ..., t_m))$, sei $D' := \{\neg\, l = r \mid [l, r] \in D\}$ die durch D bestimmte Menge von negierten Gleichungen, dann ist

$\sigma A \cup \sigma B \cup D'$

RUE-Resolvente der Klauseln C_1 und C_2.

Anders als bei der E-Resolution muß diese Regel auch zwischen Gleichungen und negierten Gleichungen angewendet werden können. Eine zweite Regel ist auf negierte Gleichungen anwendbar:

NRF Regel: Sei $C := \{\neg\, s = t, A\}$ eine Klausel,
σ eine Substitution und D ein (beliebiges) disagreement set der Terme σs und σt, sei $D' := \{\neg\, l = r \mid [l, r] \in D\}$ die durch D bestimmte Menge von negierten Gleichungen, dann ist

$\sigma A \cup D'$

NRF-Resolvente der Klausel C.

Der RUE-Resolutionskalkül besteht aus den beiden Regeln RUE und NRF und ist korrekt und vollständig. Wenn allgemein von RUE-Resolution gesprochen wird, dann ist der Kalkül mit

beiden Regeln (RUE und NRF) gemeint.

Bei der Methode von Digricoli müssen in jedem Schritt neben den Literalen, die an einer Operation beteiligt sein sollen, auch eine Substitution und ein disagreement set ausgewählt werden. Besonders problematisch ist hierbei die Auswahl der Substitution. Eine freie Wahl der Substitution käme einer Instantiierungsregel gleich mit allen Nachteilen für den Suchraum. Um dies zu vermeiden definiert Digricoli einen „most general partial unifier“ (mgpu). Der mgpu berechnet sich ähnlich wie der mgu bei Robinson, indem von links nach rechts Unterterme unifiziert und die jeweiligen partiellen Unifikatoren auf die restlichen Terme angewendet werden. Anstatt mit Mißerfolg zu terminieren, wird die Unifikation jedoch fortgesetzt, wenn zwei korrespondierende Unterterme bzgl. der Symbole auf der obersten Termebene nicht unifizierbar sind.

Beispiele:

Der mgpu von	f(a, x, h(x), g(y, b))	und
	f(h(b), b, z, g(a, z))	ist
	$\{x \leftarrow b,\ z \leftarrow h(b),\ y \leftarrow a\}$	
RUE-Resolvente von	{ P(f(a, x, h(x), g(y, b))), Q(x) }	und
	{ ¬ P(f(h(b), b, z, g(a, z))) }	ist
	{ Q(b) , ¬ a = h(b) , ¬ b = h(b) }	
Der mgpu von	h(f(x, y), g(h(x), x))	und
	h(f(a, z), g(h(z), f(a, a)))	ist
	$\{x \leftarrow a,\ z \leftarrow a,\ y \leftarrow a\}$	
NRF-Resolvente von	{ ¬ h(f(x, a), g(h(x), x)) = h(f(a, z), g(h(y), f(a, a))) }	
ist	{ ¬ a = f(a, a) }	

In beiden Beispielen sind auch andere RUE- (bzw. NRF-) Resolventen möglich, denn es gibt noch andere disagreement sets als die jeweils verwendeten, und es muß nicht der mgpu, sondern es kann eine beliebige Substitution gewählt werden. Um die Vollständigkeit zu zeigen, läßt Digricoli die Auswahl beliebiger Substitutionen zu. Für die Praxis schlägt er allerdings vor, die leere Substitution, den mgpu oder eine „dazwischen“ liegende Substitution zu wählen.

Die Wahl einer Substitution hat praktische Konsequenzen für die Beweissuche. Bei Verwendung der leeren Substitution werden viele Unterprobleme erzeugt und der Suchraum ist relativ groß. Die Wahl des mgpu kann das Finden eines Beweises verhindern, da die Instantiierung möglicherweise zu stark oder falsch ist, und so Unterprobleme erzeugt werden, die nicht lösbar sind. Angenommen im letzten Beispiel (s.o.) bestünde die Gleichheitstheorie lediglich aus der Kommutativität von f, dann könnte a nicht gleich zu f(a, a) gemacht, und somit das Literal ¬ a = f(a, a) niemals wegresolviert werden. Würde der mgpu von rechts nach links berechnet, dann wäre ein anderer (in diesem Falle günstigerer) mgpu herausgekommen.

