

Projekt LTM

Jerzy Łukaszewicz s444429

16 czerwca 2019

Język programowania

Program napisany jest w języku Python. Dzięki wektorowym zapisie danych można bez większego trudu prowadzić operacje logiczne, zgodnie z kolejnością ich wykonywania.

Problem

Zadaniem mojego programu jest doprowadzić formułę logiczną do postaci uproszczonej postaci APN, i sprawdzić czy dana formuła jest spełnialna tzn. czy istnieje takie wartościowanie, które spełnia tą formułę. W Postaci APN można to z łatwością wywnioskować gdyż formuła w APN jest niespełnialna wtedy i tylko wtedy, gdy w każdej składowej koniunkcji elementarnej występuje para przeciwnych literalów. W moim porgramie kiedy zwrócona forma APN będzie pusta, oznaczać to będzie, że dana formuła nie jest spełnialna bo w każdej składowej koniunkcji wystąpiły przeciwne literały.

Działanie algorytmu

Aby doprowadzić formułę do postaci APN należy wykonać następującą listę kroków:

1. Zamienić wszystkie wyrażenia równoważności na równoważne im formuły zapisane przy pomocy implikacji tzn.

$$(P \Leftrightarrow Q) \Leftrightarrow ((P \Rightarrow Q) \wedge (Q \Rightarrow P))$$

2. Zamienić wszystkie implikacje odpowiednio korzystając z prawa:

$$(P \Rightarrow Q) \Leftrightarrow (\neg P \vee Q)$$

3. Korzystając z praw De Morgana pozbyć się wszystkich negacji sprzed nawiasów.

$$\neg(P \wedge Q) \iff (\neg P \vee \neg Q), \neg(P \vee Q) \iff (\neg P \wedge \neg Q).$$

4. Następnie przy pomocy prawa rozdzielności koniunkcji względem alternatywy sprowadzamy formułę do postaci APN.

$$(P \wedge (Q \vee R)) \Leftrightarrow ((P \wedge Q) \vee (P \wedge R))$$

5. W ostatnim kroku upraszczamy formułę do minimalnej postaci APN. korzystając w większej mierze z praw:

$$P \Leftrightarrow (P \vee P)$$

$$P \Leftrightarrow (P \wedge P)$$

$$P \wedge 0 = 0$$

$$P \vee 0 = P$$

$$P \wedge 1 = P$$

$$P \vee 1 = 1$$

Jak poprawnie używać algorytmu!

Operatory

- negacja = neg
- koniuncja = and
- alternatywa = or
- implikacja = imp
- równoważność = iff

Symbole

Symbole należy zapisywać za TYLKO pomocą liter.

Np. P albo PP

Zapis w postaci "P1 albo 1" jest niepoprawny

Formułę należy zapisać jako wektor w języku Python, analogicznie jak w notacji polskiej

np.

$$['iff', ['or', 'A', 'B'], ['and', ['not', 'A'], 'C']]$$

oznacza taki zapis:

$$(P \vee Q) \Leftrightarrow (\neg P \wedge R)$$

Dane testowe oraz wyniki

- (a) • Wejście:

$$(P \vee Q) \Leftrightarrow (\neg P \wedge R)$$

- Wyjście:
Formuła jest spełnialna.
Uproszczona postać APN:

$$(\neg P \wedge \neg Q \wedge \neg R) \vee (\neg P \wedge Q \wedge R)$$

- (b) • Wejście:

$$(P \wedge Q \Rightarrow R) \Rightarrow (P \Rightarrow R)$$

- Wyjście:
Formuła jest spełnialna.
Uproszczona postać APN:

$$(P \wedge Q \wedge \neg R) \vee (\neg P) \vee (R)$$

- (c) • Wejście:

$$\neg P \wedge P$$

- Wyjście:
Formuła nie jest spełnialna.

- (d) • Wejście:

$$\neg(P \wedge (Q \vee R) \Rightarrow (P \wedge Q) \vee R)$$

- Wyjście:
Formuła nie jest spełnialna.

- (e) • Wejście:

$$(P \wedge (Q \vee R))$$

- Wyjście:
Formuła jest spełnialna.
Uproszczona postać APN:

$$(P \wedge Q) \vee (P \wedge R)$$

- (f) • Wejście:

$$(P \Rightarrow R) \Leftrightarrow (\neg P \wedge R)$$

- Wyjście:
Formuła jest spełnialna.
Uproszczona postać APN:

$$(P \wedge \neg R) \vee (\neg P \wedge R)$$

- (g) • Wejście: $(P \wedge Q \Rightarrow R) \Rightarrow (P \Rightarrow (Q \Rightarrow R))$

- Wyjście:
Formuła jest spełnialna.
Uproszczona postać APN:

$$(P \wedge Q \wedge \neg R) \vee \neg P \vee \neg Q \vee R$$

- (h) • Wejście: $\neg((P \Leftrightarrow Q) \vee (P \Rightarrow Q \vee P))$

- Wyjście:
Formuła nie jest spełnialna.

- (i) • Wejście: $(P \Leftrightarrow \neg R) \Rightarrow (P \wedge Q \Rightarrow R)$

- Wyjście:
Formuła jest spełnialna.
Uproszczona postać APN:

$$(P \wedge R) \vee (\neg R \wedge \neg P) \vee \neg P \vee \neg Q \vee R$$

- (j) • Wejście: $\neg(P \Rightarrow Q) \Leftrightarrow P \wedge \neg Q$

- Wyjście:
Formuła jest spełnialna.
Uproszczona postać APN:

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