Logik
Vorlesung 13: Resolution und Prolog

Andreas Maletti

30. Januar 2015

Überblick

Inhalt

Vorlesungsziele

heutige Vorlesung

1. Übertragungstheorem
2. Beweise für prädikatenlogische Resolution
3. abschließendes Beispiel
4. logische Programmierung

Bitte Fragen direkt stellen!

Überblick

Organisation

Organisation

Tutorium

am 6.02.2015 ab 11:00 Uhr im **Hs. 5**

- bringen Sie Ihre Fragen mit

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
- (wir bringen unsere Fragen mit)

Wiederholung: Resolution

Prädikatenlogik — Resolution

Definition (Resolvent)

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

$R \subseteq \mathcal{L}$ ist Resolvent von F gdw. $D_1, D_2 \in G$ existieren, so dass

1. Variablenumbenennungen subst_1 und subst_2 existieren mit $D_1' = \{\mathrm{subst}_1(L) \mid L \in D_1\}$ und $D_2' = \{\mathrm{subst}_2(L) \mid L \in D_2\}$ (keine gemeinsamen Variablen in D_1' und D_2')
2. $L_1, \ldots, L_m \in D_1'$ 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 mit allg. Unif. subst ist
3. $R = \{\mathrm{subst}(L) \mid L \in (D_1' \setminus \{L_1, \ldots, L_m\}) \cup (D_2' \setminus \{L_1', \ldots, L_n'\})\}$

Rezept für Resolventen

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

Prädikatenlogik — Resolution

Beispiel (Deduktion)

$$F = \{\{P(x),\ \neg Q(x),\ \neg R(f(x))\},\ \{P(x),\ \neg Q(x),\ \neg S(x, f(x))\},\ \{\neg P(a)\},\ \{\neg T(a)\},\ \{S(a, z),\ \neg T(z)\},\ \{T(x),\ Q(x)\},\ \{T(y),\ R(y)\}\}$$

1	$\{\neg T(a)\}$	in F
2	$\{T(x), Q(x)\}$	in F
3	$\{Q(a)\}$	Res. $\{1, 2\}$
4	$\{\neg P(a)\}$	in F
5	$\{P(x), \neg Q(x), \neg R(f(x))\}$	in F
6	$\{\neg Q(a), \neg R(f(a))\}$	$\{4, 5\}$
7	$\{P(x), \neg Q(x), \neg S(x, f(x))\}$	in F
8	$\{\neg Q(a),\ \neg S(a, f(a))\}$	$\{4, 7\}$
9	$\{\neg S(a, f(a))\}$	$\{3, 8\}$
10	$\{\neg R(f(a))\}$	$\{3, 6\}$
11	$\{S(a, z), \neg T(z)\}$	in F
12	$\{\neg T(f(a))\}$	$\{9, 11\}$
13	$\{T(y),\ R(y)\}$	in F
14	$\{R(f(a))\}$	$\{12, 13\}$
15	$\emptyset$	$\{10, 14\}$

Definition

Sei $L \in \mathcal{L}$ ein Literal.

Ein Literal $L' \in \mathcal{L}$ ist eine Grundinstanz von L gdw.

- $\mathrm{FV}(L') = \emptyset$ (variablenfrei)
- es existiert eine Substitution $\mathrm{subst}\colon V \to \mathcal{T}$, so dass $L' = \mathrm{subst}(L)$. (Instanz von L)

Beispiel

Für das Literal $P(x, f(x), g(a, y))$

- ist $P(g(f(b), a), f(g(f(b), a)), g(a, f(b)))$ eine Grundinstanz (Substitution: $[x \mapsto g(f(b), a),\ y \mapsto f(b)]$)
- ist $P(g(f(b), a), f(b), g(a, f(b)))$ keine Grundinstanz (x nicht konsistent ersetzt)
- ist $P(g(f(x), a), b, g(a, f(b)))$ keine Grundinstanz (nicht variablenfrei)

Prädikatenlogik — Resolution

Definition

Sei $M \subseteq \mathcal{L}$ ein endliche Menge von Literalen.
Eine Literalmenge $M' \subseteq \mathcal{L}$ ist eine Grundinstanz von M gdw.

