4 Termersetzungssysteme

(Norbert Eisinger, Andreas Nonnengart, Axel Präcklein)

4.1 Einführung

Ein typisches Grundprinzip für die Lösung von Gleichheitsproblemen durch menschliche Problemlöser ist die Technik, sukzessive „Gleiches durch Gleiches“ zu ersetzen.

4.1-1 Beispiel: Folgende Gleichungen axiomatisieren die algebraische Struktur „Gruppe“:

(A1)	$\forall x$	$0 + x$	=	x
(A2)	$\forall x$	$-x + x$	=	0
(A3)	$\forall x,y,z$	$(x + y) + z$	=	$x + (y + z)$
(A1')	$\forall x$	$x + 0$	=	x

Daß die vierte Gleichung überflüssig ist, da sie aus den anderen folgt, spielt zunächst keine Rolle. Die Axiome implizieren unter anderem, daß $-(-a) = a$ für beliebiges a gilt. Eine typische Darstellung des Beweises dieser Aussage sieht folgendermaßen aus:

(5)	$-(-a)$			
(6)		=	$-(-a) + 0$	mit A1'
(7)		=	$-(-a) + (-a + a)$	mit A2
(8)		=	$(-(-a) + -a) + a$	mit A3
(9)		=	$0 + a$	mit A2
(10)		=	a	mit A1 ■

Hinter diesem einfachen Beweis verbergen sich aber verhältnismäßig komplizierte Operationen. Wir wollen einmal anhand des Übergangs von Zeile (6) nach Zeile (7) exemplarisch verfolgen, was in den einzelnen Schritten eigentlich genau passiert ist. Wir haben zunächst in dem aktuellen Term $-(-a) + 0$ in Zeile (6) den Unterterm 0 für die Anwendung einer Gleichung ausgewählt. Als anzuwendende Gleichung haben wir die Instanz $-a + a = 0$ von Axiom A2 gewählt, deren rechte Seite mit dem ausgewählten Unterterm übereinstimmt. Diesen haben wir schließlich durch die linke Seite $-a + a$ der gewählten Instanz von A2 ersetzt und so den aktuellen Term $-(-a) + (-a + a)$ der nächsten Zeile erzeugt.

Ein derartiger Übergang von einem Term zu einem anderen mit Hilfe einer Gleichung heißt ein *Termersetzungsschritt.* Wichtig ist, daß ein solcher Schritt nur die anzuwendende Gleichung instantiiert, nicht aber den Term, auf den sie angewandt wird. Dies ist der Unterschied zu einem *Paramodulationsschritt*. Dabei wäre der ausgewählte Unterterm mit einer Seite der Gleichung unifiziert und der Term der nächsten Zeile mit dem allgemeinsten Unifikator instantiiert worden.

Von einem gegebenen Term s aus spannt eine Menge von Gleichungen einen Suchraum von

Termen auf, die durch Folgen von Termersetzungsschritten aus s erzeugt werden können. Ein Nachfolger eines Terms in diesem Suchraum ist bestimmt durch die Auswahl (i) eines Unterterms, (ii) einer Gleichung, (iii) einer Anwendungsrichtung dieser Gleichung und (iv) einer Instantiierung dieser Gleichung, so daß die entsprechende Seite der instantiierten Gleichung mit dem gewählten Unterterm übereinstimmt. Man kann leicht sehen, daß für jeden so aus s erreichbaren Term t die Gleichung $s = t$ und deshalb auch für zwei beliebige aus s erreichbare Terme t_1, t_2 die Gleichung $t_1 = t_2$ aus der Gleichungsmenge folgt.

Aufbauend auf dieser Beobachtung kann man verschiedene Deduktionssysteme für Gleichheitsprobleme konstruieren, also für die Frage, ob eine Gleichung $s = t$ aus einer Menge von Axiomgleichungen folgt (Unifikationsprobleme, das heißt, die Aufgabe des Lösens einer Gleichung, erfordern andere Techniken). Ein positiver Testkalkül ergibt sich beispielsweise, wenn man von der zu beweisenden Gleichung $s = t$ ausgeht und Termersetzungsschritte auf ihre beiden Seiten anwendet, bis man eine Instanz der elementaren Tautologie $x = x$, des logischen Axioms der Reflexivität, erreicht hat. Die Grundtechnik in diesen Deduktionssystemen besteht also darin, auf s und auf t Folgen von Termersetzungsschritten anzuwenden, bis die beiden Folgen mit syntaktisch übereinstimmenden Termen enden. In Beispiel 4.1-1 hatte die zweite Folge die Länge Null. Dieses Verfahren ist aufgrund der obigen Überlegungen offensichtlich *korrekt*, außerdem ist es, weniger offensichtlich, *vollständig*.

Der Suchraum ist allerdings im allgemeinen sehr groß und besitzt recht unangenehme Eigenschaften. Im Beispiel könnte man in Zeile (6) etwa noch einmal Axiom A1' anwenden und den Term $(-(-a) + 0) + 0$ erzeugen. Darauf ist Axiom A1' wieder in der gleichen Weise anwendbar, so daß ein unendlich langer Pfad entsteht, der keine Lösung enthält. Wir hätten die Gleichung A1' aber auch in der Gegenrichtung anwenden und damit wieder den Ausgangsterm $-(-a)$ herleiten können. Man kann jederzeit beliebig viele Nullen an einen Term anfügen und sie anschließend durch Verwendung derselben Gleichung in Gegenrichtung wieder abbauen. Schon bei Betrachtung dieser einen Gleichung ist der Suchraum geradezu überladen mit Zyklen und beliebigen Sackgassen.

Noch schlimmer wirkt sich die Instantiierung der anzuwendenden Gleichungen aus. Beim Übergang von Zeile (6) nach Zeile (7) im Beispiel ist gar nicht zu erkennen, warum man das x in Axiom A2 ausgerechnet durch a ersetzen soll. Jede andere Instantiierung hätte ebenfalls den Term 0 als rechte Seite und wäre somit anwendbar, und dies würde sich erst in Zeile (8) bei der zweiten Anwendung von Axiom A2 als Sackgasse herausstellen. Die aus klassischen Kalkülen bekannten Probleme mit der Instantiierungsregel treten hier also wieder auf. Obendrein hilft die dort gefundene Lösung, die Unifikation, hier zunächst nicht weiter. Der allgemeinste Unifikator von 0 und 0 ist schließlich die Identität und wirkt sich nicht auf x aus.

Das ursprüngliche Ziel bei der Betrachtung von Termersetzungssystemen ist eine Beschränkung der Anwendbarkeit von Gleichungen, die möglichst viele dieser Probleme vermeidet, obendrein eine *nicht-revidierende* Steuerung anstelle einer *revidierenden* erlaubt (englisch *irrevocable / tentative* control regime [Nil80]), und bei alledem die Vollständigkeit erhält.

4.2 Termersetzungsregeln

Eine *Termersetzungsregel* (englisch *rewrite rule*) ist eine gerichtete Gleichung, die in der Form $s \to t$ statt $s = t$ geschrieben wird, und die die *Variablenbedingung* erfüllt, daß alle Variablen von t auch in s vorkommen. Eine Menge von Termersetzungsregeln heißt *Termersetzungssystem* (englisch *rewrite system*). Offensichtlich läßt sich jede Menge von Gleichungen, in denen die Variablen passend verteilt sind (in den interessanteren Fällen sind sie es meistens), in ein Termersetzungssystem umwandeln.

4.2-1 Beispiel: Termersetzungssystem für die Gruppenaxiome:

(R1)	$0 + x$	$\to$	x
(R2)	$-x + x$	$\to$	0
(R3)	$(x + y) + z$	$\to$	$x + (y + z)$
(R1')	$x + 0$	$\to$	x

Natürlich könnte man auch andere Richtungen wählen. Lediglich bei der zweiten Gleichung bleibt keine Wahl, da die Gegenrichtung die Variablenbedingung verletzen würde. ■

Die Intention dahinter ist, daß eine Termersetzungsregel nur in Pfeilrichtung verwendet wird. Der Schritt von (8) nach (9) in Beispiel 4.1-1 wäre also mit dem obigen Termersetzungssystem erlaubt, der von (6) nach (7) dagegen nicht. Die Variablenbedingung hat technische Gründe. Sie bewirkt, daß von einer Gleichung jeweils höchstens eine Instanz existieren kann, deren linke Seite mit einem gegebenen Unterterm r eines aktuellen Terms übereinstimmt. Diese Instanz läßt sich durch *einseitige Unifikation* (englisch *matching*) zwischen der linken Seite der Termersetzungsregel und r berechnen.

Damit tritt das oben anhand des Übergangs von (6) nach (7) beschriebene Instantiierungsproblem nicht mehr auf. Ein Nachfolger eines Terms im Suchraum ist jetzt bereits durch die Auswahl eines Unterterms und einer Termersetzungsregel bestimmt, deren linke Seite mit dem Unterterm einseitig unifizierbar ist. Die Anwendungsrichtung der Regel und ihre Instantiierung liegen damit fest. Insgesamt verkleinert sich die Verzweigungsrate im Suchraum beträchtlich.

Allerdings kann diese Verkleinerung auf Kosten der Anwendbarkeit gehen. Beispielsweise lassen sich weder auf -(-a) noch auf a irgendwelche Termersetzungsregeln des obigen Systems anwenden. Damit kann die Gleichheit zwischen -(-a) und a mit den gerichteten Gleichungen nicht mehr wie vorher gezeigt werden, da es keine Folgen von Termersetzungsschritten gibt, die von diesen beiden Termen ausgehen und mit syntaktisch übereinstimmenden Termen enden.

Die Umwandlung einer Menge von Gleichungen in ein Termersetzungssystem gefährdet also die Vollständigkeit des darauf aufbauenden Deduktionsverfahrens. Dies liegt auf der Hand, denn jede Gleichung $s = t$ entspricht ja zwei Regeln $s \to t$ und $s \leftarrow t$, von denen bei der Umwandlung eine unter den Tisch fällt. Die Korrektheit der Ableitungen bleibt dagegen erhalten, das heißt, die Gleichheit von je zwei durch Folgen von Termersetzungsschritten in beliebiger Richtung verbundenen Termen folgt aus den Gleichungen des Termersetzungssystems. Man betrachtet mit $\to$ eine leichter handzuhabende, aber viel speziellere Relation als die Gleichheit.

