Análisis Sintáctico Ascendente

CI4721 – Lenguajes de Programación II

Ernesto Hernández-Novich
<emhn@usb.ve>

Universidad "Simón Bolívar"

UNIVERSIDAD SIMÓN BOLÍVAR

Reconocedor ascendente no determinístico

Sea $G = (N, \Sigma, P, S)$ una CFG cualquiera. Entonces el PDA extendido

$$Asc(G) = (\{q_0, q_1\}, \Sigma, N \cup \Sigma, \delta, q_0, \{q_1\})$$

con δ definida según

$$\delta(q_0, \mathbf{a}, \lambda) = \{(q_0, \mathbf{a})\}, \forall \mathbf{a} \in \Sigma \quad (1)$$

$$\delta(q_0, \lambda, \alpha) = \{A \to \alpha^R \in P : (q_0, A)\}, \forall A \in N \wedge \alpha \neq S \quad (2)$$

$$\delta(q_0, \lambda, S) = \{(q_1, \lambda)\} \cup \{A \to S \in P : (q_0, A)\} \quad (3)$$

¿Cuál es la razón del no-determinismo?

UNIVERSIDAD SIMÓN BOLÍVAR

Contexto $LR(0)$

Sea $G = (N, \Sigma, P, S)$ una CFG cualquiera.
Si tenemos una forma sentencial más derecha

$$S \underset{Right}{\overset{*}{\Rightarrow}} \beta A v \underset{Right}{\Rightarrow} \beta \alpha v$$

decimos que $\beta\alpha$ es un **contexto** $LR(0)$ para $A \to \alpha \in P$.

- Cada *shift* deben avanzar de izquierda a derecha hasta completar $\beta\alpha$.
- Cada *reduce* corresponden a sustituir α por A.

UNIVERSIDAD SIMÓN BOLÍVAR

Un ejemplo trivial

Posibles derivaciones más derechas

Para la gramática

$$S \to \mathbf{a}A$$
$$S \to \mathbf{b}B$$
$$A \to \mathbf{ab}A$$
$$A \to \mathbf{b}B$$
$$B \to \mathbf{b}B\mathbf{c}$$
$$B \to \mathbf{bc}$$

$$\begin{aligned} S &\Rightarrow \mathbf{a}A \\ &\overset{i}{\Rightarrow} \mathbf{a}(\mathbf{ab})^i A \\ &\Rightarrow \mathbf{a}(\mathbf{ab})^i \mathbf{b}B \\ &\overset{j}{\Rightarrow} \mathbf{a}(\mathbf{ab})^i \mathbf{b}\mathbf{b}^j B\mathbf{c}^j \\ &\Rightarrow \mathbf{a}(\mathbf{ab})^i \mathbf{b}\mathbf{b}^j \mathbf{b}\mathbf{c}\mathbf{c}^j \end{aligned}$$

$$\begin{aligned} S &\Rightarrow \mathbf{b}B \\ &\overset{i}{\Rightarrow} \mathbf{b}\mathbf{b}^i B\mathbf{c}^i \\ &\Rightarrow \mathbf{b}\mathbf{b}^i \mathbf{b}\mathbf{c}\mathbf{c}^i \end{aligned}$$

UNIVERSIDAD SIMÓN BOLÍVAR

Un ejemplo trivial

Podemos construir los contextos

Regla	$LR(0) - CONTEXT$
$S \to \mathbf{a}A$	$\{\mathbf{a}A\}$
$S \to \mathbf{b}B$	$\{\mathbf{b}B\}$
$A \to \mathbf{ab}A$	$\{\mathbf{a}(\mathbf{ab})^i A \mid i > 0\}$
$A \to \mathbf{b}B$	$\{\mathbf{a}(\mathbf{ab})^i \mathbf{b}B \mid i \geq 0\}$
$B \to \mathbf{b}B\mathbf{c}$	$\{\mathbf{a}(\mathbf{ab})^i \mathbf{b}\mathbf{b}^j B\mathbf{c}, \mathbf{b}\mathbf{b}^j B\mathbf{c} \mid i \geq 0, j > 0\}$
$B \to \mathbf{bc}$	$\{\mathbf{a}(\mathbf{ab})^i \mathbf{b}\mathbf{b}^j \mathbf{c}, \mathbf{b}\mathbf{b}^j \mathbf{c} \mid i \geq 0, j > 0\}$

- Los contextos se deducen de las formas sentenciales más derechas.
- Cada contexto es un lenguaje posiblemente infinito.

