TI1300: Redeneren en Logica

College 16: Resolutie en Prolog

Tomas Klos

Algoritmiek Groep

Voorbeeld

Wat is de Skolemnormaalvorm van

$$\exists x \forall y \exists u \forall v \forall w \exists z M(x, y, u, v, w, z)?$$

$$\exists x \forall y \exists u \forall v \forall w \exists z M(x, y, u, v, w, z)$$
$$\forall y \exists u \forall v \forall w \exists z M(a, y, u, v, w, z)$$
$$\forall y \forall v \forall w \exists z M(a, y, f(y), v, w, z)$$
$$\forall y \forall v \forall w M(a, y, f(y), v, w, g(y, v, w))$$

Voorbeeld

Zij F de formule

$$F = \exists x \forall y P(x, y) \rightarrow \exists z Q(z).$$

- prenexnormaalvorm(en)? 2 mogelijkheden:

$$F_2^{p,c} = \forall x \exists y \exists z (\neg P(x, y) \vee Q(z))$$
$$F_1^{p,c} = \forall x \exists y (\neg P(x, y) \vee Q(y))$$

- Skolemnormaalvorm(en)? ook 2 mogelijkheden:

$$F_1^s = \forall x (\neg P(x, f(x)) \vee Q(g(x)))$$
$$F_2^s = \forall x (\neg P(x, f(x)) \vee Q(f(x)))$$

Skolemnormaalvorm

Definitie (Skolemnormaalvorm)

Zij $F \in FORM$. Een Skolemnormaalvorm F^s van F is wat resulteert wanneer je

1. *F in prenexnormaalvorm brengt en de matrix in CNV schrijft*
2. *één voor één de existentiële kwantoren in de prenex skolemiseert.*

Een zin in Skolemnormaalvorm heeft de vorm

$$F = \forall x_1 \ldots \forall x_n (D_1 \wedge \ldots \wedge D_m),$$

waar elke D_i een disjunctie van literalen is (een clause of clausule). (Neem in elke clausule verschillende variabelen.)

We schrijven F als $\{D_1, \ldots, D_n\}$ (zonder kwantoren).

Atomen en Literalen

Definitie (Literaal)

Een literaal is een formule van de vorm A of $\neg A$ als A een atoom is.

Definitie (Atoom)

Een atoom is een formule van de vorm $P(t_1, \ldots, t_n)$ of $(t_1 = t_2)$.

Definitie (Complement van een literaal)

Zij L een literaal en A een atoom.

- *Als $L = A$, dan is $\overline{L} = \neg A$ het complement van L.*
- *Als $L = \neg A$ dan is $\overline{L} = A$ het complement van L.*

TU Delft

Resolutie in de predicatenlogica

Propositielogica:

$$\neg p \vee q, \neg q \vee r \vdash \neg p \vee r.$$

Predicatenlogica:

$$\forall x(M(x) \rightarrow S(x)), M(s) \vdash S(s)?$$

2 obstakels:

1. kwantoren: Skolemnormaalvorm
2. variabelen: unificatie

TU Delft

Resolutie

Resolutie zonder variabelen (dus met gronddisjuncties)

$D_1 = P(f(a))$
$D_2 = \neg P(f(a)) \vee Q(a)$
Resolvent van D_1 en D_2 naar $P(f(a))$ is ... $Q(a)$.

TU Delft

Resolutie met variabelen (2de obstakel)

Resolutie met variabelen: $\forall x(M(x) \rightarrow S(x)), M(s) \models S(s)$

$$D_1 = \neg M(x) \vee S(x)$$
$$D_2 = M(s)$$

Wat is de resolvent? Als we $[x := s]$ nemen? $S(s)$.

TU Delft

Resolutie met variabelen, ander voorbeeld

Resolutie met variabelen

$D_1 = P(x) \vee Q(x)$
$D_2 = \neg P(f(y)) \vee R(y)$
Resolvent van D_1 en D_2 is ...?
Met *gelijktijdige* substitutie $\sigma = [x := f(a), y := a]$:
$D_1\sigma = P(f(a)) \vee Q(f(a))$
$D_2\sigma = \neg P(f(a)) \vee R(a)$
Dan is de resolvent ... $Q(f(a)) \vee R(a)$.

Is er nog een substitutie?

Ja, de substitutie $\tau = [x := f(y)]$:
$D_1\tau = P(f(y) \vee Q(f(y))$
$D_2\sigma = \neg P(f(y)) \vee R(y)$
dan is de resolvent ... $Q(f(y)) \vee R(y)$.

Unificatie

Definitie (Unificator of 'unifier')

Een substitutie die twee literalen aan elkaar gelijk maakt heet unificator. Het zoeken naar zo'n substitutie heet unificatie.

Meest algemene unificator (mgu = most general unifier)

Beschouw:

- de substitutie $\sigma = [x := f(a), y := a]$
- de substitutie $\tau = [x := f(y)]$

Beide unificeren ze $P(x)$ en $P(f(y))$, τ legt minder vast: beter!

Een unifier θ is meest algemeen als er voor alle unifiers κ een substitutie λ bestaat zodat $\kappa = \theta\lambda$.

