TEMA 3. ANÁLISIS ASCENDENTE

Gramáticas de Precedencia

PROCESADORES DE LENGUAJES
4º Informática

http://ccia.ei.uvigo.es/docencia/PL

9 de febrero de 2011

3.1 Analizadores sintácticos de Desplazamiento-Reducción

Definición (Algoritmo salto-reducción)

Sea $\mathcal{G} = (N, \Sigma, P, S)$ *una GIC con* $P = \{P_1, P_2, \ldots, P_n\}$.
Un algoritmo de salto-reducción $\mathfrak{a}(f, g)$ *para* $\mathcal{G}$*, está definido por un par de funciones :*

- f*: función de salto-reducción*

$$f : \Gamma^* \times (\Sigma \cup \{\$\})^* \rightarrow \{\text{SALTO, REDUCCION, ERROR, ACEPTAR}\}$$

- g*: función de reducción*

$$g : \Gamma^* \times (\Sigma \cup \{\$\})^* \rightarrow \{1, 2, \cdots, n\}$$

con: $\$$*: símbolo de fin de cadena y fin de pila*
Γ*: alfabeto de la pila* ($\Gamma = N \cup \Sigma \cup \{\$\}$) $\diamond$

Funcionamiento (análisis ascendente)

- Recorrido de la entrada de IZQ. a DER.
- Uso de una pila de símbolos
- *función* f*:* decide la acción a realizar a partir del contenido de la pila y del texto que queda por analizar
  - Si *acción*=SALTO: se añade a la pila el símbolo actual de entrada y se avanza una posición en la entrada
  - Si *acción*=REDUCIR: la función g determina la regla a reducir (regla P_i)
    - Se elimina de pila los símbolos del lado derecho de la regla P_i
    - Se añade a pila el símbolos del lado izquierdo de la regla P_i

Definición (Configuración)

Una configuración *de un analizador salto-reducción es un triple:*

$$(\$X_1X_2\cdots X_m,\ a_1a_2\cdots a_n\$,\ P_1P_2\cdots P_r)$$

- $\$X_1X_2\cdots X_m$*: representa la pila con* $\begin{cases} X_m \text{ en la } \underline{\text{cima}} \\ X_i \in N \cup \Sigma \end{cases}$
- $a_1a_2\cdots a_n\$$*: es lo que queda por analizar de la entrada donde:* $a_i \in \Sigma$*,* a_1*: símbolo actual y* $\$$*: fin de entrada*
- $P_1P_2\cdots P_r$*: cadena de reglas usadas para reducir el texto original* w *a* $X_1X_2\cdots X_ma_1a_2\cdots a_n$*.*

Configuración incial: $(\$, w\$, \varepsilon)$ $\diamond$

Definición (Acción)

Una acción *para un analizador salto-reducción* $\mathfrak{A}(f, g)$ *está determinado por las funciones* f *y* g *que relacionan pares de configuraciones (*$\overset{r}{\vdash}$*: reducción,* $\overset{s}{\vdash}$*: salto).*

- $f(\alpha, aw) = \text{SALTO} \Rightarrow (\alpha, aw, \Pi) \overset{s}{\vdash} (\alpha a, w, \Pi)$
- $f(\alpha\beta, w) = \text{REDUCCION}$
  $\left.\begin{array}{l} g(\alpha\beta, w) = i \\ P_i \equiv A \to \beta \end{array}\right\} \Rightarrow (\alpha\beta, w, \Pi) \overset{r_i}{\vdash} (\alpha A, w, \Pi i)$
- $f(\alpha, w) = \text{ACEPTAR} \Rightarrow (\alpha, w, \Pi) \vdash \text{ACEPTAR}$
- $f(\alpha, w) = \text{ERROR}$*, en otro caso*

Π *representará el conjunto de reglas empleadas en un análisis sintáctico por la derecha de la cadena* w $\diamond$

Nota: Normalmente f y g no dependerán de la totalidad de la pila, sino únicamente de algunos símbolos de su cima.

Se puede resumir la notación: $f(\gamma\alpha, xw') \equiv f(\alpha, x) \quad g(\gamma\alpha, xw') \equiv g(\alpha, x)$

En la práctica no será necesario definir las funciones f y g explícitamente. Se consultará directamente la tabla de relaciones de precedencia.

