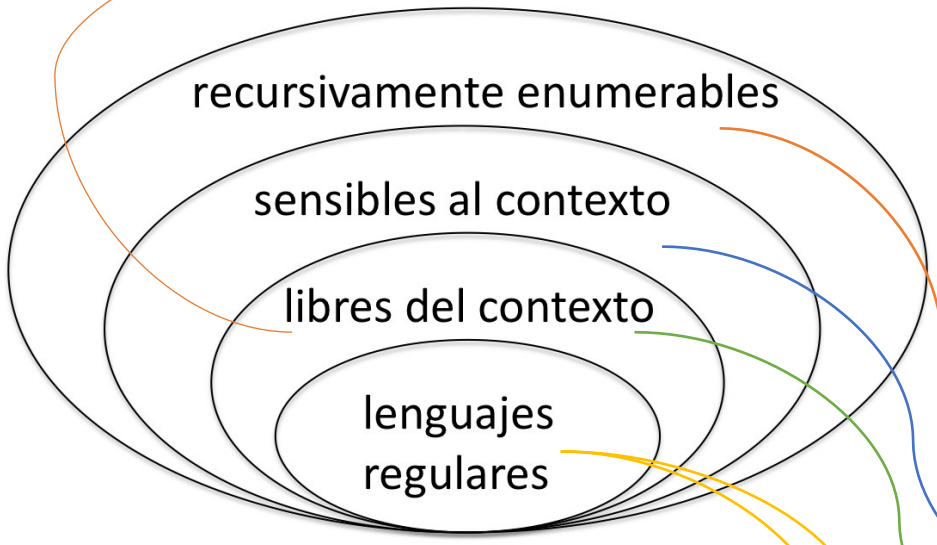
Gramáticas libres de contexto

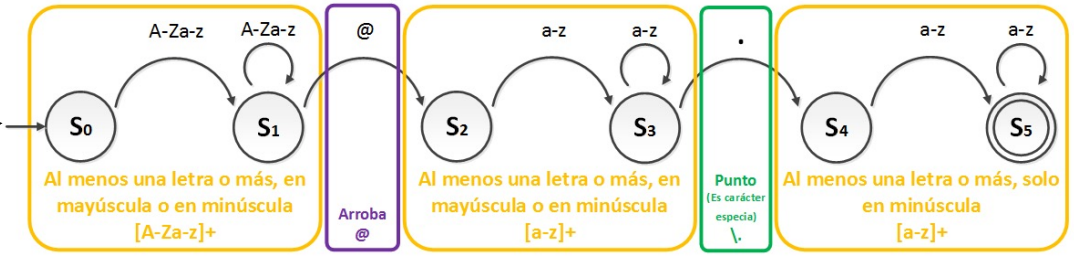
- Prof. Maureen Murillo
- Teoría de la Computación
- Escuela de Ciencias de la Computación e Informática
- Universidad de Costa Rica

Basado en capítulo 5 de libro de Hopcroft y otros

Ubicándonos…

¿Para qué? ¿Cuándo son necesarios?





[A-Za-z]+@[a-z]+\.[a-z]+

Máquina de Turing

Máq.Turing c/memoria limitada

Autómata de pila

Autómata finito ✅

Lenguaje regular ✅

¿Cuándo nos sirve una gramática libre de contexto?

```
<data>
  <topics>
    <topic>Afganistán</topic>
          ….
    <topic>salud</topic>
  </topics>

  <sites>
    <site>
          ….
    </site>
  </sites>
</data>
```

```
while (…)  {
   if (…)  {
      while (…) {
           …
      }
   }
   while (…) {
        …
   }
}
```

¿Cómo saber que cada símbolo de apertura tiene uno de cierre? 🤔

¿Se puede escribir una expresión regular para esto? 🤦‍♀️

Contando los símbolos… tal vez… ¿Es suficiente contar? ⁉️

Necesitamos **memoria**… algo que recuerde ❇️
Necesitamos una forma de especificar una **estructura**, una **sintaxis**.

Ejemplo de sintaxis de oraciones en español:
sujeto + verbo + complemento directo + complemento indirecto

¿Qué es una gramática libre de contexto (CFG)?

