Rezolucja w rachunku predykatów

Joanna Józefowska

Instytut Informatyki

Poznań, rok akademicki 2009/2010

Formuły ustalone

Term oraz atom nazywamy **ustalonym** wtw, gdy nie zawiera zmiennych.
Formuła jest **ustalona** wtw, gdy nie zawiera ani kwantyfikatorów, ani zmiennych.
Formułę A' nazywamy **ustaloną instancją formuły** A (która nie zawiera kwantyfikatorów), jeśli A' można otrzymać z formuły A przez podstawienie za zmienne formuły A termów ustalonych.
Klauzula C' jest **klauzulą ustaloną** wtw, gdy jest ustaloną instancją klauzuli C, czyli można ją otrzymać z C przez zastąpnienie wszystkich zmiennych w C termami ustalonymi.
Literał ustalony jest ustaloną instancją literału.

$p(a), f(5)$

$p(a) \vee q(b)$

$A = p(x) \wedge q(y)$
$A' = p(a) \wedge q(b)$

$C = r(x) \vee s(y)$
$C' = r(a) \vee s(b)$

$\neg p(a)$

Przedrostkowa koniunkcyjna postać normalna

Formuła jest w **koniunkcyjnej postaci normalnej** wtw, gdy jest koniunkcją alternatyw literałów.
Klauzula jest w **przedrostkowej koniunkcyjnej postaci normalnej** wtw, gdy jest postaci:

$$Q_1 x_1 Q_2 x_2 \dots Q_n x_n M$$

gdzie $Q_i, i = 1, \dots, n$ są kwantyfikatorami, a M jest formułą w koniunkcyjnej postaci normalnej, niezawierającą kwantyfikatorów.
Ciąg $Q_1 x_1 Q_2 x_2 \dots Q_n x_n$ jest nazywany **przedrostkiem** (prefiksem), a M **matrycą** formuły.

Postać klauzulowa formuł

Formuła zamknięta jest w postaci klauzulowej wtw, gdy jest w przedrostkowej koniunkcyjnej postaci normalnej i jej prefix zawiera wyłącznie kwantyfikatory uniwersalne.

$$\forall x_1 \forall x_2 \ldots \forall x_n M$$

Skolemizacja - sprowadzanie formuły do postaci klauzulowej

$$\exists x \forall y\; p(x,y) \rightarrow \forall y \exists x\; p(x,y)$$

Krok 1

Przemianuj zmienne kwantyfikowane w ten sposób, aby żadna zmienna nie występowała w dwóch kwantyfikatorach.

$$\exists x \forall y\; p(x,y) \rightarrow \forall w \exists z\; p(z,w)$$

Skolemizacja

$$\exists x \forall y\; p(x,y) \rightarrow \forall w \exists z\; p(z,w)$$

Krok 2

Usuń wszystkie binarne operatory logiczne oprócz $\vee$ oraz $\wedge$.

$$\neg \exists x \forall y\; p(x,y) \;\vee\; \forall w \exists z\; p(z,w)$$

Skolemizacja

$$\neg \exists x \forall y\; p(x,y) \;\vee\; \forall w \exists z\; p(z,w)$$

Krok 3

Przesuń operatory negacji do środka, usuwając podwójną negację, tak, aby negacja występowała tylko przy formułach atomowych. Skorzystaj z równoważnosci:
$\neg \forall x\; A(x) \equiv \exists x\; \neg A(x)$ oraz $\neg \exists x\; A(x) \equiv \forall x\; \neg A(x)$

$$\forall x \exists y\; \neg p(x,y) \;\vee\; \forall w \exists z\; p(z,w)$$

Skolemizacja

$\forall x \exists y \neg p(x,y) \ \vee \ \forall w \exists z \ p(z,w)$

Krok 4 (żadna zmienna nie występuje w dwóch kwantyfikatorach!)

Wydobądź kwantyfikatory z matrycy stosując równoważności:
$A \ op \ QxB(x) \equiv Qx(A \ op \ B(x))$
oraz
$QxA(x) \ op \ B \equiv Qx(A(x) \ op \ B)$
gdzie $op \in \{\vee, \wedge\}$.