- $\mathrm{FV}(L') = \emptyset$ für alle $L' \in M'$ (variablenfrei)
- es existiert eine Substitution $\mathrm{subst}\colon V \to \mathcal{T}$, so dass $M' = \{\mathrm{subst}(L) \mid L \in M\}$. (Instanzen von M)

Beispiel

Für die Menge $\{P(x, f(x), g(a, y)),\ Q(x)\}$

- ist $\{P(f(b), f(f(b)), g(a, a)),\ Q(f(b))\}$ eine Grundinstanz (Substitution: $[x \mapsto f(b),\ y \mapsto a]$)
- ist $\{P(f(b), f(f(b)), g(a, a))\}$ keine Grundinstanz (keine Instanz von $Q(x)$)
- ist $\{P(f(b), f(f(b)), g(a, a)),\ Q(a)\}$ keine Grundinstanz (inkonsistente Ersetzung von x)

Prädikatenlogik — Resolution

Theorem (Übertragungstheorem)

Seien M_1 und M_2 zwei endliche Mengen von prädikatenlogischen Literalen und M_1' und M_2' zwei entsprechende Grundinstanzen.

Falls R' ein aussagenlogischer Resolvent von $\{M_1', M_2'\}$ ist, dann existiert ein Resolvent von $\{M_1, M_2\}$, so dass R' eine Grundinstanz von R ist.

$$
\begin{tikzcd}
M_1 \arrow[dd, "\text{Grundinstanz}"'] \arrow[dr, dash, "\text{Res.}"'] & & M_2 \arrow[dd, "\text{Grundinstanz}"] \arrow[dl, dash, "\text{Res.}"] \\
& R \arrow[dd, "\text{Grund-}\ \text{instanz}"] & \\
M_1' \arrow[dr, dash, "\text{Res.}"'] & & M_2' \arrow[dl, dash, "\text{Res.}"] \\
& R' &
\end{tikzcd}
$$

Prädikatenlogik — Resolution

$$
\begin{tikzcd}
\{P(x), R(x,y)\} \arrow[dd, "\text{Grundinstanz}"'] \arrow[dr, dash, "\text{Res.}"] & & \{\neg P(f(x)), Q(a)\} \arrow[dd, "\text{Grundinstanz}"] \arrow[dl, dash, "\text{Res.}"'] \\
& \{R(f(x''), y'), Q(a)\} \arrow[dd, "\text{Grund-}"', "\text{instanz}"] & \\
\{P(f(b)), R(f(b), a)\} \arrow[dr, dash, "\text{Res.}"] & & \{\neg P(f(b)), Q(a)\} \arrow[dl, dash, "\text{Res.}"'] \\
& \{R(f(b), a), Q(a)\} &
\end{tikzcd}
$$

Beweis des Übertragungstheorems (1/3).

Seien zunächst subst_1 und subst_2 Variablenumbenennungen, so dass $\left(\bigcup_{L \in N_1} \mathrm{FV}(L)\right) \cap \left(\bigcup_{L \in N_2} \mathrm{FV}(L)\right) = \emptyset$ wobei

$$N_1 = \{\mathrm{subst}_1(L) \mid L \in M_1\} \qquad \text{und} \qquad N_2 = \{\mathrm{subst}_2(L) \mid L \in M_2\}$$

Dann sind M_1' und M_2' Grundinstanzen von N_1 und N_2. Seien also $\mathrm{subst}_1' \colon V_1' \to \mathcal{T}$ und $\mathrm{subst}_2' \colon V_2' \to \mathcal{T}$ Substitutionen, so dass

$$M_1' = \{\mathrm{subst}_1'(L) \mid L \in N_1\} \qquad \text{und} \qquad M_2' = \{\mathrm{subst}_2'(L) \mid L \in N_2\}$$

Es gilt offensichtlich

1. $V_1' = \bigcup_{L \in N_1} \mathrm{FV}(L)$ und $V_2' = \bigcup_{L \in N_2} \mathrm{FV}(L)$
2. $\mathrm{FV}(\mathrm{subst}_1'(v')) = \emptyset$ für alle $v' \in V_1'$ und
   $\mathrm{FV}(\mathrm{subst}_2'(v'')) = \emptyset$ für alle $v'' \in V_2'$
3. $V_1' \cap V_2' = \emptyset$

Prädikatenlogik — Resolution

Beweis des Übertragungstheorems (2/3).

