Esquemas de Traducción

CI4721 – Lenguajes de Programación II

Ernesto Hernández-Novich
<emhn@usb.ve>

Universidad "Simón Bolívar"

UNIVERSIDAD SIMÓN BOLÍVAR

Esquema de Traducción

Un **Esquema de Traducción Dirigido por Sintaxis**

- Comienza como una Definición Dirigida por Sintaxis.
- El meta-lenguaje de operaciones es un lenguaje de programación Turing-completo – después de todo, es para traducir.
- Las acciones pueden ocurrir en *cualquier* punto de las producciones.
- Están asociadas a un reconocedor concreto.

El resultado producido para el símbolo inicial corresponde con la traducción

UNIVERSIDAD SIMÓN BOLÍVAR

El proceso de traducción

Siempre se puede

1. Construir el árbol gramatical completo.
2. Recorrer el árbol en DFS de izquierda a derecha.
3. Ejecutar las acciones a medida que se encuentran en este recorrido.

El proceso de traducción

Siempre se puede

1. Construir el árbol gramatical completo.
2. Recorrer el árbol en DFS de izquierda a derecha.
3. Ejecutar las acciones a medida que se encuentran en este recorrido.

En la práctica

- No construir todo el árbol gramatical – ahorrar espacio.
- Garantizar la ejecución de acciones en el momento oportuno.

Tipo de reconocedor y tipo de atribución
definirán la estrategia.

UNIVERSIDAD SIMÓN BOLÍVAR

Esquema de Traducción Postfijo

El esquema ideal – simple y eficiente

- La Definición Dirigida por Sintaxis es S-Atribuida.
- El reconocedor es ascendente.

Basta ejecutar las acciones al reducir cada regla.

UNIVERSIDAD SIMÓN BOLÍVAR

Esquema de Traducción Postfijo

Evaluar expresiones e imprimir el resultado

$$
\begin{array}{ll}
S \to E\# & \{\textbf{print}(E.val)\} \\
E \to E_1 + T & \{E.val \leftarrow E_1.val + T.val\} \\
E \to T & \{E.val \leftarrow T.val\} \\
T \to T_1 * F & \{T.val \leftarrow T_1.val * F.val\} \\
T \to F & \{T.val \leftarrow F.val\} \\
F \to (E) & \{F.val \leftarrow E.val\} \\
F \to \mathbf{n} & \{F.val \leftarrow \mathbf{n}.val\}
\end{array}
$$

Estas reglas permiten *evaluar* el resultado justo después de cada reducción.

UNIVERSIDAD SIMÓN BOLÍVAR

Acciones coincidiendo con reducciones

- Atributos de cada símbolo en la pila.
  - La propia pila LR o una pila "paralela".
  - Si el atributo es simple, contenerlo directamente – varios atributos simples en estructura o registro variante.
  - Si son muchos atributos diferentes o bien complejos, contener un apuntador al atributo concreto.
- Al momento de reducir $A \rightarrow XYZ$
  - Los atributos de X, Y y Z están en posiciones conocidas en la pila.
  - Al reducir, se retiran de la pila durante el cálculo de los atributos de A.
  - Los atributos calculados para A quedarán en el nuevo tope de la pila.

UNIVERSIDAD SIMÓN BOLÍVAR

Una corrida

Entrada	Pila LR	Pila Atributos	Acción

UNIVERSIDAD SIMÓN BOLÍVAR

Una corrida

Entrada	Pila LR	Pila Atributos	Acción
6*5+12#	S #	_ #	

Una corrida

Entrada	Pila LR	Pila Atributos	Acción
6*5+12#	S #	_ #	shift **n** – Empilar atributo intrínseco.
***5+12 #**	**n** S #	6 _ #	

Una corrida

Entrada	Pila LR	Pila Atributos	Acción
6*5+12#	S #	_ #	shift **n** – Empilar atributo intrínseco.
***5+12 #**	**n** S #	6 _ #	reduce $F \rightarrow \mathbf{n}$ $\{F.val \leftarrow \mathbf{n}.val\}$
***5+12 #**	F S #	6 _ #	

Una corrida

Entrada	Pila LR	Pila Atributos	Acción
6*5+12#	S #	_ #	shift **n** – Empilar atributo intrínseco.
***5+12 #**	**n** S #	6 _ #	reduce $F \rightarrow \mathbf{n}\ \{F.val \leftarrow \mathbf{n}.val\}$
***5+12 #**	F S #	6 _ #	reduce $T \rightarrow F\ \{T.val \leftarrow F.val\}$
***5+12 #**	T S #	6 _ #	