Die Vollständigkeit läßt sich nur in eingeschränkten Fällen sicherstellen, in denen die Termersetzungssysteme gewisse Zusatzeigenschaften besitzen.

4.3 Eigenschaften von Termersetzungssystemen

Ein Termersetzungssystem heißt *Noethersch* (oder *terminierend*), wenn es keine unendliche Sequenz $s_0 \to s_1 \to s_2 \to \ldots$ von Termersetzungsschritten zuläßt. Die Bedeutung dieser Eigenschaft liegt darin, daß die wiederholte Anwendung von Termersetzungsregeln eines Noetherschen Systems in jedem Fall mit einem Term endet, auf den keine Regel mehr anwendbar ist. Der Suchraum enthält keine unendlich langen Pfade und insbesondere keine Zyklen. Ein Term ist *irreduzibel* bezüglich des Termersetzungssystems, falls keine Regel auf ihn anwendbar ist, andernfalls *reduzibel*. Wendet man ausgehend von einem Term s so lange Termersetzungsschritte an, bis ein irreduzibler Term erreicht ist, spricht man auch von einer *Reduktion* von s bezüglich des Systems.

Woran kann man nun erkennen, ob ein Termersetzungssystem Noethersch ist? Im allgemeinen ist diese Eigenschaft unentscheidbar. Für die Regeln R1, R2 und R1' des obigen Systems ist jedoch offensichtlich, daß sie nur endlich oft von einem gegebenen Term aus anwendbar sind, da die rechte Seite jeweils weniger Symbole hat als die linke und ein Termersetzungsschritt somit einen kürzeren Term erzeugt. Eine Anwendung der Regel R3 läßt zwar die Anzahl der Symbole des Terms gleich, aber sie verschiebt ein Klammernpaar weiter nach rechts, was auch nur endlich oft möglich ist.

Um eine derartige Intuition zu präzisieren, sucht man nach einer *fundierten Ordnung* (englisch *well-founded ordering*) auf den Termen, also einer kleiner-Beziehung ohne unendliche absteigende Ketten. Wenn durch einen Termersetzungsschritt aus jedem reduziblen Term nur solche Terme entstehen können, die bezüglich dieser Ordnung kleiner sind, ist das Termersetzungssystem Noethersch. Eine naheliegende fundierte Ordnung auf den Termen ergibt sich durch einen einfachen Längenvergleich: Ein Term ist genau dann kleiner als ein anderer, wenn er aus weniger Symbolen besteht (Klammern nicht mitgezählt). Diese Ordnung ist fundiert, da kein Term aus weniger als einem Symbol bestehen kann. Durch strukturelle Induktion über den Aufbau der Terme kann man zeigen, daß aus jedem Term, auf den eine der Regeln R1, R2, R1' anwendbar ist, durch den Termersetzungsschritt ein bezüglich dieser Ordnung kleinerer Term entsteht. Damit ist jedenfalls das entsprechende Teilsystem Noethersch.

Dieses Kriterium ist allerdings immer noch recht unhandlich, da man jeden reduziblen Term untersuchen muß, und davon gibt es unendlich viele. Eigentlich wäre es am einfachsten, lediglich zu prüfen, daß die rechte Seite jeder Termersetzungsregel kleiner als die linke ist. Dies garantiert aber leider nicht, daß die Anwendung der Regel auch in jedem Fall einen kleineren Term erzeugt. Beispielsweise hat die Regel

$$\text{(R)} \qquad (1+1) \cdot x \quad \to \quad x + x$$

eine bezüglich der obigen Ordnung kleinere rechte als linke Seite, aber sie produziert aus $(1+1) \cdot (a+(b+c))$ den größeren Term $(a+(b+c)) + (a+(b+c))$. Um sich auf die Terme in den Regeln beschränken zu können, benötigt man zusätzliche, stärkere Eigenschaften der Ordnung.

Eine fundierte Ordnung heißt monoton, wenn die Ersetzung eines Unterterms durch einen kleineren Term auch den Gesamtterm kleiner macht. Sie heißt instantiierungsstabil, wenn die kleiner-Beziehung zwischen zwei Termen auch bei Anwendung derselben Substitution auf beide erhalten bleibt. Eine fundierte Ordnung mit diesen beiden Eigenschaften nennt man *Termordnung*. Unsere bisherige Ordnung ist zwar monoton, aber, wie die beiden Seiten von Regel R zeigen, nicht instantiierungsstabil. Wir können daraus eine Termordnung definieren, die *Längenordnung*: s sei genau dann kleiner als t, wenn s aus weniger Symbolen besteht als t und jede Variable in s höchstens so oft vorkommt wie in t.

Ein Termersetzungssystem ist genau dann Noethersch, wenn bezüglich irgendeiner Termordnung die rechte Seite jeder Regel kleiner ist als die linke Seite. Die Regeln R1, R2, R1' erfüllen diese Bedingung für die Längenordnung, die Regel R3 jedoch nicht. Das Gesamtsystem ist aber Noethersch. Es kann durchaus sein, daß ein Termersetzungssystem das Kriterium für die meisten „vernünftigen" Termordnungen verletzt und trotzdem Noethersch ist. Man ist deshalb an einem umfangreichen Repertoire von Termordnungen interessiert, mit deren Hilfe man die Terminierungs-Eigenschaft im Einzelfall nachweisen kann. Die bekanntesten dieser Ordnungen sind die „Knuth-Bendix-Ordnungen" und die „rekursiven Pfad-Ordnungen", kurz RPO, oder auch die „Polynom-Ordnungen" [Der 82, Der 85].

Für die zweite grundlegende Eigenschaft von Termersetzungssystemen benutzen wir die übliche Notation $s \rightarrow^* t$ um auszudrücken, daß ausgehend vom Term s der Term t mit einer Folge von null oder mehr Termersetzungsschritten abgeleitet werden kann. Angenommen, mit einem Termersetzungssystem gibt es von einem Term s zwei verschiedene Ableitungen zu Termen t_1 und t_2, das heißt, es gilt $s \rightarrow^* t_1$ und $s \rightarrow^* t_2$. Dann könnte es sein, daß der gesuchte Lösungsterm zwar von t_2 noch erreichbar ist, nicht aber von t_1. Eine nicht-revidierende Auswahlstrategie liefe Gefahr, die erste Alternative zu wählen und damit unwiderruflich in eine Sackgasse zu laufen. Diese Gefahr besteht nicht, wenn in jeder derartigen Situation die divergierenden Ableitungen wieder zusammengeführt werden können, wenn also von t_1 und von t_2 aus ein gemeinsamer Nachfolger t erreichbar ist. Ein Termersetzungssystem heißt *konfluent*, wenn für alle Terme s, t_1, t_2 mit $s \rightarrow^* t_1$ und $s \rightarrow^* t_2$ ein Term t existiert, so daß $t_1 \rightarrow^* t$ und $t_2 \rightarrow^* t$.

Das System in 4.2-1 ist nicht konfluent. Wir können zum Beispiel auf den Term $-0 + 0$ alternativ die Regeln R2 oder R1' anwenden und erhalten $-0 + 0 \rightarrow 0$ und $-0 + 0 \rightarrow -0$. Es gibt aber keinen Term t mit $0 \rightarrow^* t$ und $-0 \rightarrow^* t$, da 0 und -0 offensichtlich verschiedene Terme und beide irreduzibel sind, also keine der Regeln mehr anwendbar ist.

Wenn mit einem System ausgehend von irgendeinem Term s zwei verschiedene irreduzible Terme erreichbar sind, ist das System also nicht konfluent. Umgekehrt heißt das, daß mit einem konfluenten System von jedem Term s aus höchstens ein irreduzibler Term erreichbar sein kann. Wie wir oben gesehen haben, führt in einem Noetherschen System jede Folge von Termersetzungsschritten schließlich zu einem irreduziblen Term. Wenn ein Termersetzungssystem sowohl konfluent als auch Noethersch ist (dann heißt es auch *kanonisch* oder *vollständig*), gibt es demnach zu jedem Term s genau einen irreduziblen Term, mit dem alle Folgen schließlich enden. Diese sogenannte *Normalform* von s wird nach endlich vielen Term-

ersetzungsschritten von s aus erreicht, wie auch immer man die Schritte auswählt.

Für ein kanonisches Termersetzungssystem ist das eingangs beschriebene Deduktionsverfahren vollständig. Von zwei gegebenen Termen s und t ausgehend wendet man jeweils Termersetzungsschritte an, bis dies nicht mehr möglich ist, und erzeugt so die Normalformen s' und t'. Die Gleichung $s = t$ folgt genau dann aus den Gleichungen des Termersetzungssystems, wenn s' und t' syntaktisch übereinstimmen. Da die Reihenfolge der Regelanwendung nur die Länge einer Ableitung, aber nicht ihr Ergebnis beeinflußt, kann das Verfahren mit einer nichtrevidierenden Auswahlstrategie gesteuert werden.

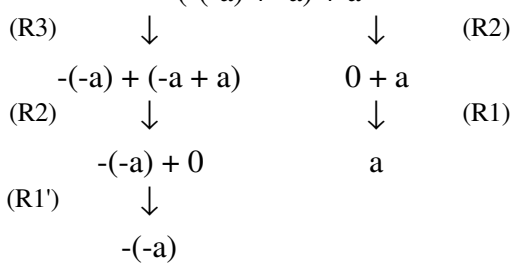
Das Problem ist allerdings, daß man erst einmal ein konfluentes und Noethersches Termersetzungssystem für die gewünschte Menge von Gleichungen finden muß. Wie im Beispiel 4.2-1 für die Gruppenaxiome ist es oft möglich, eine Menge von Gleichungen in ein Termersetzungssystem umzuwandeln, das Noethersch, aber nicht konfluent ist. In solchen Fällen versucht man, das System durch Hinzufügen von daraus folgenden Gleichungen so zu „vervollständigen", daß es konfluent wird, ohne die Terminierungseigenschaft zu verlieren und ohne die Gleichungstheorie zu verändern.

