

Laborator 3

Generare de şiruri pornind de la o gramatică independentă de context

1. Considerații teoretice:

O gramatică (formală) clasică în sens Chomsky este un cvadruplu:

$$G = (N, T, S, P)$$

unde N, T sunt alfabet(e) (vocabulare) disjuncte ($N \cap T = \emptyset$), $S \in N$ și P este un set finit de perechi (w, v) astfel încât $w, v \in (N \cup T)^*$ și w conține cel puțin o literă din N . Elementele mulțimii N se numesc neterminale (sau simboluri auxiliare), elementele lui T se numesc terminale, S se numește simbol de start sau axiomă. Perechile (w, v) din P se numesc reguli de rescriere sau producții și se notează uzual $w \rightarrow v$. Neterminalele apar cel puțin o dată în partea stângă (în șirul w) a unei producții iar terminalele apar numai în partea dreaptă a producțiilor.

Pentru o gramatică dată $G = (N, T, S, P)$ și două șiruri $w, v \in (N \cup T)^*$ se definește relația numită derivare imediată sau directă (într-un singur pas) notată $w \Rightarrow_G v$ dacă și numai dacă există $u_1, u_2 \in (N \cup T)^*$ astfel încât $w = u_1 \alpha u_2$, $v = u_1 \beta u_2$ și există o regulă $\alpha \rightarrow \beta \in P$. Se spune că α derivează direct în β . Șirul sau simbolul α se poate alege ca cel mai din stânga sau cel mai din dreapta din șirul w , derivarea numindu-se extrem stânga sau extrem dreapta.

Prelungirea tranzitivă a relației \Rightarrow_G se notează cu \Rightarrow_G^+ iar închiderea tranzitivă și reflexivă a relației \Rightarrow_G se notează cu \Rightarrow_G^* . Pentru o gramatică $G = (N, T, S, P)$ și $w, v \in (N \cup T)^*$, o **derivare** $w \Rightarrow_G^* v$ are loc dacă și numai dacă sau $w = v$ sau există $z \in (N \cup T)^*$ astfel încât $w \Rightarrow_G^* z$ și $z \Rightarrow_G v$. Altfel spus w derivează în v în unul sau mai mulți pași.

Limbajul generat de o asemenea gramatică este definit de:

$$L(G) = \{ w : S \Rightarrow_G^* w \text{ și } w \in T^* \}$$

În alte cuvinte $L(G)$ este setul (mulțimea) de șiruri terminale generate de derivări secvențiale plecând de la simbolul de start S . În continuare sunt prezentate câteva exemple de gramatici care generează limbajele din exemplu 1.1.

Exemplu 1.2

Fie $G = (N, T, S, P)$ o gramatică astfel încât:

$$N = \{ S, A, B \}$$

$$T = \{ a, b \}$$

$$P = \{ S \rightarrow aA, A \rightarrow bB, A \rightarrow aA, B \rightarrow b, B \rightarrow bB, B \rightarrow \lambda \}$$

Limbajul generat de G este limbajul (2):

$$L = \{ a^n b^k : n \geq 1 \text{ and } k \geq 1 \}$$

Pentru a genera şirul a^2b^3 derivarea extrem stânga (plecând de la simbolul de start) este:

$$S \rightarrow a\underline{A} \rightarrow aa\underline{A} \rightarrow aab\underline{B} \rightarrow aabb\underline{B} \rightarrow aabbb = a^2b^3$$

Pentru acelaşi şir a^2b^3 derivarea extrem dreapta este aceeaşi pentru că avem un singur simbol neterminal în şirul de derivat.

Gramaticile de tip 2 = **independentă de context: CF** (context-free grammar) au producţiile de forma:

$$A \rightarrow w \quad \text{cu } A \in N, w \in (N \cup T)^*$$

2. Cerinţe:

Programul primeşte ca intrare dintr-un fişier elementele unei gramatici:

- mulţimea neterminalilor
- terminalelor
- producţiile gramaticii
- simbolul de start

2.1. Să se implementeze structurile de date interne pentru o gramatică generală citită dintr-un fişier şi completarea lor cu datele din fişierul de intrare (cu elementele gramaticii)

2.2 Să se implementeze mecanismul de generare de şiruri prin derivări succesive plecând de la simbolul de start. Se vor genera unul sau mai multe şiruri

2.3 Generarea unui şir se va termina într-unul din cazurile:

- şirul este format numai din terminale
- lungimea şirului este mai mare ca şi 60 caractere

3. Indicaţii de implementare:

3.1. Programul trebuie să fie general pentru orice gramatică independentă de context cum ar fi cele două gramatici de mai jos:

3.1.1. gramatica independentă de context: $G1$

$$P = \{E \rightarrow E + T \mid T,$$

$$T \rightarrow T * F \mid F,$$

$$F \rightarrow a \mid (E)\}, \quad N = \{E, T, F\}, \quad T = \{a, +, *, (,)\}, \quad S = E$$

3.1.2. pentru gramatica $G2$

$$P = \{E \rightarrow TE^l, E_l \rightarrow +T E_l \mid \lambda$$

$$T \rightarrow FT^l, T^l \rightarrow *F T^l \mid \lambda$$

$$F \rightarrow a \mid (E), \quad N = \{E, E^l, T, T^l, F\}, \quad T = \{a, +, *, (,)\}, \quad S = E$$

3.2. Se va alege de la început tipul de derivare

- prin derivare extrem dreapta
- prin derivare extrem stânga

3.3. Se va stabili de către programator politica de alegere a variantelor pentru producții cu mai multe alternative variante.

Ex. pt. $E \rightarrow E+T \mid T$ se poate alege

1. $E \rightarrow E+T$

2. $E \rightarrow T$

Se poate opta pentru o secvență 1,2,1,2,1,2.....

Sau un număr aleatoriu dintre 1 și 2

3.4. Procesul de generare începe cu simbolul de start întotdeauna

Se poate implementa în orice limbaj de programare, preferabil C,C++