Das Problem, zwei Terme unter einer Gleichheitstheorie zu unifizieren (bzw. die Lösung eines

solchen Problems), ist unabhängig von der Reihenfolge der Argumente, d.h. gleichmäßige Permutationen für beide Terme ändern nicht die Lösbarkeit oder die Lösung des Problems. Durch die Anwendung von partiellen Unifikatoren auf andere Probleme ist der von Digricoli definierte mgpu jedoch von der Reihenfolge der Argumente abhängig. Die Reihenfolge, durch die der mgpu bestimmt wird, ist willkürlich und verursacht die Probleme bei der Auswahl einer Substitution.

2.4 Planen: Equality Graphs

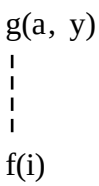
Das Klauselgraphverfahren (Kap. II) wurde von J. Siekmann und G. Wrightson durch Einbeziehung der Paramodulation erweitert [SW80]. Das in diesem Abschnitt vorgestellte Verfahren ist als Weiterentwicklung dieses „paramodulierten“ Klauselgraphverfahrens einerseits und von Digricoli's Kalkül andererseits entstanden. Wie Digricoli's Verfahren basiert auch dies auf dem Prinzip der Abstandsverminderung, wobei jedoch das Problem der Auswahl einer Substitution durch einen anderen Ansatz zur partiellen Unifikation gelöst wird.

Das im folgenden vorgestellte Verfahren, Gleichheitsgraphen als Beweis für die Unifizierbarkeit zweier Terme (bzgl. einer Gleichheitstheorie) zu konstruieren (ECOP: Equality Graph Construction Procedure), unterscheidet sich grundsätzlich von den oben vorgestellten Methoden. In den Abschnitten 2.1 bis 2.3 wurden spezielle prädikatenlogische Kalküle zur Behandlung der Gleichheit vorgestellt, welche aus wenigen, relativ einfachen Regeln bestehen. Diese Regeln manipulieren eine logische Formel, so daß der Wahrheitswert dieser Formel erhalten bleibt.

Bei ECOP werden keine Formeln manipuliert, sondern mit Hilfe von Regeln eines Produktionssystems wird ein Graph für zwei Terme konstruiert, der einen Unifikator dieser Terme und einen Beweis für die Gleichheit der entsprechenden Instanzen (also die Lösung von Gleichheitsproblemen) darstellt. Ein solcher Graph hat keinen Wahrheitswert, erfüllt aber bestimmte Eigenschaften, so daß ein Lösungsgraph für zwei Terme s und t genau dann konstruierbar ist, wenn eine Substitution σ existiert, so daß die Gleichheit von σs und σt semantisch aus der gegebenen Gleichungsmenge E folgt (Korrektheit und Vollständigkeit).

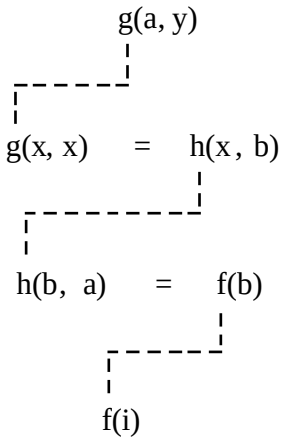
Wegen der Komplexität des Regelsystems wird das zugrundeliegende Prinzip lediglich anhand von Beispielen erläutert. Dabei wird das Problem, zwei Terme s und t unter der Gleichheitstheorie E zu unifizieren, im folgenden durch $< s =_E t >$ notiert.