UNIVERSIDAD SIMÓN BOLÍVAR

Una corrida

Entrada	Pila LR	Pila Atributos	Acción
6*5+12#	S #	_ #	shift **n** – Empilar atributo intrínseco.
***5+12 #**	**n** S #	6 _ #	reduce $F \rightarrow \mathbf{n}\ \{F.val \leftarrow \mathbf{n}.val\}$
***5+12 #**	F S #	6 _ #	reduce $T \rightarrow F\ \{T.val \leftarrow F.val\}$
***5+12 #**	T S #	6 _ #	shift *
5+12 #	* T S #	_ 6 _ #	

Una corrida

Entrada	Pila LR	Pila Atributos	Acción
6*5+12#	S #	_ #	shift **n** – Empilar atributo intrínseco.
***5+12 #**	**n** S #	6 _ #	reduce $F \rightarrow \mathbf{n}\ \{F.val \leftarrow \mathbf{n}.val\}$
***5+12 #**	F S #	6 _ #	reduce $T \rightarrow F\ \{T.val \leftarrow F.val\}$
***5+12 #**	T S #	6 _ #	shift *
5+12 #	* T S #	_ 6 _ #	shift **n** – Empilar atributo intrínseco.
+12 #	**n** * T S #	5 _ 6 _ #	

Una corrida

Entrada	Pila LR	Pila Atributos	Acción
6*5+12#	S #	_ #	shift **n** – Empilar atributo intrínseco.
***5+12 #**	**n** S #	6 _ #	reduce $F \rightarrow \mathbf{n}\ \{F.val \leftarrow \mathbf{n}.val\}$
***5+12 #**	F S #	6 _ #	reduce $T \rightarrow F\ \{T.val \leftarrow F.val\}$
***5+12 #**	T S #	6 _ #	shift *
5+12 #	* T S #	_ 6 _ #	shift **n** – Empilar atributo intrínseco.
+12 #	**n** * T S #	5 _ 6 _ #	reduce $F \rightarrow \mathbf{n}\ \{F.val \leftarrow \mathbf{n}.val\}$
+12 #	F * T_1 S #	5 _ 6 _ #	

UNIVERSIDAD SIMÓN BOLÍVAR

Una corrida

Entrada	Pila LR	Pila Atributos	Acción
6*5+12#	S #	_ #	shift **n** – Empilar atributo intrínseco.
***5+12 #**	**n** S #	6 _ #	reduce $F \rightarrow \mathbf{n}\ \{F.val \leftarrow \mathbf{n}.val\}$
***5+12 #**	F S #	6 _ #	reduce $T \rightarrow F\ \{T.val \leftarrow F.val\}$
***5+12 #**	T S #	6 _ #	shift *
5+12 #	* T S #	_ 6 _ #	shift **n** – Empilar atributo intrínseco.
+12 #	**n** * T S #	5 _ 6 _ #	reduce $F \rightarrow \mathbf{n}\ \{F.val \leftarrow \mathbf{n}.val\}$
+12 #	F * T_1 S #	5 _ 6 _ #	reduce $T \rightarrow T{*}F\ \{T.val \leftarrow T_1.val * F.val\}$
+12 #	T S #	30 _ #	

UNIVERSIDAD SIMÓN BOLÍVAR

Una corrida

Entrada	Pila LR	Pila Atributos	Acción
6*5+12#	S #	_ #	shift **n** – Empilar atributo intrínseco.
***5+12 #**	**n** S #	6 _ #	reduce $F \rightarrow \mathbf{n}\ \{F.val \leftarrow \mathbf{n}.val\}$
***5+12 #**	F S #	6 _ #	reduce $T \rightarrow F\ \{T.val \leftarrow F.val\}$
***5+12 #**	T S #	6 _ #	shift *
5+12 #	* T S #	_ 6 _ #	shift **n** – Empilar atributo intrínseco.
+12 #	**n** * T S #	5 _ 6 _ #	reduce $F \rightarrow \mathbf{n}\ \{F.val \leftarrow \mathbf{n}.val\}$
+12 #	F * T_1 S #	5 _ 6 _ #	reduce $T \rightarrow T{*}F\ \{T.val \leftarrow T_1.val * F.val\}$
+12 #	T S #	30 _ #	reduce $E \rightarrow T\ \{E.val \leftarrow T.val\}$
+12 #	E S #	30 _ #	