Nota: Utilizaremos $\vdash$ para referirnos tanto a $\overset{s}{\vdash}$ como para $\overset{r}{\vdash}$, cuando no sea necesario especificar el tipo de acción.

$\overset{+}{\vdash}$ y $\overset{*}{\vdash}$ tienen el significado habitual de cierre transitivo y cierre reflexivo y transitivo, respectivamente.

Definición Decimos que $\mathfrak{A}(w) = \Pi$, $w \in \Sigma^*$, si $\exists \Pi / (\$, w\$, \epsilon) \overset{*}{\vdash} (\$S, \$, \Pi)$, con $\mathfrak{A}(w) =$ ERROR en otro caso,

Definición (Algoritmo válido)

Decimos que un algoritmo de salto-reducción $\mathfrak{A}$ es válido para una GIC $\mathcal{G}$ sii $\mathcal{L}(\mathcal{G}) = \{w / \mathfrak{A}(w) \neq \text{ERROR}\}$

En ese caso, si $\mathfrak{A}(w) = \Pi$, decimos que Π es un análisis sintáctico por la derecha de w. $\diamond$

Ejemplo: Para $\mathcal{G}$ definida por: $\left[\begin{array}{ll} (1) & S \to SaSb \\ (2) & S \to \varepsilon \end{array} \right]$

Función f: $\forall \alpha \in \Gamma^*, \forall x \in (\Sigma \cup \{\$\})^*$

$f(\alpha S, cx) =$ SALTO si $c \in \{a, b\}$
$f(\alpha c, dx) =$ REDUCC si $c \in \{a, b\}$ y $d \in \{a, b\}$
$f(\$, ax) =$ REDUCC
$f(\alpha b, \$) =$ REDUCC
$f(\alpha X, \$) =$ ERROR si $X \in \{S, a\}$
$f(\$, bx) =$ ERROR
$f(\$S, \$) =$ ACEPTAR
$f(\$, \$) =$ ERROR

Función g: $\forall \alpha \in \Gamma^*, \forall x \in (\Sigma \cup \{\$\})^*$

$g(\$, ax) = 2$
$g(\alpha a, cx) = 2,$ si $c \in \{a, b\}$
$g(\$SaSb, cx) = 1,$ si $c \in \{a, \$\}$
$g(\alpha aSaSb, cx) = 1,$ si $c \in \{a, b\}$
$g(\alpha, x) =$ ERROR, en otro caso

Análisis de $w = aabb$, mediante $\mathfrak{a}_{=}(f, g)$

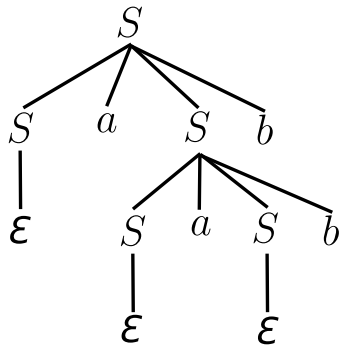
$(\$, aabb\$, \varepsilon) \vdash (\$S, aabb\$, 2) \vdash (\$Sa, abb\$, 2) \vdash (\$SaS, abb\$, 22) \vdash$
$(\$SaSa, bb\$, 22) \vdash (\$SaSaS, bb\$, 222) \vdash (\$SaSaSb, b\$, 222) \vdash$
$(\$SaS, b\$, 2221) \vdash (\$SaSb, \$, 2221) \vdash (\$S, \$, 22211) \vdash$ ACEPTAR

De modo que $\mathfrak{a}(w) = 22211$

La derivación resultante sería (el orden de aplicación de las reglas es inverso al resultado del algoritmo):

$$S \Rightarrow SaSb \Rightarrow SaSaSbb \Rightarrow SaSabb \Rightarrow Saabb \Rightarrow aabb$$

Y el árbol de derivación resultante sería:



3.2 Gramáticas de precedencia simple

Tipo más simple de gramáticas que admiten analizadores salto-reducción

Definición (Relaciones de precedencia de Wirth-Weber)

Sea $\mathcal{G} = (N, \Sigma, P, S)$ *una GIC, se definen las* relaciones de precedencia de Wirth-Weber *sobre* $N \cup \Sigma$ *de la forma:*

