

# Lenguajes Formales y de Programación. B-

Tutor: Douglas Omar Arreola Martínez

Clase 8

# Aplicación del nuevo reglamento de evaluación



Actividad	% sobre 20 pts	Valor de 20 pts	Valor de 24 pts	% sobre 24 pts	Suma
Tarea 1	1	0,2	0,2	0,8333333333	3,3333333333
Tarea 2	1	0,2	0,2	0,8333333333	
Tarea 3	1	0,2	0,2	0,8333333333	
Tarea 4	1	0,2	0,2	0,8333333333	
Corto 1	2	0,4	0,4	1,666666667	5
Corto 2	2	0,4	0,4	1,666666667	
Corto 3	2	0,4	0,4	1,666666667	
HT 1	2,5	0,5	0,5	2,083333333	4,166666667
HT 2	2,5	0,5	0,5	2,083333333	
Práctica	15	3	3	12,5	12,5
Proyecto 1	25	5	7	29,16666667	29,16666667
Proyecto 2	35	7	9	37,5	37,5
Examen Final	10	2	2	8,333333333	8,333333333
<b>Total</b>	<b>100</b>	<b>20</b>	<b>24</b>	<b>100</b>	<b>100</b>

# Examen Corto No. 2

**UEDI**

**Día: Sábado 12 de marzo**

**Hora: 21:00 - 21:30**

# Hoja de Trabajo



# HT #1



Analizar el lenguaje propuesto para el proyecto 1 y mediante una tabla de cuatro columnas describir el nombre del token, el patrón que lo define, la respectiva expresión regular y en la última columna anotar mínimo dos ejemplos de los lexemas que pueden ser reconocidos con dichos patrones.

Luego de definir todas las expresiones regulares agruparlas en una sola expresión final mediante el operador **Or** ( `|` ) y a partir de allí realizar todo el proceso del método del árbol.

# HT #1



Entregar un documento con extensión **.pdf** en el apartado disponible en UEDI **y también** subir dicho documento al repositorio utilizado para el desarrollo del proyecto 1.

Fecha de Entrega: **Domingo 13** de marzo antes de las **23:59 horas**.

## Elementos:

L = [A-Z, a-z, Ñ, ñ]

D = [0-9]

Nombre	Patrón	Expresión Regular	Ejemplos
Identificador o ID	Inicia con una letra a la que le sigue cero o muchas veces cualquier combinación de más letras, dígitos o sub guiones	$L (L \mid D \mid \_ )^*$	<ul style="list-style-type: none"><li>- a</li><li>- b1</li><li>- mi_variable</li></ul>
Entero	Número compuesto como mínimo por un solo dígito al cual le pueden, o no, seguir más dígitos.	$D D^*$	<ul style="list-style-type: none"><li>- 0</li><li>- 10</li><li>- 1050</li></ul>
-	-	-	-



# Expresiones Regulares



# Operadores de Lenguajes

Operación	Símbolo	Descripción	Presedencia
Cerradura de Kleene	*	Cero o más veces	1o
Cerradura positiva	+	Una o más veces	
Cero o una instancia	?	Cero o una vez	
Concatenación	▪	Uno y lo otro (ambas opciones)	2o
Unión o Alternación		Uno o lo otro (solo una de las opciones)	3o

# Método del Árbol



# Pasos

1. Concatenar símbolo de aceptación al final de la ER.
2. Construir el árbol binario de sintaxis.
3. Identificar cada hoja con terminales.
4. Calcular por cada nodo del árbol:
  - 4.1. Anulable
  - 4.2. Primera Posición (First)
  - 4.3. Última Posición (Last)
5. Calcular Siguientes
6. Construir Tabla de Transiciones
  - 6.1. Optimizar Tabla
7. Dibujar el Diagrama de Estados Finitos (Autómata)

# Ejemplo:

$$(L|D)^*L?|\`#\`L$$

$L = [a-zA-ZñÑ]$

$D = [0-9]$

# 1. Agregar Símbolo de Aceptación

$(L|D)^*L?'#L$

# 1. Agregar Símbolo de Aceptación

$(L|D)^*L?|'\# 'L$

$\rightarrow ( (L|D)^*L?|'\# 'L )\$$

## 1. Agregar Símbolo de Aceptación

$(L|D)^*L?|'\# 'L$

$\rightarrow ( (L|D)^*L?|'\# 'L )\$$

---

## 2. Construir Árbol Binario de Sintaxis

$( (L|D)^*L?|'\# 'L )\$$



## 1. Agregar Símbolo de Aceptación

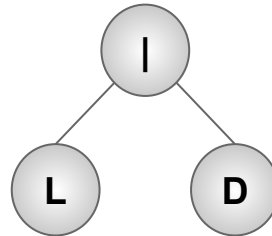
$(L|D)^*L?'#\text{'L}$

$\rightarrow ( (L|D)^*L?'#\text{'L} )\$$

---

## 2. Construir Árbol Binario de Sintaxis

$( (L|D)^*L?'#\text{'L} )\$$



## 1. Agregar Símbolo de Aceptación

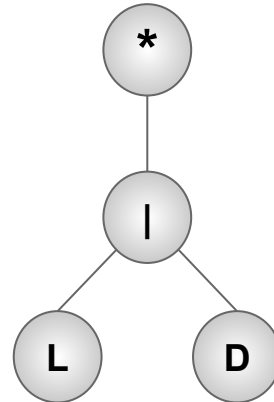
$(L|D)^*L?'#\text{'L}$

$\rightarrow ( (L|D)^*L?'#\text{'L} )\$$

---

## 2. Construir Árbol Binario de Sintaxis

$( (L|D)^*L?'#\text{'L} )\$$



## 1. Agregar Símbolo de Aceptación

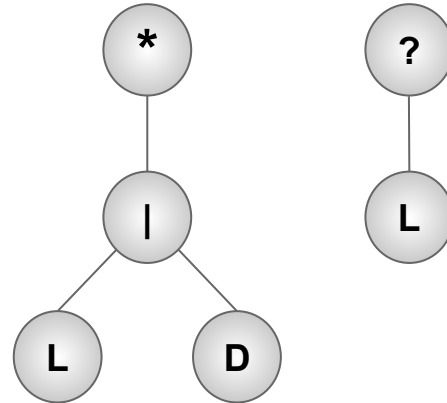
$(L|D)^*L?' \# 'L$

$\rightarrow ( (L|D)^*L?' \# 'L ) \$$

---

## 2. Construir Árbol Binario de Sintaxis

$( (L|D)^*L?' \# 'L ) \$$



## 1. Agregar Símbolo de Aceptación

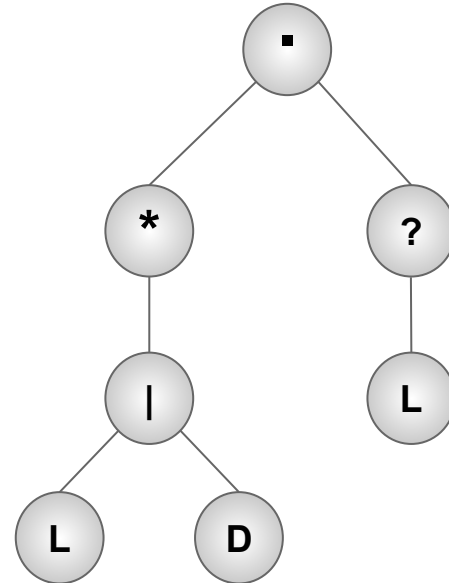
$(L|D)*L?'#\text{'L}$

$\rightarrow ( (L|D)*L?'#\text{'L} )\$$

---

## 2. Construir Árbol Binario de Sintaxis

$( (L|D)*L?'#\text{'L} )\$$



## 1. Agregar Símbolo de Aceptación

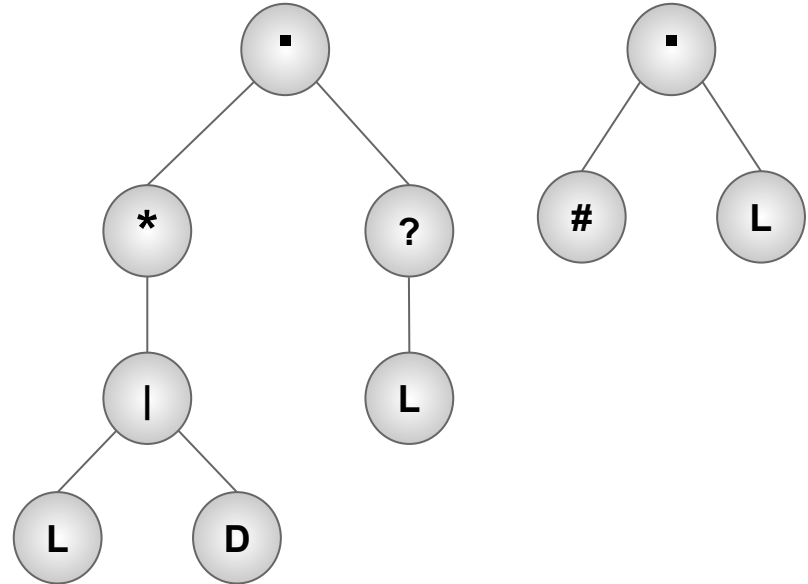
$(L|D)^*L?' \# 'L$

$\rightarrow ( (L|D)^*L?' \# 'L ) \$$

---

## 2. Construir Árbol Binario de Sintaxis

$( (L|D)^*L?' \# 'L ) \$$



## 1. Agregar Símbolo de Aceptación

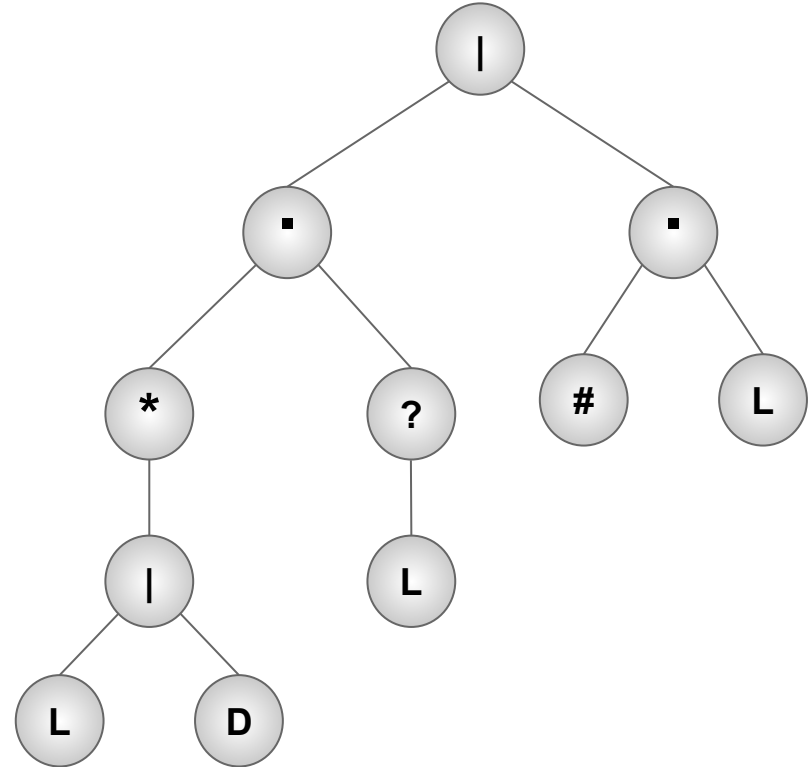
$(L|D)*L?' \# 'L$

$\rightarrow ( (L|D)*L?' \# 'L ) \$$

---

## 2. Construir Árbol Binario de Sintaxis

$( (L|D)*L?' \# 'L ) \$$



## 1. Agregar Símbolo de Aceptación

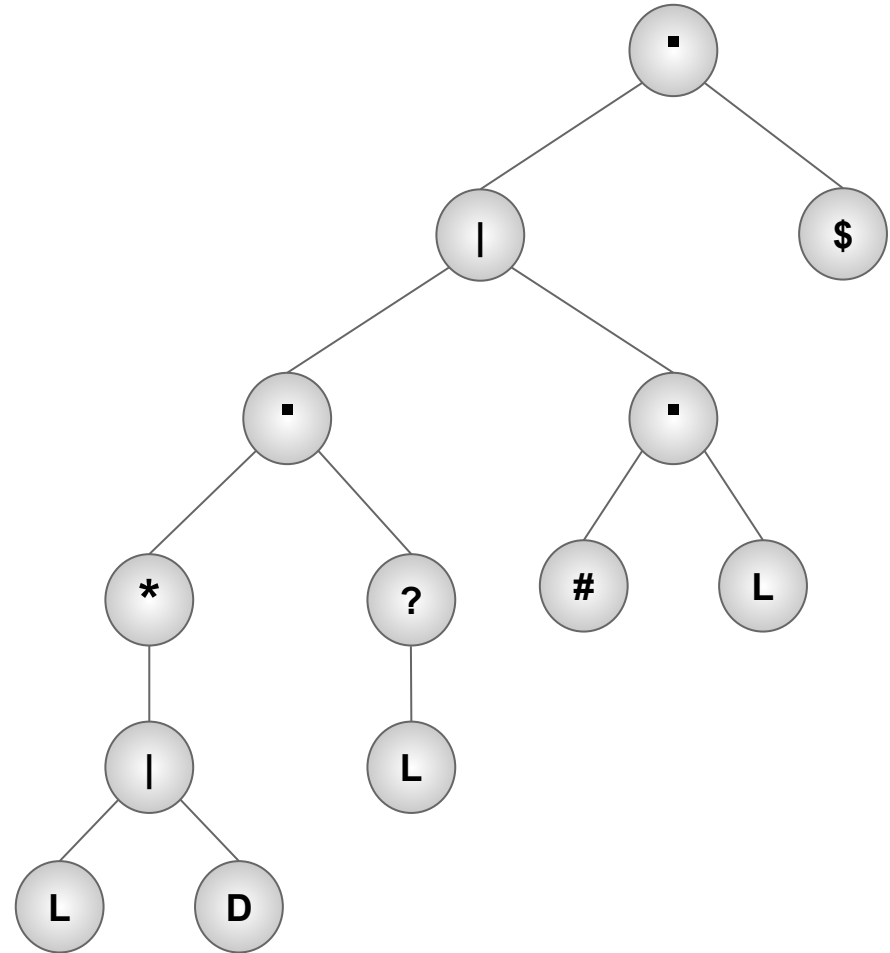
$(L|D)*L?|'\# 'L$

$\rightarrow ( (L|D)*L?|'\# 'L ) \$$

---

## 2. Construir Árbol Binario de Sintaxis

$( (L|D)*L?|'\# 'L ) \$$



## 1. Agregar Símbolo de Aceptación

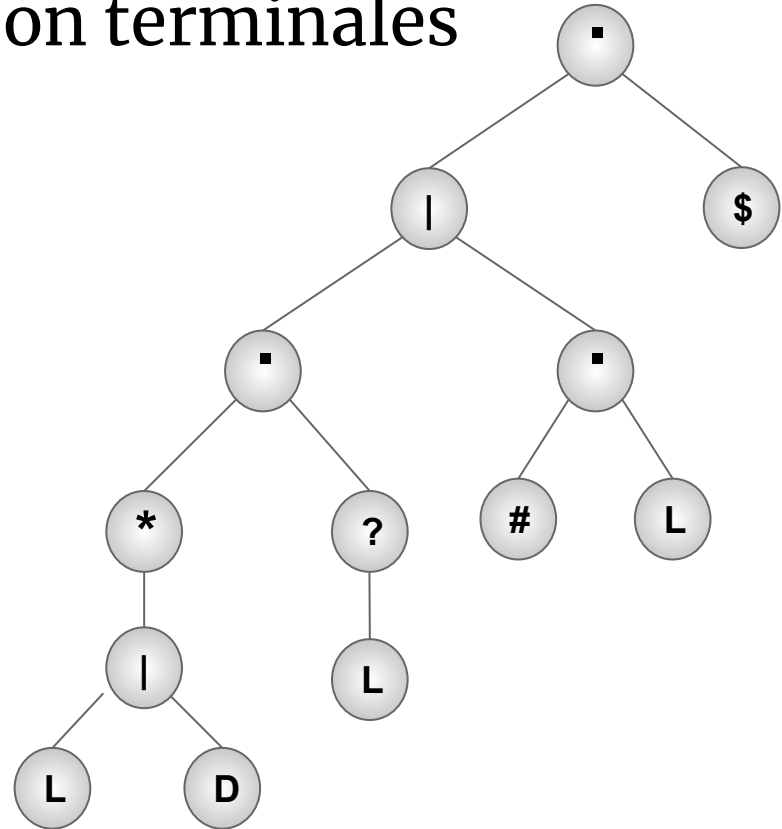
$(L|D)*L?|'\# 'L$   
 $\rightarrow ( (L|D)*L?|'\# 'L )\$$

