



Instituto Politecnico Nacional



ESCOM “ESCUELA SUPERIOR DE CÓMPUTO”

TEORÍA COMPUTACIONAL

PRÁCTICA 6: AMBIGÜEDAD EN GLC

PROFA: Luz María Sánchez García

ALUMMNO: Rojas Alvarado Luis Enrique

GRUPO: 2CM11

INTRODUCCIÓN

El propósito de la práctica consistió en que en base a un lenguaje generado a partir de las reglas dadas por la práctica se requiere realizar el árbol de derivación para generar distintas cadenas que posteriormente serán aceptadas o no aceptadas en la gramática y verificar tanto el tipo de gramática como si es una gramática ambigua.

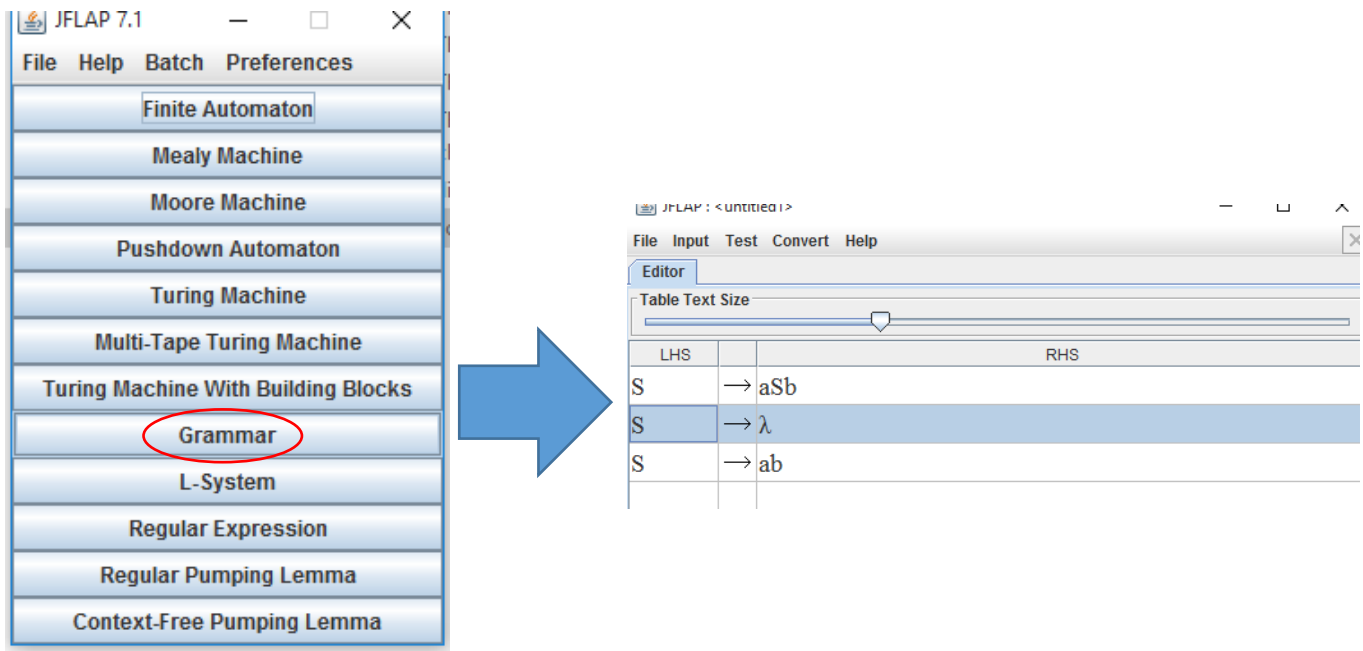
PLANTEAMIENTO DEL PROBLEMA

El problema que se presenta es que por medio de una gramática dada se tienen que ingresar 5 cadenas válidas y 5 cadenas no válidas, para posteriormente generar su árbol de derivación y la tabla de derivación para observar cómo se llega a la cadena ingresada.

