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

Riduzioni

Operazioni base del parsing LR:

Una sottostringa (handle) della forma di frase attuale α, 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

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

e la stringa id + id $*$ id.

Esempio di parsing bottom up

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

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

Esempio di parsing bottom up

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

$$E \Longrightarrow TE' \Longrightarrow T+TE' \Longrightarrow T+T \Longrightarrow T+FT' \Longrightarrow T+F*FT' \Longrightarrow T+F*$$
$$F \Longrightarrow T+F*\texttt{id} \Longrightarrow T+\texttt{id}*\texttt{id} \Longrightarrow FT'+\texttt{id}*\texttt{id} \Longrightarrow F+\texttt{id}*\texttt{id} \Longrightarrow \texttt{id}+\texttt{id}*\texttt{id}$$

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 $A \longrightarrow \alpha$. Viene effettuata una reduce, passando da

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

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

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 tringa è rifiutata

Parsing shift-reduce

Pila	Input	Azione
\$	`id+id*id$`	shift
\$id	`+id*id$`	reduce $F \longrightarrow$ `id`
\$$F$	`+id*id$`	reduce $T' \longrightarrow \varepsilon$
\$$FT'$	`+id*id$`	reduce $T \longrightarrow FT'$
\$$T$	`+id*id$`	shift
\$$T$+	`id*id$`	shift
\$$T$ + id	`*id$`	reduce $F \longrightarrow$ `id`
\$$T + F$	`*id$`	shift
\$$T + F*$	`id$`	shift
\$$T + F *$ id	`$`	reduce $F \longrightarrow$ `id`
\$$T + F * F$	`$`	reduce $T' \longrightarrow \varepsilon$
\$$T + F*FT'$	`$`	reduce $T' \longrightarrow *FT'$

Parsing shift-reduce

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

Parsing shift-reduce

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

Parsing shift-reduce

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

Derivazione destra:

$$S \stackrel{*}{\Longrightarrow} \alpha Az \Longrightarrow \alpha\beta Byz \Longrightarrow \alpha\beta\gamma yz$$

dove si sono applicate le produzioni $A \longrightarrow \beta By$ e $B \longrightarrow \gamma y$, con $\alpha, \beta, \gamma \in (V_T \cup V_N)^*$ e $y, z \in V_T^*$

Nel parsing, si ha la corrispondente sequenza di azioni

Pila	Input	Azione
$\$\alpha\beta\gamma$	yz$	reduce $B \longrightarrow \gamma$
$\$\alpha\beta B$	z$	shift y
$\$\alpha\beta By$	z$	reduce $A \longrightarrow \beta y$
$\$\alpha A$	z$	

Parsing shift-reduce

Derivazione destra:

$$S \overset{*}{\Longrightarrow} \alpha Bx Az \Longrightarrow \alpha Bxyz \Longrightarrow \alpha\gamma xyz$$

dove si sono applicate le produzioni $A \longrightarrow y$ e $B \longrightarrow \gamma y$, con $\alpha, \gamma \in (V_T \cup V_N)^*$ e $x, y, z \in V_T^*$

Nel parsing, si ha la corrispondente sequenza di azioni

Pila	Input	Azione
$\$\alpha\gamma$	xyz$	reduce $B \longrightarrow \gamma$
$\$\alpha B$	xyz$	shift xy
$\$\alpha Bxy$	z$	reduce $A \longrightarrow y$
$\$\alpha BxA$	z$	

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?