UNIVERSIDAD SIMÓN BOLÍVAR

Una corrida

Entrada	Pila LR	Pila Atributos	Acción
6*5+12#	S #	_ #	shift **n** – Empilar atributo intrínseco.
***5+12 #**	**n** S #	6 _ #	reduce $F \to \mathbf{n}\ \{F.val \leftarrow \mathbf{n}.val\}$
***5+12 #**	F S #	6 _ #	reduce $T \to F\ \{T.val \leftarrow F.val\}$
***5+12 #**	T S #	6 _ #	shift *
5+12 #	* T S #	_ 6 _ #	shift **n** – Empilar atributo intrínseco.
+12 #	**n** * T S #	5 _ 6 _ #	reduce $F \to \mathbf{n}\ \{F.val \leftarrow \mathbf{n}.val\}$
+12 #	F * T_1 S #	5 _ 6 _ #	reduce $T \to T*F\ \{T.val \leftarrow T_1.val * F.val\}$
+12 #	T S #	30 _ #	reduce $E \to T\ \{E.val \leftarrow T.val\}$
+12 #	E S #	30 _ #	shift +
12 #	+ E S #	_ 30 _ #	

UNIVERSIDAD SIMÓN BOLÍVAR

Una corrida

Entrada	Pila LR	Pila Atributos	Acción
6*5+12#	S #	_ #	shift **n** – Empilar atributo intrínseco.
***5+12 #**	**n** S #	6 _ #	reduce $F \rightarrow \mathbf{n}\ \{F.val \leftarrow \mathbf{n}.val\}$
***5+12 #**	F S #	6 _ #	reduce $T \rightarrow F\ \{T.val \leftarrow F.val\}$
***5+12 #**	T S #	6 _ #	shift *
5+12 #	* T S #	_ 6 _ #	shift **n** – Empilar atributo intrínseco.
+12 #	**n** * T S #	5 _ 6 _ #	reduce $F \rightarrow \mathbf{n}\ \{F.val \leftarrow \mathbf{n}.val\}$
+12 #	F * T_1 S #	5 _ 6 _ #	reduce $T \rightarrow T{*}F\ \{T.val \leftarrow T_1.val * F.val\}$
+12 #	T S #	30 _ #	reduce $E \rightarrow T\ \{E.val \leftarrow T.val\}$
+12 #	E S #	30 _ #	shift +
12 #	+ E S #	_ 30 _ #	shift **n** – Empilar atributo intrínseco.
#	**n** + E S #	12 _ 30 _ #	

UNIVERSIDAD SIMÓN BOLÍVAR

Una corrida

Entrada	Pila LR	Pila Atributos	Acción
6*5+12#	S #	_ #	shift **n** – Empilar atributo intrínseco.
***5+12 #**	**n** S #	6 _ #	reduce $F \to \mathbf{n}\ \{F.val \leftarrow \mathbf{n}.val\}$
***5+12 #**	F S #	6 _ #	reduce $T \to F\ \{T.val \leftarrow F.val\}$
***5+12 #**	T S #	6 _ #	shift *
5+12 #	* T S #	_ 6 _ #	shift **n** – Empilar atributo intrínseco.
+12 #	**n** * T S #	5 _ 6 _ #	reduce $F \to \mathbf{n}\ \{F.val \leftarrow \mathbf{n}.val\}$
+12 #	F * T_1 S #	5 _ 6 _ #	reduce $T \to T{*}F\ \{T.val \leftarrow T_1.val * F.val\}$
+12 #	T S #	30 _ #	reduce $E \to T\ \{E.val \leftarrow T.val\}$
+12 #	E S #	30 _ #	shift +
12 #	+ E S #	_ 30 _ #	shift **n** – Empilar atributo intrínseco.
#	**n** + E S #	12 _ 30 _ #	reduce $F \to n\ \{F.val \leftarrow \mathbf{n}.val\}$
#	F + E S #	12 _ 30 _ #	

