Vorlesung “Logik”

10-201-2108-1

10. PL1 – Grundresolution und Unifikation

Ringo Baumann
Professur für Formale Argumentation
und Logisches Schließen

26. Juni 2025
Leipzig

In der letzten Vorlesung

Herbrand-Modellsatz
Satz von Löwenheim-Skolem
Satz von Herbrand
Algorithmus von Gilmore

Fahrplan für diese Vorlesung

- Grundresolution
- Allgemeine Substitution
- Unifikation
- Prädikatenlogische Resolution

Grundresolution

- Idee: verwende aussagenlogische Resolution zum Nachweis der Unerfüllbarkeit
- dies ist möglich aufgrund des Satzes von Herbrand und des Algorithmus von Gilmore

Proposition (Herbrand, 1930)

Sei $\phi = \forall x_1 \ldots \forall x_n \, \xi$ gleichheitsfreier Satz in SNF. Es gilt:

ϕ erfüllbar gdw. $E(\phi)$ im aussagenlogischen Sinne erfüllbar

Algorithmus von Gilmore

Sei ϕ gleichheitsfreier Satz in SNF und sei $\{\phi_1, \phi_2, \phi_3, \ldots\}$ eine Aufzählung von $E(\phi)$. Für i = 1,2,3,... teste:

- *Ist $\phi_1 \wedge \ldots \wedge \phi_i$ aussagenlogisch unerfüllbar?*
- *Falls ja, halte an mit Ausgabe "ϕ unerfüllbar"*

Grundresolutionsalgorithmus

Grundresolutionsalgorithmus

Sei ϕ gleichheitsfreier Satz in SNF mit Matrix in KNF und $\{\phi_1, \phi_2, \ldots\}$ eine Aufzählung von $E(\phi)$. Für $i = 1, 2, \ldots$ teste:

- $\Box \in \mathrm{Res}^*(M(\phi_1) \cup \ldots \cup M(\phi_i))$ *?*
- *Falls ja, halte an mit Ausgabe "ϕ unerfüllbar"*

$$
\begin{aligned}
\phi &= \forall x\,(P(x) \wedge \neg P(f(f(x)))) \\
D(\phi) &= \{c, f(c), f(f(c)), f(f(f(c))), \ldots\} \\
E(\phi) &= \{\underbrace{P(c) \wedge \neg P(f(f(c)))}_{\phi_1},\ \underbrace{P(f(c)) \wedge \neg P(f(f(f(c))))}_{\phi_2}, \ldots\} \\
M(\phi_1) &= \{\{P(c)\},\ \{\neg P(f(f(c)))\}\} \\
\textstyle\bigcup_{i=1}^{2} M(\phi_i) &= \{\{P(c)\},\ \{\neg P(f(f(c)))\},\ \{P(f(c))\},\ \{\neg P(f(f(f(c))))\}\} \\
\textstyle\bigcup_{i=1}^{3} M(\phi_i) &= \{\{P(c)\},\ \{\neg P(f(f(c)))\},\ \{P(f(c))\},\ \{\neg P(f(f(f(c))))\}, \\
&\qquad \{P(f(f(c)))\},\ \{\neg P(f(f(f(f(c)))))\}\}
\end{aligned}
$$

$$\Box \in \mathrm{Res}^*(M(\phi_1) \cup M(\phi_2) \cup M(\phi_3))$$

Grundresolutionsalgorithmus

Grundresolutionsalgorithmus

Sei ϕ gleichheitsfreier Satz in SNF mit Matrix in KNF und $\{\phi_1, \phi_2, \ldots\}$ eine Aufzählung von $E(\phi)$. Für $i = 1, 2, \ldots$ teste:

- $\Box \in \mathrm{Res}^*(M(\phi_1) \cup \ldots \cup M(\phi_i))$?
- *Falls ja, halte an mit Ausgabe "ϕ unerfüllbar"*

- Algorithmus ist korrekt und vollständig bzgl. Unerfüllbarkeit
- Aber! extrem ineffizient, da alle Grundinstanzen betrachtet werden

