Logik
Vorlesung 12: Unifikation und Resolution

Andreas Maletti

23. Januar 2015

Überblick

Inhalt

Vorlesungsziele

heutige Vorlesung

1. Unifikationsalgorithmus
2. Anwendungen Logik
3. prädikatenlogische Resolution

Bitte Fragen direkt stellen!

Überblick

Organisation

Organisation

Evaluierung

- externer Dienstleister (Uni Bonn)
- läuft seit Montag (19.01.2015)
- bitte nehmen Sie teil
  (und geigen dem Vorlesenden die Meinung)

Übungsserien

- 7. Übungsserie mit Inhalten aus der gesamten Vorlesung
- wir werden Ihre besten 6 Serien für die Vorleistung bewerten
  (Möglichkeit einen Ausrutscher oder Krankheit auszugleichen)

Organisation

Prüfung

am 17.02.2015 um 13:00 Uhr im **Hs. 3**

- 1 DIN-A4-Blatt mit Notizen als Hilfsmittel zugelassen (beliebig beschrieben oder bedruckt)
- keine weiteren Hilfsmittel zugelassen
- Abmeldung noch bis 25. Januar möglich
- Bitte: melden Sie sich per TOOL ab,
  wenn Sie nicht mitschreiben möchten (oder wenn Sie aufgrund fehlender Vorleistung nicht teilnehmen können)

Tutorium

am 6.02.2015 um 11 Uhr im **Hs. 5**

Wiederholung: Unifikation

Prädikatenlogik — Unifikation

Definition (Unifikator)

Sei $M = \{L_1, \dots, L_n\}$ eine endliche Menge von (prädikatenlogischen) Literalen.

- Ein Substitution $\text{subst} \colon V \to \mathcal{T}$ heißt Unifikator für M gdw.

$$\text{subst}(L_1) = \cdots = \text{subst}(L_n)$$

(Unifikator macht alle Literale gleich)

- Unifikator $\text{subst} \colon V \to \mathcal{T}$ für M ist der allgemeinste Unifikator für M gdw. für jeden Unifikator $\text{subst}' \colon V' \to \mathcal{T}$ für M eine Substitution $\text{subst}'' \colon V'' \to \mathcal{T}$ existiert, so dass $\text{subst}' = \text{subst}\,\text{subst}''$

(jeden Unif. erhält man per Substitution aus allg. Unif.)

Prädikatenlogik — Unifikation

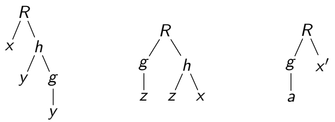
Beispiel

- drei Literale $R(x, h(y, g(y)))$, $R(g(z), h(z, x))$ und $R(g(a), x')$
- Unifikator

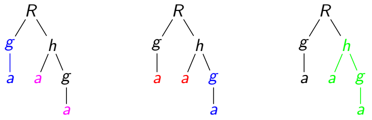
$$[x \mapsto g(a),\ y \mapsto a,\ z \mapsto a,\ x' \mapsto h(a, g(a))]$$

- ist sogar allgemeinster Unifikator

Darstellung als Syntaxbaum:



Nach Substitution:



Prädikatenlogik — Unifikation

Aufgabe

Welche der folgenden Literale sind unifizierbar?

- $P(x, y)$ und $P(y, x)$ ✓
  (Unifikator $[x \mapsto y]$)
- $P(f(x), g(y))$ und $P(y, x)$ ✗
  (schaukelt sich auf)
- $P(h(x, y), y)$ und $P(z, z)$ ✗
  (schaukelt sich auch auf)
- $P(f(z), f(x))$ und $P(y, y)$ ✓
  (Unifikator $[y \mapsto f(z), x \mapsto z]$)
- $P(x, g(y))$ und $P(z, x)$ und $P(y, x)$ ✗
  (schaukelt sich auf)

Unifikationsalgorithmus

Prädikatenlogik — Unifikationsalgorithmus

Algorithmus

Eingabe: endliche Menge M von Literalen

1. setze subst auf $\emptyset$
2. solange $|\{\text{subst}(L) \mid L \in M\}| > 1$
   1. finde erste Position, an der sich subst(L_1) und subst(L_2) unterscheiden für Literale $L_1, L_2 \in M$
   2. falls keines der Symbole an dieser Position eine Variable ist, dann liefere **nicht unifizierbar**
   3. sei x die Variable an der Position und t der Term an der Position im anderen Literal (evtl. auch eine Variable)
   4. falls $x \in \text{Var}(t)$, dann liefere **nicht unifizierbar**
   5. setze subst auf subst$[x \mapsto t]$
3. liefere subst

Theorem

Sei M eine endliche Menge von Literalen.

