

PROVE IN ITINERE

Corso Automi, Linguaggi e Traduttori (Laurea e Laurea on-line). Anno Accademico 2009-10.

Docente: Alberto Pettorossi

Prova 1. Cocke-Younger-Kasami parser per un linguaggio generato da una grammatica di tipo 2 estesa.

Scrivere e documentare un programma in Java o in C++ che, data una qualsiasi grammatica G di tipo 2 estesa e una parola w , verifichi se w appartiene o no al linguaggio generato da G utilizzando l'algoritmo di Cocke-Younger-Kasami. In particolare, data una grammatica di tipo 2 estesa, si implementino:

- (i) l'algoritmo che elimina le unit productions,
- (ii) l'algoritmo che elimina le ε -productions e
- (iii) l'algoritmo che costruisce la grammatica in Chomsky normal form equivalente alla grammatica data.

Il programma deve stampare la recognition matrix relativa alla parola data w .

Si aggiungano alla fine del programma, come commento, alcune prove di esecuzione.

Prova 2. LR(1) parsing.

Scrivere e documentare un programma in Java o in C++ che, data una qualsiasi grammatica G di tipo 2 estesa e una parola w , verifichi se G è una grammatica LR(1) e, nel caso che lo sia, (i) verifichi se w appartiene o no al linguaggio generato da G utilizzando l'algoritmo di LR(1) parsing e (ii) stampi l'automa finito relativo all'LR(1) parsing. Si ricorda che nella grammatica data G ci possono essere anche unit productions e ε -productions.

Si aggiungano alla fine del programma, come commento, alcune prove di esecuzione.

Si supponga che la grammatica della Prova 1 e la grammatica della Prova 2 siano date in input da stringhe che ne codificano le produzioni. Tali stringhe siano generate dalla seguente grammatica con assioma \tilde{G} (ma non tutte le stringhe generate a partire dall'assioma \tilde{G} codificano le produzioni di una grammatica di tipo 2 estesa):

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

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 l'*empty string* fatta da zero caratteri). Per esempio, la grammatica di tipo 2 estesa le cui produzioni sono:

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

e il cui assioma è S , è data in input dalla stringa: **S->aAb|SA;A->aA|e|S**.

Si raccomanda la precisione e la chiarezza nell'esposizione.