§2 Podstawy Prologu

- ```
  p(a).
  p(x) :- q(x), p(a).
  q(5).
  ```

- $p(a)$

  $\forall x\,(q(x) \wedge p(a)) \rightarrow p(x)$

  $q(b)$

- $p(a) \wedge \forall x(q(x) \wedge p(a)) \rightarrow p(x) \wedge q(5)$

  $=: M$

```
?- p(5).
yes
```

$M \models p(5)$.

```
?- p(c)
no
```

$M \not\models p(c)$

1. Jakie formuły w ten sposób można opisać?
2. Jak sprawdzić $M \models g$?
```

2.1 Postaci normalne

(i) prenex form

$$Q_1 x_1 \dots Q_n x_n \; A(x_1, \dots x_n)$$

$Q_i \in \{\forall, \exists\}$, A bez kwantyfikatorów

Przykład:

$$(\neg \forall x\, p(x,y)) \wedge q(u) \wedge \neg(\forall z\, r(z) \vee s(w))$$

$\models\models 1$

$$\exists x \exists z (\neg p(x,y) \wedge q(u) \wedge \neg r(z) \wedge \neg s(w))$$

Twierdzenie:

Dla $F \in PL1$ istnieje $F' \in PL1_{prenex}$, taka że $F \models\models 1 F'$.

↳ ale: w Prologu nie mamy możliwości opisowania kwantyfikatora $\exists$.

↳ eliminacja $\exists$

Skolem

(ii) Clause form

$$\forall x_1, \ldots \forall x_m \, A(x_1, \ldots x_m)$$

$$A = C_1 \wedge \ldots \wedge C_k \qquad C_i : \text{clauses}$$

$$C_i \equiv L_{i_1} \vee \ldots \vee L_{i_{k_i}}$$

L_{ij} : literały, tzn.

$$L_{ij} \equiv p(x_1, \ldots x_m) \text{ lub } L_{ij} = \neg p(x_1, \ldots x_m)$$

Przykład:

$$F \equiv \forall x ((\forall y \, p(x,y)) \rightarrow (\neg \forall y (q(x,y) \rightarrow r(x,y))))$$

1. Eliminacja $\rightarrow$

$$\forall x \, (\neg \forall y \, p(x,y)) \vee (\neg \forall y (\neg q(x,y) \vee r(x,y)))$$

2. $\neg$ do środka

$$\forall x \, (\exists y \neg p(x,y)) \vee (\exists y (q(x,y) \wedge \neg r(x,y)))$$

3. Zmiana nazw zmiennych

$$\forall x\,(\exists y \neg p(x,y)) \vee (\exists z\,(q(x,z) \wedge \neg r(x,z)))$$

4. Eliminacja $\exists$ (Skolemizacja)

$$\forall x \neg p(x, f(x)) \vee (q(x, g(x)) \wedge \neg r(x, g(x)))$$

5. Koniunkcja dysjunkcji

$$\forall x\,(\neg p(x, f(x)) \vee q(x, g(x))) \wedge (\neg p(x, f(x)) \vee \neg r(x, g(x))) =: F'$$

Twierdzenie

Dla $F \in PL1$ istnieje $F' \in PL1_{\forall se}$, taka że F' jest spełnialna wtw. F jest spełnialna.

Zbiór klauzuli formuły F:

$$M_F = M_{F'} = \{\neg p(x, f(x)) \vee q(x, g(x)),\ \neg p(x, f(x)) \vee \neg r(x, g(x))\}$$

Teraz:

$F \models g$ wtw. $F \cup \{\neg g\}$ nie jest spełnialna

wtw. $M_{F \cup \{\neg g\}}$ nie jest spełnialna

czyli

dla programu w Prologu P mamy

?- g. wtw. $M_{P \cup \{\neg g\}}$ nie jest spełnialna.

yes

2.2 Systemy inferencji

zamiast $\models$, dedukcja na podstawie składni

do tego: aksjomaty i reguły

Przykład:

(i) $$\frac{A,\ A \to B}{B}$$

(ii) $$\frac{A \vee B}{A}$$

↳ krok po kroku "idziemy" od aksjomatów do twierdzeń.

Dowód dla P_n, to

ciąg $P_1, \dots P_n$ taki że

$\forall i \leq n$: P_i "=" Axiom lub

P_i "=" wynik zastosowania reguły

