Programowanie deklaratywne

Zadanie 1:

<u>**Algorytm:** Sprawdzanie tautologiczności formuł KRZ za pomocą rezolucji zdaniowej:</u>

<u>Dane:</u> formuła A.

<u>Wynik:</u> 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 Σ(KPN(B)).
4) Szukamy derywacji klauzuli pustej ze zbioru Σ(KPN(B)).
5) Jeżeli tak (udało się otrzymać klauzulę pustą).
6) W innym wypadku odp. nie.

Zastępujemy:
$(A \leftrightarrow B) \text{ -> } (A \rightarrow B) \wedge (B \rightarrow A)$
$(A \rightarrow B) \text{ -> } \neg A \vee B$
$\neg(A \rightarrow B) \text{ -> } A \wedge \neg B$

<u>Przykłady</u>:

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
   $\Sigma(\text{KPN}(B)) = \{\{\neg p, \neg q, r\}, \{\neg p, q\}, \{p\}, \{\neg r\}\}$
4) Sprawdzamy czy ze zbioru Σ(KPN(B)) da się wyprowadzić klauzulę pustą.

$\{\neg p, \neg q, r\}$ $\{p\}$
RZ
$\{\neg q, r\}$ $\{\neg p, q\}$
RZ
$\{r, \neg p\}$ $\{p\}$
RZ
$\{r\}$ $\{\neg r\}$
RZ
$\square$

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

Odp.TAK

II) **A: (p↔)→(p → q)**

1) ¬(p↔)→(p → q)
2) Sprowadzamy tę formułę do postaci KPN:
   (p ↔ q)∧¬(p → q)
   (p → q)∧(q → p) ∧¬(p → q)
   (¬p v q) ∧ (¬q v p) ∧ p ∧ ¬ q - KPN(B)
3) Przedstawiamy tę formułę w postaci klauzulowej
   Σ(KPN(B)) = {{¬p, q} , {¬q, p}, {p}, {¬ q}}
4) Sprawdzamy czy ze zbioru Σ(KPN(B)) da się wyprowadzić klauzulę pustą.

{¬p, q} {p}
RZ
{q} {¬q}
RZ
□

Odp.TAK

III) **A: (p v q)→(p ∧ q)**

1) ¬[(p v q)→(p ∧ q)]
2) Sprowadzamy tę formułę do postaci KPN:
   (p v q)∧ ¬(p ∧ q)
   (p v q) ∧ (¬p v ¬q) - KPN(B)
3) Przedstawiamy tę formułę w postaci klauzulowej
   Σ(KPN(B)) = {{p, q} , {¬p, ¬q}}
4) Sprawdzamy czy ze zbioru Σ(KPN(B)) da się wyprowadzić klauzulę pustą.

{¬p, ¬q} {p, q}
RZ
{¬q, q} {¬p, ¬q}
RZ
{¬p, ¬q}
⋮ itd.

Nieważne jakie podstawienia zastosujemy nigdy nie dojdziemy do rozwiązania.

{¬p, ¬q} {p, q}
RZ
{¬q, q} {p, q}
RZ
{p, q}
⋮ itd.

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

{p, q} {¬p,¬ q}
RZ
{q, ¬q} {p, q}
RZ
{p, q}
⋮ itd.

{p, q}   {¬p, ¬q}

RZ

{q, ¬q}   {¬p, ¬q}

RZ

{¬p, ¬q}

⋮ itd.

Odp: NIE

Zadanie 2:

<u>**Algorytm**: Sprawdzanie tautologii formuł LPR z wykorzystaniem rezolucji zdaniowej:</u>

<u>Dane:</u> formuła A.

<u>Wynik:</u> Odpowiedź tak jeżeli A jest tautologią, odpowiedź nie w innym przypadku.

1) Niech B będzie formułą PNF(¬A).
2) Znajdujemy formułę SKOL(B).
3) W formule 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 Σ(C).
5) Szukamy derywacji klauzuli pustej ze zbioru gr(Σ(C)).
6) Jeżeli klauzula pusta została otrzymana: odpowiedź tak.
7) Jeżeli algorytm się zatrzymał ze względu na brak możliwości stosowania reguły rezolucji: odpowiedź nie.

<u>Przykłady:</u>

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

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

   Sprowadzamy tę formułę do postaci PNF:

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

   Zastępujemy
   $\neg\exists x\, P(x) \to \forall x\, \neg P(x)$
   $\neg\forall x\, P(x) \to \exists x\, \neg P(x)$

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

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

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

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

   Wyciągamy ∃x i ∀x przed nawias. Najpierw ∃x (jeżeli to możliwe)

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

   $\exists x\, \forall y\, [(P(x) \vee Q(x)) \wedge \neg\, P(y) \wedge \neg Q(y)]$ – PNF(¬A)

2) Sprowadzamy tę formułę do postaci Skolema:

   $\forall y\, [(P(a) \vee Q(a)) \wedge \neg\, P(y) \wedge \neg Q(y)]$ – SKOL(PNF(¬A))

   Jeżeli ∃x jest na początku to można je usunąć i zamienić x w P(x) na stałą (a,b,c,…)

3) C:= $(P(a) \vee Q(a)) \wedge \neg\, P(y) \wedge \neg Q(y)$
4) Przedstawiamy tę formułę w postaci klauzulowej

   Σ(C) = {{P(a),Q(a)},{¬P(y)},{¬Qy)}}
5) Sprawdzamy czy ze zbioru gr(Σ(C)) da się wyprowadzić klauzulę pustą:

   gr(Σ(C)) = {{P(a),Q(a)},{¬P(a)},{¬Q(a)}$\boxed{\dots}$}

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

{P(a),Q(a)} {¬P(A)}
RZ
{Q(a)} {¬Q(a)}
RZ
□

Odp. TAK

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

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

2) Sprowadzamy tę formułę do postaci Skolema:
   $\forall u\ \forall z\ [\neg P(a,u) \wedge P(z,b)]$ – SKOL(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 gr(Σ(C)) da się wyprowadzić klauzulę pustą:
   $gr(\Sigma(C)) = \{\{\neg P(a,b)\},\{P(a,b)\}, \dots \}$

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

RZ

□

Odp. TAK