UNIVERSIDAD SIMÓN BOLÍVAR

Gramáticas $LR(0)$

Definición

Diremos que una gramática $G = (N, \Sigma, P, S)$ con símbolo inicial no recursivo es **LR(0)** si

$$\forall \alpha \in (N \cup \Sigma)^* \wedge v \in \Sigma^*$$

cuando

$$\alpha \in LR(0) - CONTEXT(A \to \beta_1)$$
$$\alpha v \in LR(0) - CONTEXT(B \to \beta_2)$$

entonces $v = \lambda$, $A = B$ y $\beta_1 = \beta_2$.

UNIVERSIDAD SIMÓN BOLÍVAR

Gramáticas $LR(0)$

Definición

Diremos que una gramática $G = (N, \Sigma, P, S)$ con símbolo inicial no recursivo es **LR(0)** si

$$\forall \alpha \in (N \cup \Sigma)^* \wedge v \in \Sigma^*$$

cuando

$$\alpha \in LR(0) - CONTEXT(A \to \beta_1)$$
$$\alpha v \in LR(0) - CONTEXT(B \to \beta_2)$$

entonces $v = \lambda$, $A = B$ y $\beta_1 = \beta_2$.

No hay ningún contexto $LR(0)$ de una regla
que sea *prefijo* de un contexto $LR(0)$ de otra regla.

UNIVERSIDAD SIMÓN BOLÍVAR

Hacia el determinismo

Keep *shifting* until you can *reduce*

Los contextos $LR(0)$ proveen información para actuar ante un α particular:

1. Si $\alpha \in LR(0) - CONTEXT(A \to \beta)$, se puede reducir con $A \to \beta$.
2. Si α es un *prefijo viable* de algún $LR(0) - CONTEXT$, se avanza (*shift*) con algún símbolo (terminal o no-terminal) para extenderlo.
3. En caso contrario, la entrada se rechaza.

UNIVERSIDAD SIMÓN BOLÍVAR

¿Qué pasa si hay más de una alternativa?

Conflictos $LR(0)$

- Si para un α particular encontramos

$$\alpha \in LR(0) - CONTEXT(A \to \beta_1)$$
$$\alpha \in LR(0) - CONTEXT(B \to \beta_2)$$

con $A \to \beta_1 \neq B \to \beta_2$ tenemos un **conflicto *reduce/reduce LR(0)*** pues no podemos decidir cuál regla reducir.

UNIVERSIDAD SIMÓN BOLÍVAR

¿Qué pasa si hay más de una alternativa?

Conflictos $LR(0)$

- Si para un α particular encontramos

$$\alpha \in LR(0) - CONTEXT(A \to \beta_1)$$
$$\alpha \in LR(0) - CONTEXT(B \to \beta_2)$$

con $A \to \beta_1 \neq B \to \beta_2$ tenemos un **conflicto *reduce/reduce LR(0)*** pues no podemos decidir cuál regla reducir.

- Si para un α particular encontramos

$$\alpha \in LR(0) - CONTEXT(A \to \beta_1)$$
$$\alpha\gamma \in LR(0) - CONTEXT(B \to \beta_2)$$

con $\gamma \neq \lambda$ tenemos un conflicto ***shift/reduce LR(0)*** pues no podemos decidir si reducir o continuar avanzando en la entrada.

De prefijo hasta contexto

Items $LR(0)$ de la gramática

Sea $G = (N, \Sigma, P, S)$ y $A \to \alpha \in P$, diremos que

$$[A \to \alpha_1 \cdot \alpha_2]$$

es un **item LR(0)** para cualesquiera $\alpha_1, \alpha_2 \in (N \cup \Sigma)^*$ y $\alpha = \alpha_1\alpha_2$

- Un item de la forma $[A \to \alpha\cdot]$ se denomina *item completo*.
- Una producción $A \to \alpha$ genera $|\alpha| + 1$ items – uno por cada posible posición del marcador en el lado derecho.
- Una producción $A \to \lambda$ solamente genera el item $[A \to \cdot]$

Omitiremos los corchetes para ahorrar espacio.