Um die Konfluenz bei diesem Vorgehen festzustellen, nutzt man aus, daß in Noetherschen Systemen bereits eine schwächere Eigenschaft ausreicht. Ein Termersetzungssystem heißt *lokal konfluent*, wenn für alle Terme s, t_1, t_2 mit $s \to t_1$ und $s \to t_2$ ein Term t existiert, so daß $t_1 \to^* t$ und $t_2 \to^* t$. Der Unterschied zur Konfluenz liegt darin, daß man nicht alle durch beliebige Folgen von Termersetzungsschritten aus s erreichbaren Terme t_1 und t_2 betrachten muß, sondern lediglich solche, die durch genau einen Schritt aus s entstehen. Jedes konfluente System ist auch lokal konfluent. In Noetherschen Systemen gilt auch die Umkehrung, so daß man dort nur die lokale Konfluenz sicherstellen muß, um ein kanonisches Termersetzungssystem zu erhalten. Wie im nächsten Abschnitt gezeigt wird, kann man diese Eigenschaft sogar anhand der Regeln selbst erkennen und braucht nicht jeden Term s zu untersuchen.

4.4 Kritische Ausdrücke und kritische Paare

Wenn man in einem gegebenen Gleichheitsbeweis die Gleichheitszeichen durch Pfeile ersetzt, kann man erkennen, welche Schritte die durch ein Termersetzungssystem festgelegte Anwendungsrichtung einhalten. Es ist in dieser Darstellung der Beweise nützlich, die Terme so anzuordnen, daß alle Pfeile nach unten zeigen.

4.4-1 Beispiel: Darstellung des Beweises 4.1-1 mit Richtungen des Termersetzungssystems 4.2-1:



■

Ausgehend von -(-a) wurden also zunächst Gleichungen entgegen der Pfeilrichtung angewandt, wodurch in der Termordnung größere Terme entstanden. Anschließend wurden die Terme durch Verwendung von Gleichungen in Pfeilrichtung wieder verkleinert. Bei dieser Betrachtungsweise scheint die eigentliche Idee des Beweises in dem obersten Term zu liegen, und man kann sich leicht klarmachen, daß nicht-triviale Gleichheitsbeweise im allgemeinen mehrere solcher „Gipfelterme“ benutzen. Diese Terme zeichnen sich dadurch aus, daß zwei verschiedene Regeln auf sie anwendbar sind. Das mit kanonischen Systemen mögliche Beweisverfahren setzt voraus, daß es zwischen zwei als gleich nachzuweisenden Termen auch einen „V-förmigen“ Beweis ohne Zwischengipfel gibt. Wenn dies, wie im Beispiel, nicht der Fall ist, versucht man, die möglichen Gipfel durch zusätzliche Regeln zu „untertunneln“.

Im Beispiel kann der „Gipfelterm“ $(-(-a) + -a) + a$ mit der Regel R2 oder mit der Regel R3 umgeformt werden. Es gibt unendlich viele Terme, auf die beide Regeln anwendbar sind, aber alle sind Instanzen des Terms $(-v + v) + z$ oder enthalten Instanzen davon als Unterterm. Diesen *kritischen Ausdruck* für R2 und R3 kann man durch sogenannte *Superposition* der Regeln berechnen. Um R2 mit R3 zu „überlagern“, hält man die linke Seite $(x + y) + z$ von R3 fest. Für diesen Term und für jeden seiner nicht-variablen Unterterme prüft man, ob er mit der (umbenannten) linken Seite $-v + v$ von R2 unifizierbar ist. Die Anwendung des jeweiligen Unifikators auf die festgehaltene linke Seite ergibt einen kritischen Ausdruck für die beiden Regeln. In unserem Fall gibt es offensichtlich genau eine Möglichkeit, R2 mit R3 zu überlagern, und keine Möglichkeit, R3 mit R2 zu überlagern. Die Superposition der beiden Regeln liefert also genau einen kritischen Ausdruck, nämlich $(-v + v) + z$.

Nach Konstruktion sind sowohl R2 als auch R3 auf den kritischen Ausdruck anwendbar:

$$\begin{array}{ccccc} & & (-v + v) + z & & \\ \text{(R3)} & \swarrow & & \searrow & \text{(R2)} \\ & -v + (v + z) & & 0 + z & \end{array}$$

Durch die beiden Termersetzungsschritte erhalten wir ein *kritisches Paar* von Termen für die Regeln R2 und R3. Nach den Überlegungen in Teil 4.2 folgt die Gleichheit der beiden Terme eines kritischen Paars aus den Gleichungen des Termersetzungssystems, und dasselbe gilt auch für alle daraus abgeleiteten irreduziblen Terme. Der linke der beiden Terme ist bereits irreduzibel, auf den rechten ist R1 anwendbar, und es entsteht der irreduzible Term z. Also folgt auch die Gleichung $-v + (v + z) = z$ aus den vier gegebenen Gruppenaxiomen. Wir können somit eine weitere Regel in unser Termersetzungssystem aufnehmen, nämlich

$$\text{(R4)} \qquad -v + (v + z) \;\rightarrow\; z,$$

die den kritischen Ausdruck „untertunnelt“.

Das erweiterte Termersetzungssystem ist immer noch Noethersch, was man mit einer der vorher diskutierten Termordnungen zeigen könnte. Darüberhinaus gilt: wann immer aus zwei Termen t_1 und t_2 mit den Regeln R1 bis R4 syntaktisch übereinstimmende irreduzible Terme ableitbar sind und somit die Gleichung $t_1 = t_2$ aus den entsprechenden Gleichungen folgt, so folgt diese Gleichung auch aus den Gleichungen des ursprünglichen Systems, da die neue Regel R4 ja selbst eine Folgerung daraus ist.

Wenn wir jetzt in einer weiteren Superposition R2 mit der neuen Regel R4 überlagern, entsteht der kritische Ausdruck $-(-x)+(-x+x)$ mit dem kritischen Paar $-(-x)+0$ und x. Der erste Term läßt sich mit Regel R1' zu $-(-x)$ reduzieren, und wir können folgende Regel zu unserem Termersetzungssystem hinzufügen:

(R6) $\quad -(-x) \quad \rightarrow \quad x.$

Mit dem jetzt erhaltenen System läßt sich die Gleichung $-(-a)=a$ sofort in der beschriebenen Weise zeigen. Der rechte Term ist irreduzibel, aus dem linken entsteht durch Anwendung von R6 derselbe normalisierte Term. Da die Normalformen übereinstimmen, folgt die Gleichung aus den gegebenen Gleichungen.

Was haben wir nun eigentlich bisher getan? Der Schlüssel für einen Gleichheitsbeweis mittels Termersetzungsregeln liegt darin, geeignete „Gipfelterme“ zu finden. Die Konstruktion eines kritischen Ausdrucks durch Superposition zweier Regeln liefert in gewissem Sinn einen allgemeinsten derartigen „Gipfelterm“ für die beiden Regeln. Durch die Hinzunahme einer neuen aus dem kritischen Paar abgeleiteten Regel wird sozusagen ein „Tunnel“ unter dem kritischen Ausdruck hindurch geschaffen. Sollte dies durch wiederholte Superposition für alle kritischen Ausdrücke gelingen, die in beliebigen Gleichheitsbeweisen vorkommen können, entstünde ein System, mit dem stets auch „V-förmige“ Beweise existieren.

Mit den hinzugefügten Regeln versucht man, ein Noethersches, aber noch nicht konfluentes Termersetzungssystem zu einem konfluenten zu vervollständigen. Ein kritischer Ausdruck s mit einem kritischen Paar t_1, t_2 von Termen entspricht gerade einer für die lokale Konfluenz relevanten Situation $s \rightarrow t_1$ und $s \rightarrow t_2$. Mit einem kanonischen System muß ein Term t existieren mit $t_1 \rightarrow^* t$ und $t_2 \rightarrow^* t$. Also wendet man die Regeln des (Noetherschen) Systems auf die beiden Terme des kritischen Paars an, bis irreduzible Terme t_1', t_2' entstehen. Stimmen diese syntaktisch überein, so ist s unkritisch für die lokale Konfluenz und man kann den nächsten kritischen Ausdruck untersuchen. Wenn keine weiteren kritischen Ausdrücke übrigbleiben, ist das System lokal konfluent und, da es Noethersch ist, auch konfluent und damit kanonisch. Wenn t_1', t_2' aber syntaktisch verschieden sind, so folgt zumindest die Gleichung $t_1' = t_2'$ aus den Gleichungen des Systems. Fügt man jetzt die Regel $t_1' \rightarrow t_2'$ hinzu, dann ist in dem erweiterten System der Term t_2' sowohl von t_1 als auch von t_2 aus erreichbar und damit ist s jetzt auch unkritisch für die lokale Konfluenz. Jede mit dem erweiterten System ableitbare Gleichung folgt auch aus den Gleichungen des ursprünglichen Systems.

Allerdings muß man sicherstellen, daß bei dieser Vervollständigung auch das erweiterte System noch Noethersch bleibt. Dies ist der Fall, wenn die neue Regel bezüglich der ursprünglich gewählten Termordnung eine kleinere rechte als linke Seite besitzt. Außerdem entstehen durch die neue Regel weitere Superpositionsmöglichkeiten, so daß zusätzliche kritische Paare gebildet werden können. Es ist allerdings nicht gesagt, daß die fortgesetzte Superposition auch irgendwann einmal endet.

Damit sind drei Ergebnisse der Vervollständigung möglich. Im günstigsten Fall lassen sich keine neuen kritischen Ausdrücke bilden, und das Verfahren terminiert *erfolgreich* mit einem kanonischen Termersetzungssystem. Dieses kann in sehr einfacher Weise als Entscheidungsverfahren dafür eingesetzt werden, ob eine Gleichung aus den ursprünglich gegebenen Axiom-

gleichungen folgt. Das Verfahren entscheidet das Wortproblem in der durch die Axiome spezifizierten algebraischen Struktur. Zweitens kann es passieren, daß die beiden aus einem kritischen Paar abgeleiteten irreduziblen Terme sich nicht zu einer Termersetzungsregel „richten“ lassen, da keiner bezüglich der vorgegebenen Termordnung kleiner ist als der andere. In diesem Fall kann das Verfahren nicht sinnvoll weiterarbeiten und terminiert *erfolglos*. Dann bleibt noch die Möglichkeit, den gesamten Prozeß mit einer „stärkeren“ Ordnung neu zu starten und zu hoffen, daß er diesmal erfolgreich terminiert. Schließlich ist es möglich, daß das Vervollständigungsverfahren überhaupt nicht terminiert, weil immer wieder neue „echt“ kritische Paare und daraus auch immer wieder neue Regeln erzeugt werden. Dann existiert im allgemeinen für die ursprünglich gegebene Axiommenge kein endliches kanonisches Termersetzungssystem, was wegen der Unentscheidbarkeit der Gleichheit leider vorkommt.

