

Programowanie deklaratywne

Zadanie 1:

Algorytm: Sprawdzanie tautologiczności formuł KRZ za pomocą rezolucji zdaniowej:

Dane: formuła A.

Wynik: Odpowiedź tak, jeżeli formułą A jest tautologią KRZ, odpowiedź nie w przeciwnym wypadku.

- 1) Bierzemy formułę B identyczną z formułą $(\neg A)$.
- 2) Sprowadzamy formułę B do KPN(B).
- 3) Formułę KPN(B) przedstawiamy w postaci klauzulowej $\Sigma(KPN(B))$.
- 4) Szukamy derywacji klauzuli pustej ze zbioru $\Sigma(KPN(B))$.
- 5) Jeżeli tak (udało się otrzymać klauzulę pustą).
- 6) W innym wypadku odp. nie.

Przykłady:

I) **A: $(p \rightarrow (q \rightarrow r)) \rightarrow [(p \rightarrow q) \rightarrow (p \rightarrow r)]$**

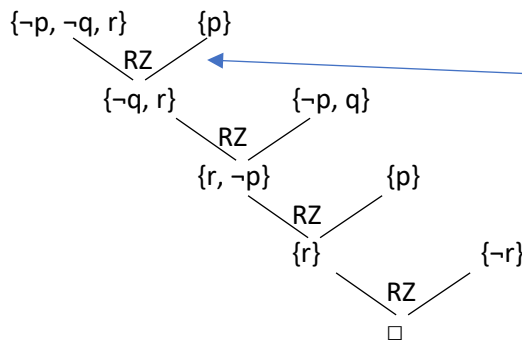
- 1) $\neg[(p \rightarrow (q \rightarrow r)) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r))]$
- 2) Sprowadzamy tę formułę do postaci KPN:
 $(p \rightarrow (q \rightarrow r)) \wedge \neg[(p \rightarrow q) \rightarrow (p \rightarrow r)]$
 $(\neg p \vee (q \rightarrow r)) \wedge (p \rightarrow q) \wedge \neg(p \rightarrow r)$
 $(\neg p \vee \neg q \vee r) \wedge (\neg p \vee q) \wedge p \wedge \neg r$ - KPN(B)
- 3) Przedstawiamy tę formułę w postaci klauzulowej
 $\Sigma(KPN(B)) = \{\{\neg p, \neg q, r\}, \{\neg p, q\}, \{p\}, \{\neg r\}\}$
- 4) Sprawdzamy czy ze zbioru $\Sigma(KPN(B))$ da się wyprowadzić klauzulę pustą.

Zastępujemy:

$(A \leftrightarrow B) \rightarrow (A \rightarrow B) \wedge (B \rightarrow A)$

$(A \rightarrow B) \rightarrow \neg A \vee B$

$\neg(A \rightarrow B) \rightarrow A \wedge \neg B$

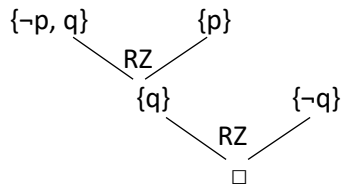


W jednej rezolucji zdaniowej (RZ) możemy zredukować tylko jedną parę zmiennych.

Odp. TAK

II) A: $(p \leftrightarrow q) \rightarrow (p \rightarrow q)$

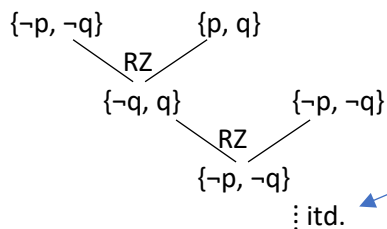
- 1) $\neg(p \leftrightarrow q) \rightarrow (p \rightarrow q)$
- 2) Sprowadzamy tę formułę do postaci KPN:
 $(p \leftrightarrow q) \wedge \neg(p \rightarrow q)$
 $(p \rightarrow q) \wedge (q \rightarrow p) \wedge \neg(p \rightarrow q)$
 $(\neg p \vee q) \wedge (\neg q \vee p) \wedge p \wedge \neg q$ - KPN(B)
- 3) Przedstawiamy tę formułę w postaci klauzulowej
 $\Sigma(\text{KPN}(B)) = \{\{\neg p, q\}, \{\neg q, p\}, \{p\}, \{\neg q\}\}$
- 4) Sprawdzamy czy ze zbioru $\Sigma(\text{KPN}(B))$ da się wyprowadzić klauzulę pustą.