tzn. $\exists$ reguła $\frac{A_1 \dots A_m}{A}$:

P_i "=" A ∧

$\forall i = 1, \dots m \; \exists j < i : A_i$ "=" P_j

w tym przypadku P_n jest twierdzeniem (systemu)

i piszemy $\vdash P_n$.

$\vdash$ poprawny, jeśli $\vdash P \Rightarrow \models P$

$\vdash$ zupełny, jeśli $\models P \Rightarrow \vdash P$

Cel: system inferencji dla $M \models g$

wystarcie:

system inferencji dla $K_M \cup \{\neg g\}$ nie jest spełnialna.

↳ rezolucja

2.3 Rezolucja

obserwacje:

(i) $\{p, \neg p\}$ nie jest spełnialna

piszemy: $\square$

(ii) $\{p \vee q, \neg p \vee r\}$ jest spełnialna

$\Rightarrow$ $\{q, r\}$ jest spełnialna

Def: $k_1 = L_1 \vee \ldots \vee L_m$

$k_2 = L_1' \vee \ldots \vee L_{m'}'$

takie że $\exists_{j,j'}: L_{j'} \equiv \neg L_j$

Rezolwenta $Res(k_1, k_2)$, to

$$L_1 \vee \ldots L_{j-1} \vee L_{j+1} \vee \ldots \vee L_m \vee L_1' \vee \ldots \vee L_{j'-1}' \vee L_{j'+1}' \vee \ldots L_{m'}'$$

jako reguła:

$$\frac{L_1 \vee \ldots \vee L_m \qquad\qquad L_1' \vee \ldots \vee L_{m'}'}{L_1 \vee \ldots \vee L_{j-1} \vee L_{j+1} \vee \ldots \vee L_{m'}'}$$

jako inferencje:

$$M \cup \{k_1, k_2\} \underset{res}{\vdash} M \cup \{k_1, k_2, Res(k_1, k_2)\}$$

Twierdzenie

K nie jest spełnialna wtw.

$$\exists K' : K \underset{RS}{\vdash^*} K' \cup \{\square\}$$

Przykłady:

(i)
$$\frac{p \vee q \vee r \qquad s \vee \neg q}{p \vee r \vee s}$$

(ii) $K = \{\neg p \vee q, p, \neg q\}$

$$\dfrac{\dfrac{\neg p \vee q \qquad p}{q} \qquad \neg q}{\square}$$

czyli K nie można spełnić,
tzn. $\{\neg p \vee q, p\} \models q$

(iii) $(p \to q) \to r \;\models\; p \to (q \to r)$

$M \cup \{\neg q\} = \{(p \to q) \to r,\ \neg(p \to (q \to r))\}$

$K_{M \cup \{\neg q\}} = \{p \vee r,\ \neg q \vee r,$ [redacted] $p, q, \neg r\}$

$$\dfrac{\dfrac{\neg q \vee r \qquad \neg r}{\neg q} \qquad q}{\square}$$

(iv) $p \to (q \to r) \overset{?}{\models} (p \to q) \to r$

$K = \{\neg p \vee \neg q \vee r,\ \neg p \vee q,\ \neg r\}$

$$\dfrac{\dfrac{\neg p \vee \neg q \vee r \quad \neg r}{\neg p \vee \neg q} \quad \neg p \vee q}{\neg p}$$

↳ kontrprzykład: $f(p) = 0,\ f(r) = 0$

Zmienne:

$p(x_1) \vee q(x_2)$

$\neg p(a) \vee r(x_1, x_2)$

↳ $\left\{ \begin{array}{l} p(a) \vee q(b) \\ \neg p(a) \vee r(a,b) \end{array} \right\} \underset{RS}{\vdash} q(b) \vee r(a,b)$

b jest dowolna, ale jakie wybierać,

$b, c, f(a), \ldots$?

Def. Substitucja s

$s: Vars \to \overline{Term}$

taka że $|\{x \in Vars \mid s(x) \neq x\}| < \infty$

piszemy: $s = [x_1/a, x_2/f(b), \ldots x_n/c]$

Substytucja σ: Term $\to$ Term

$\sigma(a) = a$, a stała

$\sigma(x) = s(x)$, x zmienna

$\sigma(p(t_1, \ldots t_n)) = p(\sigma(t_1), \ldots \sigma(t_n))$

Def. Unifikator

Substytucja σ unifikuje t_1 i t_2 ($t_1 =_\sigma t_2$), jeżeli $\sigma(t_1) = \sigma(t_2)$.