UNIVERSIDAD SIMÓN BOLÍVAR

Una corrida

Entrada	Pila LR	Pila Atributos	Acción
6*5+12#	S #	_ #	shift **n** – Empilar atributo intrínseco.
***5+12 #**	**n** S #	6 _ #	reduce $F \to \mathbf{n}\ \{F.val \leftarrow \mathbf{n}.val\}$
***5+12 #**	F S #	6 _ #	reduce $T \to F\ \{T.val \leftarrow F.val\}$
***5+12 #**	T S #	6 _ #	shift *
5+12 #	* T S #	_ 6 _ #	shift **n** – Empilar atributo intrínseco.
+12 #	**n** * T S #	5 _ 6 _ #	reduce $F \to \mathbf{n}\ \{F.val \leftarrow \mathbf{n}.val\}$
+12 #	F * T_1 S #	5 _ 6 _ #	reduce $T \to T*F\ \{T.val \leftarrow T_1.val * F.val\}$
+12 #	T S #	30 _ #	reduce $E \to T\ \{E.val \leftarrow T.val\}$
+12 #	E S #	30 _ #	shift +
12 #	+ E S #	_ 30 _ #	shift **n** – Empilar atributo intrínseco.
#	**n** + E S #	12 _ 30 _ #	reduce $F \to n\ \{F.val \leftarrow \mathbf{n}.val\}$
#	F + E S #	12 _ 30 _ #	reduce $T \to F\ \{T.val \leftarrow F.val\}$
#	T + E_1 S #	12 _ 30 _ #	

UNIVERSIDAD SIMÓN BOLÍVAR

Una corrida

Entrada	Pila LR	Pila Atributos	Acción
6*5+12#	S #	_ #	shift **n** – Empilar atributo intrínseco.
***5+12 #**	**n** S #	6 _ #	reduce $F \to \mathbf{n}\ \{F.val \leftarrow \mathbf{n}.val\}$
***5+12 #**	F S #	6 _ #	reduce $T \to F\ \{T.val \leftarrow F.val\}$
***5+12 #**	T S #	6 _ #	shift *
5+12 #	* T S #	_ 6 _ #	shift **n** – Empilar atributo intrínseco.
+12 #	**n** * T S #	5 _ 6 _ #	reduce $F \to \mathbf{n}\ \{F.val \leftarrow \mathbf{n}.val\}$
+12 #	F * T_1 S #	5 _ 6 _ #	reduce $T \to T{*}F\ \{T.val \leftarrow T_1.val * F.val\}$
+12 #	T S #	30 _ #	reduce $E \to T\ \{E.val \leftarrow T.val\}$
+12 #	E S #	30 _ #	shift +
12 #	+ E S #	_ 30 _ #	shift **n** – Empilar atributo intrínseco.
#	**n** + E S #	12 _ 30 _ #	reduce $F \to n\ \{F.val \leftarrow \mathbf{n}.val\}$
#	F + E S #	12 _ 30 _ #	reduce $T \to F\ \{T.val \leftarrow F.val\}$
#	T + E_1 S #	12 _ 30 _ #	reduce $E \to E{+}T\ \{E.val \leftarrow E_1.val + T.val\}$
#	E S #	42 _ #	

UNIVERSIDAD SIMÓN BOLÍVAR

Una corrida

Entrada	Pila LR	Pila Atributos	Acción
6*5+12#	S #	_ #	shift **n** – Empilar atributo intrínseco.
***5+12 #**	**n** S #	6 _ #	reduce $F \to \mathbf{n}$ $\{F.val \leftarrow \mathbf{n}.val\}$
***5+12 #**	F S #	6 _ #	reduce $T \to F$ $\{T.val \leftarrow F.val\}$
***5+12 #**	T S #	6 _ #	shift *
5+12 #	* T S #	_ 6 _ #	shift **n** – Empilar atributo intrínseco.
+12 #	**n** * T S #	5 _ 6 _ #	reduce $F \to \mathbf{n}$ $\{F.val \leftarrow \mathbf{n}.val\}$
+12 #	F * T_1 S #	5 _ 6 _ #	reduce $T \to T*F$ $\{T.val \leftarrow T_1.val * F.val\}$
+12 #	T S #	30 _ #	reduce $E \to T$ $\{E.val \leftarrow T.val\}$
+12 #	E S #	30 _ #	shift +
12 #	+ E S #	_ 30 _ #	shift **n** – Empilar atributo intrínseco.
#	**n** + E S #	12 _ 30 _ #	reduce $F \to n$ $\{F.val \leftarrow \mathbf{n}.val\}$
#	F + E S #	12 _ 30 _ #	reduce $T \to F$ $\{T.val \leftarrow F.val\}$
#	T + E_1 S #	12 _ 30 _ #	reduce $E \to E+T$ $\{E.val \leftarrow E_1.val + T.val\}$
#	E S #	42 _ #	shift #
#	# E S #	_ 42 _ #	accept $\{\mathbf{print}(E.val)\}$

Imprime la respuesta.

