Egzamin licencjacki — 10 września 2010

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

Rozważmy gramatykę bezkontekstową ze zbiorem symboli nieterminalnych $\{S, A, B\}$, zbiorem symboli terminalnych $\{a, b\}$, symbolem startowym S i produkcjami

$$\begin{aligned} S &\rightarrow aB \mid bA \\ A &\rightarrow SaS \mid Sa \mid a \\ B &\rightarrow SbS \mid Sb \mid b \end{aligned}$$

Udowodnij indukcyjnie, że każde słowo generowane przez tę gramatykę zawiera tyle samo liter a i b.

Matematyka II

1. Sprawdzić, czy wektory $v_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$, $v_2 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$, $v_3 = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$ są liniowo niezależne
   (a) w przestrzeni R^3 nad ciałem R,
   (b) w przestrzeni Z_2^3 nad ciałem Z_2.
2. W pierścieniu Z_7 obliczyć wartość wyrażenia $\frac{2}{3} + \frac{4}{5}$.
3. Dana jest grupa G i ustalony jej element: a. Sprawdzić czy przekształcenie $f : G \rightarrow G$, określone wzorem $f(x) = axa^{-1}$ jest homomorfizmem tej grupy w siebie.

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 $\{a, b, c\}$ dana jest za pomocą następującego zbioru produkcji:

$$\{S \rightarrow aSc\ ,\ S \rightarrow aSbSc,\ S \rightarrow \varepsilon\}$$

a) Czy gramatyka G_1 jest jednoznaczna (odpowiedź uzasadnij)? **(3)**

b) Niech $A_1 = L(G_1) \cap \mathcal{L}((a^*b^*))$. Opisz, jakie słowa należą do A_1. Odpowiedź uzasadnij. **(2)**

c) Niech
$$A_2 = L(G_1) \setminus \mathcal{L}((a+b+c)^*b(a+b+c)^*)$$
Podaj wyrażenie regularne lub gramatykę bezkontekstową definiującą A_2. **(2)**

d) Niech $A_3 = L(G_1) \cap \mathcal{L}(a^*b^*c^*)$. Przedstaw wyrażenie regularne lub gramatykę bezkontekstową definiującą A_3 **(2)**

e) Wśród języków $L(G_1)$, A_2 oraz A_3 wybierz taki, który nie jest regularny. Krótko uzasadnij dlaczego nie jest. **(1)**

Część 2. Zadanie to ma dwa warianty, z których musisz wybrać jeden. Jeżeli w odpowiedzi znajdą się oba, to będzie sprawdzany tylko pierwszy.

Wariant funkcjonalny

Możesz używać Haskella albo SML-a. W specyfikacji zadania używamy typów Haskellowych.

a) Napisz funkcję `rev :: [a] -> [a]`, która odwraca listę, czyli `rev [1,2,3] = [3,2,1]`. Funkcja powinna działać możliwie jak najefektywniej oraz nie powinna korzystać z funkcji łączącej listy. **(2)**

b) Przedstaw alternatywną implementację funkcji odwracającej listy, w której podstawowym priorytetem jest zwięzłość i czytelność definicji (tym razem można użyć funkcji łączącej listy). Wyjaśnij dokładnie, w czym ta definicja jest gorsza od poprzedniej. **(3)**

c) Napisz funkcję `listToNumber :: [Int] -> Int`, która bierze listę cyfr stanowiących zapis liczby w systemie dziesiętnym i zwraca liczbę odpowiadającą temu ciągowi cyfr, czyli: `listToNumber [1,2,3] == 123`. **(2)**

d) Napisz funkcję `numberToList :: Int -> [Int]`, która wykonuje odwrotną operację, czyli przykładowo: `numberToList 234 == [2,3,4]`. **(3)**

Wariant logiczny

W tym wariancie powinieneś używać Prologa.

a) Napisz predykat `rev(+L,-R)`, prawdziwy, gdy `R` jest listą `L` przeczytaną wspak, czyli `rev([1,2,3],[3,2,1])`. Predykat powinien działać możliwie jak najefektywniej oraz nie powinien korzystać z `append`. **(2)**

b) Przedstaw alternatywną implementację predykatu odwracającego listy, w której podstawowym priorytetem jest zwięzłość i czytelność definicji (tym razem można użyć apppenda). Wyjaśnij dokładnie, w czym ta definicja jest gorsza od poprzedniej. **(3)**

c) Napisz predykat `listToNumber(+L,-N)`, prawdziwy gdy lista cyfr jest zapisem dziesiętnym liczby N, czyli: `listToNumber([1,2,3], 123)`. **(2)**

d) Napisz predykat `numberToList(+N,-L)`, który wykonuje odwrotną operację, czyli przykładowo: `numberToList(234, [2,3,4])`. **(3)**

Matematyka dyskretna

Oblicz postać zwartą sumy:

$$\sum_{k=1}^{n} \frac{1}{k(k+2)}$$

Algorytmy i struktury danych

Za poprawne rozwiązanie wszystkich zadań z tej części można otrzymać 9 punktów. 3 punkty dają ocenę dostateczną, 4 — dostateczną z plusem, 5 — dobrą, 6 — dobrą z plusem, 7 — ocenę bardzo dobrą.