Sei also nun $\text{subst}' = \text{subst}'_1\text{subst}'_2$. Dann gilt insbesondere

$$M'_1 = \{\text{subst}'(L) \mid L \in N_1\} \qquad \text{und} \qquad M'_2 = \{\text{subst}'(L) \mid L \in N_2\}$$

Nach Voraussetzung ist R' ein aussagenlogischer Resolvent von $\{M'_1, M'_2\}$. Also existiert $L \in M'_1$ mit $\overline{L} \in M'_2$, so dass $R' = \left(M'_1 \setminus \{L\}\right) \cup \left(M'_2 \setminus \{\overline{L}\}\right)$ Seien also

$$\{L_1, \ldots, L_m\} = \{L' \in N_1 \mid \text{subst}'(L') = L\}$$
$$\{L'_1, \ldots, L'_n\} = \{L' \in N_2 \mid \text{subst}'(L') = \overline{L}\}$$

Also ist subst′ ein Unifikator für

$$M = \{\overline{L_1}, \ldots, \overline{L_m}, L'_1, \ldots, L'_n\}$$

und damit sind M_1 und M_2 resolvierbar. Sei subst ein allgemeinster Unifikator für M.

Prädikatenlogik — Resolution

Beweis des Übertragungstheorems (3/3).

Wir betrachten den prädikatenlogischen Resolventen

$$R = \left\{\mathrm{subst}(L') \mid L' \in \left(N_1 \setminus \{L_1, \ldots, L_m\}\right) \cup \left(N_2 \setminus \{L'_1, \ldots, L'_n\}\right)\right\}$$

Da subst ein allgemeinster Unifikator für M und subst' ein Unifikator für M ist, gibt es eine Substitution subst'', so dass $\mathrm{subst}' = \mathrm{subst}\,\mathrm{subst}''$. Weiterhin gilt

$$\begin{aligned}
R' &= \left(M'_1 \setminus \{L\}\right) \cup \left(M'_2 \setminus \{\overline{L}\}\right) \\
&= \left(\{\mathrm{subst}'(L') \mid L' \in N_1\} \setminus \{\mathrm{subst}'(L_1), \ldots, \mathrm{subst}'(L_m)\}\right) \\
&\quad\; \cup \left(\{\mathrm{subst}'(L') \mid L' \in N_2\} \setminus \{\mathrm{subst}'(L'_1), \ldots, \mathrm{subst}'(L'_n)\}\right) \\
&= \left\{\mathrm{subst}'(L') \mid L' \in \left(N_1 \setminus \{L_1, \ldots, L_m\}\right) \cup \left(N_2 \setminus \{L'_1, \ldots, L'_n\}\right)\right\} \\
&= \left\{\mathrm{subst}''(L') \mid L' \in R\right\}
\end{aligned}$$

womit R' eine Grundinstanz von R ist. $\square$

Beweise für Resolution

Prädikatenlogik — Resolution

Offene 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

Definition

Sei $F \in \mathcal{F}$ eine Formel mit $\mathrm{FV}(F) = \{x_{i_1}, \ldots, x_{i_n}\}$. Dann sei

$$\forall F = \forall x_{i_1} \cdots \forall x_{i_n} F$$

der Allabschluss von F

Theorem

Sei F eine Aussage in konjunktiver Normalform mit $F = \forall x_{i_1} \cdots \forall x_{i_n} G$ und G quantorenfrei. Dann sind folgende Formeln äquivalent:

$$F \qquad \text{und} \qquad \forall G \qquad \text{und} \qquad \bigwedge_{D \in G} \forall D$$

Beweis.

Offensichtlich sind F und $\forall G$ äquivalent, denn $\mathrm{FV}(G) \subseteq \{x_{i_1}, \ldots, x_{i_n}\}$ und F kann nur zusätzlich Variablen binden, die überhaupt nicht in G vorkommen. Weiterhin ist G äquivalent zu $\bigwedge_{D \in G} D$, denn F ist in konjunktiver Normalform. Also sind auch $\forall G$ und $\bigwedge_{D \in G} \forall D$ äquivalent gemäß der klassischen Äquivalenz zwischen

$$\forall x(F_1 \wedge F_2) \qquad \text{und} \qquad \forall x F_1 \wedge \forall x F_2 \qquad \square$$

Beispiel

- Sei $F = \forall x \forall y \forall z \big(P(x) \land \neg Q(z)\big)$
- Dann ist $G = \big(P(x) \land \neg Q(z)\big)$
- $\forall G = \forall x \forall z \big(P(x) \land \neg Q(z)\big)$
  (und diese Formel ist äquivalent zu F)
- $\bigwedge_{D \in G} \forall D = \big(\forall x P(x) \land \forall z \neg Q(z)\big)$
  (auch diese Formel ist äquivalent zu F)

Notizen

- Allabschluss macht implizite Allquantifikation explizit
- technisches Hilfsmittel für folgende Beweise

Prädikatenlogik — Resolution

Theorem

Sei R ein Resolvent zweier Disjunktionsglieder $\{D_1, D_2\}$. Dann ist $(\forall D_1 \land \forall D_2) \to \forall R$ eine Tautologie. (d.h. $\forall R$ folgt aus $\forall D_1 \land \forall D_2$)

Beweis (1/2).