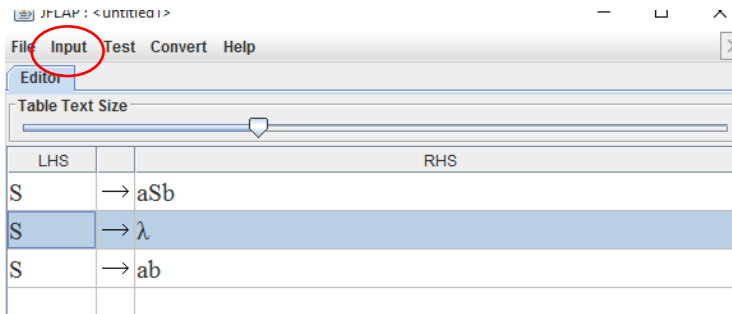
Para darle solución se optó por usar la herramienta de JFLAP ya que ofrece la opción para generar la gramática e ingresar las cadenas válidas y no válidas y posteriormente generar el árbol de derivación mostrando cómo se genera al final la cadena.

DISEÑO DE LA SOLUCIÓN

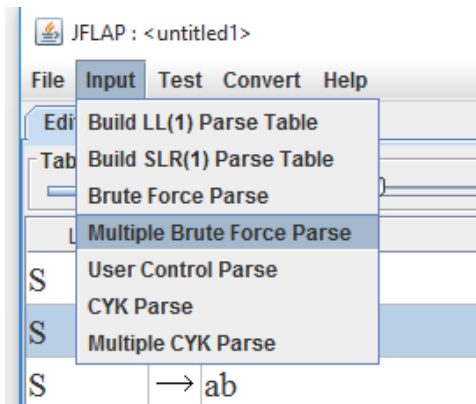
Como ya se comentó anteriormente se utilizó la herramienta JFLAP en la interfaz de gramática y posteriormente se hace el diseño de las reglas de la gramática.



Para ingresar las expresiones simplemente nos tenemos que ir a la pestaña de input.

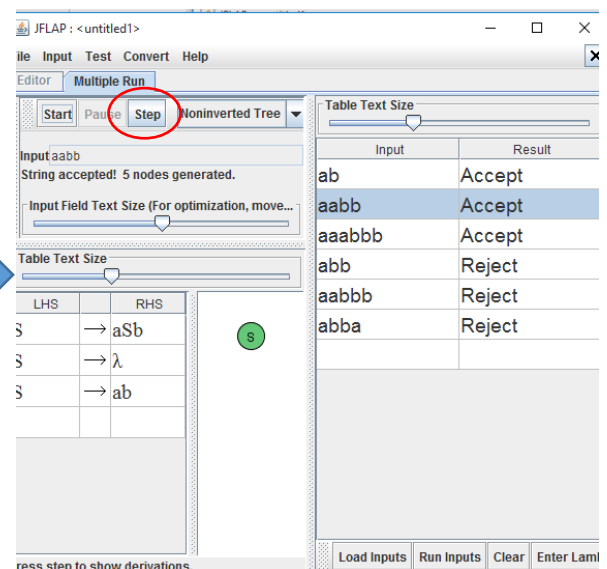
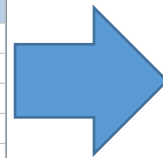
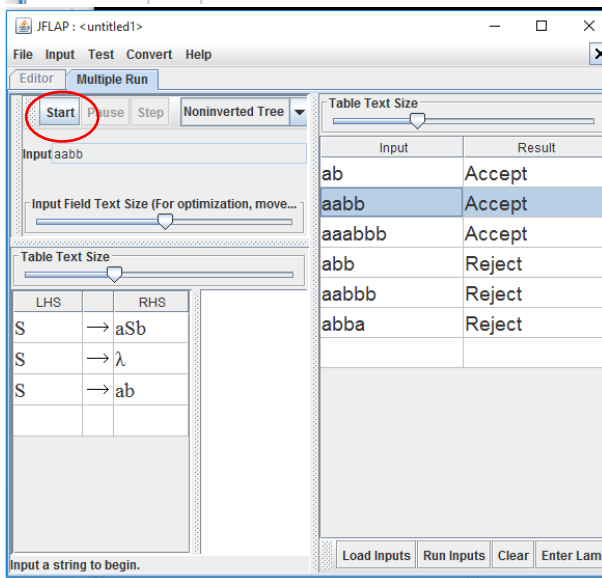


Una vez ahí se selecciona la opción de Multiple Brute Force Parse para agregar múltiples cadenas y verificar si son válidas o no válidas para la gramática declarada.



