

Procesadores de Lenguaje

**ESTUDIO DE LOS REQUERIMIENTOS
DE UN ALGORITMO DE ANÁLISIS
SINTÁCTICO ASCENDENTE**

Juan Antonio Pérez Ortiz

Copyright (c) 2003 Juan Antonio Pérez Ortiz

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, version 1.2, or any later version published by the Free Software Foundation; with no Invariant Section, no Front-Cover Texts, and no Back-Cover Texts.

Se concede permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre de GNU, versión 1.2, o cualquier versión posterior publicada por la Free Software Foundation; sin secciones invariantes ni textos de portada o contraportada.

Repaso del análisis descendente

- La cadena se recorre una única vez de izquierda a derecha.
- Se usa una pila para almacenar la parte del árbol predicha y aún no explorada.
- Dos acciones básicas:
 - Emparejar
 - Expandir
- El algoritmo produce una derivación por la izquierda de la cadena de entrada.
- El árbol se recorre (construye) de la raíz a las hojas.

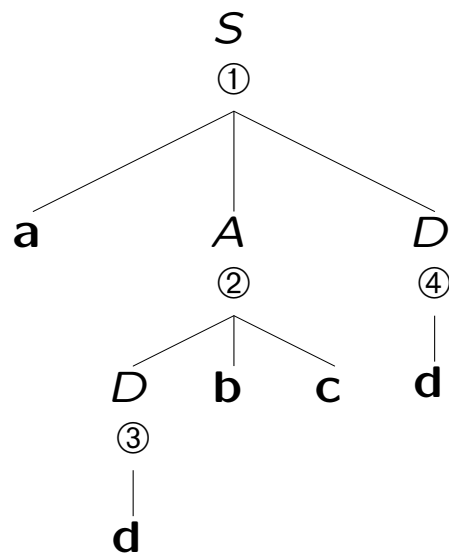
Repaso del análisis descendente

G_0 :

$$\begin{aligned} S &\longrightarrow a A D \\ A &\longrightarrow D b c \mid e \\ D &\longrightarrow d \end{aligned}$$

Cadena: **a d b c d \$**

Árbol de análisis sintáctico (los nodos intermedios se numeran según el orden en el que se van visitando en el análisis descendente):



De las hojas a la raíz

¿Podemos escribir un algoritmo que construya un árbol de análisis sintáctico de las hojas a la raíz leyendo la entrada de izquierda a derecha?

¿Nos interesa hacerlo por algún motivo en especial?

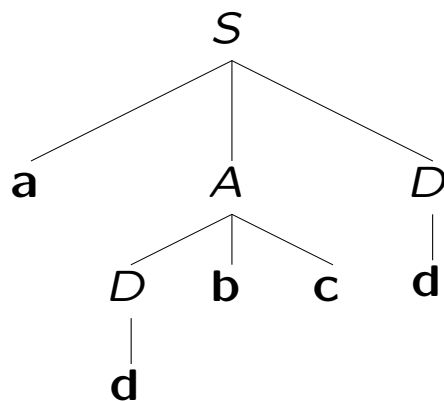
¿Cuáles serían los requerimientos de este algoritmo?

Sea G_0 :

$$\begin{array}{lcl} S & \longrightarrow & \mathbf{a A D} \\ A & \longrightarrow & \mathbf{D b c} \mid \mathbf{e} \\ D & \longrightarrow & \mathbf{d} \end{array}$$

Cadena: **a d b c d \$**

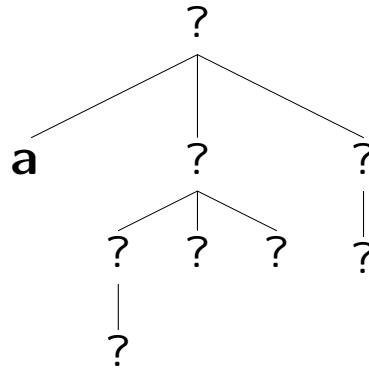
Objetivo:



