Logik für Informatik
Sommersemester 2003
Mathias Kegelmann

Technische Universität Darmstadt
Fachbereich Mathematik
1. Juli 2003

XI. Resolution und Unifikation

Diese Woche wollen wir uns mit einigen Details des Resolutionsverfahrens beschäftigen, die noch offen geblieben sind. Beginnen wir aber damit, die einzelnen Schritte zu wiederholen:

Resolutionsverfahren

1. Wir wollen $\vDash A$, also die Allgemeingültigkeit einer Formel A nachweisen.
2. Stattdessen zeigen wir, dass $\neg A$ unerfüllbar ist.
3. Zunächst bringen wir dazu $\neg A$ in Pränex-Form.
4. Dann skolemisieren wir das Resultat.
5. Schließlich bringen wir die nun quantorenfreie Formel in konjunktive Normalform und kodieren das Ergebnis als Klauselmenge M; eine Klausel $K \in M$ entspricht dabei einem Disjunktionsterm und M der Konjunktion aller Klauseln.
6. Jetzt iteriert man folgenden **Resolutionsschritt**:
   (a) Man wählt zwei Klauseln $K, K' \in M$
   (b) und Substitutionen σ, σ', sodass für ein Literal A gilt $A \in K[\sigma]$ und $\neg A \in K'[\sigma']$.
   (c) Wir bilden daraus die *Resolvente*[1]
   $$R := \big(K[\sigma] \setminus \{A\}\big) \cup \big(K'[\sigma'] \setminus \{\neg A\}\big)$$
   und fügen sie zur Klauselmenge hinzu, d.h. die neue Klauselmenge ist
   $$M' := M \cup \{R\}.$$
7. Mit diesem Resolutionsschritt fügt man sukzessive immer neue Klauseln zur Klauselmenge hinzu, bis man die leere Klausel $\emptyset$ erhält. Sobald man das geschafft hat, ist die Allgemeingültigkeit von A bewiesen.

1 Korrektheit und Vollständigkeit der prädikatenlogischen Resolution

1.1 Korrektheit

Es stellt sich natürlich die Frage, ob dieses Verfahren korrekt ist. Um das präzise formulieren zu können, müssen wir zunächst definieren, wann wir eine Klauselmenge unerfüllbar nennen. Die Begriffe sind gerade so definiert, wie man es aufgrund der Erfüllbarkeit für Formeln erwarten würde, wenn man bedenkt, dass Klauselmengen für implizit universell quantifizierte Formeln in konjunktiver Normalform stehen.

Definition 11.1. Eine Klausel K heißt **erfüllbar**, wenn es ein Modell $\mathcal{M}$ gibt, sodass es für alle Variablenbelegungen ρ *mindestens eine* Formel $A \in K$ mit $[\![A]\!]\rho = \mathsf{w}$ gibt. Wir nennen dieses $\mathcal{M}$ auch ein **Modell** für K.

Eine Klauselmenge M nennen wir **erfüllbar**, wenn es ein $\mathcal{M}$ gibt, das ein Modell für *alle* Klauseln $K \in M$ ist. Wieder nennen wir $\mathcal{M}$ ein **Modell** für M.

Um einzusehen, dass diese Definition tatsächlich die Erfüllbarkeit des universellen Abschlusses der einer Klauselmenge M entsprechenden Formel fasst, machen wir folgende Beobachtung, die auch später noch nützlich sein wird.

Proposition 11.2. *Seien* $A_1, \ldots, A_n \in F_\Sigma(X)$ quantorenfreie *Formeln, die nur die Variablen* $x_1, \ldots, x_m$ *enthalten. Dann sind* $\forall x_1 \cdots \forall x_m . A_1 \wedge \cdots \wedge A_n$ *und* $(\forall x_1 \cdots \forall x_m . A_1) \wedge \cdots \wedge (\forall x_1 \cdots \forall x_m . A_n)$ *äquivalent.*

Mit den Schritten (1)–(5) des Resolutionsverfahrens haben wir uns schon letzte Woche beschäftigt; wir wissen bereits, dass A genau dann eine Tautologie ist, wenn die zugehörige Klauselmenge M unerfüllbar ist. Wir müssen also nur noch überlegen, dass der Resolutionsschritt (6) Erfüllbarkeit erhält. Dazu befassen wir uns zunächst mit der Substitution, also dem Teilschritt (6b).