1. Der Unifikationsalgorithmus terminiert bei Eingabe M.
2. Wenn L keinen Unifikator hat, dann liefert der Unifikationsalgorithmus **nicht unifizierbar**.
3. Wenn L einen Unifikator hat, dann liefert der Unifikationsalgorithmus einen allgemeinsten Unifikator für M.

Beweis (1/7).

1. Wir behaupten, dass die Anzahl der vorkommenden Variablen $\bigcup_{L \in M} \mathrm{FV}(\mathrm{subst}(L))$ mit jeder Iteration kleiner wird. Entweder die Iteration bricht ab oder die Substitution wird um eine Variable x, die in $\bigcup_{L \in M} \mathrm{FV}(\mathrm{subst}(L))$ vorkommt, erweitert. Danach wird x durch einen Term t ohne Vorkommen von x ersetzt. Also kommt x danach nicht mehr in $\bigcup_{L \in M} \mathrm{FV}(\mathrm{subst}(L))$ vor.

Beweis (2/7).

2. Sei M nicht unifizierbar. Da die Schleife nicht normal ablaufen kann (denn dann wäre $|\{\text{subst}(L) \mid L \in M\}| \leq 1$ und L damit unifizierbar) und der Algorithmus gemäß 1 terminiert, muss der Algorithmus **nicht unifizierbar** liefern.

3. Sei M unifizierbar und sei subst_i die Substitution subst nach der i-ten Iteration. Nach 1 terminiert der Algorithmus; sei n die Iteration, nach (oder in) der der Algorithmus terminiert.

   Es gilt $\text{subst}_0 = \emptyset$ und subst_n ist die potentielle Ausgabe. Wir stellen folgende Behauptungen auf:

   - Für jeden Unifikator subst' für M und alle $0 \leq i \leq n$ existiert eine Substitution subst'', so dass $\text{subst}' = \text{subst}_i\, \text{subst}''$.
   - Im i-ten Durchlauf der Schleife terminiert der Algorithmus entweder erfolgreich (und liefert subst_n) oder erweitert die Substitution subst.

   Wir beweisen zunächst einen Spezialfall per Induktion.

Prädikatenlogik — Unifikationsalgorithmus

Beweis (3/7).

Sei $\mathrm{subst}'\colon V' \to \mathcal{T}$ ein Unifikator für M. Gelte zunächst, dass

$$\left(\bigcup_{L\in M} \mathrm{FV}(L)\right) \cap \left(\bigcup_{v\in V'} \mathrm{Var}(\mathrm{subst}(v))\right) = \emptyset$$

(M und Ersetzungsterme haben keine gemeinsamen Variablen)

Zusätzlich sei die gesuchte Substitution $\mathrm{subst}''\colon V'' \to \mathcal{T}$ eine Einschränkung von subst'; d.h. $V'' \subseteq V'$ und $\mathrm{subst}''(v'') = \mathrm{subst}'(v'')$ für alle $v'' \in V''$.

- IA: Sei $i = 0$. Dann gilt $\mathrm{subst}_0 = \emptyset$ und damit auch $\mathrm{subst}' = \emptyset\,\mathrm{subst}' = \mathrm{subst}_0\,\mathrm{subst}'$. Wir setzen also $\mathrm{subst}'' = \mathrm{subst}'$, denn dies ist offensichtlich eine Einschränkung von subst'.
- Sei $i > 0$ und per Induktionshypothese gelten die Behauptungen für die vorherige Iteration $i - 1$.

Prädikatenlogik — Unifikationsalgorithmus

Beweis (4/7).

