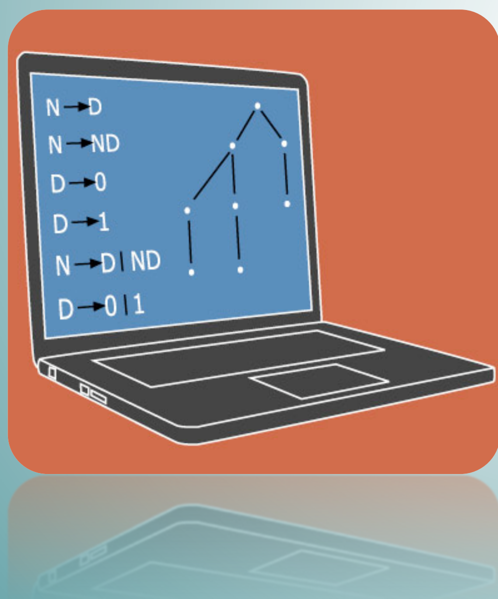
Procesadores de Lenguaje

Repaso TALF



Cristina Tirnauca

Dpto. de Matemáticas,

Estadística y Computación

La Jerarquía de Chomsky

Cuatro niveles de lenguajes formales

Del nivel más restrictivo al más general:

Tipo 3: Lenguajes regulares
(gramáticas regulares, expresiones regulares, autómatas finitos deterministas o indeterministas, ...);

Tipo 2: Lenguajes libres de contexto (*context-free* o incontextuales)
(gramáticas incontextuales, autómatas con pila, ...);

Tipo 1: Lenguajes sensibles al contexto (*context-sensitive*)
(gramáticas sensibles al contexto, autómatas linealmente acotados);

Tipo 0: Lenguajes recursivamente enumerables
(gramáticas arbitrarias, máquinas de Turing...).

Gramáticas incontextuales

¿De qué constan?

Una gramática $G = (V, \Sigma, S, P)$ está dada por:

1. Dos alfabetos
   - el de símbolos gramaticales V (símbolos no terminales o variables), y
   - el alfabeto de entrada Σ (o de símbolos terminales)
2. un símbolo gramatical S especialmente designado como inicial,
3. y un conjunto de reglas gramaticales (o "producciones")

   (donde cada regla consta de dos palabras sobre la unión de ambos alfabetos $V \cup \Sigma$, y la primera de las dos palabras no es vacía).

En las gramáticas incontextuales, que serán las únicas que emplearemos en esta asignatura, la primera secuencia de cada regla consta de exactamente un símbolo gramatical.

Ejemplos de gramáticas incontextuales

Y dos notaciones

Lenguaje: los números en binario

Gramática $G = (V, \Sigma, N, P)$ con $V = \{N, D\}$ y $\Sigma = \{0, 1\}$.

Notación BNF (Backus Naur Form): **Notación TALF**:

- Los caracteres del alfabeto de entrada entresin comillas simples..
- Los símbolos gramaticales muchas veces con una sola letra mayúscula; el primero en aparecer es el símbolo inicial.

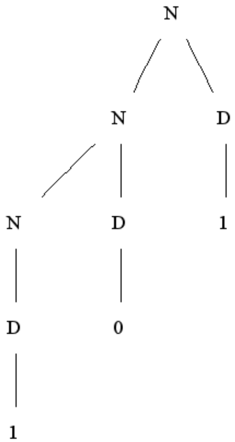
Las reglas en notación BNF	**Las reglas en notación TALF**
N : D	N $\rightarrow$ D
N : N D	N $\rightarrow$ N D
D : '0'	D $\rightarrow$ 0
D : '1'	D $\rightarrow$ 1
N : D | N D	N $\rightarrow$ D | N D
D : '0' | '1'	D $\rightarrow$ 0 | 1

Árboles de análisis sintáctico

Y derivaciones izquierda y derecha

Un árbol y tres derivaciones:

N	N	N
N D	N D	N D
N 1	N D D	N 1
N D 1	D D D	N D 1
D D 1	1 D D	N 0 1
1 D 1	1 0 D	D 0 1
1 0 1	1 0 1	1 0 1
	(izquierda)	(derecha)



Gramática para expresiones

Versión simplificada

Algunas operaciones entre enteros sin signo

E : N | E OP E

N : D | N D

D : '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'

OP : '+' | '*'

Observamos la propiedad de recursividad izquierda.

Ambigüedad:

Las expresiones con más de un operador admiten varios árboles de análisis sintáctico: 12+34*56

Ambigüedad

E indeterminismo

Una gramática libre de contexto es ambigua si existe una palabra que tiene al menos dos árboles de derivación.
NOTA: algunos lenguajes de programación tienen grámaticas ambiguas (solución: utilizar tabla de símbolos).
NOTA: en general, no se puede decidir si una gramática libre de contexto es o no ambigua.

Un lenguaje libre de contexto es inherentemente ambiguo si todas las gramáticas libres de contexto que generan el lenguaje son ambiguas.
Ejemplo de lenguaje inherentemente ambiguo:
$\{a^n b^m c^m d^n \mid n, m > 0\} \cup \{a^n b^n c^m d^m \mid n, m > 0\}$.

El conjunto de los lenguajes libres de contexto deterministas es un subconjunto propio del conjunto de los lenguajes libres de contexto que no son inherentemente ambiguo.
Ejemplo de lenguaje indeterminista: el lenguaje generado por la gramática $S \to 0S0 \mid 1S1 \mid \lambda$.

Expresiones regulares

Y los lenguajes correspondientes

Σ: alfabeto finito.

- $\emptyset$ es una expresión regular $\qquad L(\emptyset) = \emptyset$
- λ es una expresión regular $\qquad L(\lambda) = \lambda$
- a es una expresión regular, $\forall a \in \Sigma$ $\qquad L(a) = \{a\}$
- si α y β son expresiones regulares, también lo son:
  - $\alpha + \beta$ $\qquad L(\alpha + \beta) = L(\alpha) \cup L(\beta)$
  - $\alpha \cdot \beta$ $\qquad L(\alpha \cdot \beta) = L(\alpha) \cdot L(\beta)$
  - α^* $\qquad L(\alpha^*) = L(\alpha)^*$

Autómatas finitos

Y los lenguajes aceptados

Un autómata finito $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$
Caso general (sin λ-transiciones):
$\delta : Q \times \Sigma \to 2^Q$
Deterministas:
$\delta : Q \times \Sigma \to Q$
$L(\mathcal{A}) = \{w \mid \delta(q_0, w) \cap F \neq \emptyset\}$

Autómatas con pila

Aceptar por pila vacía o estado final

Un autómata con pila $\mathcal{P} = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$.

Caso general:
$\delta : Q \times \Sigma \cup \{\lambda\} \times \Gamma \rightarrow 2^{Q \times \Gamma^*}$

Deterministas:
1. $|\delta(q, a, X))| \leq 1 \forall q \in Q, a \in \Sigma \cup \{\lambda\}, X \in \Gamma$
2. si $|\delta(q, a, X))| = 1$ para $a \in \Sigma$ entonces $|\delta(q, \lambda, X))| = 0$

Estado final: $L(\mathcal{P}) = \{w \mid (q_0, w, Z_0) \Rightarrow^* (q, \lambda, \gamma), q \in F\}$
Cinta y pila vacía: $N(\mathcal{P}) = \{w \mid (q_0, w, Z_0) \Rightarrow^* (q, \lambda, \lambda)\}$