Resolutiemethode

Resolutie-algoritme

Om na te gaan of een zin F in Skolemnormaalvorm onvervulbaar is:

Stap 1: Genereer, *indien mogelijk*, een nieuwe resolvent R van disjuncties uit F; *anders* **Stop**: F is vervulbaar

Stap 2: *Als* $R = \square$ *dan* **Stop**: F is onvervulbaar, *anders*: $F \leftarrow F \cup \{R\}$ en ga naar 1.

Semi-beslissingsmethode:
Als F onvervulbaar is, stopt het algoritme, anders *soms* niet.

Voorbeeld

$$\forall x(M(x) \rightarrow S(x)), M(s) \models S(s)$$

1. Schrijf $F = P_1 \wedge P_2 \wedge \ldots \wedge \neg C$ in Skolemnormaalvorm
2. $F = \{\neg M(x) \vee S(x), M(s), \neg S(s)\}$
3. Probeer $\square$ af te leiden:

1	$\neg M(x) \vee S(x)$	element van F
2	$M(s)$	element van F
3	$\neg S(s)$	element van F
4	$S(s)$	resolvent (1, 2) met mgu $[x := s]$
5	$\square$	resolvent (3, 4)

Oefening

Bewijs met resolutie dat

1. $\forall x(A(x) \to B(x)) \vdash \exists x A(x) \to \exists x B(x)$.
2. $\forall x A(x) \to \forall x B(x) \vdash \exists x(A(x) \to B(x))$

Prolog: **Pro**gramming in **log**ic

Alain Colmerauer, Bob Kowalski (1972)

Niet algemene clauses, maar 'beperkt' tot Horn clauses:

Definitie (Horn clause)

Een disjunctie met maximaal één positieve literaal is een Horn clause.

- *Een clause met één positieve literaal is een program clause.*
- *Een clause met géén positieve literaal is een goal clause.*

Een clause met geen negatieve literalen heet een feit.

$\forall x(M(x) \to S(x)), M(s) \models S(s)$?

premissen: $\neg M(x) \vee S(x)$
$M(s)$

conclusie: $S(s)$.

vraag: Is $\{\neg M(x) \vee S(x), M(s), \neg S(s)\}$ vervulbaar?

Voorbeeld

program clause ('rule'): $\neg M(x) \vee S(x) \equiv M(x) \to S(x)$.
In Prolog: `s(X):-m(X).` ("$S(x)$, als $M(x)$")
`s(X)` is de head, `m(X)` is de body.
Ook: `p(X):-q(X),r(X).` $(P(x) \vee \neg Q(x) \vee \neg R(x))$

feit: `m(S).`

goal clause: `:-s(S).`

Voordelen van Horn clauses

- Je kunt er evenveel mee als met elke andere (programmeer-)taal.
- Resolutie werkt veel efficiënter dan met algemene clauses.
- Het geeft intuïtieve programma's.
- Horn clauses zijn gesloten onder resolutie.

Intuïtieve programma's

$$
\begin{aligned}
\text{Eén positieve literaal (rule):} \quad & \neg A_1 \vee \ldots \vee \neg A_m \vee A_n \\
\equiv \quad & \neg(A_1 \wedge \ldots \wedge A_m) \vee A_n \\
\equiv \quad & (A_1 \wedge \ldots \wedge A_m) \rightarrow A_n \\
\text{Geen negatieve literaal (feit):} \quad & A \\
\text{Geen positieve literaal (goal clause):} \quad & \neg A_1 \vee \ldots \vee \neg A_m \\
\equiv \quad & \neg(A_1 \wedge \ldots \wedge A_m)
\end{aligned}
$$

TU Delft

Werkwijze Prolog

1. Vul een 'programma' (knowledge base) met kennis:
   - feiten
   - regels
2. Laad de knowledge base in Prolog
3. Stel vragen ('negatie van de conclusie')

Prolog zoekt substituties en 'probeert $\square$ af te leiden.'

TU Delft

Voorbeeld van resolutie in Prolog

Predicatenlogica:

formules: $(B \wedge C) \rightarrow A \equiv \neg(B \wedge C) \vee A \equiv$

1. $\neg B \vee \neg C \vee A$
2. $\neg D \vee B$
3. $\neg E \vee C$
4. D
5. E

$\neg$-conclusie: $\neg A$

Prolog:

rules:
```
A:-B,C.

B:-D.
C:-E.
```

facts:
```
D.
E.
```

goal:
```
:-A.
```

Resolutie:

- Resolvent($\neg B \vee \neg C \vee A$ en $\neg A$) $= \neg B \vee \neg C$
- Resolvent($\neg B \vee \neg C$ en $\neg D \vee B$) $= \neg C \vee \neg D$
- Resolvent($\neg C \vee \neg D$ en D) $= \neg C$
- Resolvent($\neg C$ en $\neg E \vee C$) $= \neg E$
- Resolvent($\neg E$ en E) $= \square$.

TU Delft

Nog een voorbeeld

Voorbeeld met variabelen

```
f(a).
f(b).
g(a).
g(b).
h(b).
k(X):-f(X),g(X),h(X).
```

Vraag: (?- is de prompt)

```
?- k(Y).
```

TU Delft

Zelf doen?

- Vele varianten van Prolog zijn gratis beschikbaar
- `http://learnprolognow.org/` (uitgebreid: ook als boek beschikbaar)

Hartelijk bedankt voor jullie aandacht

en

succes met je tentamen!