Egzamin licencjacki/inżynierski — 19 czerwca 2016

Informacja dla zdających egzamin na kierunku informatyka:
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.

Informacja dla zdających egzamin na kierunku indywidualne studia informatyczno-matematyczne:
Z sześciu poniższych zestawów zadań (Matematyka I, Programowanie, Matematyka dyskretna, Algorytmy i struktury danych, Metody numeryczne, Języki formalne i złożoność obliczeniowa) należy wybrać i przedstawić na osobnych kartkach rozwiązania trzech zestawów.

Informacja dla wszystkich zdających:
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

Tło zadania: Bardzo częstym błędem w dowodach indukcyjnych dla formuł rachunku zdań jest przeprowadzanie dowodu względem liczby wystąpień spójników logicznych (lub zmiennych) i rozważanie w kroku indukcyjnym formuł z dopisanym jednym spójnikiem i jedną zmienną na końcu. Odpowiada to indukcji strukturalnej dla zbiorów formuł generowanych przez gramatyki podobne do poniższej. W tym zadaniu chcemy pokazać, że takie dowody w ogólnym przypadku są niepoprawne.

Zadanie: Rozważmy zbiór formuł rachunku zdań generowany przez gramatykę

$$\begin{array}{lcl} F & \to & V \mid (F \wedge V) \mid (F \vee V) \\ V & \to & p \mid q \mid r \mid s \end{array}$$

gdzie F jest symbolem startowym. Udowodnij, że istnieje formuła zbudowana ze zmiennych zdaniowych p, q, r, s, spójników koniunkcji $\wedge$ i alternatywy $\vee$ oraz nawiasów, która nie jest równoważna żadnej formule generowanej przez tę gramatykę.

Wskazówka: Formuły generowane produkcją $F \to (F \vee V)$ są spełnione przez wiele wartościowań; formuły generowane produkcją $F \to (F \wedge V)$ są spełnione tylko przez szczególne wartościowania.

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

Programowanie

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

$$S \to SabS, S \to baS, S \to cS,\ \ S \to \varepsilon$$

Gramatyka G_2 z symbolem startowym S nad alfabetem $\{a, b, c\}$ dana jest za pomocą następującego zbioru produkcji:

$$S \to cSc, S \to ab, S \to aaS,\ \ S \to \varepsilon$$

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

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

c) Napisz jednoznaczną gramatykę definiującą $L(G_1)$. Uzasadnij, że rzeczywiście definiuje ona ten język **(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

$$L(G_1) \cap (L(G_2) \cup \mathcal{L}((ab)^*))$$

Możesz używać języka wybranego z następującej listy: C, C++, Java, C#, Python, Ruby, Go, AWK, Rust. **(4)**

Zadanie 2. Będziemy rozważać następujące *warunki elementarne*:

```
<wyrażenie1> <symbol_relacyjny> <wyrażenie2>
```

oraz

```
<zmienna> : <lista_liczb>
```

<zmienna> to zmienna prologowa, <symbol_relacyjny> to jeden z symboli: `==`, `<`, `>`, `=<`, `>=`.

Napisz predykat, `solve/1`, który bierze jako argument listę warunków elementarnych i unifikuje występujące w nim zmienne w ten sposób, by wszystkie warunki były spełnione (predykat powinien znajdywać wszystkie rozwiązania). Możesz założyć, że każda zmienna występuje przynajmniej raz w warunku „z dwukropkiem”. Przykładowe wywołanie:

```
?- solve([X:[1,2,3], Y:[1,2,3], Z:[1,2,3], X + Y > 2*Z, X =< Y]).
```

(6p)

Zadanie 3. **(4p)** Napisz w Haskellu funkcję `compress`, która bierze listę i zamienia w niej każdy podciąg kolejnych równych elementów na pierwszy element tego podciągu. Przykładowo `compress([1,1,1,2,2,3,3,1,1,1]` powinno dać listę `[1,2,3,1]`. Podaj najbardziej ogólny typ tej funkcji.