Zum Beispiel:

für $\xi = P(x) \wedge \neg P(f(f(x)))$ reicht $[x/c]$ und $[x/f(f(c))]$ aus

$\Rightarrow$ zielgerichtete Instantiierung

Allgemeine Substitution

- Substitution [x/t] (VL8) ist ein Spezialfall – ersetzen einer Variablen durch einen Term
- jetzt: mehrfache gleichzeitige Variablenersetzung

Definition

Sei $V \subseteq \mathcal{V}$ und $\sigma : V \to \mathcal{T}$. Wir definieren die Substitution σ, $\sigma : \mathcal{T} \to \mathcal{T}$ mit $t \mapsto t\sigma$ rekursiv durch:

- für $x \in \mathcal{V}$: $x\sigma = \begin{cases} \sigma(x), & \text{falls } x \in V \\ x, & \text{sonst} \end{cases}$
- für $c \in \mathcal{C}$: $c\sigma = c$
- für $f \in \mathcal{F}$ mit $ar(f) = n \geq 1$ und $t_1, \dots, t_n \in \mathcal{T}$

$$f(t_1, \dots, t_n)\sigma = f(t_1\sigma, \dots, t_n\sigma)$$

$$\text{sei } V = \{x, y\}\,, \quad \sigma(x) = h(y), \quad \sigma(y) = g(y, x)$$

$$f(h(x), y)\sigma = f(h(h(y)), g(y, x))$$

Notation und Eigenschaften

- für Substitution $\sigma : V \to \mathcal{T}$ mit $V = \{x_1, \ldots, x_n\}$ endlich und $x_i\sigma = t_i$ für $1 \le i \le n$, schreiben wir auch $[x_1/t_1, \ldots, x_n/t_n]$
- Simultane vs. Sequentielle Ersetzung:

$$f(h(x), y)[x/h(y)][y/g(y, x)] = f(h(h(g(y, x))), g(y, x))$$
$$f(h(x), y)[x/h(y),\ y/g(y, x)] = f(h(h(y)), g(y, x))$$

- für zwei Substitution σ und τ und eine Variable x sei:

$$x(\sigma\tau) = (x\sigma)\tau \qquad \text{(Komposition)}$$

- via Terminduktion folgt $t(\sigma\tau) = (t\sigma)\tau$ für beliebige Terme t
- Komposition ist assoziativ, d. h. $(\sigma_1\sigma_2)\sigma_3 = \sigma_1(\sigma_2\sigma_3)$
  Beweis:

$$\begin{aligned} t((\sigma_1\sigma_2)\sigma_3) &= (t(\sigma_1\sigma_2))\sigma_3 \\ &= ((t\sigma_1)\sigma_2)\sigma_3 \\ &= (t\sigma_1)(\sigma_2\sigma_3) \\ &= t(\sigma_1(\sigma_2\sigma_3)) \end{aligned}$$

Kommutativität

- Vertauschung in bestimmten Fällen möglich

Proposition

Gegeben Substitution $\sigma : V \to \mathcal{T}$ mit $x \notin V$ und $x \notin var(y\sigma)$ für alle $y \in V$. Für jeden Term $t \in \mathcal{T}$ gilt: $[x/t]\sigma = \sigma[x/t\sigma]$

- Einschränkungen wichtig

1. $f(x)[x/c]\underbrace{[x/d]}_{\sigma} = f(c) \neq f(d) = f(x)\underbrace{[x/d]}_{\sigma}\underbrace{[x/c]}_{[x/c\sigma]} \qquad (x \in V)$
2. $g(x,y)[x/c]\underbrace{[y/x]}_{\sigma} = g(c,x) \neq g(c,c) = g(x,y)\underbrace{[y/x]}_{\sigma}\underbrace{[x/c]}_{[x/c\sigma]}$
   $(x \in var(y\sigma))$

Kommutativität

Proposition

Gegeben Substitution $\sigma : V \to \mathcal{T}$ und $x \notin V$ als auch $x \notin var(y\sigma)$ für alle $y \in V$. Für jeden Term $t \in \mathcal{T}$ gilt: $[x/t]\sigma = \sigma[x/t\sigma]$

Beweis: Zeige $u[x/t]\sigma = u\sigma[x/t\sigma]$ für Terme u mittels Termind.