Sei E := {g(x, x) = h(x, b), h(u, v) = h(v, u), h(b, a) = f(b), b = c, c = i} und sei $<$ g(a, y) $=_E$ f(i) $>$ das gegebene Gleichheitsproblem, dann ist

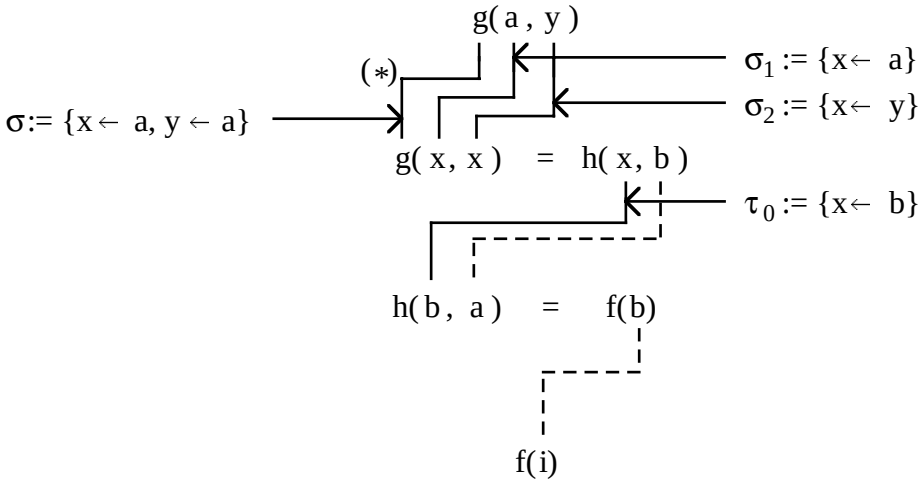


der initiale Gleichheitsgraph. Die einzige Information, die dieser Graph enthält, ist, daß das

Problem, die Terme g(a, y) und f(i) zu unifizieren (unter der Theorie E), noch zu lösen ist. Die wesentliche Differenz zwischen den beiden Termen liegt in den unterschiedlichen Funktionssymbolen g und f. Dieser Unterschied muß mit Hilfe von Gleichungen aus der gegebenen Menge E behoben werden. Zwei dieser Gleichungen können zu einer Kette g(x, x) = h(x, b) - - - - h(b, a) = f(b) verbunden und in den Graphen eingefügt werden, wodurch der Unterschied in den Funktionssymbolen auf oberster Termebene behoben wird:



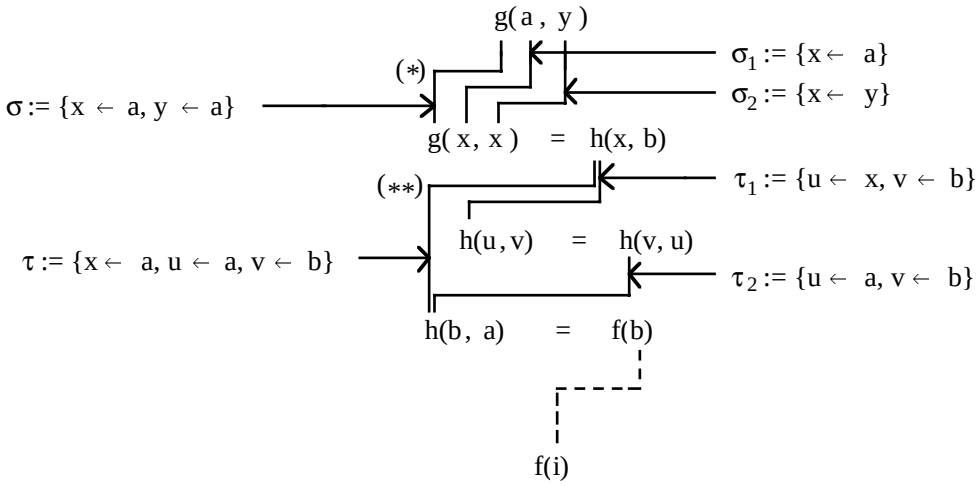
Durch das Einfügen der Gleichungskette entstehen drei Unterprobleme, die noch zu lösen sind: $< g(a, y) =_E g(x, x) >$, $< h(x, b) =_E h(b, a) >$ und $< f(b) =_E f(i) >$. In allen drei Fällen sind jeweils für die beiden Terme die Symbole auf oberster Termebene gleich. Für jedes dieser Unterprobleme werden durch die korrespondierenden Unterterme weitere Unterprobleme definiert, von denen einige triviale Lösungen haben. Diese Situation wird in dem Graphen



