Egzamin licencjacki — 6 lipca 2007

Z zestawu sześciu zadań (Matematyka I, Matematyka II, Programowanie, Matematyka dyskretna, Algorytmy i struktury danych i Metody numeryczne) poniżej należy wybrać i przedstawić rozwiązanie 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. **(3 punkty)** Nie używając znaku negacji (można używać znaku $\notin$) napisz formuły równoważne zanegowanym formułom poniżej.
   (a) $(\forall i \leq n \;\; i \in X) \Rightarrow n+1 \in X$
   (b) $\forall i \leq n \;\; (i \in X \Rightarrow n+1 \in X)$
   (c) $\forall n \, \exists i \;\; ((i \leq n \wedge i \notin X) \vee n+1 \in X)$
2. **(1 punkt)** Czy dla dowolnych zbiorów A, B, C równość $A \cap B = A \cap C$ implikuje równość $B = C$? Odpowiedź uzasadnij.
3. **(2 punkty)** Sformułuj zasadę indukcji matematycznej.
4. **(3 punkty)** Niech liczby F_n dla $n \in \mathbb{N}$ będą zdefiniowane równaniami
   - $F_0 = 0$,
   - $F_1 = -1$,
   - $F_{n+2} = F_n + F_{n+1}$ dla wszystkich $n \in \mathbb{N}$.

   Udowodnij indukcyjnie, że dla wszystkich $n \in \mathbb{N}$ zachodzi $F_n \leq 0$.

Matematyka II

1. Podaj przykład trzech wektorów, które są liniowo niezależne w przestrzeni $\mathbb{R}^3$ nad ciałem $\mathbb{R}$ oraz liniowo zależne w przestrzeni $\mathbb{Z}_3^3$ nad ciałem $\mathbb{Z}_3$. Podany przykład uzasadnij.

2. Wykaż, że przy dowolnie ustalonym elemencie b grupy $\mathbf{G}$ odwzorowanie $a \mapsto bab^{-1}$ jest izomorfizmem grupy $\mathbf{G}$ na siebie.

3. W pierścieniu $\mathbb{Z}_8$ rozwiąż równanie $2x = 4$.

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:

$$\{S \to aSb\ ,\ S \to bSa\ \ S \to SS, S \to Sb, S \to bS\ ,\ S \to \varepsilon\}$$

a) Czy gramatyka ta jest jednoznaczna (odpowiedź uzasadnij)? **(2)**

b) Czy jest możliwe usunięcie jednej produkcji z tej gramatyki i otrzymanie w ten sposób gramatyki G_2, takiej że $L(G_1) = L(G_2)$. Odpowiedź uzasadnij. **(3)**

c) Niech $A_1 = L(G_1) \cap \mathcal{L}((aa^*b)^*)$. Czy A_1 jest językiem regularnym? Udpowiedź uzasadnij **(2)**

d) Niech $A_2 = L(G_1) \cap \mathcal{L}(a^*b^*a^*)$. Przedstaw możliwie prostą gramatykę bezkontekstową lub wyrażenie regularne generujące A_2. **(3)**

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 odwracania listy, czyli, na przykład, zamiany `[1,2,3,4]` na `[4,3,2,1]`. 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).

Wariant funkcjonalny

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

a) Napisz funkcję `append :: [a] -> [a] -> [a]`, która łączy dwie listy (tzn wykonuje ich konkatenację). **(2)**

b) Wykorzystując funkcję `append` napisz funkcję `reverse :: [a] -> [a]`, która odwraca listę. Postaraj się, by rozwiązanie było możliwie naturalne. **(2)**

c) Dlaczego powyższy program wykonuje $O(N^2)$ operacji, gdzie N jest długością listy? **(2)**

d) Napisz wersję funkcji z punktu b), która wykonuje $O(N)$ operacji. **(4)**

Wariant logiczny

W tym wariancie powinieneś używać Prologa.

a) Napisz predykat `append(A,B,C)`, prawdziwy, gdy listy `A` oraz `B` po konkatenacji dadzą listę `B`. **(2)**

b) Wykorzystując predykat `append` napisz predykat `reverse(A,B)`, prawdziwy, gdy lista `B` jest odwróconą listą `A`. Postaraj się, by rozwiązanie było możliwie naturalne. **(2)**

c) Dlaczego powyższy program wykonuje $O(N^2)$ operacji, gdzie N jest długością listy? **(2)**

d) Napisz wersję predykatu z punktu b), która wykonuje $O(N)$ operacji. **(4)**

Matematyka dyskretna

Rozwiąż zależność rekurencyjną $a_n = 4a_{n-1} - 4a_{n-2}$ przy warunkach początkowych $a_0 = 0, a_1 = 2$.

Algorytmy i struktury danych

Rozważmy problem *mnożenia długich liczb całkowitych*: dla zadanych n–cyfrowych liczb $A = (a_{n-1}a_{n-2}\dots a_0)$ i $B = (b_{n-1}b_{n-2}\dots b_0)$ należy obliczyć iloczyn $C = A \cdot B$, przy czym $C = (c_{2n-1}c_{2n-2}\dots c_0)$ będzie już liczbą $(2n)$–cyfrową. Opisz algorytm oparty na metodzie *dziel i zwyciężaj*, który efektywnie rozwiązuje to zadanie. Uzasadnij poprawność przedstawionego rozwiązania i przeanalizuj jego złożoność obliczeniową (możesz założyć, że n jest potęgą 2).

Podaj przykłady dwóch innych problemów, które można rozwiązać techniką *dziel i zwyciężaj* i oszacuj złożoność obliczeniową zastosowanych algorytmów.

Metody numeryczne

1. (a) Niech dane będą parami różne punkty $x_0, x_1 \dots, x_n$ i liczby $y_0, y_1 \dots, y_n$. Podaj postać Newtona wielomianu interpolacyjnego L_n spełniającego następujące warunki:

$$1)\ L_n \in \Pi_n; \qquad 2)\ L_n(x_i) = y_i \quad (i = 0, 1, \dots, n).$$

   (b) Znajdź wielomian interpolacyjny $L_5 \in \Pi_5$ dla danych

x_k	-3	-2	-1	1	2	3
y_k	-3	-2	-1	2007	2	3

.

2. (a) Zaproponuj efektywny algorytm obliczania wyznacznika nieosobliwej macierzy $A \in \mathbb{R}^{n \times n}$ (n może być duże). Jaka jest złożoność Twojego algorytmu?