- Sei u eine Variable. Zwei Fälle: 1. $u = x$. Es gilt: $x[x/t]\sigma = t\sigma$. Da $x \notin V$ folgt $x\sigma[x/t\sigma] = x[x/t\sigma] = t\sigma$.
  2. $u = z \neq x$. Es gilt: $z[x/t]\sigma = z\sigma$ und ebenso, $z\sigma[x/t\sigma] = z\sigma$ da $x \notin var(z\sigma)$.
- Sei $u = c$ Konstante. Folglich, $c[x/t]\sigma = c\sigma = c\sigma[x/t\sigma]$
- Sei $u = f(t_1, \ldots, t_n)$. Es gilt:

$$
\begin{aligned}
f(t_1, \ldots, t_n)[x/t]\sigma &= f(t_1[x/t], \ldots, t_n[x/t])\sigma \\
&= f(t_1[x/t]\sigma, \ldots, t_n[x/t]\sigma) \\
&= f(t_1\sigma[x/t\sigma], \ldots, t_n\sigma[x/t\sigma]) \qquad \text{(IV)} \\
&= f(t_1\sigma, \ldots, t_n\sigma)[x/t\sigma] \\
&= (f(t_1, \ldots, t_n)\sigma)[x/t\sigma]
\end{aligned}
$$

Unifikation

- Ziel: Literale durch Substitution in gleiche (bzw. komplementäre) Gestalt überführen
- Sei σ eine Substitution und $P(t_1, \ldots, t_n)$ eine atomare Aussage. Wir setzen:

$$P(t_1, \ldots, t_n)\sigma = P(t_1\sigma, \ldots, t_n\sigma)$$
$$(\neg P(t_1, \ldots, t_n))\sigma = \neg P(t_1\sigma, \ldots, t_n\sigma)$$

Definition

Sei $S = \{L_1, \ldots, L_n\}$ eine Menge von Literalen und σ eine Substitution. σ heißt Unifikator für S, falls $L_1\sigma = \ldots = L_n\sigma$.

$[x/g(c),\ y/f(g(c))]$ ist Unifikator für $S = \{P(x, y),\ P(g(c), f(x))\}$ da

$$P(x, y)[x/g(c),\ y/f(g(c))] = P(g(c),\ f(g(c))), \text{ und}$$
$$P(g(c), f(x))[x/g(c),\ y/f(g(c))] = P(g(c),\ f(g(c))).$$

- nicht jede Menge S ist unifizierbar (kein Unifikator für S)
- falls unifizierbar, dann Unifikator nicht eindeutig bestimmt

Hörsaalübung

Definition

Sei $S = \{L_1, \ldots, L_n\}$ eine Menge von Literalen und σ eine Substitution. σ heißt Unifikator für S, falls $L_1\sigma = \ldots = L_n\sigma$.

Vervollständigen Sie nachfolgende Tabelle. Bei Fragen konsultieren Sie die Person rechts oder links von Ihnen. (3 min)

Literal 1	Literal 2	Unifizierbar	Unifikator
$P(x, y)$	$P(g(c), f(x))$	Ja	$[x/g(c),\ y/f(g(c))]$
$Q(x, f(y))$	$P(g(c), f(x))$	Nein	Q vs. P
$\neg R(x)$	$\neg R(f(x))$	Nein	für jede $[x/t]$: $t \neq f(t)$
$\neg Q(g(x), a)$	$\neg Q(f(y), a)$	Nein	g vs. f
$P(x, g(y))$	$P(f(y), g(z))$	Ja	$[x/f(y),\ z/y]$
$Q(x, f(z))$	$Q(f(z), x)$	Ja	$[x/f(z)]$

Allgemeinster Unifikator

Definition

Seien σ_1 und σ_2 Substitutionen. σ_1 ist allgemeiner als σ_2 ($\sigma_1 \leq \sigma_2$), falls eine Substitution τ existiert, so daß: $\sigma_1 \tau = \sigma_2$.

