Teoría de Lenguajes

Gramáticas Libres de Contexto

Lenguaje

Sistema de comunicación estructurado para el que existe un contexto de uso y ciertos principios combinatorios formales (wikipedia)

- Alfabeto
- Reglas

Reglas de producción

V Nom

Art Nom

Los perros comen huesos

SN

SV

O → SN SV
SN → Art Nom
SV → V Nom
Art → Los
Nom → perros | huesos
V → comen

Reglas de producción

```
procedure pepe (j:integer);
var  a:string;
begin
        if j > 100 then
             a:= “Ganaste”;
        else
             a:= “Lo siento, perdiste”
        endif;
        write (a);
  end;
```

<procedure declaration> ::= <procedure heading> <block>

<procedure heading> ::= ***procedure*** <identifier> ;

| ***procedure*** <identifier> (<formal parameter section> {;<formal parameter section>});

<formal parameter section> ::= <parameter group> | ***var*** <parameter group> | Ɛ

…

<block> ::= <label declaration part> <constant definition part> <type definition part> <variable declaration part> <procedure and function declaration part> <statement part>

….

<if statement> ::= ***if*** <expression> ***then*** <statement> | ***if*** <expression> ***then*** <statement> ***else*** <statement>

….

Lenguaje

Un lenguaje es el conjunto de strings sobre elementos de un alfabeto

Sea Σ un alfabeto

Definimos Σ^*

$\varepsilon \in \Sigma^*$

si $a \in \Sigma$, $w \in \Sigma^*$ entonces $wa \in \Sigma^*$

Trabajamos con $L \subseteq \Sigma^*$

Ejemplos

1) $L = \{ 0^k1^k \ / \ k > 0 \}$ $(\Sigma = \{0,1\})$

   $01 \in L$

   si $w \in L$ entonces $0w1 \in L$

2) $L'' = \{2^t \ / \ t \geq 0 \}$ $(\Sigma = \{2\})$

   $\varepsilon \in L''$

   si $w \in L''$ entonces $w2 \in L''$

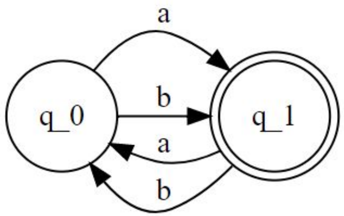
3) $L' = \{ 0^k1^k2^t \ / \ k > 0, t \geq 0 \}$ $(\Sigma = \{0,1,2\})$

   si $w \in L$ y $x \in L''$ entonces $wx \in L'$

vamos a estudiar en la teoría de lenguajes

Formalismos

- autómatas



- gramáticas

S → a
S → b
S → aA
S → bA
A → aS
A → bS

Gramáticas Libres de Contexto

$G = (V,T,P,S)$

Es un formalismo para especificar lenguajes

- V : conjunto finito de variables
- T : conjunto finito de terminales ($T \equiv \Sigma$)
- P : conjunto de reglas de producción
- S : símbolo inicial $S \in V$

$$A \rightarrow \alpha \ / \ \alpha \in (V \cup T)^*$$

$$A \in V$$

Ejemplos

1) $L = \{ 0^k 1^k \ / \ k > 0 \}$ $(\Sigma = \{0,1\})$

   $01 \in L$

   si $w \in L$ entonces $0w1 \in L$

$A \rightarrow 01$
$A \rightarrow 0A1$

2) $L'' = \{2^t \ / \ t \geq 0 \}$ $(\Sigma = \{2\})$

   $\varepsilon \in L''$

   si $w \in L''$ entonces $w2 \in L''$

$B \rightarrow \varepsilon$
$B \rightarrow B2$

3) $L' = \{ 0^k 1^k 2^t \ / \ k > 0, t \geq 0 \}$ $(\Sigma = \{0,1,2\})$

   si $w \in L$ y $x \in L''$ entonces $wx \in L'$

$S \rightarrow AB$

Ejemplo

$L' = \{ 0^k 1^k 2^t \ / \ k > 0, t \geq 0 \}$

$G = (V,T,P,S)$ $V = \{S,A,B\}$ $T = \{ 0,1,2\}$

$P = \{$

$S \rightarrow AB$

$A \rightarrow 01$

$A \rightarrow 0A1$

$B \rightarrow \varepsilon$

$B \rightarrow B2$

$\}$

Definiciones

Derivación directa:

Dada una $G = (V,T,P,S)$ y $A \rightarrow \alpha \in P$

$$\beta A \gamma \Rightarrow \beta \alpha \gamma \quad \beta, \gamma \in (V \cup T)^*$$

Derivación:

$$\alpha_1 \overset{*}{\Rightarrow} \alpha_k \;/\; \alpha_1 \Rightarrow \alpha_2 \Rightarrow \ldots \Rightarrow \alpha_k$$

en cada paso se aplica una derivación directa

Definiciones

Lenguaje generado por una gramática $G:(V,T,P,S)$

$$L(G) = \{ x \in T^* \ / \ S \overset{*}{\Rightarrow} x \}$$

__Lenguaje Libre de Contexto__

Es aquél que puede ser generado por una gramática libre de contexto

$$\mathcal{L} = L(G)$$

Definiciones

Ejemplo: Sea $G : (V,T,P,S)$

$V = \{ S,X \}$ $T=\{ a,b,c,+,-,*,/\}$ $P=\{ S \rightarrow S + S$

$\mid S - S$

$\mid S * S$

$\mid S / S$

$\mid X$

$X \rightarrow a \mid b \mid c \mid aX \mid bX \mid cX \}$

$a + b * c \in L(G)$?

