Egzamin z programowania
część II
wersja biała

We wszystkich zadaniach testowych obowiązuje następująca punktacja: 0 punktów za mniej niz dwie poprawne odpowiedzi, 2 punkty za dwie poprawne odpowiedzi, 5 puntów za trzy, 8 za cztery i 10 za wszystkie poprawne odpowiedzi.

Ocena za **cały** egzamin liczona jest zgodnie z zasadą: **dst** za 60 punktów i każde dadatkowe 20 punktów powoduje podniesienie oceny o pół stopnia. Ocenę **bdb** można zatem otrzymać za 140p.

Zadanie 1. (24p) Funkcje `until` oraz `move` zdefiniowane są następująco:

```
fun until f p x = if (p x) then x else until f p (f x);
fun move (x::xs,ys) = (xs,x::ys);
fun len (x::xs) = 1 + len xs |
    len [] = 0;
fun second (x,y) = y;
```

a) Podaj typy zdefiniowanych powyżej funkcji (**4p.**).
b) Napisz w SML-u funkcję `repeat n f x`, która oblicza $f^n(x)$, przy czym $f^0(x) = x$, a $f^{n+1}(x) = f(f^n(x))$ dla $n > 0$ (**5p.**). Jaki typ ma ta funkcja? (**2p.**).
c) Napisz (nie używając jawnie rekurencji) funkcję `reverse` odwracającą listę za pomocą funkcji `repeat` (**6p.**).
d) Napisz (nie używając jawnie rekurencji) funkcję `reverse` za pomocą funkcji `until` (**7p.**).

W punktach c) i d) możesz korzystać ze zdefiniowanych powyżej funkcji. W obu tych punktach możesz zdefiniować w sumie co najwyżej jedną funkcję pomocniczą.

Zadanie 2. (18p) Funkcje `foldleft` oraz `foldright` spełniają następujące specyfikacje:

$$\texttt{foldleft}(op\circ)(e,[x_1,x_2,\ldots,x_n]) = (\ldots((e\circ x_1)\circ x_2)\cdots\circ x_n)$$
$$foldright(op\circ)(e,[x_1,x_2,\ldots,x_n]) = (x_1\circ(x_2\circ\ldots(x_n\circ e)\ldots)$$

Dla obu tych funkcji podaj definicje i typ główny w SML-u (po 3 punkty za definicję i po 2 punkty za typ). Udowodnij (**8p.**) następujące twierdzenie:

```
foldright (op::)  (xs,ys) = xs @ ys
```

Zadanie 3. (13p) Predykat `maxlist(X,L)` zachodzi, gdy `X` jest największym elementem z zawierającej liczby listy `L`. Predykat `maxtree(X,T)` zachodzi, gdy `X` jest największym elementem w drzewie binarnym `T`[1].

a) Zaimplementuj w Prologu (**7p.**) ogonową wersję predykatu `maxlist`. (Musisz napisać pomocniczy predykat trójargumentowy i ...).
b) Zaimplementuj w Prologu (**6p.**) predykat `maxtree`.

Zadanie 4. (19p) Mamy następujący program prologowy:

```
member(X,[X|_]).
member(X,[_|Ys]):- member(X,Ys).

nonmember1(X,Ys):- \+ member(X,Ys).

connected(X,X).
connected(X,Y):- edge(X,Z), connected(Z,Y).
```

a) Napisz predykat `nonmember2` (wersja predykatu `nonmember1`) nie korzystający z negacji, lecz z różności (`X\=Y`).(**4p.**)

[1] Przypominam, że drzewo to albo liść (`lf`, albo term `bt(X,T1,T2)`, gdzie `T1` oraz `T2` to drrzewa.

b) Podaj przykład zapytania, w którym `nonmember1` oraz `nonmember2` dają te same wyniki oraz zapytania, w którym dają wyniki różne. Odpowiedź wyjaśnij.(**4p.**)
c) Graf skierowany zadany jest przez fakty `edge(X,Y)`, gdzie `X` i `Y` to stałe przedstawiające wierzchołki grafu. Podaj fakty definiujące graf oraz zapytanie postaci `?- connected(X,Y)`, gdzie `X` i `Y` to stałe, które powoduje niekończące się obliczenia.(**4p.**)
d) Dokończ następujący program (**7p.**):

```
connected2(X,Y):- conn(X,Y,[]).
```

by predykad `connected2` był prawdziwy, gdy w grafie jest połączenie pomiędzy `X` a `Y`. Ponadto powstały program nie powinien się zapętlać dla stałych argumentów `X` oraz `Y`.

Zadanie 5. (10p) Czy prawdziwe są następujące stwierdzenia dotyczące Prologa:
a) Po zamianie kolejności atomów `T1=T2, T3=T4` otrzymujemy program dający ten sam wynik.
b) Równanie postaci `X=T`, gdzie `X` jest zmienną, a `T` jest termem, ma zawsze rozwiązanie w dzedzinie termów skończonych.
c) Po dopisaniu cięcia `!` do poprawnego programu otrzymujemy zawsze poprawny program, który może szybciej dochodzić do odpowiedzi.
d) Predykat `append` zdefiniowany

```
append([],Xs,Xs).
append([X|Xs],Ys,[X|Zs]):- append(Xs,Ys,Zs).
```

prawdziwy jest jedynie wówczas, gdy jego trzema argumentami są listy (spełniające pewne dodatkowe warunki).
e) Więz `X #/= Y` z GNU-Prologa działa tak samo jak standardowe `X/=Y`, którego wykonanie zawieszono do momentu, w którym obie zmienne będą ustalone.

Zadanie 6. (10p) Dla każdej z poniższych funkcji podaj jej najbardziej ogólny typ lub powiedz, że funkcja nie ma typu
a) `f x y = x 1 y`
b) `f x y = (x y, y x)`
c) `f x y = (x::x)@y)`
d) `f x y = x@(y::x)`
e) `f x y = (x y) = 0`

Zadanie 7. (10p) Mamy następujące definicje w programie z C++.

```
class A { (...) };
class B public A { };
A p(...);
B r(...);
A* pa = new A(...);
B* pb = new B(...);
```

Czy prawdziwe są następujące stwierdzenia (dotyczące programów w C++):
a) Jeżeli wywołanie jakiejś metody jest poprawne dla obiektu z klasy `A`, to jest też poprawne dla obiektu z klasy `B`.
b) Podstawienie `pb=pa` jest poprawne.
c) Podstawienie `a=b` jest poprawne.
d) Podczas konstruowania obiektu z klasy `B` zawsze wykonuje się konstruktor klasy `A`.
e) Dzięki konstrukcji `template` można pisać programy, które po skompilowaniu dają mniejszy kod wynikowy.