Beispiel: $S = \{P(x), P(y)\},\ \sigma_1 = [x/y],\ \sigma_2 = [x/c,\ y/c]$

Es gilt: $\sigma_1 \leq \sigma_2$, da $[x/y][y/c] = [x/c,\ y/c]$

Proposition

Die Relation $\leq$ ist transitiv.

Beweis: Seien $\sigma_1, \sigma_2, \sigma_3, \tau_1, \tau_2$ Substitutionen mit $\sigma_1 \tau_1 = \sigma_2$ und $\sigma_2 \tau_2 = \sigma_3$. Folglich $\sigma_3 = \sigma_2 \tau_2 = (\sigma_1 \tau_1)\tau_2 \underbrace{=}_{\text{Assoziativität}} \sigma_1(\tau_1 \tau_2) = \sigma_3$. $\square$

Proposition

Falls σ_1 Unifikator einer Literalmenge S und gilt $\sigma_1 \leq \sigma_2$, dann auch σ_2 Unifikator der Menge S.

Allgemeinster Unifikator

Proposition

Falls σ_1 Unifikator einer Literalmenge S und gilt $\sigma_1 \leq \sigma_2$, dann auch σ_2 Unifikator der Menge S.

Beweis: Da σ_1 Unifikator von S gilt $L_i\sigma_1 = L_j\sigma_1$ für alle $L_i, L_j \in S$. Sei also $t = L\sigma_1$ für alle $L \in S$. Da $\sigma_1 \leq \sigma_2$ existiert ein τ mit $\sigma_1\tau = \sigma_2$. Somit gilt für jedes $L \in S$: $L\sigma_2 = L(\sigma_1\tau) = (L\sigma_1)\tau = t\tau$.

Definition

Sei $S = \{L_1, \ldots, L_n\}$ eine Menge von Literalen und σ eine Substitution. σ heißt allgemeinster Unifikator (mgu) für S, falls:

1. σ ist Unifikator für S, und
2. falls τ Unifikator für S, dann $\sigma \leq \tau$.

Achtung! Obwohl Name allgemeinster Unifikator kann es mehrere geben: $S = \{P(x), P(y)\}$, $\sigma_1 = [x/y]$, $\sigma_2 = [y/x]$

Unifikationsalgorithmus

- Ziel: Auffinden eines allgemeinsten Unifikators im Falle der Unifizierbarkeit

Unifikationsalgorithmus

Eingabe: endliche, nichtleere Menge S von Literalen

1. *setze* $\sigma = []$ *(leere Substitution)*
2. *solange* $|\{L\sigma \mid L \in S\}| > 1$, *(noch nicht unifiziert)*
   finde erste Position, an der sich $L_1\sigma$ *und* $L_2\sigma$ *mit* $L_1, L_2 \in S$ *unterscheiden*
   - *falls, an dieser Position weder* $L_1\sigma$ *noch* $L_2\sigma$ *eine Variable aufweist, gib "nicht unifizierbar" aus und stoppe (Clash)*
   - *sonst, d.h. ein Zeichen Variable* x *und andere Term* t
     - *falls,* $x \in var(t)$*, gib "nicht unifizierbar" aus und stoppe (Cycle)*
     - *andernfalls, erweitere Substitution:* $\sigma := \sigma[x/t]$
3. *gib "unifizierbar mit mgu* σ*" aus und stoppe*

