

# Uniwersytet Wrocławski, Informatyka

Egzamin wstępny na studia drugiego stopnia — Programowanie

18 lutego 2015

Obie kartki należy podpisać na marginesie imieniem i nazwiskiem.

Każde zadanie jest warte 20 punktów.

Czas trwania egzaminu: 60 minut.

**Zadanie 1.** Zaprogramuj w Prologu taki predykat `czworkami(?)`, że `cel` czworkami (`L`) jest spełniony wówczas, gdy `L` unifikuje się z listą, której liczba elementów jest podzielna przez cztery. Nie wolno używać żadnych predykatów standardowych ani definiować własnych predykatów pomocniczych.

**Zadanie 2.** Liczby jedyńkowo-dwójkowe, to ciągi cyfr 1 i 2. Ciąg  $\langle c_1, \dots, c_k \rangle$ , gdzie  $c_i \in \{1, 2\}$ , reprezentuje liczbę

$$\sum_{i=1}^k c_i \cdot 2^{i-1}.$$

Reprezentacja jest jednoznaczna. Liczbę zero reprezentuje ciąg pusty, liczbę 13 — ciąg 122 (bo z powyższej definicji wynika, że najmniej znacząca cyfra znajduje się po lewej stronie, a  $1 \cdot 2^0 + 2 \cdot 2^1 + 2 \cdot 2^2 = 13$ ). Zaprogramuj taki predykat  $\text{sum}(+, +, ?)$ , że cel  $\text{sum}(N, M, S)$  jest spełniony wówczas, gdy  $S$  unifikuje się listą cyfr 1 i 2 będącą reprezentacją liczby będącej sumą liczb reprezentowanych przez argumenty  $N$  i  $M$ . Predykat powinien działać deterministycznie dla podanego trybu użycia. W przypadku gdy  $N$  i  $M$  nie są listami cyfr 1 i 2 działanie predykatu może być dowolne.

**Zadanie 3.** Algorytm sortowania list *Bottom-Up Mergesort* polega na podzieleniu sortowanej listy na listę list jednoelementowych, a następnie na iterowaniu scalania list parami tak długo, aż pozostanie pojedyncza lista. Np. dla listy  $[6,3,7,9,4,5,1,2]$  kolejne etapy sortowania wyglądają następująco:

```
[[6], [3], [7], [8], [4], [5], [1], [2], [9]]
[[3,6], [7,8], [4,5], [1,2], [9]]
[[3,6,7,8], [1,2,4,5], [9]]
[[1,2,3,4,5,6,7,8], [9]]
[[1,2,3,4,5,6,7,8,9]]
```

Zaprogramuj w Haskellu funkcje

```
merge :: Ord a => [a] -> [a] -> [a]
```

```
bum :: Ord a => [a] -> [a]
```

Pierwsza scala dwie posortowane listy (tak jak funkcja biblioteczna o tej samej nazwie z modułu `Data.List.Utils`), druga implementuje opisany wyżej algorytm sortowania. Możesz korzystać z funkcji zdefiniowanych w preludium standardowym (`map`, `$` itp.).

**Zadanie 4.** Podaj typy podanych wyrażeń w Haskellu (wpisz kreskę, jeśli podane wyrażenie nie posiada typu).

<code>map map</code>	
<code>map . map</code>	
<code>flip map</code>	
<code>flip . map</code>	
<code>map flip</code>	
<code>map . flip</code>	
<code>flip flip</code>	
<code>flip . flip</code>	

gdzie

```
map _ [] = []
```

```
map f (x:xs) = f x : map f xs
```

```
(x . y) z = x (y z)
```

```
flip x y z = x z y
```