3 Unifikationstheorie

(Hans-Jürgen Bürckert)

3.1 Robinson-Unifikation

Die meisten Deduktionssysteme benutzen als Schlußregel Robinsons Resolutionsprinzip [Rob 65]. Deren Hauptoperation ist die Unifikation zweier Literale mit verschiedenem Vorzeichen und gleichem Prädikatensymbol. Unter (syntaktischer) Unifikation versteht man dabei die Ersetzung der in den Literalen vorkommenden freien Variablen durch Terme, so daß die Literale (ohne die Vorzeichen) danach zeichenweise gleich sind. Eine etwas andere Sichtweise erhält man, wenn man statt der Literale deren Argumentlisten unifiziert. Da die Literale bereits das gleiche Prädikatensymbol besitzen, liefern beide Vorgehensweisen dasselbe Resultat. Die Aufgabe, die beiden Argumentlisten, etwa $(s_1,\ldots,s_n)$ und $(t_1,\ldots,t_n)$, zu unifizieren, kann nun aber als das Auflösen des Gleichungssystems $\Gamma = \langle\, s_1 = t_1,\ldots,\ s_n = t_n \,\rangle$ nach den vorkommenden Variablen betrachtet werden. Dies geschieht nach den folgenden Rechenregeln, die bereits von J. Herbrand so ähnlich formuliert wurden [Her 30]. (x, y, z_i sind Variablen, p_i, q_i und s_i sind beliebige Terme, t ist ein Term, aber keine Variable)

(1) Entferne jede Gleichung t = t aus Γ.

(2) Ersetze jede Gleichung $f(p_1,\ldots,p_n) = f(q_1,\ldots,q_n)$ durch die Gleichungen $p_1 = q_1,\ldots,\ p_n = q_n$.

(3) Substituiere für jede Gleichung x = t, wenn x nicht unter den Variablen des Terms t ist, alle weiteren Vorkommen von x in Γ durch t. Analog für Gleichungen x = y.

(4) Ersetze jede Gleichung t = x durch x = t.

(5) Falls eine Gleichung $f(p_1,\ldots,p_n) = g(q_1,\ldots,q_m)$ in Γ vorkommt, gibt es keine Lösung.

(6) Falls eine Gleichung x = t in Γ ist, so daß x in t vorkommt, dann gibt es keine Lösung.

Das Gleichungssystem ist gelöst, wenn es nur noch aus Gleichungen $z_i = s_i$ $(1 \le i \le N)$ besteht, wobei die Variablen z_i sonst nicht mehr in Γ vorkommen. Der Unifikator der Argumentlisten, von denen wir ausgingen, ist dann die Substitution $\{z_1 \leftarrow s_1,\ldots,\ z_N \leftarrow s_N\}$. Diese Substitution ist gerade der allgemeinste Unifikator der Argumentlisten, der auch von Robinson's bekanntem Unifikationsalgorithmus berechnet wird [Rob 65]. In Regel (1) genügt es auch, nur Gleichungen der Form x = x zu entfernen (Regel (2) entfernt auch Gleichungen c = c, da man Konstanten c üblicherweise als nullstellige Funktionen betrachtet).

Beispiel: Das Gleichungssystem $\langle$ f(x, g(a, y)) = f(h(y), g(y, a)), g(x, h(y)) = g(z, z) $\rangle$ wird durch mehrmaliges Anwenden der Regel (2) in das System $\langle$ x = h(y), a = y, y = a, x = z, h(y) = z $\rangle$, mit den Regeln (3) und (4) in $\langle$ x = h(a), y = a, a = a, h(a) = h(a), z = h(a) $\rangle$, und schließlich mit Regel (1) in das gelöste System $\langle$ x = h(a), y = a, z = h(a) $\rangle$ überführt. Das

Ausgangssystem hat also den allgemeinsten Unifikator $\{x \leftarrow h(a), y \leftarrow a, z \leftarrow h(a)\}$, der die beiden Seiten der Gleichungen jeweils gleich macht. ■

3.2 Theorieunifikation

Wie bereits in den Beiträgen über die Gleichheitsbehandlung ausgeführt wurde, bereiten Formeln, in denen die Gleichheit vorkommt, ziemliche Schwierigkeiten beim automatischen Beweisen - und nicht nur dort. Sehr schön sieht man das zum Beispiel bei Formeln, die die *Kommutativität* gewisser Funktionssymbole definieren, d.h. wenn etwa $\forall x\ \forall y.\ g(x, y) = g(y, x)$ gilt. Diese Kommutativitätsformel führt unter Umständen zu einem ständigen Vertauschen der Argumente von Termen mit führendem kommutativen Funktionssymbol. Daher wurde schon relativ früh versucht, solche Gleichungsformeln aus den Formelmengen zu entfernen und durch entsprechend veränderte Schlußregeln zu ersetzen. G. Plotkin schlug vor, die Resolutionsregel dahingehend zu ändern, daß die Unifikation durch eine die herausgenommenen Gleichungsformeln berücksichtigende Unifikation ersetzt wird. Er gab auch die wesentlichen Kriterien an, unter denen das geschehen kann, ohne daß die Vollständigkeit des Kalküls verloren geht [Plo 72]. Vorausgesetzt, das Gleichheitsprädikat kommt nur in unären Klauseln vor, d.h. die Klauselmenge enthält endlich viele Klauseln $l_1 = r_1, \ldots, l_n = r_n$, und es tritt sonst in keiner weiteren Klausel auf, dann darf die Unifikation durch eine sogenannte *Theorieunifikation* ersetzt werden, die die Gleichungsaxiome $l_1 = r_1, \ldots, l_n = r_n$ berücksichtigt. Diese relativ starke Einschränkung kann noch wesentlich abgeschwächt werden, was aber hier die Darstellung der Idee nur unnötig komplizieren würde.