UNIVERSIDAD SIMÓN BOLÍVAR

Manipulación explícita de la pila

$$
\begin{array}{ll}
S \to E\# & \left\{ \begin{array}{l} \textbf{print}(stack[top-1].val); \\ top = top - 1; \end{array} \right\} \\
E \to E_1 + T & \left\{ \begin{array}{l} stack[top-2].val = stack[top-2].val + stack[top].val; \\ top = top - 2; \end{array} \right\} \\
E \to T & \\
T \to T_1 * F & \left\{ \begin{array}{l} stack[top-2].val = stack[top-2].val * stack[top].val; \\ top = top - 2; \end{array} \right\} \\
T \to F & \\
F \to (E) & \left\{ \begin{array}{l} stack[top-2].val = stack[top-1].val; \\ top = top - 2; \end{array} \right\} \\
F \to \textbf{n} &
\end{array}
$$

La manipulación es generada automáticamente por la herramienta reconocedora.

UNIVERSIDAD SIMÓN BOLÍVAR

Traducción a postfijo

A veces ni siquiera hay que manipular la pila

$$
\begin{array}{ll}
S \to E\# & \\
E \to E_1 + T & \{\textbf{print}('+');\} \\
E \to T & \\
T \to T_1 * F & \{\textbf{print}('*');\} \\
T \to F & \\
F \to (E) & \\
F \to \textbf{n} & \{\textbf{print}(\textbf{n}.val);\}
\end{array}
$$

UNIVERSIDAD SIMÓN BOLÍVAR

¿Y si el reconocedor es descendente?

Eliminar recursión izquierda ...

- Para eliminar la recursión izquierda de las reglas

$$A \to A\alpha$$
$$A \to \beta$$

aplicamos el método de reescritura para obtener

$$A \to \beta R$$
$$R \to \alpha R$$
$$R \to \lambda$$

UNIVERSIDAD SIMÓN BOLÍVAR

¿Y si el reconocedor es descendente?

Eliminar recursión izquierda ...

- Para eliminar la recursión izquierda de las reglas

$$A \to A\alpha$$
$$A \to \beta$$

aplicamos el método de reescritura para obtener

$$A \to \beta R$$
$$R \to \alpha R$$
$$R \to \lambda$$

- Se preserva el orden de los *terminales* generados por la regla.

¡Consideremos las acciones como terminales!

UNIVERSIDAD SIMÓN BOLÍVAR

En el traductor a postfijo

Aplicando esa técnica al traductor a postfijo, observamos

$$E \to E_1 + T \quad \{\textbf{print}('+');\}$$
$$E \to T$$

En el traductor a postfijo

Aplicando esa técnica al traductor a postfijo, observamos

$$E \to E_1 + T \quad \{\textbf{print}('+');\}$$
$$E \to T$$

Si consideramos la acción como un terminal y aplicamos el algoritmo general de eliminación de recursión izquierda, nos queda

$$E \to TR$$
$$R \to +T \ \{\textbf{print}('+');\} \ R$$
$$R \to \lambda$$

Ahora tenemos una acción "en medio"

Acciones dentro de las producciones

$$A \to X\{\text{acción}\}Y$$

- La acción debe efectuarse tan pronto se haya reconocido X
  - Reconocedor ascendente – efectuarla tan pronto aparezca el estado asociado con X en el tope de la pila.
  - Reconocedor descendente
    - Si Y es no terminal, efectuarla antes de expandirlo.
    - Si Y es terminal, efectuarla antes de consumirlo.
- Generalmente son viables cuando son L-Atribuidas.

UNIVERSIDAD SIMÓN BOLÍVAR

Expresiones, una vez más

$$
\begin{array}{lcll}
S & \rightarrow & E\# & \{\textbf{print}(E.val)\} \\
E & \rightarrow & T & \{U.i \leftarrow T.val\} \\
 & & U & \{E.val \leftarrow U.s\} \\
U & \rightarrow & +T & \{U_1.i \leftarrow U.i + T.val\} \\
 & & U_1 & \{U.s \leftarrow U_1.s\} \\
U & \rightarrow & \lambda & \{U.s \leftarrow U.i\} \\
T & \rightarrow & F & \{V.i \leftarrow F.val\} \\
 & & V & \{T.val \leftarrow V.s\} \\
V & \rightarrow & *F & \{V_1.i \leftarrow V_i.i * F.val\} \\
 & & V_1 & \{V.s \leftarrow V_1.s\} \\
V & \rightarrow & \lambda & \{V.s \leftarrow V.i\} \\
F & \rightarrow & (E) & \{F.val \leftarrow E.val\} \\
F & \rightarrow & \mathbf{n} & \{F.val \leftarrow \mathbf{n}.val\}
\end{array}
$$