Sei $I = (U, \cdot^I)$ ein Modell für $\forall D_1 \land \forall D_2$; d.h. $(\forall D_1)^I = 1$ und $(\forall D_2)^I = 1$. Z.zg. I ist ein Modell für $\forall R$.

Seien subst_1 und subst_2 die Variablenumbenennungen,

$$D_1' = \{\text{subst}_1(L) \mid L \in D_1\} \qquad \text{und} \qquad D_2' = \{\text{subst}_2(L) \mid L \in D_2\},$$

$L_1, \ldots, L_m \in D_1'$ und $L_1', \ldots, L_n' \in D_2'$ die gewählten Literale und subst ein allg. Unifikator für $\{\overline{L_1}, \ldots, \overline{L_m}, L_1', \ldots, L_n'\}$, so dass

$$R = \left\{\text{subst}(L) \mid L \in \left(D_1' \setminus \{L_1, \ldots, L_m\}\right) \cup \left(D_2' \setminus \{L_1', \ldots, L_n'\right)\right\}$$

Sei $L = \text{subst}(\overline{L_1}) = \cdots \text{subst}(\overline{L_m}) = \text{subst}(L_1') = \cdots = \text{subst}(L_n')$.

Beweis (2/2).

Dann gilt

$$\big(\{\mathrm{subst}(L') \mid L' \in D_1'\} \setminus \{\overline{L}\}\big) \cup \big(\{\mathrm{subst}(L') \mid L' \in D_2'\} \setminus \{L\}\big) \subseteq R$$

Für einen Widerspruch nehmen wir an, es sei $(\forall R)^I = 0$. Sei $\mathrm{FV}(R) = \{x_{i_1}, \ldots, x_{i_\ell}\}$. Dann existieren $u_1, \ldots, u_\ell \in U$, so dass $R^{I[x_{i_1} \mapsto u_1]\cdots[x_{i_\ell} \mapsto u_\ell]} = 0$. Sei $I' = I[x_{i_1} \mapsto u_1]\cdots[x_{i_\ell} \mapsto u_\ell]$ diese Widerlegung.

Es folgt, dass sowohl $\mathrm{subst}(L')^{I'} = 0$ für alle $L' \in D_1' \setminus \{\overline{L}\}$ als auch $\mathrm{subst}(L')^{I'} = 0$ für alle $L' \in D_2' \setminus \{L\}$. Aus $(\forall D_1)^I = 1$ und $(\forall D_2)^I = 1$ folgen jedoch

$$\Big(\bigvee_{L' \in D_1'} \mathrm{subst}(L')\Big) = 1 \qquad \text{und} \qquad \Big(\bigvee_{L' \in D_2'} \mathrm{subst}(L')\Big) = 1$$

Also muss das jeweils fehlende Literal $\overline{L}$ und L in I' wahr sein; d.h. $(\overline{L})^{I'} = L^{I'} = 1$. Dies ist offensichtlich widersprüchlich. $\square$

Prädikatenlogik — Resolution

Theorem (Korrektheit)

Sei F eine Aussage in konjunktiver Normalform.
Wenn per Resolution die leere Menge aus F herleitbar ist, dann ist F unerfüllbar

Beweis.

Sei $F = \forall x_{i_1} \cdots \forall x_{i_n} G$ mit G quantorenfrei. Aus dem vorherigen Theorem folgt, dass $\emptyset = \forall \emptyset$ eine Folgerung aus $\bigwedge_{D \in G} \forall D$ ist. (jeder Resolvent ist Folgerung vorheriger Resolventen oder Disjunktionsglieder von G; mit modus ponens ist daher auch die leere Menge eine Folgerung der Disjunktionsglieder von G) Da $\emptyset$ unerfüllbar ist, muss auch $\bigwedge_{D \in G} \forall D$ unerfüllbar sein. Dies ist jedoch äquivalent zu F vermittels der Hilfsaussage zum Allabschluss. Also ist auch F unerfüllbar. $\square$

Prädikatenlogik — Resolution

Theorem (Vollständigkeit)

Sei F eine <u>Aussage</u> in konjunktiver Normalform.
Wenn F unerfüllbar ist, dann ist aus F per Resolution $\emptyset$ herleitbar.

Beweis (1/2).