Beispiel: Unter Berücksichtigung der Kommutativität der Funktion g erhält man für das Gleichungssystem $\langle\ f(x, g(a, y)) = f(h(y), g(y, a)),\ g(x, h(y)) = g(z, z)\ \rangle$ in obigem Beispiel folgenden Unifikator $\{x \leftarrow h(y), z \leftarrow h(y)\}$, der offensichtlich sogar allgemeiner ist als der oben berechnete. Man erhält den obigen Unifikator durch weitere Instantiierung, indem man y durch a substituiert. ■

Man will natürlich auch jetzt eine allgemeinste, unifizierende Substitution haben, allerdings zeigt sich, daß dies im allgemeinen nicht mehr möglich ist. Es kann mehr als einen allgemeinsten Theorieunifikator geben. Man sieht dies sehr schön am Beispiel der Kommutativität. Das Gleichungssystem $\langle\ g(x\ y) = g(a\ b)\ \rangle$ hat die beiden unabhängigen Lösungen $\{x \leftarrow a, y \leftarrow b\}$ und bei vertauschten Untertermen $\{x \leftarrow b, y \leftarrow a\}$, wenn g kommutativ ist. Es existiert keine weitere, also auch keine gemeinsame, allgemeinere Lösung. Es kann sogar noch unangenehmer sein, wenn etwa eine Funktion f assoziativ ist, wenn also $\forall x\ y\ z.\ f(x, f(y, z)) = f(f(x, y), z)$ gilt. In diesem Fall hat etwa das Gleichungssystem $\langle\ f(x, a) = f(a, x)\ \rangle$ unendlich viele unabhängige Lösungen:

$$\{x \leftarrow f(a, a)\},\ \{x \leftarrow f(a, f(a, a))\},\ \{x \leftarrow f(a, f(a, f(a, a)))\}, \ldots$$

Alle weiteren Lösungen sind im Sinne der Assoziativität von f gleich zu einer dieser Lösungen, d.h. der für x substituierte Term ist bis auf Umklammerung einer der hier angegebenen. Allerdings erspart man sich auch in diesem unangenehmen Fall gegenüber der Resolution (bzw. Paramodulation) ohne Theorieunifikation etwa das ständige Umklammern von Termen mit

assoziativem Funktionssymbol.

Eine solche Menge E := $\{l_1 = r_1,\ldots,l_n = r_n\}$ von unären Klauseln mit dem Gleichheitsprädikat (*Gleichheitsaxiome*) induziert eine *Gleichheitstheorie* auf der Menge aller Terme, der sogenannten *Termalgebra* $\mathbb{T}$, über einer *Signatur* $\mathbb{F}$, d.h. einer Menge von Funktionssymbolen verschiedener Stelligkeiten $n \geq 0$, die mindestens die Symbole der Gleichheitsaxiome in E enthält. Diese Gleichheitstheorie ist die kleinste Äquivalenzrelation $=_E$ auf $\mathbb{T}$, die alle Termpaare (l_i, r_i) für $l_i = r_i$ aus E enthält und abgeschlossen ist gegenüber der Termbildung und der Substitution von Variablen:

(i) $s_1 =_E t_1,\ldots,s_n =_E t_n$ und f ist ein n-stelliges Funktionssymbol
$$\Rightarrow \quad f(s_1,\ldots,s_n) =_E f(t_1,\ldots,t_n)$$

(ii) $s =_E t$ und σ ist eine Substitution
$$\Rightarrow \quad \sigma s =_E \sigma t.$$

Man kann zeigen, daß zwei Terme genau dann in dieser Relation $=_E$ sind, wenn man ihre Gleichheit aus den Axiomen E ableiten kann. Wir bezeichnen die Theorie mit dem Paar (E, $\mathbb{F}$) oder auch kurz nur mit E und nennen sie E-Gleichheit über der Signatur $\mathbb{F}$.

Beispiel: Sei C := $\{g(x, y) = g(y, x)\}$ die Theorie der Kommutativität. C-gleiche Terme sind dann:

$$g(a, b) =_C g(b, a) \text{ oder } f(x, g(a, b), z) =_C f(x, g(b, a), z).$$ ■

Ein Unifikationsproblem unter der Theorie E - ein *E-Unifikationsproblem* - ist dann ein Gleichungssystem $\Gamma = \langle s_1 = t_1,\ldots,s_n = t_n \rangle_E$. Eine Lösung von Γ ist eine Substitution σ mit $\sigma s_i =_E \sigma t_i$ (für alle i mit $1 \leq i \leq n$) und sie heißt *E-Unifikator* von Γ; die Menge der E-Unifikatoren bezeichnen wir mit $U_E(\Gamma)$ oder auch $U_E(s_1 = t_1,\ldots,s_n = t_n)$. Vorausgesetzt man hat ein Verfahren, das für jedes Gleichungssystem die Lösungsmenge $U_E(\Gamma)$ – oder besser noch eine möglichst kleine repräsentative Teilmenge $\mu U_E(\Gamma)$ – berechnet, dann kann man die Resolutionsregel wie folgt abwandeln:

Klausel1: $P(s_1,\ldots,s_n),\ K_1,\ldots,\ K_m$

Klausel2: $\neg\ P(t_1,\ldots,t_n)\ ,\ L_1,\ldots,\ L_k \qquad \sigma \in U_E(s_1 = t_1,\ldots,s_n = t_n)$

E-Resolvente: $\sigma K_1,\ldots,\ \sigma K_m,\ \sigma L_1,\ldots,\ \sigma L_k$

Im allgemeinen sind die Mengen $U_E(\Gamma)$ unendlich und man möchte sie daher durch eine möglichst kleine repräsentative Teilmenge ersetzen. Bei der üblichen syntaktischen Unifikation - sie entspricht der Unifikation bezüglich der Theorie mit der leeren Axiomatisierung - gelingt dies sogar mit einelementigen Teilmengen, wie Robinson gezeigt hat (siehe oben). Wir betrachten also *minimale und vollständige* Mengen von E-Unifikatoren – auch Mengen *allgemeinster* E-Unifikatoren oder Lösungsbasen genannt – $\mu U_E(\Gamma)$, die den folgenden Bedingungen genügen:

(1) Korrektheit: $\mu U_E(\Gamma) \subseteq U_E(\Gamma)$

(2) Vollständigkeit: Für alle $\delta \in U_E(\Gamma)$ existiert ein $\sigma \in \mu U_E(\Gamma)$ mit $\delta x =_E \lambda\sigma x$ (für alle x in Γ)

(3) Minimalität: Für alle $\sigma,\tau \in U_E(\Gamma)$ mit $\tau x =_E \lambda\sigma x$ (für alle x in Γ) ist $\sigma = \tau$

Das heißt: (1) alle allgemeinsten E-Unifikatoren lösen Γ, (2) jeder beliebige E-Unifikator ist Instanz eines allgemeinsten E-Unifikators, und (3) verschiedene allgemeinste E-Unifikatoren sind keine Instanzen voneinander. Unifikatormengen $cU_E(\Gamma)$ mit (1) und (2) heißen auch vollständige Mengen von E-Unifikatoren. $U_E(\Gamma)$ selbst ist trivialerweise eine vollständige Lösungsmenge.