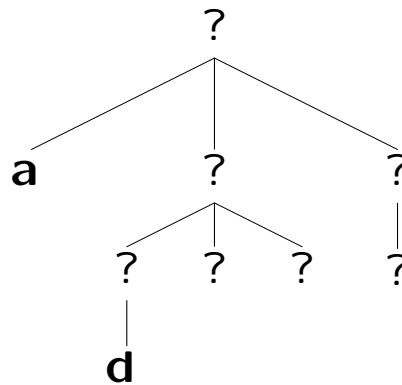
- Dos acciones básicas:
 - Leer un terminal
 - Ascender

Traza del análisis ascendente

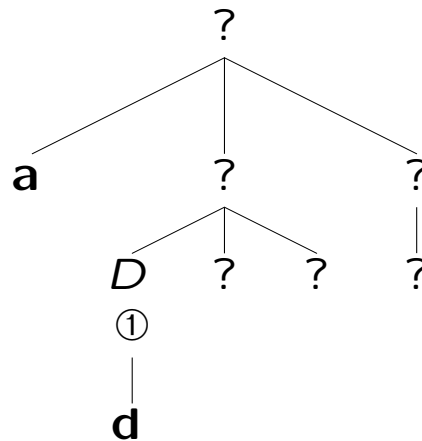
Paso 1. Leer **a** :



Paso 2. Leer **d** :

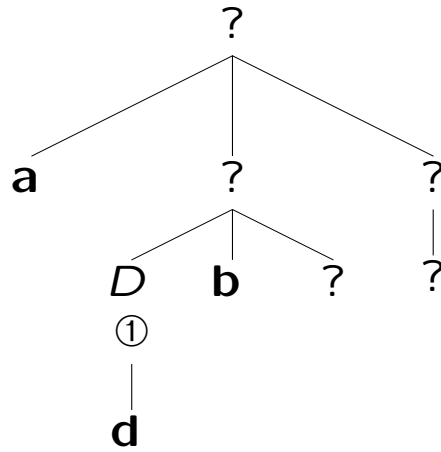


Paso 3. Ascender a D ①:

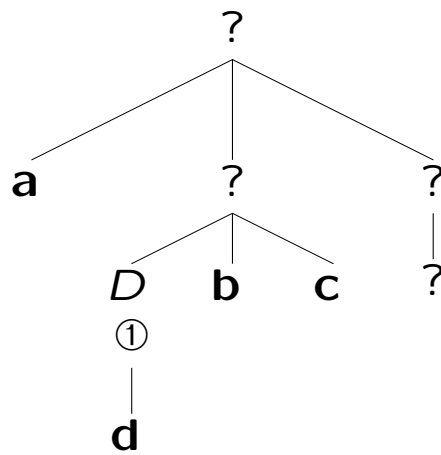


Traza del análisis ascendente

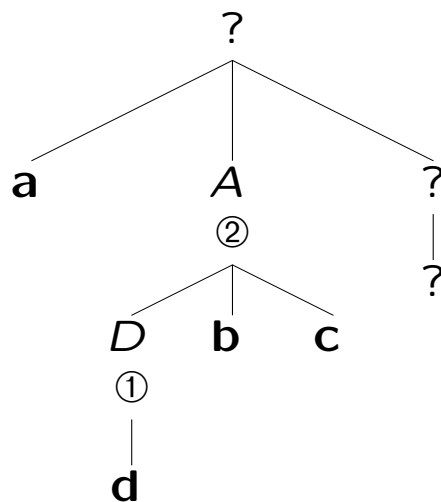
Paso 4. Leer **b** :



Paso 5. Leer **c** :

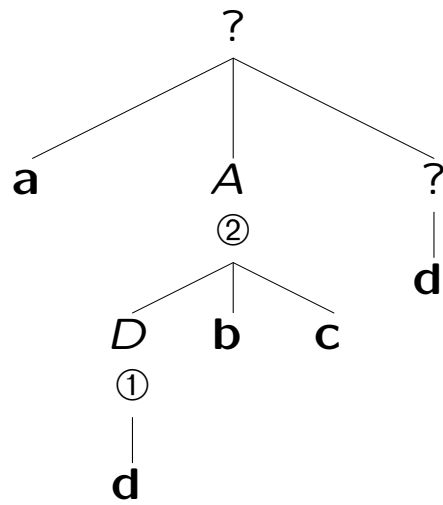


Paso 6. Ascender a A ②:

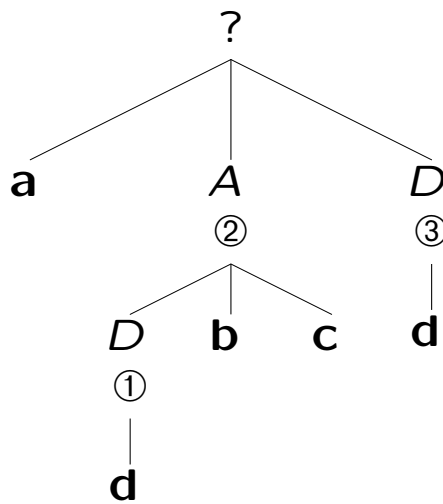


Traza de un análisis ascendente

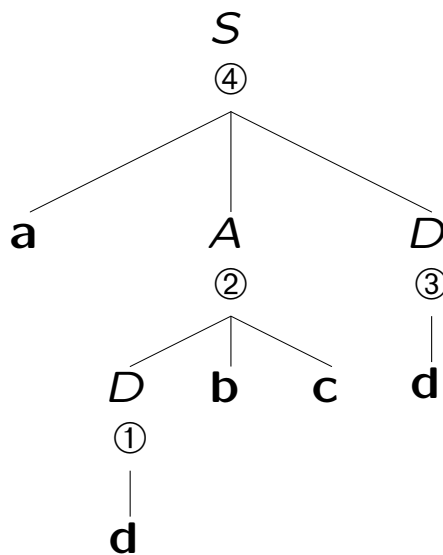
Paso 7. Leer **d** :



Paso 8. Ascender a *D* ③:



Paso 9. Ascender a *S* ④:

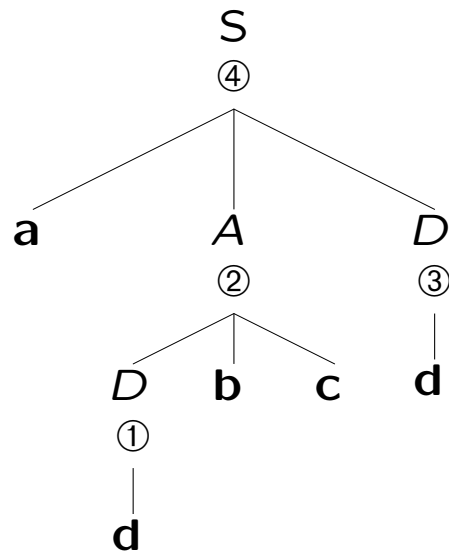


Boceto n.º 0

- Dos acciones básicas:
 - Leer \equiv desplazar
 - Ascender \equiv reducir
- No podemos ascender (reducir) hasta el padre hasta que no tengamos todos sus hijos.
- En cuanto tengamos todos los hijos “ascendemos” (reducimos) hasta el padre.
- No podemos “olvidar” ningún hermano hasta que no hayamos ascendido hasta el padre. Necesitamos un lugar donde ir almacenándolos; una pila es adecuada.

La pila de un analizador ascendente

Árbol de análisis sintáctico:



Traza:

Pila	Entrada	Acción
\$	a d b c d \$	desplazar
\$ a	d b c d \$	desplazar
\$ a d	b c d \$	reducir por $D \rightarrow d$ ①
\$ a D	b c d \$	desplazar
\$ a D b	c d \$	desplazar
\$ a D b c	d \$	reducir por $A \rightarrow D b c$ ②
\$ a A	d \$	desplazar
\$ a A d	\$	reducir por $D \rightarrow d$ ③
\$ a A D	\$	reducir por $S \rightarrow a A D$ ④
\$ S	\$	aceptar

Análisis por desplazamiento y reducción

- Reducir por $A \rightarrow \alpha$: desapilar $|\alpha|$ elementos de la pila y apilar A .
- Desplazar: apilar el símbolo de preanálisis y actualizar el preanálisis al siguiente símbolo de la entrada.