Also existiert eine Substitution subst''_{i-1}, die eine Einschränkung von subst' ist, so dass $\mathrm{subst}' = \mathrm{subst}_{i-1}\,\mathrm{subst}''_{i-1}$. Wir unterscheiden zwei Fälle:

- Sei $|\{\mathrm{subst}_{i-1}(L) \mid L \in M\}| \leq 1$. Dann terminiert der Algorithmus bereits bei $i-1 = n$. Dies ist also unmöglich.
- Sei p die gewählte Position, an der sich die zwei gewählten Literale $L'_1 = \mathrm{subst}_{i-1}(L_1)$ und $L'_2 = \mathrm{subst}_{i-1}(L_2)$ mit $L_1, L_2 \in M$ unterscheiden. Da subst' ein Unifikator für M ist, gilt $\mathrm{subst}'(L_1) = \mathrm{subst}'(L_2)$. Dies liefert

  $$\mathrm{subst}''_{i-1}(L'_1) = \mathrm{subst}'(L_1) = \mathrm{subst}'(L_2) = \mathrm{subst}''_{i-1}(L'_2)$$

  Also können an der Position p in L'_1 und L'_2 nicht zwei verschiedene Funktions- oder Relationssymbole stehen. Aufgrund des Unterschieds an p, steht o.B.d.A. in L'_1 eine Variable x an p und in L'_2 ein Term $t \neq x$.

Prädikatenlogik — Unifikationsalgorithmus

Beweis (5/7).

Dann gilt offenbar $\text{subst}''_{i-1}(x) = \text{subst}''_{i-1}(t)$. Es bleibt zunächst zu zeigen, dass x nicht in t vorkommt. Dies ist offensichtlich falls t eine Variable $t \neq x$ oder eine Konstante ist. Sei $t = f_j^k(t_1, \ldots, t_k)$ für $j, k \in \mathbb{N}$ und Terme $t_1, \ldots, t_k \in \mathcal{T}$. Dann gilt

$$\text{subst}''_{i-1}(x) = \text{subst}''_{i-1}(t) = f_j^k(\text{subst}''_{i-1}(t_1), \ldots, \text{subst}''_{i-1}(t_k))$$

Sollte t_ℓ nun x enthalten, so erhalten wir eine unlösbare Gleichung, denn dann wäre der Term auf der rechten Seite (wobei bereits $\text{subst}''_{i-1}(t_\ell)$ mind. so groß wie $\text{subst}''_{i-1}(x)$ ist) immer echt größer als der Term $\text{subst}''_{i-1}(x)$.

Also kommt x nicht in t vor und die zweite Behauptung ist bewiesen (kein Abbruch wegen verschiedener Funktions- oder Relationssymbole oder Vorkommen von x in t). Des Weiteren ist $\text{subst}_i = \text{subst}_{i-1}[x \mapsto t]$.

Beweis (6/7).

Wir betrachten die Substitution $\mathrm{subst}''_i \colon V''_i \to \mathcal{T}$, so dass $V''_i = V''_{i-1} \setminus \{x\}$ und $\mathrm{subst}''_i(v'') = \mathrm{subst}''_{i-1}(v'')$ für alle $v'' \in V''_i$. Dies ist offenbar wieder eine Einschränkung von subst', da subst''_{i-1} eine solche Einschränkung ist.

Außerdem gilt

$$
\begin{aligned}
\mathrm{subst}_i \, \mathrm{subst}''_i &= \mathrm{subst}_{i-1} \, [x \mapsto t] \, \mathrm{subst}''_i \\
&= \mathrm{subst}_{i-1} \, \mathrm{subst}''_i \, [x \mapsto \mathrm{subst}''_i(t)] && \text{(Vertauschung)} \\
&= \mathrm{subst}_{i-1} \, \mathrm{subst}''_i \, [x \mapsto \mathrm{subst}''_{i-1}(t)] && (x \notin \mathrm{Var}(t)) \\
&= \mathrm{subst}_{i-1} \, \mathrm{subst}''_{i-1} && (\text{Def. } \mathrm{subst}''_i) \\
&= \mathrm{subst}' && \text{(IH.)}
\end{aligned}
$$

Damit ist auch die erste Behauptung bewiesen und beide Behauptungen sind im Spezialfall bewiesen.

Beweis (7/7).