Matematyka dyskretna

Permutację π zbioru $\{1, 2, 3, \ldots, n\}$ nazywamy spójną gdy nie istnieje $k : 1 < k < n$ dla którego π przekształca podzbiór $\{1, 2, \ldots, k\}$ na siebie. Niech c_n będzie liczbą spójnych permutacji $\{1, 2, 3, \ldots, n\}$. Pokaż, że

$$\sum_{i=1}^{n} c_i(n-i)! = n!.$$

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: najdłuższy podciąg rosnący (5 punktów)

Dany jest n–elementowy ciąg liczb $S = (s_0, s_1, \ldots s_{n-1})$. Należy wyznaczyć najdłuższy podciąg roznący w tym ciągu.

Przedstaw efektywną metodę rozwiązującą to zadanie. Precyzyjnie opisz swój algorytm. Uzasadnij jego poprawność. Oszacuj złożoność obliczeniową opisanego rozwiązania.

Zadanie 2: przeciąganie elementu do korzenia w drzewie BST (4 punkty)

Niech T będzie drzewem BST, w którym każdy węzeł przechowuje jedynie element oraz dwa wskaźniki (na lewego i prawego syna). Zaprojektuj algorytm, który dla zadanej wartości v tak zmodyfikuje drzewo T, że węzeł z wartością v znajdzie się w korzeniu — jeżeli element v znajdował się w drzewie, to należy je odpowiednio przeorganizować zachowując porządek symetryczny w BST, a jeżeli elementu v nie było w drzewie, to należy go najpierw wstawić a potem przeorganizować drzewo. Najważniejsze jest to, aby wysokość drzewa po tej operacji nie wzrosła więcej niż o 1.

Opisz ideę rozwiązania. Zapisz swój algorytm w pseudokodzie wraz z komentarzami. Oszacuj jego złożoność obliczeniową opisanej metody. Uzasadnij, że twoja procedura zachowa pierwotną wysokość drzewa ± 1.

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 dana będzie funkcja ciągła f mająca w przedziale (a, b) dokładnie jedno miejsce zerowe α i spełniająca warunek $f(a)f(b) < 0$. Sformułuj i krótko uzasadnij algorytm znajdowania miejsca zerowego α funkcji f z błędem bezwzględnym mniejszym niż zadane $\varepsilon > 0$.

2. $\boxed{\textbf{4 punkty}}$ Załóżmy, że wartości funkcji ciągłej f znane są jedynie w punktach

$$x_k := \frac{k}{2048} \qquad (k = 0, 1, \ldots, 1024).$$

Zaproponuj metodę obliczania przybliżonej wartości całki $\int_0^{\frac{1}{2}} f(x)\,\mathrm{d}x$.

3. $\boxed{\textbf{4 punkty}}$ Niech dana będzie nieosobliwa macierz trójprzekątniowa $A \in \mathbb{R}^{n\times n}$ postaci

$$A := \begin{bmatrix} b_1 & c_1 & & & & & \\ a_2 & b_2 & c_2 & & & & \\ & a_3 & b_3 & c_3 & & & \\ & & a_4 & b_4 & c_4 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & & & a_n & b_n \end{bmatrix}.$$

Zaproponuj algorytm rozwiązywania układu równań liniowych $Ax = b$, gdzie $b \in \mathbb{R}^n$ jest danym wektorem. Podaj jego złożoność.

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:

$$A = \begin{bmatrix} 5 & 8 & 16 \\ 4 & 1 & 8 \\ -4 & -4 & 11 \end{bmatrix}.$$

Zadanie 2. (5 punktów)
Podać współczynniki rozwinięcia wielomianu $L_3(x) = x^3 - 7x^2 + 12x - 4$ względem bazy (w bazie): $1,\ x-1,\ (x-1)(x-2),\ (x-1)(x-2)(x-3)$.

Języki formalne i złożoność obliczeniowa

- (1,5 punktu) Podaj definicje redukcji wielomianowej oraz klas NP i PSPACE.
- (1,5 punktu) Udowodnij, że problem "Dla danych k deterministycznych automatów skończonych, czy przecięcie języków rozpoznawanych przez te automaty jest niepuste?" jest rozstrzygalny.
- (2 punkty) Określ dokładną złożoność obliczeniową powyższego problemu.

Na zaliczenie trzeba zdobyć 3 punkty. Wtedy ocena to liczba zdobytych punktów