dargestellt. Die durchgezogenen Linien stellen gelöste Unterprobleme dar, die jeweils mit einer Substitution markiert sind (leere Substitutionen werden weggelassen). Diese Substitutionen zusammen mit eventuell verwendeten Gleichungen stellen die Lösungen der betreffenden (Unter-) Probleme dar. Die Substitutionen σ_1 und σ_2 , welche hier Lösungen für Teilprobleme sind, müssen kompatibel, d.h. selbst unifizierbar sein. Das Ergebnis einer entsprechenden Unifikation wird durch eine eigene mit (∗) gekennzeichnete Linie dargestellt und mit dem Unifikator (hier σ) markiert.

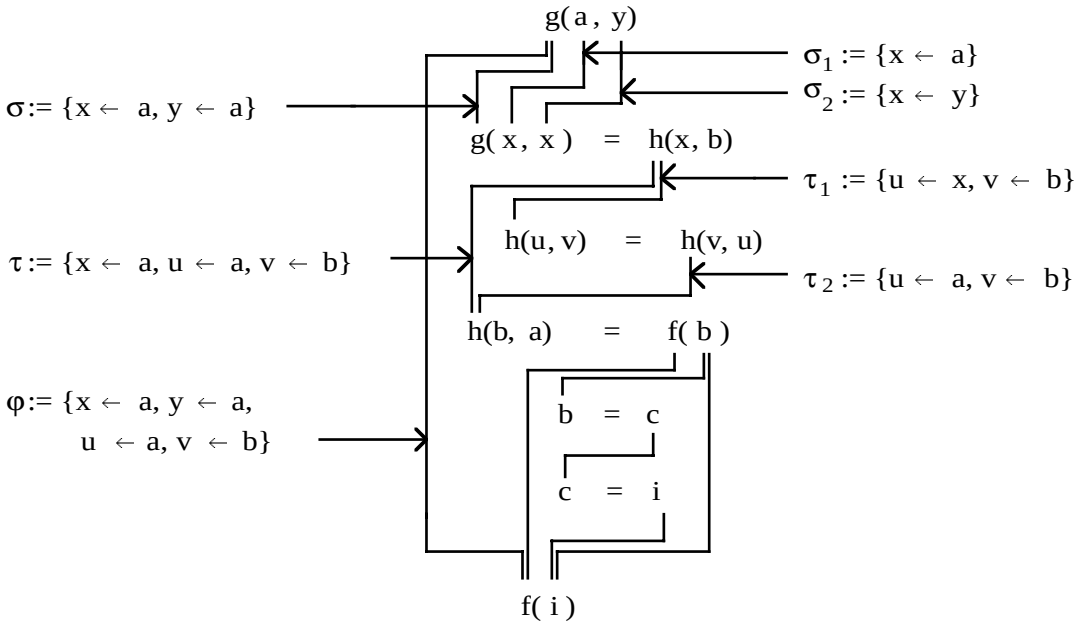
Gestrichelte Linien (- - - -) kennzeichnen ungelöste Unterprobleme, die in einem weiteren

Schritt ausgewählt werden können. Angenommen das Unterproblem b - - - a sei ausgewählt: Aus der gegebenen Menge von Gleichungen E gibt es keine Teilmenge, die den Unterschied zwischen a und b beheben und in den Graphen eingesetzt werden könnte; dieses Unterproblem ist nicht lösbar. Anstatt das Problem b - - - a zu lösen, kann aber ein neues Unterproblem auf höherer Termstufe erzeugt werden, hier h(x, b) - - - h(b, a). Zwischen diesen beiden Termen kann nun die Gleichung h(u, v) = h(v, u) eingefügt werden mit dem Resultat:



Die Substitutionen entlang einer Kette zwischen zwei Termen müssen auf Kompatibilität geprüft werden. Das Ergebnis der Unifikation dieser Substitutionen wird wieder durch eine eigene durchgezogene Linie dargestellt. In unserem Beispiel verbindet die einelementige Kette h(u, v) = h(v, u) die Terme h(x, b) und h(b, a), die Substitutionen τ_1 und τ_2 müssen unifiziert werden, und das Ergebnis wird mit der durch (**) gekennzeichneten Linie dargestellt, die mit dem Unifikator τ markiert ist.