Lemma 11.3. *Sei M eine Klauselmenge, $\mathcal{M}$ ein Modell für M, $K \in M$ eine Klausel und σ eine Substitution. Dann ist $\mathcal{M}$ auch ein Modell für $K[\sigma]$.*

[1] **Achtung:** Im Skript 10 war die Resolvente als $\big(K[\sigma] \cup K'[\sigma']\big) \setminus \{A, \neg A\}$ angegeben. Das ist fast immer das selbe wie $\big(K[\sigma] \setminus \{A\}\big) \cup \big(K'[\sigma'] \setminus \{\neg A\}\big)$, aber wenn sie sich unterscheiden, dann ist die Klausel aus Skript 10 **falsch**!

Beweis. Das folgt praktisch unmittelbar aus der Definition; wenn man es genau nimmt, verwendet man aber das Substitutionslemma 7.4, wenigstens in der Form, dass $[\![A[\sigma]]\!]\rho = [\![A]\!]\rho'$ für eine andere Variablenbelegung ρ' ist. □

Damit kann man jetzt sofort die prädikatenlogische Verallgemeinerung von Satz 5.8 und Korollar 5.9 zeigen, dass jedes Modell für M auch ein Modell für die Resolvente R aus (6c) ist. Da mit jeder erfüllbaren Klauselmenge trivialerweise auch jede Teilmenge erfüllbar ist, sehen wir dass in Schritt (6) die Klauselmenge M genau dann erfüllbar ist, wenn es die neue Klauselmenge M' ist. Also erhält die Durchführung eines Resolutionsschritts Erfüllbarkeit. Da die leere Klausel unter keiner Belegung gilt, also jede Klauselmenge, die $\emptyset$ enthält, unerfüllbar ist, erhalten wir somit folgendes Korollar

Korollar 11.4. *Sei M eine Klauselmenge, für die wir in endlich vielen Resolutionsschritten die leere Klausel $\emptyset$ erzeugen können. Dann ist M unerfüllbar.*

Dies zeigt, dass Resolution korrekt ist: Kann man die leere Klausel aus der zu $\neg A$ gehörenden Klauselmenge erzeugen, so ist A eine Tautologie.

1.2 Vollständigkeit

Die nächste Frage ist nun, ob Resolution auch vollständig ist. Wie immer ist Vollständigkeit schwieriger zu zeigen als Korrektheit. Aber verblüffenderweise ist es für Resolution *ohne Gleichheit* verhältnismäßig einfach.

Satz 11.5 (Herbrand). *Sei $A \in F$ eine* quantorenfreie *Formel, in der* = *nicht vorkommt. Dann ist der universelle Abschluss von A genau dann erfüllbar, wenn die Menge aller Substitutionsinstanzen*

$$S_A := \{A[\sigma] \mid \sigma\colon V \to T_\Sigma(V)\}$$

rein aussagenlogisch[2] *konsistent ist.*

Beweis. Zunächst bemerken wir, dass aufgrund des aussagenlogischen Vollständigkeitssatzes für unendliche Formelmengen (Korollar 4.3) die Menge S_A genau dann konsistent ist, wenn sie erfüllbar ist.

Hat der universelle Abschluss von A ein Modell, so gibt uns dies wegen Lemma 11.3 sofort eine aussagenlogische Belegung, die alle Formeln in S_A wahr macht.

Für die Rückrichtung nehmen wir nun an, S_A sei rein aussagenlogisch konsistent. Dann gibt es eine aussagenlogische Belegung $\rho\colon \{R(t_1, \ldots, t_{\alpha(R)}) \mid R \in \mathcal{R}, t_1, \ldots, t_{\alpha(R)} \in T_\Sigma(V)\} \to \{\mathsf{w}, \mathsf{f}\}$, die alle Formeln $B \in S_A$ wahr macht. Aus ρ konstruieren wir uns nun ein Modell wie folgt. Als Grundmenge verwenden wir die Termalgebra $T_\Sigma(V)$ und wir interpretieren die Operationen aus Ω wie in der Termalgebra üblich. Die Relationssymbole $R \in \mathcal{R}$ interpretieren wir in $T_\Sigma(V)$ wie folgt:

$$R^{T_\Sigma(V)} := \left\{(t_1, \ldots, t_{\alpha(R)}) \in \big(T_\Sigma(V)\big)^{\alpha(R)} \,\middle|\, \rho\big(R(t_1, \ldots, t_{\alpha(R)})\big) = \mathsf{w}\right\}$$