Definiciones

Ejemplo: Sea G : (V,T,P,S)

$V = \{ S,X \}$ $T=\{ a,b,c,+,-,*,/\}$ $P=\{ S \rightarrow S + S \mid S - S \mid S * S \mid S / S \mid X$
$X \rightarrow a \mid b \mid c \mid aX \mid bX \mid cX \}$

$a + b * c \in L(G)$?

Vemos si se puede derivar

$$S \Rightarrow S + S \Rightarrow S + S * S \Rightarrow X + S * S \Rightarrow a + S * S \Rightarrow a + S * X \Rightarrow a + X * X \Rightarrow$$
$$\Rightarrow a + b * X \Rightarrow a + b * c \quad \surd$$

Definiciones

Derivación de más a la derecha:

es aquella en que en cada derivación, se elige la variable de **más a la derecha**

Derivación de más a la izquierda:

es aquella en que en cada derivación, se elige la variable de **más a la izquierda**

Definiciones

Derivación de más a la derecha:

$S \Rightarrow S + \mathbf{S} \Rightarrow S + S * \mathbf{S} \Rightarrow S + S * \mathbf{X} \Rightarrow S + \mathbf{S} * c \Rightarrow S + \mathbf{X} * c \Rightarrow \mathbf{S} + b * c \Rightarrow$

$\Rightarrow \mathbf{X} + b * c \Rightarrow a + b * c$ √

Derivación de más a la izquierda:

$S \Rightarrow \mathbf{S} + S \Rightarrow \mathbf{X} + S \Rightarrow a + \mathbf{S} \Rightarrow a + \mathbf{S} * S \Rightarrow a + X * \mathbf{S} \Rightarrow a + b * \mathbf{S} \Rightarrow$

$\Rightarrow a + b * \mathbf{X} \Rightarrow a + b * c$ √

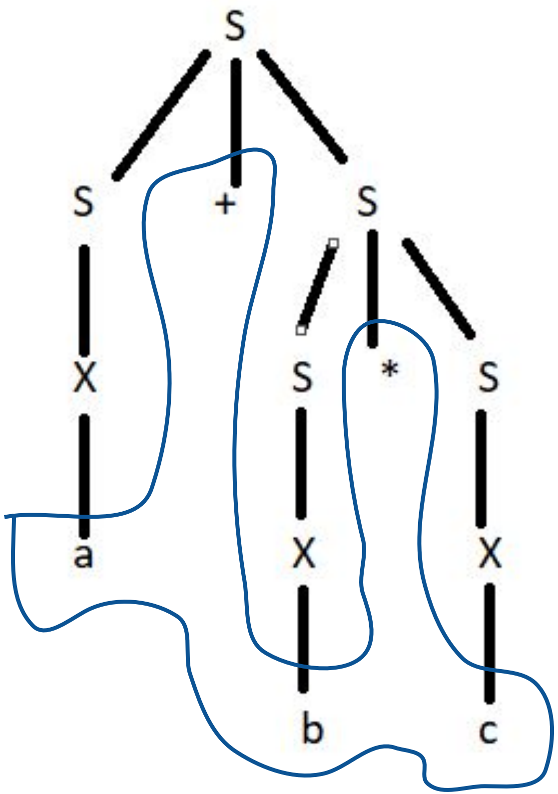
Definiciones

Dada una G:(V,T,P,S), se define árbol de derivación para una tira $w \in L(G)$ a una estructura que:

- la raíz es S
- cada nodo interno $\in$ V
- cada hoja $\in T \cup \{\varepsilon\}$
- si un nodo está etiquetado con A y sus hijos son $x_1, x_2, .., .x_k$ entonces existe $A \rightarrow x_1x_2...x_k \in P$

Ejemplo

a + b * c



Definiciones

Teorema

Dada una G:(V,T,P,S) y una w ∈ L(G)

$$S \overset{*}{\Rightarrow} w \quad \Leftrightarrow \quad \text{existe un árbol de derivación para } w$$

Definiciones

<u>Gramática ambigua</u> G:(V,T,P,S) si existe alguna $w \in L(G)$ con al menos:

- 2 derivaciones de más a la izquierda distintas o
- 2 derivaciones de más a la derecha distintas o
- 2 árboles de derivación distintos

Ejemplo de ambigüedad

Estructura de control ::= **if** <Condicion> **then** <Sentencia> **else** <Sentencia>

Estructura de control ::= **if** <Condicion> **then** <Sentencia>

Sentencia ::= <Asignacion> | <Sentencia compuesta> | <Estructura de control> ...

Sentencia compuesta ::= **begin** <Sentencia> {; <Sentencia> } **end**

if c1 then if c2 then S1 else S2

Ejemplo de ambigüedad (cont.)

if c1 then if c2 then S1 else S2