4.5 Das Knuth-Bendix-Verfahren

Das oben beschriebene Vervollständigungsverfahren wurde erstmals von D. Knuth und P. Bendix vorgestellt [KB70] und trägt daher den Namen dieser Autoren. Es läßt sich folgendermaßen beschreiben:

Eingabe: Eine endliche Menge von Axiomgleichungen und eine Termordnung

Ergebnis: Ein kanonisches Termersetzungssystem für die gegebenen Gleichungen oder eine Meldung „Die Termordnung ist zu schwach“

```
initialisiere leere Regelmenge;
while Gleichungsmenge nicht leer do
    entnimm eine Gleichung aus der Gleichungsmenge;
    reduziere diese Gleichung mit den Regeln der Regelmenge;
    if reduzierte Gleichung nicht von der Form t = t für einen beliebigen Term t then
    if reduzierte Gleichung läßt sich mit der Termordnung nicht richten
    then return „Die Termordnung ist zu schwach“
    else bilde aus der reduzierten Gleichung eine neue Regel;
         nimm diese Regel in die Regelmenge auf;
         reduziere alle Regeln der Regelmenge untereinander;
         if eine Regel verletzt die Termordnung
         then entferne die Regel aus der Regelmenge;
              if die Regel hat nicht die Form t = t
              then nimm die Regel als Gleichung in die Gleichungsmenge auf
         fi fi;
         bilde alle kritischen Paare zwischen der neuen Regel und den übrigen Regeln;
         nimm die kritischen Paare als Gleichungen in die Gleichungsmenge auf;
    fi fi
od
return Regelmenge
```

Die Beschreibung enthält einige Indeterminismen, zum Beispiel bei der Auswahl der

Gleichungen. Obwohl das Ergebnis stark davon abhängt, wie diese genau behandelt werden, spricht man oft vom *Knuth-Bendix-Algorithmus*. Das ist gerechtfertigt, da für alle Eingaben dieselbe „faire“ Abarbeitungsstrategie gewählt werden kann. Mit den Axiomen A1, A2 und A3 als Eingabe können beispielsweise folgendermaßen neue Regeln erzeugt werden (der Inhalt der Gleichungsmenge ist nicht protokolliert):

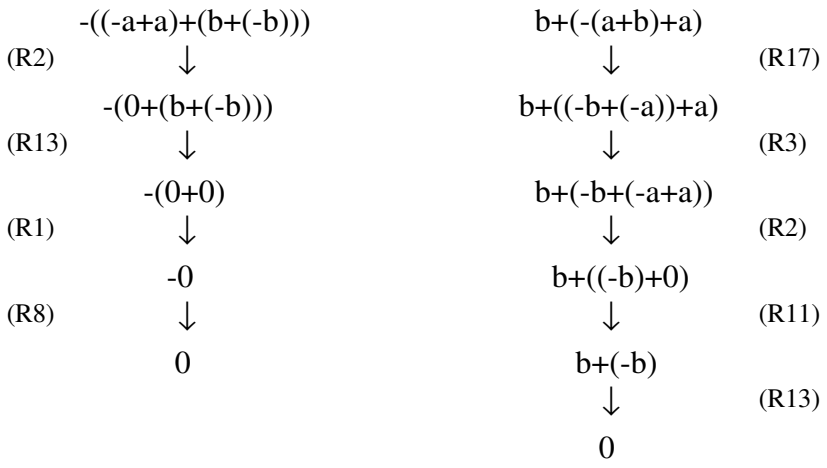
aus	kritischer Ausdruck	angewandte Regeln	neue Regel
A1			R1: 0+x → x
A2			R2: -x+x → 0
A3			R3: (x+y)+z → x+(y+z)
R3, R2	(-x+x)+y	R3 und R2, R1	R4: -x+(x+y) → y
R1, R4	(-0)+(0+x)	R1 und R4	R5: -0+x → x
R2, R4	-(-x)+(-x+x)	R2 und R4	R6: -(-x)+0 → x
R5, R4	-(-0)+((-0)+x)	R5 und R4	R7: -(-0)+x → x
R7, R2	-(-0)+(-0)	R7 und R2	R8: -0 → 0
		R5 wird mit R8, R1 zu x = x und entfällt.	
		R7 wird mit R8, R8, R1 zu x = x und entfällt.	
R6, R4	-(-(-x))+(-(-x)+0)	R6 und R4	R9: -(-(-x))+x → 0
R4, R4	-(-x)+(-x+(x+y))	R4 und R4	R10: -(-x)+y → x+y
		R9 wird mit R10, R2 zu 0 → 0 und entfällt.	
		R6 wird mit R10 zu	R11: x+0 → x
R11, R10	-(-x)+0	R11 und R10, R11	R12: -(-x) → x
		R10 wird mit R12 zu x+y = x+y und entfällt.	
R12, R2	-(-x)+(-x)	R12 und R2	R13: x+(-x) → 0
R12, R4	-(-x)+(-x+y)	R12 und R4	R14: x+(-x+y) → y
R13, R3	(x+y)+(-(x+y))	R3 und R13	R15: x+(y+(-(x+y))) → 0
R15, R4	-y+(y+(x+(-(y+x))))	R4 und R15,R11	R16: x+(-(y+x)) → -y
		R15 wird mit R16, R13 zu 0 = 0 und entfällt.	
R16, R4	-y+(y+(-(x+y)))	R4 und R16	R17: -(x+y) → (-y)+(-x)
		R16 wird mit R17, R14 zu -y = -y und entfällt.	

Nun findet das Verfahren keine weitere Gleichung mehr, die in eine Regel umgewandelt werden kann, und terminiert deshalb mit Erfolg. Insgesamt entsteht das folgende kanonische Termersetzungssystem für die Gruppenaxiome A1, A2, A3:

R1:	0+x	→	x
R2:	-x+x	→	0
R3:	(x+y)+z	→	x+(y+z)
R4:	-x+(x+y)	→	y
R8:	-0	→	0
R11:	x+0	→	x
R12:	-(-x)	→	x
R13:	x+(-x)	→	0
R14:	x+(-x+y)	→	y
R17:	-(x+y)	→	(-y)+(-x)

Um jetzt zum Beispiel zu entscheiden, ob die Gleichung -((-a+a)+(b+(-b))) = b+(-(a+b)+a) aus

den Gruppenaxiomen folgt, berechnen wir einfach mit diesem Termersetzungssystem die Normalformen der beiden Terme, wobei die Reihenfolge der Regelanwendung beliebig ist.



Da die beiden Normalformen syntaktisch übereinstimmen, folgt die Gleichung aus den Axiomen A1, A2 und A3.

Für die Gleichung 0+(a+b) = 0+(b+a) erhalten wir dagegen die beiden Normalformen a+b und b+a, und da diese syntaktisch verschieden sind, folgt die Gleichung nicht aus den Gruppenaxiomen (denn nicht in jeder Gruppe gilt das Kommutativgesetz).

4.6 Knuth-Bendix-Vervollständigung modulo einer Äquivalenzrelation

Das Standard-Knuth-Bendix-Verfahren setzt eine feste Termordnung voraus, bezüglich der die Regeln gerichtet werden. Leider existiert keine stärkste Termordnung, die alle anderen möglichen Termordnungen impliziert und die man deshalb immer verwenden könnte. Bei erfolgloser Terminierung bleibt deshalb nichts anderes übrig, als das gesamte Verfahren mit einer für den vorliegenden Fall hoffentlich besser geeigneten Termordnung noch einmal von vorne zu beginnen.

In manchen Fällen ist aber auch diese Vorgehensweise aussichtslos, da gewisse Gleichungen grundsätzlich durch keine Termordnung zu einer Termersetzungsregel gerichtet werden können. Gäbe es zum Beispiel eine Termordnung, bezüglich der eine Seite des Kommutativgesetzes x+y = y+x kleiner ist als die andere, so müßte wegen der Instantiierungsstabilität der Term z+z echt kleiner sein als er selbst, was den Eigenschaften einer Ordnungsrelation widerspricht. Jede Termordnung kann deshalb nur eine Halbordnung sein, in der die Terme x+y und y+x unvergleichbar sind. Enthält eine Menge von Axiomen eine Gleichung zwischen zwei derart unvergleichbaren Termen oder entsteht eine solche Gleichung im Zuge der Vervollständigung, dann stößt das Verfahren an seine Grenzen.

Zur Verringerung dieser Schwierigkeiten wurden verschiedene Erweiterungen des Standard-Knuth-Bendix-Verfahrens untersucht. Die Grundidee ist meistens, gewisse Gleichungen nicht in Regeln umzuformen, sonder irgendwie anders zu behandeln.

Die Erweiterung von Huet [Hue77] basiert auf einer Verallgemeinerung des Konfluenzbegriffs

auf die Konfluenz modulo einer Äquivalenzrelation. Die nicht in Regeln umgewandelten Gleichungen werden als Erzeugende einer Äquivalenzrelation $\approx$ auf den Termen betrachtet. Das Termersetzungssystem ist *konfluent modulo* $\approx$, wenn für alle Terme s_1, s_2, t_1, t_2 mit $s_1 \approx s_2$ und $s_1 \to^* t_1$ und $s_2 \to^* t_2$ zwei Terme $t_1' \approx t_2'$ existieren mit $t_1 \to^* t_1'$ und $t_2 \to^* t_2'$. Wie man leicht sieht, entspricht unsere bisherige Konfluenz gerade der Konfluenz modulo der Identität.

Um ein Termersetzungssystem so zu vervollständigen, daß es konfluent modulo $\approx$ wird, müssen im Knuth-Bendix-Verfahren nicht nur paarweise linke Regelseiten, sondern zusätzlich linke Regelseiten mit linken oder rechten Seiten der nicht in Regeln umgewandelten Axiome überlagert werden, um kritische Paare und somit neue Regeln zu erzeugen.