Für jede quantorenfreie Formel $B \in F_\Sigma$ und jede Variablenbelegung $\sigma\colon V \to T_\Sigma(V)$ gilt nun in diesem Modell $[\![B]\!]\sigma = [\![B[\sigma]]\!]\rho$, wobei der zweite Ausdruck die *aussagenlogische* Semantik bezeichnet, wie man sofort per Induktion über den Aufbau von B zeigt. Insbesondere gilt also für jede solche Variablenbelegung σ die Beziehung $[\![A]\!]\sigma = [\![A[\sigma]]\!]\rho = \mathsf{w}$, da $A[\sigma] \in S_A$ und ρ alle Elemente aus S_A wahr macht. Somit gilt in unserem Termmodell der universelle Abschluss von A. □

Solche Modelle, wie wir sie gerade im Satz gesehen haben, deren unterliegende Algebra die Termalgebra ist, heißen auch **Herbrand-Modelle**.

Aus der Vollständigkeit der Resolution für die Aussagenlogik und dem Satz von Herbrand erhalten wir nun unmittelbar die Vollständigkeit der Resolution für Prädikatenlogik ohne Gleichheit:

Korollar 11.6. *Sei M eine Klauselmenge, in der das Symbol für Gleichheit* = *nicht vorkommt. Dann ist M genau dann unerfüllbar, wenn wir in endlich vielen Resolutionsschritten die leere Klausel $\emptyset$ erzeugen können.*

Beweis. Kann man mit Resolutionsschritten aus einer Klauselmenge M die leere Klausel erzeugen, so ist nach Korollar 11.4 M unerfüllbar.

Sei umgekehrt M unerfüllbar. Diese Klauselmenge kodiert eine quantorenfreie Formel A in konjunktiver Normalform, deren universeller Abschluss somit ebenfalls unerfüllbar ist. Nach Satz 11.5 ist die Menge der Substitutionsinstanzen S_A somit rein aussagenlogisch inkonsistent und wegen dem Kompaktheitssatz 4.2 gilt dies auch für eine endliche Teilmenge $S \subseteq S_A$. Wegen der Vollständigkeit der Resolution für Aussagenlogik kann man deshalb mit endlich vielen aussagenlogischen Resolutionsschritten die leere Klausel aus S herleiten. Dies kann man nun mit der prädikatenlogischen Resolution dadurch nachspielen, dass man nacheinander die notwendigen Substitutionen bildet und die so entstehenden Klauseln dann resolviert. □

[2] D.h. wenn wir die atomaren Formeln $R(t_1, \ldots, t_{\alpha(R)})$, für $R \in \mathcal{R}$ und $t_1, \ldots, t_{\alpha(R)} \in T_\Sigma(V)$ als *atomare Aussagen* (statt den gewöhnlichen $p_0, p_1, \ldots$) auffassen.

2 Resolution und Gleichheit

Die Gleichheitsrelation = hat gewisse Eigenschaften, die wir im Kalkül des Natürlichen Schließens durch die beiden Regeln (R) und (Eq) gefasst haben. Bei der Resolution müssen wir uns aber anders behelfen.

2.1 Gleichheitsaxiome als Klauseln

Das einfachste Vorgehen ist es, die Eigenschaften der Gleichheit durch Axiome auszudrücken, die den Regeln (R), (S) und (T) aus Abschnitt VIII.3 entsprechen. Von der Regel (Eq) brauchen wir nur den Spezialfall, der es uns erlaubt, bei den Argumenten für die Operations- bzw. die Relationssymbole gleiches für gleiches zu ersetzen. Diese Eigenschaften kann man direkt durch Klauseln ausdrücken:

