

## **Egzamin licencjacki/inżynierski — 07 lutego 2020**

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 rozwiązania 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

Rozważmy niewielki fragment funkcyjnego języka programowania (dokładniej, jest to rachunek lambda z typami prostymi).

**Syntaktyka.** Język (zbiór termów, czyli programów w naszym języku) jest zadany gramatyką

$$t ::= c \mid x \mid \text{fn } x \Rightarrow t \mid t_1 t_2$$

gdzie  $c$  przebiega stałe całkowite,  $x$  przebiega zmienne,  $\text{fn } x \Rightarrow t$  oznacza funkcję przyjmującą parametr  $x$  i zwracającą wartość wyliczoną przez  $t$ , natomiast  $t_1 t_2$  jest aplikacją funkcji  $t_1$  do argumentu  $t_2$ . Przyjmujemy też standardowe konwencje dotyczące zmiennych wolnych i związanych (definicja funkcji  $\text{fn } x \Rightarrow t$  wiąże zmienną  $x$ ).

**System typów.** Typy zadane są gramatyką

$$\tau ::= \text{int} \mid \tau \rightarrow \tau$$

Będziemy używać notacji  $\Gamma \vdash t : \tau$  oznaczającej „w środowisku  $\Gamma$  term  $t$  ma typ  $\tau$ ”. Środowiska  $\Gamma$  przypisują tutaj zmiennym typy, tzn. są zbiorami par postaci  $x : \tau$ , gdzie  $x$  jest zmienną a  $\tau$  typem. Zakładamy przy tym, że zmienna może mieć przypisany co najwyżej jeden typ, w szczególności pisząc  $\Gamma \cup \{x : \tau\}$  milcząc zakładamy, że zmienna  $x$  nie występuje w środowisku  $\Gamma$ . Mamy następujące reguły typowania:

$$\frac{}{\Gamma \vdash c : \text{int}} \quad \frac{}{\Gamma \cup \{x : \tau\} \vdash x : \tau} \quad \frac{\Gamma \cup \{x : \tau_1\} \vdash t : \tau_2}{\Gamma \vdash \text{fn } x \Rightarrow t : \tau_1 \rightarrow \tau_2} \quad \frac{\Gamma \vdash t_1 : \tau_1 \rightarrow \tau_2 \quad \Gamma \vdash t_2 : \tau_1}{\Gamma \vdash t_1 t_2 : \tau_2}$$

Pierwsza reguła mówi, że przy dowolnych założeniach każda stała całkowita jest typu  $\text{int}$ . Druga, że przy założeniu  $x : \tau$  zmienna  $x$  jest typu  $\tau$ . Trzecia, że jeśli przy dodatkowym założeniu  $x : \tau_1$  można udowodnić  $t : \tau_2$ , to funkcja  $\text{fn } x \Rightarrow t$  jest typu  $\tau_1 \rightarrow \tau_2$ . Czwarta, że jeśli  $t_1$  jest funkcją typu  $\tau_1 \rightarrow \tau_2$  a  $t_2$  argumentem typu  $\tau_1$  to aplikacja  $t_1 t_2$  ma typ  $\tau_2$ .

**Zadanie.** Udowodnij indukcyjnie następujący lemat o podstawianiu: dla dowolnego kontekstu  $\Gamma$ , termów  $t, t'$  i typów  $\tau_x, \tau_t$ , jeśli

- zmienna  $x$  ma wolne wystąpienia w termie  $t$ ,
- $\Gamma \cup \{x : \tau_x\} \vdash t : \tau_t$ , oraz
- $\Gamma \vdash t' : \tau_x$ ,

to  $\Gamma \vdash t[x \mapsto t'] : \tau_t$ . Tutaj  $t[x \mapsto t']$  oznacza wynik podstawienia termu  $t'$  za zmienną  $x$  w termie  $t$ .

## Programowanie

Za tę część 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  $\{0, 1, +, *, [, ]\}$  dana jest za pomocą następującego zbioru produkcji:

$$\{S \rightarrow S + S, S \rightarrow S * S, S \rightarrow [S], S \rightarrow 0, S \rightarrow 1\}$$

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

- a) Czy  $[1 + 1] * [0 + 1]$  należy do  $L(G_1)$ ? Odpowiedź uzasadnij. **(1)**
- b) Dlaczego  $G_1$  nie jest jednoznaczna? Dlaczego dla tej konkretnie gramatyki brak jednoznaczności jest problemem **(2)**
- c) Przedstaw wyrażenie regularne lub gramatykę bezkontekstową generującą zbiór

$$A_1 = \mathcal{L}(((\text{lewy})^* 0^* (\text{prawy})^* (\text{plus})^*)^*) \cap L(G_1)$$

. Słowa *(lewy)*, *(prawy)*, *(plus)* oznaczają odpowiednie elementy alfabetu (nawiasy i operator). Odpowiedź uzasadnij. **(3)**

