## Análisis Sintáctico Descendente

➢ **Primeros:** $(N \cup \Sigma)^* \rightarrow \wp(\Sigma \cup \{\epsilon\})$

$$\texttt{PRI}(\alpha) = \{a \in \Sigma \mid \alpha \stackrel{*}{\Rightarrow} a\beta\} \cup \{\epsilon \mid \alpha \stackrel{*}{\Rightarrow} \epsilon\}; \qquad \alpha, \beta \in (N \cup \Sigma)^*$$

➢ **Siguientes:** $N \rightarrow \wp(\Sigma \cup \{\$\})$

$$\texttt{SIG}(A) = \{a \in \Sigma \mid S \stackrel{*}{\Rightarrow} \alpha A a \beta\} \cup \{\$ \mid S \stackrel{*}{\Rightarrow} \alpha A\}; \qquad \alpha, \beta \in (N \cup \Sigma)^*; A \in N$$



## ASD: Condición LL(1)

➢ **Condición LL(1):**

Una gramática incontextual es **LL(1)** si, para cualquier par de producciones $(A \rightarrow \alpha$ y $A \rightarrow \beta)$, se cumple:

$$\boxed{\texttt{PRI}(\alpha \cdot \texttt{SIG}(A)) \cap \texttt{PRI}(\beta \cdot \texttt{SIG}(A)) = \emptyset}$$

➢ **Propiedades:**

P1.- Si una gramática es **LL(1)**, entonces no es ambigua

P2.- Si una gramática es **LL(1)**, entonces no es recursiva a izquierdas

P3.- Si una gramática es **LL(1)**, entonces no presenta problemas de factorización por la izquierda



## Conjunto de Primeros

PRIMEROS : $(N \cup \Sigma)^* \rightarrow \wp(\Sigma \cup \{\epsilon\})$

```
input   α ∈ (N ∪ Σ)*  and  G = (N, Σ, P, S);
output  PRI(α);

PRI(α) = ∅;   i = 1;
if α == x ∈ Σ   then PRI(α) = {x}
if α == ε       then PRI(α) = {ε};
if α == B ∈ N   then
    for (B → β) ∈ P  do  PRI(α) = PRI(α) ∪ PRI(β);
if α == X1 X2 ... Xm  with  m > 1  then
    repeat
        PRI(α) = PRI(α) ∪ ( PRI (Xi) − {ε });     i = i + 1;
    until (ε ∉ PRI(Xi)  ∨  (i > m);
    if (i > m) ∧ (ε ∈ PRI(Xi))  then PRI(α) = PRI(α) ∪ {ε};
```



## Conjunto de Siguientes

SIGUIENTES : $N \rightarrow \wp(\Sigma \cup \{\$\})$

```
input   G = (N, Σ, P, S);
output  SIG(A);     ∀A ∈ N;

SIG(S) = {$};
for all A ∈ N : A ≠ S do  SIG(A) = ∅;
repeat
    for (A ∈ N) ∧ ∀(B → αAβ) ∈ P
        SIG(A) = SIG(A) ∪ ( PRI(β) − {ε} );
        if ε ∈ PRI(β) then SIG(A) = SIG(A) ∪ SIG(B);
until   no se modifique  SIG(A),     ∀A ∈ N
```

# Ejemplo de ASD - LL(1) 1/3

| | | |
|---|---|---|
| E | ::= | E + T |
| E | ::= | T |
| T | ::= | T ∗ F |
| T | ::= | F |
| F | ::= | ( E ) |
| F | ::= | a |

$\Rightarrow$

| | | | | | |
|---|---|---|---|---|---|
| E | ::= | T E' | → | PRI( T E' · SIG(E)) | ={ a, ( } |
| E' | ::= | + T E' | → | PRI( + T E' · SIG(E')) | ={ + } |
| E' | ::= | $\epsilon$ | → | PRI( SIG(E')) | ={ ), $ } |
| T | ::= | F T' | → | PRI( F T' · SIG(T)) | ={ a, ( } |
| T' | ::= | ∗F T' | → | PRI( ∗ F T' · SIG(T')) | ={ ∗ } |
| T' | ::= | $\epsilon$ | → | PRI( SIG(T')) | ={ +, ), $ } |
| F | ::= | ( E ) | → | PRI( ( E ) · SIG(F)) | ={ ( } |
| F | ::= | a | → | PRI( a · SIG(F)) | ={ a } |

SIG(E') = SIG(E) = { $, ) }; SIG(T') = SIG(T) = { +, $, ) }; SIG(F) = {∗, +, $, ) };



