

## Laboratorium 5

27.11.2014, 04.12.2014

### Zadanie 1.

Wczytaj plik *dpn.pl*. Zdefiniowana w nim procedura `remove` służy do usuwania danego elementu z listy. Przeanalizuj tę procedurę, a następnie:

- usuń **2** z listy [3,4,2,5,2,1,2],
- usuń **a** z listy [a,b,a,c,a,d,a,e,a],
- usuń **p** z listy [p,q,r,neg p,p,neg q],
- usuń **neg p** z listy [p,q,neg p,r,neg p],
- zastanów się, dlaczego po wypisaniu odpowiedzi Prolog nie zgłasza się od razu znakiem „?-”
- popraw procedurę `remove` wstawiając w odpowiednim miejscu predykat cięcia *cut*.
- jak usunąć „Warning: Singleton variables [X]”?

### Przypomnienie:

#### REGUŁY REDUKCJI

do postaci klauzulowej (KPN):

$$\left| \frac{\neg\neg Z}{Z} \quad \frac{\neg\top}{\perp} \quad \frac{\neg\perp}{\top} \quad \frac{\beta}{\beta_1 \quad \beta_2} \quad \frac{\alpha}{\alpha_1 \mid \alpha_2} \right|$$

dualnej klauzulowej (DPN):

$$\left| \frac{\neg\neg Z}{Z} \quad \frac{\neg\top}{\perp} \quad \frac{\neg\perp}{\top} \quad \frac{\alpha}{\alpha_1} \quad \frac{\beta}{\beta_1 \mid \beta_2} \right|$$

Aby sprowadzić formułę *A* do postaci klauzulowej (KPN):

```
begin
  Niech S oznacza  $\langle [A] \rangle$ ;
  while jakiś element S zawiera nie-literał do
    wybierz z S element D zawierający nie-literał;
    wybierz z D nie-literał N;
    zastosuj odpowiednią regułę redukcyjną do N;
    niech S oznacza nowo utworzoną formułę
  end
end
```

Aby sprowadzić formułę *A* do postaci dualnej klauzulowej (DPN):

```
begin
  Niech S oznacza  $\langle [A] \rangle$ ;
  while jakiś element S zawiera nie-literał do
    wybierz z S element C zawierający nie-literał;
    wybierz z C nie-literał N;
    zastosuj odpowiednią regułę redukcyjną do N;
    niech S oznacza nowo utworzoną formułę
  end
end
```

## Zadanie 2.

Zapytaj o postać dualną klauzulową (DPN) formuł z zajęć konwersatoryjnych:

- (a)  $p \vee \neg p$
- (b)  $p \wedge \neg p$
- (c)  $(p \rightarrow (\perp \rightarrow r)) \rightarrow ((p \rightarrow \perp) \rightarrow (p \rightarrow r))$
- (d)  $(p \downarrow q) \uparrow \neg(p \downarrow q)$
- (e)  $\neg((p \rightarrow (q \rightarrow r)) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r)))$
- (f)  $(p \leftarrow (q \uparrow r)) \downarrow \neg p$
- (g)  $(p \vee (r \rightarrow q)) \rightarrow (q \leftarrow r)$
- (h)  $(\perp \vee p) \rightarrow (\perp \wedge q)$

## Zadanie 3.

- 1) Zapytaj o postać dualną klauzulową formuł:
  - a)  $p \text{ and } q \text{ imp } p$
  - b)  $(p \text{ and } q) \text{ imp } p$Dlaczego otrzymujemy różne odpowiedzi? Sprawdź teraz komponenty tych formuł.
- 2) Nadaj spójnikowi implikacji priorytet niższy od koniunkcji i alternatywy i jeszcze raz wykonaj polecenia z punktu 1). Co zauważasz?

## Zadanie 4.

Mechanizm śledzenia (ang. tracing) umożliwia przejrzanie wykonywania kolejnych kroków przez program. Włączenie śledzenia w Prologu następuje po użyciu polecenia „trace”:  
?-trace.

Wyróżnia się cztery różne typy zdarzeń w czasie pracy programu:

- CALL – określa próbę spełnienia celu lub podcelu,
- EXIT – oznacza sytuację, gdy cel lub podcel został spełniony,
- REDO – określa ponowienie próby spełnienia celu lub podcelu (nawracanie),
- FAIL – oznacza, że próba spełnienia celu lub podcelu zawodzi.

Prześledź w Prologu poszukiwanie odpowiedzi na pytanie o dualną klauzulową postać wybranej formuły.

## Zadanie 5.

Uzupełnij program *kpn.pl* dodając odpowiednie klauzule (reguły) definiujące predykat *singlestep*. Jako wzór wykorzystaj klauzule definiujące *singlestep* w programie *dpn.pl* oraz podany wyżej algorytm generowania KPN. Zapytaj o postać klauzulową kilku wybranych formuł.