

## Egzamin licencjacki/inżynierski — 5 września 2018

Z sześciu poniższych zestawów zadań (Matematyka I, Programowanie, Matematyka dyskretna, Algorytmy i struktury danych, Metody numeryczne, Matematyka II) należy wybrać i przedstawić na osobnych kartkach rozwiązanie trzech zestawów.

Za brakujące (do trzech) zestawy zostanie wystawiona ocena niedostateczna z urzędu. Egzamin uważa się za zaliczony, jeśli student rozwiąże z oceną dostateczną co najmniej 2 zestawy. Wtedy ocena z egzaminu jest średnią arytmetyczną ocen z trzech wybranych zestawów. Na rozwiązanie przeznaczona jest czas  $3 \times 40 = 120$  minut. Po wyjściu z sali egzaminacyjnej w czasie egzaminu nie ma możliwości powrotu do tej sali i kontynuowania pisania egzaminu.

### Matematyka I — Logika dla informatyków

Zbiór  $L(X)$  wszystkich skończonych list nad danym zbiorem  $X$  jest zdefiniowany indukcyjnie w następujący sposób:

- $\text{nil}$  jest skończoną listą nad zbiorem  $X$ ;
- jeśli  $x$  jest elementem zbioru  $X$  oraz  $xs$  jest skończoną listą nad zbiorem  $X$  to  $x : xs$  jest skończoną listą nad zbiorem  $X$ .

Dla wszystkich zbiorów  $X$  operacja konkatencji list  $++ : L(X) \times L(X) \rightarrow L(X)$  jest zdefiniowana w następujący sposób. Dla wszystkich elementów  $x \in X$  oraz wszystkich list  $xs, ys \in L(X)$  przyjmujemy

$$\begin{aligned}\text{nil} ++ ys &= ys, \\ (x : xs) ++ ys &= x : (xs ++ ys).\end{aligned}$$

Dla wszystkich zbiorów  $X$  definiujemy funkcję  $\text{length} : L(X) \rightarrow \mathbb{N}$  w następujący sposób. Dla dowolnego elementu  $x \in X$  oraz dowolnej listy  $xs \in L(X)$  przyjmujemy

$$\begin{aligned}\text{length}(\text{nil}) &= 0, \\ \text{length}(x : xs) &= \text{length}(xs) + 1.\end{aligned}$$

1. Sformułuj zasadę indukcji w takiej postaci, żeby można było jej użyć w dowodzie w punkcie 2.
2. Korzystając z zasady indukcji sformułowanej w punkcie 1 udowodnij indukcyjnie, że dla dowolnego zbioru  $X$  oraz dowolnych list  $xs, ys \in L(X)$  zachodzi równość

$$\text{length}(xs ++ ys) = \text{length}(xs) + \text{length}(ys).$$

**Uwaga:** To jest zadanie z logiki. Przy ocenianiu zwrócimy szczególną uwagę na poprawność i klarowność rozumowania, w szczególności na poprawność użytej zasady indukcji, odpowiednie sformułowanie i użycie wszystkich założeń, odpowiednie użycie kwantyfikatorów i nawiasów itp.

## Programowanie

Za zadanie można otrzymać 20 punktów. Aby otrzymać ocenę dostateczną, należy zdobyć 7 punktów, próg dla dst+ to 9p, dla db – 11p, dla db+ 13p, dla bdb – 15p.

**Zadanie 1.** Gramatyka  $G_1$  z symbolem startowym  $S$  nad alfabetem  $\{a, b, c\}$  dana jest za pomocą następującego zbioru produkcji:

$$\{S_1 \rightarrow aS_1a, S_1 \rightarrow bS_1b, S \rightarrow \varepsilon, S \rightarrow S_1S, S \rightarrow cS, \}$$

Gramatyka  $G_2$  z symbolem startowym  $S$  nad alfabetem  $\{a, b, c\}$  dana jest za pomocą następującego zbioru produkcji:

$$\{S \rightarrow aSa, S \rightarrow cSc, S \rightarrow \varepsilon, S \rightarrow a, S \rightarrow c, \}$$

Dla gramatyki  $G$  przez  $L(G)$  rozumieć będziemy język generowany przez  $G$ . Dla wyrażenia regularnego  $r$  przez  $\mathcal{L}(r)$  rozumiemy język opisany przez wyrażenie  $r$ .

- Czy *abbabb* należy do  $L(G_1)$ ? Odpowiedź uzasadnij. **(1)**
- Czy gramatyka  $G_1$  jest jednoznaczna? Odpowiedź krótko uzasadnij. **(2)**
- Przedstaw wyrażenie regularne lub gramatykę bezkontekstową generującą zbiór

$$A_1 = L(G_1) \cap L(G_2)$$

**(3).** Odpowiedź uzasadnij.

- Napisz w języku imperatywnym funkcję, która bierze jako wejście napis i zwraca wartość logiczną, równą *True* wtedy i tylko wtedy, gdy ten napis należy do zbioru  $L(G_1)$ . Możesz używać języka wybranego z następującej listy: Możesz używać języka wybranego z następującej listy: C, C++, Java, C#, Python, Ruby, Go, AWK, Rust. **(4)**

**Zadanie 2. (6p)** Napisz w Haskellu dwuargumentową funkcję `nth`, która dla listy i liczby  $n$  zwraca  $n$ -ty element listy wejściowej (3p). Napisz również funkcję `f2`, która nie wywołuje funkcji `nth` i zwraca te same wartości, co funkcja `f` zdefiniowana poniżej

$$f\ k\ xs = nth\ k\ (qsort\ xs)$$

**Zadanie 3. (4p)** Napisz w Prologu predykat `sorted/1`, który sprawdza, czy lista liczb jest posortowana (i wywoływany jest na listach w pełni ukonkretnionych). Twój predykat powinien:

- Korzystać z `append`, porównania liczb i negacji.
- Nie wykorzystywać rekurencji, ani żadnych innych predykatów.

