Egzamin licencjacki/inżynierski — 9 lutego 2017

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

Zbiór $L(X)$ wszystkich skończonych list nad danym zbiorem X jest zdefiniowany indukcyjnie w następujący sposób:

- nil jest skończoną listą nad zbiorem X;
- jeśli x jest elementem zbioru X oraz xs jest skończoną listą nad zbiorem X to $x : xs$ jest skończoną listą nad zbiorem X.

Dla wszystkich zbiorów X operacja konkatenacji list $+\!\!+ : L(X) \times L(X) \to L(X)$ jest zdefiniowana w następujący sposób. Dla wszystkich elementów $x \in X$ oraz wszystkich list $xs, ys \in L(X)$ przyjmujemy

$$\begin{aligned} \mathsf{nil} +\!\!+ ys &= ys, \\ (x : xs) +\!\!+ ys &= x : (xs +\!\!+ ys). \end{aligned}$$

Dla wszystkich zbiorów X i Y definiujemy funkcję $foldr : Y^{X\times Y} \times Y \times L(X) \to Y$ w następujący sposób. Dla dowolnej funkcji $f : X \times Y \to Y$, dowolnych elementów $c \in Y$ i $x \in X$ oraz dowolnej listy $xs \in L(X)$ przyjmujemy

$$\begin{aligned} foldr(f, c, \mathsf{nil}) &= c, \\ foldr(f, c, x : xs) &= f(x, foldr(f, c, xs)). \end{aligned}$$

1. Sformułuj zasadę indukcji w takiej postaci, żeby można było jej użyć w dowodzie w punkcie 2.
2. Korzystając z zasady indukcji sformułowanej w punkcie 1 udowodnij indukcyjnie, że dla dowolnych zbiorów X, Y, dowolnej funkcji $f : X \times Y \to Y$, dowolnego elementu $c \in Y$ oraz dowolnych list $xs, ys \in L(X)$ zachodzi równość

$$foldr(f, c, xs +\!\!+ ys) = foldr(f, foldr(f, c, ys), xs).$$

Uwaga: To jest zadanie z logiki. Przy ocenianiu zwrócimy szczególną uwagę na poprawność i klarowność rozumowania, w szczególności na poprawność użytej zasady indukcji, odpowiednie sformułowanie i użycie wszystkich założeń, odpowiednie użycie kwantyfikatorów i nawiasów itp.

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.

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

$$A \to aa, A \to \varepsilon, B \to bbb, B \to \varepsilon, S \to AxBxS, S \to \varepsilon$$

Gramatyka G_2 z symbolem startowym S nad alfabetem $\{a, b, x\}$ dana jest za pomocą

$$A \to aaa, A \to \varepsilon, B \to bb, B \to \varepsilon, S \to AxBxS, S \to \varepsilon$$

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 $aaxxaax$ należy do $L(G_1)$? Odpowiedź uzasadnij. **(1p)**

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

c) Przedstaw wyrażenie regularne (o ile to możliwe) lub gramatykę bezkontekstową (w przeciwnym przypadku) generującą zbiór $A_1 = L(G_1) \cap L(G_2)$ Odpowiedź uzasadnij. **(3p)**

d) Wyjaśnij słownie, jakie napisy znajdują się w A_1. 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 $L(G_1) \cap L(G_2)$. Możesz używać języka wybranego z następującej listy: C, C++, Java, C#, Python, Ruby, PHP, AWK, Pascal. **(4p)**

Zadanie 2. W języku Python mamy dwa operatory działające na napisach: `+` oznaczający konkatenacje oraz `*` oznaczający mnożenie przez stałą (czyli k-krotną konkatenecje z samym sobą. Przykładowo: `2 * ("ali" + 2 * "ba" + " ") == "alibaba alibaba "`. Napisz w Prologu predykat `compute/2`, który oblicza wartość napisowego wyrażenia pythonowego (z dodawaniem, mnożeniem, nawiasami i stałymi napisami). Możesz skorzystać z `append`, możesz również definiować nowe predykaty. **(5p)**

Zadanie 3. Napisz w Haskellu funcję `porzadkowalne(S)`, która wykorzystuje mechanizm list comprehension i zwraca wartość logiczną równą True wtedy i tylko wtedy, gdy napisy na liście L da się ułożyć w ten sposób, by każdy poprzedni był podnapisem (ciągłym) kolejnego. **(5p)**

Matematyka dyskretna

Niech G będzie grafem n-wierzchołkowym, który składa się z dwóch składowych spójnych będących klikami. Pokaż, że

$$m(G) \geq (n^2 - 2n)/2.$$

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: zapętlona lista jednokierunkowa (4 punkty)

Dana jest lista jednokierunkowa. O liście tej nie wiemy, czy jest prawidłowo zbudowana (zakończona wskaźnikiem pustym) czy może jest zapętlona (wskazuje na jakiś węzeł ze środka listy — w tym również na pierwszy albo na ostatni węzeł).