UNIVERSIDAD SIMÓN BOLÍVAR

Otro ejemplo trivial

El conjunto de items $LR(0)$ de la gramática

Regla	Items $LR(0)$	$LR(0) - CONTEXT$
$S \to A$	$S \to \cdot A$ $S \to A\cdot$	$\{A\}$
$A \to \mathbf{a}A\mathbf{b}$	$A \to \cdot\mathbf{a}A\mathbf{b}$ $A \to \mathbf{a}\cdot A\mathbf{b}$ $A \to \mathbf{a}A\cdot\mathbf{b}$ $A \to \mathbf{a}A\mathbf{b}\cdot$	$\{\mathbf{a}^i A\mathbf{b} \mid i > 0\}$
$A \to \mathbf{a}$	$A \to \cdot\mathbf{a}$ $A \to \mathbf{a}\cdot$	$\{\mathbf{a}^i \mid i > 0\}$
$A \to \lambda$	$A \to \cdot$	$\{\mathbf{a}^i \mid i \geq 0\}$

¿Puede identificar los conflictos?

UNIVERSIDAD SIMÓN BOLÍVAR

Máquina característica $LR(0)$

Sea $G = (N, \Sigma, P, S)$. Construiremos el $\lambda - NFA$

$$M = (Q, N \cup \Sigma, \delta, q_0, Q)$$

Q contiene todos los items $LR(0)$ además del estado q_0, mientras

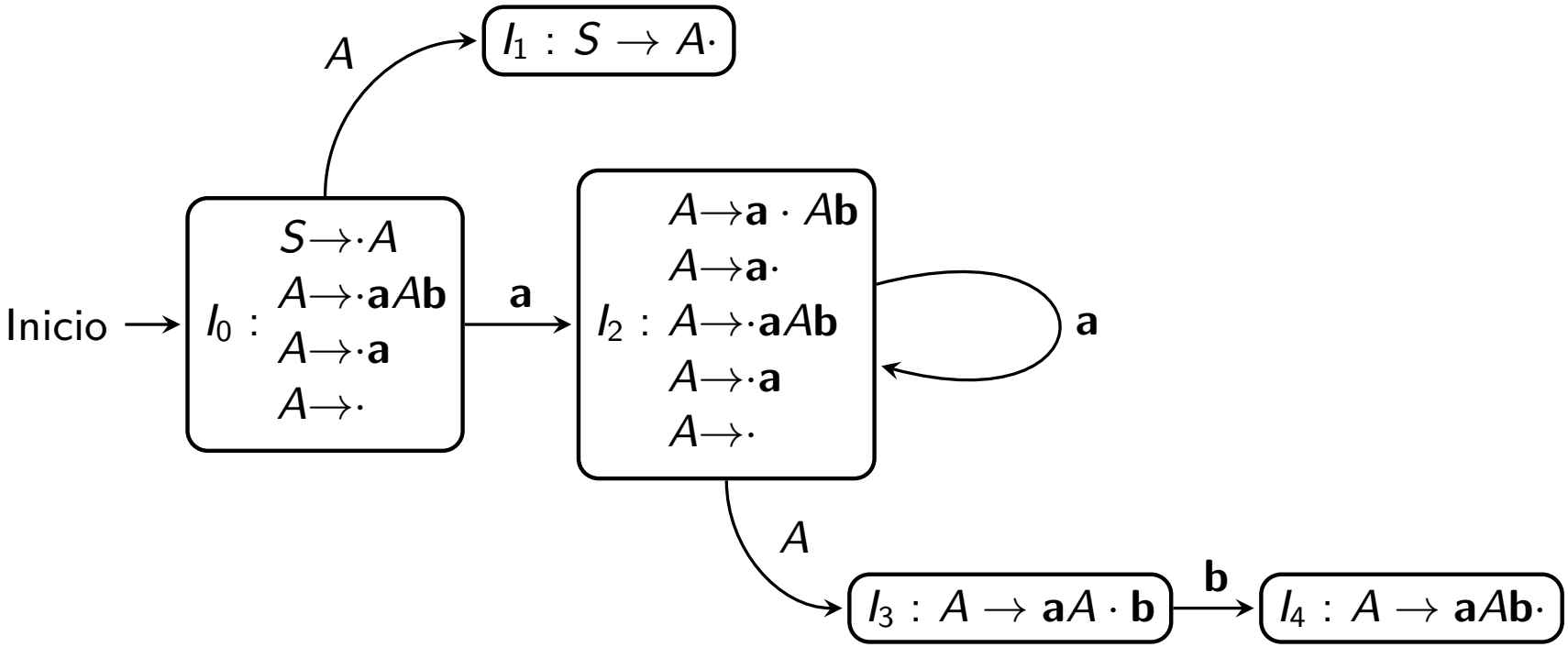
$$\begin{aligned}
\delta(q_0, \lambda) &= \{S \to \cdot\alpha \mid S \to \alpha \in P\} \\
\delta(A \to \alpha \cdot \mathbf{a}\beta, \mathbf{a}) &= \{A \to \alpha\mathbf{a} \cdot \beta\} \\
\delta(A \to \alpha \cdot B\beta, B) &= \{A \to \alpha B \cdot \beta\} \\
\delta(A \to \alpha \cdot B\beta, \lambda) &= \{B \to \cdot\gamma \mid B \to \gamma \in P\}
\end{aligned}$$

