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

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

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

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

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

$$S \to aSb, S \to \varepsilon, S \to cSd, S \to SS$$

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 $aabbacdb$ 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 $A_1 = L(G_1) \cap \mathcal{R}(a^*c^*b^*d^*)$ Odpowiedź uzasadnij. **(3p)**

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 $L(G_1) \cap A_2$, gdzie $A_2 = \mathcal{R}(a^*c^*d^*b^*)$. Możesz używać języka wybranego z następującej listy: C, C++, Java, C#, Python, Ruby, PHP, AWK, Pascal. **(4p)**

Zadanie 2. Napisz w Prologu predykat `p/1`, który jako argument bierze stałą listę `L` i kończy się sukcesem wtedy i tylko wtedy, gdy lista ta zawiera tylko jedną wartość (być może powtórzoną kilkukrotnie). Implementując `p` nie powinieneś używać rekurencji, możesz natomiast używać negacji, predykatu `member` oraz unifikacji. **(4p)** Jaki jest wynik następujących wywołań Twojego predykatu:

```
?- p([1,1,1,1,X]).
?- p([2,2,2,2,A,B,A]).
```

Zadanie 3. Napisz w Haskellu funcję `parzysteSumy(L)`, która dla listy liczb `L` (nie zawierającej powtórzeń) zwraca listę wszystkich liczb parzystych, które są sumą pewnego podzbioru elementów listy `L`. Kolejność elementów w wyniku nie ma znaczenia, żadna liczba w wyniku nie powinna się powtarzać. Skoncentruj się na czytelności kodu, możesz definiować funkcje pomocnicze. Dla każdej funkcji pomocniczej opisz znaczenie jej argumentów i wyniku. **(5p)**

Matematyka dyskretna

Niech G będzie grafem, w którym wierzchołki mają stopnie 5 i 6. Pokaż, że G ma co najmniej 5 wierzchołków stopnia 6 lub co najmniej 6 wierzchołków stopnia 5.

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: usuwanie powtórzeń (5 punktów)

Dana jest n–elementowa tablica zapełniona liczbami rzeczywistymi. Niektóre liczby w tej tablicy moga się powtarzać. Skonstruuj efektywny algorytm (działający poniżej czasu $O(n^2)$), który usunie z tej tablicy wszystkie powtórzenia.

Zapisz swój algorytm w pseudokodzie i dopisz komentarz do tego kodu. Oszacuj złożoność obliczeniową opisanej metody i uzasadnij jej poprawność.

Zadanie 2: graf dwukolorowy (4 punkty)

Dany jest spójny graf prosty $G = (V, E)$. Skonstruuj algorytm, który sprawdzi czy graf ten da się pokolorować dwoma kolorami. Kolorowanie takie polega na takim przyporządkowaniu dwóch różnych kolorów wierzchołkom grafu, aby żadne sąsiadujące ze sobą nie były pokolorowane tak samo.

Precyzyjnie opisz swój algorytm i oszacuj jego złożoność obliczeniową. Uzasadnij poprawność opisanej metody.

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. $\boxed{\textbf{4 punkty}}$ Niech dany będzie wielomian $w(x) := a_1x/3! - a_3x^3/5! + a_5x^5/7! - a_7x^7/9!$. Rozważmy następujący algorytm obliczania jego wartości w punkcie $x \in \mathbb{R}$:

```
w:=a[7]

  for n from 3 downto 1
    do
      w:=a[2*n-1]-x^2/(2*n+3)/(2*n+2)*w
    od

return(w*x/2/3)
```

   Przyjmując, że a_1, a_3, a_5, a_7 oraz x są liczbami maszynowymi, sprawdź czy algorytm ten jest algorytmem numerycznie poprawnym.

2. $\boxed{\textbf{4 punkty}}$ Ile i jakich operacji arytmetycznych należy wykonać, aby dla danych parami różnych punktów $x_0, x_1, \ldots, x_n$ wyznaczyć ilorazy różnicowe $f[x_0, x_1, \ldots, x_k]$ dla $k = 0, 1, \ldots, n$? Jakie jest zastosowanie ilorazów różnicowych w interpolacji wielomianowej?

3. $\boxed{\textbf{4 punkty}}$ Podaj definicję naturalnej interpolacyjnej funkcji sklejanej trzeciego stopnia (w skrócie: NIFS3). Czy istnieją takie stałe a, b, c, d, że funkcja

$$f(x) = \begin{cases} -x & \text{dla} \quad -2 \le x \le -1, \\ ax^3 + bx^2 + cx + d & \text{dla} \quad -1 \le x \le 1, \\ x & \text{dla} \quad 1 \le x \le 2 \end{cases}$$

   jest NIFS3?

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:

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

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

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