---

## 2. Construir Árbol Binario de Sintaxis

$( (L|D)*L?|'\# 'L )\$$

## 3. Numerar Hojas con terminales





## 1. Agregar Símbolo de Aceptación

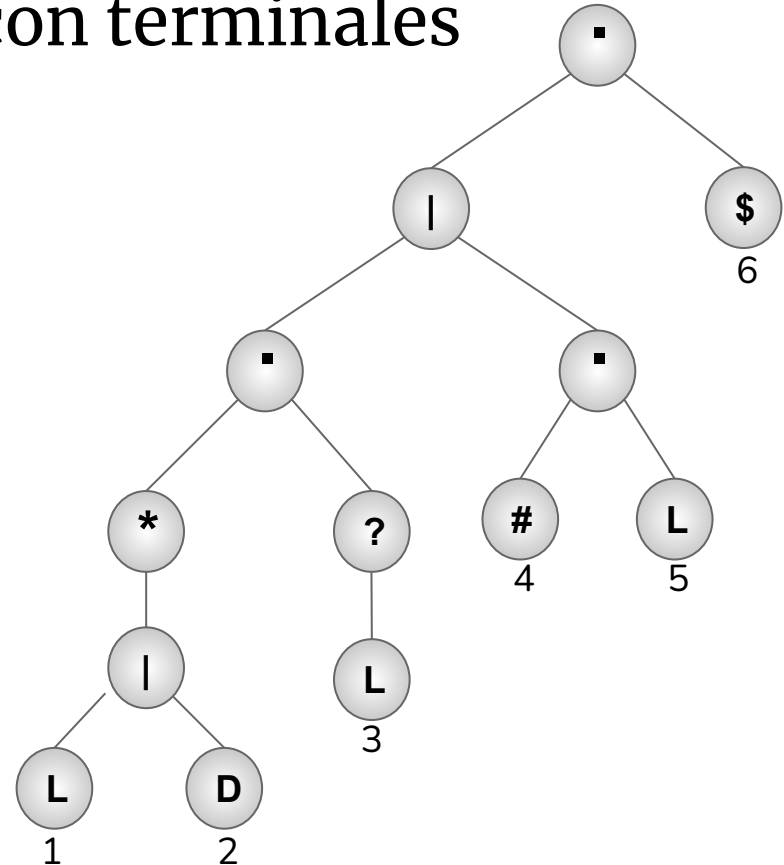
$(L|D)*L?|'\# 'L$   
→  $((L|D)*L?|'\# 'L)\$$

---

## 2. Construir Árbol Binario de Sintaxis

$((L|D)*L?|'\# 'L)\$$

## 3. Numerar Hojas con terminales



# \*\* Mate de Cómputo 1

INPUT		OUTPUT
A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

INPUT		OUTPUT
A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

## 4. Calcular

### 4.1. Anulable

A	B	OR	AND
V	V	V	V
V	F	V	F
F	V	V	F
F	F	F	F

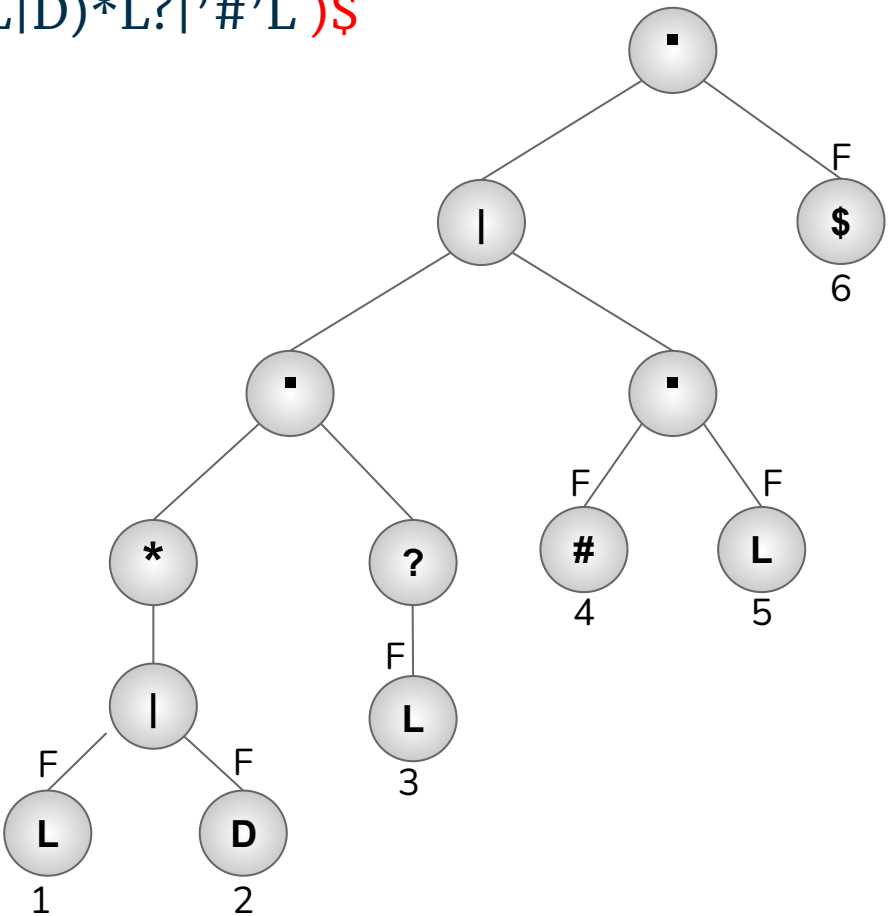
Nodo	Anulable
C1 (hoja)	<b>F</b>
C1   C2	If isNullable(C1)    isNullable(C2) <b>V</b> else <b>F</b>
C1 C2	If isNullable(C1) && isNullable(C2) <b>V</b> else <b>F</b>
C1*	<b>V</b>
C1+	If isNullable(C1) <b>V</b> else <b>F</b>
C1?	<b>V</b>

# 4. Calcular

## 4.1. Anulable

Nodo	Anulable
C1 (hoja)	<b>F</b>
C1   C2	If isNullable(C1)    isNullable(C2) <b>V</b> else <b>F</b>
C1 C2	If isNullable(C1) && isNullable(C2) <b>V</b> else <b>F</b>
C1*	<b>V</b>
C1+	If isNullable(C1) <b>V</b> else <b>F</b>
C1?	<b>V</b>

**( (L|D)\*L?'# 'L )\$**

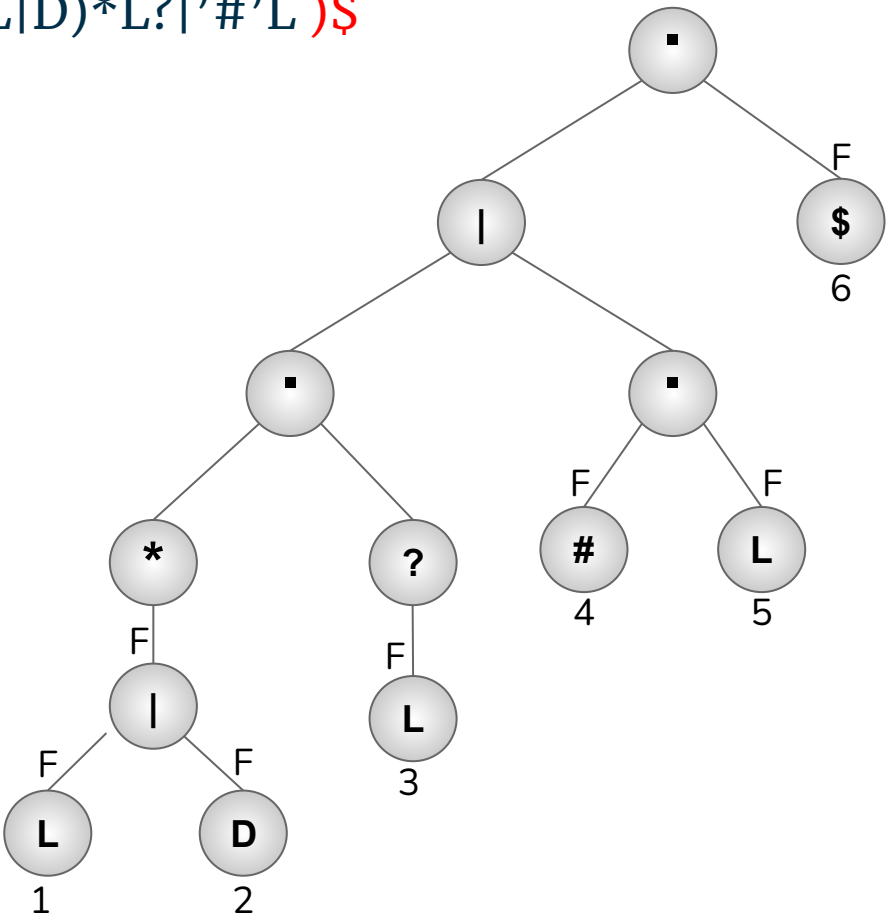


# 4. Calcular

## 4.1. Anulable

Nodo	Anulable
C1 (hoja)	<b>F</b>
C1   C2	If isNullable(C1)    isNullable(C2) <b>V</b> else <b>F</b>
C1 C2	If isNullable(C1) && isNullable(C2) <b>V</b> else <b>F</b>
C1*	<b>V</b>
C1+	If isNullable(C1) <b>V</b> else <b>F</b>
C1?	<b>V</b>

**( (L|D)\*L?'# 'L )\$**

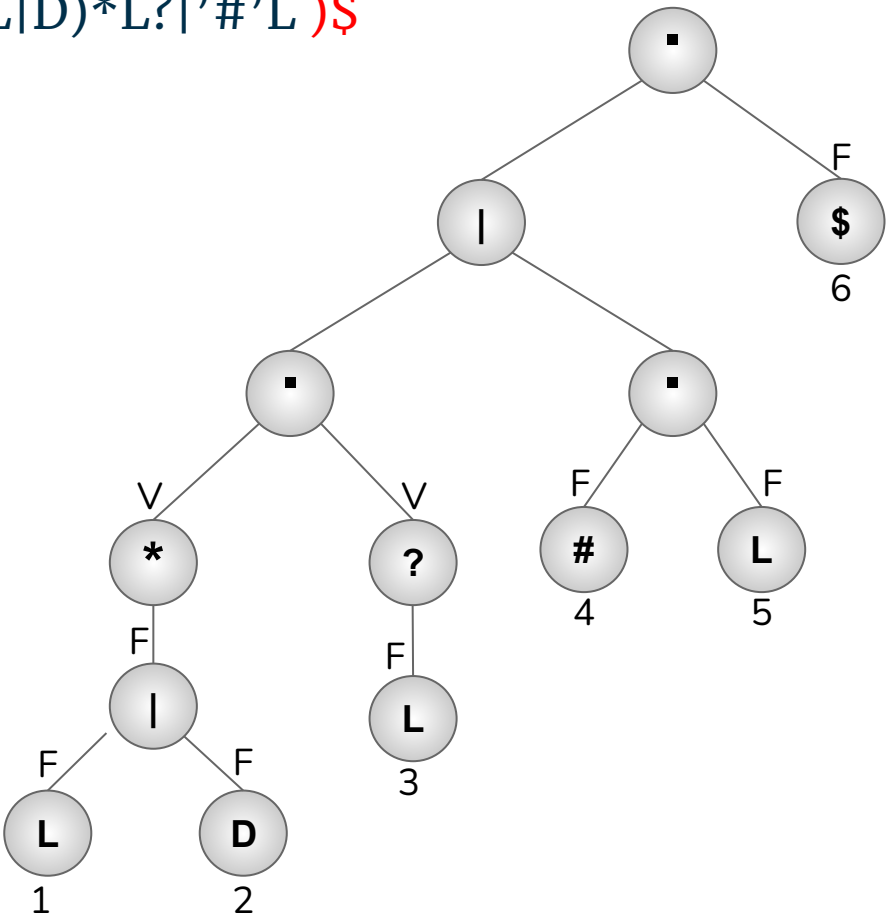


# 4. Calcular

## 4.1. Anulable

Nodo	Anulable
C1 (hoja)	<b>F</b>
C1   C2	If isNullable(C1)    isNullable(C2) <b>V</b> else <b>F</b>
C1 C2	If isNullable(C1) && isNullable(C2) <b>V</b> else <b>F</b>
C1*	<b>V</b>
C1+	If isNullable(C1) <b>V</b> else <b>F</b>
C1?	<b>V</b>

**( (L|D)\*L?'# 'L )\$**

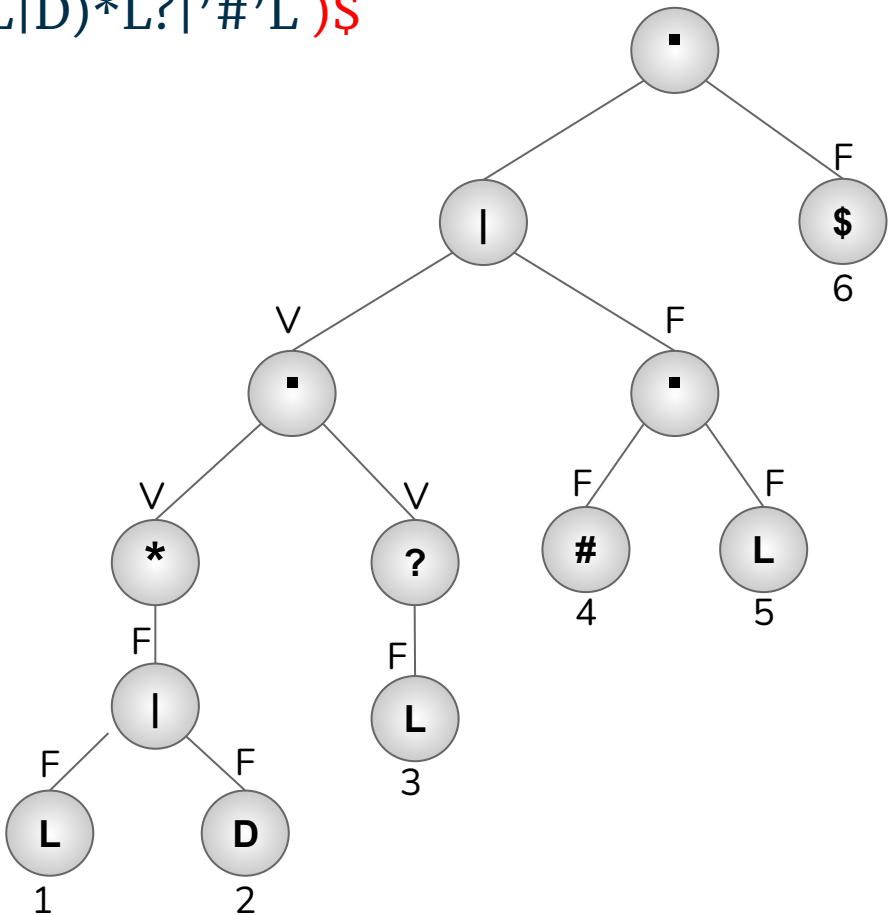


# 4. Calcular

## 4.1. Anulable

Nodo	Anulable
C1 (hoja)	<b>F</b>
C1   C2	If isNullable(C1)    isNullable(C2) <b>V</b> else <b>F</b>
C1 C2	If isNullable(C1) && isNullable(C2) <b>V</b> else <b>F</b>
C1*	<b>V</b>
C1+	If isNullable(C1) <b>V</b> else <b>F</b>
C1?	<b>V</b>

**( (L|D)\*L?|'#'L )\$**

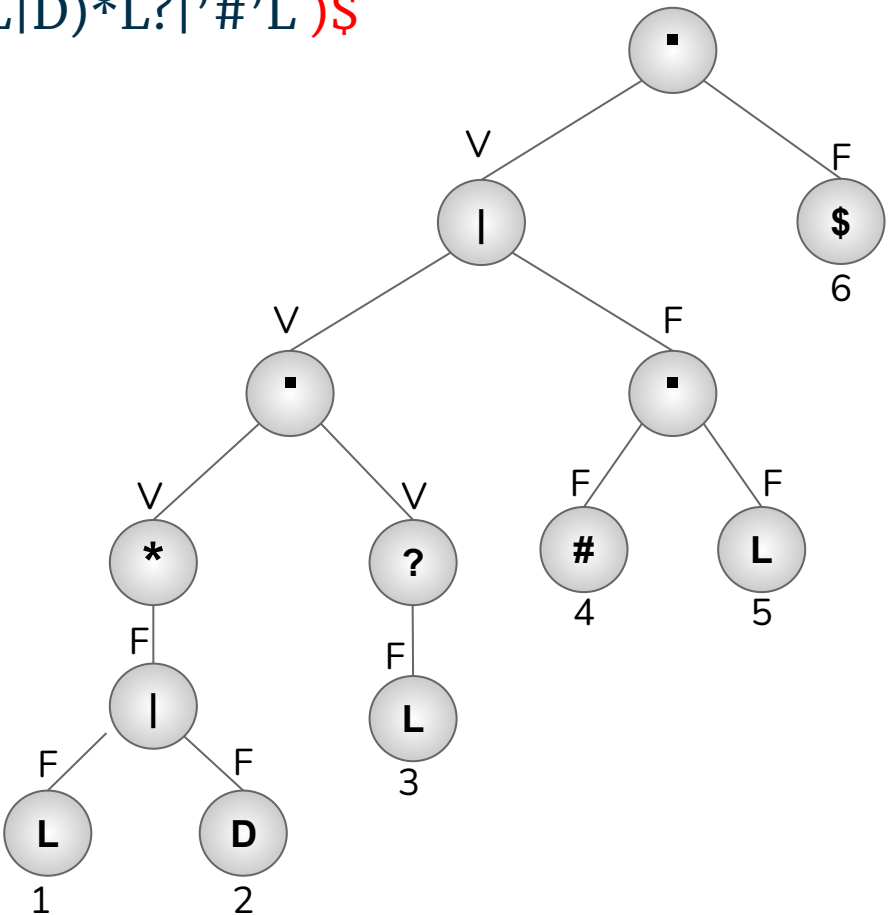


# 4. Calcular

## 4.1. Anulable

Nodo	Anulable
C1 (hoja)	<b>F</b>
C1   C2	If isNullable(C1)    isNullable(C2) <b>V</b> else <b>F</b>
C1 C2	If isNullable(C1) && isNullable(C2) <b>V</b> else <b>F</b>
C1*	<b>V</b>
C1+	If isNullable(C1) <b>V</b> else <b>F</b>
C1?	<b>V</b>