- M acepta los prefijos viables de la gramática – el cómputo de M "avanza" hasta encontrar el lado derecho de una producción.
- Se puede construir determinístico en una sola operación.

UNIVERSIDAD SIMÓN BOLÍVAR

Sigue el ejemplo trivial

La máquina característica $LR(0)$ de la gramática



- Conflicto *shift/reduce* en I_0 – *shift* **a** o *reduce* $A \to \lambda$.
- Conflictos *shift/reduce* en I_2 – *shift* **a**, *reduce* $A \to \lambda$ o *reduce* $A \to \mathbf{a}$.
- Conflicto *reduce/reduce* en I_2 – *reduce* $A \to \lambda$ o *reduce* $A \to \mathbf{a}$.

UNIVERSIDAD SIMÓN BOLÍVAR

¿Y esa máquina sirve?

Theorem

Sea G una CFG y M su máquina característica $LR(0)$, entonces

$$A \to \alpha \cdot \beta \in \hat{\delta}(q_0, w)$$

si y sólo si $w = p\alpha$ con $p\alpha\beta$ contexto $LR(0)$ de $A \to \alpha\beta$

UNIVERSIDAD SIMÓN BOLÍVAR

¿Y esa máquina sirve?

Theorem

Sea G una CFG y M su máquina característica $LR(0)$, entonces

$$A \to \alpha \cdot \beta \in \hat{\delta}(q_0, w)$$

si y sólo si $w = p\alpha$ con $p\alpha\beta$ contexto $LR(0)$ de $A \to \alpha\beta$

Theorem

Sea G una CFG con símbolo inicial no recursivo. Entonces G es $LR(0)$ si y sólo si $\hat{\delta}$ de su máquina característica determinística $LR(0)$ satisface:

1. *Si $\hat{\delta}(q_s, u)$ contiene **únicamente** un item $A \to \alpha\cdot$ con $\alpha \neq \lambda$.*
2. *Si $\hat{\delta}(q_s, u)$ contiene un item $A \to \cdot$ entonces el marcador siempre es seguido por un **no terminal** en el resto de los items.*

UNIVERSIDAD SIMÓN BOLÍVAR

¿Cómo aprovechamos la máquina?

Reconocedor basado en la máquina característica

input: $G = (N, \Sigma, P, S)$ CFG $LR(0)$, $w \in \Sigma^*$ y la Máquina $LR(0)$ de G

$u, v, error \leftarrow \lambda, w,$ **false**
repeat
  if $\hat{\delta}(q_0, u)$ contiene $A \to \alpha\cdot$ con $u = p\alpha$ **then**
    $u \leftarrow pA$
  else if $\hat{\delta}(q_0, u)$ contiene $A \to \alpha \cdot \beta \wedge v \neq \lambda$ **then**
    $shift(u, v)$
  else
    $error \leftarrow$ **true**
  end if
until $u = S \vee error$
if $u = S$ **then**
  aceptar
else
  rechazar
end if