...terminiert, korrekt und vollständig

Unifikationsalgorithmus

Gegeben die folgenden beiden Literale:

$$L_1 = P(f(z, g(a, y)),\ h(z)), \quad L_2 = P(f(f(u, v), w),\ h(f(a, b)))$$

Substitution σ	**Unterscheidungsstelle**
$\sigma = []$	$L_1\sigma = P(f(z,\ g(a, y)),\ h(z))$ $L_2\sigma = P(f(f(u, v),\ w),\ h(f(a, b)))$
$\sigma = [z/f(u, v)]$	$L_1\sigma = P(f(f(u, v),\ g(a, y)),\ h(f(u, v)))$ $L_2\sigma = P(f(f(u, v),\ w),\ h(f(a, b)))$
$\sigma = [z/f(u, v)][w/g(a, y)]$	$L_1\sigma = P(f(f(u, v),\ g(a, y)),\ h(f(u, v)))$ $L_2\sigma = P(f(f(u, v),\ g(a, y)),\ h(f(a, b)))$
$\sigma = [z/f(u, v)][w/g(a, y)][u/a]$	$L_1\sigma = P(f(f(a, v),\ g(a, y)),\ h(f(a, v)))$ $L_2\sigma = P(f(f(a, v),\ g(a, y)),\ h(f(a, b)))$
$\sigma = [z/f(a, b)][w/g(a, y)]$ $[u/a][v/b]$	$L_1\sigma = P(f(f(a, b),\ g(a, y)),\ h(f(a, b)))$ $L_2\sigma = P(f(f(a, b),\ g(a, y)),\ h(f(a, b)))$

Unifikationsalgorithmus

Theorem (Robinson, 1965)

Für jede nichtleere, endliche Menge S von Literalen gilt:

(A) *Der Unifikationsalgorithmus terminiert.*

(B) *Falls S nicht unifizierbar, so terminiert der Algorithmus mit Ausgabe "nicht unifizierbar".*

(C) *Falls S unifizierbar, so terminiert der Algorithmus und liefert einen allgemeinsten Unifikator von S.*

(A) Die Anzahl der Variablen in S ist endlich. Eine nichtterminierende Schleifenbegehung erfordert eine Variable x und Term t mit $x \notin var(t)$ (die Unterscheidungsstelle) und setzt $\sigma := \sigma[x/t]$. Dadurch verschwindet die Variable x aus allen Literalen $L\sigma[x/t]$ mit $L \in S$. Folglich wird die Anzahl der Variablen in jedem Durchlauf echt kleiner. Daraus folgt, dass die Schleife entweder innerhalb eines Durchlaufs abbricht oder regulär verlassen wird, d.h. $|\{L\sigma \mid L \in S\}| = 1$, woraufhin der Algorithmus gemäß Schritt 3 terminiert.

Unifikationsalgorithmus

Theorem (Robinson, 1965)

Für jede nichtleere, endliche Menge S von Literalen gilt:

(A) *Der Unifikationsalgorithmus terminiert.*

(B) *Falls S nicht unifizierbar, so terminiert der Algorithmus mit Ausgabe "nicht unifizierbar".*

(C) *Falls S unifizierbar, so terminiert der Algorithmus und liefert einen allgemeinsten Unifikator von S.*

(B) Sei S nicht unifizierbar. Somit kann die Schleife nicht regulär verlassen werden, denn andernfalls wäre $|\{L\sigma \mid L \in S\}| = 1$ und damit S doch unifizierbar. Da nach (A) der Algorithmus terminiert, muß innerhalb einer Schleifenbegehung abgebrochen werden. Dies ist nur durch die Ausgabe "nicht unifizierbar" möglich.

Unifikationsalgorithmus

Theorem (Robinson, 1965)

Für jede nichtleere, endliche Menge S von Literalen gilt:

(A) *Der Unifikationsalgorithmus terminiert.*

(B) *Falls S nicht unifizierbar, so terminiert der Algorithmus mit Ausgabe "nicht unifizierbar".*

(C) *Falls S unifizierbar, so terminiert der Algorithmus und liefert einen allgemeinsten Unifikator von S.*

(C) Ohne Beweis (siehe z.B. Schöning).

Aus (C) folgt, dass jede unifizierbare Menge einen mgu besitzt.

Prädikatenlogische Resolution

Definition

Substitution $\sigma : V \to \mathcal{T}$ heißt Variablenumbenennung falls: $\sigma(x) \in \mathcal{V} \setminus V$ für alle $v \in V$, und σ ist injektiv.

