

PLF, Egzamin

18.06.2019

1. Przetłumacz poniższy opis na program w Prologu. Pamiętaj o konwencji dotyczącej małych liter i stałych w Prologu.

Jeśli klocek X stoi na kloku Y, to X jest powyżej Y. Jeśli klocek X stoi na kloku Y i klocek Y jest powyżej klocka Z, to klocek X jest powyżej klocka Z. Jeśli klocek X jest powyżej klocka Y, to Y jest szerszy niż X. bu jest klockiem. fu jest klockiem. tu jest klockiem. mu jest klockiem. bu stoi na fu. fu stoi na tu. bu jest powyżej mu.

Jak zapisać i czy program w Prologu pozwala udowodnić, że

- (a) bu stoi na mu.
 - (b) tu jest szerszy niż bu.
 - (c) mu jest szerszy niż bu.
 - (d) istnieje klocek szerszy niż fu.
2. Dany jest zbiór klauzul: $K_1 = \{P(x, f(x))\}$, $K_2 = \{\neg P(x, f(g(z))), H(x, g(z))\}$, $K_3 = \{\neg H(x, x), P(x, x)\}$, $K_4 = \{\neg P(g(c), x), ok(c)\}$.

Skonstruuj SLD-rezolucję klauzuli pustej dla tezy $ok(c)$ (czyli wywód klauzuli pustej z $\{\neg ok(c)\}$ w oparciu o klauzule K_1, \dots, K_4).

3. Napisz predykat w Prologu $rowne(L1, L2)$, który sprawdzi, czy listy L1 i L2 zawierają te same elementy. Możesz założyć, że elementy obu list są ukonkretnione (spełniają predykat $ground$). Dopuszczalne jest użycie predykatu $member(X, L)$.
4. Napisz program $polacz :: [a] \rightarrow [a] \rightarrow [[a]]$, który dla argumentów postaci $[x1, \dots, xn]$ i $[y1, \dots, yn]$ zwróci listę postaci $[[x1, y1], \dots, [xn, yn]]$. Jeśli argumenty nie są równej długości algorytm ma połączyć tyle par elementów ile jest na krótszej liście.
5. Napisz funkcję fun , która będzie miała typ

$fun :: ((a, b) \rightarrow c) \rightarrow (b \rightarrow a) \rightarrow b \rightarrow c$.

6. Jaki typ będą miały poniższe wyrażenia:

(a) `foldr (\ x y -> x + y) (0::Int),`

(b) `flip flip True,`

gdzie

`flip::(a->b->c) -> b->a->c,`

`foldr:: Foldable t => (a->b->b) -> b -> t a -> b.`