4.6-1 Beispiel: Betrachten wir folgendes Axiom- und Regelsystem für die Vervollständigung nach Huet:

(C)	x+y	=	y+x
(A)	(x+y)+z	=	x+(y+z)
(R1)	0+x	$\to$	x

Das nur aus R1 bestehende Termersetzungssystem ist bereits kanonisch, und das Standard-Knuth-Bendix-Verfahren würde, ohne Berücksichtigung von A und C, an dieser Stelle erfolgreich terminieren. Durch Superposition der linken Seite von C mit R1 entsteht jedoch ein weiteres kritisches Paar $x + 0$ und x, aus dem sich sofort eine neue Regel ergibt:

(R1')	x+0	$\to$	x

Die rechte Seite von C kann auch mit R1 überlagert werden, das kritische Paar wird aber nun zu x = x reduziert und generiert deshalb keine neue Regel. Zwischen A und R1 oder R1' lassen sich ebenfalls kritische Paare bilden, die aber alle auf die Form y+z = y+z oder umbenannte Varianten führen. Auch die anderen Superpositionsmöglichkeiten bilden keine „echt“ kritischen Paare mehr, so daß für das aus R1 und R1' bestehende System die Konfluenz modulo der durch A und C erzeugten Äquivalenzrelation $\approx$ garantiert ist.

Um nun zum Beispiel zu entscheiden, ob die Gleichung b+(a+0) = (a+0)+(0+b) aus A, C und der R1 zugrundeliegenden Gleichung folgt, berechnen wir mit den Regeln R1 und R1' die beiden Normalformen b+a und a+b. Diese irreduziblen Terme stimmen nicht syntaktisch überein, aber sie gehören zur selben Äquivalenzklasse unter A und C, das heißt, $b+a \approx a+b$ gilt. Also folgt die Gleichung aus den Axiomen. ■

Leider gibt es in manchen Fällen Probleme mit dieser Methode, etwa wenn man die Regel

(R2)	-x+x	$\to$	0

hinzunimmt. Benutzen wir nur C als Nicht-Regel-Gleichung und R2 als einzige Regel, so entsteht durch die Vervollständigung lediglich noch

(R2')	x+(-x)	$\to$	0.

Das System R2, R2' ist aber nicht konfluent modulo der durch C definierten Äquivalenzrelation $\approx$. Beispielsweise gilt $-(b+a)+(a+b) \approx -(a+b)+(a+b)$, wobei der erste Term irreduzibel ist und

aus dem zweiten der irreduzible Term 0 ableitbar ist. Die Terme -(b+a)+(a+b) und 0 stehen nicht in der Relation ≈ zueinander, und aus ihnen sind im Widerspruch zur Konfluenzforderung auch keine zwei Terme mit dieser Eigenschaft ableitbar. Damit läßt sich also die offensichtlich aus C und R2 folgende Gleichung -(b+a)+(a+b) = 0 nicht beweisen.

Die Entstehung eines konfluenten Regelsystems (modulo ≈) ist dann garantiert, wenn alle Regeln linkslinear sind, d.h. auf der linken Seite jede Variable höchstens einmal enthalten. R2 verstößt gegen diese ziemlich restriktive Bedingung, was zeigt, wie stark die Anwendbarkeit des Huetschen Verfahrens dadurch beschränkt wird. Das Beispiel kann aber mit einigen Erweiterungen des Knuth-Bendix-Verfahrens behandelt werden (siehe zum Beispiel Hsiang und Rusinowitch [HR86b] sowie Bachmair und Dershowitz [BD87]).

Man kann auch die Reduktionsrelation so erweitern, daß außer den Regeln Gleichungen berücksichtigt werden. Ist zum Beispiel die linke Seite einer nicht richtbaren Gleichung s = t mit dem zu reduzierenden Term einseitig unifizierbar mit einer Substitution μ, und ist außerdem μ(t) bezüglich der Termordnung kleiner als μ(s), dann kann die Instanz der Gleichung als Regel μ(s) →μ(t) zur Reduktion herangezogen werden.

4.7 Das Knuth-Bendix-Verfahren als Beweisprozedur für Gleichungen

Die ursprüngliche Bedeutung des Knuth-Bendix-Verfahrens liegt in seiner Verwendbarkeit für Gleichheitsbeweise. Wird aus einer Menge von Axiomgleichungen ein endliches kanonisches Termersetzungssystem erzeugt, ist dadurch ein effizientes Entscheidungsverfahren zur Lösung von Gleichheitsproblemen gegeben. Diese Beweisprozedur basiert auf dem *Ergebnis* des Knuth-Bendix-Algorithmus. Wenn der Algorithmus nicht terminiert, kann man ihn in gewisser Weise selbst als Beweisprozedur verwenden, da er fortwährend neue Regeln produziert, die für einen Gleichheitsbeweis von Nutzen sein können und die Rolle von Lemmata spielen. Mit der jeweils gerade erzeugten Regelmenge lassen sich beliebige Paare von Eingabetermen reduzieren und somit auch Gleichheitsbeweise führen. Sind die beiden gefundenen irreduziblen Terme identisch, folgt die Gleichheit der beiden Eingabeterme in jedem Fall aus den Axiomen. Nur ist nicht gewährleistet, daß zwei Eingabeterme verschieden sind, wenn die gefundenen irreduziblen Terme nicht übereinstimmen.

Es kann sogar gezeigt werden [Hue81], daß jede aus den Axiomen folgende Gleichheit nachgewiesen werden kann, wenn man das Verfahren nur ausreichend viele Regeln erzeugen läßt. Man erhält also eine Semi-Entscheidungsprozedur, wenn man in die Reduzierungsphase des Knuth-Bendix-Algorithmus die Reduktion der Anfrageterme integriert und bei nicht übereinstimmenden irreduziblen Termen die Regelgenerierung fortsetzt. Falls der Algorithmus erfolglos terminiert (zum Beispiel, weil das Kommutativitätsaxiom vorkommt), können keine weiteren Regeln mehr erzeugt werden. Damit ergibt sich ein zwar korrektes, aber nicht vollständiges Verfahren zum Führen von Gleichheitsbeweisen. Varianten, in denen eine erfolglose Terminierung vermieden wird, sind unter dem Schlagwort „unfailing Knuth-Bendix-Algorithm“ untersucht worden [Lan77, HR86b].

Auf demselben Prinzip basiert die Idee, mit Hilfe kanonischer Termersetzungssysteme Theorie-

Unifikation bezüglich eben dieser Theorie durchzuführen. Diese Technik wird „Narrowing“ genannt und im Abschnitt über Unifikationstheorie genauer erläutert.

4.8 Knuth-Bendix-Vervollständigung mit Theorieunifikation

Der Ansatz von Peterson und Stickel [PS81] unterscheidet ebenfalls zwischen Axiomen, die eine Äquivalenzrelation ≈ auf den Termen definieren, und solchen, die in Regeln umgewandelt werden. Die Äquivalenzklassen bestehen also aus Termen, die durch Anwendung der Nicht-Regel-Gleichungen ineinander umgewandelt werden können. Die Regeln sollen statt auf Terme auf Äquivalenzklassen von Termen angewandt werden, und man strebt ein System an, das kanonisch auf diesen Klassen ist. Da bei der Anwendung aber immer nur ein einzelner Term vorliegt, muß man beliebige andere Repräsentanten derselben Klasse mit berücksichtigen. Das leistet in bestimmten Fällen die Theorieunifikation (siehe Abschnitt Unifikationstheorie). Man kann die Nicht-Regel-Gleichungen als Axiomatisierung der Theorie betrachten, bezüglich der unifiziert wird. Wenn zwischen diesen Gleichungen und den Regeln eine bestimmte Kompatibilität besteht, wird die Regelmenge im wesentlichen nach dem Standardverfahren vervollständigt, nur verwendet man für die Superposition zur Bildung kritischer Paare anstelle des üblichen einen speziellen Unifikationsalgorithmus für die vorgegebene Theorie. Eine Verallgemeinerung dieser Art von Vervollständigung auf beliebige Theorien mit gegebenem Unifikationsalgorithmus wurde von Bachmair und Dershowitz entwickelt [BD87].

4.8-1 Beispiel: Wir benutzen das Axiom- und Regelsystem aus Beispiel 4.6-1 für die Theorievervollständigung nach Peterson und Stickel. Die Gleichungen A und C werden durch einen AC-Unifikationsalgorithmus behandelt. Die einzige Superpositionsmöglichkeit ist die triviale von R1 mit sich selbst, die auch unter der AC-Unifikation keinen „echt“ kritischen Ausdruck ergibt. Also ist das nur aus R1 bestehende System konfluent auf den durch A und C definierten Äquivalenzklassen.

Um zu entscheiden, ob die Gleichung b+(a+0) = (a+0)+(0+b) aus A, C und der R1 zugrundeliegenden Gleichung folgt, wenden wir die einzige Regel R1 auf die beiden Terme an, solange dies geht. Mit einseitiger AC-Unifikation unter Kommutativität ist R1 auf b+(a+0) anwendbar, und es entsteht der normalisierte Term b+a. Auf den zweiten Term ist, wieder mit einseitiger AC-Unifikation, die Regel zweimal anwendbar und erzeugt den normalisierten Term a+b. Die beiden Normalformen stimmen zwar nicht syntaktisch überein, aber sie gehören zur selben Äquivalenzklasse unter A und C (dies kann man zum Beispiel dadurch feststellen, daß man sie mit dem AC-Unifikationsalgorithmus unifiziert und die Identität erhält). Die aus den Klassen der Ausgangsterme abgeleiteten normalisierten Klassen sind somit identisch. Also folgt die Gleichung aus den Axiomen. ■

Im Gegensatz zum Verfahren von Huet erlaubt die Methode von Peterson und Stickel auch eine Vervollständigung des Systems R1, R2 unter A und C und damit das Beweisen von Gleichungen in Abelschen Gruppen. Die Herleitung eines konfluenten Systems wird im folgenden Beispiel gezeigt.