Sei $F = \forall x_{i_1} \cdots \forall x_{i_n} G$ mit G quantorenfrei die unerfüllbare Aussage. Gemäß Grundresolution existiert eine Folge von Disjunktionsgliedern $D'_1, \ldots, D'_k \subseteq \mathcal{L}$, so dass $D'_k = \emptyset$ und für jedes $1 \le j \le k$ gilt

- D'_j ist eine Grundinstanz eines Disjunktionsgliedes von G oder
  $(D'_j = D[x_{i_1} \mapsto t_1] \cdots [x_{i_n} \mapsto t_n]$ für $t_1, \ldots, t_n \in H(F)$, $D \in G)$
- D'_j ist Resolvent von $\{D'_\ell, D'_m\}$ mit $\ell < j$ und $m < j$.

Wir konstruieren nun eine Deduktion $(D_1, \ldots, D_k)$ aus diesen Disjunktionsgliedern $D'_1, \ldots, D'_k$, so dass D'_j eine Grundinstanz von D_j für alle $1 \le j \le k$ ist. Sei $1 \le j \le k$ und seien $(D_1, \ldots, D_{j-1})$ bereits konstruiert.

Beweis (2/2).

Wir unterscheiden zwei Situationen gemäß Herkunft von D'_j:

1. Falls D'_j eine Grundinstanz des Disjunktionsgliedes $D \in G$ ist, dann setzen wir $D_j = D$. Dies erfüllt offensichtlich die Anforderungen.
2. Sei D'_j Resolvent von $\{D'_\ell, D'_m\}$ mit $\ell < j$ und $m < j$. Aufgrund des Übertragungstheorems existiert ein prädikatenlogischer Resolvent R von $\{D_\ell, D_m\}$, so dass D'_j eine Grundinstanz von R ist. Wir setzen $D_j = R$ und erfüllen damit wieder die Anforderungen.

Da $D'_k = \emptyset$ nur eine Grundinstanz von $\emptyset$ ist, muss $D_k = \emptyset$ gelten. Damit haben wir die leere Menge hergeleitet. □

Prädikatenlogik — Resolution

Korollar

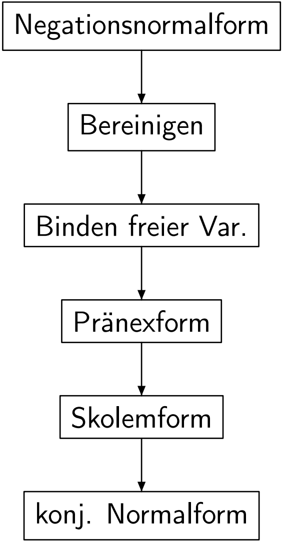
Prädikatenlogische Resolution ist korrekt und vollständig für Unerfüllbarkeit von Aussagen in konjunktiver Normalform.

Notizen

- immer noch problematisch für Implementation:
  - viele Wahlmöglichkeiten
    (Disjunktionsglieder, Literale)
  - viele nutzlose Resolventen (Ineffizienz)
  - sehr großer Suchraum
  - (sogar unendlich viele Resolventen bildbar)
- Heuristiken (Suchstrategien) notwendig
  (Verbot nutzloser Resolutionsschritte,
  Priorisierung bestimmter Resolutionsschritte)
- aber Vollständigkeit sollte erhalten werden
  (es dürfen keine notwendigen Schritte verboten werden)

Abschließendes Beispiel

Prädikatenlogik — Vollständiges Beispiel



Erhält

1. Äquivalenz (NNF)
2. Äquivalenz (Bereinigen)
3. Erfüllbarkeit (Binden)
4. Äquivalenz (Pränexform)
5. Erfüllbarkeit (Skolemform)
6. Äquivalenz (KNF per Distr.)
   Erfüllbarkeit (KNF per TSEITIN)

Prädikatenlogik — Vollständiges Beispiel

Beispiel

- Ausgangsformel:

$$\Big(Q(x) \lor \forall z P(x, g(z)) \lor \exists x \forall y \big(P(f(x), y) \land Q(a)\big)\Big)$$
$$\to \exists x \big(Q(x) \lor P(x, g(z))\big)$$

- wir möchten Allgemeingültigkeit (Tautologie) testen
- also negieren wir zunächst

$$\neg\Bigg(\Big(Q(x) \lor \forall z P(x, g(z)) \lor \exists x \forall y \big(P(f(x), y) \land Q(a)\big)\Big)$$
$$\to \exists x \big(Q(x) \lor P(x, g(z))\big)\Bigg)$$