Una CFG es una gramática formal en la que cada regla de producción es de la forma: V → w
Es una notación recursiva.

Componentes

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

- T: Símbolos terminales (alfabeto): elementos que no se dividen (tokens en el análisis sintáctico)
- V: Símbolos no terminales (variables): elementos que se componen de partes
- S: Símbolo inicial: es uno de los símbolos no terminales en donde inicia el análisis
- P: Producciones o reglas: relación entre un símbolo no terminal y una secuencia de símbolos terminales y/o no terminales. Es la definición recursiva del lenguaje. Formato:

Símbolo no terminal → lista de símbolos terminales y/o no terminales

cabeza — cuerpo

Ejemplos

Dos gramáticas que generan números binarios:

S –> 0	S –> 0S
S –> 1	S –> 1S
S –> 0S	S –> ε
S –> 1S	

¿Cuál es la diferencia en cuanto a las hileras binarias que reconocen ambas gramáticas? 🤔

Gramática que genera números binarios pares:

S –> B0
B –> 0B
B –> 1B
B –> ε

Ejemplo de cadena que reconoce:

10110

¿Qué pasa si no tuviera la última producción?

Escriba una gramática que reconozca palíndromos binarios.

P –> ε
P –> 0
P –> 1
P –> 0P0
P –> 1P1

Otra manera de especificar las producciones anteriores:

P –> ε | 0 | 1 | 0P0 | 1P1

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

$G_{bin} = (\{S\}, \{0,1\}, A, S)$
$G_{binPares} = (\{S, B\}, \{0,1\}, A, S)$
$G_{pal} = (\{P\}, \{0,1\}, A, P)$

donde A es el conjunto de producciones mostrado en cada ejemplo.

Herramienta en línea: https://web.stanford.edu/class/archive/cs/cs103/cs103.1156/tools/cfg/

Árboles de derivación (parse trees)

Derivación es el proceso de aplicar las producciones de una gramática para determinar que cierta cadena pertenece al lenguaje de una variable, desde la cabeza hasta el cuerpo.

Los **árboles de derivación** representan la estructura que aplica una gramática a las cadenas de su lenguaje mediante el proceso de derivación. Es la forma en la que se suele representar la estructura de los programas.

G_{pal} = ({*P*}, {0,1}, *A*, *P*), donde A es:

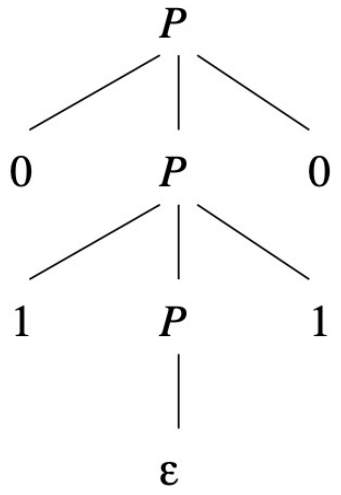
P –> ε

P –> 0

P –> 1

P –> 0P0

P –> 1P1



Cada ramificación del árbol corresponde a una producción

Figura 5.5. Un árbol de derivación para la derivación $P \overset{*}{\Rightarrow} 0110$.

Tipos de derivación

Ejemplo: gramática que acepta oraciones en español con la palabras: *niña, bola, la, hermosa, fuerte, golpea, recoge.*

ORACION –> S P
S –> OBJ
P –> V CD
CD –> OBJ
OBJ –> SUST | ART OBJ | OBJ ADJ
ART –> la
ADJ –> hermosa | fuerte
V –> golpea | recoge
SUST –> niña | bola

¿Oración correcta?
La niña hermosa golpea la bola fuerte

Izquierda: en cada paso, el símbolo no terminal más a la izquierda es el que se sustituye.