1. $\mathbf{X} \lessdot \mathbf{Y} \Leftrightarrow \exists A \to \alpha \mathbf{X} B \beta \in P$ *tal que* $B \overset{+}{\Rightarrow} \mathbf{Y}\gamma$
2. $\mathbf{X} \doteq \mathbf{Y} \Leftrightarrow \exists A \to \alpha \mathbf{X}\mathbf{Y} \beta \in P$
3. $\mathbf{X} \gtrdot \mathbf{a} \Leftrightarrow \exists A \to \alpha B \mathbf{Y} \beta \in P$ *tal que* $\begin{cases} B \overset{+}{\Rightarrow} \gamma \mathbf{X} \\ Y \overset{*}{\Rightarrow} \mathbf{a}\delta \end{cases}$

donde $\mathbf{X}, \mathbf{Y} \in (N \cup \Sigma \cup \{\$\})$ *y* $\mathbf{a} \in \Sigma$.

(Observar que tenemos $Y = a$, $\delta = \varepsilon$ *si* $Y \overset{0}{\Rightarrow} a\delta$*)* $\diamond$

Nota: Mayor precedencia si *"está más abajo"* (se reduce antes)

En un árbol de derivación:

1. $X \lessdot Y$ si X está en un NIVEL SUPERIOR a Y
2. $X \doteq Y$ si X está en el MISMO NIVEL que Y
3. $X \gtrdot a$ si X está en un NIVEL SUPERIOR o IGUAL a a y, además, X va inmediatamente antes de a.

Graficamente:

Para el símbolo \$: $\begin{cases} \$ \lessdot \mathbf{Z}, & \forall Z \text{ tal que } S \overset{+}{\Rightarrow} \mathbf{Z}\alpha \\ \mathbf{Z} \gtrdot \$, & \forall Z \text{ tal que } S \overset{+}{\Rightarrow} \alpha\mathbf{Z} \end{cases}$

Definición (Gramática propia y gramática inversible)

Una GIC $\mathcal{G} = (N, \Sigma, P, S)$ *sin reglas nulas* ($A \overset{+}{\Rightarrow} A$)*, sin símbolos inútiles y sin reglas-*ε *se dice que es una* gramática propia

Una GIC $\mathcal{G} = (N, \Sigma, P, S)$ *se dice* inversible *si y sólo si:*

$$\forall \left\{ \begin{array}{l} A \to \alpha \ \in P \\ B \to \alpha \ \in P \end{array} \right\} \Rightarrow A = B \quad \left(\begin{array}{l} \textit{no hay 2 reglas con} \\ \textit{igual parte derecha} \end{array} \right)$$

$\diamond$

Definición (Gramática de precedencia y precedencia simple)

Sea $\mathcal{G} = (N, \Sigma, P, S)$ *una GIC propia, tal que como mucho existe una única relación Wirth-Weber entre cualquier par de símbolos de* $N \cup \Sigma$*. Entonces* $\mathcal{G}$ *es una* gramática de precedencia*.*

Una gramática de precedencia, que además sea inversible, se dice que es una gramática de precedencia simple $\diamond$

Ejemplo: Tabla de relaciones de precendecia para $\mathcal{G}$ definida por $\left[\begin{array}{l} S \to aSSb \\ S \to c \end{array} \right]$

	S	a	b	c	\$
S	$\doteq$	$\lessdot$	$\doteq$	$\lessdot$	
a	$\doteq$	$\lessdot$		$\lessdot$	
b		$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$
c		$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$
\$		$\lessdot$		$\lessdot$	

Lema: Propagación de relaciones precedencia

Sea $\mathcal{G} = (N, \Sigma, P, S)$ una GIC propia, tenemos:

1. $\left.\begin{array}{l} \mathbf{X} \lessdot A \text{ ó } \mathbf{X} \doteq A \\ A \to \mathbf{Y}\alpha \in P \end{array}\right\} \Rightarrow \mathbf{X} \lessdot \mathbf{Y}$
2. $\left.\begin{array}{l} A \lessdot \mathbf{a} \text{ ó } A \doteq \mathbf{a} \text{ ó } A \gtrdot \mathbf{a} \\ A \to \alpha\mathbf{X} \in P \end{array}\right\} \Rightarrow \mathbf{X} \gtrdot \mathbf{a}$

■

Teorema: (Base del funcionamiento de los analizadores de precedencia)

