Bottom-up parsing

Clase 25

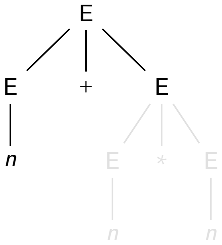
IIC 2223

Prof. Cristian Riveros

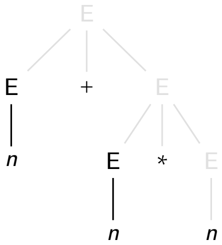
Dos estrategias para hacer parsing

$$E \;\rightarrow\; E + E \;\mid\; E * E \;\mid\; n$$

Top-down parsing



Bottom-up parsing



Outline

Advertencias

Bottom-up parsing

Prefijos viables

Outline

Advertencias

Bottom-up parsing

Prefijos viables

ADVERTENCIA! (1)

Cambio en la notación de stack

Notación (desde ahora y hasta el término del curso)

Para un stack $q_0 \dots q_{n-1} q_n$, usaremos a
q_n como el **tope de stack** y $q_0 \dots q_{n-1}$ como la cola de stack.

Para un PDA $\mathcal{P}$ y una transición $(\overbrace{p_0 \dots p_i}^{\alpha}, a, \overbrace{q_0 \dots q_j}^{\beta})$ de $\mathcal{P}$, p_i es el símbolo en el tope del stack y q_j será el símbolo del tope del stack resultante.

La relación $\vdash_{\mathcal{P}}$ de **siguiente-paso** quedará como:

$$(\gamma \cdot \alpha, a \cdot u) \ \vdash_{\mathcal{P}} \ (\gamma \cdot \beta, u)$$

Como **recordatorio**, (algunas veces) marcaremos el tope del stack:

$$q_0 \dots q_{n-1} q_n^{\downarrow}$$

ADVERTENCIA! (2)

Gramática aumentada

Sea $\mathcal{G} = (V, \Sigma, P, S)$ una gramática libre de contexto reducida.

Definición

Se define la **gramática aumentada** de $\mathcal{G}$ como:

$$\mathcal{G}' = (V \cup \{S'\}, \Sigma, P \cup \{S' \to S\}, S')$$

tal que S' es una variable nueva con $S' \notin V$.

> *"Usaremos S' para saber cuando hemos llegado al final de una derivación."'*

Desde ahora, trabajaremos siempre con una gramática aumentada.

Outline

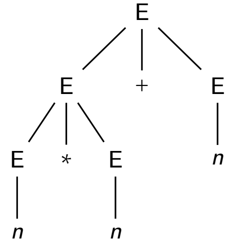
Advertencias

Bottom-up parsing

Prefijos viables

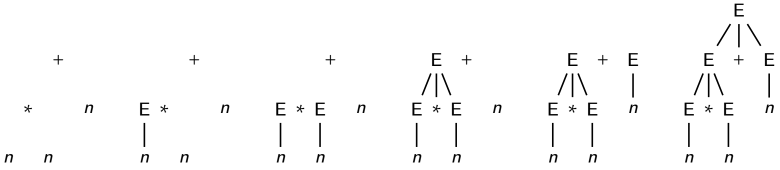
Derivación por la derecha y parsing

$$E \;\rightarrow\; E+E \;\mid\; E*E \;\mid\; n$$



$$E \underset{\mathrm{rm}}{\Rightarrow} E+E \underset{\mathrm{rm}}{\Rightarrow} E+n \underset{\mathrm{rm}}{\Rightarrow} E*E+n \underset{\mathrm{rm}}{\Rightarrow} E*n+n \underset{\mathrm{rm}}{\Rightarrow} n*n+n$$

$$n*n+n \underset{\mathrm{rm}}{\Leftarrow} E*n+n \underset{\mathrm{rm}}{\Leftarrow} E*E+n \underset{\mathrm{rm}}{\Leftarrow} E+n \underset{\mathrm{rm}}{\Leftarrow} E+E \underset{\mathrm{rm}}{\Leftarrow} E$$



Derivación por la derecha y parsing

$$n * n + n \underbrace{\underset{\text{rm}}{\Leftarrow}}_{E \to n} E * n + n \underbrace{\underset{\text{rm}}{\Leftarrow}}_{E \to n} E * E + n \underbrace{\underset{\text{rm}}{\Leftarrow}}_{E \to E * E} E + n \underbrace{\underset{\text{rm}}{\Leftarrow}}_{E \to n} E + E \underbrace{\underset{\text{rm}}{\Leftarrow}}_{E \to E + E} E$$

Stack	**Input**	**Operación**
	$n * n + n$	
n	$* n + n$	`shift`
E	$* n + n$	`reduce` $E \to n$
$E *$	$n + n$	`shift`
$E * n$	$+ n$	`shift`
$E * E$	$+ n$	`reduce` $E \to n$
E	$+ n$	`reduce` $E \to E * E$
$E +$	n	`shift`
$E + n$	.	`shift`
$E + E$	.	`reduce` $E \to n$
E	.	`reduce` $E + E$

Los **reduce** nos entregan una derivación por la derecha (invertida).