**( (L|D)\*L?|'#'L )\$**



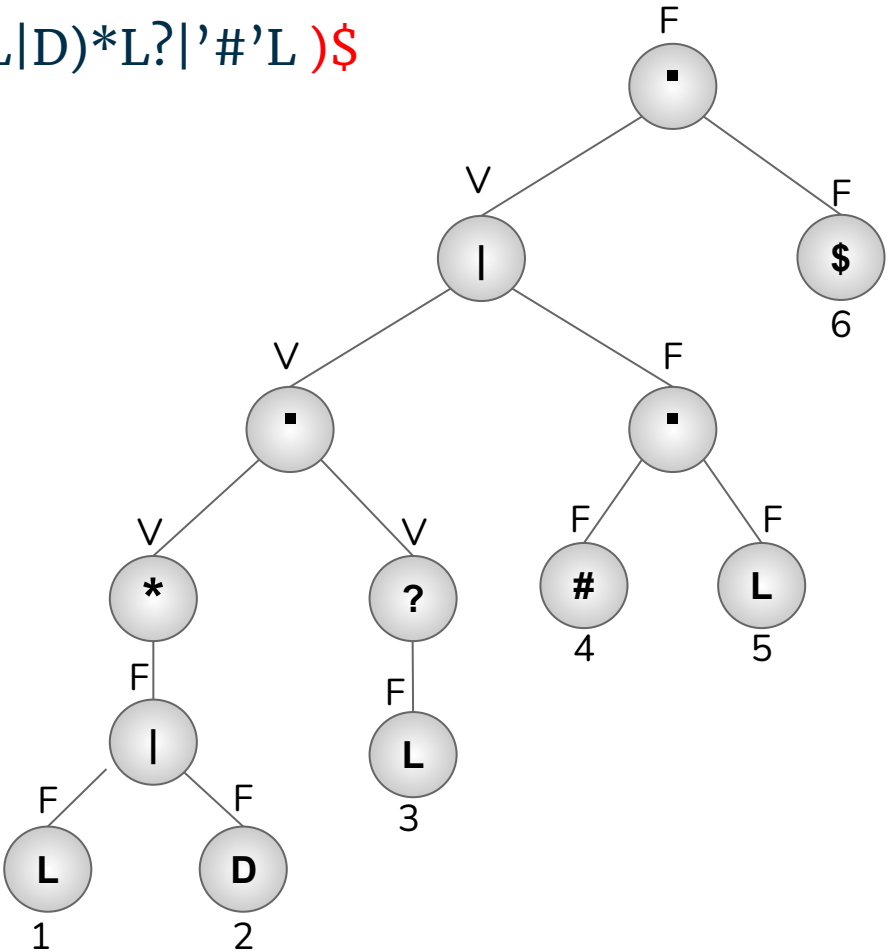


# 4. Calcular

## 4.1. Anulable

Nodo	Anulable
C1 (hoja)	<b>F</b>
C1   C2	If isNullable(C1)    isNullable(C2) <b>V</b> else <b>F</b>
C1 C2	If isNullable(C1) && isNullable(C2) <b>V</b> else <b>F</b>
C1*	<b>V</b>
C1+	If isNullable(C1) <b>V</b> else <b>F</b>
C1?	<b>V</b>

$((L|D)^*L?'# 'L)\$$



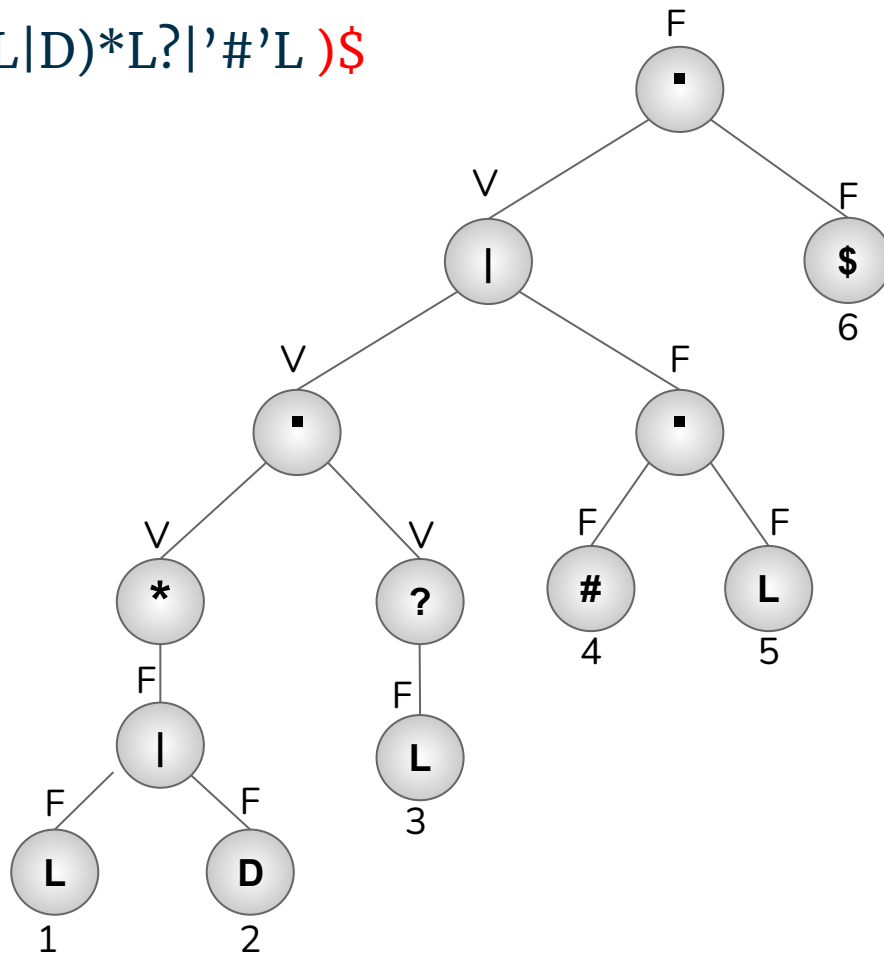
## 4. Calcular

4.1. Anulable

4.2. Primeros

Nodo	Primeros ó F(x)
C1 (hoja)	<b>C1</b>
C1   C2	<b>F(C1) U F(C2)</b>
C1 C2	If isNullable(C1) <b>F(C1) U F(C2)</b> else <b>F(C1)</b>
C1*	<b>F(C1)</b>
C1+	<b>F(C1)</b>
C1?	<b>F(C1)</b>

$((L|D)^*L?'# 'L)\$$



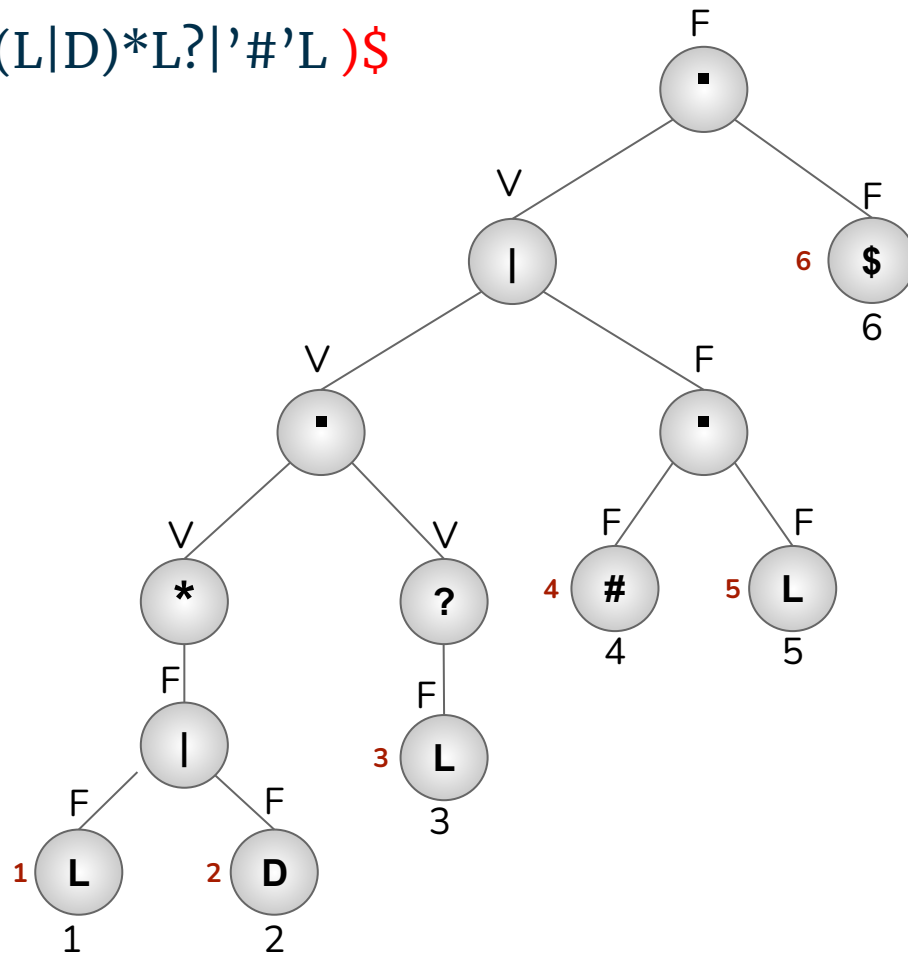
## 4. Calcular

4.1. Anulable

4.2. Primeros

Nodo	Primeros ó F(x)
C1 (hoja)	<b>C1</b>
C1   C2	<b>F(C1) U F(C2)</b>
C1 C2	If isNullable(C1) <b>F(C1) U F(C2)</b> else <b>F(C1)</b>
C1*	<b>F(C1)</b>
C1+	<b>F(C1)</b>
C1?	<b>F(C1)</b>

$((L|D)^*L?' \# 'L) \$$



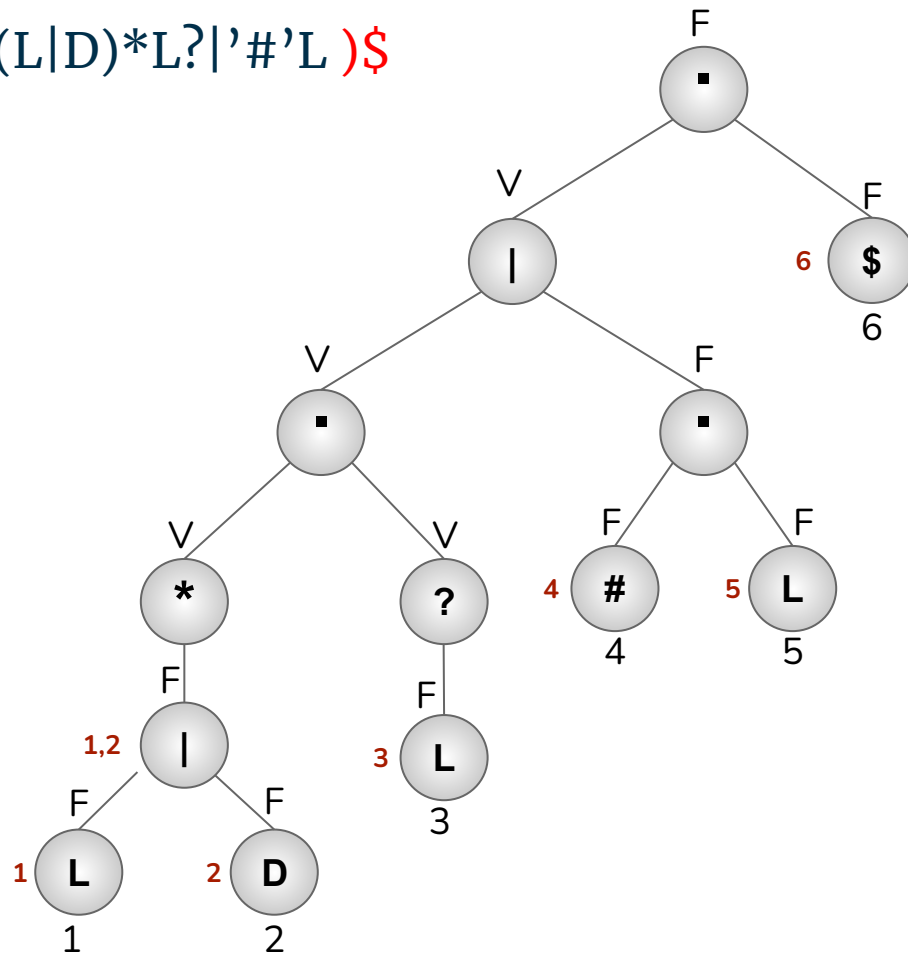
## 4. Calcular

4.1. Anulable

4.2. Primeros

Nodo	Primeros ó F(x)
C1 (hoja)	<b>C1</b>
C1   C2	<b>F(C1) U F(C2)</b>
C1 C2	If isNullable(C1) <b>F(C1) U F(C2)</b> else <b>F(C1)</b>
C1*	<b>F(C1)</b>
C1+	<b>F(C1)</b>
C1?	<b>F(C1)</b>

$((L|D)^*L?' \# 'L) \$$



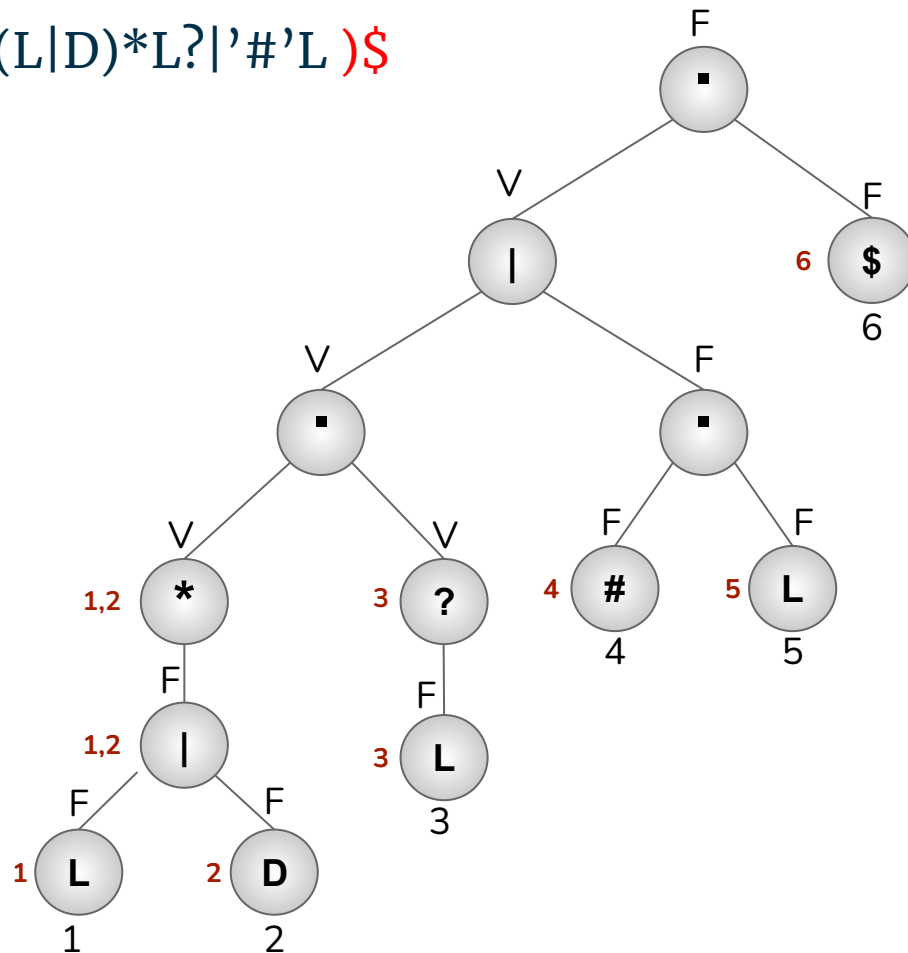
## 4. Calcular

4.1. Anulable

4.2. Primeros

Nodo	Primeros ó F(x)
C1 (hoja)	<b>C1</b>
C1   C2	<b>F(C1) U F(C2)</b>
C1 C2	If isNullable(C1) <b>F(C1) U F(C2)</b> else <b>F(C1)</b>
C1*	<b>F(C1)</b>
C1+	<b>F(C1)</b>
C1?	<b>F(C1)</b>