Sea $\mathcal{G} = (N, \Sigma, P, S)$ una GIC propia. Para cualquier derivación de la forma:

$$\$S\$ \overset{n}{\Rightarrow} X_p X_{p-1} \cdots X_{k+1} \mathbf{A} a_1 \cdots a_q \Rightarrow X_p X_{p-1} \cdots X_{k+1} \mathbf{X_k} \cdots \mathbf{X_1} a_1 \cdots a_q$$

(se aplicó la regla $A \to X_k \cdots X_1$)

se verifica:

1. $X_{i+1} \lessdot X_i$ ó $X_{i+1} \doteq X_i, \quad \forall p < i < k$
2. $X_{k+1} \lessdot X_k$
3. $X_{i+1} \doteq X_i, \quad \forall k > i \geq 1$
4. $X_1 \gtrdot a_1$

■

Corolario:

Si además $\mathcal{G}$ es una gramática de precedencia, ***no habrá otras relaciones*** *entre esos símbolos.* ■

Corolario:

Toda gramática de precedencia simple es ***no ambígua*** ■

Teorema: (algoritmo de análisis)

Existe un algoritmo para la generación de un analizador sintáctico de precedencia simple ■

Demostración:

El algoritmo el siguiente:

Generación de analizadores de precedencia simple

Entrada: Una GIC $\mathcal{G} = (N, \Sigma, P, S)$ de precedencia simple con $P = \{P_1, \ldots, P_n\}$

Salida: El algoritmo de precedencia simple $\mathfrak{a}(f, g)$ para $\mathcal{G}$

La **función de acciones** f dependerá sólo de $\begin{cases} \text{el TOP de la pila} \\ \text{el siguiente símbolo de entrada} \end{cases}$

Se define $f : (N \cup \Sigma \cup \{\$\}) \times (\Sigma \cup \{\$\}) \to \{\text{SALTO, REDUCC., ACEPTAR, ERROR}\}$ como:

- $f(X, a) = \text{SALTO}$ si $X \lessdot a$ ó $X \doteq a$
- $f(X, a) = \text{REDUCCIÓN}$ si $X \gtrdot a$
- $f(\$S, \$) = \text{ACEPTAR}$
- $f(X, a) = \text{ERROR}$ en otro caso

Resumen: $\begin{cases} \text{1: recorrer entrada de Izq. a Der., consultando la cima de la pila} \\ \text{2: SALTO (meter en pila) mientras tengamos "}\doteq\text{" ó "}\lessdot\text{"} \\ \text{3: REDUCIR al encontrar un "}\gtrdot\text{"} \end{cases}$

La **función de reducción** g depende sólo de la pila

Se define $g : (N \cup \Sigma \cup \{\$\}) \to \{1, 2, \ldots, n\}$ como:

- $g(X_{k+1}\mathbf{X_k}\mathbf{X_{k-1}}\cdots\mathbf{X_1}, \varepsilon) = i$ si $\begin{cases} X_{k+1} \lessdot X_k \\ X_{j+1} \doteq X_j, \ \forall k > j \geq 1 \\ P_i = A \to X_k X_{k-1} \cdots X_1 \end{cases}$
- $g(\alpha, \varepsilon) = \text{ERROR}$ en otro caso

Resumen: $\begin{cases} \text{1: buscar cadena a reducir recorriendo pila hasta encontrar primer "}\lessdot\text{"} \\ \text{2: sólo puede haber una regla con ese lado derecho} \end{cases}$

La demostración de que el algoritmo es el correcto es trivial por construcción.

Ejemplo: Analizador de precedencia simple para $\mathcal{G}$ definida por $\left[\begin{array}{l} S \to aSSb \\ S \to c \end{array}\right]$

Anteriormente habíamos calculado la tabla de relaciones de precedencia:

	S	a	b	c	\$
S	$\doteq$	$\lessdot$	$\doteq$	$\lessdot$	
a	$\doteq$	$\lessdot$		$\lessdot$	
b		$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$
c		$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$
\$		$\lessdot$		$\lessdot$	

de la cual se puede deducir automáticamente f.

