

14/2/2019

Esercizio 2. Data la grammatica (le lettere minuscole sono simboli terminali)

$$\begin{aligned} S &\rightarrow Ab \mid Bc \\ A &\rightarrow aA' \\ A' &\rightarrow d \mid \varepsilon \\ B &\rightarrow aB' \\ B' &\rightarrow d \mid \varepsilon \end{aligned}$$

Si dimostri, costruendo l'opportuna tabella, che la grammatica è $LL(1)$. Nel caso in cui non lo sia, verificare se esiste k tale che la grammatica è $LL(k)$. Motivare la risposta.

5/6/2019

Esercizio 2. Data la grammatica (le lettere minuscole sono simboli terminali)

$$\begin{aligned} A &\rightarrow Ba \mid C \\ B &\rightarrow AA \\ C &\rightarrow Cc \mid b \end{aligned}$$

1. Trasformarla, rimuovendo la (mutua) ricorsione sinistra;
2. verificare se la grammatica ottenuta è $LL(1)$ costruendo l'opportuna tabella.

19 Dicembre 2019

Esercizio 1 (6 punti) Scrivere le definizioni **formali** di nullable, first, e follow per grammatiche $LL(1)$.

19 Febbraio 2020

Esercizio 1 (7 punti). Data la grammatica (le lettere minuscole sono simboli terminali, A è il simbolo iniziale)

$$\begin{aligned} A &\rightarrow BC \\ B &\rightarrow aB \mid \varepsilon \\ C &\rightarrow CbB \mid c \end{aligned}$$

Riscrivere la grammatica rimuovendo la ricorsione sinistra e verificare se la grammatica è $LL(1)$ costruendo l'opportuna tabella. Nel caso non lo sia, esiste un k per cui essa è $LL(k)$? Motivare la risposta.

18/9/2020

Esercizio 2 (punti 9) Si consideri la seguente grammatica:

$$\begin{array}{l} S \rightarrow S B \mid y \\ B \rightarrow B x \mid A x \\ A \rightarrow z \mid z S y \end{array}$$

1. verificare, costruendo la tabella che la grammatica non è LL(1);
2. modificare la grammatica per renderla LL(1) e dimostrarlo costruendo la tabella.

28/5/2021

I programmi di un linguaggio di programmazione sono blocchi **Dec Stm** dove

- **Dec** sono sequenze di dichiarazioni di identificatori interi (int);
- **Stm** sono sequenze di comandi che possono essere
 - assegnamenti di una espressione **Exp** a una variabile;
 - iterazioni while (la guardia del condizionale è una espressione intera, la semantica è quella di C).
- **Exp** possono essere costanti intere, identificatori o espressioni con somma.

Esercizi

1. (**punti 6**) definire l'input *completo* di ANTLR per la grammatica del linguaggio di sopra;

9/7/2021

- la sintassi delle espressioni **Exp** è:

$$\text{Exp} : \text{Exp '+' Exp} \mid \text{Exp '-' Exp} \mid X \mid N;$$

dove N sono i naturali, e X sono gli identificatori.

Esercizi

1. (**punti 6**) trasformare la grammatica delle espressioni in modo da eliminare la ricorsione sinistra. Quindi verificare, costruendo l'opportuna tabella, che la grammatica ottenuta sia LL(1). [Assumere che N ed X siano simboli terminali.]