

Egzamin licencjacki — 9 września 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 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 gramatykę bezkontekstową G z jednym symbolem nieterminalnym S , zbiorem symboli terminalnych $\{ (,) \}$, symbolem startowym S i produkcjami

$$S \rightarrow (S)S \mid \varepsilon.$$

Mówimy, że słowo w nad alfabetem $\{ (,) \}$ jest *ciągami poprawnie rozstawionych nawiasów* jeśli $\text{balance}(w) = 0$ oraz $\text{balance}(u) \geq 0$ dla każdego prefiksu u słowa w , gdzie funkcja balance jest zdefiniowana indukcyjnie w następujący sposób:

$$\begin{aligned} \text{balance}(\varepsilon) &= 0, \\ \text{balance}(x(&= \text{balance}(x) + 1, \\ \text{balance}(x) &= \text{balance}(x) - 1. \end{aligned}$$

Dla przykładu $((()())())$ jest ciągiem poprawnie rozstawionych nawiasów, zaś $((()))(()$ nim nie jest.

Udowodnij indukcyjnie, że każdy ciąg poprawnie rozstawionych nawiasów jest generowany przez gramatykę G .

Matematyka II — Algebra

- (4p. + 4p.) Z pomocą algorytmu Euklidesa:
 - Znaleźć największy wspólny dzielnik dla 13 i 8.
 - Znaleźć odwrotność 8 w \mathbb{Z}_{13}^*
- (4p. + 4p.) Rozważamy układ równań postaci $Ax = 0$, gdzie $A \in \mathbb{R}^{n \times n}$.
 - Wykazać, że zbiór rozwiązań tego układu jest podprzestrzenią liniową przestrzeni \mathbb{R}^n .
 - Podać przykład bazy podprzestrzeni rozwiązań, jeśli

$$A = \begin{bmatrix} 2 & 1 & 3 \\ 1 & 0 & 2 \\ 3 & 1 & 5 \end{bmatrix}$$

- (4p.) Obliczyć macierz odwrotną do macierzy

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{bmatrix}.$$

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.

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

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

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

$$\{S \rightarrow aSa, S \rightarrow bSb, S \rightarrow b, S \rightarrow a, S \rightarrow \varepsilon\}$$

Dla gramatyki G przez $L(G)$ rozumiemy język generowany przez G .

- Czy $abbbaab$ należy do $L(G_1)$? Odpowiedź uzasadnij. **(1p.)**
- Wskaż najkrótsze słowo, które zawiera litery a oraz b i nie należy do $L(G_1)$. **(1p.)**
- Co to znaczy, że gramatyka jest jednoznaczna. Czy gramatyka G_1 jest jednoznaczna (odpowiedź uzasadnij)? **(2p.)**
- Czy $L(G_1)$ jest językiem regularnym? Odpowiedź uzasadnij. **(3p.)**
- Niech $A_1 = L(G_1) \cap L(G_2)$. Opisz, jakie słowa należą do A_1 , odpowiedź uzasadnij¹. **(3p.)**

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

Wariant funkcjonalny

Możesz używać Haskell'a albo OCaml'a. W specyfikacji zadania używamy typów Haskellowych.

- Napisz funkcję `sort :: [Int] -> [Int]`, która sortuje rosnąco listę liczb całkowitych. Możesz definiować funkcje pomocnicze, postaraj się by kod był możliwie jak najbardziej czytelny (oczywiście nie możesz korzystać z bibliotecznej funkcji sortującej). **(5p.)**
- Napisz funkcję liczącą długość listy, w której używamy ogonowej rekurencji. Podaj jej typ. **(3p.)**
- Napisz funkcję `setSize :: [Int] -> Int`, która zwraca liczbę **różnych** elementów na liście będącej argumentem. Skoncentruj się na zwięzłości kodu. **(2p.)**

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

Wariant logiczny

W tym wariancie powinieneś używać Prologa.

- Napisz predykat `sort(L1,L2)`, który unifikuje `L2` z posortowaną rosnąco listą liczb całkowitych `L1`. Możesz definiować predykaty pomocnicze, postaraj się by kod był możliwie jak najbardziej czytelny (oczywiście nie możesz korzystać z bibliotecznego predykatu sortującego). (5p.)
- Napisz predykat liczącą długość listy, w którym używamy ogonowej rekurencji. Opisz znaczenie argumentów i sposób użycia tego predykatu. (3p.)
- Napisz predykat `setSize(L,N)`, który unifikuje `N` z liczbą **różnych** elementów na liście `L`. Skoncentruj się na zwięzłości kodu. (2p.)