UNIVERSIDAD SIMÓN BOLÍVAR

Un ejemplo completo

Gramática G_{AE} de expresiones aditivas

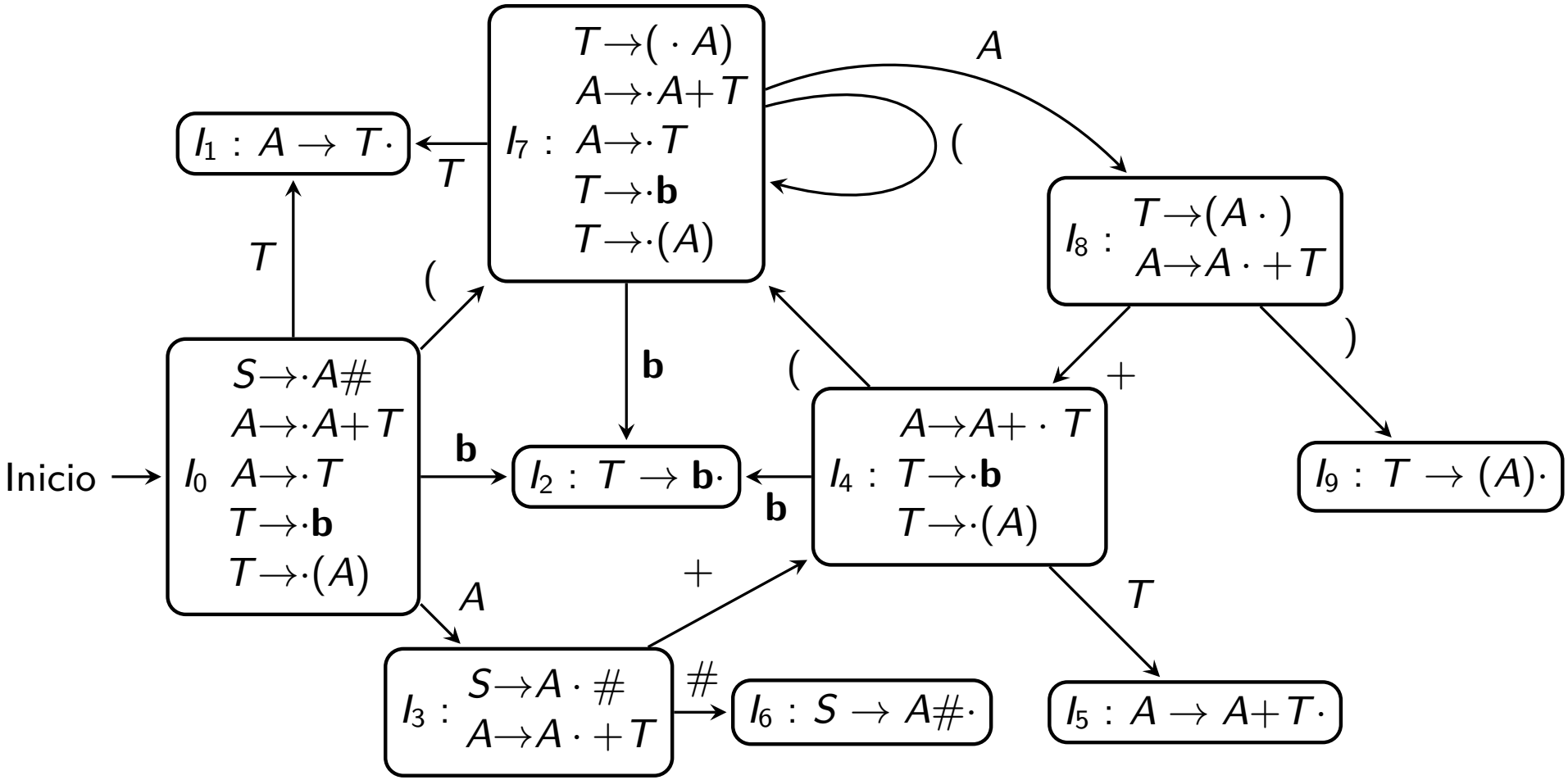
$$
\begin{aligned}
S &\to A\# \\
A &\to A{+}T \\
A &\to T \\
T &\to \mathbf{b} \\
T &\to (A)
\end{aligned}
$$

- $\#$ es el terminador – mismo propósito que para $LL(k)$
- **b** abstrae números e identificadores.

