Egzamin licencjacki — 4 lipca 2008

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

Za poprawne rozwiązanie całego zadania można otrzymać 9 punktów. 3 punkty dają ocenę dostateczną, 4 — dostateczną z plusem, 5 — dobrą, 6 — dobrą z plusem, a 7 — ocenę bardzo dobrą.

1. **(2 punkty)** Podaj formułę równoważną formule $p \Leftrightarrow (q \wedge r)$ i mającą:
   (a) koniunkcyjną postać normalną
   (b) dysjunkcyjną postać normalną
2. **(1 punkt)** Wykaż, że jeśli a i b są dodatnimi liczbami rzeczywistymi spełniającymi warunki $a \leq 1$ i $b \geq 1$ to $a + b \geq ab + 1$.
3. **(1 punkt)** Sformułuj zasadę indukcji matematycznej.
4. **(5 punktów)** Wykaż indukcyjnie, że jeśli iloczyn dodatnich liczb rzeczywistych $x_1 x_2 \ldots x_n$ wynosi 1 to $x_1 + x_2 + \ldots + x_n \geq n$.

Matematyka II

Rozważmy przekształcenie $\mathcal{L} : \mathbb{R}^3 \to \mathbb{R}^3$ dane wzorem $\mathcal{L}((x, y, z)) = (x - y + 2z, x - y, z)$.

1. (**2 pkt.**) Udowodnij, że $\mathcal{L}$ jest przekształceniem liniowym.
2. (**2 pkt.**) Jak wygląda jądro przekształcenia $\mathcal{L}$? Podaj jego dowolną bazę.
3. (**2 pkt.**) Jak wygląda obraz przekształcenia $\mathcal{L}$? Podaj jego dowolną bazę.
4. (**2 pkt.**) Skonstruuj ortogonalną bazę obrazu $\mathcal{L}$. Użyj standardowego iloczynu skalarnego.

Skala ocen: 3 punkty – *dostateczny*, 4 punkty – *+dostateczny*, 5 punktów – *dobry*, 6 punktów – *+dobry*, 7 punktów – *bardzo dobry*.

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:

$$\{S \to aB,\ S \to bA,\ S \to \varepsilon,\ A \to bAA,\ A \to aS,\ B \to aBB,\ B \to bS\}$$

a) Opisz zbiór słów akceptowanych przez tę gramatykę. **(1p)**

b) Zdefiniuj gramatykę zawierającą **tylko** jeden symbol nieterminalny, która akceptuje ten sam język. **(2p)**

c) Niech $A_1 = L(G_1) \cap \mathcal{L}(a^*b^*)$. Czy A_1 jest językiem regularnym? Udpowiedź uzasadnij **(1p)** W zależności od odpowiedzi na Twoje pytanie podaj wyrażenie regularne lub gramatykę bezkontekstową definiującą A_1 **(2p)**.

d) Niech $A_2 = L(G_1) \cap \mathcal{L}((ab^*)^*)$. Przedstaw możliwie prostą gramatykę bezkontekstową lub wyrażenie regularne generujące A_2. **(1p)** Napisz funkcję w języku C, `int in_a2(char *s)` zwracającą wartość logiczną, mówiącą czy napis `s` jest słowem, które należy do A_2. **(3p)**

Przypominam, że $L(G)$ to język generowany przez gramatykę G, a $\mathcal{L}(r)$ to język generowany przez wyrażenie regularne r.

Część 2. Będziemy rozważać zadanie sortowania ciągów binarnych, to jest takich, ciągów liczb naturalnych, które zawierają jedynie zera i jedynki. Ta część egzaminu ma dwa warianty (do wyboru przez studenta, w przypadku rozwiązania obu sprawdzany jest **tylko** ten wariant, który w odpowiedzi pojawia się jako pierwszy). W obu wariantach należy zaprezentować rozwiązania działające w **czasie liniowym**.