$\forall w(\forall x \exists y \neg p(x,y) \ \vee \ \exists z \ p(z,w))$
$\forall w \exists z \forall x \exists y \ (\neg p(x,y) \ \vee \ \ p(z,w))$

Skolemizacja

$$\forall w \exists z \forall x \exists y \ (\neg p(x,y) \ \vee \ \ p(z,w))$$

Krok 5

Korzystając z praw rozdzielczości, przekształć matrycę formuły do koniunkcyjnej postaci normalnej.

$$\forall w \exists z \forall x \exists y \ (\neg p(x,y) \ \vee \ \ p(z,w))$$

Skolemizacja

$$\forall w \exists z \forall x \exists y \ (\neg p(x,y) \ \vee \ \ p(z,w))$$

Krok 6 - funkcje Skolema

- $\exists x$ - kwantyfikator egzystencjalny wystepujący w formule A,
- $y_1, ..., y_n$ zmienne kwantyfikowane uniwersalnie przed $\exists x$,
- f nowy symbol funkcyjny o arności n.

Usuń $\exists x$ i zastąp każde wystąpienie zmiennej x termem $f(y_1, ..., y_n)$. Jeśli żaden kwantyfikator uniwersalny nie poprzedza $\exists x$, to zastąp x nową stałą a (funkcją 0-argumentową).

$$\forall w \forall x \exists y \ (\neg p(x,y) \ \vee \ \ p(g(w),w))$$
$$\forall w \forall x \ (\neg p(x,f(w,x)) \ \vee \ \ p(g(w),w))$$

Notacja

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

$$(\neg p(x,f(w,x)) \ \vee \ \ p(g(w),w)) \wedge r(a)$$

$$\{\{\neg p(x,f(w,x)), p(g(w),w)\}, \{r(a)\}\}$$

Formuła jest zbiorem klauzul.
Klauzula jest zbiorem literałów.

Klauzula pusta i pusty zbiór klauzul

Oznaczenia

- Klauzulę pustą oznaczamy przez $\square$.
- Pusty zbiór klauzul oznaczamy $\emptyset$.

Lemat

- Klauzula pusta jest niespełnialna.
- Formuła pusta jest formułą prawdziwą.

... Adam i orzeszki

$$\frac{\neg\textbf{jedzenie(jabłko)} \vee \text{lubi(adam, jabłko)},\ \textbf{jedzenie(jabłko)}}{\text{lubi(adam, jabłko)}}$$

Reguła rezolucji dla klauzul ustalonych

- Niech C_1 i C_2 będą klauzulami ustalonymi, takimi, że $l \in C_1$ oraz $l^- \in C_2$.
- Klauzule C_1 i C_2 nazywamy **kolidującymi** i mówimy, że kolidują względem literałów komplementarnych l i l^-.
- **Rezolwentą** klauzul C_1 i C_2 nazywamy klauzulę C postaci $C = Rez(C_1, C_2) = (C_1 \setminus \{l\}) \cup (C_2 \setminus \{l^-\})$.
- Klauzule C_1 i C_2 nazywamy **klauzulami macierzystymi** dla C.

Przypomnienie

- **Klauzula ustalona** = klauzula bez kwantyfikatorów i zmiennych.
- **Klauzula** = alternatywa literałów.
- **Literał** = atom lub negacja atomu.
- **Atom** = p(lista termów).

Terminologia

$$\frac{\neg\textbf{jedzenie(jabłko)} \vee \textbf{lubi(adam, jabłko)},\ \textbf{jedzenie(jabłko)}}{\textbf{lubi(adam, jabłko)}}$$

Klauzule macierzyste
Literały kolidujące
Rezolwenta

Spełnialność rezolwenty klauzul spełnialnych

Twierdzenie

Rezolwenta klauzul C_1 i C_2 jest spełnialna wtedy i tylko wtedy, gdy klauzule C_1 i C_2 są (równocześnie) spełnialne.