Bottom-up parser

Sea $\mathcal{G}' = (V, \Sigma, P, S')$ una gramática libre de contexto aumentada.

Definición

El **apilador bottom-up** de $\mathcal{G}$ (bottom-up-PDA) es un PDA alternativo:

$$\mathcal{P}_{\uparrow} = (Q, \Sigma, \Delta, q_0, F)$$

- $Q = V \cup \Sigma \cup \{\$\}$
- $q_0 = \$$
- $F = \{S'\}$

Bottom-up parser

Sea $\mathcal{G}' = (V, \Sigma, P, S')$ una gramática libre de contexto aumentada.

Definición

El **apilador bottom-up** de $\mathcal{G}$ (bottom-up-PDA) es un PDA alternativo:

$$\mathcal{P}_\uparrow \;=\; (\underbrace{V \cup \Sigma \cup \{\$\}}_{Q}, \Sigma, \Delta, \underbrace{\$}_{q_0}, \underbrace{\{S'\}}_{F})$$

Tres tipos de transiciones en $\Delta \subseteq Q^+ \times (\Sigma \cup \{\epsilon\}) \times Q^*$:

Shift:	$q \xrightarrow{a} q\,a$	para $q \in V \cup \Sigma \cup \{\$\}$ y $a \in \Sigma$
Reduce:	$\alpha \xrightarrow{\epsilon} X$	si $X \to \alpha \in P$
Termino:	$\$S' \xrightarrow{\epsilon} S'$	

Bottom-up parser

Shift: $q \xrightarrow{a} q\,a$ para $q \in V \cup \Sigma \cup \{\$\}$ y $a \in \Sigma$

Reduce: $\alpha \xrightarrow{\epsilon} X$ si $X \to \alpha \in P$

Termino: $\$S' \xrightarrow{\epsilon} S'$

Ejemplo: $E \;\to\; E + E \;|\; E * E \;|\; n$

$\$$	$n * n + n$	
$\$n$	$* n + n$	`shift`
$\$E$	$* n + n$	`reduce` $n \xrightarrow{\epsilon} E$
$\$E *$	$n + n$	`shift`
$\$E * n$	$+ n$	`shift`
$\$E * E$	$+ n$	`reduce` $n \xrightarrow{\epsilon} E$
$\$E$	$+ n$	`reduce` $E * E \xrightarrow{\epsilon} E$
$\$E +$	n	`shift`
$\$E + n$	.	`shift`
$\$E + E$	.	`reduce` $n \xrightarrow{\epsilon} E$
$\$E$	.	`reduce` $E + E \xrightarrow{\epsilon} E$
$\$S'$	.	`reduce` $E \xrightarrow{\epsilon} S'$
S'	.	`termino`

Correctitud de bottom-up parser

Teorema

Sea $\mathcal{G} = (V, \Sigma, P, S)$ una gramática libre de contexto. Entonces:

$$\mathcal{L}(\mathcal{G}) \; = \; \mathcal{L}(\mathcal{P}_\uparrow)$$

Demostración ($\subseteq$)

PD: Si $S' \underset{\mathrm{rm}}{\overset{\star}{\Rightarrow}} \alpha A y \underset{\mathrm{rm}}{\overset{\star}{\Rightarrow}} xy$, entonces $(\$, xy) \vdash^{*}_{\mathcal{P}_\uparrow} (\$\alpha A, y)$.

Inducción en el largo de la derivación $\alpha A y \underset{\mathrm{rm}}{\overset{\star}{\Rightarrow}} xy$.

Caso inductivo: Suponemos que $S' \underset{\mathrm{rm}}{\overset{\star}{\Rightarrow}} \alpha A y \underset{\mathrm{rm}}{\Rightarrow} \alpha\beta y \underset{\mathrm{rm}}{\overset{\star}{\Rightarrow}} xy$.