La pestaña generará una nueva ventana dentro de la aplicación para que se puedan ingresar las cadenas deseadas.

Para generar el árbol se selecciona la opción de start en la parte superior de la gramática y después seleccionar step hasta que el árbol esté completamente formado.



Una vez generado el árbol, para generar la tabla de derivación se requiere desplegar la pestaña que en este momento dice “Noninverted Tree” y seleccionar la opción “Derivation Table”

JFLAP : <untitled1>

File Input Test Convert Help

Editor Multiple Run

Start Pause Step **Noninverted Tree**

Input aabb
String accepted! 5 nodes generated.

Input Field Text Size (For optimization, move...)

Table Text Size

LHS	RHS
S	→ aSb
S	→ λ
S	→ ab

Derived ab from S. Derivations complete.

Table Text Size

Input	Result
ab	Accept
aabb	Accept
aaabbb	Accept
abb	Reject
aabbb	Reject
abba	Reject

Load Inputs Run Inputs Clear Enter Lamb

JFLAP : <untitled1>

File Input Test Convert Help

Editor Multiple Run

Start Pause Step **Derivation Table**

Input aabb
String accepted! 5 nodes generated.

Input Field Text Size (For optimization, move...)

Table Text Size

LHS	RHS
S	→ aSb
S	→ λ
S	→ ab

Table Text Size

Input	Result
ab	Accept
aabb	Accept
aaabbb	Accept
abb	Reject
aabbb	Reject
abba	Reject

Derived ab from S. Derivations complete.

Load Inputs Run Inputs Clear Enter Lamb

De ésta manera obtendremos tanto las cadenas válidas y no válidas así como el árbol de derivación y la tabla de derivación. Para así poder sacar una conclusión sobre el lenguaje que maneja la gramática, que tipo de expresión es y si es ambigua o no.

IMPLEMENTACIÓN DE LA SOLUCIÓN

1. $S \rightarrow aSb \mid ab \mid \epsilon$
2. $S \rightarrow SS^+ \mid SS^* \mid A$
 $A \rightarrow 0 \mid 1$
3. $S \rightarrow (S)S \mid \epsilon$
4. $S \rightarrow Aa \mid b$
 $A \rightarrow Ac \mid Sd \mid \epsilon$
5. $S \rightarrow aSbS \mid bSaS \mid \epsilon$
6. $S \rightarrow 0 \mid 1 \mid S+S \mid S^* \mid SS \mid (S)$
7. $S \rightarrow a \mid (L)$
 $L \rightarrow SL \mid \epsilon$
8. $S \rightarrow aSbb \mid abb \mid \epsilon$
9. $S \rightarrow AB$
 $A \rightarrow aAb \mid \epsilon$
 $B \rightarrow cBd \mid \epsilon$

10. $A \rightarrow a$
 $E \rightarrow b$
 $A \rightarrow azb$
 $A \rightarrow aX$
 $E \rightarrow E$
 $G \rightarrow g$
 $X \rightarrow XE$
 $D \rightarrow eI$
 $X \rightarrow z$
 $Y \rightarrow b$
 $I \rightarrow fG$
 $x \rightarrow Xb$
 $E \rightarrow d$
11. $S \rightarrow abS \mid Sab \mid aSb \mid SS \mid e$
12. $S \rightarrow aSbS \mid bSaS \mid e$

FUNCIONAMIENTO

- 1) $L(G1) = \{a^n b^n \mid n=m=0,1,2,\dots\}$ Tipo 2, No ambigua

The screenshot displays a software interface for a Noninverted Tree grammar. The interface is divided into several sections:

- Top Left:** A control panel with buttons for "Start", "Pause", and "Stop". Below these, it shows the input "ab" and a message "String accepted! 4 nodes generated." There is also an "Input Field Text Size" slider.
- Top Right:** A "Table Text Size" slider.
- Bottom Left:** A table showing the grammar rules (LHS and RHS):

LHS	RHS
S	$\rightarrow aSb$
S	$\rightarrow \lambda$
S	$\rightarrow ab$
- Bottom Center:** A derivation tree diagram showing a root node 'S' (green) with two children, 'a' (yellow) and 'b' (yellow).
- Bottom Right:** A table showing the results of the grammar's operation for various input strings:

Input	Result
ab	Accept
aabb	Accept
aaabbb	Accept
aaaabbbb	Accept
aaaaabbbbb	Accept
abaabaabb	Reject
abb	Reject
aabbb	Reject
abba	Reject
ababababa	Reject

At the bottom of the interface, there is a status bar that reads "Derived ab from S. Derivations complete."