4.8-2 Beispiel: Die Theorie ist gegeben durch:

(C)	x+y	=	y+x
(A)	(x+y)+z	=	x+(y+z)

Als Eingabegleichungsmenge verwenden wir die beiden Gruppenaxiome A1 und A2. Damit ergibt sich folgender Programmablauf für eine Abelsche Gruppe. Wir geben diesmal zwei kritische Ausdrücke an, wenn die AC-äquivalenten Terme intuitiv sehr verschieden aussehen.

aus	kritische Ausdrücke	angewandte Regeln	neue Regel
A1			R1: 0+x → x
A2			R2: -x+x → 0
R1, R2	-0+0	R1 und R2	R3: -0 → 0
R2, A	x+(-y+y)	A und R1, R2	R4: (x+(-y))+y → x
R4, R4	(x+(-(-x)))+(-x), (-(-x)+(-x))+x	R4 und R4	R5: -(-x) → x
R4, R4	(-y+(-(y+x)))+(y+x), ((x+(-y))+y)+(-(y+x))	R4 und R4	R6: x+(-(y+x)) → -y
R4, R6	-y+(-((x+y)+-y))	R6 und R4	R7: -(x+y) → (-y)+(-x)
		R6 wird mit R7, R4 zu -y = -y und entfällt.	

Nun findet das Verfahren keine weitere Gleichung mehr, die in eine Regel umgewandelt werden kann, und terminiert deshalb mit Erfolg. Insgesamt entsteht das folgende kanonische Termersetzungssystem für die Gruppenaxiome A1, A2 modulo der Theorie {C, A}:

R1:	0+x	→	x
R2:	-x+x	→	0
R3:	-0	→	0
R4:	(x+(-y))+y	→	x
R5:	-(-x)	→	x
R7:	-(x+y)	→	(-y)+(-x)

■

Zur Reduktion von Termen bezüglich eines gegebenen Regelsystems reicht es aus, einen geeigneten Algorithmus für einseitige Unifikation unter der Theorie anzuwenden, was im allgemeinen wesentlich effizienter ist als die Verwendung eines Algorithmus für beidseitige Unifikation [GD88].

Diese Art der Vervollständigung scheitert aber oft daran, daß sie nur in Zusammenarbeit mit einem geeigneten Theorieunifikationsalgorithmus funktioniert. Es gibt zwar für einige wichtige Theorien bekannte Unifikationsalgorithmen, z.B. für Assoziativität und Kommutativität, aber häufig muß man den gewünschten Algorithmus erst selbst erfinden. Insbesondere lassen sich Algorithmen für verschiedene Theorien im allgemeinen nicht zu einem für die Vereinigung der Theorien kombinieren. Noch schlimmer wirkt sich aus, daß in manchen Theorien (etwa der

Assoziativität allein) unendlich viele allgemeinste Unifikatoren für ein Paar von Termen existieren können. Dann muß man einen unvollständigen Algorithmus verwenden und ist nicht mehr sicher, daß ein konfluentes System auch erreicht wird.

Einen Algorithmus nur für Kommutativität zu verwenden ist selten nützlich, da die Ordnungen durch die Theorie eingeschränkt werden. Im Fall der Polynomordnungen erfordert die Verträglichkeit mit der Kommutativität eine Einschränkung auf symmetrische Polynome. Damit läßt sich dann allerdings die Assoziativitätsgleichung nicht mehr richten.

Die Diskussion über die Probleme von AC- (assoziativ und kommutativ) und AC1-Unfikation (AC mit neutralem Element 1) läßt sich am besten anhand von Beispielen führen.

4.8-3 Beispiel: Die Aufgabe sei, die beiden Terme f(x,y) und f(u,v) zu unifizieren. Sowohl im theoriefreien Fall als auch für den Fall, daß man nur Assoziativität betrachtet, haben sie den eindeutigen allgemeinsten Unifikator $\{x\leftarrow u, y\leftarrow v\}$. Für AC1 und AC erwartet man intuitiv die beiden Unifikatoren $\sigma_1 = \{x\leftarrow u, y\leftarrow v\}$ und $\sigma_2 = \{x\leftarrow v, y\leftarrow u\}$. Diese reichen allerdings nicht aus, da die Vertauschungsinformation vom AC1-Unifikationsalgorithmus in den Unifikator eingearbeitet werden muß. Damit ergibt sich:

$\sigma = \{x\leftarrow f(v_1,v_2), y\leftarrow f(v_3,v_4), u\leftarrow f(v_1,v_3), v\leftarrow f(v_2,v_4)\}$ mit v_1, v_2, v_3, und v_4 als neuen Variablen. σ ist allgemeiner als σ_1 und σ_2, da $\sigma_1 =_{AC1} \sigma \circ \{v_1\leftarrow u, v_2\leftarrow 1, v_3\leftarrow 1, v_4\leftarrow v\}$ und $\sigma_2 =_{AC1} \sigma \circ \{v_1\leftarrow 1, v_2\leftarrow v, v_3\leftarrow u, v_4\leftarrow 1\}$, wobei 1 das zu f gehörende neutrale Elenent ist.

Die Menge von AC-Unifikatoren wird dann durch Instantiierung mit dem entsprechenden neutralen Element gewonnen:

$\tau_1 = \{x\leftarrow f(v_1,v_2), y\leftarrow f(v_3,v_4), u\leftarrow f(v_1,v_3), v\leftarrow f(v_2,v_4)\}$,

$\tau_2 = \{x\leftarrow v_2, y\leftarrow f(v_3,v_4), u\leftarrow v_3, v\leftarrow f(v_2,v_4)\}$,

$\tau_3 = \{x\leftarrow v_1, y\leftarrow f(v_3,v_4), u\leftarrow f(v_1,v_3), v\leftarrow v_4\}$,

$\tau_4 = \{x\leftarrow f(v_1,v_2), y\leftarrow v_4, u\leftarrow v_1, v\leftarrow f(v_2,v_4)\}$,

$\tau_5 = \{x\leftarrow f(v_1,v_2), y\leftarrow v_3, u\leftarrow f(v_1,v_3), v\leftarrow v_2\}$,

$\tau_6 = \{x\leftarrow u, y\leftarrow v\}$ und

$\tau_7 = \{x\leftarrow v, y\leftarrow u\}$. ■

Die Menge von AC-Unifikatoren hat also den Vorteil, daß kleinere Unifikatoren ausgewählt werden können, während das System bei AC1-Unifkation gezwungen ist, immer den kompliziertesten zu wählen. Desweiteren ist die Ordnung für AC1-Vervollständigung so stark eingeschränkt, daß sich praktisch keine auftretende Gleichung mehr richten läßt, wie das folgende Beispiel zeigt.

4.8-4 Beispiel: Nehmen wir die schon mehrmals verwendete Gleichung $-(-x) = x$. Sie gilt in den von uns betrachteten Gruppenbeispielen und läßt sich sowohl theoriefrei als auch mit AC-Unifikation zu $-(-x) \rightarrow x$ richten. Sie ließe sich auch bei Verwendung von AC1-Unifikation mit geeigneter Ordnung richten, allerdings schlägt hier das „Prinzip der allgemeinsten Unifikatoren“ zu, und führt zur Herleitung der allgemeineren Gleichung $u+(z+(-(z+(y+(-(x+y))))))) = x+u$, die mit keiner AC1-verträglichen Ordnung richtbar ist. ■

Ein weiterer Ansatz von Jouannaud und Kirchner [JK84] umfaßt sowohl nichtabbrechende Vervollständigung als auch Theorievervollständigung. Diesen betrachten wir hier nicht näher, da er keine zusätzlichen neuen Ideen integriert.

4.9 Das Knuth-Bendix-Verfahren als Beweisprozedur für Klauseln

Es gibt zwei Möglichkeiten, das Knuth-Bendix-Verfahren als Beweisprozedur für beliebige Formeln der Prädikatenlogik erster Stufe einzusetzen.

Die Grundidee der ersten Variante geht auf Hsiang und Dershowitz [HD83] zurück und besteht darin, Prädikatensymbole und Junktoren als Boolesche Funktionen aufzufassen, so daß die Unterscheidung zwischen Termen und Formeln aufgehoben wird. Um eine Klauselmenge zu widerlegen, geht man von einem kanonischen Termersetzungssystem für die Boolesche Algebra aus und fügt für jede Klausel C in der Menge eine Termersetzungsregel $C \rightarrow \mathit{true}$ hinzu. Das so erhaltene System wird dann mit Hilfe des Knuth-Bendix-Verfahrens vervollständigt. Die Klauselmenge ist genau dann widersprüchlich, wenn dabei die Regel $\mathit{false} \rightarrow \mathit{true}$ entsteht. (Die Vorgehensweise entspricht in manchen Fällen der Konsistenzmethode, nämlich dann, wenn genau die beiden Konstruktorterme *false* und *true* betrachtet werden.)

Es gibt verschiedene Abwandlungen dieser Grundidee, sei es durch die Erweiterung um Gleichheitstheorien oder durch den Verzicht auf Klauselnormalformbildung. Eigenschaften dieser Verfahren werden in [Pau85] untersucht. Die wesentliche Aussage ist jeweils, daß aus einer Formelmenge eine Formel $\mathcal{F}$ genau dann folgt, wenn bei der Vervollständigung genügend viele Regeln erzeugt werden können, um $\mathcal{F}$ zu *true* zu normalisieren. In diesem Rahmen kann man die Resolution als ein spezielles Vervollständigungsverfahren auffassen, welches nur bestimmte kritische Paare berechnet.

Diese Art der Vervollständigungsmethode ist insofern eleganter als die übliche Kombination der Resolution mit weiteren Ableitungsregeln wie der Paramodulation, als sie einen einheitlichen Grundmechanismus für die Behandlung von Gleichheits- und anderen Beweisproblemen verwendet. Wann welche Methode Effizienzvorteile bringt, ist derzeit noch nicht restlos geklärt, obwohl im Hornklauselfall die Vervollständigungsmethode leichte Nachteile gegenüber der Resolution aufzuweisen scheint [Die86]. Der Hauptgewinn der Übersetzung der beiden Verfahren ineinander liegt wohl darin, daß damit ein Weg geschaffen wird, Weiterentwicklungen und Verbesserungen der einen Methode auch der anderen zugänglich zu machen.

Die zweite Möglichkeit der Verwendung der Vervollständigungsidee beim Beweisen wurde erstmals von Peterson veröffentlicht [Pet83] und dann von Hsiang und Rusinowitch [HR86a], Rusinowitch [Rus87], Kapur und Zhang [ZK88] sowie von Bachmair und Ganzinger [BG90] sukzessive verbessert. Wir stellen hier die Variante von Kapur und Zhang vor, da diese sich am einfachsten erklären läßt.