Wird schließlich das Unterproblem b - - - i ausgewählt, kann die Kette b = c - - - c = i in den Graphen eingefügt werden:



Da σ und τ ihrerseits (zum Unifikator φ) unifizierbar sind, ist das gegebene Gleichheitsproblem

vollständig gelöst.

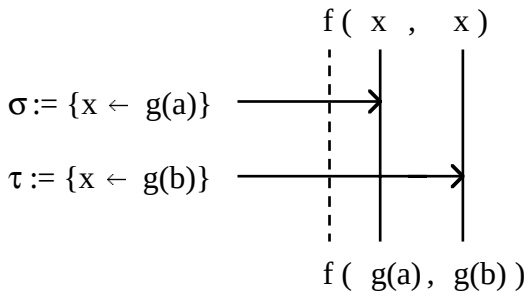
Beginnend mit einem leeren Graphen wird also ein Lösungsgraph mit einer Folge von Transformationen konstruiert, wobei jeder mögliche Zwischengraph eine Lösung für einen gewissen Abstraktionsgrad des Problems darstellt ([Pla81]) und als globaler Plan für die Suche nach einer Lösung für das Originalproblem verwendet wird. Die Abstraktion wird mit jedem Schritt abgeschwächt, wobei der Graph – und damit der Plan – verfeinert wird. Die gestrichelten Linien (ungelöste Unterprobleme) kennzeichen die Positionen, an denen eine Abstraktion benutzt wird. In unserem Beispiel kann eine Abstraktion wie folgt formuliert werden: Berücksichtige nicht das zweite Argument der Funktion h und auch nicht das erste Argument der Funktion f. Normalerweise kann eine solche Abstraktion nicht einheitlich für alle Stellen, an denen eine Funktion vorkommt, ausgedrückt werden, sondern lediglich in Abhängigkeit von speziellen Positionen. In jedem Schritt enthält der Graph die Information, ob und wie die bereits gefundenen Teillösungen zusammenpassen sowie die Information über die Unterprobleme, die noch gelöst werden müssen.

In dem obigen Beispiel konnten die gebildeten Teillösungen immer zu einer Lösung auf höherer Ebene verknüpft werden. Häufig ist jedoch die Verknüpfung von Teillösungen nicht unmittelbar möglich. Um einen Konflikt mit nicht kompatiblen Teillösungen aufzulösen, gibt es in ECOP zwei Möglichkeiten:

1. Die zu unifizierenden Substitutionen werden unter der gegebenen Gleichheitstheorie unifiziert.

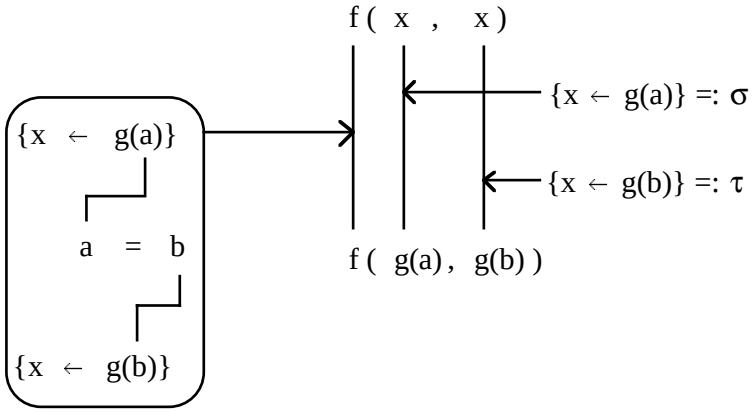
2. Teilgraphen werden durch neue Teillösungen für dasselbe Unterproblem ersetzt. Aus der durch den Konfikt gegebenen Information können Positionen im Graph gefunden werden, die den Konflikt verursacht haben. Im Sinne des „dependency directed backtracking“ [SS77] können nun lokal Teilgraphen durch andere Graphen ersetzt werden, ohne daß andere Teillösungen und der globale Plan zerstört werden. Das Prinzip der Untergraphersetzung ist in [Blä87] ausführlicher beschrieben.