σ jest najbardziej ogólnym unifikatorem dla t_1 i t_2 ($\sigma = mgu(t_1, t_2)$), jeżeli

(i) $t_1 =_\sigma t_2$

(ii) $\forall \sigma'$: $t_1 =_{\sigma'} t_2$: $\exists$ substytucja τ: $\sigma' = \tau \circ \sigma$

Przykład:

- $t_1 = p(a, y, z)$, $t_2 = p(x, b, c)$

  $\sigma = [x/a, y/b, z/c]$ unifikuje t_1 i t_2

  $\sigma' = [x/a, y/b]$ jest $mgu(t_1, t_2)$

  $\tau = [z/c] \rightsquigarrow \sigma = \tau \circ \sigma'$

- $f(x, g(a, b)) =_\sigma f(g(y, b), x)$

  dla $\sigma = [x / g(a, b)]$

Def: Robinson (1965)

$k_1 = L_1 \vee \dots \vee L_m$, $k_2 = L_1' \vee \dots \vee L_n'$

takie że

$$\exists j, j' : \quad L_j \equiv p(t_1, \dots t_k) \wedge$$
$$L_{j'}' \equiv \neg p(t_1', \dots t_k')$$

oraz $\sigma = mgu(p(t_1, \dots t_k), p(t_1', \dots t_k'))$

Resolwenta $Res(k_1, k_2)$ to

$\sigma(L_1 \vee \dots L_{j-1} \vee L_{j+1} \vee \dots L_m \vee L_1' \vee \dots L_{j'-1}' \vee L_{j'+1}' \vee \dots L_n')$

Przykład:

$k_1 = p(x_1) \vee q(x_2)$

$k_2 = \neg p(a) \vee r(x_1, x_2)$

$\sigma = mgu(p(x), p(a)) = [x/a]$

$\hookrightarrow Res(k_1, k_2) = q(x_2) \vee r(a, x_2)$

Algorytm Unifikacji

obliczenie mgu dla t_1 i t_2
na podstawie struktur termów

t_1 i t_2 unifikują się, jeżeli

- mają ten sam operator
- argumenty się unifikują, czyli
  - jeden z nich jest zmienną lub
  - są tą samą stałą lub
  - unifikują się rekurencyjnie

Przykłady:

(i) $t_1 = f(x, g(a,b))$

$t_2 = f(g(y,b), x)$

$x \quad g(y,b) \quad \rightsquigarrow \sigma(x) = g(y,b)$

$g(a,b) \quad x$

$=$

$g(y,b) \quad \rightsquigarrow \sigma(y) = a$

$\hookrightarrow \sigma = [x/g(a,b), y/a]$

(ii) $t_1 = f(g(x), a)$

$t_2 = f(g(y), b)$

$\sigma = mgu(t_1, t_2)$ nie istnieje

(iii) $t_1 = h(x), \quad t_2 = h(f(x))$

$\sigma(x) = f(x)$?

$\sigma(t_2) = h(f(f(x))) \neq h(f(x)) = \sigma(t_1)$

$\hookrightarrow$ occurs check

Algorytm unify

Input: $\Xi = \{u_1 = v_1, \ldots, u_n = v_n\}$

Output: σ = mgu Ξ, o ile σ istnieje

1. unify($\{\}$) = $\{\}$

2. unify($\{f(t_1, \ldots t_n) = g(t'_1, \ldots t'_m)\} \cup \Xi$) =
   if $f \neq g$
   then "σ nie istnieje"
   else unify($\{t_1 = t'_1, \ldots, t_n = t'_n\} \cup \Xi$)

3. unify($\Xi \cup \{x = t\}$) =
   unify($\Xi \cup \{t = x\}$) =
   if x wystąpi w temie t
   then "σ nie istnieje"
   else unify($\Xi[x/t]$) $\cup$ $\{[x/t]\}$

4. unify($\Xi \cup \{t = t\}$) = unify(Ξ)

Uwaga

Prolog nie wykonuje occurs check

unify_with_occurs_check/2

Przykład:

$$\text{unify}(\{f(x, g(a,b)) = f(g(y,b), x)\})$$

$$= \text{unify}(\{x = g(y,b),\ g(a,b) = x\})$$

$$= \text{unify}(\{g(a,b) = g(y,b)\}) \cup [x / g(y,b)]$$

$$= \text{unify}(\{a = y,\ b = b\}) \cup [x / g(y,b)]$$

$$= \text{unify}(\{a = y\}) \cup [x / g(y,b)]$$

$$= \text{unify}(\{\}) \cup [y / a, x / g(y,b)]$$

$$= [y / a, x / g(y,b)]$$

czyli $\sigma = [y/a,\ x/g(a,b)]$