Beispiel: Sei C := {g(x, y) = g(y, x)} wieder die Theorie der Kommutativität und betrachten wir das C-Unifikationsproblem $\langle$ g(a, h(u)) = g(v, w) $\rangle_C$. Dann sind sowohl δ := {u ← a, v ← a, w ← h(a)} als auch σ := {v ← a, w ← h(u)} und τ := {v ← h(u), w ← a} C-Unifikatoren, aber δ ist eine Instanz von σ mit der Instantiierungssubstitution λ := {u ← a}, während σ und τ allgemeinste C-Unifikatoren sind, in diesem Beispiel bereits alle. ■

3.3 Eigenschaften von Lösungsmengen

Die folgenden Theoreme zeigen, daß wir hier die „richtige“ Form der Repräsentation gewählt haben. Das Eindeutigkeitstheorem garantiert, daß verschiedene minimale und vollständige Mengen von E-Unifikatoren eines Gleichungssystems Γ äquivalent im Sinne der folgenden durch wechselseitige Instanzenbildung induzierten *E-Äquivalenz* von Substitutionen sind: $\sigma \equiv_E \tau \Leftrightarrow \exists\ \lambda,\mu$ mit $\sigma x =_E \lambda\tau x$ und $\tau x =_E \mu\sigma x$ für alle x in Γ, d.h. wenn jede eine Instanz der anderen ist. Das Repräsentationstheorem zeigt, daß die allgemeinsten Unifikatoren *genau* die Menge der Unifikatoren repräsentieren: Eine Substitution löst Γ genau dann, wenn sie Instanz eines allgemeinsten Unifikators ist. Das Vererbungstheorem schließlich besagt, daß man die Lösungsbasen schrittweise berechnen kann: Um ein Gleichungssystem zu lösen, kann man die allgemeinsten Unifikatoren einer der Gleichungen berechnen, diese auf die anderen Gleichungen anwenden und die resultierenden Restsysteme analog lösen. Prinzipiell könnte man auch andere Formen der Repräsentation der Lösungsmengen verwenden. Für diese sollten dann aber entsprechende Theoreme gelten.

Eindeutigkeitstheorem:

a) Die Lösungsbasen eines Gleichungssystems Γ haben gleiche Kardinalität und unterscheiden sich höchstens um E-Äquivalenz ihrer Elemente.

b) Ersetzt man Elemente einer Lösungsbasis von Γ durch E-äquivalente Substitutionen, so erhält man wieder eine Lösungsbasis. ■

Repräsentationstheorem:

a) Für jede vollständige Menge $cU_E(\Gamma)$ von E-Unifikatoren gilt:

δ löst Γ genau dann, wenn $\delta x =_E \lambda\sigma x$ für alle x in Γ mit $\sigma \in cU_E(\Gamma)$ und einer Substitution λ.

b) Die Lösungsbasen sind die kleinsten Mengen mit dieser Eigenschaft, d.h. wenn man eine Substitution aus einer Lösungsbasis entfernt, ist sie nicht mehr vollständig. ■

Vererbungstheorem:

Sei $cU_E(s_1 = t_1)$ eine vollständige Menge von E-Unifikatoren von $\langle s_1 = t_1 \rangle_E$ und seien $cU_E(\sigma s_2 = \sigma t_2)$ vollständige Mengen von E-Unifikatoren für $\langle \sigma s_2 = \sigma t_2 \rangle_E$ für alle Lösungen $\sigma \in cU_E(s_1 = t_1)$. Dann ist die Menge $\{\tau\sigma: \tau \in cU_E(\sigma s_2 = \sigma t_2), \sigma \in cU_E(s_1 = t_1)\}$ eine vollständige Lösungsmenge für das System aus beiden Gleichungen $\langle s_1 = t_1, s_2 = t_2 \rangle_E$. ■

Wie wir bereits gesehen haben, ist es möglich, daß unendlich viele allgemeinste Unifikatoren existieren (siehe das Assoziativitätsbeispiel von oben). Es kann sogar sein, daß überhaupt keine Menge allgemeinster E-Unifikatoren existiert, die Forderung nach Vollständigkeit und nach Minimalität können unverträglich sein. Betrachten wir dazu ein Beispiel.

Beispiel: Sei L = {append(x, nil) = x, first(append(x,y)) = first(x), first(nil) = nil } eine Theorie für eine Konstante 'nil', ein einstelliges und ein zweistelliges Funktionssymbol 'first' und 'append'.

Dann hat das L-Unifikationsproblem $\langle$ first(x) = nil $\rangle_L$ die vollständige Menge von L-Unifikatoren

$cU_L(\text{first}(x) = \text{nil}) = \{\sigma_n : n \geq 0\}$ mit

$\sigma_0 = \{x \leftarrow \text{nil}\}$

$\sigma_1 = \{x \leftarrow \text{append}(\text{nil}, x_1)\}$

$\sigma_2 = \{x \leftarrow \text{append}(\text{append}(\text{nil}, x_1), x_2)\}$

…

$\sigma_n = \{x \leftarrow \text{append}(\text{append}(\ldots (\text{append}(\text{nil}, x_1), x_2), \ldots , x_n)\}$

…

Aber jeder Unifikator σ_n ist allgemeiner als der Vorgänger σ_{n-1}, man substituiere x_n durch 'nil' und wende das erste Axiom aus L an, dann erhält man den Vorgänger. Man hat hier also eine Kette von immer allgemeiner werdenden Unifikatoren und da sie vollständig ist, gibt es keine allgemeinsten. ■

3.4 Die Unifikationshierarchie

Eine der wichtigsten Aufgaben in der Unifikationstheorie ist es daher, für spezielle Theorien oder gar ganze Klassen von Theorien Existenztheoreme zu finden, also Aussagen der Form:

> Für die Theorie E oder die Klasse $\mathcal{K}$ von Theorien existiert für jedes Unifikationsproblem stets eine Lösungsbasis.