2) $L(G_2) = \{01^*, 10+, 01^*+, 10^*+, 01+01^*\}$ Tipo 2, No ambigua

Start

Pause

Step

Noninverted Tree

Input 01*1+

String accepted! 54 nodes generated.

Input Field Text Size (For optimization, move one of the window size adjusters around this window after resizing L...

Table Text Size

LHS	RHS
S	→ SS+
S	→ SS*
S	→ A
A	→ 0
A	→ 1

Derived 1 from A. Derivations complete.

Start

Pause

Step

Derivation Table

Input 01*1+

String accepted! 54 nodes generated.

Input Field Text Size (For optimization, move one of the window size adjusters around this window after resizing L...

Table Text Size

LHS	RHS
S	→ SS+
S	→ SS*
S	→ A
A	→ 0
A	→ 1

Table Text Size

S	S
S→SS+	SS+
S→SS*	SS*S+
S→A	AS*S+
S→A	AA*S+
S→A	AA*A+
A→0	0A*A+
A→1	01*A+
A→1	01*1+

Derived 1 from A. Derivations complete.

Table Text Size

Input	Result
01*1+	Accept
10*0+	Accept
11+0*	Accept
01+01+*	Accept
11+11++	Accept
11*+11*	Reject
00+*01+	Reject
010*10*	Reject
1010+11*	Reject
10*10+	Reject

Load Inputs

Run Inputs

Clear

Enter Lambda

Table Text Size

Input	Result
01*1+	Accept
10*0+	Accept
11+0*	Accept
01+01+*	Accept
11+11++	Accept
11*+11*	Reject
00+*01+	Reject
010*10*	Reject
1010+11*	Reject
10*10+	Reject

Load Inputs

Run Inputs

Clear

Enter Lambda

3) $L(G3) = \{(), (()), ()(), (())(), \dots\}$ Tipo 2, No ambigua

Start **Pause** **Step** **Noninverted Tree**

Input: ()()()

String accepted! 64 nodes generated.

Input Field Text Size (For optimization, move one of the window size adjusters around this window after resizing t...)

Table Text Size

LHS	RHS
S	→ (S)S
S	→ λ

Table Text Size

Input	Result
()	Accept
(())	Accept
(())()	Accept
(())()	Accept
(())(())	Accept
(())	Reject
(())	Reject
(())(Reject
(())(()))(())	Reject
(())(())()	Reject

```

graph TD
    S1((S)) --- L1((L))
    S1 --- J1((J))
    S1 --- S2((S))
    L1 --- L2((L))
    L1 --- L3((L))
    L1 --- L4((L))
    L2 --- Lambda1((λ))
    L3 --- Lambda2((λ))
    J1 --- L5((L))
    J1 --- S3((S))
    J1 --- J2((J))
    S2 --- S4((S))
    S2 --- S5((S))
    S2 --- S6((S))
    S4 --- Lambda3((λ))
    S5 --- Lambda4((λ))
    S6 --- Lambda5((λ))
    S3 --- L6((L))
    S3 --- S7((S))
    S3 --- S8((S))
    S7 --- Lambda6((λ))
    S8 --- Lambda7((λ))
    S8 --- S9((S))
    S9 --- Lambda8((λ))
    
```

The diagram shows a parse tree for the input string "()()()". The root node is a green circle labeled 'S'. It has three children: a yellow circle 'L', a yellow circle 'J', and another green circle 'S'. The first 'L' has four children, all yellow circles labeled 'L'. The first two of these lead to yellow circles labeled 'λ'. The 'J' node has three children: a yellow circle 'L', a green circle 'S', and a yellow circle 'J'. This second 'S' node has three children: two green circles 'S' and one yellow circle 'J'. Each of these 'S' nodes has three children: two yellow circles 'L' leading to 'λ', and one green circle 'S' which leads to a final 'λ' at the bottom right.

Derived λ from S. Derivations complete.

Load Inputs
Run Inputs
Clear
Enter Lambda

StartPauseStepDerivation Table

Input: $()()()$
String accepted! 64 nodes generated.