Man kann sich bei der Betrachtung des Kalküls auf Paramodulation zwischen Klauseln beschränken, da Resolution als Spezialfall der Paramodulation gesehen werden kann. Der

normale Resolutionsschritt, bei dem aus zwei Klauseln $L \vee L_1 \vee ...$ und $\neg L' \vee L'_1 \vee ...$ mit $\sigma(L) = \sigma(L')$ die Klausel $\sigma(L_1 \vee ... \vee L'_1 \vee ...)$ hergeleitet wird, kann folgendermaßen als Paramodulationsschritt dargestellt werden: Aus $L = \mathit{true} \vee L_1 \vee ...$ und $L' = \mathit{false} \vee L'_1 \vee ...$ ergibt sich der Paramodulant $\sigma(\mathit{true} = \mathit{false} \vee L_1 \vee ... \vee L'_1 \vee ...)$, aus dem das Literal $\mathit{true} = \mathit{false}$ trivialerweise als falsch gelöscht werden kann.

Der nächste Schritt in der Idee ist, die Reduktionsordnung (Termordnung) von Termen auf Literale zu erweitern. Dies geschieht, indem man die Prädikatssymbole genauso behandelt wie Funktionssymbole. Das bezüglich dieser Ordnung maximale Literal in einer Klausel wird nun als Kopf einer Regel ausgewählt, die restlichen Literale werden im Hinblick auf die Reduktion mit Hilfe solcher Regeln Bedingung genannt. Die Klausel $L \vee L_1 \vee ... \vee L_n$ mit maximalem Literal L wird dann als Regel L unless $L_1 \vee ... \vee L_n$ geschrieben. Ist L eine richtbare Gleichung, so verwenden wir wieder $\rightarrow$ statt = als Gleichheitssymbol. Sind mehrere maximale, das heißt unvergleichbare, Literale in einer Klausel, werden mehrere Regeln gebildet. Analog dazu kann man, falls der Kopf der Klausel aus einer nichtrichtbaren Gleichung besteht, zwei Regeln generieren. Man erhält ein widerlegungsvollständiges Paramodulationsverfahren, wenn man auf den linken Seiten von Kopfliteralen paramoduliert.

Die Reduktion kann auf Klauseln mit Bedingungen erweitert werden, indem ein Entscheidungsverfahren für notwendige Bedingungen an die Gültigkeit der Bedingungsliterale angegeben wird. Eine Möglichkeit wäre die Reduktion der Bedingungen mit den vorhanden Gleichungen und einer nachfolgenden Überprüfung, ob das Literal zu *false* reduziert wurde.

Dieses Verfahren hat zwei große Vorteile. Zum einen wird praktisch ganz normale Resolution ausgeführt, wenn keine Gleichungen vorhanden sind, zum anderen läuft der Knuth-Bendix-Algorithmus ab, wenn nur einfache Gleichungen als Axiome gegeben sind. In beiden Randfällen wird also automatisch ein bekanntermaßen gutes Verfahren angewandt, man hat damit eine alle Vorteile erhaltende Kopplung der beiden Verfahren.

Im folgenden wird wieder anhand eines Beispiels das Verfahren erläutert. Es ist der Sammlung von Pelletier [Pel86] entnommen.

4.9-1 Beispiel: Wenn in einer Welt mit zwei Objekten für zwei verschiedene Konstanten a und b sowohl P(a) als auch P(b) gilt, dann gilt P für alle Objekte. Es ergeben sich fünf Klauseln:

(C1)	$P(a)$	Es gilt P(a).
(C2)	$P(b)$	Es gilt P(b).
(C3)	$(a=b) = \mathit{false}$	a und b sind verschieden.
(C4)	$\forall x \; x=c_1 \vee x=c_2$	Es gibt zwei Objekte in der Welt.
(C5)	$\neg P(c_3)$	Negation von „P gilt für alle Objekte“.

Im Beweis geben wir triviale Umformungen $t=t \rightarrow \mathit{true}$, $\mathit{true}=\mathit{false} \rightarrow \mathit{false}$ sowie das Weglassen von *false*-Literalen nicht an. Für die Operatoren wurde die Ordnung $a > b > c_1 > c_2 > c_3 > P$ zugrundegelegt.

aus	kritischer Ausdruck	angewandte Regeln	neue Regel
A1			R1: P(a) → *true*
A2			R2: P(b) → *true*
A3			R3: a=b → *false*
A4			R4: x=c_1 unless x=c_2
T1			R5: P(c_3) → *false*
R4, R1	P(a)	R4 und R1	R6: a → c_2 unless P(c_1) = *true*
R6, R1	P(a)	R6 und R1	R7: P(c_1) → *true* unless P(c_2) = *true*
R4, R2	P(b)	R4 und R2	R8: b → c_2 unless P(c_1) = *true*
R4, R5	P(c_3)	R4 und R5	R9: P(c_1) → *false* unless c_3=c_2
R6, R3	a	R6 und R3	R10: c_2=b → *false* unless P(c_1) = *true*
R10, R8	c_2=b	R10 und R8	R11: P(c_1) → *true*
		R9 wird mit R11 zu	R12: c_2 → c_3
		R4 wird mit R12 zu	R13: x=c_1 unless x=c_3
R13, R3	a=b	R13 und R3	R14: a → c_3 unless (c_1=b) = *false*
R14, R1	P(a)	R14 und R1, R5	R15: (c_1=b) → *false*
R13, R15	c_1=b	R13 und R15	R16: b → c_3
		R2 wird mit R16, R5 zu *false*, womit das Theorem bewiesen ist.	■

Bei der Bildung von R6 ergibt sich zunächst durch Überlappung von x in R4 und a in R1 die Klausel P(c_1)=*true* ∨ x=c_2. Das maximale Literal in dieser Klausel ist x=c_2, weshalb dann die Regel R6 konstruiert wird.

4.10 Das Knuth-Bendix-Verfahren als Induktionsbeweiser

Eine andere wichtige Anwendung des Knuth-Bendix-Verfahrens liegt im Bereich von Induktionsaussagen. Zu einer Menge von Axiomgleichungen kann es verschiedene Modelle geben. Zeichnet man einige Funktions- und Konstantensymbole als sogenannte Konstruktorsymbole mit gewissen Eigenschaften aus, legt man damit ein bis auf Isomorphie eindeutiges *Standardmodell* fest, dessen Universum gerade aus den mit den ausgezeichneten Symbolen aufgebauten Konstruktortermen besteht (man kann auch ohne Konstruktorsymbole ein *initiales Modell* auszeichnen). Bisher haben wir nur *allgemeine Folgerungen* aus einer Menge von Gleichungen betrachtet, das sind Gleichungen, die in allen Modellen der Menge gelten. Oft interessiert man sich auch für *induktive Folgerungen*, das sind Gleichungen, die im Standardmodell, aber nicht unbedingt in anderen Modellen der Menge gelten. Es gibt also im allgemeinen mehr induktive als allgemeine Folgerungen aus einer Menge von Gleichungen.

4.10-1 Beispiel: Die natürlichen Zahlen kann man mit den Konstruktorsymbolen 0 und suc (Nachfolgerfunktion) aufbauen, wobei 0 der Nachfolger von keiner und jede andere natürliche Zahl der Nachfolger von genau einer natürlichen Zahl ist. Die Addition in den natürlichen Zahlen wird durch die beiden folgenden Gleichungen spezifiziert:

(A0)	0 + z	=	z
(A1)	suc(x) + y	=	suc(x + y)

Die Gleichung $suc(0) + y = suc(y)$ ist eine allgemeine Folgerung aus diesen Axiomen. Man kann sie leicht beweisen, indem man erst A1 und dann A0 auf die linke Seite anwendet. Die Gleichungen $(x+y)+z = x+(y+z)$ und $x+y = y+x$, die Assoziativität und die Kommutativität von +, sind dagegen induktive, aber keine allgemeinen Folgerungen. Sie gelten im Standardmodell mit den „normalen" natürlichen Zahlen als Universum und der „normalen" Addition als Interpretation von +, aber zum Beispiel nicht im folgenden Modell:

Das Universum sei die Menge $\{0, 1, 2, \ldots\} \cup \{\ldots, -1.5, -0.5, 0.5, 1.5, 2.5, \ldots\}$, also die „normalen" natürlichen Zahlen vereinigt mit den um ein Halb nach oben verschobenen ganzen Zahlen. Der Nachfolger sei jeweils die um eins größere Zahl, dann ist jedes Element bis auf 0 der Nachfolger von genau einem anderen. Die Funktion + interpretieren wir als die „normale" Addition, außer wenn zwei „Punkt-5-Zahlen" x und y verknüpft werden; dann sei das Ergebnis die „Punkt-5-Zahl" $x - (y - 0.5)$. Man verifiziert leicht, daß damit alle Axiome erfüllt sind. In einer derartigen Interpretation gilt $(9.5 + 4.5) + 3.5 = 5.5 + 3.5 = 2.5$ aber $9.5 + (4.5 + 3.5) = 9.5 + 1.5 = 8.5$, außerdem $9.5 + 4.5 = 5.5$ aber $4.5 + 9.5 = -4.5$. Diese Interpretation erfüllt also die Gleichungen A0 und A1, aber weder die Assoziativität noch die Kommutativität. ■

Um induktive Folgerungen ableiten zu können, muß man geeignete *Induktionsaxiome* hinzufügen. Für unser Beispiel drücken diese aus, daß alle natürlichen Zahlen über die Nachfolgerabbildung von 0 aus erreichbar sind. Dies ist für die „Punkt-5-Zahlen" nicht der Fall, die obige Interpretation erfüllt die um die Induktionsaxiome erweiterte Formelmenge also nicht. Die Wirkung der Induktionsaxiome besteht gerade darin, nur das Standardmodell zuzulassen und damit alle anderen Interpretationen auszuschließen. Leider sind Induktionsaxiome nicht in Prädikatenlogik erster Stufe formulierbar, sie lassen sich durch Formeln erster Stufe nur *approximieren*. Diese approximierenden Formeln können in einem klassischen Beweisverfahren zwar als zusätzliche Axiome verwendet werden, um induktive Folgerungen zu erhalten. Jedoch ist diese Vorgehensweise unvollständig, da man immer Sätze findet, die nicht ableitbar sind, obwohl sie im Standardmodell gelten.