Wariant funkcjonalny

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

a) Napisz funkcję `binsort :: [Int] -> [Int]`, która sortuje w czasie liniowym listę zer i jedynek. **(8p)**. Zwróć uwagę na czytelność rozwiązania.

b) Czy zdefiniowana przez Ciebie funkcja ma następującą własność: *długość listy będącej argumentem jest równa długości listy będącej wynikiem funkcji.* Odpowiedź uzasadnij **(2p)**

Wariant logiczny

W tym wariancie powinieneś używać Prologa.

a) Napisz działający w czasie liniowym predykat `binsort(A,B)`, o następującej własności: jeżeli zostanie wywołany z pierwszym argumentem będącym listą zer i jedynek, wówczas po zakończeniu działania drugi argument powinien zunifikować się z listą, będącą uporządkowaną permutacją pierwszego argumentu. Zwróć uwagę na czytelność rozwiązania. **(8p)**

b) Czy zdefiniowany przez Ciebie predykat ma następującą własność: *Po zakończonym sukcesem działaniu predykatu, oba argumenty zunifikowane są z pewnymi listami o tej samej długości.* Odpowiedź uzasadnij **(2p)**

Matematyka dyskretna

Czy można znaleźć taką potęgę liczby 3, która w układzie dziesiętnym kończy się cyframi 00001? Odpowiedź uzasadnij.

Algorytmy i struktury danych

Za poprawne rozwiązanie całego zadania można otrzymać 9 punktów. 3 punkty dają ocenę dostateczną, 4 — dostateczną z plusem, 5 — dobrą, 6 — dobrą z plusem, 7 — ocenę bardzo dobrą.

Część 1 (4 punkty). Jakie operacje powinna efektywnie realizować struktura danych zwana *słownikiem*? Wymień trzy znane Ci przykłady efektywnej realizacji tej struktury danych, zbudowane na drzewach ukorzenionych. Spośród wymienionych wybierz i opisz jedną wybraną implementację słownika. Przeanalizuj złożoność czasową poszczególnych operacji.

Część 2 (5 punktów). Niech S będzie słownikiem, którego elementy pochodzą ze zbioru z porządkiem liniowym z relacją $\leq$. Do operacji słownikowych dodajemy jeszcze operację *lessthan(x)* określającą ile elementów w słowniku przyjmuje wartości mniejsze od x, zdefiniowaną formalnie w następujący sposób:

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

Zaprojektuj taki słownik, w którym każda operacja słownikowa i operacja *lessthan(x)* mają pesymistyczną złożoność czasową $O(\log n)$, gdzie $n = |S|$.

W zadaniu wykorzystaj znaną Ci strukturę danych realizującą efektywnie operacje słownikowe. Procedurę *lessthan(x)* napisz w pseudokodzie i opisz jej działanie. Krótko ale precyzyjnie opisz, jak zmodyfikowałeś pozostałe procedury słownikowe.

Metody numeryczne

1. Podaj definicję naturalnej funkcji sklejanej trzeciego stopnia.

2. Korzystając z definicji, znajdź naturalną funkcję sklejaną trzeciego stopnia dla danych

$$\begin{array}{c||c|c|c} x_k & -1 & 0 & 1 \\ \hline y_k & 2 & 0 & 2 \end{array}.$$

3. Jak widomo, przy wyznaczaniu naturalnej funkcji sklejanej trzeciego stopnia należy m.in. rozwiązać następujący układ równań liniowych:

$$\begin{cases} 2M_1 + \mu_1 M_2 = f_1, \\ \lambda_2 M_1 + 2M_2 + \mu_2 M_3 = f_2, \\ \lambda_3 M_2 + 2M_3 + \mu_3 M_4 = f_3, \\ \lambda_4 M_3 + 2M_4 + \mu_5 M_5 = f_4, \\ \quad \ddots \quad \ddots \quad \ddots \quad \vdots \quad \vdots \\ \lambda_{n-2} M_{n-3} + 2M_{n-2} + \mu_{n-2} M_{n-1} = f_{n-2}, \\ \lambda_{n-1} M_{n-2} + 2M_{n-1} = f_{n-1}, \end{cases}$$

gdzie $\mu_k := 1 - \lambda_k$ oraz λ_k i f_k $(k = 1, 2 \ldots, n-1)$ są dane. Zaproponuj algorytm rozwiązywania podanego układu, którego zlożoności wynosi $O(n)$.