Input Field Text Size (For optimization, move one of the window size adjusters around this window after resizing L...

Table Text Size

LHS		RHS
S	\rightarrow	$(S)S$
S	\rightarrow	λ

Table Text Size

	S
$S \rightarrow (S)S$	$(S)S$
$S \rightarrow (S)S$	$(S)(S)S$
$S \rightarrow (S)S$	$((S)S)(S)S$
$S \rightarrow (S)S$	$((S)S)((S)S)S$
$S \rightarrow \lambda$	$()(S)((S)S)S$
$S \rightarrow \lambda$	$()()((S)S)S$
$S \rightarrow \lambda$	$()()()((S)S)S$
$S \rightarrow \lambda$	$()()()()((S)S)S$
$S \rightarrow \lambda$	$()()()()()((S)S)S$

Derived λ from S. Derivations complete.

Table Text Size

Input	Result
$()$	Accept
$()()$	Accept
$()()()$	Accept
$()()()$	Accept
$()()()$	Accept
$()()()$	Accept
$()()$	Reject
$()()$	Reject
$()()$	Reject
$()()$	Reject
$()()$	Reject
$()((()))()()$	Reject
$()()()()$	Reject

Load InputsRun InputsClearEnter Lambda

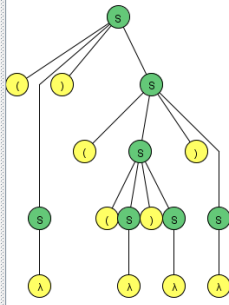
StartPauseStepNoninverted Tree

Input: $()()()$
String accepted! 26 nodes generated.

Input Field Text Size (For optimization, move one of the window size adjusters around this window after resizing L...

Table Text Size

LHS		RHS
S	\rightarrow	$(S)S$
S	\rightarrow	λ



Derived λ from S. Derivations complete.

Table Text Size

Input	Result
$()$	Accept
$()()$	Accept
$()()()$	Accept
$()()()$	Accept
$()()()$	Accept
$()()()$	Accept
$()()$	Reject
$()()$	Reject
$()()$	Reject
$()()$	Reject
$()()$	Reject
$()((()))()()$	Reject
$()()()()$	Reject

Load InputsRun InputsClearEnter Lambda

StartPauseStepDerivation Table

Input: $()()()$
String accepted! 26 nodes generated.

Input Field Text Size (For optimization, move one of the window size adjusters around this window after resizing L...

Table Text Size

LHS		RHS
S	\rightarrow	$(S)S$
S	\rightarrow	λ

Table Text Size

	S
$S \rightarrow (S)S$	$(S)S$
$S \rightarrow (S)S$	$(S)(S)S$
$S \rightarrow (S)S$	$(S)((S)S)S$
$S \rightarrow \lambda$	$()((S)S)S$
$S \rightarrow \lambda$	$()()((S)S)S$
$S \rightarrow \lambda$	$()()()((S)S)S$
$S \rightarrow \lambda$	$()()()()((S)S)S$

Derived λ from S. Derivations complete.

Table Text Size

Input	Result
$()$	Accept
$()()$	Accept
$()()()$	Accept
$()()()$	Accept
$()()()$	Accept
$()()()$	Accept
$()()$	Reject
$()()$	Reject
$()()$	Reject
$()()$	Reject
$()()$	Reject
$()((()))()()$	Reject
$()()()()$	Reject

Load InputsRun InputsClearEnter Lambda

StartPauseStepDerivation Table

Input bdca

String accepted! 6 nodes generated.

Input Field Text Size (For optimization, move one of the window size adjusters around this window after resizing L...

Table Text Size

LHS		RHS
S	→	Aa
S	→	b
A	→	Ac
A	→	Sd
A	→	λ

Table Text Size

S	→ Aa	Aa
A	→ Ac	Aca
A	→ Sd	Sdca
S	→ b	bdca

Table Text Size

Input	Result
b	Accept
bdca	Accept
bdada	Accept
cccca	Accept
a	Accept
ba	Reject
bbbbba	Reject
bdcdcdcdca	Reject
cccdada	Reject
bcccca	Reject

Load InputsRun InputsClearEnter Lambda

StartPauseStepNoninverted Tree

Input bdada

String accepted! 8 nodes generated.