Theorien E, für die gewisse Gleichungssysteme keine Lösungsbasis besitzen, heißen auch vom *Unifikationstyp 0* ($E \in \mathcal{U}_0$). Die Existenztheoreme befassen sich also mit der Frage: $E \notin \mathcal{U}_0$ oder $\mathcal{K} \cap \mathcal{U}_0 = \emptyset$.

Allgemeiner noch ist man darüberhinaus daran interessiert, die Theorien, die nicht Typ 0 sind, danach zu klassifizieren, ob einige Unifikationsprobleme unendliche Lösungsbasen haben (die Theorie E ist vom Typ ∞ oder *infinitär*, $E \in \mathcal{U}_\infty$), oder ob die Lösungsbasen aller Gleichungssysteme endlich sind (E ist vom Typ ω oder *finitär*, $E \in \mathcal{U}_\omega$) bzw. noch besser ob alle einelementig sind (E ist vom Typ 1 oder *unitär*, $E \in \mathcal{U}_1$). Die letzteren sind insbesondere in der Praxis von Bedeutung, da auch bei einer finitären Theorie die Anzahl der allgemeinsten Unifikatoren zwar endlich ist, aber beliebig groß werden kann.

Einen Überblick über untersuchte und gemäß dieser *Unifikationshierarchie* klassifizierte Theorien gibt J. Siekmann [Sie 88], der auch eine umfangreiche Bibliographie zur Unifikationstheorie angibt.

Die Hauptaufgabe in der „praktischen" Unifikationstheorie ist es nun, für spezielle Theorien Unifikationsalgorithmen zu entwickeln. Das sind Algorithmen (oder Verfahren), die als Eingabe ein E-Unifikationsproblem erhalten und eine – möglichst minimale – Menge von E-Unifikatoren bezüglich der speziellen Theorie E zurückliefern.

Als Mindestvoraussetzung für eine Theorie stellt sich damit folgendes

Problem 1: *(Unifizierbarkeitsproblem)*

> *Ist die Unifizierbarkeit eines E-Unifikationsproblems in der gegebenen Theorie entscheidbar?*

Daß dies im allgemeinen nicht der Fall ist, zeigt die bekannte Unentscheidbarkeit von Hilberts zehntem Problem: Gibt es ein Verfahren das die Lösbarkeit einer Polynomgleichung über den ganzen Zahlen (Diophantische Gleichung) entscheidet? Antwort: nein.

Für Implementierungen ist natürlich dann auch die Frage nach der Anzahl der zurückgelieferten Unifikatoren relevant. Ob überhaupt ein Algorithmus existieren kann oder nur ein Aufzählverfahren, folgt unmittelbar aus der Lage der zu untersuchenden Theorie in der Unifikationshierarchie.

Problem 2: *(Hierarchieproblem)*

> *Welchen Unifikationstyp hat die gegebene Theorie?*

Schließlich stellt sich die letzte Frage fast von allein: Existiert überhaupt ein Algorithmus, der minimale Mengen von Unifikatoren berechnet bzw. aufzählt? Hierbei wollen wir noch eine kleine Unterscheidung zwischen finitären und infinitären Theorien treffen.

Ein E-Unifikationsalgorithmus heißt *typkonform*, wenn er eine Menge Ψ von E-Unifikatoren berechnet – oder aufzählt – mit:

- Ψ ist immer eine vollständige Menge von E-Unifikatoren.
- Der Algorithmus terminiert (mit endlichem Ψ), falls eine endliche, vollständige Lösungsmenge existiert.
- Ψ ist eine Lösungsbasis, falls keine endliche, vollständige Menge existiert.

Mit anderen Worten: Ein typkonformer Algorithmus berechnet immer eine endliche, vollständige Lösungsmenge, falls eine solche existiert, oder er zählt eine unendliche, aber minimale auf. Natürlich darf die Theorie dann nicht vom Typ 0 sein. Insbesondere berechnen typkonforme Algorithmen für finitäre Theorien immer endliche, vollständige Mengen. Die Aufzählung einer unendlichen, vollständigen Menge wäre nicht sehr sinnvoll, da die gesamte Lösungsmenge bereits vollständig und aufzählbar ist. Ideal sind natürlich *minimale* Algorithmen, die immer minimale, vollständige Lösungsmengen berechnen oder aufzählen.

Damit ergibt sich

Problem 3: *(typkonformer Algorithmus)*

> *Gibt es für eine gegebene Theorie – die nicht vom Typ 0 ist – einen typkonformen beziehungsweise einen minimalen Algorithmus?*

3.5 Einige Resultate für spezielle Theorien

Wir geben eine tabellarische Übersicht über diejenigen Theorien, die seit Robinson in der Unifikationstheorie näher untersucht wurden.

