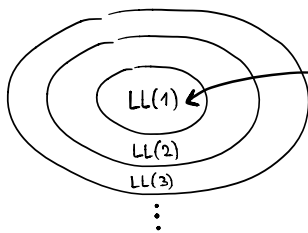


Stavamo parlando di top-down parsing predittivo.



Dal punto di vista della programmazione, ci si sforza di progettare delle grammatiche di questo tipo.

Esercizio

Grammatica che genera linguaggio delle espressioni aritmetiche

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid \text{id}$

First Follow:
 $E: (, \text{id} \quad \$, +,)$
 $T: (, \text{id} \quad *$
 $F: (, \text{id}$

	id	+	*	()	\$
E	$E \rightarrow E + T$			$E \rightarrow E + T$		
	$E \rightarrow T$			$E \rightarrow T$		
T	$T \rightarrow T * F$			$T \rightarrow T * F$		
	$T \rightarrow F$			$T \rightarrow F$		
F	$F \rightarrow \text{id}$			$F \rightarrow (E)$		

Riprendiamo il concetto di **ricorsione a sinistra**.

La grammatica di prima presenta ricorsione immediata a sinistra:

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid \text{id}$

Un'altra grammatica:

$S \rightarrow Bb$

$B \rightarrow Sa$

Mostra ricorsione in due passi:

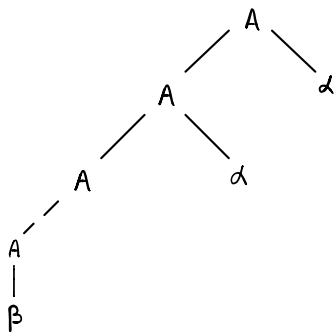
$S \Rightarrow Bb \Rightarrow Sab$

Ossia, $S \Rightarrow S ab$

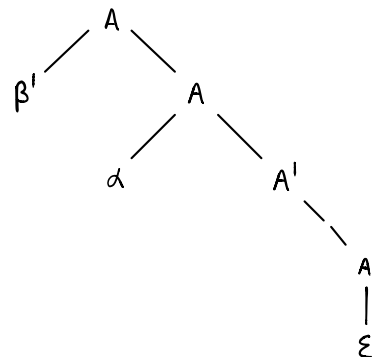
Come modificare una grammatica per eliminarne la ricorsione a sinistra ma mantenendola equivalente?

Focus: vediamo il caso delle grammatiche immediatamente ricorsive a sinistra.

$A \rightarrow A\alpha \mid \beta$ con $\beta \neq A\gamma$ per qualunque γ (gamma), e con $\alpha \neq \varepsilon$



Tentiamo: aggiungo A' , simbolo nuovo.
 $A \rightarrow \beta A'$
 $A' \rightarrow \alpha A' \mid \varepsilon$



In generale, la ricorsione immediata è in una forma

$A \rightarrow A\alpha_1 \mid \dots \mid A\alpha_n \mid \beta_1 \mid \dots \mid \beta_k$ con $\beta_i \neq A\gamma_i \quad \forall i = 1 \dots k$ e $\alpha_j \neq \varepsilon \quad \forall j = 1 \dots n$

Per eliminare la ricorsione a sinistra:

$A \rightarrow \beta_1 A' \mid \dots \mid \beta_k A'$

$A' \rightarrow \alpha_1 A' \mid \dots \mid \alpha_n A' \mid \varepsilon$

Focus: Vediamo una grammatica con ricorsione non immediata.

$A \rightarrow Ba \mid b$

$B \rightarrow Bc \mid Ad \mid b$

- Occupiamoci della produzione $B \rightarrow Bc$: la teniamo, non ci dà troppo fastidio perché sappiamo come gestire la ricorsione immediata: $B \rightarrow Bc \mid \text{Bad} \mid \text{bd} \mid b$
- La produzione scomoda è $B \rightarrow Ad$, che trasformiamo in:
 $B \rightarrow \text{bdB}' \mid \text{bB}'$
 $B' \rightarrow \text{cB}' \mid \text{adB}' \mid \varepsilon$

Esercizio

Eliminare la ricorsione immediata a sinistra e determinare se la grammatica trasformata è LL(1)

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid \text{id}$

Diventa:

$E \rightarrow \text{TE}'$

$E' \rightarrow +\text{TE}' \mid \varepsilon$

$T \rightarrow \text{FT}'$

$T' \rightarrow *\text{FT}' \mid \varepsilon$

$F \rightarrow (E) \mid \text{id}$

La grammatica trasformata è LL(1) in quanto, se facessimo la tabella di parsing top-down predittivo, non vi sarebbero entry multiply-defined (è in linea con le nostre attese).

Vedremo che:

TOP DOWN: è fastidiosa la ricorsione a sinistra.

BOTTOM UP: è fastidiosa la ricorsione a destra.