$((L|D)^*L?' \# 'L) \$$



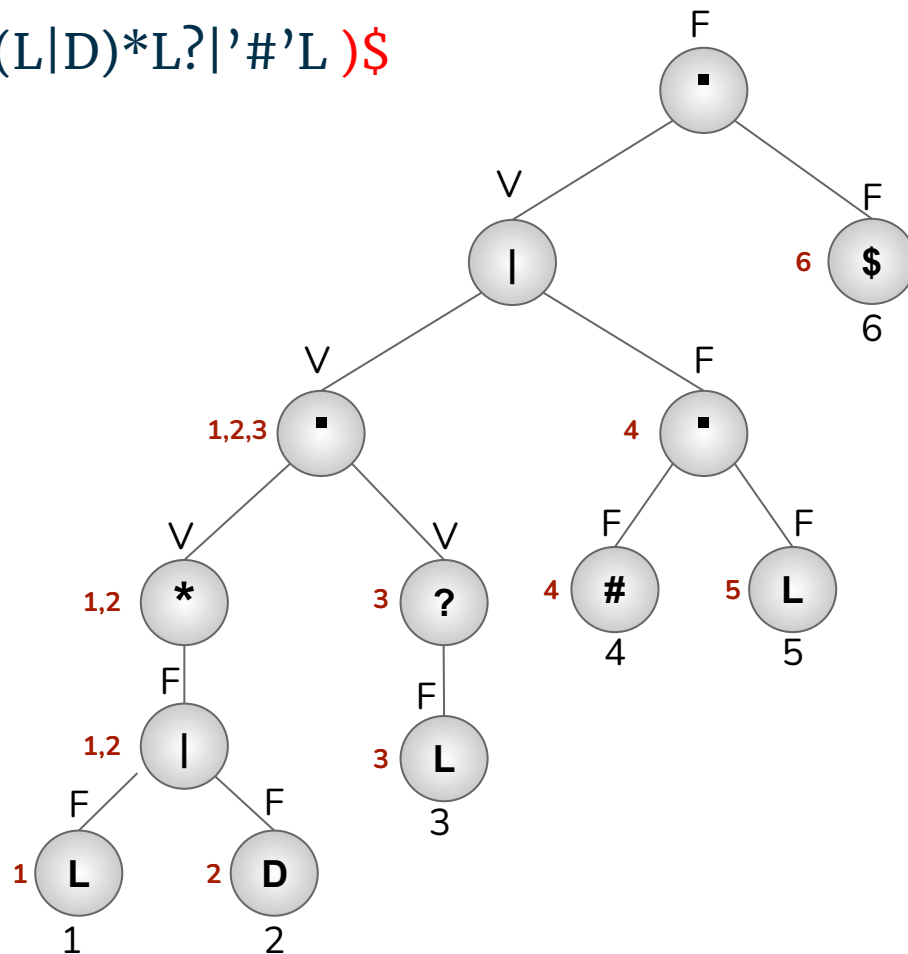
## 4. Calcular

4.1. Anulable

4.2. Primeros

Nodo	Primeros ó F(x)
C1 (hoja)	<b>C1</b>
C1   C2	<b>F(C1) U F(C2)</b>
C1 C2	If isNullable(C1) <b>F(C1) U F(C2)</b> else <b>F(C1)</b>
C1*	<b>F(C1)</b>
C1+	<b>F(C1)</b>
C1?	<b>F(C1)</b>

$((L|D)^*L?' \# 'L) \$$



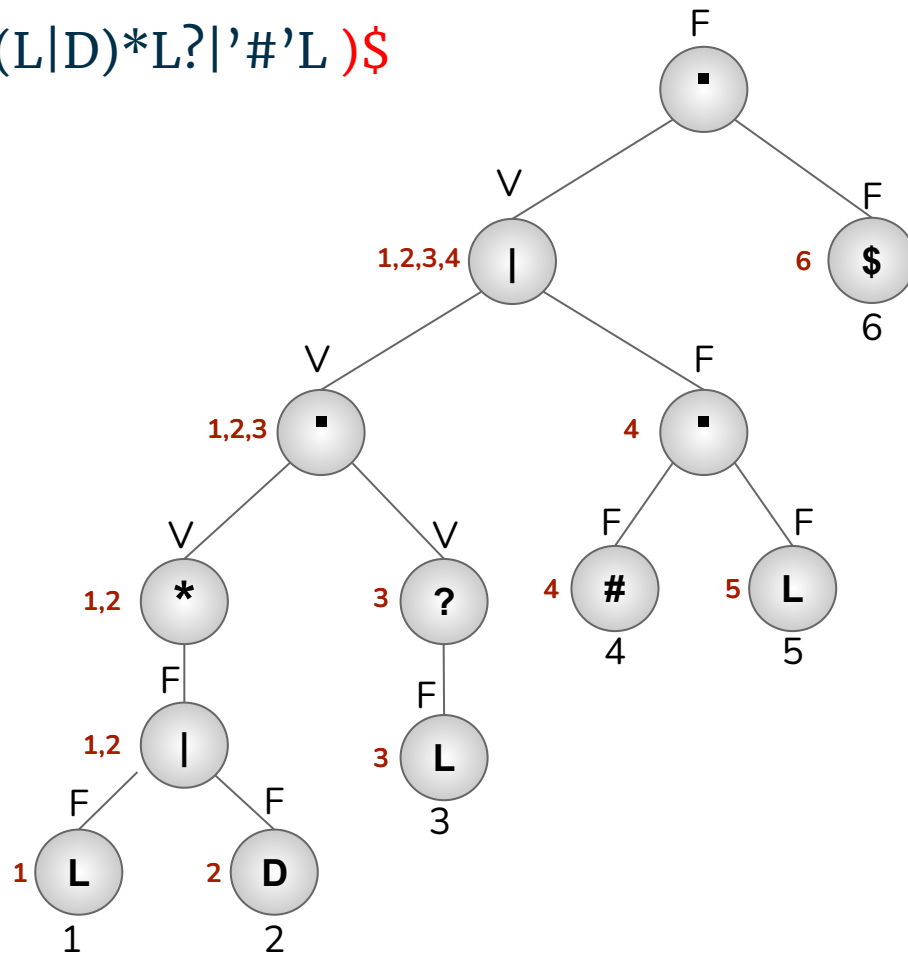
## 4. Calcular

4.1. Anulable

4.2. Primeros

Nodo	Primeros ó F(x)
C1 (hoja)	<b>C1</b>
C1   C2	<b>F(C1) U F(C2)</b>
C1 C2	If isNullable(C1) <b>F(C1) U F(C2)</b> else <b>F(C1)</b>
C1*	<b>F(C1)</b>
C1+	<b>F(C1)</b>
C1?	<b>F(C1)</b>

$((L|D)^*L?'# 'L)\$$



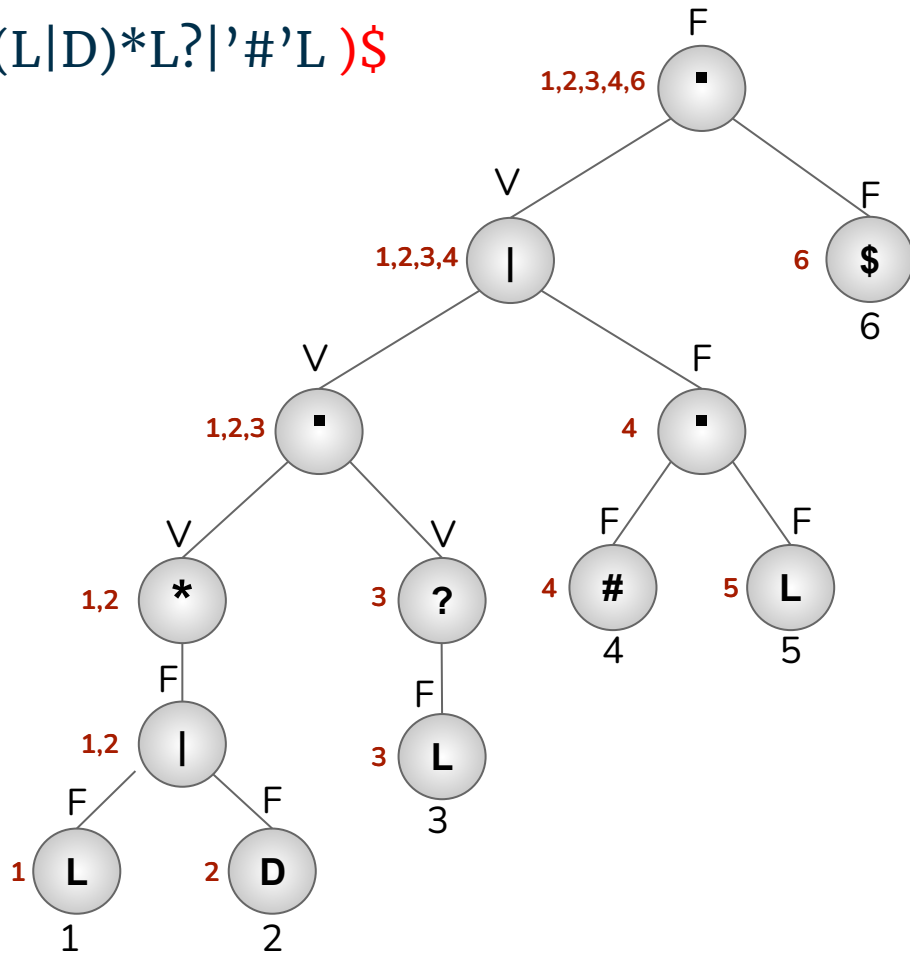
## 4. Calculular

### 4.1. Anulable

## 4.2. Primeros

Nodo	Primeros ó F(x)
C1 (hoja)	<b>C1</b>
C1   C2	<b>F(C1) U F(C2)</b>
C1 C2	If isNullable(C1) <b>F(C1) U F(C2)</b> else <b>F(C1)</b>
C1*	<b>F(C1)</b>
C1+	<b>F(C1)</b>
C1?	<b>F(C1)</b>

**( (L|D)\*L?|'##'L )\$**





## 4. Calculular

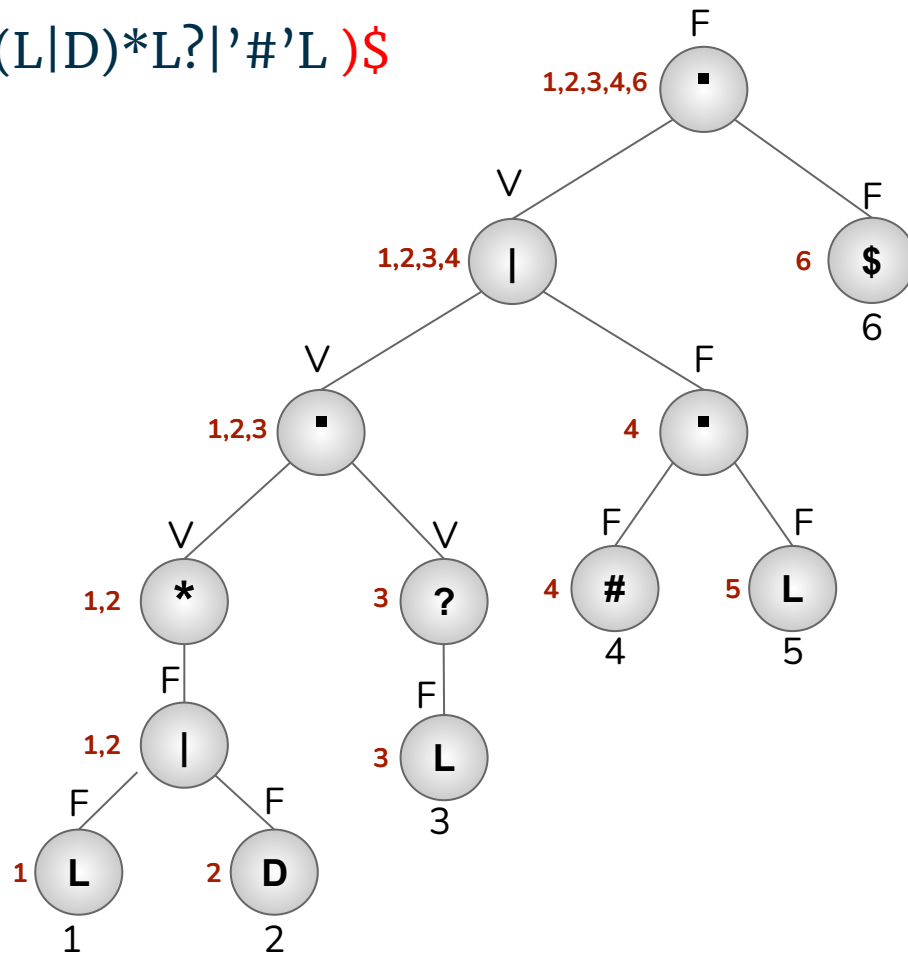
### 4.1. Anulable

## 4.2. Primeros

### 4.3. Últimos

Nodo	Últimos ó L(x)
C1 (hoja)	<b>C1</b>
C1   C2	<b>L(C1) U L(C2)</b>
C1 C2	If isNullable(C2) <b>L(C1) U L(C2)</b> else <b>L(C2)</b>
C1*	<b>L(C1)</b>
C1+	<b>L(C1)</b>
C1?	<b>L(C1)</b>

**( (L|D)\*L?|' #'L )\$**



## 4. Calculular

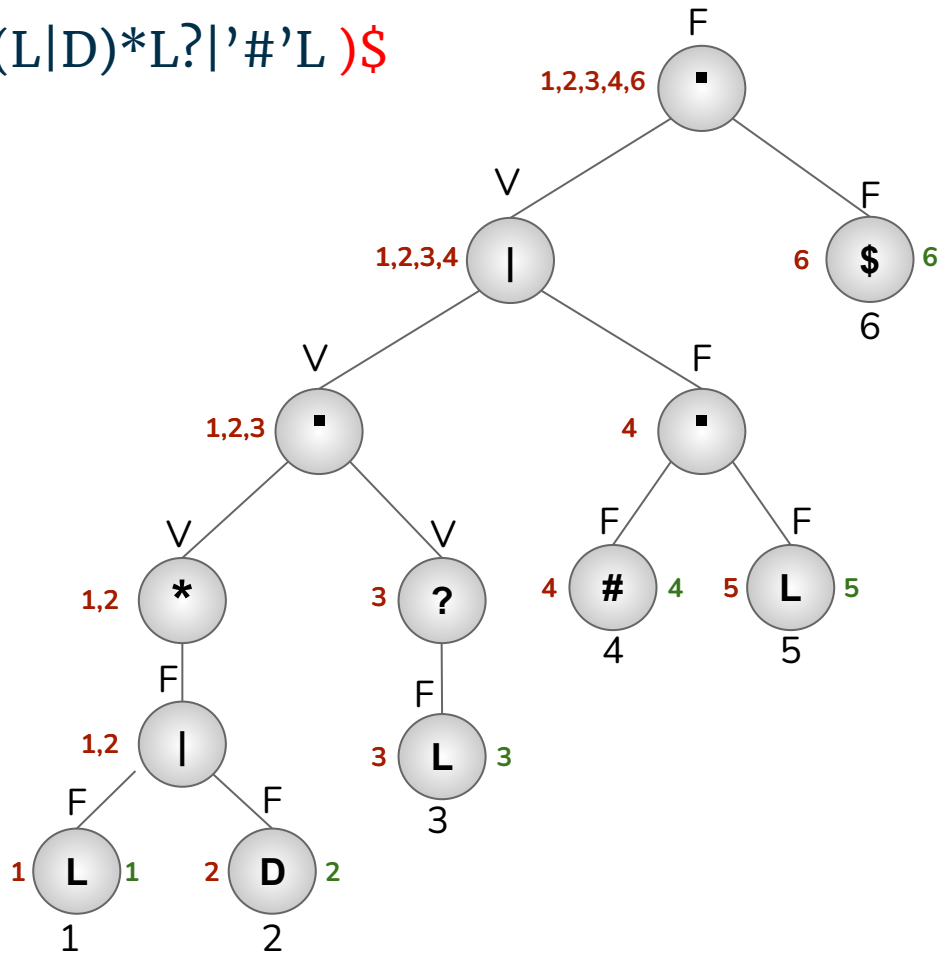
### 4.1. Anulable

## 4.2. Primeros

### 4.3. Últimos

Nodo	Últimos ó L(x)
C1 (hoja)	<b>C1</b>
C1   C2	<b>L(C1) U L(C2)</b>
C1 C2	If isNullable(C2) <b>L(C1) U L(C2)</b> else <b>L(C2)</b>
C1*	<b>L(C1)</b>
C1+	<b>L(C1)</b>
C1?	<b>L(C1)</b>