Boceto n.º 1

- Al principio apilar \$.
- Reducir en cuanto se detecte un sufijo de la pila que coincida con la parte derecha de una regla.
- Desplazar, en otro caso.
- Aceptar si la pila tiene $\$S$ y el preanálisis es \$.

¿Qué ventajas aporta el análisis ascendente?

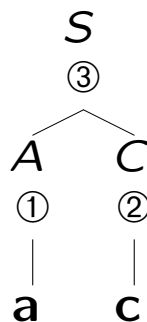
¿Es el análisis ascendente más potente que el descendente?

G_1 :

$$\begin{aligned} S &\longrightarrow AB \mid AC \\ A &\longrightarrow a \\ B &\longrightarrow b \\ C &\longrightarrow c \end{aligned}$$

G_1 no es LL(1).

Árbol de análisis sintáctico:

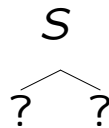


Traza del análisis ascendente:

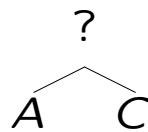
Pila	Entrada	Acción
\$	a c \$	desplazar
\$ a	c \$	reducir por $A \rightarrow a$ ①
\$ A	c \$	desplazar
\$ A c	\$	reducir por $C \rightarrow c$ ②
\$ A C	\$	reducir por $S \rightarrow AC$ ③
\$ S	\$	aceptar

Propiedades

- El poder observar toda la parte derecha de una regla antes de tomar una decisión confiere cierta ventaja a los analizadores ascendentes sobre los descendentes.
- Análisis descendente:



- Análisis ascendente:



Reglas con la misma parte derecha

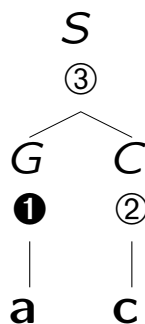
¿Qué pasa si dos reglas tienen la misma parte derecha?

G_2 :

$$\begin{array}{l}
 S \longrightarrow FB \mid GC \\
 F \longrightarrow a \\
 G \longrightarrow a \mid b \\
 B \longrightarrow b \\
 C \longrightarrow c
 \end{array}$$

G_2 no es ambigua.

Árbol de análisis sintáctico:



Traza:

Pila	Entrada	Acción
\$	a c \$	desplazar
\$ a	c \$	¿reducir por $G \rightarrow a$ o $F \rightarrow a$? ❶
...

Reglas con la misma parte derecha

G_2 :

$$\begin{array}{l}
 S \longrightarrow FB \mid GC \\
 F \longrightarrow a \\
 G \longrightarrow a \mid b \\
 B \longrightarrow b \\
 C \longrightarrow c
 \end{array}$$

Traza cuando se reduce por $F \rightarrow a$:

Pila	Entrada	Acción
\$	a c \$	desplazar
\$ a	c \$	reducir* por $F \rightarrow a$
\$ F	c \$	desplazar
\$ F c	\$	reducir por $C \rightarrow c$
\$ F C	\$	¡no podemos seguir!

Traza cuando se reduce por $G \rightarrow a$:

Pila	Entrada	Acción
\$	a c \$	desplazar
\$ a	c \$	reducir por $G \rightarrow a$
\$ G	c \$	desplazar
\$ G c	\$	reducir por $C \rightarrow c$
\$ G C	\$	reducir por $S \rightarrow G C$
\$ S	\$	aceptar

¿Cómo podemos asegurar que tomamos la decisión correcta?

Boceto n.º 2

- Al principio apilar \$.
- Reducir en cuanto se detecte un sufijo adecuado.
- Desplazar, en otro caso.
- Aceptar si la pila tiene $\$S$ y el preanálisis es \$.

