Egzamin licencjacki/inżynierski — 01 lipca 2019

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

Rozważmy niewielki fragment funkcyjnego języka programowania (dokładniej, jest to rachunek lambda z typami prostymi, semantyką dużych kroków i gorliwą strategią ewaluacji).

Syntaktyka. Język (zbiór termów, czyli programów w naszym języku) jest zadany gramatyką

$$t ::= c \mid x \mid \mathsf{fn}\ x \Rightarrow t \mid t_1 t_2$$

gdzie c przebiega stałe całkowite, x przebiega zmienne, $\mathsf{fn}\ x \Rightarrow t$ oznacza funkcję przyjmującą parametr x i zwracającą wartość wyliczoną przez t, natomiast $t_1 t_2$ jest aplikacją funkcji t_1 do argumentu t_2. Przyjmujemy też standardowe konwencje dotyczące zmiennych wolnych i związanych (definicja funkcji $\mathsf{fn}\ x \Rightarrow t$ wiąże zmienną x).

Semantyka. Wartościami (czyli końcowymi wynikami obliczeń) są stałe całkowite i funkcje.

$$v ::= c \mid \mathsf{fn}\ x \Rightarrow t$$

Mamy następujące dwie reguły obliczania:

$$\frac{}{v \Downarrow v} \qquad \frac{t_1 \Downarrow \mathsf{fn}\ x \Rightarrow t_1' \quad t_2 \Downarrow v_2 \quad t_1'[x \mapsto v_2] \Downarrow v}{t_1 t_2 \Downarrow v}$$

gdzie $t[x \mapsto v]$ oznacza wynik podstawienia wartości v za zmienną x w termie t. Pierwsza reguła mówi, że wartości nie trzeba obliczać (ona sama jest już wynikiem obliczeń). Druga reguła mówi, że aby obliczyć wartość aplikacji t_1 do argumentu t_2, najpierw trzeba obliczyć t_1 do wartości, która musi być funkcją, następnie obliczyć wartość argumentu, podstawić wyliczoną wartość argumentu za parametr i na koniec obliczyć wartość termu otrzymanego po podstawieniu. Na przykład $((\mathsf{fn}\ x \Rightarrow \mathsf{fn}\ y \Rightarrow x)7)(\mathsf{fn}\ z \Rightarrow z) \Downarrow 7$ bo $((\mathsf{fn}\ x \Rightarrow \mathsf{fn}\ y \Rightarrow x)7) \Downarrow \mathsf{fn}\ y \Rightarrow 7$, $\mathsf{fn}\ z \Rightarrow z \Downarrow \mathsf{fn}\ z \Rightarrow z$ oraz $(\mathsf{fn}\ y \Rightarrow 7)\ (\mathsf{fn}\ z \Rightarrow z) \Downarrow 7$.

System typów. Typy zadane są gramatyką

$$\tau ::= \mathsf{int} \mid \tau \to \tau$$

Mamy następujące reguły typowania:

$$\frac{}{\Gamma \vdash c : \mathsf{int}} \qquad \frac{}{\Gamma,\ x : \tau \vdash x : \tau} \qquad \frac{\Gamma,\ x : \tau_1 \vdash t : \tau_2}{\Gamma \vdash \mathsf{fn}\ x \Rightarrow t : \tau_1 \to \tau_2} \qquad \frac{\Gamma \vdash t_1 : \tau_1 \to \tau_2 \quad \Gamma \vdash t_2 : \tau_1}{\Gamma \vdash t_1 t_2 : \tau_2}$$

Pierwsza reguła mówi, że przy dowolnych założeniach każda stała całkowita jest typu int. Druga, że przy założeniu $x : \tau$ zmienna x jest typu τ. Trzecia, że jeśli przy dodatkowym założeniu $x : \tau_1$ można udowodnić $t : \tau_2$, to funkcja $\mathsf{fn}\ x \Rightarrow t$ jest typu $\tau_1 \to \tau_2$. Czwarta, że jeśli t_1 jest funkcją typu $\tau_1 \to \tau_2$ a t_2 argumentem typu τ_1 to aplikacja $t_1 t_2$ ma typ τ_2.