Die erste Möglichkeit Unifikationskonflikte zu lösen wird nun noch an einem Beispiel erläutert. Sei $< f(x, x) =_E f(g(a), g(b)) >$ ein Gleichheitsproblem mit $E := \{a = b\}$. Durch die korrespondierenden Unterterme von f(x, x) und f(g(a), g(b)) werden trivial lösbare Unterprobleme gebildet:



Die Lösungen dieser Unterprobleme sind nun aber nicht unmittelbar kompatibel. Anstatt irgendwelche Gleichungen für trival lösbare Probleme (z.B. zwischen x und g(a)) einzusetzen, um so andere Teillösungen zu erhalten, werden die Substitutionen σ und τ selbst unter der

Gleichheitstheorie unifiziert, mit dem Resultat:



Korrespondierend zu verschiedenen Arten von Unterproblemen gibt es also auch verschiedene Arten von Gleichheitsgraphen:

1. Gleichheitsgraphen zur Darstellung einer lokalen Unifikation von zwei Termen unter einer Theorie E, wobei die unifizierende Substitution nie auf andere Teilprobleme angewendet wird.
2. Gleichheitsgraphen zur Darstellung der Unifikation einer Menge von Substitutionen unter der Theorie.

Anders als bei der RUE-Resolution ist hier die Lösung von Gleichheitsproblemen unabhängig von der Reihenfolge der Argumente in den zu unifizierenden Termen. Dies wird durch ein andere Art von partieller Unifikation erreicht, wobei Unifikatoren von Teilproblemen nicht auf andere Unterprobleme angewendet werden. Dadurch sind die einzelnen Teillösungen voneinander unabhängig, und bei der Wahl einer alternativen Lösung für ein Unterproblem behalten die Lösungen anderer Unterprobleme ihre Gültigkeit. Durch die Unabhängigkeit von Teillösungen werden Untergraphersetzungen (s.o.) im Sinne eines „dependency directed backtracking“ erst möglich, und das Problem der Auswahl einer Substitution entfällt.

ECOP kann als universeller Unifikationsalgorithmus (siehe [Sie86]) betrachtet werden, und könnte in einem E-Resolutionskalkül dazu dienen, die Voraussetzungen für das Ausführen von E-Resolutionsschritten zu schaffen.

2.5 Schlußbemerkungen

Unter den Aspekten der Einschränkung von Suchräumen einerseits und gleichzeitiger Erhaltung einer breiten Anwendbarkeit andererseits wurden viele spezielle Methoden zur Behandlung der Gleichheit entwickelt, wobei die Tendenz zu mehr zielorientierten Verfahren besteht. Am Anfang dieser Entwicklung stand die Berücksichtigung lokaler Ziele auf der Basis der Unifikation, um willkürliche Instantiierungen überflüssig zu machen. Danach wurden immer mehr Verfahren entwickelt, die darauf beruhen, globalere Ziele zu berücksichtigen, wobei das Setzen und Verfolgen von Zielen auch über größere Bereiche unmittelbar vom Kalkül unterstützt wird. Die Berücksichtigung von Zielen hat sich als ein wesentliches Prinzip für eine

Suchraumeinschränkung herausgestellt.