Es sind dies im wesentlichen die durch folgende Axiome definierten Theorien:

$Ø(f)$	$:= \{ \}$	(leere Theorie; freie Funktion)
$A(f)$	$:= \{ f(x, f(y, z)) = f(f(x, y), z) \}$	(Assoziativität)
$C(f)$	$:= \{ f(x, y) = f(y, x) \}$	(Kommutativität)
$I(f)$	$:= \{ f(x, x) = x) \}$	(Idempotenz)
$D_l(f,g)$	$:= \{ f(g(x, y), z) = g(f(x, z), f(y, z)) \}$	(Links-Distributivität)
$D_r(f,g)$	$:= \{ f(x, g(y, z)) = g(f(x, y), f(x, z)) \}$	(Rechts-Distributivität)
$D(f,g)$	$:= D_l(f,g) \cup D_r(f,g)$	(Distributivität)
$Inv_l(f,i,e)$	$:= \{ f(i(x), x) = x \}$	(Links-Inverse)
$Inv_r(f,i,e)$	$:= \{ f(x, i(x)) = x \}$	(Rechts-Inverse)
$Inv(f,i,e)$	$:= Inv_l(f,i,e) \cup Inv_r(f,i,e)$	(Inverse)
$N_l(f,e)$	$:= \{ f(e, x) = x \}$	(Links-Neutrales)
$N_r(f,e)$	$:= \{ f(x, e) = x \}$	(Rechts-Neutrales)
$N(f,e)$	$:= N_l(f,e) \cup N_r(f,e)$	(Neutrales)
$Iv(f)$	$:= \{ f(f(x)) = x \}$	(Involution)
$E(h,f)$	$:= \{ h(f(x_1,\ldots, x_n)) = f(h(x_1),\ldots, h(x_n)) \}$	(Endomorphismus)
$AE(h,f)$	$:= \{ h(f(x_1,\ldots, x_n)) = f(h(x_n),\ldots, h(x_1)) \}$	(Anti-Endomorphismus)

Durch Kombinationen aus diesen Axiomen lassen sich bereits viele wichtige mathematische Theorien aufbauen: Monoide (A+N), Gruppen (A+N+Inv) und andere.

Spezielle Kombinationen, die in der Unifikationstheorie untersucht wurden, sind zum Beispiel

– $AC(f) := A(f) \cup C(f)$

– $AI(f) := A(f) \cup I(f)$

– $ACI(f) := A(f) \cup C(f) \cup I(f)$

Auch folgende Kombinationen - die sogenannten Minus-Theorien - sind unter anderem für theoretische Untersuchungen zur Unifikationshierarchie von Interesse:

– $Iv(-) \cup AE(-, f_2) = \{ -(-x) = x, f_2(-x, -y) = -f_2(y, x) \}$

– $Iv(-) \cup AE(-, f_1) \cup AE(-, f_2) = \{ -(-x) = x, f_1(-x) = -f_1(x), f_2(-x, -y) = -f_2(y, x) \}$.

Die erstere ist nämlich vom Typ ω, während die zweite vom Typ ∞ ist [Kir 86].

Den Stand der Forschung bezüglich unserer drei Problemstellungen für einige dieser Theorien spiegelt die folgende - unvollständige - Tabelle wider:

Theorie	entscheidbar	Typ	Algorithmus
∅	ja	unitär	minimal
A(f)	ja	infinitär	typkonform
C(f)	ja	finitär	typkonform
I(f)	ja	finitär	typkonform
AC(f)	ja	finitär	minimal
AI(f)	ja	0	?
D(f,g)	?	infinitär	typkonform*
D(f,g)∪A(f)	nein	infinitär	typkonform*
D(f,g)∪AC(f)	nein	infinitär	typkonform*
E(f,g)	ja	unitär	minimal
$H(f,g_1,g_2)$	ja	unitär	minimal
Minus-Theorien	ja	(in)finitär**	typkonform
Abel'sche Gruppen	ja	finitär	minimal
Boole'sche Ringe	ja	unitär***	minimal

* Für nicht lösbare Unifikationsprobleme terminiert der Algorithmus nicht.

** Falls alle Anti-Endomorphismen geradstellig sind, ist die Theorie finitär, falls ein gerad- und ein ungeradstelliger existieren ist sie infinitär.

*** Ohne freie Funktionssymbole, mit freien Funktionsymbolen ist sie mindestens finitär.

? Offenes Problem.

Ausführlichere Tabellen findet man bei [Kir 85] und bei [Sie 88].

3.6 Kombination von Theorien und universelle Unifikation

Im mehr theoretischen Fragen zugewandten Teil der Unifikationstheorie geht es darum, die Gleichheitstheorien nach für die Unifikation relevanten, etwa algebraischen Kriterien zu klassifizieren und die Stellung solcher Klassen im Verhältnis zu den Klassen der Unifikationshierarchie zu untersuchen [BHS 88]. Da dies jedoch gewisse vertiefte Kenntnisse etwa in Universeller Algebra erfordern würde, und weil die meisten Leser an diesen recht

theoretischen Fragestellung vermutlich weniger interessiert sein dürften, wollen wir hier nicht näher darauf eingehen.

Eine andere wichtige Fragestellung ergibt sich, wenn man Unifikationsalgorithmen für verschiedene Gleichheitstheorien hat und diese miteinander kombinieren möchte. Man baut ja im allgemeinen nicht für jede Theorie E einen speziellen Beweiser mit der um diese E-Unifikation erweiterten Resolution, sondern integriert die verschiedenen Algorithmen in einen einzigen Beweiser. Ganz allgemein kann eine solche Kombination von Gleichheitstheorien nicht funktionieren. Die Algorithmen für kommutative Funktionen und für assoziative Funktionen unterscheiden sich grundlegend von den Algorithmen für Funktionen, die gleichzeitig assoziativ und kommutativ sind. Wenn allerdings in den Axiomen der einzelnen Theorien keine gemeinsamen Funktionssymbole vorkommen, dann können die Gleichheits-theorien und die zugehörigen Algorithmen kombiniert werden.

Dabei zeigt sich, daß dieser Fall (bisher) auch nur dann gelöste werden kann, wenn die Einzelverfahren Terme mit beliebigen freien Funktionssymbolen unifizieren können. Dies ist allerdings auch aus anderem Grunde die wohl wichtigste Kombinationsfrage: die Kombination einer Theorie mit der leeren Theorie. Das heißt, kann man einen E-Unifikationsalgorithmus so erweitern, daß er auch Terme mit freien Funktionssymbolen, also Funktionssymbolen, die nicht in der Axiomatisierung der Theorie vorkommen, unifizieren kann? Gerade bei automatischen Beweisern, die auf der Resolution beruhen, ist dies ausschlaggebend für deren Erweiterung auf E-Unifikation. Da die Beweisaufgaben in Klauselform gestellt werden müssen, werden insbesondere die ursprünglichen Formeln skolemisiert, das bedeutet aber, daß freie Funktionen in den meisten Fällen vorhanden sind.