Definition

Sei $\phi = \forall x_1 \ldots \forall x_n \xi$ gleichheitsfreier Satz in SNF mit Matrix ξ in KNF und $M(\xi) = \{D_1, \ldots, D_m\}$. Eine Klausel R heißt (prädikatenlogische) Resolvente von D_i und D_j (bzw. von ϕ), falls:

1. es existieren Variablenumbenennungen σ_1,σ_2, so daß:
$$frei(D_i\sigma_1) \cap frei(D_j\sigma_2) = \varnothing$$
2. es existieren nichtleere $D_i' \subseteq D_i$ und $D_j' \subseteq D_j$ mit σ ist mgu von $\overline{D_i'}\sigma_1 \cup D_j'\sigma_2$, wobei $\overline{D_i'} = \{\overline{L} \mid L \in D_i'\}$ und
3. $R = \Big((D_i\sigma_1 \setminus D_i'\sigma_1) \cup (D_j\sigma_2 \setminus D_j'\sigma_2)\Big)\sigma$

Prädikatenlogische Resolution

Beispiele:

1. Sei $\phi = \forall x \underbrace{(P(x) \wedge \neg P(f(f(x))))}_{\xi}$
   - $M(\xi) = \{\underbrace{\{P(x)\}}_{D_1}, \underbrace{\{\neg P(f(f(x)))\}}_{D_2}\}$
   - Variablenumbennung $\sigma_1 = [x/y]$ und $\sigma_2 = []$ liefert

     $$D_1\sigma_1 = \{P(y)\} \text{ und } D_2\sigma_2 = \{\neg P(f(f(x)))\}$$

   - für $D_1' = D_1$ und $D_2' = D_2$ ist $\sigma = [y/f(f(x))]$ mgu von

     $$\overline{D_1'}\sigma_1 \cup D_2'\sigma_2 = \{\neg P(y), \neg P(f(f(x)))\}$$

   - $R = \big((D_i\sigma_1 \setminus D_i'\sigma_1) \cup (D_j\sigma_2 \setminus D_j'\sigma_2)\big)\sigma = (\emptyset \cup \emptyset)\sigma = \emptyset = \Box$

Prädikatenlogische Resolution

Beispiele:

2. $\phi = \forall x\,\forall z\;\underbrace{((P(f(x)) \vee P(z) \vee \neg Q(z)) \wedge (\neg P(x) \vee R(g(x),a)))}_{\xi}$

   - $M(\xi) = \{\underbrace{\{P(f(x)),\ P(z),\ \neg Q(z)\}}_{D_1},\ \underbrace{\{\neg P(x),\ R(g(x),a)\}}_{D_2}\}$
   - Variablenumbenennung: $\sigma_1 = []$, $\sigma_2 = [x/y]$ ergibt

     $$D_1\sigma_1 = \{P(f(x)),\ P(z),\ \neg Q(z)\}, \quad D_2\sigma_2 = \{\neg P(y),\ R(g(y),a)\}$$

   - für $D_1' = \{P(f(x)), P(z)\}$ und $D_2' = \{\neg P(y)\}$ ist $\sigma = [z/f(x)][y/f(x)]$ mgu von

     $$\overline{D_1'}\sigma_1 \cup D_2'\sigma_2 = \{\neg P(f(x)), \neg P(z), \neg P(y)\}$$

   - Resolvente $R = ((D_1\sigma_1 \setminus D_1'\sigma_1) \cup (D_2\sigma_2 \setminus D_2'\sigma_2))\,\sigma$

     $$\begin{aligned} &= (\{\neg Q(z)\} \cup \{R(g(y),a)\})\sigma \\ &= \{\neg Q(f(x)),\ R(g(f(x)),a)\} \end{aligned}$$

Vorlesung “Logik”

10-201-2108-1

10. PL1 – Grundresolution und Unifikation

Ringo Baumann
Professur für Formale Argumentation
und Logisches Schließen

26. Juni 2025
Leipzig