Egzamin licencjacki — 1 lipca 2011

Z zestawu sześciu zadań (Matematyka I, Matematyka II, Programowanie, Matematyka dyskretna, Algorytmy i struktury danych i Metody numeryczne) poniżej należy wybrać i przedstawić na osobnych kartkach rozwiązania trzech zadań. Za brakujące (do trzech) zadania zostanie wystawiona ocena nieostateczna z urzędu. Egzamin uważa się za zaliczony, jeśli student rozwiąże z oceną dostateczną co najmniej 2 zadania. Wtedy ocena z egzaminu jest średnią arytmetyczną ocen z trzech wybranych zadań. Na rozwiązanie zadań przeznacza się czas 3x40=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 wszystkich skończonych list nad danym zbiorem X jest zdefiniowany indukcyjnie w następujący sposób:

- 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.

Rozważmy dwie funkcje $f : L \to \mathbb{N}$ oraz $g : \mathbb{N} \times L \to \mathbb{N}$ zdefiniowane indukcyjnie w następujący sposób:

$$
\begin{aligned}
f(\mathsf{nil}) &= 0 \\
f(x : xs) &= 1 + f(xs) \\
g(n, \mathsf{nil}) &= n \\
g(n, x : xs) &= g(n+1, xs)
\end{aligned}
$$

1. Udowodnij indukcyjnie, że dla wszystkich liczb naturalnych $n \in \mathbb{N}$ oraz wszystkich skończonych list $xs \in L$ zachodzi równość
$$g(n, xs) = n + g(0, xs).$$
2. Udowodnij indukcyjnie, że dla wszystkich skończonych list $xs \in L$ zachodzi równość
$$f(xs) = g(0, xs).$$

   Uwaga: W dowodzie w punkcie 2 możesz skorzystać z punktu 1 jako lematu, nawet jeśli go nie udowodniłeś.

Matematyka II

1. Niech $V = \{(x, y, z) \in \mathbb{R}^3 : x + y + z = 0\}$.
   (a) Wykazać że jest to podprzestrzeń przestrzeni liniowej $\mathbb{R}^3$.
   (b) Jaki jest wymiar tej podprzestrzeni?

(c) Podać, wraz z uzasadnieniem, przykład bazy tej podprzestrzeni.

2. Przekształcenie liniowe $L : \mathbb{R}^3 \to \mathbb{R}^3$ dane jest wzorem

$$L(x, y, z) = (x + y,\ 2x - y + z,\ x + y - 2z).$$

Znaleźć macierz tego przekształcenia w bazie $f_1 = (1, 0, 0)^T$, $f_2 = (1, 1, 0)^T$, $f_3 = (1, 1, 1)^T$.

3. Obliczyć macierz odwrotną do macierzy

$$\mathrm{A} = \begin{bmatrix} 2 & 2 & 2 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}.$$

Programowanie

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

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

$$\{A \to a,\ A \to Aa,\ B \to b,\ B \to bB, S \to SS,\ S \to ASB,\ S \to BSA, S \to \varepsilon\}$$

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

$$\{S \to abSba\ ,\ S \to aa,\ S \to \varepsilon\}$$

Dla gramatyki G przez $L(G)$ rozumieć będziemy język generowany przez G.

a) Czy $aabbab$ należy do $L(G_1)$? Odpowiedź uzasadnij. **(1)**

b) Wskaż najkrótsze słowo, które zawiera litery a oraz b i nie należy do $L(G_1)$. **(1)**

c) Co to znaczy, że gramatyka jest jednoznaczna. Czy gramatyka G_1 jest jednoznaczna (odpowiedź uzasadnij)? **(2)**

d) Czy $L(G_1)$ jest językiem regularnym? Odpowiedź uzasadnij. **(3)**

e) Niech $A_1 = L(G_1) \cap L(G_2)$. Opisz, jakie słowa należą do A_1, odpowiedź uzasadnij[1]. **(3)**

Część 2. Będziemy rozważać binarne drzewo poszukiwań, jako strukturę służącą do zapamiętania zbioru liczb całkowitych. Zadanie to ma dwa warianty, z których musisz wybrać jeden. Jeżeli w odpowiedzi znajdą się oba, to będzie sprawdzany tylko pierwszy. Postaraj się, by Twoje implementacje były efektywne.