Respecto a g, tenemos que:

$$\begin{array}{ll} g(XaSSb) = 1 & \text{si } X \in \{S, a, \$\} \\ g(Xc) = 2 & \text{si } X \in \{S, a, \$\} \\ g(\alpha) = \text{ERROR} & \text{en otro caso} \end{array}$$

Consideremos ahora la entrada $w = accb$, $\mathfrak{A}$ analizaría la entrada como:

$$(\$, accb\$, \varepsilon) \vdash (\$a, ccb\$, \varepsilon) \vdash (\$ac, cb\$, \varepsilon) \vdash (\$aS, cb\$, 2) \vdash (\$aSc, b\$, 2) \vdash (\$aSS, b\$, 22) \vdash (\$aSSb, \$, 22) \vdash (\$Sb, \$, 221) \vdash \text{ACEPTAR}$$

Mientras que para la entrada $= acb$ tendríamos:

$$(\$, acb\$, \varepsilon) \vdash (\$a, cb\$, \varepsilon) \vdash (\$ac, b\$, \varepsilon) \vdash (\$aS, b\$, 2) \vdash (\$aSb, \$, 2) \vdash \text{ERROR}$$

Conflictos de análisis

- Si la gramática no es de predecencia simple, aparecerá más de una relación en alguna casilla de la tabla de precedencia
- Conflictos **salto-reducción**
  - Hay casillas con "$\doteq$" y "$\gtrdot$" o con "$\lessdot$" y "$\gtrdot$"
  - Se puede seguir metiendo en la pila (salto)[$\doteq$, $\lessdot$] o bien reducir la regla que corresponda [$\gtrdot$].
  - *Condición necesaria (pero no suficiente):* el lado derecho de una regla es prefijo del lado derecho de otra distinta.
- Conflictos **reducción-reducción**
  - Hay casillas con "$\doteq$" y "$\lessdot$"
  - Se puede reducir la secuencia actualmente seleccionada en la cima de la pila [$\lessdot$] o seguir buscando [$\doteq$] en la pila otra secuencia mayor para reducir una regla distinta.
  - *Condición necesaria (pero no suficiente):* el lado derecho de una regla es sufijo del lado derecho de otra distinta.
- También hay conflicto reducción-reducción si la gramática no es inversible (hay 2 reglas con igual lado derecho, $A \to \beta$ y $B \to \beta$)

3.3 Gramáticas de precedencia débil

Idea básica: Relajar la relaciones de precedencia simple $\lessdot, \doteq, \gtrdot$

- Se exige que "$\gtrdot$" sea disjunta de "$\lessdot$" y "$\doteq$"
- Se permite que "$\lessdot$" y "$\doteq$" no sean disjuntos
  (Se permiten casillas con esas 2 relaciones [conflictos reducción-reducción])

Se complica la delimitación de la secuencia a reducir en la pila

1. Al detectar una relación $\gtrdot$, se busca entre los símbolos de la cima de la pila la regla o reglas cuya parte derecha encaje con esos símbolos
2. Si hay más de una regla, se escoge aquella cuya parte derecha tenga mayor longitud

Para evitar conflictos sólo una regla debe verificar (2)

Definición (Gramática de precedencia débil)

Sea $\mathcal{G} = (N, \Sigma, P, S)$ una GIC propia. Decimos que $\mathcal{G}$ es una gramática de precedencia débil *si y sólo si:*

1. La relación de precedencia $\gtrdot$ es disjunta del conjunto $\{\lessdot, \doteq\}$

2. $\left\{ \begin{array}{l} A \to \alpha X \beta \\ B \to \beta \end{array} \right\} \Rightarrow X \not\lessdot B \;\; y \;\; X \not\doteq B$

$\diamond$

La condición (2) garantiza que ante dos reducciones posibles sólo se llegará a aplicar la reducción más larga, dado que B no podría reducirse.

- Como $\left\{ \begin{array}{l} X \not\lessdot B \\ X \not\doteq B \end{array} \right\}$, no habrá ambiguedad en la pila a la hora de decidir si reducir "β" o "seguir buscando" para reducir "$\alpha X \beta$"
- Siempre que la cima de la pila sea "$\alpha X \beta$" se reducirá la regla $A \to \alpha X \beta$ (regla más larga) y no $B \to \beta$

Ejemplo: La gramática G_1 es de precedencia débil.

$$G_1 : \left| \begin{array}{llllll} E & \to E+T & T & \to T*F & F & \to (E) \\ & \mid +T & & \mid F & & \mid constante \\ & \mid T & & & & \end{array} \right|$$

Su tabla de relaciones de precedencia sería:

	E	T	F	a	$($	$)$	$+$	$*$	$\$$
E						$\doteq$	$\doteq$		
T						$\gtrdot$	$\gtrdot$	$\doteq$	$\gtrdot$
F						$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$
a						$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$
$($	$\lessdot, \doteq$	$\lessdot$	$\lessdot$	$\lessdot$	$\lessdot$		$\lessdot$		
$)$						$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$
$+$		$\lessdot, \doteq$	$\lessdot$	$\lessdot$	$\lessdot$				
$*$			$\doteq$	$\lessdot$	$\lessdot$				
$\$$	$\lessdot$	$\lessdot$	$\lessdot$	$\lessdot$	$\lessdot$		$\lessdot$		

Se cumple la primera condición (sólo hay conflictos $\lessdot, \doteq$)

Para la segunda condición se deben estudiar los pares de reglas conflictivas:

$$\left\{ \begin{array}{l} E \to E+T \\ E \to +T \end{array} \right\} \quad \left\{ \begin{array}{l} E \to +T \\ E \to T \end{array} \right\} \quad \left\{ \begin{array}{l} T \to T*F \\ T \to F \end{array} \right\}$$

Podemos ver que, en efecto:

$$\begin{array}{lll} E \not\lessdot E & + \not\lessdot E & * \not\lessdot T \\ E \not\doteq E & + \not\doteq E & * \not\doteq T \end{array}$$

Lema:

Sea $\mathcal{G} = (N, \Sigma, P, S)$ una gramática de precedencia débil, y sea la derivación:

$$\$S\$ \stackrel{*}{\Rightarrow} \gamma C w \Rightarrow \delta X \beta w \ / \ \exists \ \begin{array}{l} A \to \alpha X \beta \in P, \ |\alpha| \geq 1 \\ B \to \beta \in P \end{array}$$

Entonces la última producción aplicada no fue $B \to \beta$ ■

Lema:

Sea $\mathcal{G} = (N, \Sigma, P, S)$ una gramática de precedencia débil, tal que $B \to \beta \in P$ y $\mathcal{G}$ inversible, y sea la derivación $\$S\$ \stackrel{}{\Rightarrow} \gamma C w \Rightarrow \delta X \beta w$, entonces:*

$$\nexists A \to \alpha X \beta \in P \Rightarrow \{ \begin{array}{l} C = B \\ \gamma = \delta X \end{array}$$

Esto es, la última regla aplicada ha sido $B \to \beta$. ■

Teorema: (algoritmo de análisis)

Existe un algoritmo de salto/reducción para las gramáticas de precedencia débil, inversibles. ■

Demostración: El algoritmo es el siguiente:

Generación de analizadores de precedencia débil

Entrada: Una GIC $\mathcal{G} = (N, \Sigma, P, S)$ de precedencia débil con $P = \{P_1, \ldots, P_n\}$

Salida: El algoritmo de precedencia débil $\mathfrak{a}(f, g)$ para $\mathcal{G}$

Función de acciones f: *(IDEM que precedencia simple)*

- $f(X, a) =$ SALTO si $X \lessdot a$ ó $X \doteq a$
- $f(X, a) =$ REDUCCIÓN si $X \gtrdot a$
- $f(\$S, \$) =$ ACEPTAR
- $f(X, a) =$ ERROR en otro caso

Función de reducción g:

- $g(X\beta, \varepsilon) = i$ si $\left\{ \begin{array}{l} P_i \equiv B \to \beta \in P \\ \nexists A \to \alpha X \beta \in P \end{array} \right.$
- $g(\alpha, \varepsilon) =$ ERROR en otro caso

Se busca en la pila la **reducción más larga** que se corresponda con el lado derecho de alguna regla.

La demostración de que el algoritmo es el correcto es trivial por construcción.

■

Resolución de conflictos

Supongamos un conflicto del tipo $X \doteq Y, X \gtrdot Y$. Puesto que $X \doteq Y$, sabemos que $\exists A \rightarrow \alpha X Y \beta \in P$. Reemplazando en esta regla X por B, y añadiendo a P la regla $B \rightarrow X$, eliminaremos la relación $X \doteq Y$, y, por tanto, el conflicto.

El único problema que puede presentarse es la inversibilidad de la nueva gramática. Para evitarlo, debemos asegurarnos que no existe ninguna otra regla con X como parte derecha.