Daß dieses Kombinationsproblem nicht trivial ist, zeigen Resultate, bei denen etwa gezeigt wurde, daß die Unifikation durch die Hinzunahme freier Konstanten bereits unentscheidbar werden kann. Ein relativ einfaches Beispiel mag demonstrieren, daß die Hinzunahme freier Konstanten oder Funktion die Unifikation zumindest verändert bzw. erschwert. Nimmt man etwa die Theorie eines assoziativen und kommutativen Funktionssymbols f mit einem neutralen Element e, aber ohne weitere Funktionen oder Konstanten. Dann ist jede Gleichung trivialerweise lösbar. Die Terme sind nur aus den Symbolen f und e und Variablen aufgebaut und man kann die Terme immer dadurch gleichmachen, daß man alle Variablen durch e substituiert. Beide Terme werden dann gleich zu e. Wenn man jedoch freie Konstanten zuläßt, dann werden die Probleme ungleich komplizierter. Bekannte Verfahren für diese als AC1-Unifikation bekannt gewordene Aufgabe reduzieren das Problem auf das Lösen linearer diophantischer Gleichungen, was bekanntlich keine triviale Aufgabe ist.

Hat man für eine Theorie noch kein spezielles Unifikationsverfahren, so kann man natürlich auch die volle Gleichheitsbehandlung für das Lösen der Unifikationsprobleme verwenden, oder sogenannte *universelle Unifikationsalgorithmen*. Das sind Verfahren, die als Eingabe ein Gleichungssystem *und* die Axiome einer Gleichheitstheorie erhalten, und vollständige Lösungsmengen bezüglich der Eingabetheorie berechnen oder aufzählen. Der Vorteil dieser Verfahren ist, daß sie für jede beliebige Theorie funktionieren. Ihr Haupt-Nachteil ist, daß sie natürlich sehr ineffizient sind.

Die folgenden Rechenregeln definieren ein solches Verfahren [GS 87]:

(1) Entferne jede Gleichung $t = t$ aus Γ.

(2) Ersetze jede Gleichung $f(p_1,\ldots,p_n) = f(q_1,\ldots,q_n)$ durch die Gleichungen $p_1 = q_1,\ldots, p_n = q_n$.

(3) Substituiere für jede Gleichung $x = t$, wenn x nicht unter den Variablen des Terms t ist, alle weiteren Vorkommen von x in Γ durch t. Analog für Gleichungen $x = y$.

(4) Ersetze jede Gleichung $t = x$ durch $x = t$.

(5) Ersetze eine Gleichung $f(p_1,\ldots,p_n) = t$ durch $f(p_1,\ldots,p_n) = f(q_1,\ldots,q_n)$ und $s = t$, wenn $f(q_1,\ldots,q_n) = s$ oder $s = f(q_1,\ldots,q_n)$ Variante eines Axioms der Theorie ist. Anmerkung: Diese Regel darf nicht wieder auf $f(p_1,\ldots,p_n) = f(q_1,\ldots,q_n)$ angewandt werden!

(6) Ersetze $x = f(q_1,\ldots,q_n)$ durch $x = f(v_1,\ldots,v_n)$ und $v_1 = q_1,\ldots, v_n = q_n$ und substituiere alle weiteren Vorkommen von x in Γ durch $f(v_1,\ldots,v_n)$, wenn x unter den Variablen der q_i vorkommt.

Die ersten vier Regeln sind dieselben wie beim Regelsystem für die Robinson-Unifikation, aber die beiden dort angegebenen Nicht-Unifizierbarkeitsregeln gelten jetzt im allgemeinen nicht mehr. Stattdessen haben wir zwei neue Regeln, die die Gleichungen mithilfe der Axiome beziehungsweise durch Einführung neuer Variablen (und Auffalten) in neue Gleichungen transformieren, auf die wieder die anderen Regeln angewandt werden können. Unter der Variante eines Axioms bei Regel (5) versteht man eine Kopie des Axioms in der sämtliche Variablen durch neue bisher nicht vorgekommene Variable substituiert wird.

Mit diesem Regelsystem kann man eine vollständige Lösungsmenge für die Ausgangsgleichungen erzeugen, indem man sie solange anwendet, bis man ein gelöstes System erhält und dieses in eine Substitution transformiert. Dabei muß man garantieren, daß alle Pfade im Suchraum erfaßt werden. Das Regelsystem definiert nämlich in folgendem Sinne einen vollständigen Unifikationsalgorithmus:

Theorem: Sei E eine Theorie und sei Γ ein Gleichungssystem. Wenn Γ unter E lösbar ist, dann existiert zu jeder Lösung δ von Γ eine Folge von Regelanwendungen, die mit einem gelösten System terminiert, so daß δ eine Instanz der zum gelösten System gehörenden Substitution ist. ■

Falls für die Theorie eine Axiomatisierung existiert, deren Gleichungen gerichtet werden können, so daß man ein kanonisches Termersetzungssystem erhält (siehe den Beitrag über Termersetzungssysteme), dann genügt es in Regel (5), nur Varianten $f(q_1,\ldots,q_n) \rightarrow s$ von gerichteten Axiomen zu betrachten. Es gilt auch dann das obige Theorem.

