

**Zadanie z programowania.
Egzamin licencjacki
30 czerwca 2006**

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

$$\{S \rightarrow aSb, S \rightarrow SS, S \rightarrow \varepsilon\}$$

- a) Czy gramatyka ta jest jednoznaczna (odpowiedź uzasadnij)? **(2)**
- b) Niech $A_1 = L(G) \cap \mathcal{L}(a^*b^*)$. Przedstaw możliwie prostą gramatykę bezkontekstową lub wyrażenie regularne generujące A_1 . **(2)**
- c) Niech $A_2 = L(G) \cap \mathcal{L}((ab)^*)$. Przedstaw możliwie prostą gramatykę bezkontekstową lub wyrażenie regularne generujące A_2 . **(2)**
- d) Niech $A_3 = L(G) \cap \mathcal{L}(ab^*)$. Przedstaw możliwie prostą gramatykę bezkontekstową lub wyrażenie regularne generujące A_3 . **(2)**
- e) Dla zbiorów A_1, A_2, A_3 odpowiedz (z krótkim uzasadnieniem) na pytanie: *Czy zbiór jest regularny?* **(2)**

Przypominam, że $L(G)$ to język generowany przez gramatykę G , a $\mathcal{L}(r)$ to język generowany przez wyrażenie regularne r .

Część 2. Będziemy rozważać algorytm sortowania list przez wstawianie, który polega na braniu kolejnych elementów z listy wejściowej i wstawianiu ich do listy wynikowej na odpowiedniej pozycji. Ta część egzaminu ma dwa warianty (do wyboru przez studenta, w przypadku rozwiązania obu sprawdzany jest **tylko** ten wariant, który w odpowiedzi pojawia się jako pierwszy).

Wariant funkcjonalny

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

- a) Napisz funkcję `insert :: [Int] -> Int -> [Int]`, która dla posortowanej listy liczb całkowitych i liczby całkowitej zwraca *posortowaną* listę liczb całkowitych powstałą w wyniku wstawienia elementu (drugiego argumentu) do listy (pierwszego argumentu) **(5)**
- b) Napisz funkcję `in_sort :: [Int] -> [Int]`, która sortuje listę liczb całkowitych opisaną wyżej metodą *sortowania przez wstawianie*. **(5)**

Wariant logiczny

W tym wariantcie powinieneś używać Prolog'a.

- a) Zdefiniuj predykat `insert(+Lista,+Element,-Wynik)`, który przy założeniu, że lista `Lista` jest posortowaną listą liczb, unifikuje `Wynik` z listą powstałą w wyniku wstawienia `Elementu` do `Listy`, w ten sposób, że nie tracimy właściwości posortowania. **(5)**
- b) Zdefiniuj predykat `in_sort(+Lista,-Wynik)`, który będzie używany do sortowania `Listy` opisaną wyżej metodą *sortowania przez wstawianie*. **(5)**