Análisis Sintáctico Descendente

➢ Primeros: $(N \cup T)^* \to \wp(T \cup \{\epsilon\})$

$\text{PRIMEROS}(\alpha) = \{a \in T \mid \alpha \overset{*}{\Rightarrow} a\beta\} \cup \{\epsilon \mid \alpha \overset{*}{\Rightarrow} \epsilon\}; \qquad \alpha, \beta \in (N \cup T)^*$

➢ Siguientes: $N \to \wp(T \cup \{\$\})$

$\text{SIGUIENTES}(A) = \{a \in T \mid S \overset{*}{\Rightarrow} \alpha A a \beta\} \cup \{\$ \mid S \overset{*}{\Rightarrow} \alpha A\};$

$\alpha, \beta \in (N \cup T)^*; A \in N$

ASD: Condición LL(1)

➢ Condición LL(1):

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

$$\text{PRIMEROS}(\alpha \cdot \text{SIGUIENTES}(A)) \cap \text{PRIMEROS}(\beta \cdot \text{SIGUIENTES}(A)) = \emptyset$$

➢ 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

Conjunto de Primeros

Función: PRIMEROS ($\alpha \in (N \cup T)^*$):conjunto de $(T \cup \{\epsilon\})$
Dada $G = (N, T, P, S)$, **con** $C = \emptyset$ conjunto de $(T \cup \{\epsilon\})$
Método
si $\alpha == x \in T$ **entonces** $C = \{x\}$;
si $\alpha == \epsilon$ **entonces** $C = \{\epsilon\}$;
si $\alpha == B \in N$ **entonces**
para toda $(B \to \beta) \in P$ **hacer** $C = C \cup \text{PRIMEROS}(\beta)$;
si $\alpha == X_1 X_2 \ldots X_m$ **con** $m > 1$ **entonces**
$i = 0$;
repetir
$i = i + 1; \qquad C = C \cup (\text{PRIMEROS}(X_i) - \{\epsilon\})$;
hasta $(\epsilon \notin \text{PRIMEROS}(X_i) \vee (i = m)$;
si $(i == m) \wedge (\epsilon \in \text{PRIMEROS}(X_i))$ **entonces** $C = C \cup \{\epsilon\}$;
Devolver C;
Fin

Conjunto de Siguientes

Función: SIGUIENTES: vector $[N]$ de conjuntos de $(T \cup \{\$\})$
Dada $G = (N, T, P, S)$, **con** C: vector $[N]$ de conjuntos de $(T \cup \{\$\})$
Método
para todo $A \in N$ **hacer** $C[A] = \emptyset$;
$C[S] = \{\$\}$;
mientras $C[A]$ no se modifique para ningún $A \in N$ **hacer**
para todo $A \in N$ **y para toda** $(B \to \alpha A \beta) \in P$ **hacer**
$C[A] = C[A] \cup (\text{PRIMEROS}(\beta) - \{\epsilon\})$;
si $\epsilon \in \text{PRIMEROS}(\beta)$ **entonces** $C[A] = C[A] \cup C[B]$;
Devolver C;
Fin

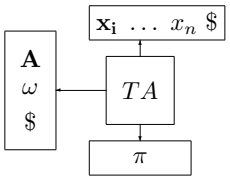
Construcción de analizadores LL(1)

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

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

si $A \in N\ (k : A \to \alpha) \in \mathcal{P}$ **derivar**$-k$ sii $x_i \in$ PRIMEROS ($\alpha\cdot$ SIGUIENTE(A))

si $A \in T$ **consumir** 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 $TA[x_i, A] =$ `derivar`$-k$

o

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

Construcción de analizadores LL(1)

Algoritmo: Construcción de la TA-LL(1)

Entrada $G = (N, T, P, S)$;

Salida TA: $((N \cup T \cup \{\$\}) \times (T \cup \{\$\})) \to$ {derivar-k, sacar, aceptar, error};

Método

Inicializar la TA con la acción *error*;

para todo $(k : A \to \alpha) \in P$ **hacer**

  para todo $a \in$PRIMEROS$(\alpha\cdot$ SIGUIENTES$(A))$ **hacer** TA [A,a] = derivar-k;

para todo $a \in T$ **hacer** TA [a,a] = sacar;