Ein wesentlich bekannteres universelles Verfahren zum Lösen von E-Unifikationsproblemen unter beliebigen Theorien erhält man mit der Paramodulation. Sie wurde eigentlich eingeführt für die allgemeine Behandlung der Gleichheit im Resolutionsverfahren, wenn wir Gleichheitsliterale in den Klauseln haben (vgl. den Abschnitt über allgemeine Gleichheits-

verfahren). Aber natürlich kann man dies insbesondere für den Spezialfall anwenden, daß unsere Klauseln nur Gleichheitsliterale enthalten und wenn wir nur unäre Klauseln bestehend aus jeweils genau einem positiven Gleichheitsliteral, also eine Gleichheitstheorie E haben. Dies bedeutet, man kann Paramodulation für die universelle Unifikation einsetzen, in dem man das zu unifizierende Gleichungssystem, welches ja eine existenzquantifizierte Konjunktion darstellt, negiert (es wird dann eine Klausel mit negativen Gleichheitsliteralen) und zu den Klauseln der Axiomatisierung der Gleichheitstheorie E gibt. Die so erhaltene Klauselmenge kann man dann mittels der Paramodulationsregel bearbeiten. Allerdings erhält man so noch kein universelles Unifikationsverfahren, sondern lediglich ein Testverfahren für die Unifizierbarkeit.

Man kann allerdings auch „konstruktiv“ vorgehen. Wie wir im Kapitel über Deduktion als Berechnung sehen werden, kann man eine Gleichheitstheorie auch als logisches Programm sehen und ein Unifikationsproblem als eine Anfrage an dieses. Dann wendet man die Paramodulationsregel mit der Gleichheitstheorie E auf das Unifikationsproblem selbst an, solange bis man ein Gleichungssystem erhält das mittels syntaktischer Unifikation gelöst werden kann. Man erhält dann einen E-Unifikator des ursprünglichen Systems, in dem die Paramodulationssubstitutionen aufsammelt und mit dem syntaktischen Unifikator dieses letzten Systems kombiniert. Man hat damit die E-Unifikation gänzlich auf die syntaktische Unifikation zurückgeführt – die Paramodulationssubstitutionen sind selbst auch syntaktische Unifikatoren.

Präzisieren wir das daraus resultierende universelle Unifikationsverfahren. Als Eingabe erhalten wir eine Gleichheitstheorie (genauer ihre Axiomatisierung) E und ein Gleichungssystem Γ. Durch wiederholte Anwendung der folgenden Paramodulationsregel transformiert man das System in ein syntaktisch unifizierbares System.

(P) Ersetze in Γ einen beliebig ausgewählten Unterterm $f(t_1,\ldots,t_n)$ durch den Term t und wende die Substitution σ auf das so erhaltene System an; dabei sei σ der syntaktische Unifikator von $f(s_1,\ldots,s_n)$ und $f(t_1,\ldots,t_n)$ für eine Variante eines passend gewählten Axioms $f(s_1,\ldots,s_n) = t$ oder $t = f(s_1,\ldots,s_n)$ der Theorie E.

Die Paramodulationsregel enthält zwei Indeterminismen: Der beliebig zu wählende Unterterm und das passend zu wählende Axiom. Über diese Indeterminismen erhält man mehrere Möglichkeiten das Eingabesystem zu transformieren. Jede Folge von Regelanwendungen, die zu einem syntaktisch unifizierbaren System führt, liefert einen E-Unifikator des Eingabesystems. Natürlich braucht das Verfahren nicht zu terminieren. Aber es definiert ähnlich wie das obige Regelsystem einen vollständigen Unifikationsalgorithmus: wenn Γ unter E lösbar ist, existiert zu jeder Lösung δ eine Folge von Paramodulationsschritten mit Substitutionen $\sigma_1,\ldots,\sigma_m$, die mit einem System Γ' terminiert, welches einen syntaktischen Unifikator σ hat, so daß δ eine Instanz der Komposition der Substitutionen $\sigma_1,\ldots,\sigma_m$, σ ist. Falls man für die Theorie wieder eine Axiomatisierung mit kanonischem Termersetzungssystem hat, kann man sich bei der Paramodulation auf gerichtete Axiome $f(s_1,\ldots,s_n) \rightarrow t$ beschränken. Die so modifizierte Regel heißt auch „Narrowing“-Regel.

3.7 Ein Beispiel: Unifikation in Booleschen Ringen

Wir wollen abschließend noch ein interessantes Beispiel einer Gleichheitstheorie betrachten, nämlich die der Booleschen Ringe. Mit diesen kann man bekanntlich sowohl Mengen als auch Schaltkreise modellieren, und für beides gibt es wichtige Anwendungen. Die Unterstützung von Mengen ist insbesondere bei mathematischen Beweisen von Bedeutung, während die Modellierung von Schaltkreisen die Verifikation von Schaltungen unterstützen kann [BS 87], [MN 87].

Beispiel: Unifikation in freien Booleschen Ringen.

Wir betrachten eine Signatur aus beliebig vielen freien Konstanten und zweistelligen (Infix-) Funktionssymbolen + und ·, sowie Konstanten 1 und 0, für die wir die folgenden Axiome haben (wir kürzen x · y durch xy ab):

$$\begin{aligned} BR := \{\ & xy = yx,\ (xy)z = x(yz),\ xx = x, \\ & x + y = y + x,\ x + (y + z) = (x + y) + z \\ & x(y + z) = xy + yz \\ & x \cdot 1 = 1 \cdot x = x,\ x + 0 = 0 + x = x,\ x + x = 0\ \} \end{aligned}$$

Beide Funktionen sind also kommutativ, assoziativ und sind über die Distributivität verbunden, · ist idempotent und + ist nilpotent, 1 und 0 sind die neutralen Elemente bezüglich · und +. Dabei entspricht · dem Schnitt bzw. 'und' und + der symmetrischen Differenz bzw. 'exklusives oder'. Die üblichen Mengen- bzw. Schaltalgebraoperationen erhält man durch folgende Übersetzung:

$$\begin{aligned} & xy = x \cap y,\ x + y = (x \cup y) \setminus (x \cap y) = (x \setminus y) \cup (y \setminus x), \\ & x \cup y = x + y + xy,\ x \setminus y = x + xy \\ & xy + y = 0 \Leftrightarrow xy = y \Leftrightarrow y \subseteq x \end{aligned}$$

Die BR-Unifikation hat folgende einfach nachzurechnende Eigenschaft: Die Substitution $\sigma = \{x \leftarrow q + x'(1 + p)\}$ ist allgemeinster Unifikator von $\langle\ px + q = 0\ \rangle_{BR}$ genau dann, wenn gilt $pq + q =_{BR} 0$.