Ahora es L-Atribuida

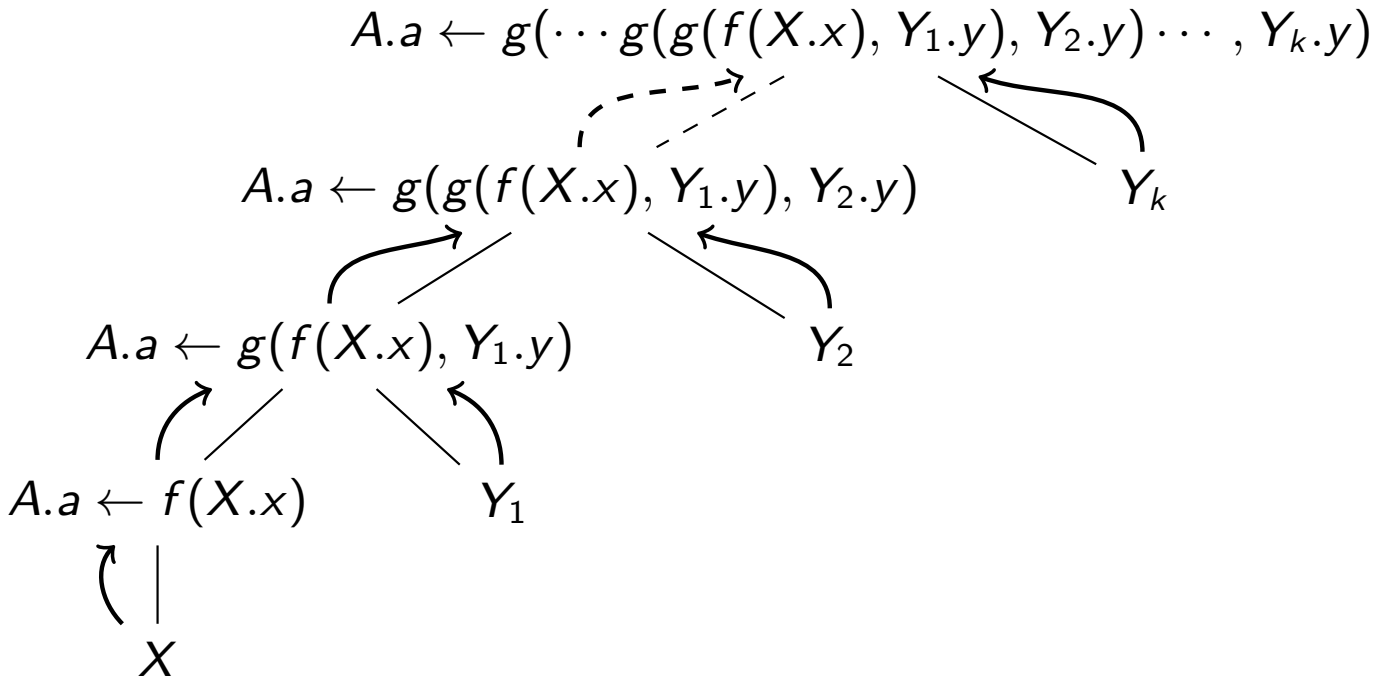
UNIVERSIDAD SIMÓN BOLÍVAR

¿Por qué funciona el método general?