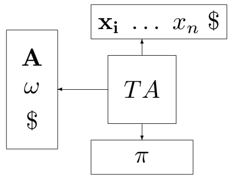
# Construcción de analizadores LL(1)

Dada una gramática $G = (N, \Sigma, \mathcal{P}, S)$ que cumple la condición LL(1):

$$S \stackrel{*}{\Rightarrow} x_1 \dots x_{i-1}\ A\ \omega\ \$ \qquad\qquad x_1 \dots x_{i-1}\ x_i\ \dots x_n\ \$$$

**si** $A \in N$ $(k : A \to \alpha) \in \mathcal{P}$ `derivar` **sii** $x_i \in$ PRI($\alpha\ \cdot$ SIG($A$))

**si** $A \in \Sigma$ `consumir` los símbolos (pila y cadena de entrada) **sii** $A = x_i$




**Configuración** $(x_i \dots x_n\ \$,\ A\ \omega\ \$,\ \pi) \vdash$

$(x_i \dots x_n\ \$,\ \alpha\ \omega\ \$,\ \pi \cdot k)$ **sii** $TA[A, x_i] = k : (A \to \alpha)$

ó

$(x_{+1} \dots x_n\ \$,\ \omega\ \$,\ \pi)$ **sii** $TA[A, x_i] =$ consumir



# Construcción de analizadores LL(1)

**Algorithm** Construcción de la TA-LL(1)

**input** $G = (N, \Sigma, P, S)$;

**output** $TA : ((N \cup \Sigma \cup \{\$\}) \times (\Sigma \cup \{\$\})) \to \{k : (A \to \alpha), consumir, aceptar, error\}$;

Inicializar $TA$ con la acción $error$;

**for** $(k : A \to \alpha) \in P$

  **for** $a \in$ PRI($\alpha\ \cdot$ SIG($A$)) **do** $TA[A, a] = k : (A \to \alpha)$;

**for** $a \in T$ **do** $TA[a, a] = consumir$;

$TA[\$, \$] = aceptar$;



# Ejemplo de ASD - LL(1) 2/3

| | | |
|---|---|---|
| E | ::= | E + T |
| E | ::= | T |
| T | ::= | T ∗ F |
| T | ::= | F |
| F | ::= | ( E ) |
| F | ::= | a |

$\Rightarrow$

| | | | | | |
|---|---|---|---|---|---|
| E | ::= | T E' | → | PRI( T E' · SIG(E)) | ={ a, ( } |
| E' | ::= | + T E' | → | PRI( + T E' · SIG(E')) | ={ + } |
| E' | ::= | $\epsilon$ | → | PRI( SIG(E')) | ={ ), $ } |
| T | ::= | F T' | → | PRI( F T' · SIG(T)) | ={ a, ( } |
| T' | ::= | ∗F T' | → | PRI( ∗ F T' · SIG(T')) | ={ ∗ } |
| T' | ::= | $\epsilon$ | → | PRI( SIG(T')) | ={ +, ), $ } |
| F | ::= | ( E ) | → | PRI( ( E ) · SIG(F)) | ={ ( } |
| F | ::= | a | → | PRI( a · SIG(F)) | ={ a } |

SIG(E') = SIG(E) = { $, ) }; SIG(T') = SIG(T) = { +, $, ) }; SIG(F) = {∗, +, $, ) };

| | a | + | ∗ | ( | ) | $ |
|---|---|---|---|---|---|---|
| E | (TE',1) | | | (TE',1) | | |
| E' | | (+TE',2) | | | ($\epsilon$,3) | ($\epsilon$,3) |
| T | (FT',4) | | | (FT',4) | | |
| T' | | ($\epsilon$,6) | (∗FT',5) | | ($\epsilon$,6) | ($\epsilon$,6) |
| F | (a,8) | | | ((E),7) | | |
| a | sacar | | | | | |
| + | | sacar | | | | |
| * | | | sacar | | | |
| ( | | | | sacar | | |
| ) | | | | | sacar | |
| $ | | | | | | aceptar |

## ASD: basado en la tabla LL(1)

**Algorithm** ASD-LL(1)

**input** $G = (N, \Sigma, P, S)$; $\omega \in \Sigma^*$

$TA : ((N \cup \Sigma \cup \{\$\}) \times (\Sigma \cup \{\$\})) \rightarrow \{k : (A \rightarrow \alpha), consumir, aceptar, error\}$;

**output** **if** $\omega \in L(G)$ **then** $\pi$ **else** $MenError(\cdot)$;

push($\$S$); $sym$ = getsym; $\pi = \epsilon$; $ok$ = FALSE;