- $\{x = x\}$
- $\{\neg x = y, y = x\}$
- $\{\neg x = y, \neg y = z, x = z\}$
- $\big\{\neg x_1 = y_1, \ldots, \neg x_{\alpha(f)} = y_{\alpha(f)}, f(x_1, \ldots, x_{\alpha(f)}) = f(y_1, \ldots, y_{\alpha(f)})\big\}$, für alle $f \in \Omega$
- $\big\{\neg x_1 = y_1, \ldots, \neg x_{\alpha(R)} = y_{\alpha(R)}, \neg R(x_1, \ldots, x_{\alpha(R)}), R(y_1, \ldots, y_{\alpha(R)})\big\}$, für alle $R \in \mathcal{R}$

Die Menge aller dieser Klauseln bezeichnen wir mit $\boldsymbol{Ax_=}$. Sind sowohl Ω als auch $\mathcal{R}$ endlich, so ist auch $Ax_=$ endlich.

Beispiel 11.7. Wir betrachten die Formel $\big(\forall x \exists y.\ R(x, y)\big) \wedge (\forall a \forall b.\ a = b) \to \exists c \forall d.\ R(d, c)$, die wir in der Übung bereits mit Natürlichem Schließen als Tautologie nachgewiesen haben. Durch Umformen erhalten wir

$$\begin{aligned} & \neg\Big(\big(\forall x \exists y.\ R(x, y)\big) \wedge (\forall a \forall b.\ a = b) \to \exists c \forall d.\ R(d, c)\Big) \\ \cong\quad & \big(\forall x \exists y.\ R(x, y)\big) \wedge (\forall a \forall b.\ a = b) \wedge \forall c \exists d.\ \neg R(d, c) \end{aligned}$$

und somit die folgenden Klauseln:[3]

$$\begin{aligned} && K_1 &= \big\{R(x, f(x))\big\} \\ && K_2 &= \{a = b\} \\ && K_3 &= \big\{\neg R(g(c), c)\big\} \\ && K_4 &= \big\{\neg x_1 = y_1, \neg x_2 = y_2, \neg R(x_1, x_2), R(y_1, y_2)\big\} \\ K_1 \text{ und } K_4[x/x_1, f(x)/x_2] : && K_5 &= \big\{\neg x = y_1, \neg f(x) = y_2, R(y_1, y_2)\big\} \\ K_3 \text{ und } K_5[g(c)/y_1, c/y_2] : && K_6 &= \big\{\neg x = g(c), \neg f(x) = c\big\} \\ K_2[x/a, g(c)/b] \text{ und } K_6 : && K_7 &= \big\{\neg f(x) = c\big\} \\ K_2[f(x)/a, c/b] \text{ und } K_6 : && K_8 &= \emptyset \end{aligned}$$

2.2 Korrektheit und Vollständigkeit

Das Hinzufügen der oben aufgeführten Klauseln, die die Gleichheit axiomatisieren, ist sicherlich korrekt, da die entsprechenden Formeln in allen Modellen gelten. Also stellt sich nur noch die Frage nach der Vollständigkeit der Resolution mit Gleichheit. Leider vertragen sich Herbrand-Modelle und Gleichheit nicht, da man die Gleichheit üblicherweise nicht in einem Termmodell interpretieren kann.

Satz 11.8. *Eine Klauselmenge M ist genau dann unerfüllbar, wenn wir in endlich vielen Resolutionsschritten aus $M \cup Ax_=$ die leere Klausel erzeugen können.*

Beweis. Da wir uns die Korrektheit bereits überlegt haben, müssen wir nur zeigen, dass man die Unerfüllbarkeit einer Klauselmenge M per Resolution nachweisen kann.

Sei deshalb M eine Klauselmenge, sodass man aus $M \cup Ax_=$ durch Resolution die leere Klausel nicht erzeugen kann. Wir ersetzen nun in $M \cup Ax_=$ systematisch die Gleichheit = durch ein neues zweistelliges Relationssymbol $\doteq$ und wir nennen die dadurch entstehende Klauselmenge N. Dann kann man aus N ebenfalls die leere Klausel nicht herleiten, und da N die Gleichheit nicht enthält, hat N ein Modell $\mathcal{M}$.