Die *Konsistenz-Methode* [KM87] verzichtet auf Approximationen und behandelt die Induktionsaxiome auf andere Weise. Sie wird auch als *induktionslose Induktion* bezeichnet. Eine Gleichung $s = t$ ist genau dann eine induktive Folgerung aus einer Menge E von Axiomgleichungen, wenn durch $E \cup \{s = t\}$ nicht mehr Grundterme identifiziert werden können als durch E selbst (weniger können es sowieso nicht sein). Diese Eigenschaft ist im allgemeinen unentscheidbar, aber das Knuth-Bendix-Verfahren kann die Überprüfung in Spezialfällen ermöglichen. Angenommen, E und $E \cup \{s = t\}$ lassen sich in kanonische Termersetzungssysteme R und R' überführen. Die Gleichung $s = t$ ist genau dann eine induktive Folgerung aus E, wenn alle bezüglich R normalisierten Grundterme auch bezüglich R' normalisiert sind. Dies ist äquivalent zu der Bedingung, daß auf jede Grundinstanz der linken Seite jeder Regel in R' auch eine Regel aus R anwendbar ist. Dann heißt R' *quasi-reduzibel* bezüglich R.

Die Quasi-Reduzibilität von R' bezüglich R ist zwar entscheidbar, aber im allgemeinen nur mit unvertretbarem Aufwand. Es gibt jedoch einfache hinreichende Bedingungen. Wenn die Regeln in R eine bestimmte Form haben, bestehen alle Grundterm-Normalformen bezüglich R nur aus Konstruktorsymbolen. Kommt nun in der linken Seite jeder Regel von R' ein Funktionssymbol

vor, das kein Konstruktorsymbol ist, so gilt dies auch für jede Grundinstanz davon. Diese Grundinstanzen können also bezüglich R nicht normalisiert sein, das heißt, wenigstens eine Regel aus R muß darauf anwendbar sein, und R' ist quasi-reduzibel bezüglich R.

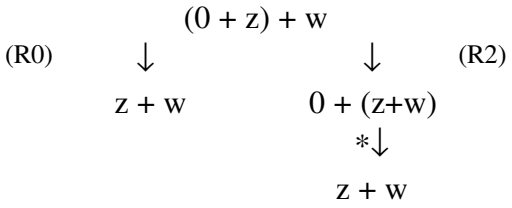
4.10-2 Beispiel: Beweis der Assoziativität der Addition in den natürlichen Zahlen mit der Konsistenzmethode. Aus den obigen Axiomen für die Addition ergibt sich das folgende kanonische Termersetzungssystem R (es existieren gar keine kritischen Paare):

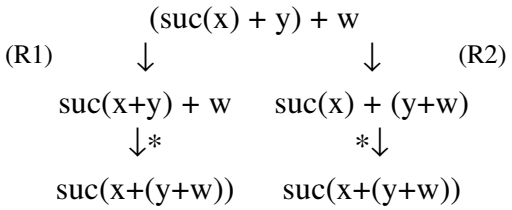
(R0)	0 + z	→	z
(R1)	suc(x) + y	→	suc(x+y)

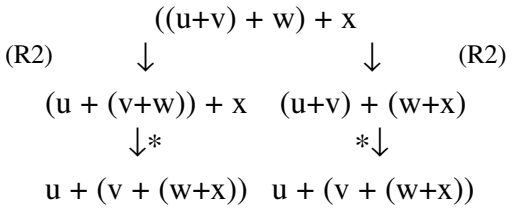
Durch Hinzunahme der Regel für die Assoziativität erhalten wir das erweiterte System R':

(R0)	0 + z	→	z
(R1)	suc(x) + y	→	suc(x+y)
(R2)	(u+v) + w	→	u + (v+w)

Auch R' ist kanonisch, es gibt nämlich genau drei kritische Paare mit jeweils syntaktisch übereinstimmenden Normalformen:







Um die Assoziativität als induktive Folgerung aus den übrigen Axiomen nachzuweisen, müssen wir zeigen, daß R' quasi-reduzibel bezüglich R ist. Die Regeln in R sind von einer Form, die garantiert, daß alle Grundterm-Normalformen bezüglich R nur aus Konstruktorsymbolen bestehen. Da die linke Seite jeder Regel in R' das Funktionssymbol + enthält, ist R' quasi-reduzibel bezüglich R, und die Assoziativität von + ist eine induktive Folgerung aus den Axiomen.■

Die Konsistenzmethode versagt, wenn das Knuth-Bendix-Verfahren keine konfluenten und terminierenden Termersetzungssysteme für die beiden Axiommengen konstruieren kann. Die Kommutativität der Addition läßt sich deshalb nicht so einfach zeigen wie die Assoziativität. Aus diesem Grunde wurde in den letzten Jahren an verschiedenen Verbesserungen des

Verfahrens zum Nachweis induktiver Eigenschaften gearbeitet [Göb85a, Göb85b, Fri86, KM87]. Alle neuen Ansätze gehen von der Tatsache aus, daß für diese Anwendung die Konfluenz der Termersetzungssysteme auf den Grundtermen ausreicht. Diese Einschränkung verringert die Anzahl der neu zu generierenden Regeln oftmals beträchtlich.

Anmerkung: Einige der Beispiele und Einzelheiten der Darstellung, insbesondere die Sichtweise der kritischen Paare in Teil 4.4, wurden aus einem Aufsatz von A. J. J. Dick entnommen [Dic84].

Literatur

BD87 L. Bachmair, N. Dershowitz: *Completion for Rewriting modulo a Congruence*, Proc. 2nd RTA (Rewriting Techniques and Applications), Bordeaux (1987)

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

Der82 N. Dershowitz: *Orderings for Term Rewriting Systems*, Theoretical Computer Science 17 (1982)

Der85 N. Dershowitz: *Termination*, Proc. 1st RTA (Rewriting Techniques and Applications), Dijon, Springer LNCS 202 (1985)

Dic84 A. J. J. Dick: *Automated Equational Reasoning and the Knuth-Bendix Algorithm: an Informal Introduction*, Research Report DoC 84/21, Imperial College, London (1984)

Die86 R. Dietrich: *Relating Resolution and Algebraic Completion for Horn Logic*, Proc. 8th CADE, Oxford (1986), 61-78

DM79 N. Dershowitz, Z. Manna: *Proving Termination with Multiset Orderings*, Communications of the ACM 22,8 (1979)

Fri86 L. Fribourg: *A Strong Restriction of the Induction Completion Procedure*, Proc. of the ICALP 86, Rennes (1986)

GD88 B. Gramlich, J. Denzinger: *Efficient AC-Matching Using Constraint Propagation*, SEKI Report SR-88-15 (1988)

Göb85a R. Göbel: *Completion of Globally Finite Term Rewriting Systems for Inductive Proofs*, Proc. GWAI 85, Dassel (1985)

Göb85b R. Göbel: *A Specialized Knuth-Bendix-Algorithm for Inductive Proofs*, Proc. of Combinatorial Algorithms in Algebraic Structures, Universität Kaiserslautern (1985)

HD83 J. Hsiang, N. Dershowitz: *Rewrite Methods for Clausal and Non-clausal Theorem Proving*, Proc. 10th ICALP (1983)

HH80 G. Huet, J. M. Hullot: *Proofs by Induction in Equational Theories with Constructors*, Proc. 5th CADE, Les Arcs (1980)

HL78 G. Huet, D. S. Lankford: *On the Uniform Halting Problem of Term Rewriting*

Systems, Rapport Laboria 283, IRIA (1978)

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

HR86a J. Hsiang, M. Rusinowitch: *A New Method for Establishing Refutational Completeness in Theorem Proving*, Proc. 8th CADE, Oxford (1986), 141-152

HR86b J. Hsiang, M. Rusinowitch: *On Word Problems in Equational Theories*, Dept. of Computer Science, SUNY at Stony Brook, New York (1986), oder in *Computational Problems in Abstract Algebra*, ed. J. Leech, 263-297, Oxford, Pergamon Press (1987)

Hue77 G. Huet: *Confluent Reductions: Abstract Properties and Applications to Term Rewriting Systems*, 18th IEEE Symposium on Foundations of Computer Science (1977)

Hue81 G. Huet: *A Complete Proof of Correctness of the Knuth-Bendix-Completion-Algorithm*, Journal of Computer and System Science 23 (1981)

Hul80 J. M. Hullot: *Canonical Forms and Unification*, Proc. 5th CADE, Les Arcs (1980)

JK84 J.-P. Jouannaud, H. Kirchner: *Completion of a Set of Rules Modulo a Set of Equations*, Proc. 11th ACM Conference on Principles of Programming Languages, Salt Lake City, Utah (1983)

Jou83 J.-P. Jouannaud: *Confluent and Coherent Equational Term Rewriting Systems: Application To Proofs In Abstract Data Types*, Centre de Rechérche en Informatique de Nancy et Greco de Programmation (1983)

KB70 D. Knuth, P. Bendix: *Simple Word Problems in Universal Algebras*, Computational Problems In Abstract Algebras, Ed. Leech, J., Pergamon Press (1970)

KM87 D. Kapur, D. Musser: *Proof by Consistency*, Artificial Intelligence 31,2 (1987)

Lan77 D. S. Lankford: *Canonical Inference*, Report ATP-32, University of Texas, Austin (1977)

Lan79 D. S. Lankford: *On Proving Term Rewriting Systems are Noetherian*, Louisiana Tech. University, Math. Dept. Rep MTP-3 (1979)

Lan81 D. S. Lankford: *A Simple Explanation of Inductionless Induction*, Louisiana Tech. University, Math. Dept. Rep MTP-14 (1981)

Mus80 D. R. Musser: *On Proving Inductive Properties of Abstract Data Types*, Proc. 7th POPL Conference, Las Vegas (1980)

Nil80 N. Nilsson: *Principles of Artificial Intelligence*, Tioga, Palo Alto, CA (1980)

Pau85 E. Paul: *Equational Methods in First Order Predicate Calculus*, Journal of Symbolic Computation 1,7 (1985), 7-29

Pel86 F. J. Pelletier: *Seventy-five Problems for Testing Automatic Theorem Provers*, Journal of Automated Reasoning 2 (1986), 191-216

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

PS81 G. E. Peterson, M. E. Stickel: *Complete Sets of Reductions for Some Equational Theories*, Journal of the ACM 28,2 (1981)

Rus87 M. Rusinowitch: *Démonstration automatique par des téchniques de reécriture*, Thèse de Doctorat d'État en Mathématique, Nancy (1987)

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