Además:

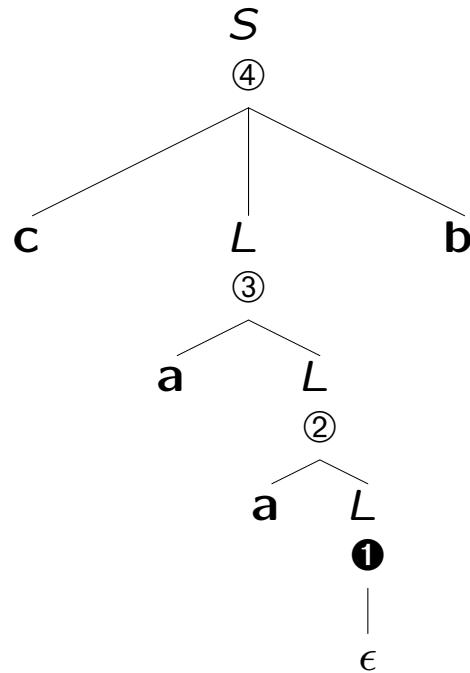
- Para saber por qué regla reducir puede ser necesario mirar el símbolo de preanálisis y compararlo con los sucesores de los no terminales de la parte izquierda de las reglas correspondientes.

Producciones ϵ

G_3 :

$$\begin{aligned} S &\longrightarrow \mathbf{c L b} \\ L &\longrightarrow \mathbf{a L} \mid \epsilon \end{aligned}$$

Árbol de análisis sintáctico:



Traza:

Pila	Entrada	Acción
\$	c a a b \$	desplazar
\$ c	a a b \$	desplazar
\$ c a	a b \$	desplazar
\$ c a a	b \$	reducir por $L \rightarrow \epsilon$ ❶
\$ c a a L	b \$	reducir por $L \rightarrow \mathbf{a L}$ ❷
\$ c a L	b \$	reducir por $L \rightarrow \mathbf{a L}$ ❸
\$ c L	b \$	desplazar
\$ c L b	\$	reducir por $S \rightarrow \mathbf{c L b}$ ❹
\$ S	\$	aceptar

La aparición de **b** como símbolo de preanálisis *dispara* la reducción por $L \rightarrow \epsilon$ ❶.

Propiedades

- Reducir no siempre implica desapilar, como puede verse con las reglas del tipo $L \rightarrow \epsilon$.
- Desplazar no equivale solo a apilar; también se avanza al siguiente símbolo de la entrada.
- La única forma de que un terminal entre en la pila es a través de un desplazamiento.
- Un no terminal solo entra en la pila tras una reducción.
- Las reglas recursivas por la derecha hacen que la pila pueda crecer indefinidamente con la entrada.

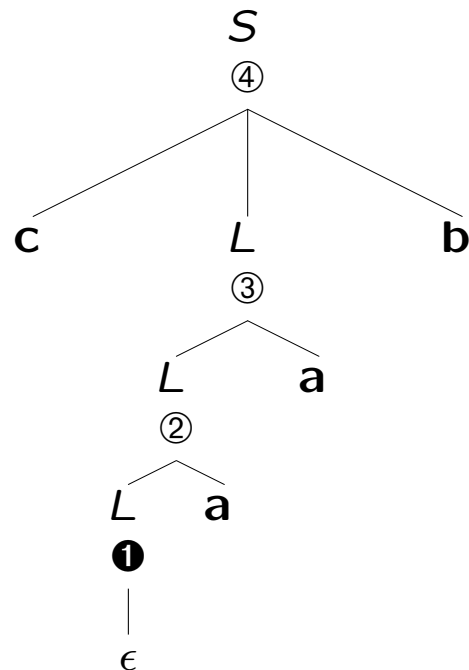
Recursividad por la izquierda

G_4 :

$$\begin{aligned} S &\longrightarrow c L b \\ L &\longrightarrow L a \mid \epsilon \end{aligned}$$

G_4 no es LL(1).

Árbol de análisis sintáctico:



Traza:

Pila	Entrada	Acción
\$	c a a b \$	desplazar
\$ c	a a b \$	reducir por $L \rightarrow \epsilon$ 1
\$ c L	a a b \$	desplazar
\$ c L a	a b \$	reducir por $L \rightarrow L a$ 2
\$ c L	a b \$	desplazar
\$ c L a	b \$	reducir por $L \rightarrow L a$ 3
\$ c L	b \$	desplazar
\$ c L b	\$	reducir por $S \rightarrow c L b$ 4
\$ S	\$	aceptar

La aparición de la primera **a** como símbolo de preanálisis *dispara* la reducción por $L \rightarrow \epsilon$ **1**.