Las producciones con recursión izquierda

$$\begin{array}{ll} A \to A_1\, Y & \{A.a \leftarrow g(A_1.a, Y.y)\} \\ A \to X & \{A.a \leftarrow f(X.x)\} \end{array}$$

sintetizan los atributos según el árbol



UNIVERSIDAD SIMÓN BOLÍVAR

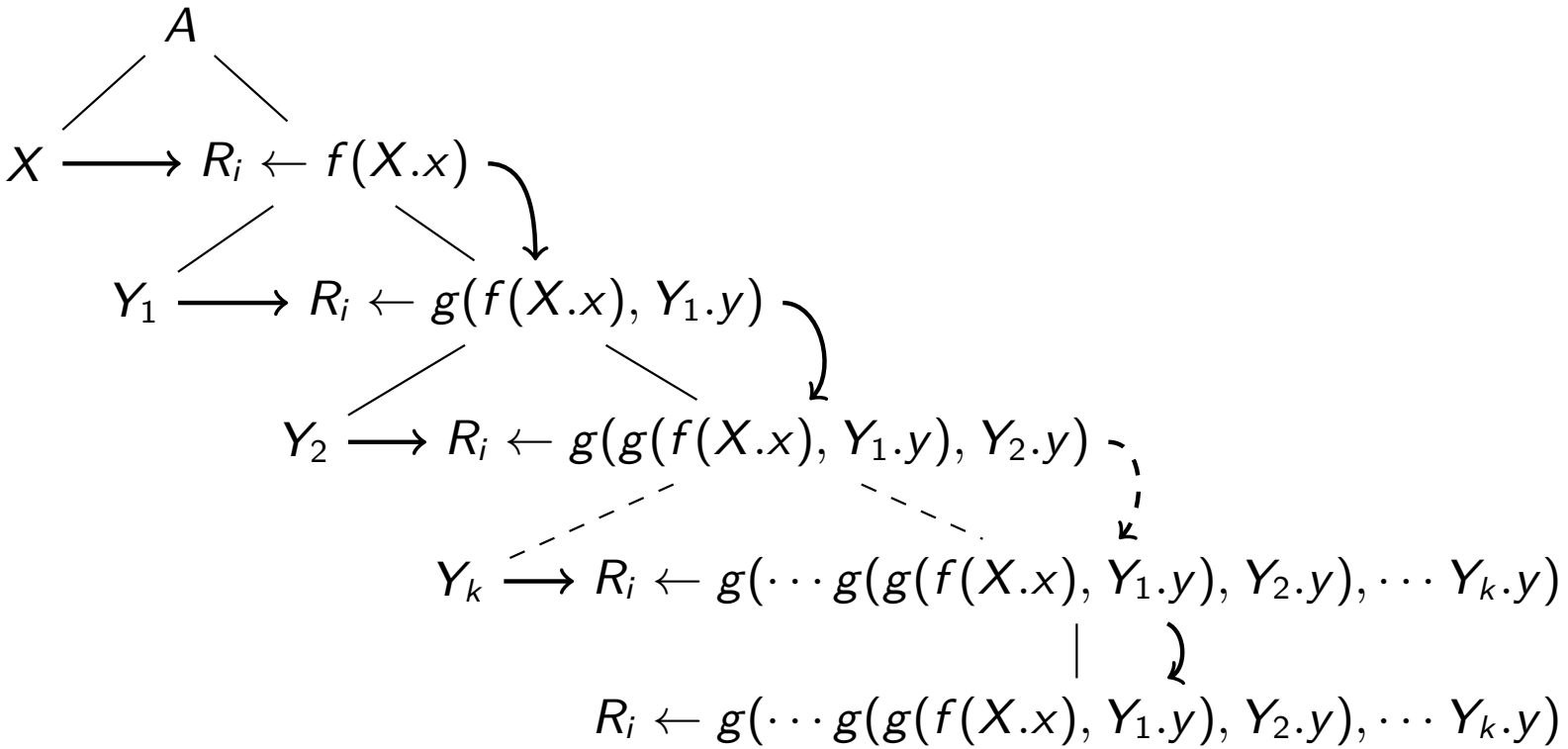
Después de la reescritura

$$
\begin{array}{lcll}
A & \rightarrow & X & \{R.i \leftarrow f(X.x)\} \\
 & & R & \{A.a \leftarrow R.s\} \\
R & \rightarrow & Y & \{R_1.i \leftarrow g(R.i, Y.y)\} \\
 & \rightarrow & R_1 & \{R.s \leftarrow R_1.s\} \\
R & \rightarrow & \lambda & \{R.s \leftarrow R.i\}
\end{array}
$$

- La generación gramatical ahora es recursiva por derecha.
- La acumulación debe operar de izquierda a derecha y de arriba abajo.
- El resultado final debe subir para llegar hasta el símbolo inicial.