Zadanie. Udowodnij indukcyjnie, że dla dowolnego termu t bez zmiennych wolnych, dowolnej wartości v i typu τ, jeśli $[] \vdash t : \tau$ i $t \Downarrow v$, to $[] \vdash v : \tau$. (Tutaj $[]$ oznacza pusty kontekst, czyli brak wszelkich założeń o typach zmiennych.)

Wskazówka. Użyj indukcji względem struktury wyprowadzenia $t \Downarrow v$. W dowodzie może się przydać następujący lemat o podstawianiu, którego nie trzeba dowodzić: *Jeśli* $\Gamma,\ x : \tau_x \vdash t : \tau_t$ *oraz* $\Gamma \vdash v : \tau_x$ *to* $\Gamma \vdash t[x \mapsto v] : \tau_t$.

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:

$$\{A \to a,\ A \to cA,\ A \to Ac,\ A \to cAc,\ B \to b,\ B \to cB,\ B \to Bc,\ B \to cBc,\\ S_1 \to \varepsilon,\ S_2 \to \varepsilon,\ S_1 \to AS_1B,\ S_2 \to BS_2A,\ S \to S_1S_2\} \quad (1)$$

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

b) Czy gramatyka G_2 jest jednoznaczna? Odpowiedź krótko uzasadnij. **(2)**

c) Zaznacz 4 produkcje gramatyki, których usunięcie nie zmienia języka generowanego przez tę gramatykę. **(1)**

d) Opisz jednym zdaniem, jakie słowa należą do zbioru: **(2)**

$$A = L(G_1) \cap \mathcal{L}((a + c)^*(b + c)^*))$$

e) 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 $L(G_1)$. Możesz używać języka wybranego z następującej listy: C, C++, Java, C#, Python, Ruby, Go, AWK, Rust. **(4)**

Zadanie 2. (6p) Mamy zadany *skracający system przepisywania napisów* P, zawierający pary: `(napis1, napis2)`, przy czym `napis2` jest krótszy niż `napis1`. Piszemy $x \Rightarrow y$, jeżeli $x = x_1x_2x_3$, $y = y_1y_2y_3$ oraz $(x_2, y_2) \in P$. Piszemy $x_1 \Rightarrow^* x_n$, jeżeli $x \Rightarrow x_1, x_1 \Rightarrow x_2, \ldots, x_{n-1} \Rightarrow x_n$.

a) Czym w Prologu jest napis: `"Ala ma kota"` (nie musisz podawać dokładnej wartości, ale generalną zasadę interpretacji napisów).

b) Napisz predykat: `wyprowadza(N1, P, N2)`, który dla napisu `N1` i listy par `P` będących skracającym systemem przepisywania, sprawdza, czy $N_1 \Rightarrow^* N_2$

c) Wskaż miejsce w Twojej definicje, w którym, jeżeli wpiszemy odcięcie (`!`) to program zacznie działać dużo szybciej i w wielu sytuacjach niepoprawnie. Wyjaśnij, dlaczego.

Uwaga: jeżeli nie znasz Prologa, możesz za 4p napisać funkcję `wyprowadza(N1, P, N2) --> bool` w języku funkcyjnym (Rocket lub Haskell)

Zadanie 3. (4p) Napisz w języku Haskell lub Rocket funkcję, która dla listy liczb naturalnych o długości 2 lub większej zwraca maksymalną wartość sumy dwóch kolejnych liczb. Możesz definiować pomocnicze funkcje (podając krótkie wyjaśnienie tego, co robią).

Matematyka dyskretna

Dysponujemy trzema rodzajami kafelków: 1×1 w kolorze czerwonym oraz 1×2 w kolorach zielonym i niebieskim. Ile jest różnych sposobów wyłożenia nimi prostokąta $2 \times n$? Odpowiedź uzasadnij.