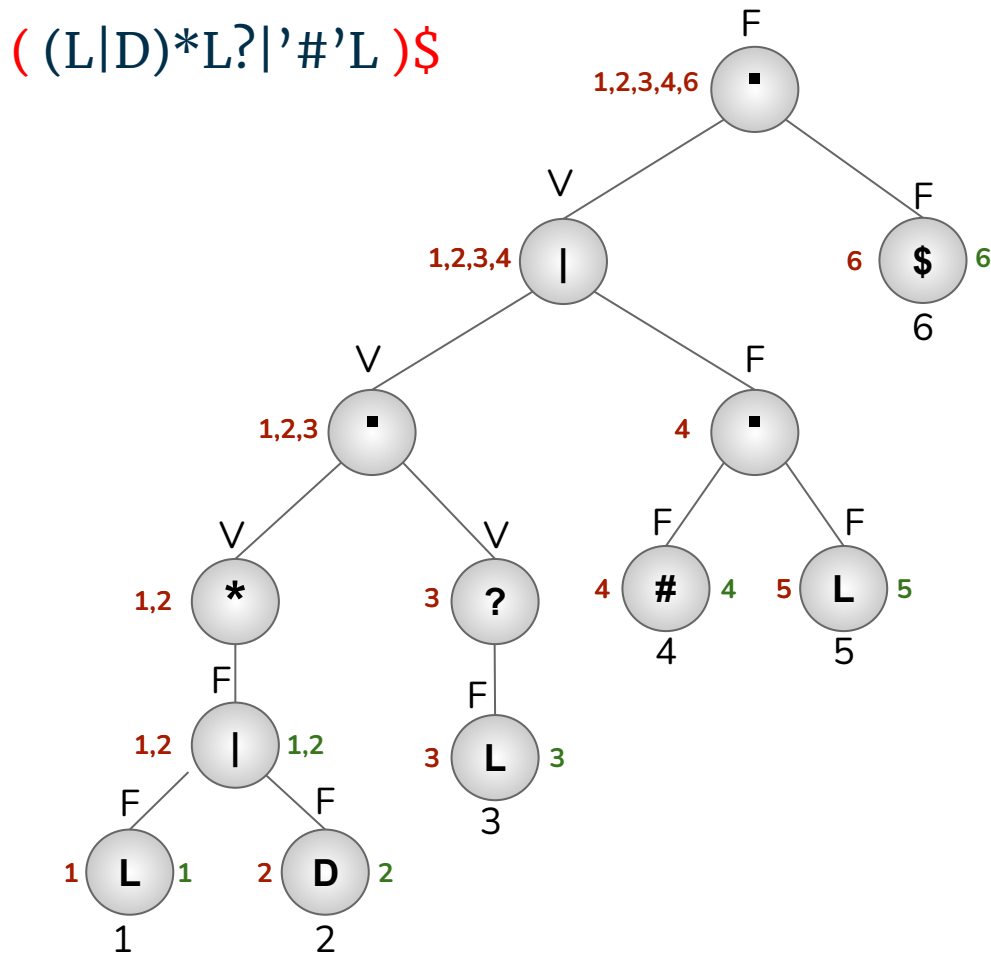
**( (L|D)\*L?|' #'L )\$**



## 4. Calcular

- 4.1. Anulable
- 4.2. Primeros
- 4.3. Últimos

Nodo	Últimos ó L(x)
C1 (hoja)	<b>C1</b>
C1   C2	<b>L(C1) U L(C2)</b>
C1 C2	If isNullable(C2) <b>L(C1) U L(C2)</b> else <b>L(C2)</b>
C1*	<b>L(C1)</b>
C1+	<b>L(C1)</b>
C1?	<b>L(C1)</b>



## 4. Calcular

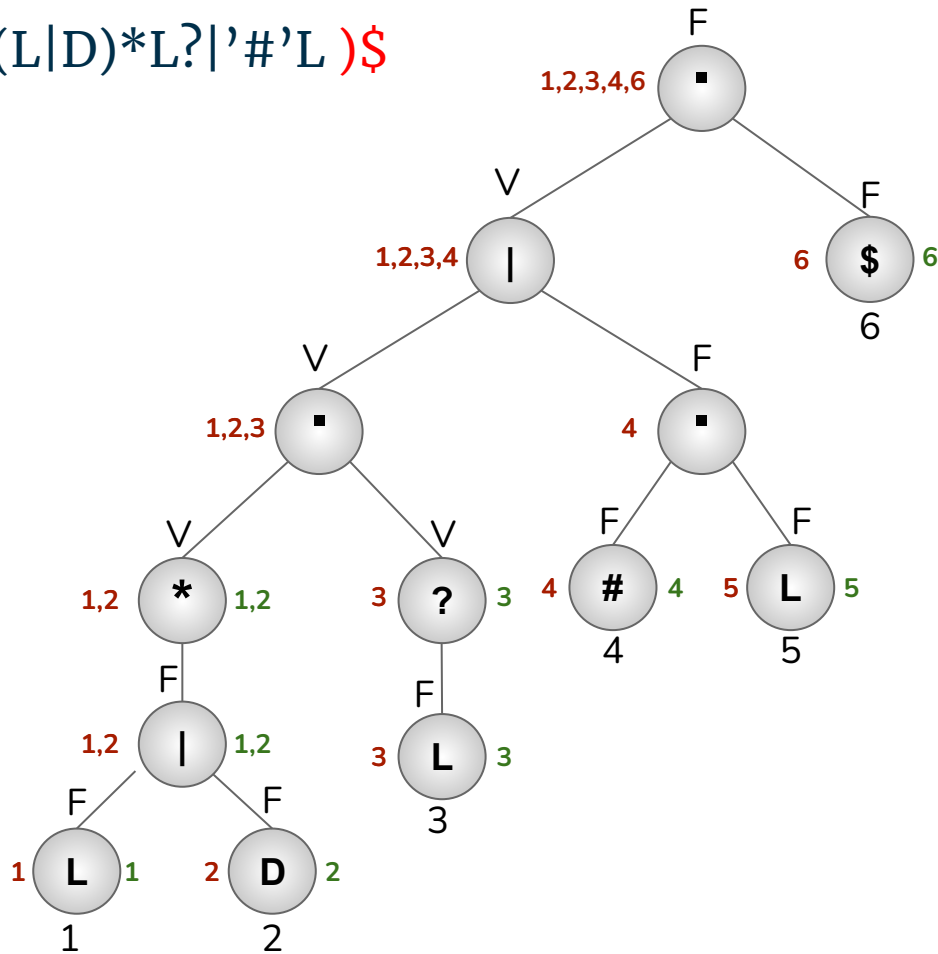
### 4.1. Anulable

## 4.2. Primeros

### 4.3. Últimos

Nodo	Últimos ó L(x)
C1 (hoja)	<b>C1</b>
C1   C2	<b>L(C1) U L(C2)</b>
C1 C2	If isNullable(C2) <b>L(C1) U L(C2)</b> else <b>L(C2)</b>
C1*	<b>L(C1)</b>
C1+	<b>L(C1)</b>
C1?	<b>L(C1)</b>

**( (L|D)\*L?|' #'L )\$**



## 4. Calcular

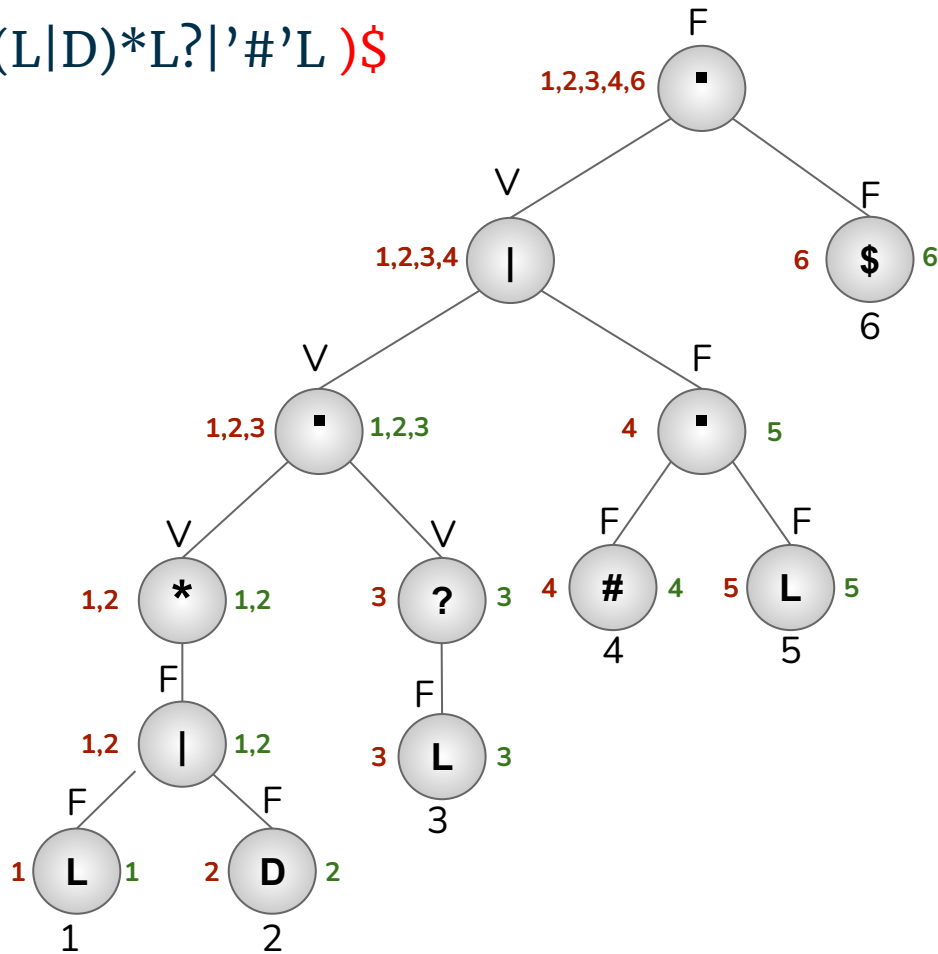
### 4.1. Anulable

## 4.2. Primeros

### 4.3. Últimos

Nodo	Últimos ó L(x)
C1 (hoja)	<b>C1</b>
C1   C2	<b>L(C1) U L(C2)</b>
C1 C2	If isNullable(C2) <b>L(C1) U L(C2)</b> else <b>L(C2)</b>
C1*	<b>L(C1)</b>
C1+	<b>L(C1)</b>
C1?	<b>L(C1)</b>

**( (L|D)\*L?|' #'L )\$**



## 4. Calculular

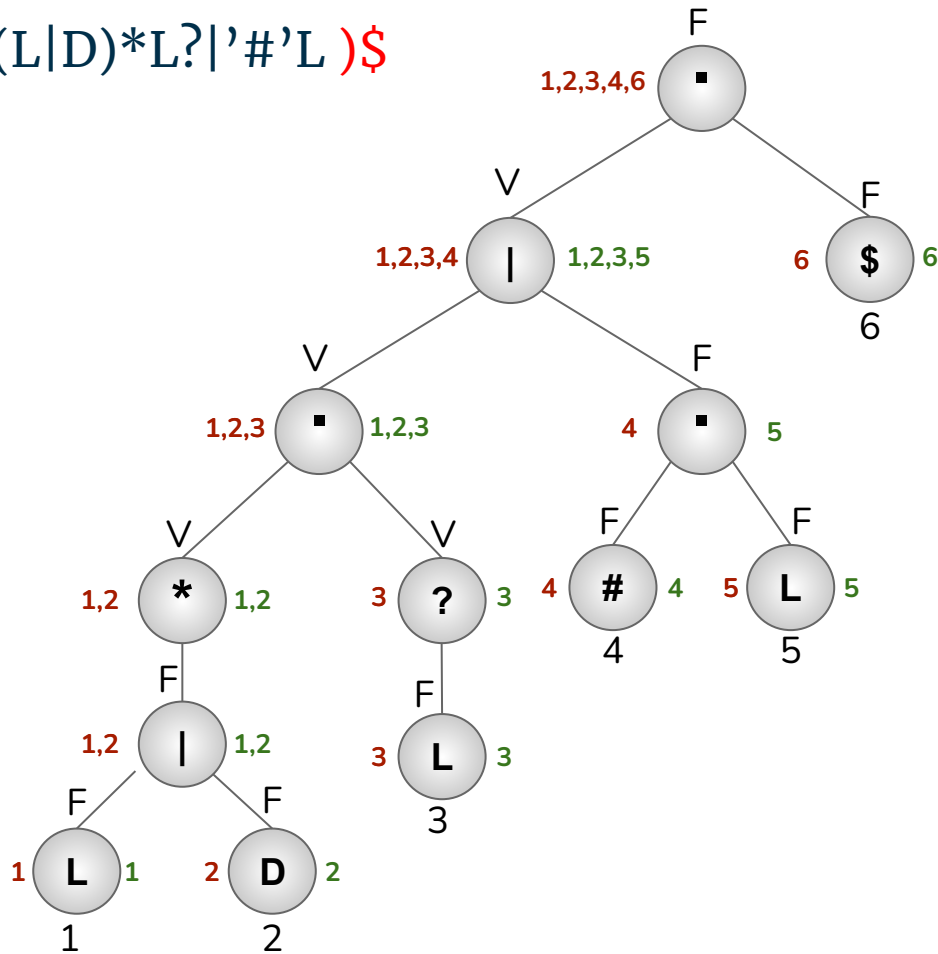
### 4.1. Anulable

## 4.2. Primeros

### 4.3. Últimos

Nodo	Últimos ó L(x)
C1 (hoja)	<b>C1</b>
C1   C2	<b>L(C1) U L(C2)</b>
C1 C2	If isNullable(C2) <b>L(C1) U L(C2)</b> else <b>L(C2)</b>
C1*	<b>L(C1)</b>
C1+	<b>L(C1)</b>
C1?	<b>L(C1)</b>

**( (L|D)\*L?|'##'L )\$**



## 4. Calculular

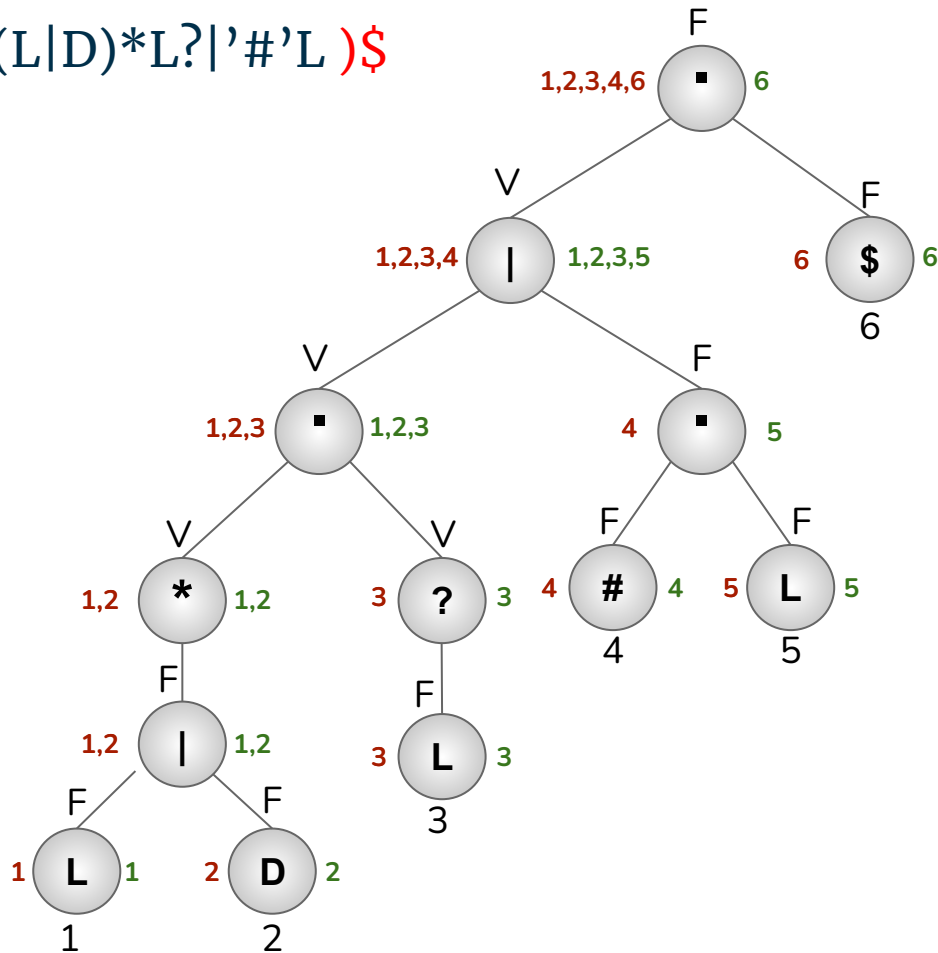
### 4.1. Anulable

## 4.2. Primeros

### 4.3. Últimos

Nodo	Últimos ó L(x)
C1 (hoja)	<b>C1</b>
C1   C2	<b>L(C1) U L(C2)</b>
C1 C2	If isNullable(C2) <b>L(C1) U L(C2)</b> else <b>L(C2)</b>
C1*	<b>L(C1)</b>
C1+	<b>L(C1)</b>
C1?	<b>L(C1)</b>