Es bleibt der allgemeine Fall. Sei also $\mathrm{subst}'\colon V' \to \mathcal{T}$ ein beliebiger Unifikator für M. Weiterhin sei $X = \bigcup_{v' \in V'} \mathrm{Var}(\mathrm{subst}'(v'))$ die Menge der Variablen in Ersetzungstermen von subst'. Wir nehmen eine Menge $Y \subseteq \mathcal{V}$ von frischen Variablen an, so dass $|X| = |Y|$. Also existiert eine Bijektion $\varphi\colon X \to Y$. Diese Bijektion ist auch eine Substitution.

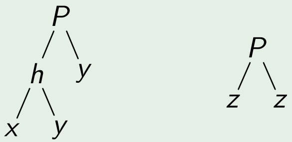
Dann ist $\mathrm{subst}'\,\varphi$ auch ein Unifikator für M, für den nun der Spezialfall anwendbar ist, denn die Variablen im Bild von φ sind frisch. Gemäß der Behauptung existiert also eine Substitution subst''_i, so dass $\mathrm{subst}'\,\varphi = \mathrm{subst}_i\,\mathrm{subst}''_i$ für alle $0 \leq i \leq n$. Folglich gilt $\mathrm{subst}' = \mathrm{subst}_i\,\mathrm{subst}''_i\,\varphi^{-1}$.

Also existiert auch für beliebige Unifikatoren eine geeignete Substitution, womit subst_n ein allg. Unifikator für M ist. $\square$

Prädikatenlogik — Unifikationsalgorithmus

einfaches Beispiel

- Literale $P\big(h(x, y), y\big)$ und $P\big(z, z\big)$
- Darstellung als Syntaxbaum

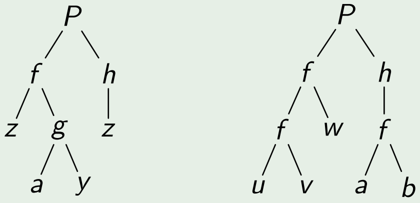


- Substitution: $[z \mapsto h(x, y)]$
- **nicht unifizierbar**, denn $y \in \mathrm{Var}(h(x, y))$

Prädikatenlogik — Unifikationsalgorithmus

weiteres Beispiel

- $P\big(f(z,g(a,y)),h(z)\big)$ und $P\big(f(f(u,v),w),h(f(a,b))\big)$
- Darstellung als Syntaxbaum



- Substitution: $[z \mapsto f(u,v)]$ $[w \mapsto g(a,y)]$ $[u \mapsto a]$ $[v \mapsto b]$
- allg. Unifikator: $[z \mapsto f(a,b),\ w \mapsto g(a,y),\ u \mapsto a,\ v \mapsto b]$

Exkurs: Anwendung

Expertensysteme

- automatische Schlussfolgerungen aus Wissensbasis
- Kodierung als Quellkode schwierig in der Wartung
  (benötigt Expertenwissen)
- stattdessen modifizierbare Wissensbasis
  und Deduktionsmechanismus (einmalig implementiert)
- es gibt auch Expertensystem-Frameworks
  (nicht so optimiert, aber nur Wissensbasis notwendig)

Prädikatenlogik — Exkurs: Anwendung

MYCIN [Shortliffe, Buchanan, 1975]

- medizinisches Diagnosesystem
- Vorreiter vieler Expertensysteme
- unterstützt Erklärungen, quantitative Folgerungen, etc.
- leicht anpassbar

Beispielregel (203)

Falls

- Entnahmestelle der Kultur Blut ist und
- Eintrittsportal des Organismus der Magen-Darm-Kanal ist und
- der Patient ein kompromittierter Wirt ist,

dann sollte die Therapie **definitiv** (1,8) Bakteroiden umfassen

Beispielregel (109)

Falls

- die Kultur von einer sterilen Quelle entnommen wurde und
- es unbekannt ist, ob der Organismus ein Kontaminant ist, und
- der Patient kein kompromittierter Wirt ist und
- der Patient aufgrund der Infektion fiebrig war und
- ein Blutbild (BB) aus der Periode der Kultur verfügbar ist und
    - die Anzahl der Leukozyten im BB über 10,5 liegt oder
    - der Anteil der neutrophilen Granulozyten im BB zum Zeitpunkt der Kulturentnahme größer als 78% war oder
    - der Anteil der Stab-Leukozyten im BB zum Zeitpunkt der Kulturentnahme größer als 10% war,

dann existiert ein stark suggestiver Nachweis (0,8), dass der Organismus kein Kontaminant ist.