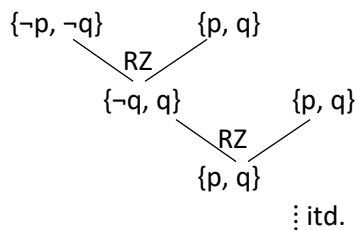
Odp. TAK

III) A: $(p \vee q) \rightarrow (p \wedge q)$

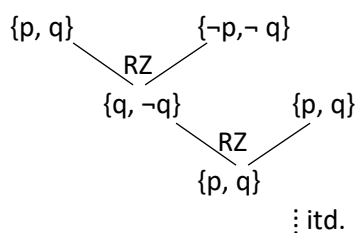
- 1) $\neg[(p \vee q) \rightarrow (p \wedge q)]$
- 2) Sprowadzamy tę formułę do postaci KPN:
 $(p \vee q) \wedge \neg(p \wedge q)$
 $(p \vee q) \wedge (\neg p \vee \neg q)$ - KPN(B)
- 3) Przedstawiamy tę formułę w postaci klauzulowej
 $\Sigma(\text{KPN}(B)) = \{\{p, q\}, \{\neg p, \neg q\}\}$
- 4) Sprawdzamy czy ze zbioru $\Sigma(\text{KPN}(B))$ da się wyprowadzić klauzulę pustą.

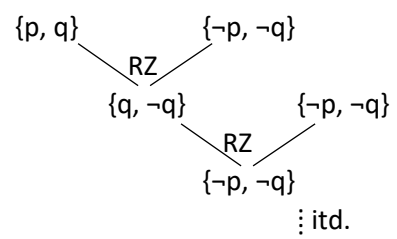


Nieważne jakie podstawienia zastosujemy nigdy nie dojdziemy do rozwiązania.



Na egzaminie trzeba rozpisać wszystkie sprawdzenia, żeby pokazać, że nie dojdzie się do klauzuli pustej (□)





Odp: NIE

Zadanie 2:

Algorytm: Sprawdzanie tautologii formuł LPR z wykorzystaniem rezolucji zdaniowej:

Dane: formuła A.

Wynik: Odpowiedź tak jeżeli A jest tautologią, odpowiedź nie w innym przypadku.

- 1) Niech B będzie formułą $\text{PNF}(\neg A)$.
- 2) Znajdujemy formułę $\text{SKOL}(B)$.
- 3) W formule $\text{SKOL}(B)$ opuszczamy kwantyfikatory i jeżeli nie jest to postać KPN, to doprowadzamy ją do tej postaci, otrzymując formułę C.
- 4) Formułę C przedstawiamy w postaci klauzuli $\Sigma(C)$.
- 5) Szukamy derywacji klauzuli pustej ze zbioru $\text{gr}(\Sigma(C))$.
- 6) Jeżeli klauzula pusta została otrzymana: odpowiedź tak.
- 7) Jeżeli algorytm się zatrzymał ze względu na brak możliwości stosowania reguły rezolucji: odpowiedź nie.

Przykłady:

I) **A: $\exists x(P(x) \vee Q(x)) \rightarrow (\exists x P(x) \vee \exists x Q(x))$**

1) $\neg [\exists x(P(x) \vee Q(x)) \rightarrow (\exists x P(x) \vee \exists x Q(x))]$

Sprawdzamy tę formułę do postaci PNF:

$$\exists x(P(x) \vee Q(x)) \wedge \neg (\exists x P(x) \vee \exists x Q(x))$$

$$\exists x(P(x) \vee Q(x)) \wedge \neg \exists x P(x) \wedge \neg \exists x Q(x)$$

$$\exists x(P(x) \vee Q(x)) \wedge \forall x \neg P(x) \wedge \forall x \neg Q(x)$$

$$\exists x(P(x) \vee Q(x)) \wedge \forall x (\neg P(x) \wedge \neg Q(x))$$

$$\exists x(P(x) \vee Q(x)) \wedge \forall y (\neg P(y) \wedge \neg Q(y))$$

$$\exists x[(P(x) \vee Q(x)) \wedge \forall y (\neg P(y) \wedge \neg Q(y))]$$

$$\exists x \forall y [(P(x) \vee Q(x)) \wedge \neg P(y) \wedge \neg Q(y)] - \text{PNF}(\neg A)$$