UNIVERSIDAD SIMÓN BOLÍVAR

Expresiones Aditivas

La máquina característica $LR(0)$ de G_{AE}



No hay conflictos – G_{AE} es $LR(0)$

Otro ejemplo completo

Gramática G_E de expresiones aditivas y multiplicativas

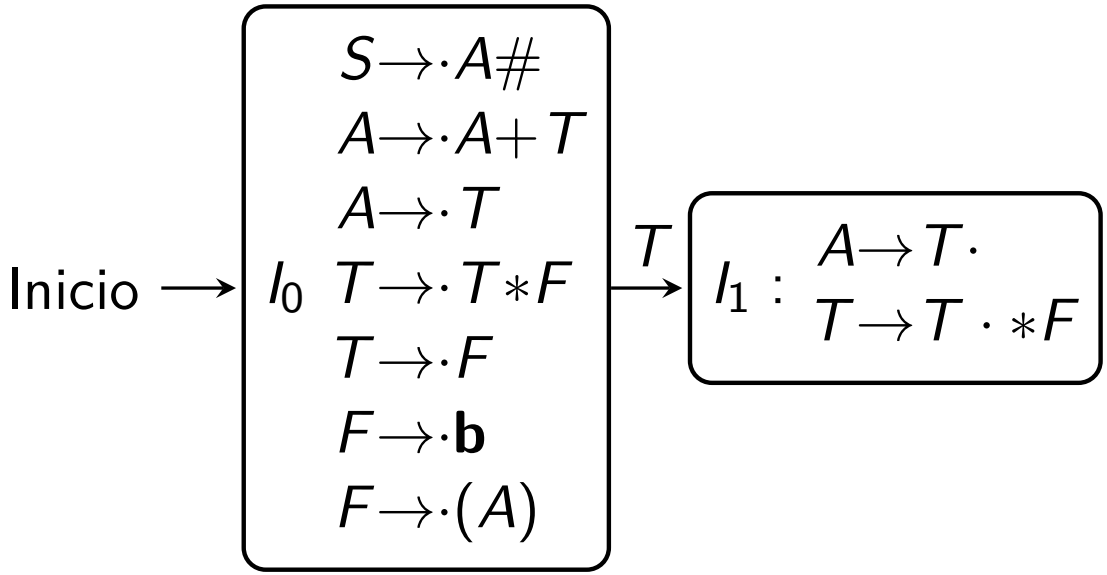
$$
\begin{aligned}
S &\to A\# \\
A &\to A{+}T \\
A &\to T \\
T &\to T{*}F \\
T &\to F \\
F &\to \mathbf{b} \\
F &\to (A)
\end{aligned}
$$

- F genera subexpresiones multiplicativas – precedencia por gramática.
- **b** abstrae números e identificadores.

UNIVERSIDAD SIMÓN BOLÍVAR

Expresiones Aditivas y Multiplicativas

Parte de la máquina característica $LR(0)$ de G_E



Conflicto *shift/reduce* en I_1 – G_E no es $LR(0)$

UNIVERSIDAD SIMÓN BOLÍVAR

Consideraciones

- Estudie la demostración del Teorema 20.4.1 del libro de texto.
- Estudie la demostración del Teorema 20.4.4 del libro de texto.
- Sin construir la máquina característica, ¿la gramática G_{AE} seguiría siendo $LR(0)$ si se agrega $A \rightarrow A - T$?
- Construya la máquina característica para G_E e identifique todos los conflictos presentes según su tipo.
- Use el algoritmo reconocedor basado en la máquina característica de G_{AE} para reconocer $b + (b + b)$ e identifique las formas sentenciales más derechas en el proceso.

UNIVERSIDAD SIMÓN BOLÍVAR

Bibliografía

- [*Sudkamp*]
  - Secciones 20.1 a 20.4
  - Ejercicios 20.1 a 20.5

UNIVERSIDAD SIMÓN BOLÍVAR