Prädikatenlogik — Vollständiges Beispiel

Beispiel

- Unerfüllbarkeit dieser Formel:

$$\neg\Bigg(\Big(Q(x) \vee \forall z P(x, g(z)) \vee \exists x \forall y \big(P(f(x), y) \wedge Q(a)\big)\Big)$$

$$\rightarrow \exists x \big(Q(x) \vee P(x, g(z))\big)\Bigg)$$

- Negationsnormalform:

$$\Big(Q(x) \vee \forall z P(x, g(z)) \vee \exists x \forall y \big(P(f(x), y) \wedge Q(a)\big)\Big)$$

$$\wedge \forall x \big(\neg Q(x) \wedge \neg P(x, g(z))\big)$$

Prädikatenlogik — Vollständiges Beispiel

Beispiel

- Negationsnormalform:

$$\Big(Q(x) \vee \forall z P(x, g(z)) \vee \exists x \forall y \big(P(f(x), y) \wedge Q(a)\big)\Big)$$
$$\wedge \forall x \big(\neg Q(x) \wedge \neg P(x, g(z))\big)$$

- Bereinigung:

$$\Big(Q(x) \vee \forall z' P(x, g(z')) \vee \exists x' \forall y \big(P(f(x'), y) \wedge Q(a)\big)\Big)$$
$$\wedge \forall x'' \big(\neg Q(x'') \wedge \neg P(x'', g(z))\big)$$

- Binden freier Variablen:

$$\exists x \exists z \bigg(\Big(Q(x) \vee \forall z' P(x, g(z')) \vee \exists x' \forall y \big(P(f(x'), y) \wedge Q(a)\big)\Big)$$
$$\wedge \forall x'' \big(\neg Q(x'') \wedge \neg P(x'', g(z))\big)\bigg)$$

Prädikatenlogik — Vollständiges Beispiel

Beispiel

- Binden freier Variablen:

$$\exists x \exists z \Big(\big(Q(x) \lor \forall z' P(x, g(z')) \lor \exists x' \forall y \big(P(f(x'), y) \land Q(a) \big) \big) \land \forall x'' \big(\neg Q(x'') \land \neg P(x'', g(z)) \big) \Big)$$

- bereinigte Pränexform:

$$\exists x \exists z \forall z' \exists x' \forall y \forall x'' \Big(\big(Q(x) \lor P(x, g(z')) \lor \big(P(f(x'), y) \land Q(a) \big) \big) \land \neg Q(x'') \land \neg P(x'', g(z)) \Big)$$

Prädikatenlogik — Vollständiges Beispiel

Beispiel

- bereinigte Pränexform:

$$\exists x \exists z \forall z' \exists x' \forall y \forall x''$$
$$\Bigg(\Big(Q(x) \vee P(x, g(z')) \vee \big(P(f(x'), y) \wedge Q(a)\big)\Big) \wedge$$
$$\neg Q(x'') \wedge \neg P(x'', g(z))\Bigg)$$

- Skolemform:

$$\forall z' \forall y \forall x'' \Bigg(\Big(Q(b) \vee P(b, g(z')) \vee \big(P(f(h(z')), y) \wedge Q(a)\big)\Big) \wedge$$
$$\neg Q(x'') \wedge \neg P(x'', g(c))\Bigg)$$

Beispiel

- Skolemform:

$$\forall z' \forall y \forall x'' \Bigg(\Big(Q(b) \vee P(b, g(z')) \vee \big(P(f(h(z')), y) \wedge Q(a) \big) \Big) \wedge \neg Q(x'') \wedge \neg P(x'', g(c)) \Bigg)$$

- konjunktive Normalform:

$$\forall z' \forall y \forall x'' \Big(\big(Q(b) \vee P(b, g(z')) \vee P(f(h(z')), y) \big) \wedge \big(Q(b) \vee P(b, g(z')) \vee Q(a) \big) \wedge \neg Q(x'') \wedge \neg P(x'', g(c)) \Big)$$

Prädikatenlogik — Vollständiges Beispiel

Beispiel

- konjunktive Normalform:

$$\forall z' \forall y \forall x'' \Big(\big(Q(b) \vee P(b, g(z')) \vee P(f(h(z')), y)\big) \wedge \\ \big(Q(b) \vee P(b, g(z')) \vee Q(a)\big) \wedge \\ \neg Q(x'') \wedge \neg P(x'', g(c)) \Big)$$

- Mengendarstellung:

$$\Big\{ \big\{Q(b),\ P(b, g(z')),\ P(f(h(z')), y)\big\}, \\ \big\{Q(b),\ P(b, g(z')),\ Q(a)\big\}, \\ \big\{\neg Q(x'')\big\}, \big\{\neg P(x'', g(c))\big\} \Big\}$$