Zadanie 1: sortowanie liczb naturalnych (4 punkty)

W n–elementowej tablicy T zapisane są liczby naturalne z zakresu od 0 do $n^2 - 1$. Zaprojektuj i opisz algorytm, który posortuje te liczby w czasie liniowym.

- (2.0 pkt.) Napisz w pseudokodzie tę procedurę albo dokładnie ją opisz. Krótko uzasadnij, że działa ona poprawnie.
- (1.0 pkt.) Oszacuj złożoność czasową i pamięciową tej procedury.
- (1.0 pkt.) Czy twój algorytm jest stabilny i czy działa w miejscu? Odpowiedź uzasadnij.

Zadanie 2: najkrótsze ścieżki w grafie (5 punktów)

Opisz kopiec dwumianowy i zastosuj tę strukturę danych w algorytmie Dijkstry, który wyznacza najkrótsze drogi z ustalonego wierzchołka do wszystkich pozostałych w spójnym grafie skierowanym $D = (V, E, w)$ z nieujemnymi wagami na krawędziach, gdzie $V = \{0, 1, \ldots, n\text{–}1\}$ to zbiór wierzchołków, $E \subseteq \{(i, j) \ : \ i, j \in V \ \wedge \ i \neq j\}$ to zbiór krawędzi a $w : E \mapsto \mathbf{R}_+$ to wagi poszczególnych krawędzi.

- (2.5 pkt.) Opisz budowę kopca dwumianowego (zacznij od drzew dwumianowych). Jak działają w tej strukturze danych operacje łączenia kopców *meld*, wstawiania *insert*, usuwania minimum *extract–max* i zmiejszania wartości klucza w kopcu *decrease–key*? Jakie są czasy wykonania tych operacji?
- (2.5 pkt.) Opisz zachłanny algorytm Dijkstry, który oblicza odległości wszystkich wierzchołków grafu D od ustalonego źródła $s \in V$. Algorytm ten buduje zbiór X zawierający wierzchłki, dla których wagi najkrótszych ścieżek ze źródła s zostały juz obliczone. Metoda ta rozpoczyna działanie od jednoelementowego zbioru $\{s\}$ i w każdym kroku iteracji dokłada jeden wierzchołek spoza Xleżący najbliżej s. Krótko ale precyzyjnie opisz działanie algorytmu Dijkstry albo zapisz go w speudokodzie wraz z komentarzami. Uzasadnij krótko jego poprawność. Wskaż, gdzie w tym algorytmie można zastosować kopce dwumianowe. Wylicz, jaki wpływ na złożoność czasową algorytmu Dijkstry na zastosowanie w nim kopców dwumianowych.

Metody numeryczne

1. Dana jest pewna macierz A rozmiaru $n \times n$, gdzie $n \approx 1\,000\,000$. O macierzy tej wiemy, że na głównej przekątnej ma elementy o wartości bezwzględnej większej niż π. Wiadomo też, że w każdym wierszu macierzy jest nie więcej niż 50 niezerowych elementów, których suma modułów nie przekracza 2π. Pozycje niezerowych elementów są znane. Zaproponuj efektywny algorytm obliczenia rozwiązania układu równań liniowych

$$Ax = b,$$

gdzie $b \in \mathbb{R}^n$. Jakie teoretyczne fakty dają gwarancję, że Twój algorytm zadziała poprawnie (tzn. obliczy dobre rozwiązanie).

2. Chcemy wyznaczyć wartość całki

$$\int_{-1}^{1} \frac{w(x)}{\sqrt{1-x^2}}\, dx,$$

gdzie w jest wielomianem 11 stopnia. Z pewnych względów możemy jednak obliczyć wartości wielomianu w tylko w 6 punktach przedziału $[-1, 1]$. Jak należy wybrać te punkty, aby za pomocą odpowiednich wzorów otrzymać możliwie najlepsze przybliżenie powyższej całki. Czy Twój wybór gwarantuje, że teoretyczny błąd przybliżenia nie przekroczy $4 \cdot 10^{-15}$? Odpowiedzi należy precyzyjnie uzasadnić.