Suponga que $\beta = \gamma B v$ y $x = uv$. Entonces:

$$\begin{array}{rcll}
(\$, \underbrace{uv}_{x} y) & \vdash^{*}_{\mathcal{P}_\uparrow} & (\$\alpha\gamma B, vy) & \text{(por HI)} \\
 & \vdash^{*}_{\mathcal{P}_\uparrow} & (\$\alpha\gamma B v, y) & \text{(con } \texttt{shift}\text{)} \\
 & \vdash^{*}_{\mathcal{P}_\uparrow} & (\$\alpha A, y) & \text{(con } \texttt{reduce}\ A \to \overbrace{\gamma B v}^{\beta}\text{)}
\end{array}$$

Correctitud de bottom-up parser

Teorema

Sea $\mathcal{G} = (V, \Sigma, P, S)$ una gramática libre de contexto. Entonces:

$$\mathcal{L}(\mathcal{G}) \;=\; \mathcal{L}(\mathcal{P}_{\uparrow})$$

Demostración ($\supseteq$)

PD: Si $(\$, xy) \vdash^{*}_{\mathcal{P}_{\uparrow}} (\$\alpha A, y)$, entonces $\alpha A y \underset{\mathrm{rm}}{\overset{\star}{\Rightarrow}} xy$.

Inducción en el largo de pasos $(\$, xy) \vdash^{*}_{\mathcal{P}_{\uparrow}} (\$\alpha A, y)$.

Caso inductivo: Suponemos que $(\$, xy) \vdash^{*}_{\mathcal{P}_{\uparrow}} (\$\alpha\beta, y) \vdash_{\mathcal{P}_{\uparrow}} (\$\alpha A, y)$.
Sea $\alpha \cdot \beta = \gamma B w$ con $w = a_1 \dots a_k \in \Sigma^*$. Entonces:

$$(\$, xy) \vdash^{*} \overbrace{(\$\gamma B, a_1 \dots a_k y) \vdash (\$\gamma B a_1, a_2 \dots a_k y) \vdash \;\dots}^{\texttt{shifts}}$$
$$\vdash (\$\gamma B a_1 \dots a_k, y) \vdash (\$\alpha A, y)$$

Por HI: $\gamma B \cdot wy = \alpha\beta y \underset{\mathrm{rm}}{\overset{\star}{\Rightarrow}} xy$. Como $A \to \beta$, entonces $\alpha A y \underset{\mathrm{rm}}{\Rightarrow} \alpha\beta y \underset{\mathrm{rm}}{\overset{\star}{\Rightarrow}} xy$.

Correctitud de bottom-up parser

Teorema

Sea $\mathcal{G} = (V, \Sigma, P, S)$ una gramática libre de contexto. Entonces:

$$\mathcal{L}(\mathcal{G}) \;=\; \mathcal{L}(\mathcal{P}_\uparrow)$$

Corolarios

1. Si $(\$, xy) \vdash^*_{\mathcal{P}_\uparrow} (\$\alpha, y) \vdash^*_{\mathcal{P}_\uparrow} (\$S', \epsilon)$, entonces $S' \underset{\text{rm}}{\overset{\star}{\Rightarrow}} \alpha y \underset{\text{rm}}{\overset{\star}{\Rightarrow}} xy$.

2. Si $(\$, w) \vdash^*_{\mathcal{P}_\uparrow} (\$S', \epsilon)$ y $\alpha_1 \xrightarrow{\epsilon} X_1, \ldots, \alpha_n \xrightarrow{\epsilon} X_n \in \Delta$ son las **transiciones "reduce"** durante la ejecución, entonces:

$$X_n \to \alpha_n, \ldots, X_1 \to \alpha_1$$

   es la secuencia de reglas de una **deriv. por la derecha** de $\mathcal{G}$ sobre w.

Demostración: ejercicio.

Outline

Advertencias

Bottom-up parsing

Prefijos viables

¿podemos usar el bottom-up PDA para hacer parsing?

Problemas

1. Conflicto **Shift-Reduce**.

Ejemplo: ¿hacemos shift o reduce?

$$\begin{array}{rcl} S & \rightarrow & ab \mid A \\ A & \rightarrow & a \end{array}$$

Stack	**Input**	Operaciones
\$	ax	
\$$a$	x	`shift`
?	?	?

¿podemos usar el bottom-up PDA para hacer parsing?

Problemas

1. Conflicto **Shift-Reduce**.
2. Conflicto **Reduce-Reduce**.

Ejemplo: ¿con cuál regla hacemos reduce?

$$
\begin{array}{rcl}
S & \rightarrow & Ac \mid aBd \\
A & \rightarrow & ab \\
B & \rightarrow & b
\end{array}
$$

Stack	**Input**	Operaciones
\$	abx	
\$$a$	bx	`shift`
\$$ab$	x	`shift`
?	?	?

¿podemos usar el bottom-up PDA para hacer parsing?