In diesem Modell wird $\doteq$ nicht durch die Gleichheit sondern durch eine zweistellige Relation interpretiert, die aber den definierenden Eigenschaften von = genügt, da N ja auch die Klauseln $Ax_{\doteq}$ enthält, die man bekommt, wenn man in $Ax_=$ das Symbol = durch $\doteq$ ersetzt. Diese Gültigkeit der Klauseln $Ax_{\doteq}$ in $\mathcal{M}$ impliziert sofort, dass $\doteq$ eine Kongruenzrelation bzgl. der Operationen $f^{\mathcal{M}}$, für $f \in \Omega$, ist. Deshalb betrachten wir das Modell, das aus der Quotientenalgebra $\mathcal{M}/\doteq$ und den Relationen

$$R^{\mathcal{M}/\doteq} := \Big\{\big([x_1], \ldots, [x_{\alpha(R)}]\big) \in (\mathcal{M}/\doteq)^{\alpha(R)} \;\Big|\; (x_1, \ldots, x_{\alpha(R)}) \in R^{\mathcal{M}}\Big\},\ R \in \mathcal{R},$$

besteht, wobei die Klauseln aus $Ax_{\doteq}$ garantieren, dass $\big([x_1], \ldots, [x_{\alpha(R)}]\big) \in R^{\mathcal{M}/\doteq}$ genau dann gilt, wenn $(x_1, \ldots, x_{\alpha(R)}) \in R^{\mathcal{M}}$. Da $\mathcal{M}$ ein Modell für N ist, ist somit $\mathcal{M}/\doteq$ ein Modell für M, was zeigt, dass M erfüllbar ist. □

[3] Eigentlich müsste die dritte Klausel, wenn man das Bringen in Pränex-Form und das anschließende Skolemisieren explizit ausführt, $\{\neg R(g(x, c), c)\}$ sein. Die hier angegebene Klauselmenge ist aber äquivalent und erspart etwas Schreibarbeit.

3 Unifikation

Wenn man beim Resolutionsverfahren zwei Klauseln K und K' resolvieren will, stellt sich die Frage, wie man geeignete Substitutionen σ und σ' berechnen kann, sodass $K[\sigma]$ und $K'[\sigma']$ eine Formel bis aus Negation gemeinsam haben. Die Antwort auf diese Frage ist der Unifikations-Algorithmus.

3.1 Unifikation zweier Literale

Wir werden das Problem gleich darauf zurückführen *eine* Substitution zu finden, die zwei Terme gleich macht. Betrachten wir das Beispiel $P(x,x), P(y,z+z)$, so sieht man sehr schnell, dass die Substitution $[z+z/x, z+z/y]$ das erreicht. Wie hätten wir diese Substitution aber systematisch finden können? Hier ist eine erste Skizze des **Unifikations-Algorithmus**:

- Wir vergleichen die beiden Terme von links nach rechts — oder noch besser die zugehörigen Syntaxbäume von der Wurzel her zu den Blättern.
- Sind beide gleich, so sind wir fertig.
- Unterscheiden sich die Terme an einem Operations- oder Relationssymbol, so kann man sie nicht durch Substitution gleich machen, d.h. wir können mit dem Ergebnis "nicht unifizierbar" abbrechen.
- Der einzige andere mögliche Unterschied ist, dass ein Term eine Variable x enthält, wo der andere einen beliebigen Term t hat. Dann wenden wir die Substitution $[t/x]$ auf beide Terme an und beginnen wieder von vorne.

Beispiel 11.9. In dem Beispiel $P(x,x), P(y,z+z)$ finden wir so zuerst die Substitution $[y/x]$ und wir erhalten somit das neue Paar $P(y,y), P(y,z+z)$. Dort bekommen wir die Substitution $[z+z/y]$, die beide Terme zu $P(z+z,z+z)$ macht. Insgesamt ist die Substitution $[z+z/x, z+z/y]$.

Betrachten wir nun ein problematisches Paar von Termen: $P(a, f(a)), P(x,x)$. Man überzeugt sich leicht davon, dass diese beiden Terme für keine Substitution gleich werden. Wie hätten wir das aber bei der Durchführung des Unifikations-Algorithmus feststellen können? Zunächst finden wir die Substitution $[x/a]$ und somit die beiden neuen Terme $P(x, f(x)), P(x,x)$. Für diese erhalten wir die Substitution $[f(x)/x]$, was problematisch ist, da x in dem Term $f(x)$, der für x ersetzt werden soll, selbst wieder vorkommt, was dazu führt, dass die beiden Terme nach der Substitution an genau der selben Stelle immer noch nicht gleich sind.