- d) 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  $A_1$ . Możesz używać języka wybranego z następującej listy: C, C++, Java, C#, Python, Ruby, Go, AWK, Rust. Możesz definiować funkcje pomocnicze, pamiętaj by odpowiednio je nazwać i krótko wyjaśnić ich działanie. **(4)**

**Uwaga:** na potrzeby reszty egzaminu fraza *język funkcyjny* oznacza języki Haskell lub Racket (do wyboru)

**Zadanie 2. (6p)** W języku funkcyjnym

- a) Napisz funkcję `maksymalny_zysk(L)`, która na wejściu bierze ciąg liczb zmiennopozycyjnych, stanowiących ciąg cen danej akcji w kolejnych dniach działania giełdy, i zwraca maksymalny zysk, jaki odnotujemy, gdy akcję tę jednokrotnie kupimy i jednokrotnie sprzedajemy (zakładamy, że na początku akcji nie mamy, zatem musimy zacząć od kupna).
- b) Napisz funkcję `sortuj_bity(L)`, która bierze na wejściu ciąg liczb całkowitych i zwraca ciąg posortowany. Mamy gwarancję, że na liście znajdują się jedynie liczby 0 i 1, funkcja powinna działać w czasie liniowym.

**Zadanie 3. (4p)** Napisz program obliczający wartość wyrażenia, złożonego z nawiasów, stałych 0, 1, oraz operatorów  $+$  i  $*$ . Operator  $+$  interpretujemy jako logiczne **or**, operator  $*$  interpretujemy jako logiczne **and**, natomiast stałe 1 i 0 to odpowiednio **True** oraz **False**. Możesz używać Prologa (wówczas Twoim celem jest napisanie predykatu `evaluate(+Expression, -Value)`), albo języka funkcyjnego (wówczas piszesz funkcję, której argumentem jest wyrażenie).

## Matematyka dyskretna

Pokaż, że dowolny graf z więcej niż  $\binom{n-1}{2} + 1$  krawędziami ma cykl Hamiltona.

## 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: element dominujący (4 punkty)

Tablica  $A[0 \dots n - 1]$  posiada element dominujący, jeśli ponad połowa elementów w tej tablicy ma taką samą wartość. Jak stwierdzić, czy w zadanej tablicy znajduje się element dominujący? A jeśli w tablicy jest element dominujący, to jaka jest jego wartość?

Zaprojektuj efektywny algorytm rozwiązujący ten problem. Opisz ideę algorytmu, uzasadnij jego poprawność, zapisz go w pseudokodzie i oszacuj jego złożoność obliczeniową.

Elementy w tablicy nie muszą należeć do zbioru uporządkowanego, muszą jednak być rozróżnialne — możesz więc zadawać pytania o równość elementów: czy  $A[i] = A[j]$ ?

### Zadanie 1: usuwanie elementów ze słownika statycznego (5 punktów)

Słownik statyczny to struktura danych, którą najpierw wypełniamy określonymi wartościami a potem tylko odpytujemy czy dana wartość znajduje się w takiej strukturze. Do implementacji słownika statycznego często wykorzystujemy tablice z haszowaniem.

W kontekście tablic z haszowaniem odpowiedz na następujące pytania:

- Co to jest funkcja haszująca i jaką spełnia rolę.
- Na czym polega metoda adresowania otwartego w tablicach z haszowaniem?
- Jakie znasz sposoby rozwiązywania konfliktów w takiej tablicy (konflikty mogą się pojawić podczas wstawiania do niej nowych elementów)?
- Opisz, jak można zaimplementować usuwanie niepotrzebnych elementów z tablicy z haszowaniem, przy zastosowaniu adresowania otwartego. Jak należy zmodyfikować procedurę wyszukiwania, aby po usunięciu elementu nadal działała poprawnie? Przeanalizuj czas działania opisanej metody.

## Metody numeryczne

Za rozwiązanie zadania 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** Opisz wybraną metodę rozwiązywania równania nieliniowego  $f(x) = 0$ .
2. **4 punkty** Sformułuj zadanie interpolacji wielomianowej Lagrange’a. Jak efektywnie, pod względem numerycznym i złożoności obliczeniowej, rozwiązać to zadanie?
3. **4 punktów** Opisz wybraną metodę przybliżonego obliczania całki  $\int_0^1 f(x)dx$ , gdzie  $f$  jest daną funkcją ciągłą.

## Matematyka II — Algebra

Zadanie 1. (3+4 punkty)

Rozłożyć wielomian  $w(x) = x^5 + x^3 + x + 1$  na czynniki nierozkładalne nad

a)  $\mathbb{Z}_2[x]$ ,

b)  $\mathbb{Z}_3[x]$ .

Zadanie 2. (3+4 punkty)

Dla macierzy

$$A = \begin{bmatrix} -1 & -1 & 2 \\ 2 & 2 & 1 \\ 6 & 2 & 6 \end{bmatrix}$$

a) wyznaczyć wartości własne,

b) obliczyć krotności algebraiczne i geometryczne tych wartości.

Progi punktowe: 4, 6, 8, 10, 12 punktów.