Prädikatenlogik — Vollständiges Beispiel

Beispiel

- Mengendarstellung:

$$\Big\{\{Q(b),\ P(b,g(z')),\ P(f(h(z')),y)\},$$
$$\{Q(b),\ P(b,g(z')),\ Q(a)\},$$
$$\{\neg Q(x'')\},\ \{\neg P(x'',g(c))\}\Big\}$$

- Resolution:

1	$\{\neg Q(x'')\}$	Element von F
2	$\{Q(b), P(b,g(z')), Q(a)\}$	Element von F
3	$\{P(b,g(z)), Q(a)\}$	Resolvent von {1, 2}
4	$\{P(b,g(z'))\}$	Resolvent von {1, 3}
5	$\{\neg P(x'',g(c))\}$	Element von F
6	$\emptyset$	Resolvent von {4, 5}

- damit Unerfüllbarkeit nachgewiesen
- Ausgangsformel ist eine Tautologie

Überblick

Ende des prüfungsrelevanten Inhalts

Inhalt

Ausblick

Prolog

Ausblick — Prolog

Motivation

- deklarative Programmiersprache
  (d.h., man beschreibt die Lösung, nicht den Weg dahin)
- geeignet für Handhabung von Wissensbanken
- integrierter Suchmechanismus
- geeignet für viele Suchprobleme

Ausblick — Prolog

Fakten

```
vater(andreas,rainer).
vater(rainer,guenther).
vater(stefanie,karlheinz).

mutter(andreas,stefanie).
mutter(rainer,annemarie).
mutter(stefanie,edeltraut).
```

Bedeutung

- kleingeschriebene Wörter sind Relationen oder Funktionen
- Punkt essentiell
- `vater` zweistellige Relation, `andreas` Konstante, etc.
- `vater(andreas,rainer).` ist Fakt
  d.h. attestiert Wahrheit dieser Relation

Ausblick — Prolog

Prolog-Interpreter

- hier SWI-Prolog (verfügbar für viele Plattformen)
- Alternativen:
  - BProlog, Ciao, ECLiPSe, GNU Prolog, Jekejeke Prolog
  - Logic Programming Associates, Poplog Prolog, P#, Quintus
  - SICStus, Strawberry, tuProlog, XSB, YAP-Prolog

Sitzung

```
?- consult('fakten.pl').
% fakten.pl compiled 0.00 sec, 10 clauses
?- vater(andreas,rainer).                              true.
?- vater(rainer,X).                            X = guenther.
?- mutter(X,Y).                  X = andreas, Y = stefanie ;
                                  X = rainer, Y = annemarie ;
                                X = stefanie, Y = edeltraut.
```

Ausblick — Prolog

Notizen

- großgeschriebene Wörter sind also Variablen
- Prolog antwortet mit erfüllender Variablenbelegung
- Prolog bietet alle erfüllenden Belegungen an
- das Komma “,” steht für Konjunktion (und)
- das Semikolon “;” steht für Disjunktion (oder)

Regel

```
kind(X,Y) :- vater(X,Y); mutter(X,Y).
```

- zu lesen als
  $\forall x \forall y \big((\mathtt{vater}(x,y) \vee \mathtt{mutter}(x,y)) \rightarrow \mathtt{kind}(x,y)\big)$
- links steht also die Konsequenz; rechts die Voraussetzung
- “`:-`” steht für $\leftarrow$

Ausblick — Prolog

Sitzung

```
?- consult('fakten.pl').
% fakten.pl compiled 0.00 sec, 10 clauses
?- kind(X,andreas).                                  false.
?- kind(andreas,X).                             X = rainer ;
                                               X = stefanie.
?- kind(andreas,X), kind(X,Y).
                                X = rainer, Y = guenther ;
                               X = rainer, Y = annemarie ;
                             X = stefanie, Y = karlheinz ;
                              X = stefanie, Y = edeltraut.
```

Ausblick — Prolog

Kompliziertere Regel

```
enkel(X,Y) :- kind(X,Z), kind(Z,Y).
```

- zu lesen als
  $\forall x \forall y \big(\exists z(\mathtt{kind}(x,z) \wedge \mathtt{kind}(z,y)) \rightarrow \mathtt{enkel}(x,y)\big)$
- Variablen, die nicht in der linken Seite vorkommen, sind also existentiell quantifiziert