Niech $\mathcal{I}$ będzie modelem klauzul C_1 i C_2.
Niech $l \in C_1$ i $l^- \in C_2$ będą literałami kolidującymi.
Jeżeli l jest spełniony w interpretacji $\mathcal{I}$, to l^- nie jest spełniony.
Zatem w klauzuli C_2 istnieje inny literał, który jest spełniony i występuje w klauzuli C, która jest w ten sposób spełniona.

$$C_1 = \{\neg p(a), r(a, f(a))\}$$

$$C_2 = \{p(a), q(a), s(f(a))\}$$

$$C = \{q(a), r(a, f(a)), s(f(a))\}$$

Spełnialność klauzul macierzystych klauzuli spełnialnej

Twierdzenie

Rezolwenta klauzul C_1 i C_2 jest spełnialna wtedy i tylko wtedy, gdy klauzule C_1 i C_2 są (równocześnie) spełnialne.

Niech $\mathcal{I}$ będzie modelem klauzuli C.
Niech $m \in C \cap C_1$ będzie literałem spełnionym w interpretacji $\mathcal{I}$.
Wtedy klauzula C_1 jest spełniona.
Z definicji rezolwenty istnieją klazule kolidujące: $l \in C_1$ i $l^- \in C_2$
Rozszerzmy interpretację $\mathcal{I}$ tak, że literał l nie jest spełniony w $\mathcal{I}$.
Wtedy l^- jest spełniony, a zatem również klauzula C_2 jest spełniona.

$$C_1 = \{\neg p(a), r(a, f(a))\}$$

$$C_2 = \{p(a), q(a), s(f(a))\}$$

$$C = \{q(a), r(a, f(a)), s(f(a))\}$$

Ogólna reguła rezolucji

- Niech C_1 i C_2 będą klauzulami nie mającymi wspólnych zmiennych.
- Niech $l_1 \in C_1$ i $l_2 \in C_2$ będą literałami takimi, że l_1 i l_2^- można uzgodnić, a σ niech będzie ich podstawieniem uzgadniającym (mgu).
- Klauzule C_1 i C_2 nazywamy kolidującymi i mówimy, że kolidują względem literałów l_1 i l_2.
- Rezolwentą klauzul C_1 i C_2 nazywamy klauzulę postaci: $Rez(C_1, C_2) = (C_1\sigma \setminus \{l_1\sigma\}) \cup (C_2\sigma \setminus \{l_2\sigma\})$

Znowu Adam i orzeszki

¬jedzenie(x_1) ∨ lubi(adam, x_1)

jedzenie(jabłko)

jedzenie(kurczak)

¬je(x_1, x_2) ∨ ¬żyje(x_1) ∨ jedzenie(x_2)

je(bogdan, orzeszki)

żyje(bogdan)

¬je(bogdan, x_2) ∨ je(zuzia, x_2)

$\sigma = \{x_1 \leftarrow \text{jabłko}\}$

$$\frac{\neg\textbf{jedzenie(jabłko)} \vee \text{lubi(adam, jabłko)},\ \textbf{jedzenie(jabłko)}}{\text{lubi(adam, jabłko)}}$$

Ogólna metoda rezolucji

- Niech $S_0 = S$.
- Załóżmy, że utworzyliśmy zbiór S_i.
- Wybierz klauzule kolidujące $C_1, C_2 \in S_i$ i niech $C = Rez(C_1, C_2)$.
- Jeśli C jest klauzulą pustą, to zakończ: zbiór S jest niespełnialny.
- W przeciwnym razie utwórz $S_{i+1} = S_i \cup \{C\}$.
- Jeśli $S_{i+1} = S_i$ dla wszystkich par literałów kolidujących, to zakończ wykonywanie: zbiór S jest spełnialny.

Czy ktoś jeszcze lubi orzeszki ?