Man muss deshalb den **Unifikations-Algorithmus** wie folgt abändern:

- Kommt die der Substitution $[t/x]$, die in einem Iterationsschritt ausgerechnet wird, die Variable x in t vor, so sind die beiden Terme nicht unifizierbar und man kann abbrechen.

3.2 Berechnung von Substitutionen für Resolution

Wie können Unifikation nun für Resolution verwenden? Im Schritt (6) nehmen Klauseln K und K', sodass K eine Formel der Form $L = R(\ldots)$ und K' eine der Form $\neg R(\ldots)$ enthält, wobei wir den Rest ohne $\neg$ mit L' bezeichnen. Im Prinzip könnten wir jetzt versuchen L und L' zu unifizieren. Das wäre aber zu restriktiv, da wir bei der Resolution ja für L und L' *verschiedene* Substitutionen verwenden dürfen. Das können wir aber dadurch erreichen, dass wir die Variablen in L' so umbenennen, dass sie von denen aus L disjunkt sind; ein Schritt, der zusätzlich auch durch Proposition 11.2 gerechtfertigt ist. Auf diese beiden Formeln wenden wir nun Unifikation an und erhalten als Ergebnis die gewünschten Substitutionen.

Beispiel 11.10. Angenommen wir haben die Klauseln $\{P(a, a+b\cdot b)\}$ und $\{\neg P(b, a\cdot a+b), Q(a)\}$. Nach Umbenennung der Variablen erhalten wir das Unifikationsproblem $P(a, a+b\cdot b), P(y, x\cdot x+y)$. Wir bekommen die Substitution $[y/a]$ und das neue Paar $P(y, y+b\cdot b), P(y, x\cdot x+y)$, daraus $[x\cdot x/y]$ und das Paar $P(x\cdot x, x\cdot x+b\cdot b), P(x\cdot x, x\cdot x+x\cdot x)$, und somit schließlich $[x/b]$ was beide Terme zu $P(x\cdot x, x\cdot x+x\cdot x)$ macht. Die Substitution dafür ist $[x\cdot x/a, x/b, x\cdot x/y]$ und für die ursprünglichen Klauseln erhält man die beiden Substitutionen $[x\cdot x/a, x/b]$ und $[x/a, x\cdot x/b]$. Also ist das Resultat der Resolution die Resolvente $\{Q(x)\}$.

3.3 Allgemeinster Unifikator

Die Substitution, die der Unifikations-Algorithmus ausrechnet, hat eine weitere wichtige Eigenschaft: Sind t und t' Terme, sowie σ eine Substitution mit $t[\sigma] = t'[\sigma]$. Dann findet Unifikation ebenfalls eine Substitution σ' mit $t[\sigma'] = t'[\sigma']$. Diese hat ferner die Eigenschaft dass σ spezieller in folgendem Sinn ist: Es gibt eine Substitution τ, sodass $t[\sigma] = t[\sigma'][\tau]$ und $t'[\sigma] = t'[\sigma'][\tau]$. Das heißt, dass der Unifikations-Algorithmus eine möglichst allgemeine Substitution ausrechnet.

Diese Eigenschaft garantiert (fast), dass man beim Resolutionsverfahren die Substitutionen verwenden kann, die durch Unifikation ausgerechnet werden: Alles was man nach beliebigen Substitutionen resolvieren kann, erhält man auch durch Unifikation und Weitersubstituieren. Insbesondere erhält man die leere Klausel durch Resolution mit durch Unifikation berechneten Substitutionen, wann immer man sie durch beliebige Substitutionen erhält.

3.4 Resolution mehrerer Literale

Das einzige Problem ist, dass durch eine speziellere Substitution eventuell mehrere Literale einer Klausel gleich werden und damit wegfallen können. Deshalb muss man im Allgemeinen entweder Unifikatoren für mehr als zwei Terme berechnen oder auch außer Resolution noch einen weiteren Schritt erlauben, der Spezialisierungen, also Substitutionsinstanzen, von bereits hergeleiteten Klauseln als neue Klauseln erzeugt.

Mehrere Terme kann man z.B. durch eine Verallgemeinerung des Unifikations-Algorithmus unifizieren, man kann aber auch erst t_1 und t_2 zu t' und dann t' mit t_3 usw. unifizieren.