**( (L|D)\*L?|' #'L )\$**

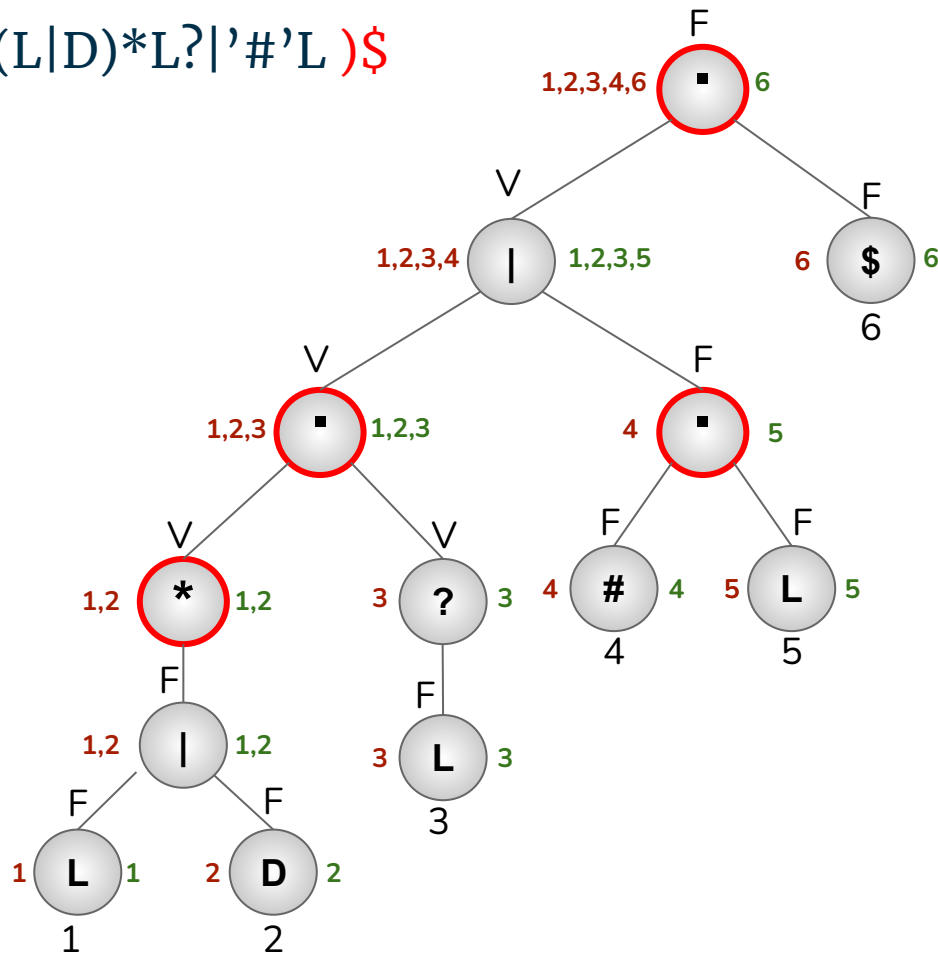


## 5. Calcular Siguietes

Solo se calcula sobre:  
\*, + y ·

Valor	Hoja	Siguietes
L	1	
D	2	
L	3	
#	4	
L	5	
\$	6	

$((L|D)*L?'#L) \$$

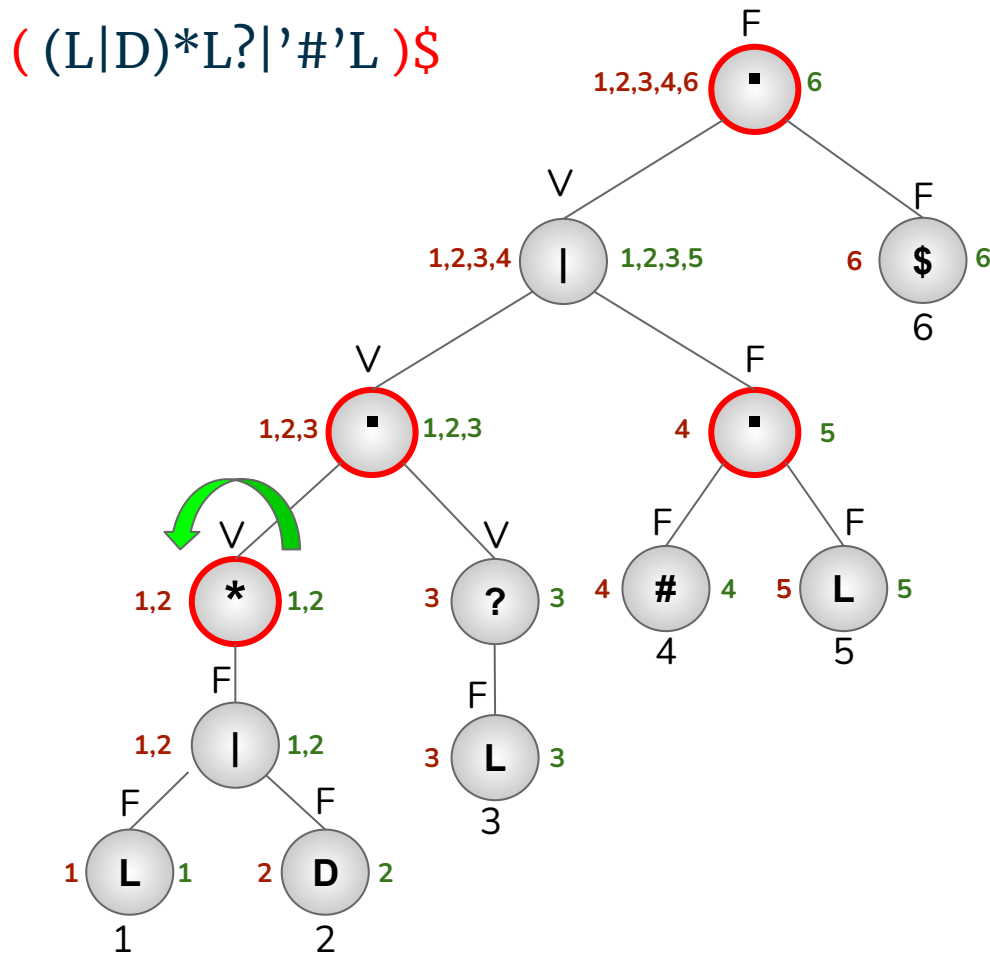




## 5. Calcular Siguientes

Solo se calcula sobre:  
\*, + y ·

Valor	Hoja	Siguientes
L	1	
D	2	
L	3	
#	4	
L	5	
\$	6	

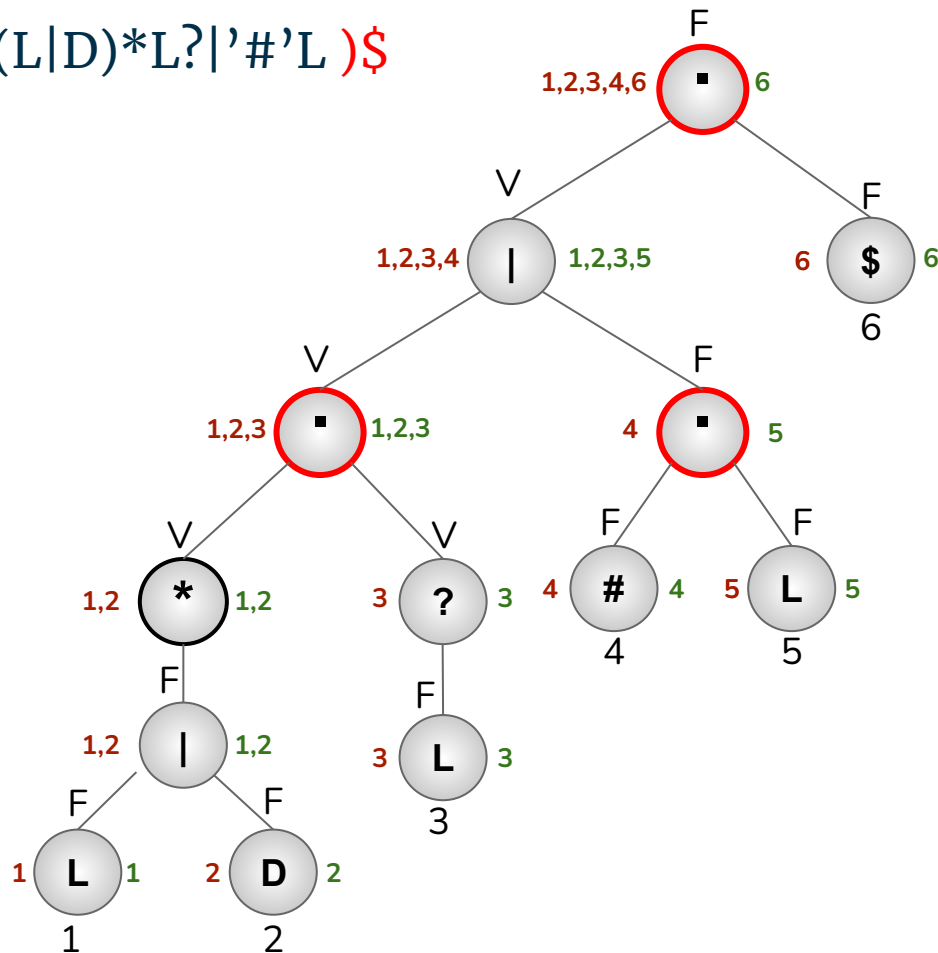


## 5. Calcular Siguietes

Solo se calcula sobre:  
\*, + y ·

Valor	Hoja	Siguietes
L	1	1,2
D	2	1,2
L	3	
#	4	
L	5	
\$	6	

$((L|D)*L?'#L)\$$

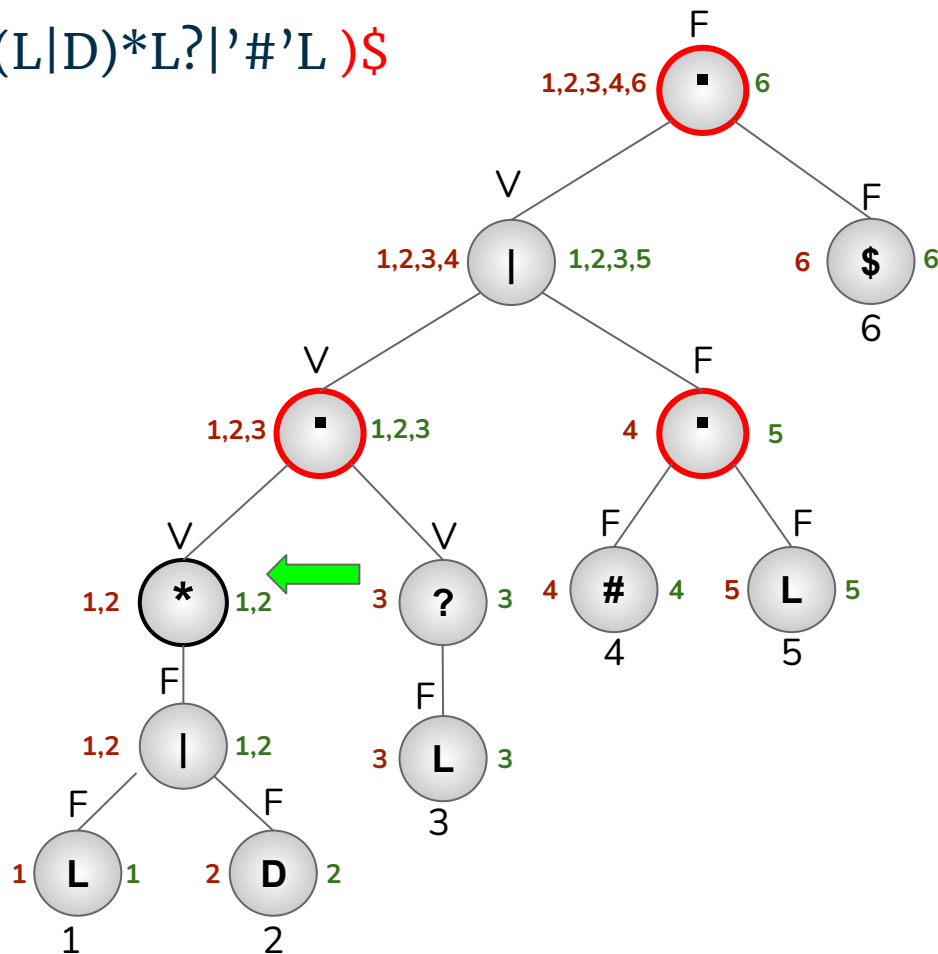


## 5. Calcular Siguietes

Solo se calcula sobre:  
\*, + y ·

Valor	Hoja	Siguietes
L	1	1,2
D	2	1,2
L	3	
#	4	
L	5	
\$	6	

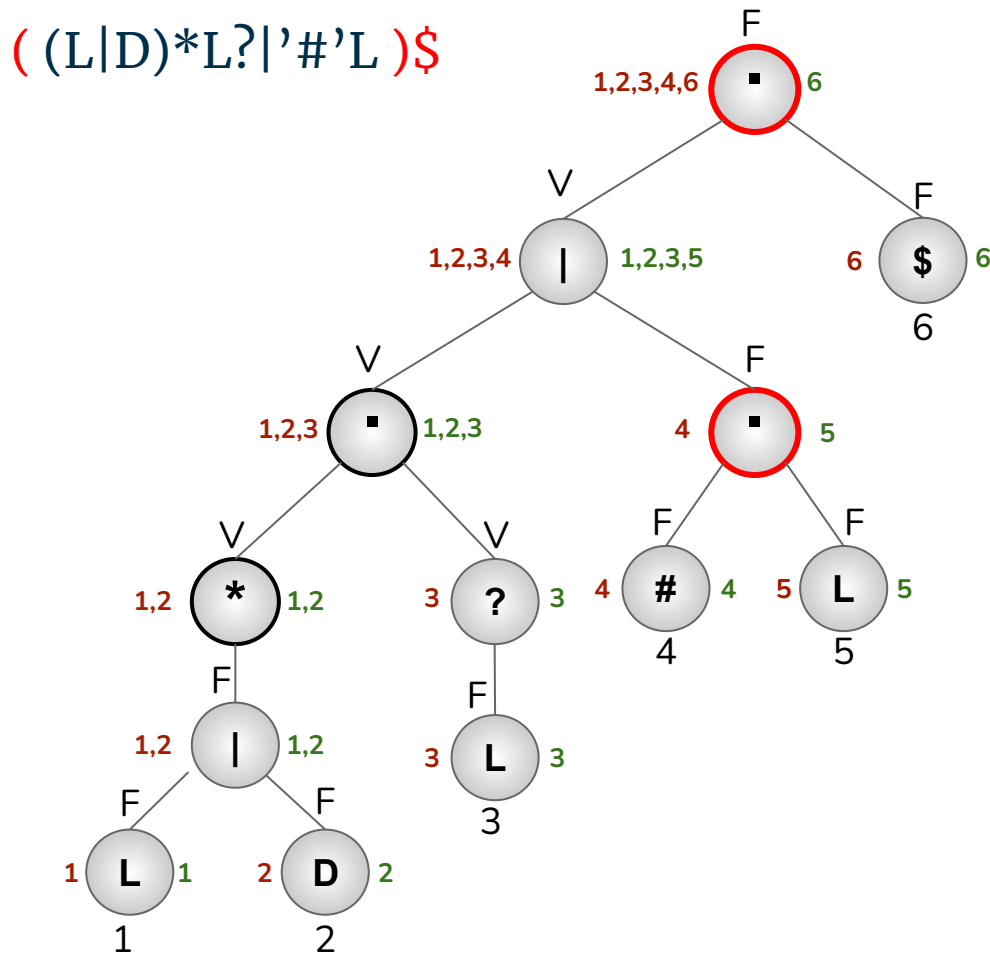
$((L|D)*L?'# 'L) \$$



## 5. Calcular Siguientes

Solo se calcula sobre:  
\*, + y ·

Valor	Hoja	Siguientes
L	1	1,2,3
D	2	1,2,3
L	3	
#	4	
L	5	
\$	6	

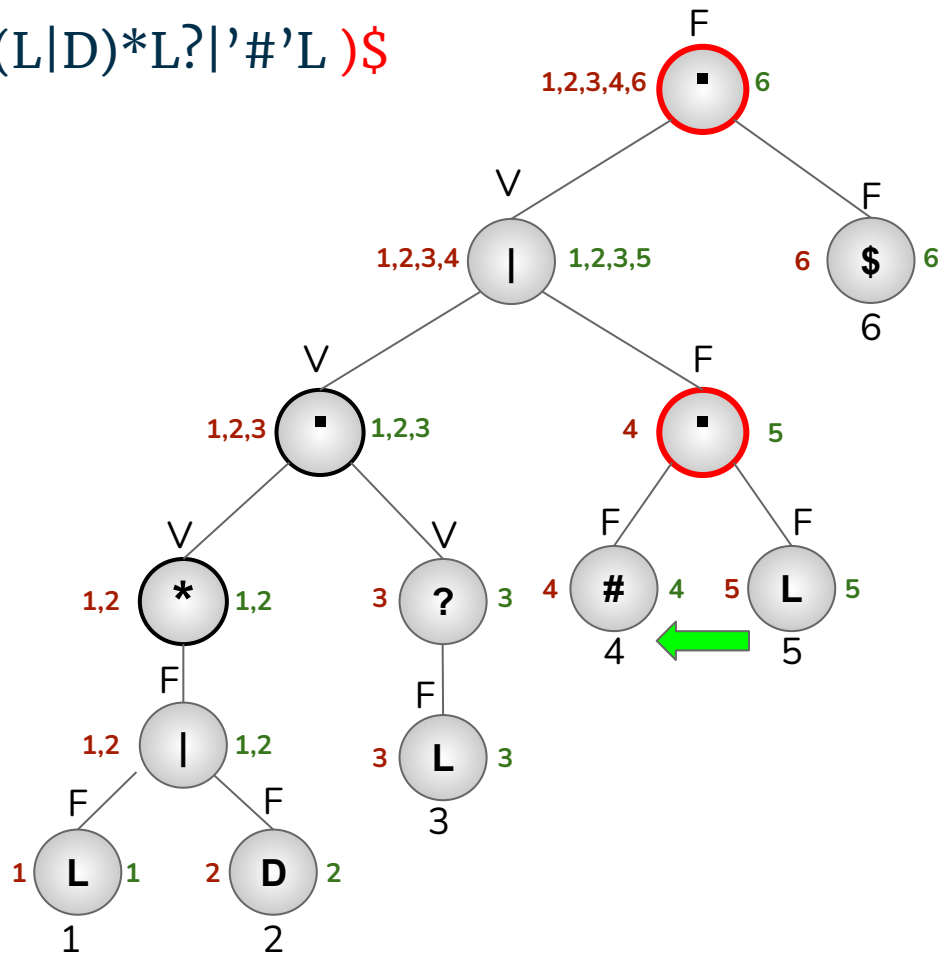


## 5. Calcular Siguientes

Solo se calcula sobre:  
\*, + y ·

Valor	Hoja	Siguientes
L	1	1,2,3
D	2	1,2,3
L	3	
#	4	
L	5	
\$	6	

$((L|D)^*L?|'\# 'L)\$$

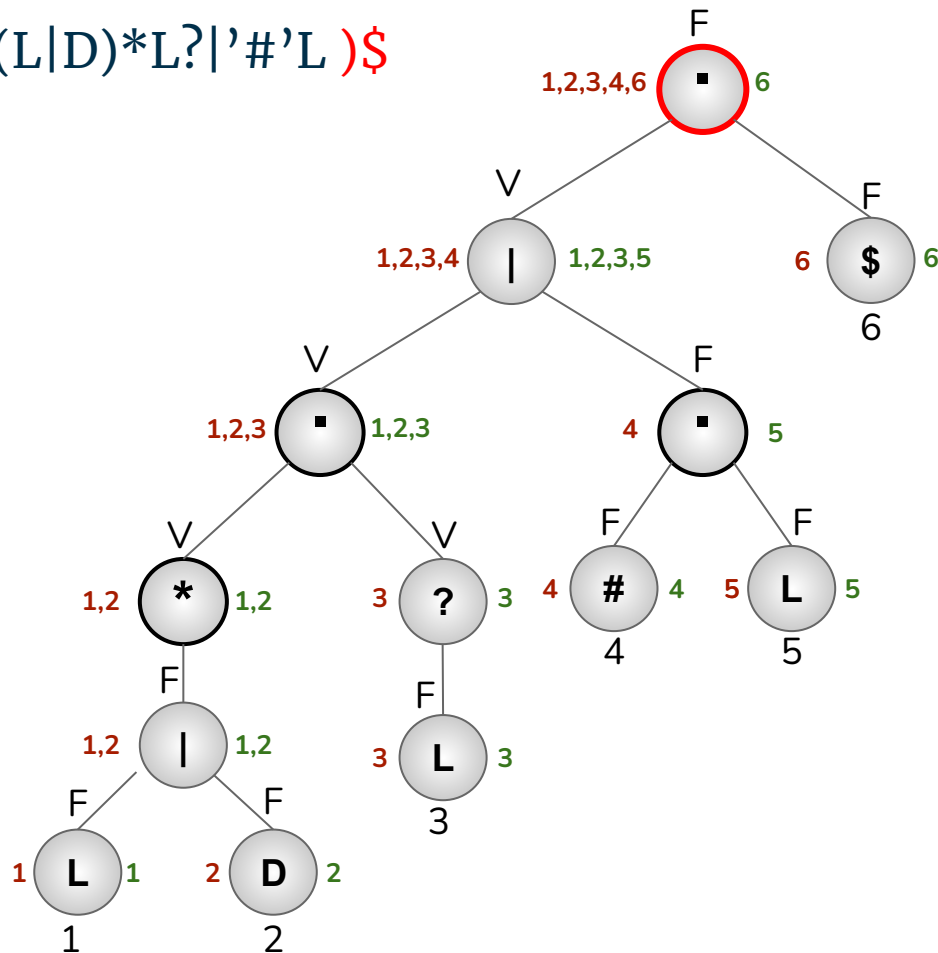


## 5. Calcular Siguietes

Solo se calcula sobre:  
\*, + y ·

Valor	Hoja	Siguietes
L	1	1,2,3
D	2	1,2,3
L	3	
#	4	5
L	5	
\$	6	

$((L|D)*L?'# 'L) \$$