Propiedades

- El tamaño de la pila permanece limitado con las reglas recursivas por la izquierda.
- Al contrario de lo que ocurre con los analizadores descendentes, en los analizadores ascendentes la recursividad por la izquierda es preferible a la recursividad por la derecha.

Más de un símbolo de preanálisis

G_5 :

$$\begin{aligned}
 S &\longrightarrow FIB \mid GIC \\
 F &\longrightarrow a \\
 G &\longrightarrow a \mid b \\
 B &\longrightarrow b \\
 C &\longrightarrow c \\
 I &\longrightarrow i
 \end{aligned}$$

A partir de este momento, no usaremos el árbol de análisis sintáctico como apoyo.

Cadena: **a i c \$**

Traza:

Pila	Entrada	Acción
\$	a i c \$	desplazar
\$ a	i c \$	¿reducir por $G \rightarrow a$ o $F \rightarrow a$?
...

¿Qué ocurre si $I \rightarrow i^n$?

Boceto n.º 3

- Al principio apilar \$.
- Reducir en cuanto se detecte un sufijo adecuado.
- Desplazar, en otro caso.
- Aceptar si la pila tiene $\$S$ y el preanálisis es \$.

Además:

- Para saber por qué regla reducir puede ser necesario mirar uno o más símbolos de preanálisis.

Mirar atrás

G_6 :

$$\begin{array}{l}
 S \longrightarrow A I D \mathbf{b} \mid E I C \mathbf{b} \\
 A \longrightarrow \mathbf{a} \\
 C \longrightarrow \mathbf{c} \mid \mathbf{d} \\
 D \longrightarrow \mathbf{d} \\
 E \longrightarrow \mathbf{e} \\
 I \longrightarrow \mathbf{i}
 \end{array}$$

Cadena: **e i d b \$**

Traza:

Pila	Entrada	Acción
\$	e i d b \$	desplazar
\$ e	i d b \$	reducir por $E \rightarrow \mathbf{e}$
\$ E	i d b \$	desplazar
\$ E i	d b \$	reducir por $I \rightarrow \mathbf{i}$
\$ E I	d b \$	desplazar
\$ E I d	b \$	¿reducir por $D \rightarrow \mathbf{d}$ o $C \rightarrow \mathbf{d}$?
...

A veces es necesario mirar debajo de la cima de la pila.

Boceto n.º 4

- Al principio apilar \$.
- Reducir en cuanto se detecte un sufijo adecuado.
- Desplazar, en otro caso.
- Aceptar si la pila tiene $\$S$ y el preanálisis es \$.
- Para saber por qué regla reducir puede ser necesario mirar uno o más símbolos de preanálisis.

Además:

- Para saber por qué regla reducir puede ser necesario consultar la historia almacenada en la pila.

Contener las reducciones

G_7 :

$$\begin{array}{l}
 S \longrightarrow a L b \\
 L \longrightarrow a L \mid \epsilon
 \end{array}$$

Cadena: **a a a b \$**

Traza:

Pila	Entrada	Acción
\$	a a a b \$	desplazar
\$ a	a a b \$	desplazar
\$ a a	a b \$	desplazar
\$ a a a	b \$	reducir por $L \rightarrow \epsilon$
\$ a a a L	b \$	reducir por $L \rightarrow a L$
\$ a a L	b \$	reducir por $L \rightarrow a L$
\$ a L	b \$	reducir* por $L \rightarrow a L$
\$ L	b \$	desplazar
\$ L b	\$	¡no podemos seguir!

¿Cómo procedemos correctamente?

- Si la cima de la pila es \$ **a L** , desplazamos.
- Si la cima de la pila es \$... **a L** , reducimos.

Contener las reducciones

G_8 :

$$\begin{array}{l}
 S \longrightarrow A B c \mid A b d \\
 A \longrightarrow a \\
 B \longrightarrow b \mid c
 \end{array}$$

Cadena: **a b d \$**

Traza:

Pila	Entrada	Acción
\$	a b d \$	desplazar
\$ a	b d \$	reducir por $A \rightarrow a$
\$ A	b d \$	desplazar
\$ A b	d \$	reducir* por $B \rightarrow b$
\$ A B	d \$	desplazar
\$ A B d	\$	¡no podemos seguir!

