

# Parsing bottom up

---

a.a. 2020-2021

Corso di Fondamenti di Informatica - 1 modulo

Corso di Laurea in Informatica

Università di Roma "Tor Vergata"

Prof. Giorgio Gambosi



# Parsing bottom up

Costruzione dell'albero sintattico dal basso verso l'alto.

Equivalentemente, costruzione di una derivazione destra (in ordine inverso rispetto alla derivazione stessa).

Parsing LR( $k$ ):

- Left-to-right: la derivazione è calcolata da sinistra a destra (dalla prima produzione applicata all'ultima)
- Rightmost derivation: la derivazione calcolata è destra
- $k$  simboli (di **look-ahead**) da considerare

Operazioni base del parsing LR:

Una sottostringa (**handle**) della forma di frase attuale  $\alpha$ , corrispondente alla parte destra di una produzione, viene sostituita dalla parte sinistra.

- Forma di frase attuale  $\alpha = \delta\eta\zeta$ , con  $\delta, \zeta \in (V_T \cup V_N)^*$ , e  $\eta \in (V_T \cup V_N)^* V_N (V_T \cup V_N)^*$
- Produzione  $A \longrightarrow \eta \in P$

Riduzione: la nuova forma di frase è  $\delta A \zeta$

# Esempio di parsing bottom up

Consideriamo ancora la grammatica

$$\begin{aligned}E &\longrightarrow TE' \\E' &\longrightarrow +TE' \mid \varepsilon \\T &\longrightarrow FT' \\T' &\longrightarrow *FT' \mid \varepsilon \\F &\longrightarrow (E) \mid \text{id}\end{aligned}$$

e la stringa  $\text{id} + \text{id} * \text{id}$ .

# Esempio di parsing bottom up

Assumiamo di poter individuare sempre la prima handle da sinistra nella forma di frase attuale.

Handle	Produzione	Result
<b>id</b> +id*id\$	$F \rightarrow id$	F+id*id
F+id*id\$	$T' \rightarrow \varepsilon$	FT'+id*id
<b>FT'</b> +id*id\$	$T \rightarrow FT'$	T+id*id
T+ <b>id</b> *id\$	$F \rightarrow id$	T+F*id
T+F* <b>id</b> \$	$F \rightarrow id$	T+F*F
T+F*F\$	$T' \rightarrow \varepsilon$	T+F*FT'
T+F* <b>FT'</b> \$	$T' \rightarrow *FT'$	T+FT'
T+ <b>FT'</b> \$	$T \rightarrow FT'$	T+T
T+T\$	$E' \rightarrow \varepsilon$	T+TE'
T+ <b>TE'</b> \$	$E' \rightarrow +TE'$	TE'
<b>TE'</b> \$	$E \rightarrow TE'$	E

# Esempio di parsing bottom up

La sequenza di produzioni individuate, lette al contrario (dal basso in alto), forniscono la derivazione destra della stringa

$$\begin{aligned} E &\Longrightarrow TE' \Longrightarrow T + TE' \Longrightarrow T + T \Longrightarrow T + FT' \Longrightarrow T + F * FT' \Longrightarrow T + F * \\ F &\Longrightarrow T + F * id \Longrightarrow T + id * id \Longrightarrow FT' + id * id \Longrightarrow F + id * id \Longrightarrow id + id * id \end{aligned}$$

Problema: come individuare handle e riduzione?

# Parsing shift-reduce

Un parser **shift-reduce** effettua parsing bottom up utilizzando:

- una pila di simboli della grammatica
- un buffer di input, in cui è contenuta la parte dell'input ancora da leggere

Il carattere  $\$ \notin V_T$  è utilizzato per marcare il fondo della pila e la fine della stringa di input, per cui inizialmente, se  $w$  è la stringa:

- la pila contiene  $\$$
- il buffer contiene  $w\$$

Pila

$\$$

Input

$w\$$

# Parsing shift-reduce: operazioni

Reduce: i simboli in cima alla pila corrispondono ad una handle per una produzione  $A \rightarrow \alpha$ . Viene effettuata una reduce, passando da

Pila	Input		Pila	Input
$\$ \beta \alpha$	$w \$$	a	$\$ \beta A$	$w \$$

Shift: i simboli in cima alla pila non corrispondono ad una handle. Il prossimo simbolo in input viene posto sulla pila, passando da

Pila	Input		Pila	Input
$\$ \alpha$	$a w \$$	a	$\$ \alpha a$	$w \$$



# Parsing shift-reduce: operazioni

Accept: la pila contiene  $\$S$ , dove  $S$  è l'assioma della grammatica, il buffer di input contiene  $\$$  (la stringa è terminata). La stringa è accettata

Error: non ci sono altre azioni eseguibili nella configurazione attuale di pila e buffer. La stringa è rifiutata

# Parsing shift-reduce

Pila	Input	Azione
\$	id+id*id\$	shift
\$ <b>id</b>	+id*id\$	reduce $F \rightarrow id$
\$F	+id*id\$	reduce $T' \rightarrow \varepsilon$
\$ <b>FT'</b>	+id*id\$	reduce $T \rightarrow FT'$
\$T	+id*id\$	shift
\$T+	id*id\$	shift
\$T+ <b>id</b>	*id\$	reduce $F \rightarrow id$
\$T+F	*id\$	shift
\$T+F*	id\$	shift
\$T+F* <b>id</b>	\$	reduce $F \rightarrow id$
\$T+F*F	\$	reduce $T' \rightarrow \varepsilon$
\$T+F* <b>FT'</b>	\$	reduce $T' \rightarrow *FT'$

# Parsing shift-reduce

Pila	Input	Azione
$\$T + FT'$	\$	reduce $T \rightarrow FT'$
$\$T + T$	\$	reduce $E' \rightarrow \varepsilon$
$\$T + TE'$	\$	reduce $E' \rightarrow +TE'$
$\$TE'$	\$	reduce $E \rightarrow TE'$
$\$E$	\$	accept

Si può mostrare che l'handle, in un parsing bottom up, apparirà sempre in cima alla pila.

# Riconoscimento di handle

Definizione di un automa a stati finiti  $\mathcal{A}$  che riconosce handle.

- Una operazione **shift** corrisponde ad una transizione di  $\mathcal{A}$  da uno stato al successivo
- Una operazione **reduce** corrisponde al riconoscimento di una handle

Come definire l'automa  $\mathcal{A}$ ?

- Alfabeto?
- Stati?
- Funzione di transizione?