**repeat**

**switch** $TA[top, sim]$ **do**

**case** $k : (A \rightarrow \alpha)$ : pop; push($\alpha$); $\pi = \pi \cdot k$;

**case** $consumir$ : pop; $sym$ = getsym;

**case** $aceptar$ : $ok$ = TRUE;

**case** $error$ : $ok$ = TRUE; $MenError(\cdot)$;

**until** $ok$



## Ejemplo de ASD - LL(1) 3/3

| | | |
|---|---|---|
| E | ::= | E + T |
| E | ::= | T |
| T | ::= | T $*$ F |
| T | ::= | F |
| F | ::= | ( E ) |
| F | ::= | a |

$\Rightarrow$

| | | | | | |
|---|---|---|---|---|---|
| E | ::= | T E' | $\rightarrow$ | PRI( T E' $\cdot$ SIG(E)) | $=\{$ a, ( $\}$ |
| E' | ::= | + T E' | $\rightarrow$ | PRI( + T E' $\cdot$ SIG(E')) | $=\{$ + $\}$ |
| E' | ::= | $\epsilon$ | $\rightarrow$ | PRI( SIG(E')) | $=\{$ ), \$ $\}$ |
| T | ::= | F T' | $\rightarrow$ | PRI( F T' $\cdot$ SIG(T)) | $=\{$ a, ( $\}$ |
| T' | ::= | $*$F T' | $\rightarrow$ | PRI( $*$ F T' $\cdot$ SIG(T')) | $=\{$ $*$ $\}$ |
| T' | ::= | $\epsilon$ | $\rightarrow$ | PRI( SIG(T')) | $=\{$ +, ), \$ $\}$ |
| F | ::= | ( E ) | $\rightarrow$ | PRI( ( E ) $\cdot$ SIG(F)) | $=\{$ ( $\}$ |
| F | ::= | a | $\rightarrow$ | PRI( a $\cdot$ SIG(F)) | $=\{$ a $\}$ |

SIG(E') = SIG(E) = { \$, ) }; SIG(T') = SIG(T) = { +, \$, ) }; SIG(F) = {$*$, +, \$, ) };

| | a | + | $*$ | ( | ) | \$ |
|---|---|---|---|---|---|---|
| E | (TE',1) | | | (TE',1) | | |
| E' | | (+TE',2) | | | ($\epsilon$,3) | ($\epsilon$,3) |
| T | (FT',4) | | | (FT',4) | | |
| T' | | ($\epsilon$,6) | ($*$FT',5) | | ($\epsilon$,6) | ($\epsilon$,6) |
| F | (a,8) | | | ((E),7) | | |
| a | sacar | | | | | |
| + | | sacar | | | | |
| $*$ | | | sacar | | | |
| ( | | | | sacar | | |
| ) | | | | | sacar | |
| \$ | | | | | | aceptar |

($a * a$\$, E\$, $\epsilon$) $\vdash$ ($a * a$\$, TE'\$, 1)
$\vdash$ ($a * a$\$, FT'E'\$, 14)
$\vdash$ ($a * a$\$, $a$T'E'\$, 148)
$\vdash$ ($*a$\$, T'E'\$, 148)
$\vdash$ ($*a$\$, $*$FT'E'\$, 1485)
$\vdash$ ($a$\$, FT'E'\$, 1485)
$\vdash$ ($a$\$, $a$T'E'\$, 14858)
$\vdash$ (\$, T'E'\$, 14858)
$\vdash$ (\$, E'\$, 148586)
$\vdash$ (\$, \$, 1485863)



## Modificación de gramáticas no LL(1)

➢ **Recursividad a izquierdas** (directa)

$A \rightarrow A\alpha_1 \mid \ldots \mid A\alpha_n \mid \beta_1 \mid \ldots \mid \beta_m$

⇨ Eliminación de la recursividad a izquierdas

$A \rightarrow \beta_1 A' \mid \ldots \mid \beta_m A'$

$A' \rightarrow \alpha_1 A' \mid \ldots \mid \alpha_n A' \mid \epsilon$

➢ **Factorización por la izquierda**

$A \rightarrow \alpha\beta_1 \mid \ldots \mid \alpha\beta_n \mid \gamma_1 \mid \ldots \mid \gamma_m$

⇨ Eliminación de la factorización por la izquierda

$A \rightarrow \alpha A' \mid \gamma_1 \mid \ldots \mid \gamma_m$

$A' \rightarrow \beta_1 \mid \ldots \mid \beta_n$