[1] W opisie nie powinieneś odwoływać się do nazw G_1 oraz G_2, odpowiedź: „słowa które należą jednocześnie do obu gramatyk" nie będzie uznana

Wariant funkcjonalny

Możesz używać Haskella albo OCamla. W specyfikacji zadania używamy typów Haskellowych. Każdy podpunkt warty jest 2.5p

a) Opisz, jak zapamiętać drzewo binarne przechowujące liczby całkowite w Haskellu.

b) Napisz funkcję `dodaj :: Int -> Tree -> Tree`, która bierze element i drzewo, a zwraca drzewo, do którego został dodany ten element (jeżeli go jeszcze w drzewie nie było)

c) Napisz funkcję `zawiera :: Int -> Tree -> Bool`, która zwraca wartość True wtedy i tylko wtedy, gdy element znajduje się w drzewie.

d) Napisz, jak zmieniłaby się definicja drzewa, oraz sygnatura i semantyka ww funkcji, gdyby drzewo miało służyć nie jako zbiór, lecz jako słownik.

Wariant logiczny

W tym wariancie powinieneś używać Prologa. Każdy podpunkt warty jest 2.5p

a) Opisz, jak zapamiętać drzewo binarne przechowujące liczby całkowite w Prologu.

b) Napisz predykat `dodaj(Element, StareDrzewo, NoweDrzewo)`, który unifikuje `NoweDrzewo` z drzewem powstałym z StaregoDrzewa po dodaniu Elementu.

c) Napisz predykat `zawiera(Element,Drzewo)`, która nie zawodzi wtedy i tylko wtedy, gdy element znajduje się w drzewie.

d) Napisz, jak zmieniłaby się definicja drzewa, oraz sygnatura i semantyka ww predykatów, gdyby drzewo miało służyć nie jako zbiór, lecz jako słownik.

Matematyka dyskretna

Znajdź funkcję $f(n)$ w postaci zwartej (zapisaną bez symboli $\sum$ i $\cdots$), taką że

$$\sum_{k=1}^{n} k^2 2^{n-k} = \Theta(f(n)).$$

Odpowiedź uzasadnij.

Algorytmy i struktury danych

Za rozwiązanie wszystkich trzech zadań z tej części można otrzymać 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 — ocenę bardzo dobrą.

Zadanie 1: sortowanie przez wstawianie w średnim przypadku (3.0 punkty)
Rozważmy standardowy algorytm *sortowania przez wstawianie*, pracujący na danych umieszczonych w tablicy.

- Zapisz w pseudokodzie ten algorytm (wraz z komentarzami).
- Napisz co to znaczy, że algorytm jest stabilny. Uzasadnij, że sortowanie przez wstawianie działa stabilnie.
- Policz, ile porównań w średnim przypadku wykona algorytm sortowania przez wstawianie. Ile porównań wykona ten algorytm w przypadku najgorszym i w przypadku najlepszym?

Zadanie 2: maksymalny podciąg niezależny (3.6 punkta)
Dla zadanego ciągu n wartości rzeczywistych $A = (a_1, a_2, \ldots, a_n)$ znajdź niezależny podciąg o największej możliwej sumie. Przez podciąg niezależny rozumiemy podciąg, którego elementy nie sąsiadują ze sobą w oryginalnym ciągu.

- Zaprojektuj algorytm, który znajdzie i zwróci maksymalny podciąg niezależny.
- Uzasadnij, że twój algorytm działa poprawnie.
- Przeanalizuj złożoność obliczeniową twojego algorytmu.

Zadanie 3: słownik z operacją `lessthan` (2.4 punkta)
Niech S będzie *słownikiem*, w którym oprócz standardowych operacji słownikowych mamy jeszcze operację *lessthan(x)*. Operacja *lessthan(x)* określa, ile elementów w słowniku przyjmuje wartości mniejsze od x, czyli:

$$S.lessthan(x) = |\{a \in S : a < x\}|$$

Zaprojektuj taki słownik, w którym wszystkie operacje słownikowe i operacja *lessthan(x)* będą miały pesymistyczną złożoność czasową $O(\log n)$, gdzie $n = |S|$.

- W zadaniu wykorzystaj jakąś znaną strukturę danych, efektywnie realizującą operacje słownikowe. Naisz, w jaki sposów zaadoptowałeś tą strukturę do tego zadania.
- Procedurę *lessthan(x)* zapisz w pseudokodzie i opisz jej działanie.
- Krótko ale precyzyjnie opisz, jak zmodyfikowałeś pozostałe procedury słownikowe.

Metody numeryczne

1. Niech dana będzie macierz nieosobliwa $A \in \mathbb{R}^{n \times n}$, o której wiadomo, że jej rozkład LU istnieje.

   (a) Zaproponuj algorytm wyznaczania tego rozkładu i podaj jego złożoność.

   (b) Opisz szczegółowo jak efektywnie obliczyć $\det(A)$ oraz A^{-1} dysponując rozkładem LU macierzy A.