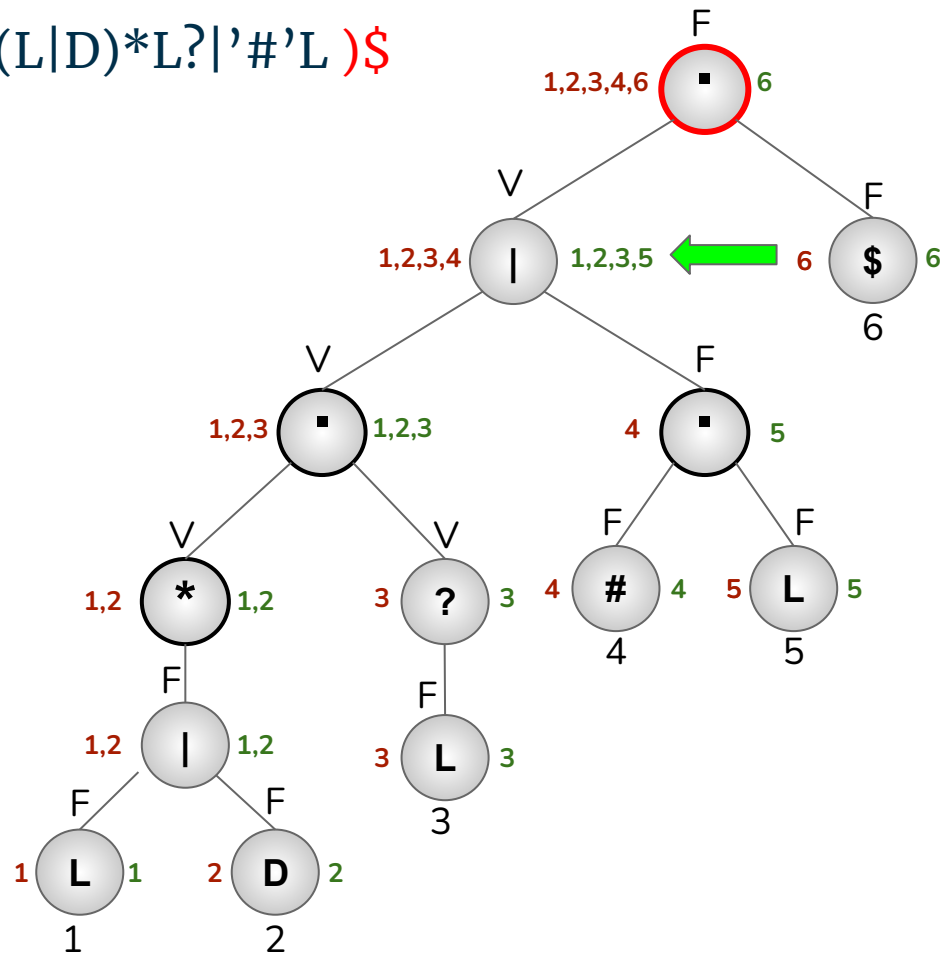


# 5. Calcular Siguietes

Solo se calcula sobre:  
\*, + y ·

Valor	Hoja	Siguietes
L	1	1,2,3
D	2	1,2,3
L	3	
#	4	5
L	5	
\$	6	

$((L|D)*L?'# 'L)\$$

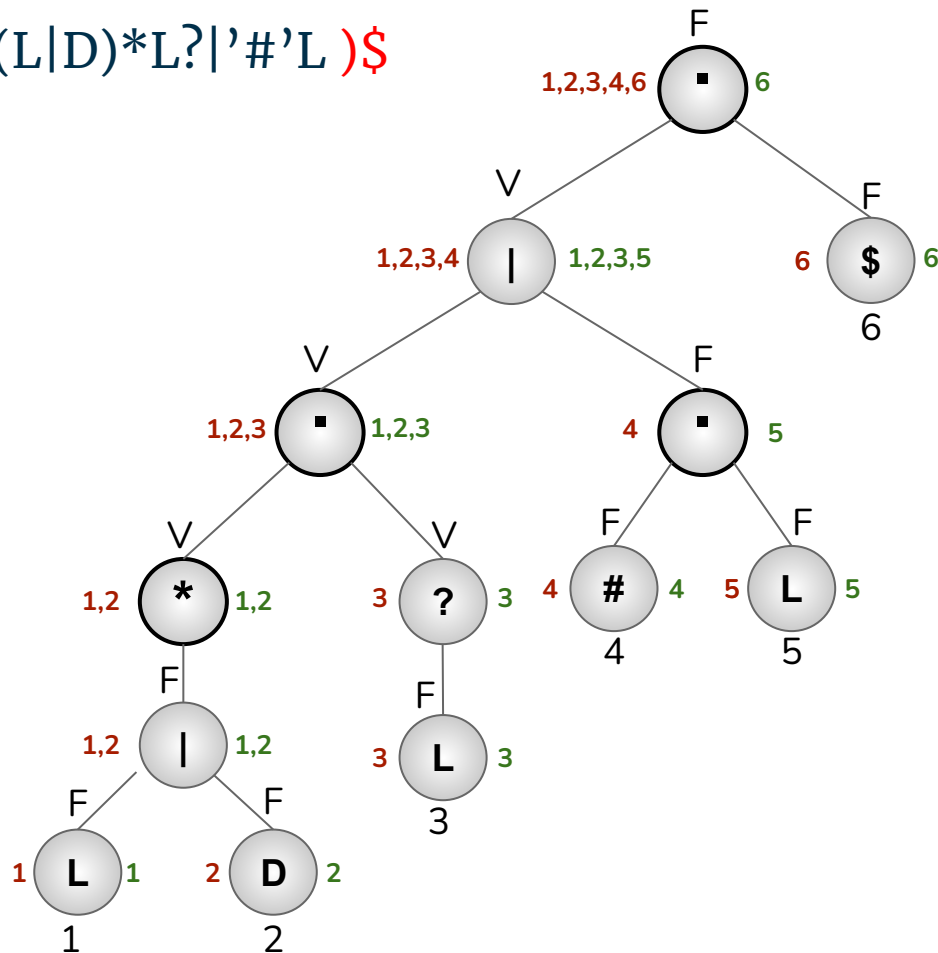


## 5. Calcular Siguietes

Solo se calcula sobre:  
\*, + y ·

Valor	Hoja	Siguietes
L	1	1,2,3,6
D	2	1,2,3,6
L	3	6
#	4	5
L	5	6
\$	6	

$((L|D)*L?'# 'L) \$$



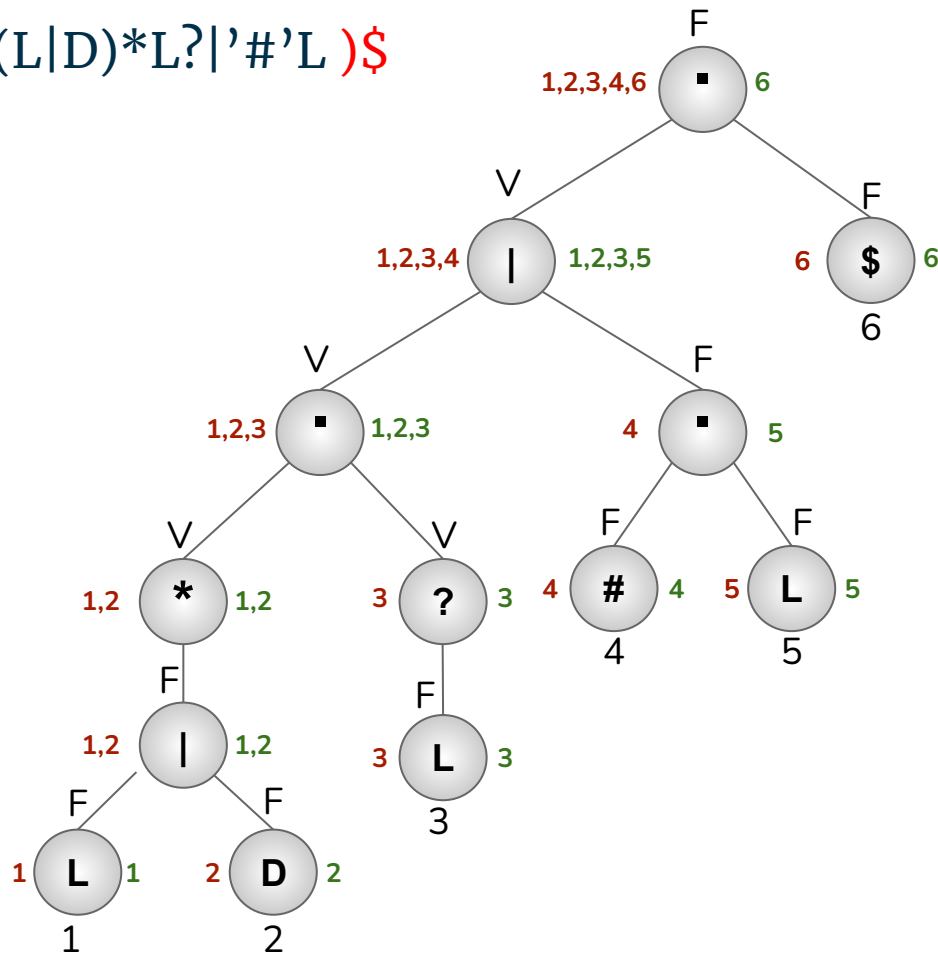


## 5. Calcular Siguietes

Solo se calcula sobre:  
\*, + y ·

Valor	Hoja	Siguietes
L	1	1,2,3,6
D	2	1,2,3,6
L	3	6
#	4	5
L	5	6
\$	6	--

$((L|D)*L?'# 'L) \$$



## 6. Construir Tabla de Transiciones

Valor	Hoja	Siguientes
L	1	1,2,3,6
D	2	1,2,3,6
L	3	6
#	4	5
L	5	6
\$	6	--

**\*\* El estado inicial es F(raíz)**

	Estado	Valores	Siguientes
Inicio	<b>S0</b>	L,D,L,#,\$ 1,2,3,4,6	

## 6. Construir Tabla de Transiciones

Valor	Hoja	Siguientes
L	1	1,2,3,6
D	2	1,2,3,6
L	3	6
#	4	5
L	5	6
\$	6	--

**\*\* El estado inicial es F(raíz)**

	Estado	Valores	Siguientes
Inicio	<b>S0</b>	L,D,L,#,\$ 1,2,3,4,6	<b>L: {1,2,3,6} = S1</b>

## 6. Construir Tabla de Transiciones

Valor	Hoja	Siguientes
L	1	1,2,3,6
D	2	1,2,3,6
L	3	6
#	4	5
L	5	6
\$	6	--

**\*\* El estado inicial es F(raíz)**

	Estado	Valores	Siguientes
Inicio	<b>S0</b>	L,D,L,#,\$ 1,2,3,4,6	L: {1,2,3,6} = <b>S1</b> D: {1,2,3,6} = S1

