

Gramatici generative

Enunț

Să se implementeze o gramatică generative $G = (V_N, V_T, S, P)$, astfel:

- se citesc din fișier elementele componente V_N, V_T , simbolul de start și producțiile
- se citește de la tastatură un număr n ,
- apoi se generează n cuvinte în gramatică.

Generarea are loc în modul următor: se pornește de fiecare dată de la simbolul de start. Cât timp există producții aplicabile se aplică random una dintre producțiile aplicabile.

Barem

1. Se cere crearea unei clase Gramatica (alta decât clasa principală - aviz celor care programează Java sau C#!). În funcția principală main se declară un obiect de tip Gramatica.

Membrii clasei vor fi: V_N, V_T, S, P – mulțimea de producții (Puteți folosi o clasă/structură producție).

Metodele clasei:

- (1) verificare() - verificările descrise la punctul 3
- (2) afisare() - afișarea frumoasă a elementelor gramaticii
- (3) generare(optiune) - generarea de cuvinte pornind de la simbolul de start. Dacă opțiunea este 0 se afișază doar cuvântul obținut, dacă opțiunea este 1 se afișază și pașii intermediari.
- (4) Constructor/constructori (cu/fără parametri)
- (5) Metode de get, set pentru membrii.

Citirea poate fi membră a clasei sau poate fi o funcție externă, care să aibă ca parametru o clasă.

Construcția clasei - **1p** (metodele de verificare și de generare sunt punctate separat la punctele 3 și 4)

2. Se cere citire din fișier a elementelor gramaticii: VN, VT, S, P (obligatoriu)

3. Se cere după citire verificarea corectitudinii gramaticii, conform definiției unei gramatici generative, adică

$$(1) VN \cap VT = \emptyset$$

$$(2) S \in VN$$

(3) pentru fiecare regulă membrul stâng conține cel puțin un neterminal

(4) există cel puțin o producție care are în stânga doar S

(5) fiecare producție conține doar elemente din VN și VT

- **2p.**

4. La generarea unui cuvânt în gramatică, producția care se aplică la momentul curent trebuie aleasă random dintre producțiile aplicabile la momentul curent - **5p**

Aici 0.5p pentru utilizarea unei randomizări bune. Trebuie să fie permisă orice gramatică, nu numai gramatici independente de context sau regulate, adică în stânga producției se pot afla mai multe caractere. (Se scad 0.5p din punctaj dacă merge doar pentru anumite tipuri de gramatici).

5. Atunci, când nu mai sunt aplicabile producții, se verifică faptul că avem un cuvânt format doar din terminale – **0.5p** + se verifică faptul că sunt generate cuvinte distincte – **0.5p**

Un punct din oficiu

Pentru un program funcțional care conține doar citirea gramaticii și afișarea elementelor gramaticii -se dă nota 3.

Sugestii de implementare:

- Se pot citi din fișier VN, VT, S, și P astfel: se pun pe prima linie elementele din VN, pe a doua elementele din VT și pe a treia simbolul de start. Apoi se poate pune numărul de producții și pentru fiecare producție stanga spațiu dreapta.
- Pentru afișare: o producție poate fi afișată astfel $S \rightarrow abS$
- VN și VT pot fi considerați vectori de caractere, S poate fi considerat de tip caracter

- Pentru producții se pot folosi doi vectori de string-uri - unul de părți stângi și altul de părți drepte (nerecomandabil) sau o clasă / structură – producție (recomandabil)
- pentru lambda se poate utiliza un caracter rezervat
- pentru punctul 4 - generarea unui cuvânt aplicând în mod aleator reguli se dă un exemplu mai jos

Exemplu:

Consider gramatica cu producțiile

1. $S \rightarrow abS$
2. $S \rightarrow aSAb$
3. $S \rightarrow aA$
4. $A \rightarrow aAb$
5. $A \rightarrow a$

Pornim generearea unui cuvânt de la S.

În acest moment sunt aplicabile producțiile 1, 2, și 3. Se alege random una dintre ele și se aplică, de ex. dacă funcția random alege pe a doua, se obține:

$S \Rightarrow aSAb$.

În acest moment sunt aplicabile toate producțiile. Se alege random una dintre ele, de exemplu ultima (5): $S \Rightarrow aSAb \Rightarrow aSab$.

În același mod putem obține: $S \Rightarrow aSAb \Rightarrow aSab \Rightarrow aAab \Rightarrow aaab$.