Input Field Text Size (For optimization, move one of the window size adjusters around this window after resizing L...

Table Text Size

LHS		RHS
S	→	Aa
S	→	b
A	→	Ac
A	→	Sd
A	→	λ

Table Text Size

S	→ Aa	Aa
A	→ Ac	Aca
A	→ Sd	Sdca
S	→ b	bdca

Table Text Size

Input	Result
b	Accept
bdca	Accept
bdada	Accept
cccca	Accept
a	Accept
ba	Reject
bbbbba	Reject
bdcdcdcdca	Reject
cccdada	Reject
bcccca	Reject

Load InputsRun InputsClearEnter Lambda

StartPauseStepDerivation Table

Input bdada

String accepted! 8 nodes generated.

Input Field Text Size (For optimization, move one of the window size adjusters around this window after resizing L...

Table Text Size

LHS		RHS
S	→	Aa
S	→	b
A	→	Ac
A	→	Sd
A	→	λ

Table Text Size

S	→ Aa	Aa
A	→ Sd	Sda
S	→ Aa	Aada
A	→ Sd	Sdada
S	→ b	bdada

Table Text Size

Input	Result
b	Accept
bdca	Accept
bdada	Accept
cccca	Accept
a	Accept
ba	Reject
bbbbba	Reject
bdcdcdcdca	Reject
cccdada	Reject
bcccca	Reject

Load InputsRun InputsClearEnter Lambda

Start Pause Step Derivation Table

Input aaabbb
String accepted! 24 nodes generated.

Input Field Text Size (For optimization, move one of the window size adjusters around this window after resizing L...

Table Text Size

LHS	RHS
S	→ bSaS
S	→ λ
S	→ aSbS

Table Text Size

LHS	RHS
S	→ aSbS
S	→ aSbS
S	→ aSbS
S	→ λ
S	→ λ
S	→ λ
S	→ λ
S	→ λ

Derived λ from S. Derivations complete.

Table Text Size

Input	Result
abab	Accept
babaab	Accept
bbbaaa	Accept
aaabbb	Accept
baab	Accept
aba	Reject
babbabbb	Reject
aabbbbaa	Reject
abaaab	Reject
a	Reject

Load Inputs Run Inputs Clear Enter Lambda

Start Pause Step Noninverted Tree

Input bbbaaa
String accepted! 24 nodes generated.

Input Field Text Size (For optimization, move one of the window size adjusters around this window after resizing L...

Table Text Size

LHS	RHS
S	→ bSaS
S	→ λ
S	→ aSbS

Derived λ from S. Derivations complete.

Table Text Size

Input	Result
abab	Accept
babaab	Accept
bbbaaa	Accept
aaabbb	Accept
baab	Accept
aba	Reject
babbabbb	Reject
aabbbbaa	Reject
abaaab	Reject
a	Reject

Load Inputs Run Inputs Clear Enter Lambda

Start Pause Step Derivation Table

Input bbbaaa
String accepted! 24 nodes generated.

Input Field Text Size (For optimization, move one of the window size adjusters around this window after resizing L...

Table Text Size

LHS	RHS
S	→ bSaS
S	→ λ
S	→ aSbS

Table Text Size

LHS	RHS
S	→ bSaS
S	→ bSaS
S	→ bSaS
S	→ λ
S	→ λ
S	→ λ
S	→ λ
S	→ λ

Derived λ from S. Derivations complete.

Table Text Size

Input	Result
abab	Accept
babaab	Accept
bbbaaa	Accept
aaabbb	Accept
baab	Accept
aba	Reject
babbabbb	Reject
aabbbbaa	Reject
abaaab	Reject
a	Reject

Load Inputs Run Inputs Clear Enter Lambda

Start Pause Step Noninverted Tree

Input babaab
String accepted! 34 nodes generated.

Input Field Text Size (For optimization, move one of the window size adjusters around this window after resizing L...

Table Text Size

LHS	RHS
S	→ bSaS
S	→ λ
S	→ aSbS

Derived λ from S. Derivations complete.

Start Pause Step Derivation Table

Input babaab
String accepted! 34 nodes generated.

Input Field Text Size (For optimization, move one of the window size adjusters around this window after resizing L...