## 6. Construir Tabla de Transiciones

Valor	Hoja	Siguientes
L	1	1,2,3,6
D	2	1,2,3,6
L	3	6
#	4	5
L	5	6
\$	6	--

**\*\* El estado inicial es F(raíz)**

	Estado	Valores	Siguientes
Inicio	<b>S0</b>	L,D,L,#,\$ 1,2,3,4,6	L: {1,2,3,6} = <b>S1</b> D: {1,2,3,6} = S1 #: {5} = <b>S2</b>

## 6. Construir Tabla de Transiciones

Valor	Hoja	Siguientes
L	1	1,2,3,6
D	2	1,2,3,6
L	3	6
#	4	5
L	5	6
\$	6	--

**\*\* El estado inicial es F(raíz)**

	Estado	Valores	Siguientes
Inicio	S0	L,D,L,#,\$ 1,2,3,4,6	L: {1,2,3,6} = S1 D: {1,2,3,6} = S1 #: {5} = S2
	S1	L,D,L,\$ 1,2,3,6	

## 6. Construir Tabla de Transiciones

Valor	Hoja	Siguientes
L	1	1,2,3,6
D	2	1,2,3,6
L	3	6
#	4	5
L	5	6
\$	6	--

**\*\* El estado inicial es F(raíz)**

	Estado	Valores	Siguientes
Inicio	<b>S0</b>	L,D,L,#,\$ 1,2,3,4,6	L: {1,2,3,6} = <b>S1</b> D: {1,2,3,6} = S1 #: {5} = <b>S2</b>
	<b>S1</b>	L,D,L,\$ 1,2,3,6	L: {1,2,3,6} = S1

## 6. Construir Tabla de Transiciones

Valor	Hoja	Siguientes
L	1	1,2,3,6
D	2	1,2,3,6
L	3	6
#	4	5
L	5	6
\$	6	--

**\*\* El estado inicial es F(raíz)**

	Estado	Valores	Siguientes
Inicio	<b>S0</b>	L,D,L,#,\$ 1,2,3,4,6	L: {1,2,3,6} = <b>S1</b> D: {1,2,3,6} = S1 #: {5} = <b>S2</b>
	<b>S1</b>	L,D,L,\$ 1,2,3,6	L: {1,2,3,6} = S1 <b>D</b> : {1,2,3,6} = S1



## 6. Construir Tabla de Transiciones

Valor	Hoja	Siguientes
L	1	1,2,3,6
D	2	1,2,3,6
L	3	6
#	4	5
L	5	6
\$	6	--

**\*\* El estado inicial es F(raíz)**

	Estado	Valores	Siguientes
Inicio	<b>S0</b>	L,D,L,#,\$ 1,2,3,4,6	L: {1,2,3,6} = <b>S1</b> D: {1,2,3,6} = S1 #: {5} = <b>S2</b>
	<b>S1</b>	L,D,L,\$ 1,2,3,6	L: {1,2,3,6} = S1 D: {1,2,3,6} = S1

## 6. Construir Tabla de Transiciones

Valor	Hoja	Siguientes
L	1	1,2,3,6
D	2	1,2,3,6
L	3	6
#	4	5
L	5	6
\$	6	--

**\*\* El estado inicial es F(raíz)**

	Estado	Valores	Siguientes
Inicio	<b>S0</b>	L,D,L,#,\$ 1,2,3,4,6	L: {1,2,3,6} = <b>S1</b> D: {1,2,3,6} = S1 #: {5} = <b>S2</b>
	<b>S1</b>	L,D,L,\$ 1,2,3,6	L: {1,2,3,6} = S1 D: {1,2,3,6} = S1
	<b>S2</b>	L 5	

## 6. Construir Tabla de Transiciones

Valor	Hoja	Siguientes
L	1	1,2,3,6
D	2	1,2,3,6
L	3	6
#	4	5
L	5	6
\$	6	--

**\*\* El estado inicial es F(raíz)**

	Estado	Valores	Siguientes
Inicio	<b>S0</b>	L,D,L,#,\$ 1,2,3,4,6	L: {1,2,3,6} = <b>S1</b> D: {1,2,3,6} = S1 #: {5} = <b>S2</b>
	<b>S1</b>	L,D,L,\$ 1,2,3,6	L: {1,2,3,6} = S1 D: {1,2,3,6} = S1
	<b>S2</b>	L 5	L: {6} = <b>S3</b>

## 6. Construir Tabla de Transiciones

Valor	Hoja	Siguientes
L	1	1,2,3,6
D	2	1,2,3,6
L	3	6
#	4	5
L	5	6
\$	6	--

**\*\* El estado inicial es F(raíz)**

	Estado	Valores	Siguientes
Inicio	S0	L,D,L,#,\$ 1,2,3,4,6	L: {1,2,3,6} = <b>S1</b> D: {1,2,3,6} = S1 #: {5} = <b>S2</b>
	S1	L,D,L,\$ 1,2,3,6	L: {1,2,3,6} = S1 D: {1,2,3,6} = S1
	S2	L 5	#: {6} = <b>S3</b>
	S3	\$ 6	--

## 6. Construir Tabla de Transiciones

Valor	Hoja	Siguientes
L	1	1,2,3,6
D	2	1,2,3,6
L	3	6
#	4	5
L	5	6
\$	6	--

**\*\* El estado inicial es F(raíz)**

	Estado	Valores	Siguientes
Inicio / Aceptación	<b>S0</b>	L,D,L,#,\$ 1,2,3,4,6	L: {1,2,3,6} = <b>S1</b> D: {1,2,3,6} = S1 #: {5} = <b>S2</b>
Aceptación	<b>S1</b>	L,D,L,\$ 1,2,3,6	L: {1,2,3,6} = S1 D: {1,2,3,6} = S1
	<b>S2</b>	L 5	L: {6} = <b>S3</b>
Aceptación	<b>S3</b>	\$ 6	--

## 6. Construir Tabla de Transiciones

	Estado	Valores	Siguientes
Inicio / Aceptación	<b>S0</b>	L,D,L,#,\$ 1,2,3,4,6	L: {1,2,3,6} = <b>S1</b> D: {1,2,3,6} = <b>S1</b> #: {5} = <b>S2</b>
Aceptación	<b>S1</b>	L,D,L,\$ 1,2,3,6	L: {1,2,3,6} = <b>S1</b> D: {1,2,3,6} = <b>S1</b>
	<b>S2</b>	L 5	L: {6} = <b>S3</b>
Aceptación	<b>S3</b>	\$ 6	--

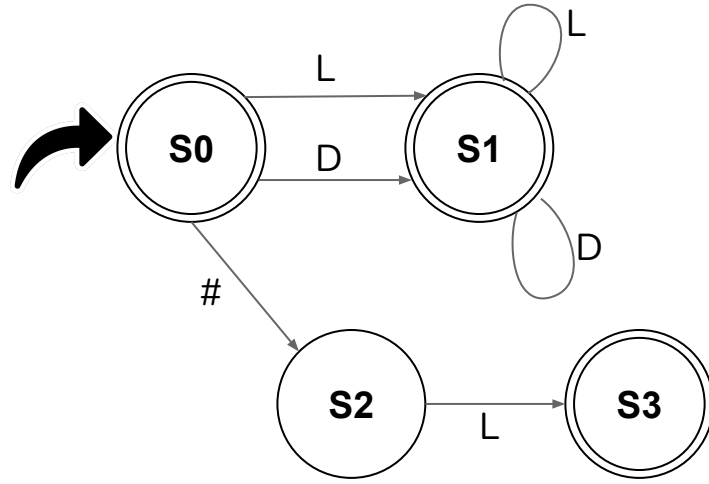
$((L|D)^*L?|' \# 'L) \$$

	Estados	$\Sigma$		
		L	D	#
o / \$	<b>S0</b>	S1	S1	S2
\$	<b>S1</b>	S1	S1	-
	<b>S2</b>	S3	-	-
\$	<b>S3</b>	-	-	-

## 7. Dibujar Diagrama de Estados

$((L|D)^*L?|'\# 'L)\$$

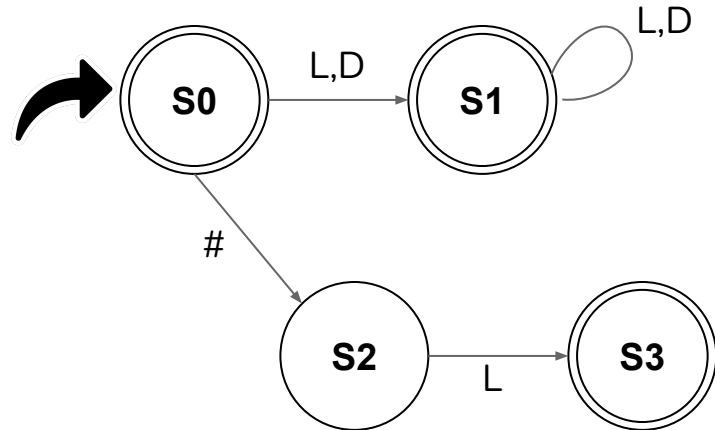
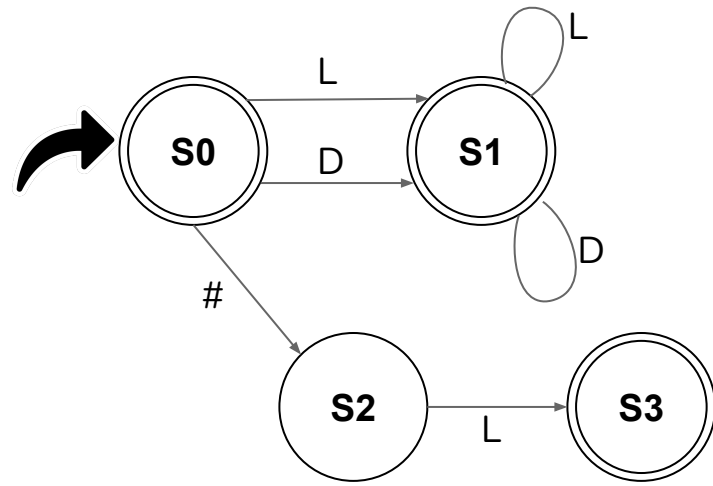
	Estados	$\Sigma$		
		L	D	#
o / \$	S0	S1	S1	S2
\$	S1	S1	S1	-
	S2	S3	-	-
\$	S3	-	-	-



## 7. Dibujar Diagrama de Estados

$((L|D)^*L?'#\text{'L})\$$

	Estados	$\Sigma$		
		L	D	#
o / \$	S0	S1	S1	S2
\$	S1	S1	S1	-
	S2	S3	-	-
\$	S3	-	-	-





# Resolución de Dudas del Proyecto 1 y HT 1



Nombre	Expresión Regular
Identificador o ID	$L (L \mid D \mid \_ )^*$
Entero	$D D^*$
.	.
.	.

Elementos:

$L = [A-Z, a-z, \tilde{N}, \tilde{n}]$

$D = [0-9]$

Agrupación de ER's para construir Expresión Regular Final

Nombre	Expresión Regular
Identificador o ID	$L(L D '_{ }')^*$
Entero	$D D^*$
.	.
.	.

Elementos:

$L = [A-Z, a-z, \tilde{N}, \tilde{n}]$

$D = [0-9]$

$(L(L|D|'_{ }')^*)|(D D^*)|\dots| \dots$

Agrupación de ER's para construir Expresión Regular Final

Nombre	Expresión Regular
Identificador o ID	$L(L D '_' )^*$
Entero	$D D^*$
.	.
.	.

Elementos:

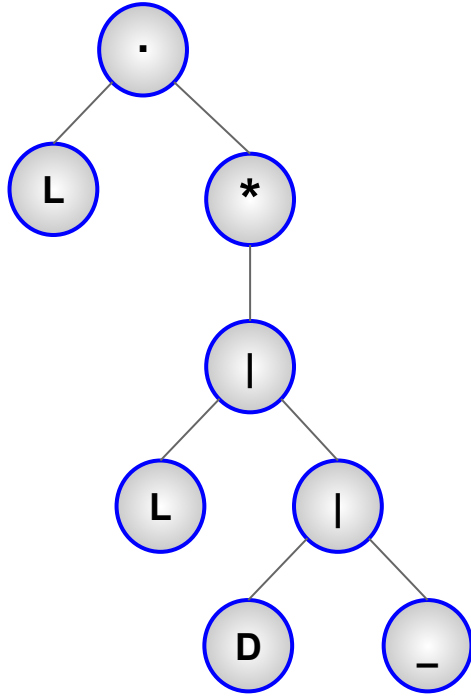
$L = [A-Z, a-z, \tilde{N}, \tilde{n}]$

$D = [0-9]$

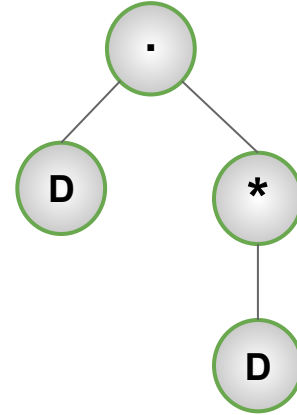
$(L(L|D|'_' )^*)|(D D^*)|\dots|\dots$

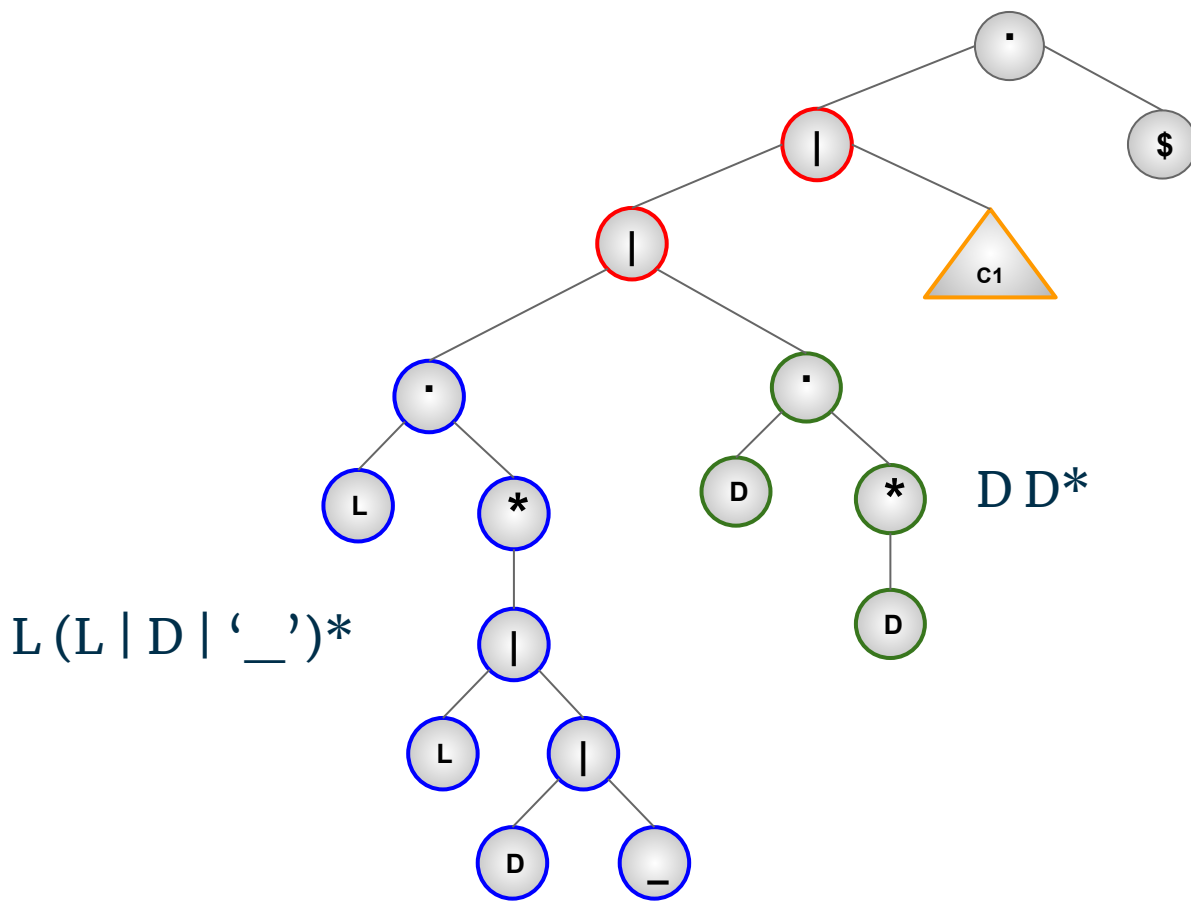
$((L(L|D|'_' )^*)|(D D^*)|\dots|\dots)\$$

$L(L \mid D \mid \text{'\_'} )^*$



$DD^* = D^+$





Agrupación de ER's para construir Expresión Regular Final

# Fin Clase 8

¿Dudas?