UNIVERSIDAD SIMÓN BOLÍVAR

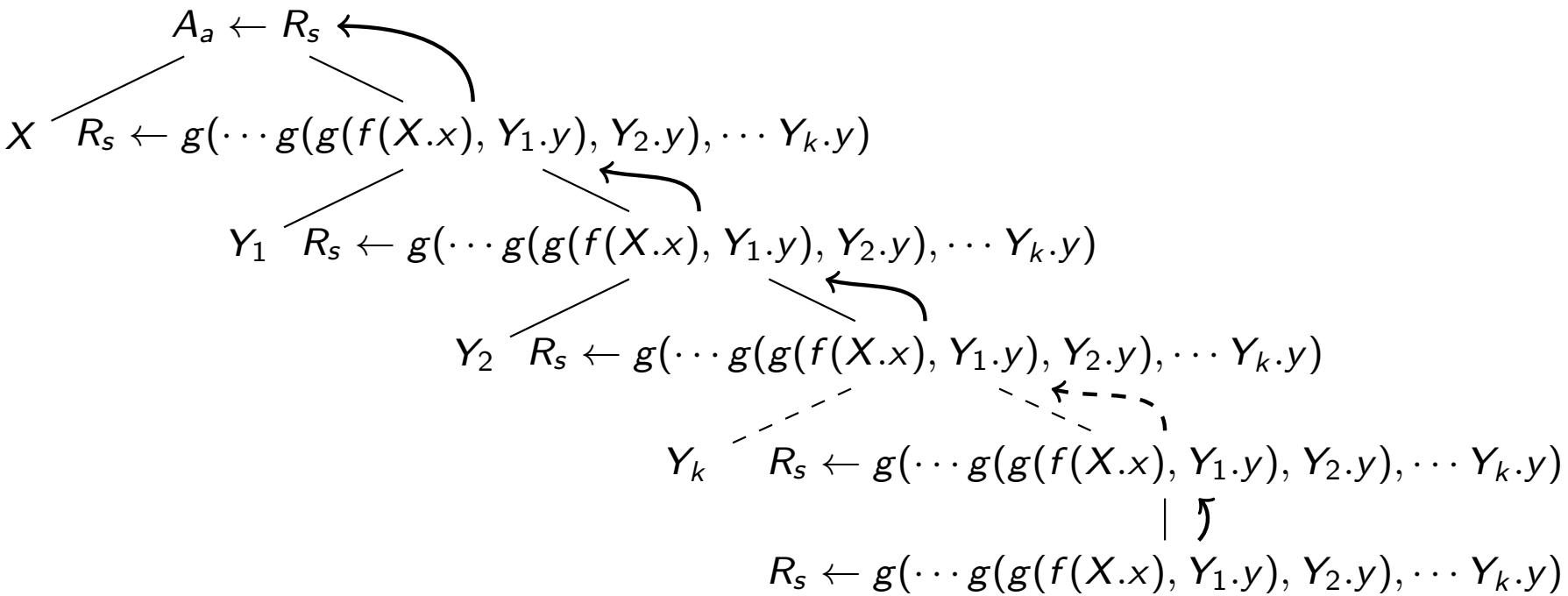
Después de la reescritura



- La acumulación de izquierda a derecha y de arriba abajo.
- El resultado final es heredado hasta el fondo del árbol.

UNIVERSIDAD SIMÓN BOLÍVAR

Después de la reescritura



El resultado final regresa a la raíz del árbol.

UNIVERSIDAD SIMÓN BOLÍVAR

¿Qué hacemos con las acciones "en medio"?

En el reconocedor descendente

Podemos mover nuevamente las acciones "en medio" hacia el final de las reglas para regresar a una definición S-Atribuida usando **no terminales marcadores**. Si se tiene

$$
\begin{aligned}
E &\rightarrow TR \\
R &\rightarrow +T\ \{\textbf{print}('+');\}\ R \\
R &\rightarrow \lambda
\end{aligned}
$$

se convierte en

$$
\begin{aligned}
E &\rightarrow TR \\
R &\rightarrow +T\ M\ R \\
R &\rightarrow \lambda \\
M &\rightarrow \lambda\ \{\textbf{print}('+');\}
\end{aligned}
$$

UNIVERSIDAD SIMÓN BOLÍVAR

Bibliografía

- [*Aho*]
  - Sección 5.4
  - Estudien los Ejemplos 5.18 y 5.19
  - Ejercicios 5.4.1 a 5.4.7
- [*Scott*]
  - Secciones 4.1 a 4.3
  - Ejercicios 4.1 a 4.5

UNIVERSIDAD SIMÓN BOLÍVAR