ORACION –> S P –> OBJ P –> ART OBJ P –> la OBJ P –> la OBJ ADJ P –>
la niña ADJ P –> la niña hermosa P –> la niña hermosa V CD –>
la niña hermosa golpea CD –> la niña hermosa golpea OBJ –>
la niña hermosa golpea ART OBJ –> la niña hermosa golpea la OBJ –>
la niña hermosa golpea la OBJ ADJ –>
la niña hermosa golpea la SUST ADJ –>
la niña hermosa golpea la bola ADJ –>
la niña hermosa golpea la bola fuerte ✅

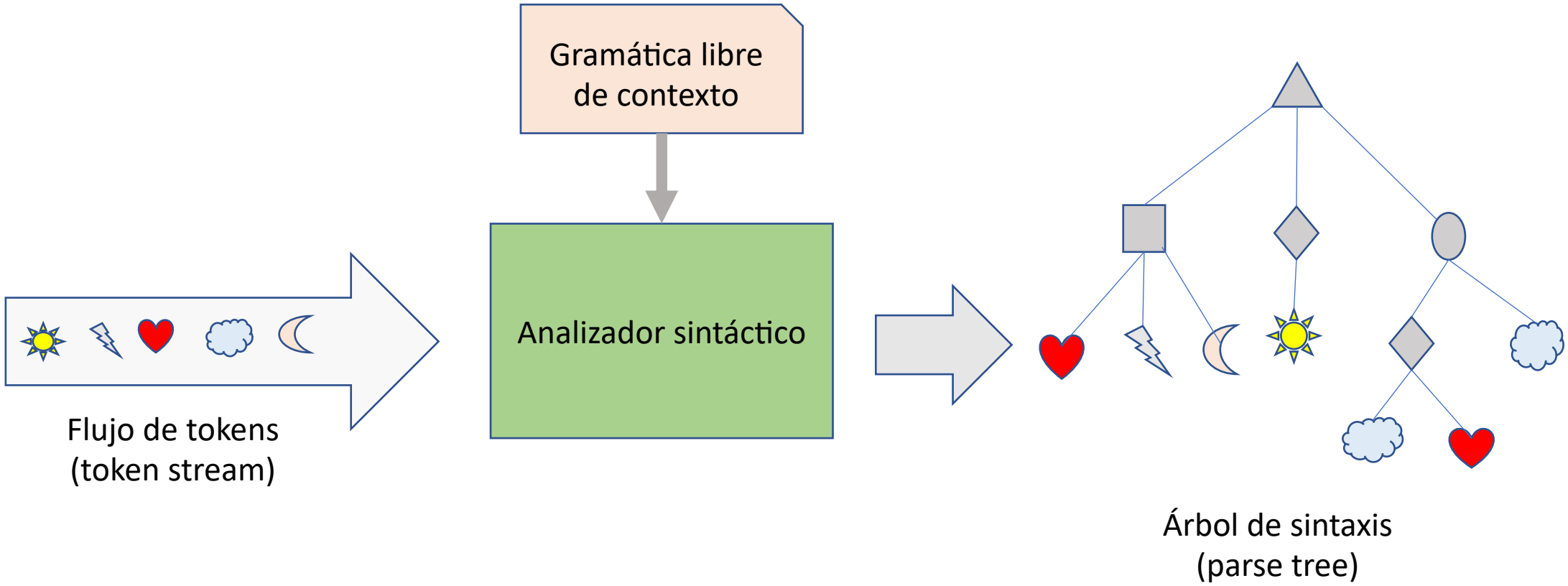
Derecha: en cada paso, el símbolo no terminal más a la derecha es el que se sustituye.

ORACION –> S P –> S V CD –> S V OBJ –> S V OBJ ADJ –>
S V OBJ fuerte –> S V ART OBJ –> S V ART bola fuerte –>
S V la bola fuerte –> S golpea la bola fuerte –> OBJ golpea la bola fuerte
–> ART OBJ golpea la bola fuerte –> ART OBJ ADJ golpea la bola fuerte –>
ART OBJ hermosa golpea la bola fuerte –>
ART SUST hermosa golpea la bola fuerte –>
ART niña hermosa golpea la bola fuerte –>
la niña hermosa golpea la bola fuerte ✅

¿Para qué nos sirven las CFG?

- Análisis sintáctico para compiladores
- Describir formatos de archivos XML, para intercambio de información en Internet
- Describir otros formatos

Análisis sintáctico



¡Naruto perdió los derechos de imagen!