Egzamin licencjacki — 2 lipca 2010

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 — Logika dla informatyków

W algebrze relacji stosowanej w teorii relacyjnych baz danych używa się m.in. operatorów sumy $\cup$, różnicy $-$, iloczynu kartezjańskiego $\times$, selekcji σ_φ i rzutu π_α. Semantyka tych operacji jest taka, że wyrażenia $E_1 \cup E_2$, $E_1 - E_2$ i $E_1 \times E_2$ oznaczają odpowiednio sumę, różnicę i iloczyn kartezjański zbiorów (oznaczanych przez) E_1 i E_2; $\sigma_\varphi(E)$ oznacza zbiór tych elementów zbioru E, które spełniają formułę φ; natomiast $\pi_\alpha(E)$ dla $E \subseteq A_1 \times \ldots \times A_n$ i $\alpha \subseteq \{A_1, \ldots, A_n\}$ oznacza zbiór krotek ze zbioru E obciętych do atrybutów ze zbioru α (np. jeśli $E \subseteq A \times B \times C$ oznacza zbiór $\{\langle a_1, b_1, c_1\rangle, \langle a_2, b_2, c_2\rangle\}$ to $\pi_{\{A,C\}}(E)$ oznacza zbiór $\{\langle a_1, c_1\rangle, \langle a_2, c_2\rangle\}$).

Udowodnij, że suma jest operatorem niezależnym od pozostałych wyżej wymienionych operatorów, tzn. nie istnieje takie wyrażenie E zbudowane z liter R i S oraz operatorów $-, \times, \sigma_\varphi, \pi_\alpha$, które oznacza ten sam zbiór co $R \cup S$.

Wskazówka: Niech R oznacza $\{a\}$ i S oznacza $\{b\}$. Udowodnij przez indukcję, że każde wyrażenie zbudowane z liter R i S oraz operatorów $-, \times, \sigma_\varphi, \pi_\alpha$ oznacza zbiór co najwyżej jednoelementowy.

Matematyka II

1. Znaleźć największy wspólny dzielnik liczb 462 i 910.

2. Rozważamy grupę $\Phi(12)$ liczb względnie pierwszych z 12, wraz z mnożeniem modulo 12.
   (a) Sprawdzić, czy jest to grupa cykliczna.
   (b) Wyznaczyć rzędy elementów tej grupy.

3. Zaproponować algorytm obliczania wartości wyznacznika

$$\begin{vmatrix} a_1 & b_1 & & & \\ c_1 & a_2 & b_2 & & \\ & c_2 & a_3 & \ddots & \\ & & \ddots & \ddots & b_{n-1} \\ & & & c_{n-1} & a_n \end{vmatrix}$$

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:

$$\{S \to aSc\ ,\ S \to R,\ R \to bRc,\ R \to \varepsilon\}$$

a) Czy gramatyka G_1 jest jednoznaczna (odpowiedź uzasadnij)? **(2)**

b) Niech $A_1 = L(G_1) \cap \mathcal{L}((a^*c^*))$. Opisz, jakie słowa należą do A_1. Czy A_1 jest regularny? **(3)**

c) Przedstaw gramatykę bezkontekstową, która generuje język A_1. **(2)**

d) Przedstaw jednoznaczną i niejednoznaczną gramatykę bezkontekstową, która generuje język A_1 (oczywiście jedną z nich może być gramatyka z poprzedniego podpunktu). Uzasadnij niejednoznaczność odpowiedniej gramatyki **(3)**.

Część 2. Załóżmy, że mamy daną listę różnych liczb naturalnych $[v_1, \ldots, v_n]$ oraz liczbę całkowitą K. Ciąg różnych elementów $[e_1, \ldots, e_m]$ nazwiemy K-ciekawym, jeżeli:

1. jest on podlistą listy $[v_1, \ldots, v_n]$
2. $e_1 \circ_1 e_2 \ldots \circ_{k-1} e_k = K$, gdzie $\circ_i$ jest znakiem $+$ albo znakiem $-$ (plus lub minus).

Przykładowo, dla listy `[1,2,3,4,5]` ciąg `[1,2,4,5]` jest 0-ciekawy, bo $1 - 2 - 4 + 5 = 0$. Twoim zadaniem będzie napisanie programu, który dla danej listy i liczby K podaje, jak wiele istnieje K-ciekawych ciągów. Zadanie to ma dwa warianty, z których musisz wybrać jeden. Jeżeli w odpowiedzi znajdą się oba, to będzie sprawdzany tylko pierwszy.

Wariant funkcjonalny

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

a) Napisz funkcję `allMult :: Int -> [[Int]]`, która dla liczby całkowitej K zwraca listę wszystkich list o długości K zawierające jedynie wartości 1 oraz -1. **(3)**

b) Napisz funkcję `dotProduct :: [int] -> [Int] -> Int`, który dla dwóch list liczb całkowitych o tej samej długości zwraca ich iloczyn skalarny, przykładowo `dotProduct [2,3] [4,5]` powinien zwrócić wartość $2 * 4 + 3 * 5$, czyli 23 **(2)**

c) Napisz funkcję `good :: [Int] -> Int -> Bool`, która dla listy `L` i wartości K sprawdza, czy między elementy listy `L` da się postawić znaki $+$ i $-$ w ten sposób, żeby powstałe w ten sposób wyrażenie miało wartość K. **(3)**

d) Jak wykorzystać funkcję `good` do rozwiązania postawionego na początku zadania problemu? Jaka jeszcze funkcja pomocnicza będzie Ci potrzebna? W tym podpunkcie nie musisz nic implementować, wystarczy krótka odpowiedź. **(2)**

Wariant logiczny

W tym wariancie powinieneś używać Prologa.