Literatur

And70 R. Anderson: *Completeness results for E-Resolution,* Proc. Spring Joint Conf., 653-656 (1970)

BG90 L. Bachmair, H. Ganzinger: *On Restrictions of Ordered Paramodulation with Simplification*, Proc. 10th CADE, Kaiserslautern (1990), 427-441

Blä87 K.H. Bläsius: *Equality Reasoning Based on Graphs*, SEKI-REPORT SR-87-01, Fachbereich Informatik, Universität Kaiserslautern, 1987 (oder Dissertation, Fachbereich Informatik, Universität Kaiserslautern, 1986)

Bra75 D. Brand: *Proving Theorems with the Modification Method*, SIAM Journal of Comp., vol 4, No. 4 (1975)

Bun83 A. Bundy: *The Computer Modelling of Mathematical Reasoning,* Academic Press, London (1983)

Dig79 V.J. Digricoli: *Resolution by Unification and Equality,* Proc. 4th Workshop on Automated Deduction, Texas (1979)

HO80 G. Huet, D. Oppen: *Equations and Rewrite Rules: A Survey,* Technical Report CSL-111, SRI International (1980)

HR78 M.C. Harrison, N. Rubin: *Another Generalization of Resolution,* JACM, vol 25, no. 3, July 1978

HR86 J. Hsiang, M. Rusinowitch: *A New Method for Establishing Refutational Completeness in Theorem Proving*, Proc. 8th CADE, Oxford (1986), 141-152

KB70 D. Knuth, P.Bendix: *Simple Word Problems in Universal Algebras*, in: *Computational Problems in Abstract Algebra.* Hrsg. Leech I., Pergamon Press, 263-297 (1970)

LH85 Y. Lim, L.J. Henschen: *A New Hyperparamodulation Strategy for the Equality Relation*, Proc. IJCAI-85, Los Angeles (1985)

Mor69 J.B. Morris: *E-Resolution: An Extension of Resolution to include the Equality Relation*, Proc. IJCAI (1969) 287-294

NSS59 A. Newell, J.C. Shaw, H. Simon: *Report on a General Problem Solving Program,* Proc. Int. Conf. Information Processing (UNESCO), Paris (1959)

Pet83 G. E. Peterson: *A Technique for Establishing Completeness Results in Theorem Proving with Equality,* SIAM Journal of Computing 12,1 (1983), 82-100

Pla81 D. Plaisted: *Theorem Proving with Abstraction,* Artifical Intelligence 16, 47-108 (1981)

Rob65 J. A. Robinson: *A Machine-Oriented Logic Based on the Resolution Principle,* JACM 12 (1965)

Rus87 M. Rusinowitch: *Démonstration automatique par des téchniques de reécriture*,

Thèse de Doctorat d'État en Mathématique, Nancy (1987)

RW69 G. Robinson, L. Wos: *Paramodulation and TP in First Order Theories with Wquality*, Machine Intelligence 4, 135-150 (1969)

Sho78 R. E. Shostak: *An Algorithm for Reasoning About Equality*, CACM, vol 21, no. 7 (1978)

Sib69 E. E. Sibert: *A Machine-oriented Logic Incorporating the Equality Axiom*, Machine Intelligence, vol 4, 103-133 (1969)

Sie86 J. Siekmann: *Universal Unification*, Proc. of European Conf. on Artificial Intelligence (ECAI) (1986)

SS77 R. M. Stallman, G. J. Sussman: *Forward Reasoning and Dependency-directed Backtracking in a System for Computer-aided Circuit Analysis*, Artificial Intelligence, Vol. 9, 2 (1977)

Sti85 M. Stickel: *Automated Deduction by Theory Resolution*, Journal of Automated Reasoning Vol. 1, No. 4 (1985), 333-356

SW80 J. Siekmann, G. Wrightson: *Paramodulated Connectiongraphs*, Acta Informatica 13, 67-86 (1980)

WRCS67 L. Wos, G. Robinson, D. Carson, L. Shalla: *The Concept of Demodulation in Theorem Proving*, J. ACM 14, 698 - 709 (1967)

ZK88 H. Zhang, D. Kapur: *First Order Theorem Proving Using Conditional Rewrite Rules*, Proc. 9th CADE, Argonne (1988), 1-20