Notizen

- Probleme:
  - Wissensbasis (Eingabe und Pflege der Regeln)
  - Eingabe der Beobachtungen (Arbeitsdopplung)
- zweites Problem verschwindet mit Einführung der elektronischen Krankenakte
- praktische Anwendungen von Expertensystemen:
  - Analyse chemischer Elemente basierend auf Messdaten
  - Arneimitteltherapiesicherheit
  - technische Anlagen (Kernreaktoren)
  - Naturkatastrophen-Vorhersage und -Auswirkungsvorhersage

Wozu Erfüllbarkeit?

- Erfüllbarkeit der Wissensbasis essentiell
  (Standardtest beim Speichern der Wissensbasis)
- eine unerfüllbare Wissensbasis erlaubt jedwede Folgerung

$\rightarrow$ unerfüllbare Wissensbasis ist nutzlos

- Modellfehler (genauer: Modellierungsfehler) existieren
  (ebenso wie Programmfehler)
- Erfüllbarkeit ist Mindest-Anforderung
  (ebenso wie syntaktische Korrektheit beim Programmieren)

Resolution

Prädikatenlogik — Resolution

Definition

Eine Substitution $\mathrm{subst}\colon V \to \mathcal{T}$ heißt Variablenumbenennung gdw.

- $\mathrm{subst}(v) \in \mathcal{V} \setminus V$ für alle $v \in V$ und
- subst injektiv ist

(verschiedene Variablen haben verschiedene Bilder)

Beispiele

Sei $V = \{x_1, x_2, x_3\}$ und $\mathrm{subst}\colon V \to \mathcal{T}$ mit

- $\mathrm{subst}(x_1) = x_5$, $\mathrm{subst}(x_2) = x_6$ und $\mathrm{subst}(x_3) = x_7$ ist eine Variablenumbenennung
- $\mathrm{subst}(x_1) = x_3$, $\mathrm{subst}(x_2) = x_4$ und $\mathrm{subst}(x_3) = x_5$ ist keine Variablenumbenennung (denn $\mathrm{subst}(x_1) \in V$)
- $\mathrm{subst}(x_1) = x_5$, $\mathrm{subst}(x_2) = x_6$ und $\mathrm{subst}(x_3) = x_5$ ist keine Variablenumbenennung (denn nicht injektiv)

Notizen

- sei wieder $\mathcal{L}$ die Menge der prädikatenlogischen Literale
- für eine Aussage $F = \forall x_{i_1} \cdots \forall x_{i_k} G$ (G ohne Quantoren) in konjunktiver Normalform, können wir (wie bisher) die Mengendarstellung (Menge von Mengen von Literalen) von G annehmen
- wann immer wir von konjunktiven Normalformen reden, können wir die Allquantoren auch "weglassen" (denn alle Variablen sind allquantifiziert)
- wie bisher nutzen wir $\overline{L}$ für die Negation des Literals $L \in \mathcal{L}$ (wieder ein Literal)

Prädikatenlogik — Resolution

Definition (Resolvent)

Sei $F = \forall x_{i_1} \cdots \forall x_{i_k} G$ eine Aussage in konjunktiver Normalform.

Ein Menge $R \subseteq \mathcal{L}$ von Literalen ist Resolvent von F gdw. zwei Mengen $\{D_1, D_2\} \subseteq G$ von Literalen existieren, so dass

1. Variablenumbenennungen subst_1 und subst_2 existieren mit $D_1' = \{\text{subst}_1(L) \mid L \in D_1\}$, $D_2' = \{\text{subst}_2(L) \mid L \in D_2\}$ und
$$\Big(\bigcup_{L \in D_1'} \text{FV}(L)\Big) \cap \Big(\bigcup_{L \in D_2'} \text{FV}(L)\Big) = \emptyset$$
2. $m, n \in \mathbb{N} \setminus \{0\}$ und Literale $L_1, \ldots, L_m \in D_1'$ und $L_1', \ldots, L_n' \in D_2'$ existieren, so dass $\{\overline{L_1}, \ldots, \overline{L_m}, L_1', \ldots, L_n'\}$ unifizierbar ist; sei subst ein allgemeinster Unifikator dafür
3. der Resolvent R ist
$$\{\text{subst}(L) \mid L \in (D_1' \setminus \{L_1, \ldots, L_m\}) \cup (D_2' \setminus \{L_1', \ldots, L_n'\})\}$$

einfaches Beispiel

Aussage $\forall x\big(P(x) \wedge \neg P(f(f(x)))\big)$ mit konjunktiver Normalform $\big\{\{P(x)\}, \{\neg P(f(f(x)))\}\big\}$ in Mengendarstellung