Damit erhält man einen rekursiven BR-Unifikationalgorithmus für Probleme $\langle\ t = 0\ \rangle_{BR}$ (es ist klar, daß alle BR-Unifikationprobleme in diese Form gebracht werden können):

1 Isoliere eine Variable x in t, d.h. transformiere $\langle\ t = 0\ \rangle_{BR}$ in die Form $\langle\ px + q = 0\ \rangle_{BR}$ (so daß x nicht in den Termen p und q vorkommt)

2 Löse das nun kleinere Problem $\langle\ pq + q = 0\ \rangle_{BR}$, d.h. berechne seinen Unifikator σ

3 $\tau := \sigma\lambda$ mit $\lambda := \{x \leftarrow q + x'(1 + p)\}$ ist allgemeinster BR-Unifikator von $\langle\ t =_{BR} 0\ \rangle$

Der Algorithmus terminiert, da bei jeder Rekursion Variablen eliminiert werden, und er berechnet den allgemeinsten BR-Unifikator. Die BR-Unifikation ist also unitär.

3.8 Schlußbemerkungen

Der hier dargestellte Stand der Unifikationstheorie zeigt die für praktische Anwendungen wichtigsten Grundlagen des Gebietes, wenn wir auch die Details der Forschung großzügig übergangen haben. Dies liegt u.a. daran, daß das Gebiet sehr stark mathematisiert ist und die Resultate oft sehr spezielle Grundlagen etwa der Universellen Algebra benötigen, so daß ihre Darstellung den Rahmen dieses Kapitels gesprengt hätte. Wir wollen hier nur anmerken, daß man bei diesen Untersuchungen die Unifikation auch als Gleichungslösen in speziellen universellen Algebren, den freien Algebren der Gleichheitstheorie, betrachtet.

Einen Zweig der Theorie haben wir hier gar nicht angeschnitten, nämlich die sortierte Unifikation. Hierbei erlaubt man zusätzlich eine Typisierung der Variablen und der Funktionssymbole, die die Termbildung und die zulässigen Substitutionen stark einschränken. Wenn man darüberhinaus die Sorten oder Typen noch hierarchisch durch eine Untersortenrelation anordnet, so kann man diese Hierachie bei der Unifikation verstärkt ausnutzen, indem man z.B. Variablen einer Sorte nur durch Terme einer kleineren Sorte ersetzen darf. Mehr dazu, auch über Anwendungen der sortierten Unifikation, findet man im Abschnitt über Sorten und Typen im Kapitel „Deduktion als Berechnung“.

Natürlich kann man sich auch von der Beschränkung auf Terme erster Stufe lösen und sich fragen, was passiert, wenn man Funktionsvariable zu läßt. Dieser Fragestellung wird im Kapitel über Unifikation für Logik höherer Stufe ausführlicher nachgegangen.

Die nachfolgende Literaturliste enthält nur die hier zitierten Referenzen. Eine ausführliche Bibliographie zur Unifikationstheorie findet man etwa in [Sie 88] (vgl. auch [Kni 88]; dort findet man auch einen Überblick über effiziente Algorthmen für die syntaktische Unifikation). Das Buch [Kir 90] enthält sämtliche Artikel des „Special Issue on Unification“ des Journal of Symbolic Computation 1988 und gibt eine relativ umfassenden Überblick über den Stand der Forschung bis Anfang 1988. Einen neueren Übersichtsartikel findet man in [JK 91], wo insbesondere auch die Sicht des Gleichungslösens in speziellen Algebren im Mittelpunkt steht.

Neuere Arbeiten zur Unifikationstheorie erweitern die Fragestellung hin zum Lösen von Gleichungen und negierten Gleichungen („Disunification“) bzw. führen abstrakter zum Lösen symbolischer „Constraints“: Man kann Unifikationsprobleme auch als Constraints über dem Quotienten $\mathbb{T}/=_E$ der Termalgebra nach der durch die Gleichheitstheorie induzierten Äquivalenzrelation sehen. Diese generellere Sicht führte dann auch konsequent zur Frage, inwiefern man die Unifikation in der Resolutionsregel durch andere Constraint-Solving-Methoden ersetzen kann (vgl. dazu auch den Abschnitt über Logisches Programmieren mit Constraints im Kapitel „Deduktion als Berechnung“).

Literatur

BHS 88 H.-J. Bürckert, A. Herold, M. Schmidt-Schauß: *Equational Theories, Unification and Decidability.* J. of Symbolic Computation, Special Issue on Unification, 1988

BS 87 W. Büttner, H. Simonis: *Embedding Boolean Expressions into Logic*

Programming. J. of Symbolic Computation, 1987

GS 87 J. Gallier, W.Snyder: *A General Complete E-Unification Procedure.* Proc. of Conf. Rewriting Techniques and Applications, Springer LNCS 256, 1987

Her 30 J. Herbrand: *Sur la Théorie de la Démonstration*, Dissertation 1930, in *Logical Writings* (W. Goldfarb, Hrsg.), Cambridge, 1971

JK 91 J.-P. Joaunnaud, C. Kirchner: *Solving Equations in Abstract Algebras: A Ruel-based Survey of Unification.* In J.-L. Lassez, G. Plotkin (Hrsg.): *Essays in Honour of Alan Robinson*. MIT, 1991

Kir 85 C. Kirchner: *Méthodes et Outils de Conception Systematique d'Algorithmes d'Unification dans les Theories Equationelles,* These d'Etat, Université de Nancy, 1985

Kir 90 C. Kirchner (Hrsg.): *Unification*. Academic Press, 1990

MN 87 U. Martin, T. Nipkow: *Unification in Boolean Rings,* J. of Automated Reasoning, 1987

Plo 72 G. Plotkin: *Building in Equational Theories,* Machine Intelligence 7, 1972

Rob 65 J.A. Robinson: *A Machine Oriented Logic Based on the Resolution Principle,* JACM 12, 1965

Sie 88 J. Siekmann: *Unification Theory,* J. of Symbolic Computation, Special Issue on Unification, 1988