## Matematyka dyskretna

Założmy, że chcemy umieścić 8 nieatakujących się wież na szachownicy. Na ile sposobów możemy to zrobić, jeśli 16 najbardziej "północno-zachodnich" pól musi być pustych? A co jeśli tylko 4 najbardziej "północno-zachodnie" pola muszą być puste?

## Algorytmy i struktury danych

Za rozwiązanie obydwu zadań z tej części można otrzymać w sumie do 9 punktów. Skala ocen: poniżej 3 punktów — ocena niedostateczna (egzamin niezdany), 3 punkty dają ocenę dostateczną, 4 — dostateczną z plusem, 5 — dobrą, 6 — dobrą z plusem, 7 albo więcej punktów daje ocenę bardzo dobrą.

### Zadanie 1: suma kwadratów (5 punktów)

Zadana jest liczba naturalna  $n > 1$ . Liczbę tę należy przedstawić w postaci *sumy kwadratów* liczb naturalnych. Suma ta ma się składać z minimalnej liczby składników.

- Opracuj i opisz efektywny algorytm rozwiązujący ten problem.
- Uzasadnij poprawność opisanego algorytmu; oszacuj jego złożoność obliczeniową (czasową i pamięciową).
- Wykaż, że strategia zachłanna nie zawsze daje optymalny wynik.

### Zadanie 2: słownik z operacjami *lessthan* i *greaterthan* (4 punkty)

Rozważmy słownik  $S$ , którego elementy pochodzą ze zbioru z porządkiem liniowym. Do standardowych operacji słownikowych *insert*( $x$ ), *delete*( $x$ ) i *search*( $x$ ) dodajemy jeszcze operację *lessthan*( $z$ ) określającą ile jest w słowniku elementów mniejszych niż  $z$  oraz operację *greaterthan*( $z$ ) określającą ile jest w słowniku elementów większych niż  $z$ :

$$\begin{aligned} S.\textit{lessthan}(z) &= |\{a \in S : a < z\}| \\ S.\textit{greaterthan}(z) &= |\{a \in S : a > z\}| \end{aligned}$$

Zaprojektuj taki słownik, w którym każda z wymienionych operacji ma pesymistyczną złożoność czasową  $O(\log n)$ , gdzie  $n = |S|$ . Procedury *lessthan*( $z$ ) i *greaterthan*( $z$ ) napisz w pseudokodzie i opisz ich działanie. Krótko ale precyzyjnie opisz, jak zmodyfikowałeś pozostałe procedury słownikowe.

## Metody numeryczne

Za rozwiązanie zadań można otrzymać łącznie 12 punktów. Otrzymanie 4 pkt. gwarantuje ocenę dostateczną, próg dla dst+ to 5.5 pkt., dla db – 7 pkt., dla db+ 9 pkt., a dla bdb – 11 pkt.

1. **4 punkty** Wyjaśnij, jakie znaczenie z punktu widzenia metod numerycznych ma pojęcie uwarunkowania zadania. Podaj odpowiedni przykład.
2. **4 punkty** Pomiary  $(t_k, c_k)$  ( $0 \leq k \leq N$ ;  $t_k > 0$ ,  $c_k > 1$ ) pewnej zależnej od czasu wielkości fizycznej  $C$  sugerują, że wyraża się ona wzorem

$$C(t) = 2^{(At^2+2018)^{-1}}.$$

Stosując aproksymację średniokwadratową, wyznacz prawdopodobną wartość parametru  $A$ .

3. **4 punkty** Niech dana będzie nieosobliwa macierz trójkątna górna  $U \in \mathbb{R}^{n \times n}$ . Zaproponuj algorytm obliczania macierzy  $U^{-1}$ . Podaj jego złożoność obliczeniową i pamięciową.

## Matematyka II — Algebra

Zadanie 1. (7 punktów)

Grupa  $G$  działa na zbiorze  $X$ . Stabilizatorem elementu  $x \in X$  nazywamy zbiór

$$S_x = \{g \in G \mid gx = x\}.$$

Wykazać, że  $S_x$  jest podgrupą grupy  $G$ .

Zadanie 2. (6 punktów)

Wektory  $x_1, \dots, x_n$  są liniowo niezależne. Niech  $ny_1 = x_1 + \dots + x_n$ ,  
 $y_k = x_k - y_1$  dla  $2 \leq k \leq n$ . Sprawdzić czy wektory  $y_k$  są liniowo niezależne.

Progi punktowe: 3, 5, 7, 9, 11 punktów.