Egzamin licencjacki/inżynierski — 9 września 2015

Z sześciu poniższych zestawów zadań (Matematyka I, Matematyka II, Programowanie, Matematyka dyskretna, Algorytmy i struktury danych i Metody numeryczne) 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 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

Konstruując odpowiednią bijekcję udowodnij, że zbiory $\mathcal{P}(\mathbb{N})$ i $\{a, b\}^{\mathbb{N}\times\{0,1\}}$ są równoliczne.

Matematyka II — Algebra

Za zadania można otrzymać 12 punktów. Aby otrzymać ocenę dostateczną, należy zdobyć 4 punkty, próg dla dst+ to 5.5p, dla db – 7p, dla db+ 9p, dla bdb – 11p.

Zadanie 1 – 6 punktów
Znajdź k takie, że wektory $v_1 = [1\ 2\ 2\ 2]$, $v_2 = [2\ 1\ 1\ 2]$, $v_3 = [1\ 0\ 2\ 1]$, $v_4 = [1\ 1\ 2\ 0]$ są liniowo zależne nad ciałem $\mathbb{Z}_k^4$.

Zadanie 2 – 3 + 3 punkty
a) Znajdź $a, b \in \mathbb{Z}$ takie, że $17a + 31b = 1$.
b) Rozwiąz równanie $14x \equiv_{17} 1$

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

$$\{S \to SS,\ S \to XY,\ X \to \varepsilon,\ X \to aXb,\ X \to Xb,\ Y \to \varepsilon,\ Y \to bYc,\ Y \to bY\} \quad (1)$$

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

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

b) Czy gramatyka G_1 jest jednoznaczna? Odpowiedź krótko uzasadnij. **(2)**

c) Przedstaw wyrażenie regularne lub gramatykę bezkontekstową generującą zbiór $A_1 = \mathcal{L}((ab + bc + ac)^*) \cap L(G_1)$ 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 $L(G_1)$. Możesz używać języka wybranego z następującej listy: C, C++, Java, C#, Python, Ruby, PHP, AWK, Pascal. **(4)**

Część 2. Rozważmy poniższy program w Prologu:

```
p(L) :-
   append(A,B,L),
   \+ append(B,A,L),!.
```

Wyjaśnij, co sprawdza ten predykat. Napisz predykat równoważny (to znaczy taki, który kończy się sukcesem i zawodzi dokładnie dla takich samych list, co predykat `p`). W nowej wersji predykatu `p` nie wolno korzystać jawnie z negacji (ale możesz sprawdzać różność termów). **(4p)**.

Część 3. Napisz w Haskellu dwie funkcje, obie biorące jako argument listę liczb całkowitych i zwracające parę liczb. Pierwszy element tej pary to liczba liczb parzystych (na liście wejściowej), drugi – liczba liczb nieparzystych. Pierwsza z tych funkcji powinna być napisana bez jawnej rekurencji z wykorzystaniem mechanizmu *list comprehension*, druga – bez wykorzystanie żadnych funkcji standardowych Haskella (oczywiście można korzystać z operatorów i relacji arytmetycznych).**(6p)**

Matematyka dyskretna

Niech a_n będzie liczbą ciągów długości n złożonych z liter a, b, c i nie zawierających podciągów aa, bb. Napisz wzór rekurencyjny na a_n.

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: łączenie drzew BST (5 punktów)

Dane są dwa drzewa BST T_1 i T_2 o wysokościach odpowiednio h_1 i h_2. Ponadto wiemy, że elementy należące do T_1 są mniejsze od elementów z T_2, czyli że dla dowolnego elementu $v_1 \in T_1$ i dla dowolnego elementu $v_2 \in T_2$ zachodzi relacja $v_1 < v_2$. Wyjaśnij, jak połączyć ze sobą te drzewa BST w jedno drzewo $T = T_1 \cup T_2$, aby wysokość h nowo powstałego drzewa T była nie większa niż $1 + \max\{h_1, h_2\}$.

- Podaj definicję drzewa BST (z porządkiem symetrycznym włącznie).
- Opisz szczegółowo efektywny algorytm rozwiązujący ten problem (procedura napisana w pseudokodzie z wyczerpującymi komentarzami).
- Uzasadnij, że opisany algorytm działa poprawnie.
- Przeanalizuj złożoność obliczeniową przedstawionej metody.

Zadanie 2: podzbiór liczb sumujący się do zadanej wartości (4 punkty)

Dany jest ciąg n różnych liczb naturalnych $a_1, a_2, \ldots, a_n$ oraz liczba naturalna t. Jak stwierdzić, czy pewien podzbiór liczb a_i sumuje się do wartości t? Zakładamy, że przy sumowaniu każdą liczbę a_i można wykorzystać co najwyżej raz.

- Zaprojektuj algorytm dynamiczny rozwiązujący to zadanie w czasie $O(nt)$.
- Uzasadnij, że opisany algorytm działa poprawnie.
- Przeanalizuj złożoność czasową i pamięciową przedstawionej metody.

Metody numeryczne

1. Znajdź naturalną interpolacyjną funkcję sklejaną trzeciego stopnia dla następujących danych:

$$\begin{array}{c||c|c|c} x_k & -1 & 0 & 1 \\ \hline y_k & 4 & 0 & 8 \end{array}\,.$$

2. Zaproponuj efektywny algorytm znajdowania rozkładu LU macierzy

$$A_n := \begin{bmatrix} a_1 & d_2 & d_3 & \cdots & d_{n-1} & c_1 \\ b_2 & a_2 & & & & c_2 \\ b_3 & & a_3 & & & c_3 \\ \vdots & & & \ddots & & \vdots \\ b_{n-1} & & & & a_{n-1} & c_{n-1} \\ b_n & e_2 & e_3 & \cdots & e_{n-1} & a_n \end{bmatrix} \in \mathbb{R}^{n\times n}.$$