Sitzung

```
?- enkel(stefanie,X).                    false.
?- enkel(X,annemarie).            X = andreas ;
                                         false.
?- enkel(andreas,X).             X = guenther ;
                                X = annemarie ;
                                X = karlheinz ;
                                 X = edeltraut.
```

Ausblick — Prolog

Kompliziertere Regel

```
opa(X) :- enkel(_,X), vater(_,X).
```

- zu lesen als
  $\forall x\,(\exists z_1 \exists z_2 (\mathtt{enkel}(z_1, x) \wedge \mathtt{vater}(z_2, x)) \rightarrow \mathtt{opa}(x))$
- "_" ist anonyme Variable
  (Variable an deren Wert wir nicht interessiert sind)
- alle Vorkommen von anonymen Variablen bezeichnen verschiedene Variablen

Sitzung

```
?- opa(X).                                   X = guenther ;
                                             X = karlheinz ;
                                                    false.
```

Ausblick — Prolog

(fast) Unifikation

```
?- f(g(a),X) = f(Z,b).          X = b, Z = g(a).
?- f(g(Y),X) = f(Z,Y).          Y = X, Z = g(X).
?- f(g(Y),X) = f(X,Y).          Y = X, X = g(X).
?- f(g(a),X) = f(X,b).                    false.
```

- "=" ist (eine Variante des) Unifikationsoperators
- (ohne Überprüfung ob Variable im Ersetzungsterm vorkommt)

Ausblick — Prolog

Ausblick — Prolog

Kartenfärbung — Bundesstaaten von Kanada

- Gesucht Färbung der Bundesstaaten, so dass benachbarte Bundesstaaten unterschiedliche Farben haben
- Anzahl der Farben vorgegeben

Ausblick — Prolog

Implementierung

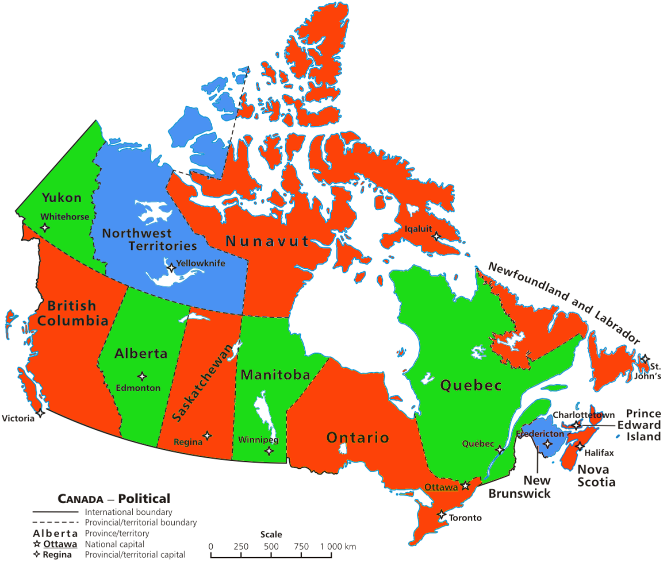
```
farbe(rot).
farbe(gruen).
farbe(blau).

nachbar(X,Y) :- X \= Y.

kanada(ON,QC,NS,NB,MB,BC,PE,SK,AB,NL,NT,YT,NU) :-
  farbe(ON), farbe(QC), farbe(NS), farbe(NB), farbe(MB),
  farbe(BC), farbe(PE), farbe(SK), farbe(AB), farbe(NL),
  farbe(NT), farbe(YT), farbe(NU),
  nachbar(ON,MB), nachbar(ON,QC), nachbar(QC,NB),
  nachbar(QC,NL), nachbar(NS,NB), nachbar(MB,SK),
  nachbar(MB,NU), nachbar(BC,YT), nachbar(BC,AB),
  nachbar(BC,NT), nachbar(SK,AB), nachbar(SK,NT),
  nachbar(AB,NT), nachbar(NT,YT), nachbar(NT,NU).
```

Ausblick — Prolog

Sitzung

```
?- consult('kanada.pl').
% kanada.pl compiled 0.00 sec, 1 clauses
?- kanada(ON,QC,NS,NB,MB,BC,PE,SK,AB,NL,NT,YT,NU).
                ON = NS = BC = PE = SK = NL = NU = rot,
                                   QC = MB = AB = YT = gruen,
                                                  NB = NT = blau ;
```

Ausblick — Prolog



Zusammenfassung

- Übertragungstheorem
- Beweise für prädikatenlogische Resolution
- abschließendes Beispiel
- Einführung Prolog

Vielen herzlichen Dank und viel Erfolg in der Prüfung.