¬jedzenie(x_1) ∨ lubi(adam, x_1)
jedzenie(jabłko)
jedzenie(kurczak)
¬je(x_2, x_3) ∨ ¬żyje(x_2) ∨ jedzenie(x_3)
je(bogdan, orzeszki)
żyje(bogdan)
¬je(bogdan, x_4) ∨ je(zuzia, x_4)
¬lubi(adam, orzeszki)
¬żyje(bogdan) ∨ jedzenie(orzeszki)
jedzenie(orzeszki)
lubi(adam, orzeszki)
□

Adam lubi orzeszki!

$\sigma_1 = \{x_2 \leftarrow \text{bogdan}, x_3 \leftarrow \text{orzeszki}\}$
$\sigma_2 = \{x_1 \leftarrow \text{orzeszki}\}$

Rezolucja - przykład

¬p(x_1) ∨ q(x_1) ∨ r(x_1, f(x_1))
¬p(x_2) ∨ q(x_2) ∨ s(f(x_2))
t(a)
p(a)
¬r(a, x_3) ∨ t(x_3)
¬t(x_4) ∨ ¬q(x_4)
¬t(x_5) ∨ ¬s(x_5)
¬q(a)
q(a) ∨ s(f(a))
s(f(a))
q(a) ∨ r(a, f(a))
r(a, f(a))
t(f(a))
¬s(f(a))
□

$\sigma_1 = \{x_4 \leftarrow a\}$
$\sigma_2 = \{x_2 \leftarrow a\}$
$\sigma_3 = \{x_1 \leftarrow a\}$
$\sigma_4 = \{x_3 \leftarrow f(a)\}$
$\sigma_5 = \{x_5 \leftarrow f(a)\}$
$\sigma = \{x_1 \leftarrow a,$
$x_2 \leftarrow a,$
$x_3 \leftarrow a,$
$x_4 \leftarrow f(a),$
$x_5 \leftarrow f(a)\}$

Lemat o podnoszeniu

Lemat

Niech C_1' i C_2' będą instancjami ustalonymi klauzul C_1 oraz C_2. Niech C' będzie rezolwentą ustaloną klauzul C_1' i C_2'. Wówczas istnieje rezolwenta C klauzul C_1 oraz C_2 taka, że C' jest instancją ustaloną C.

$$\begin{tikzcd}
C_1, C_2 \arrow[r, "\text{Rezolucja}"] \arrow[d, "\text{Instancja}"'] & C \arrow[d, "\text{Instancja}"] \\
C_1', C_2' \arrow[r, "\text{Rezolucja dla klauzul ustalonych}"'] & C'
\end{tikzcd}$$

Przykład

$$\begin{tikzcd} C_1, C_2 \arrow[r, "\text{Rezolucja}"] \arrow[d, "\text{Instancja}"] & {\color{red}C} \arrow[d, "\text{Instancja}"'] \\ C_1', C_2' \arrow[r, "\text{Rezolucja dla klauzul ustalonych}"'] & C' \end{tikzcd}$$

$C_1 = p(x) \vee q(x)$
$C_2 = \neg p(f(u)) \vee r(w)$

$\sigma_1 = \{x \leftarrow f(a), u \leftarrow a, w \leftarrow b\}$

$C_1' = p(f(a)) \vee q(f(a))$
$C_2' = \neg p(f(a)) \vee r(b)$

$C' = q(f(a)) \vee r(b)$

$\sigma_2 = \{x \leftarrow f(u)\}$

$C = q(f(u)) \vee r(w)$

$\sigma_3 = \{u \leftarrow a, w \leftarrow b\}$

Poprawność i pełność rezolucji

Poprawność

Jeżeli na podstawie ogólnej metody rezolucji można wyprowadzić klauzulę pustą, to zbiór klauzul jest niespełnialny.

Pełność

Jeśli zbiór klauzul jest niespełnialny, to stosując ogólną metodę rezolucji można wyprowadzić klauzulę pustą.

Przykładowe zadania

1. Wykazać spełnialność formuły rachunku predykatów metodą rezolucji.
2. Sprowadzić, podaną formułę rachunku predykatów do postaci klauzulowej.