Zaprojektuj algorytm, który wyznaczy długość tej pęli w zapętlonej liście (albo odpowie wartością 0 gdy lista jest prawidłowo zbudowana): przedstaw ideę rozwiązania, zapisz w pseudokodzie swój algorytm a na koniec oszacuj jego złożoność czasową.

Wskazówka: czy dwaj biegacze biegnący dookoła stadionu w tym samym kierunku ale z różnymi prędkościami mają szansę się kiedyś spotkać?

Zadanie 2: podciąg, który jest palindromem (5 punktów)

Dany jest ciąg n–literowy $S = (s_1, s_2, \ldots, s_n)$. Należy wyznaczyć najdłuższy podciąg w tym ciągu, który jest palindromem (słowo brzmiące tak samo czytane od strony lewej do prawej i od prawej do lewej). Zaprojektuj efektywny algorytm rozwiązujący to zadanie: przedstaw ideę rozwiązania, zapisz w pseudokodzie swój algorytm a na koniec oszacuj jego złożoność obliczeniową (czasową i pamięciową).

Wskazówka: jeśli pierwsza i ostatnia litera w tym ciągu będzie taka sama, to czy litery te wejdą w skład rozwiązania?

Przykład: Dla ciągu (a, b, c, a, b, a, c) rozwiązaniem może być 5–literowy podciąg *abcba*, *ababa* albo *cabac*.

Metody numeryczne

Za rozwiązanie zadań można otrzymać łącznie 12 punktów. Otrzymanie 4 pkt. gwarantuje ocenę dostateczną, próg dla dst+ to 5.5 pkt., dla db – 7 pkt., dla db+ 9 pkt., a dla bdb – 11 pkt.

1. $\boxed{\textbf{4 punkty}}$ Dla $x \approx 0$ obliczanie wartości wyrażenia $x^{-4}(\cos(2x) - 1 + 2x^2)$ może wiązać się z utratą cyfr znaczących wyniku. Zakładając, że $|x| \le \frac{1}{10}$, zaproponuj taki sposób obliczenia wartości tego wyrażenia, aby mieć pewność, że błąd bezwzględny nie przekracza 10^{-7}.

2. $\boxed{\textbf{4 punkty}}$ Podaj efektywny sposób wyznaczania m-tego wielomianu optymalnego w sensie dyskretnej aproksymacji średniokwadratowej, tj. konstrukcji wielomianu $w_m^* \in \Pi_m$ spełniającego warunek

$$||f - w_m^*||_2 = \min_{w_m \in \Pi_m} ||f - w_m||_2,$$

gdzie $||g||_2 := \sqrt{\sum_{k=0}^{N} g(x_k)^2}$, natomiast punkty $x_0 < x_1 < \ldots < x_N$ są dane. Jakie są zastosowania tego typu aproksymacji w analizie numerycznej?

3. $\boxed{\textbf{4 punkty}}$ Załóżmy, że istnieje rozkład LU macierzy $A \in \mathbb{R}^{n \times n}$. Jak efektywnie wyznaczyć ten rozkład? Jakie są jego zastosowania w analizie numerycznej?

Matematyka II — Algebra

Za zadania można otrzymać 13 punktów. Aby otrzymać ocenę dostateczną, należy zdobyć 3 punkty, próg dla dst+ to 5p, dla db – 7p, dla db+ 9p, dla bdb – 11p.

Zadanie 1. (8 punktów)
Wyznaczyć wartości i wektory własne macierzy:

$$A = \begin{bmatrix} -2 & -4 & 2 \\ -2 & 1 & 2 \\ 4 & 2 & 5 \end{bmatrix}.$$

Zadanie 2. (5 punktów)
Metodą eliminacji Gaussa rozwiązać układ równań

$$\begin{bmatrix} 1 & -3 & 1 \\ 2 & -8 & 8 \\ -6 & 3 & -15 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 4 \\ -2 \\ 9 \end{bmatrix}.$$