2) Sprawdzamy tę formułę do postaci Skolema:

$$\forall y [(P(a) \vee Q(a)) \wedge \neg P(y) \wedge \neg Q(y)] - \text{SKOL}(\text{PNF}(\neg A))$$

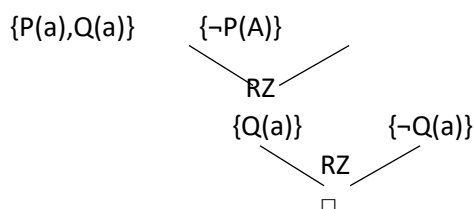
3) $C := (P(a) \vee Q(a)) \wedge \neg P(y) \wedge \neg Q(y)$

4) Przedstawiamy tę formułę w postaci klauzulowej

$$\Sigma(C) = \{\{P(a), Q(a)\}, \{\neg P(y)\}, \{\neg Q(y)\}\}$$

5) Sprawdzamy czy ze zbioru $\text{gr}(\Sigma(C))$ da się wyprowadzić klauzulę pustą:

$$\text{gr}(\Sigma(C)) = \{\{P(a), Q(a)\}, \{\neg P(a)\}, \{\neg Q(a)\} \dots\}$$



Odp. TAK

Zastępujemy

$$\neg \exists x P(x) \rightarrow \forall x \neg P(x)$$

$$\neg \forall x P(x) \rightarrow \exists x \neg P(x)$$

Wyciągamy $\exists x$ i $\forall x$ przed nawias. Najpierw $\exists x$ (jeżeli to możliwe)

Jeżeli $\exists x$ jest na początku to można je usunąć i zamienić x w $P(x)$ na stałą (a, b, c, \dots)

Oznacza, że poza podanymi wartościami $P(y)$ (u nas a) są jeszcze inne.

II) **A: $\neg \forall x \exists y P(x,y) \rightarrow \exists x \forall y \neg P(x,y)$**

1) $\neg [\neg \forall x \exists y P(x,y) \rightarrow \exists x \forall y \neg P(x,y)]$

Sprowadzamy tę formułę do postaci PNF:

$$\neg \forall x \exists y P(x,y) \wedge \neg \exists x \forall y \neg P(x,y)$$

$$\exists x \forall y \neg P(x,y) \wedge \forall x \exists y \neg \neg P(x,y)$$

$$\exists x \forall y \neg P(x,y) \wedge \forall x \exists y P(x,y)$$

$$\exists x \forall y \neg P(x,y) \wedge \forall z \exists y P(z,y)$$

$$\exists x [\forall y \neg P(x,y) \wedge \forall z \exists y P(z,y)]$$

$$\exists x [\forall u \neg P(x,u) \wedge \forall z \exists y P(z,y)]$$

$$\exists x \exists y [\forall u \neg P(x,u) \wedge \forall z P(z,y)]$$

$$\exists x \exists y \forall u [\neg P(x,u) \wedge \forall z P(z,y)]$$

$$\exists x \exists y \forall u \forall z [\neg P(x,u) \wedge P(z,y)] \text{ -- PNF}(\neg A)$$

2) Sprowadzamy tę formułę do postaci Skolema:

$$\forall u \forall z [\neg P(a,u) \wedge P(z,b)] \text{ -- SKOL}(\text{PNF}(\neg A))$$

3) $C := \neg P(a,u) \wedge P(z,b)$

4) Przedstawiamy tę formułę w postaci klauzulowej

$$\Sigma(C) = \{\{\neg P(a,u)\}, \{P(z,b)\}\}$$

5) Sprawdzamy czy ze zbioru $\text{gr}(\Sigma(C))$ da się wyprowadzić klauzulę pustą:

$$\text{gr}(\Sigma(C)) = \{\{\neg P(a,b)\}, \{P(a,b)\}, \dots\}$$

$$\{\neg P(a,b)\} \quad \{P(a,b)\}$$

RZ

□

Odp. TAK

W dwóch $P(\dots)$ występuje zmienna x , więc żeby wyciągnąć $\exists x$ przed nawias w jednej należy ją zamienić na dowolną inną (u nas z).