Matematyka dyskretna

W wyborach wystawiono jedną urnę. Kandydaci A i B otrzymali po n głosów. Każda kolejność wrzucenia głosów do urny jest tak samo prawdopodobna. Oblicz prawdopodobieństwo, że w każdym momencie wyborów w urnie było co najmniej tyle samo głosów na kandydata A co na kandydata B . Możesz wykorzystać powszechnie znane wzory i twierdzenia bez dowodzenia ich.

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), 3p. dają ocenę dostateczną, 4p. — dostateczną z plusem, 5p. — dobrą, 6p. — dobrą z plusem, 7p. — ocenę bardzo dobrą.

Zadanie 1: sortowanie punktów w kole (3.0p.)

Danych jest n punktów $\{p_1, p_2, \dots, p_n\}$ leżących w kole jednostkowym, co oznacza, że dla każdego punktu $p_i = (x_i, y_i)$ jest spełniony warunek $0 < x_i^2 + y_i^2 \leq 1$ dla $i = 1, 2, \dots, n$. Zakładamy, że punkty w tym kole są rozmieszczone jednostajnie, czyli prawdopodobieństwo zdarzenia, że punkt znajduje się w dowolnym ustalonym obszarze wewnątrz koła jednostkowego jest proporcjonalne do pola tego obszaru.

Zaprojektuj algorytm działający średnio w czasie liniowym $O(n)$, sortujący n punktów według ich odległości $d_i = \sqrt{x_i^2 + y_i^2}$ od początku układu współrzędnych.

- Opisz dokładnie twój algorytm (albo zapisz go w pseudokodzie wraz z komentarzami).
- Napisz co to znaczy, że algorytm działa w miejscu. Czy twój algorytm działa w miejscu?
- Oszacuj oczekiwany czas działania algorytmu.

Zadanie 2: problem plecakowy (2.4p.)

Opisz ciągi problem plecakowy:

- Opisz dokładnie na czym polega ciągły problem plecakowy (co jest dane i jakich oczekujemy wyników). Jaką techniką rozwiązuje się to zadanie? Napisz w pseudokodzie algorytm rozwiązujący to zadanie.

- b) Udowodnij (niewprost) poprawność przedstawionego algorytmu.
- c) Na czym polega różnica między ciągłym a dyskretnym problemem plecakowym? Czy przedstawiony algorytm dla wersji ciągłej będzie działał również dla wersji dyskretnej problemu plecakowego? Odpowiedź uzasadnij.

Zadanie 3: zbiór dynamiczny z medianą (3.6p.)

Zaprojektuj strukturę danych, która umożliwi efektywne wykonywanie następujących operacji na zbiorze dynamicznym:

- a) *insert*(x) — wstawienie elementu x do zbioru,
- b) *find-median*() — wskazanie mediany zbioru,
- c) *extract-median*() — usunięcie mediany ze zbioru.

Wskazanie mediany powinno działać w stałym czasie; pozostałe operacje w czasie logarytmicznym względem aktualnej liczby elementów w zbiorze.

Do rozwiązania tego zadania wykorzystaj jakąś znaną strukturę danych, która efektywnie realizuje operacje kolejki priorytetowej. Krótko ale precyzyjnie opisz działanie wymienionych procedur.

Metody numeryczne

- Niech dane będą parami różne liczby rzeczywiste x_0, x_1, \dots, x_n oraz odpowiadające im wartości $y_0, y_1, \dots, y_n \in \mathbb{R}$. Niech $L_n \in \Pi_n$ oznacza wielomian interpolacyjny dla tych danych, tzn.

$$L_n(x_i) = y_i \quad (i = 0, 1, \dots, n).$$

Zaproponuj algorytm wyznaczania wartości

$$L_n(z_0), L_n(z_1), \dots, L_n(z_m),$$

gdzie z_0, z_1, \dots, z_m ($m \in \mathbb{N}$; m może być duże) są dane. Jaka jest jego złożoność?