Problemas

1. Conflicto **Shift-Reduce**.
2. Conflicto **Reduce-Reduce**.
3. Configuraciones **no-viables**.

(Prefijos viables, reducibles y handles)

Sea $\mathcal{G} = (V, \Sigma, P, S)$ una gramática y $\mathcal{G}'$ su gramática aumentada.

Definiciones

- $\alpha \in (V \cup \Sigma)^*$ es un **prefijo viable** de $\mathcal{G}$ ssi
  existe una derivación $S' \overset{\star}{\underset{\text{rm}}{\Rightarrow}} \alpha\beta w$ tal que $\beta \in (V \cup \Sigma)^*$ y $w \in \Sigma^*$.

¿cuáles son prefijos viables de $\mathcal{G}$?

$$E \rightarrow E + E \mid E * E \mid n$$

- $E + E * E$ ✓
- $E + E *$ ✓
- $n + n *$ ✓
- $E\ E$ ✗
- $E + n * E$ ✗

(Prefijos viables, reducibles y handles)

Sea $\mathcal{G} = (V, \Sigma, P, S)$ una gramática y $\mathcal{G}'$ su gramática aumentada.

Definiciones

- $\alpha \in (V \cup \Sigma)^*$ es un **prefijo viable** de $\mathcal{G}$ ssi existe una derivación $S' \underset{\text{rm}}{\overset{\star}{\Rightarrow}} \alpha\beta w$ tal que $\beta \in (V \cup \Sigma)^*$ y $w \in \Sigma^*$.
- $\alpha \cdot \beta \in (V \cup \Sigma)^*$ es **reducible** a $\alpha \cdot X$ ssi existe una derivación $S' \underset{\text{rm}}{\overset{\star}{\Rightarrow}} \alpha X w \underset{\text{rm}}{\Rightarrow} \alpha\beta w$ con $w \in \Sigma^*$.

  En cuyo caso, decimos que $X \to \beta$ es un **handle** de $\alpha\beta$.

¿cuáles son reducciones válidas con sus resp. handles?

$$E \to E + E \mid E * E \mid n$$

- $E + E * E$ es reducible a $E + E$. ✓
- $E + E + n$ es reducible a $E + E + E$. ✓
- $n + E + E$ es reducible a $n + E$. ✗

(Prefijos viables, reducibles y handles)

Sea $\mathcal{G} = (V, \Sigma, P, S)$ una gramática y $\mathcal{G}'$ su gramática aumentada.

Definiciones

- $\alpha \in (V \cup \Sigma)^*$ es un **prefijo viable** de $\mathcal{G}$ ssi
  existe una derivación $S' \underset{\text{rm}}{\overset{\star}{\Rightarrow}} \alpha\beta w$ tal que $\beta \in (V \cup \Sigma)^*$ y $w \in \Sigma^*$.
- $\alpha \cdot \beta \in (V \cup \Sigma)^*$ es **reducible** a $\alpha \cdot X$ ssi
  existe una derivación $S' \underset{\text{rm}}{\overset{\star}{\Rightarrow}} \alpha X w \underset{\text{rm}}{\Rightarrow} \alpha\beta w$ con $w \in \Sigma^*$.

  En cuyo caso, decimos que $X \to \beta$ es un **handle** de $\alpha\beta$.
- $\alpha \cdot \beta \in (V \cup \Sigma)^*$ es un **prefijo reducible** ssi
  $\alpha \cdot \beta$ es un prefijo viable y existe X tal que $\alpha \cdot \beta$ es reducible a $\alpha \cdot X$.

Si $\alpha\beta$ es un prefijo viable y $X \to \beta$, ¿es $\alpha\beta$ reducible a αX?

¿podemos usar el bottom-up PDA para hacer parsing?

Problemas

1. Conflicto **Shift-Reduce**.
2. Conflicto **Reduce-Reduce**.
3. Configuraciones **no-viables**.

Ejemplo: ¿es esta configuración viable?

$$
\begin{aligned}
S &\rightarrow ab \mid B \\
B &\rightarrow b
\end{aligned}
$$

Stack	**Input**	Operaciones
$\$$	ab	
$\$a$	b	`shift`
$\$ab$	.	`shift`
$\$aB$	.	`reduce` $b \xrightarrow{\epsilon} B$
✕	✕	✕

¿podemos usar el bottom-up PDA para hacer parsing?

Problemas

1. Conflicto **Shift-Reduce**.
2. Conflicto **Reduce-Reduce**.
3. Configuraciones **no-viables**.

¿cómo determinamos si tenemos una **configuración/prefijo viable**?