TA [$,$] = aceptar;

Fin

ASD: basado en la tabla LL(1)

Algoritmo: ASD-LL(1)

Entrada $G = (N, T, P, S)$; $\omega \in T^*$;

TA: $((N \cup T \cup \{\$\}) \times (T \cup \{\$\})) \to$ {derivar-k, sacar, aceptar, error};

Salida **si** $\omega \in L(G)$ **entonces** π **else** *MenError*();

Método

apilar($\$S$); sim = *ObtSim*; $\pi = \epsilon$; fin = *falso*;

repetir

  caso de que TA [cima, sim] **sea**

  derivar-k $(A \to \alpha)$: *desapilar*; *apilar*(α); $\pi = \pi \cdot k$;

  sacar: *desapilar*; sim = *ObtSim*;

  aceptar: fin = verdad;

  error: fin = verdad; *MenError*();

hasta fin

Fin

Ejemplo de ASD - LL(1)

E	::=	E + T	E	::=	T E'	→ PRIMEROS(T E' ·SIGUIENTES(E)) ={ a, (}
E	::=	T	E'	::=	+ T E'	→ PRIMEROS(+ T E' ·SIGUIENTES(E')) ={ + }
T	::=	T * F ⇒	E'	::=	ϵ	→ PRIMEROS(SIGUIENTES(E')) ={), $ }
T	::=	F	T	::=	F T'	→ PRIMEROS(F T' ·SIGUIENTES(T)) ={ a, (}
F	::=	(E)	T'	::=	*F T'	→ PRIMEROS(* F T' ·SIGUIENTES(T')) ={ * }
F	::=	a	T'	::=	ϵ	→ PRIMEROS(SIGUIENTES(T')) ={ +,), $ }
			F	::=	(E)	→ PRIMEROS((E) ·SIGUIENTES(F)) ={ (}
			F	::=	a	→ PRIMEROS(a ·SIGUIENTES(F)) ={ a }

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

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

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

Análisis Sintáctico Descendente Recursivo

$S \rightarrow$ **begin** L S **end** | ϵ

$L \rightarrow$ **id** L | ϵ

```
procedimiento sacar (x);
  si (sim == x ) ObtSim;
  sino MenError()
fin procedimiento
```

```
procedimiento S;
   caso_de_que sim sea:
   begin: sacar (begin); L(); S(); sacar (end);
   $, end:
   en_cualquier_otro_caso: MenError()
fin procedimiento
```

```
procedimiento L;
   caso_de_que sim sea:
   id: sacar(id); L();
   begin, end:
   en_cualquier_otro_caso: MenError()
fin procedimiento
```

PRIMEROS (**begin** L S **end** · SIGUIENTES (S)) = { **begin** }
PRIMEROS (SIGUIENTES (S)) = {$, **end** }
PRIMEROS (**id** L · SIGUIENTES (L)) = { **id** }
PRIMEROS (SIGUIENTES (L)) = { **begin**, **end** }

Modificación de gramáticas no LL(1)

➢ Recursividad a izquierdas

$A \rightarrow A\alpha_1 \mid A\alpha_2 \mid \ldots \mid A\alpha_n \mid \beta_1 \mid \beta_2 \mid \ldots \mid \beta_m$

⇨ Eliminación de la recursividad a izquierdas

$A \rightarrow \beta_1 A' \mid \beta_2 A' \mid \ldots \mid \beta_m A'$

$A' \rightarrow \alpha_1 A' \mid \alpha_2 A' \mid \ldots \mid \alpha_n A' \mid \epsilon$

➢ Factorización por la izquierda

$A \rightarrow \alpha\beta_1 \mid \alpha\beta_2 \mid \ldots \mid \alpha_n\beta_m \mid \gamma$

⇨ Eliminación de la factorización por la izquierda

$A \rightarrow \alpha A' \mid \gamma$

$A' \rightarrow \beta_1 \mid \beta_2 \mid \ldots \mid \beta_m$