

Zadanie 1

Niech L będzie językiem generowanym przez gramatykę $G = \langle \{X\}, A, P, X \rangle$, gdzie $A = \{a_1, a_2, \dots, a_m\}$ ($m \geq 1$), zaś P składa się z $m+2$ produkcji:
 $X \rightarrow \varepsilon \mid a_1 X a_1 \mid a_2 X a_2 \mid \dots \mid a_m X a_m \mid XX$. Napisz funkcję

Należy(const B: array[1..2*n] of 1..m):Boolean;

przyjmującą wartość true wtw, gdy słowo $a_{B[1]} a_{B[2]} \dots a_{B[m]}$ należy do języka L .

2n

Zadanie 2

Napisz funkcję

ListaWszystkich(d1, d2: drzewo): lista;

która dla dwóch drzew binarnych wyszukiwań $d1$ i $d2$ utworzy listę wszystkich wartości występujących w $d1$ lub $d2$. Wartości w liście powinny być rosnące (powtarzające się elementy powinny pojawić się w liście tylko jednokrotnie). W drzewach i liście `typ=Integer`. Można założyć, że w drzewach nie ma skrajnych elementów typu `Integer`.

Zadanie 3

Napisz procedurę

FormatujLiczby(var f, g:text; k:Integer),

która przepisze plik f , zawierający liczby oddzielone spacjami i końcami wierszy, do pliku g formatując każdą z napotkanych liczb. Liczbą nazywamy maksymalny ciąg znaków następującej postaci: poprzedzony ciągiem (być może pustym) minusów niepusty ciąg cyfr, po którym może wystąpić kropka dziesiętna i znów (być może pusty) ciąg cyfr. Formatowanie polega na:

1. Zastąpieniu każdego niepustego ciągu minusów przez jeden minus.
2. Usunięciu zbędnych wiodących zer
3. Obcięciu liczby po kropce dziesiętnej do co najwyżej k miejsc, $k=0$ jednocześnie usuwając kończące część ułamkową zera. Jeśli tak obcięta część ułamkowa ma wartość zero to w liczbie sformatowanej w ogóle nie powinna pojawić się kropka dziesiętna.

Przykład: dla $k = 2$, liczba -07.3333 sformatuje się do -7.33 , liczba -0.200 do -0.2 , liczba $2.$ do 2 , liczba 1.00003 do 1 .

Podpisane zadania oddajemy na osobnych kartkach. Wszystkie rozwiązania należy uzasadnić i podać koszty wykonania algorytmów.