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: triangulacja wielokąta wypukłego (4 punkty)

Triangulacją wielokąta wypukłego o n wierzchołkach nazywamy zbiór $n-3$ przekątnych takich, że żadne dwie się nie przecinają (pomijając wspólne końce). Zauważmy, że triangulacja definiuje podział wnętrza wielokąta na $n-2$ rozłączne trójkąty. Koszt triangulacji wielokąta zdefiniujmy jako sumę długości przekątnych wyznaczających podział na trójkąty.

Podaj efektywny algorytm wyznaczania triangulacji o minimalnym koszcie dla zadanego wielokąta wypukłego. Opisz ideę algorytmu, uzasadnij jego poprawność, zapisz go w pseudokodzie i oszacuj jego złożoność obliczeniową (czasową i pamięciową).

Zadanie 2: zbiory rozłączne i spójne składowe w grafie (5 punktów)

Opisz budowę *drzewiastej struktury danych dla zbiorów rozłącznych*:

- Jakie podstawowe operacje są wykonywane na zbiorach rozłącznych? Jaki jest stan początkowy zbiorów rozłącznych?
- Jak zbiory rozłączne są reprezentowane w strukturze drzewiastej? Co jest reprezentantem zbioru?
- Na czym polega kompresja ścieżki i kiedy się ją wykonuje? Na czy polega łączenie według rangi? Jakie znaczenie mają rangi w węzłach?
- Napisz w pseudokodzie implementację operacji *union* i *find* z uwzględnieniem rang i kompresją ścieżek.
- Jaki jest koszt czasowy wykonania ciągu m operacji *union* i *find* na n–elementowym uniwersum?

Następnie opisz *algorytm wyznaczania spójnych składowych* w grafie prostym. Wykorzystaj w tym algorytmie drzewiastą strukturę danych dla zbiorów rozłącznych. Przeanalizuj czas działania opisanego algorytmu.

Metody numeryczne

Za rozwiązanie zadania 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. $\boxed{\textbf{4 punkty}}$ Opisz metodę bisekcji.

2. $\boxed{\textbf{3 punkty}}$ Podaj definicję naturalnej interpolacyjnej funkcji sklejanej trzeciego stopnia (NIFS3). Znajdź NIFS3 dla danych

$$\begin{array}{c||c|c|c} x_k & -1 & 0 & 1 \\ \hline y_k & 0 & 1 & 2 \end{array}.$$

3. $\boxed{\textbf{5 punktów}}$ Wykorzystaj schemat Hornera do zaproponowania algorytmu obliczania punktu $P_n(t)$ $(0 \le t \le 1)$ leżącego na krzywej Béziera stopnia n,

$$P_n(t) := \sum_{k=0}^{n} W_k B_k^n(t) \qquad (W_k \in \mathbb{R}^2), \qquad B_k^n(t) := \binom{n}{k} t^k (1-t)^{n-k},$$

i działającego w czasie $O(n)$.

Matematyka II — Algebra

1. Rozważamy przestrzeń wielomianów stopnia ≤ 4 nad $\mathbb{Z}_7$. Niech $T(w) = w' - 2w''$ (pochodne).
   (a) (2p) Wyznaczyć macierz przekształcenia T (w dowolnej bazie).
   (b) (2p) Podać bazę jądra.
   (c) (2p) Znaleźć bazę obrazu.

2. (4p) Niech $A \in \mathbb{R}^{n\times n}$. Wiadomo, że $A^2 = A$. Co można powiedzieć o wartosciach własnych macierzy A? (odpowiedź uzasadnić).

3. (4p) Niech T będzie odwracalnym przekształceniem liniowym z przestrzeni V w przestrzeń W, $T : V \to W$. Udowodnić, że T^{-1} też jest przekształceniem liniowym.

Progi punktowe: 5; 6.5; 8; 10; 12.

Języki formalne i złożoność obliczeniowa

Zadanie. pSAT to wersja problemu SAT, w której dodatkowo zakładamy, że każda klauzula zawiera wyłącznie zmienne zanegowane lub wyłącznie zmienne niezanegowane. Udowodnij, że pSAT jest NP-zupełny.