¿Cómo procedemos correctamente?

- Si el símbolo de preanálisis es **d** , desplazamos.
- Si el símbolo de preanálisis es **c** , reducimos.

Boceto n.º 5

- Al principio apilar \$.
- Aceptar si la pila tiene \$S y el preanálisis es \$.
- Para saber por qué regla reducir puede ser necesario mirar uno o más símbolos de preanálisis.
- Para saber por qué regla reducir puede ser necesario consultar la historia almacenada en la pila.

Además:

- A veces la simple detección de un sufijo adecuado de la pila no debe disparar una reducción. Consultar el símbolo de preanálisis o cierto contexto de la pila, puede ayudar a determinar si reducir o desplazar.
- Cuando no se detecte un sufijo adecuado, desplazar.

Detección de entradas incorrectas

Hasta ahora nos hemos centrado en cadenas correctas.

G_9 :

$$\begin{array}{lcl} S & \longrightarrow & C B d \\ C & \longrightarrow & a \mid c \\ B & \longrightarrow & b \end{array}$$

Cadena incorrecta: **a d c d \$**

Traza:

Pila	Entrada	Acción
\$	a d c d \$	desplazar
\$ a	d c d \$	reducir por $C \rightarrow a$
\$ C	d c d \$	desplazar*
\$ C d	c d \$	desplazar
\$ C d c	d \$	reducir por $C \rightarrow c$
\$ C d C	d \$	desplazar
\$ C d C d	\$	error

Si la cadena es incorrecta, nunca se alcanzará la combinación $\$ S \parallel \$$.

Pero para poder indicar (en un mensaje de error) cuál ha sido el primer terminal inesperado debemos tener cuidado de no desplazar terminales que produzcan configuraciones inválidas en la pila.

Así, tras reducir por $C \rightarrow a$, solo deberíamos desplazar si el símbolo de preanálisis es **b** (un sucesor), y dar un error si es otro.

Una gramática más compleja

G_{10} :

$$\begin{array}{l} E \longrightarrow E + T \mid T \\ T \longrightarrow F * T \mid F \\ F \longrightarrow \mathbf{f} \mid (E) \end{array}$$

Problema propuesto: identifica con la gramática G_{10} algunos de los requerimientos esbozados anteriormente.

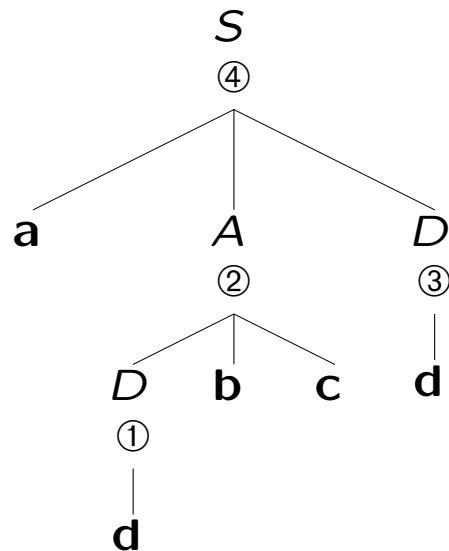
Usa cadenas como $\mathbf{f} * \mathbf{f} + \mathbf{f} \$$.

Inversa de una derivación por la derecha

G_0 :

$$\begin{array}{lcl} S & \longrightarrow & a A D \\ A & \longrightarrow & D b c \mid e \\ D & \longrightarrow & d \end{array}$$

Árbol de análisis sintáctico:



Derivación por la derecha:

$$S \xRightarrow{④} a A D \xRightarrow{③} a A d \xRightarrow{②} a D b c d \xRightarrow{①} a d b c d$$

El orden en que se van produciendo las reducciones en el analizador ascendente corresponde con la *inversa* de una derivación por la derecha.

A continuación...

- ¿Cómo podemos construir un algoritmo que considere todos los requerimientos anteriores para una gramática general?