1. Variablenumbenennungen $[x \mapsto y]$ und $[x \mapsto z]$ liefern

$$D_1' = \{P(y)\} \qquad \text{und} \qquad D_2' = \{\neg P(f(f(z)))\}$$

2. allgemeinster Unifikator $[y \mapsto f(f(z))]$ für

$$\{\neg P(y), \neg P(f(f(z)))\}$$

3. Resolvent $\emptyset$

Prädikatenlogik — Resolution

Rezept für Resolventen

1. Variablen umbenennen
2. Literale auswählen und allg. Unifikator berechnen
3. wie bisher Literale entfernen, aber Unifikator anwenden

typische Fehler

- Variablenumbenennung vergessen

  $$\{\neg P(x), \neg P(f(f(x)))\}$$

  ist nicht unifizierbar

- Unifikator nicht auf Literale angewandt

  $$D'_1 = \{P(y), R(y)\} \qquad \text{und} \qquad D'_2 = \{\neg P(f(f(z)))\}$$

  mit Unifikator $[y \mapsto f(f(z))]$ liefert Resolvent $\{R(f(f(z)))\}$

weiteres Beispiel

konjunktive Normalform
$\{\{P(f(x)), P(z), \neg Q(z)\}, \{\neg P(x), R(g(x), a)\}\}$

1. Variablenumbenennungen $[x \mapsto x', z \mapsto z']$ und $[x \mapsto x'']$ liefern

$$D_1' = \{P(f(x')), P(z'), \neg Q(z')\}$$
$$D_2' = \{\neg P(x''), R(g(x''), a)\}$$

2. allgemeinster Unifikator $[z' \mapsto f(x'), x'' \mapsto f(x')]$ für

$$\{\neg P(f(x')), \neg P(z'), \neg P(x'')\}$$

3. Resolvent $\{\neg Q(f(x')), R(g(f(x')), a)\}$

Aufgabe

Sind folgende Mengen von Literalen resolvierbar?

- $\{P(x), Q(x,y)\}$ und $\{\neg P(f(x))\}$ ✓
  Resolvent: $\{Q(f(x'), y')\}$
- $\{P(g(x)), Q(f(x))\}$ und $\{\neg P(f(x))\}$ ✗
  keine unifizierbaren Literale
- $\{P(x), P(f(x))\}$ und $\{\neg P(y), Q(x,y)\}$ ✓
  Resolventen:
  $\{P(f(x')), Q(x'', x')\}$ und $\{P(x'), Q(x'', f(x'))\}$
  (kein Resolvent enthält nur das Q-Literal)

Beispiel (Deduktion)

$F = \{\{P(f(x))\}, \{\neg P(x), Q(x, f(x))\}, \{\neg Q(f(a), f(f(a)))\}\}$

1	$\{P(f(x))\}$	Element von F
2	$\{\neg P(x), Q(x, f(x))\}$	Element von F
3	$\{Q(f(x'), f(f(x')))\}$	Resolvent von {1, 2}
4	$\{\neg Q(f(a), f(f(a)))\}$	Element von F
5	$\emptyset$	Resolvent von {3, 4}

Prädikatenlogik — Resolution

Notizen

- Deduktionen und Resolutionsalgorithmus wie in der Aussagenlogik
- Korrektheit und Vollständigkeit auch wie in Aussagenlogik

Fragen

Sei $F \in \mathcal{F}$ eine prädikatenlogische Aussage in konjunktiver NF

- Korrektheit: Wenn per Resolution die leere Menge aus F herleitbar ist, dann ist F unerfüllbar
- Vollständigkeit: Wenn F unerfüllbar ist, dann ist aus F per Resolution die leere Menge herleitbar

Zusammenfassung

- Unifikationsalgorithmus
- Expertensysteme
- prädikatenlogische Resolution

Siebente Übungsserie ist bereits verfügbar.