Ejemplo: Sea $\mathcal{G}$ la GIC definida por las reglas:

$$\begin{aligned} S &\rightarrow 0S11 \\ &\mid 011 \end{aligned}$$

cuya tabla de relaciones de precedencia es:

	S	0	1	$\$$
S			$\doteq$	
0	$\doteq$	$\lessdot$	$\doteq$	
1			$\doteq, \gtrdot$	$\gtrdot$
$\$$		$\lessdot$		

Debido al conflicto de desplazamiento-reducción en $1 \doteq 1, 1 \gtrdot 1$, la gramática no es de precedencia simple. En particular, la relación $1 \doteq 1$ es consecuencia de ambas reglas de la gramática.

Para solucionar el problema, podemos construír una nueva gramática $\mathcal{G}'$, con las reglas:

$$\begin{aligned} S &\rightarrow 0SA1 \\ &\mid 0A1 \\ A &\rightarrow 1 \end{aligned}$$

con $\mathcal{L}(\mathcal{G}) = \mathcal{L}(\mathcal{G}')$ y $\mathcal{G}'$ inversible.

Las relaciones de precedencia de la nueva gramática son:

	S	A	0	1	$\$$
S		$\doteq$		$\lessdot$	
A				$\doteq$	
0	$\doteq$	$\doteq$	$\lessdot$	$\lessdot$	
1				$\gtrdot$	$\gtrdot$
$\$$			$\lessdot$		

con lo que $\mathcal{G}'$ es de precedencia simple en inversible

Transformaciones similares pueden usarse para eliminar conflictos del tipo $X \gtrdot Y, X \lessdot Y$ ó $X \lessdot Y, X \doteq Y$.

En caso de que los cambios introducidos de ese modo en la gramática original destruyan la inversibilidad de la misma, tendremos que aplicar otro tipo de técnicas basadas en la eliminación de reglas.

Ejemplo: Sea $\mathcal{G}$ la GIC inversible definida por las reglas:

$$\begin{array}{llll} E \to E + T & T \to T * F & F \to a & L \to L, E \\ \quad | \; T & \quad | \; F & | \; (E) & \quad | \; E \\ & & | \; a(L) & \end{array}$$

cuyas relaciones de precedencia se definen en la siguiente tabla:

	E	T	F	L	a	$($	$)$	$+$	$*$	$,$	$\$$
E							$\doteq, \gtrdot$	$\doteq$		$\gtrdot$	
T							$\gtrdot$	$\gtrdot$	$\doteq$	$\gtrdot$	$\gtrdot$
F							$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$
L							$\doteq$			$\doteq$	
a						$\doteq$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$
$($	$\doteq, \lessdot$	$\lessdot$	$\lessdot$	$\doteq, \lessdot$	$\lessdot$	$\lessdot$					
$)$							$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$
$+$		$\doteq, \lessdot$	$\lessdot$		$\lessdot$	$\lessdot$					
$*$			$\doteq$		$\lessdot$	$\lessdot$					
$,$	$\doteq, \lessdot$	$\lessdot$	$\lessdot$		$\lessdot$	$\lessdot$					
$\$$	$\lessdot$	$\lessdot$	$\lessdot$		$\lessdot$	$\lessdot$					

Claramente, $\mathcal{G}$ no es de precedencia débil, debido al conflicto de desplazamiento-reducción, producido por las relaciones:

$E \doteq)$ originada por la regla $F \to (E)$
$E \gtrdot)$ originada por las reglas $F \to a(L)$ y $L \to L, E$

Usando la misma técnica que en el ejemplo anterior, se cambiaría $F \to (E)$ por:

$$F \to (E')$$
$$E' \to E$$

Pero la gramática resultante no sería inversible, al haber dos reglas con la misma parte derecha:

$$E' \to E$$
$$L \to E$$

Otra posibilidad: eliminar $F \to a(L)$, sustituyendo L por su parte derecha. La gramática resultante sería:

$$\begin{array}{llll} E \to E + T & T \to T * F & F \to a & L \to L, E \\ \quad \mid T & \quad \mid F & \quad \mid (E) & \quad \mid E \\ & & \quad \mid a(L, E) & \\ & & \quad \mid a(E) & \end{array}$$

con lo que $E \not\gtrdot)$, y la nueva gramática es de precedencia débil.