a) Napisz predykat `allMult(L,N)`, prawdziwy wówczas gdy `L` jest listą o `N` elementach, równych `1` lub `-1`. Predykat powinien móc działać jako generator list o ustalonej długości **(3)**

b) Napisz predykat `dotProduct(A,B,Prod)` prawdziwy, gdy liczba Prod jest iloczynem skalarnym list A i B liczb całkowitych o tej samej długości, przykładowo zachodzi fakt: `dotProduct([2,3],[4,5],23)`. **(2)**

c) Napisz predykat `good(L,K)`, prawdziwy, gdy dla listy `L` i wartości K, da się między elementy listy `L` postawić znaki $+$ i $-$ w ten sposób, żeby powstałe w ten sposób wyrażenie miało wartość K. **(3)**

d) Jak wykorzystać predykat `good` do rozwiązania postawionego na początku zadania problemu? Jaki predykat pomocniczy będzie Ci potrzebny? Z jakiego predykatu standardowego wygodnie skorzystać. W tym podpunkcie nie musisz nic implementować, wystarczy krótka odpowiedź. **(2)**

Matematyka dyskretna

Niech $\delta(G)$ i $\Delta(G)$ oznaczają odpowiednio najmniejszy i największy stopień wierzchołka w grafie G. Niech $n(G)$ oznacza liczbę wierzchołków grafu G. Pokaż, że jeśli G jest dwudzielny, to

$$\delta(G) + \Delta(G) \leq n(G).$$

Algorytmy i struktury danych

Za poprawne rozwiązanie obu zadań z tej części można otrzymać 9 punktów. 3 punkty dają ocenę dostateczną, 4 — dostateczną z plusem, 5 — dobrą, 6 — dobrą z plusem, 7 — ocenę bardzo dobrą.

Zadanie 1: stabilny podział (4 punkty)

W sortowaniu szybkim *quick–sort* korzysta się z procedury *partition*, która dzieli dane względem piwota p (element dzielący) na dwie części: w pierwszej mają się znaleźć elementy $\leq p$ a w drugiej elementy $\geq p$. Napisz własną wersję procedury podziału *partition* $(A[0...n{-}1], p)$, która będzie stabilna, i która na danych umieszczonych w n–elementowej tablicy dokona trójpodziału, czyli podzieli dane na trzy części: w pierwszej znajdą się elementy $< p$, w drugiej $= p$ a w trzeciej $> p$.

- (2.0 pkt.) Napisz w pseudokodzie tę procedurę albo dokładnie ją opisz. Krótko uzasadnij, że działa ona poprawnie.
- (1.0 pkt.) Oszacuj złożoność czasową i pamięciową tej procedury.
- (1.0 pkt.) Napisz co to znaczy, że algorytm jest stabilny. Uzasadnij, że twoja procedura działa stabilnie.

Zadanie 2: spójne składowe w grafie (5 punktów)

Opisz drzewiastą strukturę danych dla zbiorów rozłącznych i zastosuj ją do wyznaczenia spójnych składowych w zadanym grafie prostym $G(V, E)$, gdzie $V = \{0, 1, \ldots, n-1\}$ to zbiór wierzchołków a $E \subseteq \{\{i, j\} \;:\; i, j \in V \wedge i \neq j\}$ to zbiór krawędzi.

- (2.5 pkt.) Opisz budowę drzewiastej struktury danych dla zbiorów rozłącznych. Jak tę strukturę się inicjalizuje i jak działają operacje *union* i *find* na tej strukturze? Na czym polega łączenie według rozmiaru/rangi? Na czym polega kompresja ścieżki podczas wyszukiwania? Jaka jest złożoność czasowa wykonania pojedynczych operacji *union* i *find*? Jaki jest koszt wykonania ciągu operacji *union*/*find* na zbiorach rozłącznych z łączeniem według rozmiarów/rang i kompresją ścieżek?

- (2.5 pkt.) Opisz algorytm wyznaczenia spójnych składowych w grafie G wykorzystujący drzewiastą postać zbiorów rozłącznych z łączeniem według rozmiarów/rang i kompresją ścieżek. Uzasadnij poprawność swojej metody i przeanalizuj jej złożoność obliczeniową.

Metody numeryczne

1. Za pomocą urządzenia pomiarowego wyznaczono wartości $y_0, y_1, \ldots, y_n$ pewnej nieznanej funkcji f w parami różnych punktach $x_0, x_1, \ldots, x_n$ (n jest duże). Z ułożenia punktów (x_i, y_i) na płaszczyźnie wyraźnie widać, że gdyby nie drobne błędy pomiaru, szukana funkcja przypominałaby pewien wielomian niewysokiego (ale nieznanego) stopnia. Zaproponuj dobry sposób wyznaczenia tego wielomianu. Dokładniej, należy:

   (a) nazwać i krótko sformułować teoretyczne zagadnienie, które pasuje do rozpatrywanego problemu (nie bój się używać wzorów),

   (b) podać (jeśli istnieje) wzór na rozwiązanie tego zagadnienia,

   (c) krótko opisać (np. za pomocą kroków) ideę algorytmu, który rozwiąże opisany w zadaniu problem.

2. Układ scalony EL2010 potrafi jedynie dodawać, odejmować i mnożyć liczby rzeczywiste (układ nie ma oddzielnej arytmetyki liczb całkowitych). Czy używając tego układu da się sprawnie i dokładnie obliczać (przybliżać) wartości

   $$\frac{1}{\sqrt{a}},$$

   gdzie $a \in \mathbb{R}_+$? Jeśli uważasz, że nie, to uzasadnij swoją odpowiedź. Jeśli uważasz, że tak, to wyprowadź potrzebne wzory i zapisz swój algorytm w pseudokodzie.