

PROVE IN ITINERE

Corso Automi, Linguaggi e Traduttori (Laurea e Laurea on-line) 2007-08

Docente: Alberto Pettorossi

Si raccomanda la precisione e la chiarezza dell'esposizione.

Prova 1. Parsing per un linguaggio di tipo 2 esteso.

Scrivere e documentare un programma in Java o in C++ che data una qualsiasi grammatica G_2 di tipo 2 estesa (o context-free estesa) e una parola w , faccia il parsing di w secondo la grammatica G_2 . Si ammette che nella grammatica data ci possano essere anche ε -productions. In particolare, si implementino gli algoritmi: (i) "from above", (ii) "from below", (iii) avoiding unit-production, (iv) avoiding ε -productions, e (v) avoiding left-recursion.

La grammatica viene data in input come una stringa che ne codifica le produzioni. Tale stringa è generata dalla seguente grammatica con assioma G :

$$\begin{aligned} G &\rightarrow H. \\ H &\rightarrow P \mid P ; H \quad (P \text{ denota tutte le produzioni per lo stesso nonterminale } L) \\ P &\rightarrow L \rightarrow R \\ L &\rightarrow A \mid B \mid \dots \mid Z \\ R &\rightarrow e \mid R_{ne} \\ R_{ne} &\rightarrow T_{ne} \mid T_{ne} \mid R_{ne} \quad (T_{ne} \text{ denota il right-hand-side non-empty di una produzione}) \\ T_{ne} &\rightarrow U \mid U T_{ne} \\ U &\rightarrow a \mid b \mid c \mid d \mid f \mid \dots \mid z \mid A \mid B \mid \dots \mid Z \end{aligned}$$

Il simbolo nonterminale a sinistra della prima produzione è l'assioma della grammatica. I simboli nonterminali sono caratteri maiuscoli da **A** a **Z**. I simboli terminali sono caratteri minuscoli da **a** a **z**, escluso il carattere **e** che è riservato per la codifica della stringa vuota ε (cioè la empty string fatta da zero caratteri).

Per esempio, la grammatica di tipo 2 estesa le cui produzioni sono:

$$S \rightarrow aAb \mid SA \quad A \rightarrow aA \mid S \mid \varepsilon$$

e il cui assioma è S , è data come la stringa

S->aAb|SA;A->aA|S|e.

Prova 2. Membership per un linguaggio di tipo 1.

Scrivere e documentare un programma in Java o in C++ che, data una grammatica G_1 di tipo 1 e una parola w , verifichi se w appartiene al linguaggio $L(G_1)$ generato da G_1 .

Si ricorda che una grammatica è di tipo 1 se e solo se le sue produzioni sono del tipo:

$$\text{stringa1} \rightarrow \text{stringa2}$$

ove: (i) stringa1 e stringa2 sono stringhe di terminali e non-terminali, e (ii) la lunghezza della stringa stringa2 è maggiore o uguale a quella della stringa stringa1 .

Si ricorda anche che per la ricerca di un nodo in un albero di possibili alternative, si può usare lo schema di backtracking.

La grammatica viene data in input come una stringa che ne codifica le produzioni. Tale stringa è generata dalla seguente grammatica con assioma G :

$$\begin{aligned} G &\rightarrow H. \\ H &\rightarrow P \mid P ; H \quad (P \text{ denota tutte le produzioni per lo stesso left-hand-side } T_{ne}) \\ P &\rightarrow T_{ne} \rightarrow R_{ne} \\ R_{ne} &\rightarrow T_{ne} \mid T_{ne} \mid R_{ne} \\ T_{ne} &\rightarrow U \mid U T_{ne} \\ U &\rightarrow a \mid b \mid \dots \mid z \mid A \mid B \mid \dots \mid Z \end{aligned}$$

Il simbolo nonterminale a sinistra della prima produzione è l'assioma della grammatica. I simboli nonterminali sono caratteri maiuscoli da **A** a **Z**. I simboli terminali sono caratteri minuscoli da **a** a **z**.

Per esempio, la grammatica di tipo 1 le cui produzioni sono:

$$S \rightarrow ABc \mid a \quad AB \rightarrow acB \quad Bc \rightarrow ab \quad ac \rightarrow cBc \mid bc \mid cb$$

e il cui assioma è S , è data come la stringa

S->ABc|a;AB->acB;Bc->ab;ac->cBc|bc|cb.