Table Text Size

LHS	RHS
S	→ bSaS
S	→ λ
S	→ aSbS

Table Text Size

S	S
S → bSaS	bSaS
S → aSbS	baSbSaS
S → aSbS	baSbSaaSbS
S → λ	babSaaSbS
S → λ	babaaSbS
S → λ	babaabS
S → λ	babaab

Derived λ from S. Derivations complete.

Table Text Size

Input	Result
abab	Accept
babaab	Accept
bbbbaa	Accept
aaabbb	Accept
baab	Accept
aba	Reject
babbabbb	Reject
aabbbbaa	Reject
abaaab	Reject
a	Reject

6) $L(G_6) = \{0, 1, 01, 011, 10, 1+0, (0+1), (1+0), (1+01^*)\}$ Tipo 2, Ambigua

Start Pause Step Noninverted Tree

Input 0+1(0)*
String accepted! 580 nodes generated.

Input Field Text Size (For optimization, move...

Table Text Size

LHS	RHS
S	→ (S)
S	→ SS
S	→ S*
S	→ S...
S	→ 0
S	→ 1

Derived 0 from S. Derivations complete.

Table Text Size

Input	Result
10	Accept
0	Accept
(1+0)	Accept
0+1(0)*	Accept
(1+01*)	Accept
10+*	Reject
(10+*)	Reject
+01*	Reject
(10)+*	Reject

Start Pause Step Noninverted Tree

Input $0+1(0)^*$
String accepted! 423 nodes generated.

Input Field Text Size (For optimization, move...)

Table Text Size

LHS	RHS
S	$\rightarrow 1$
S	$\rightarrow 0$
S	$\rightarrow S+S$
S	$\rightarrow S^*$
S	$\rightarrow SS$
S	$\rightarrow (S)$

Derived 0 from S. Derivations complete.

Input	Result
10	Accept
0	Accept
(1+0)	Accept
$0+1(0)^*$	Accept
$(1+01^*)$	Accept
$10+^*$	Reject
$(10+^*)$	Reject
$+01^*$	Reject
$(10)+^*$	Reject

Load Inputs Run Inputs Clear Enter Lamb

7) $L(G7)=\{(), a, (a), ((a)), ((a)(a)), \dots (), (()), \dots\}$ Tipo 2, No ambigua

Start Pause Step Noninverted Tree

Input $((()))$
String accepted! 44 nodes generated.

Input Field Text Size (For optimization, move...)

Table Text Size

LHS	RHS
S	$\rightarrow a$
S	$\rightarrow (L)$
L	$\rightarrow SL$
L	$\rightarrow \lambda$

Derived λ from L. Derivations complete.

Input	Result
a	Accept
(a)	Accept
()	Accept
$((a)(a))$	Accept
$((()))$	Accept
$(a)()$	Reject
$()(a)()$	Reject
$a(a)$	Reject
$(a())a$	Reject
$a(a)a()$	Reject

Load Inputs Run Inputs Clear Enter Lamb

Start Pause Step Derivation Table

Input(())
String accepted! 44 nodes generated.

Input Field Text Size (For optimization, move...

Table Text Size

LHS	RHS
S	→ a
S	→ (L)
L	→ SL
L	→ λ

Table Text Size

S	→ (L)
L	→ SL
L	→ SL
S	→ (L)
S	→ (L)
L	→ λ
L	→ λ
L	→ λ

Table Text Size

Input	Result
a	Accept
(a)	Accept
()	Accept
((a)(a))	Accept
(())	Accept
(a())	Reject
()(a())	Reject
a(a)	Reject
(a())a	Reject
a(a)a()	Reject

Load Inputs

Run Inputs

Clear

Enter Lamb

Start Pause Step Noninverted Tree

Input((a)(a))
String accepted! 273 nodes generated.

Input Field Text Size (For optimization, move...

Table Text Size

LHS	R...
S	→ a
S	→ (...)
L	→ SL
L	→ λ

Table Text Size

Table Text Size

Input	Result
a	Accept
(a)	Accept
()	Accept
((a)(a))	Accept
(())	Accept
(a())	Reject
()(a())	Reject
a(a)	Reject
(a())a	Reject
a(a)a()	Reject

Load Inputs

Run Inputs

Clear

Enter Lamb

Start Pause Step Derivation Table

Input ((a)(a))
String accepted! 273 nodes generated.

Input Field Text Size (For optimization, move...)

Table Text Size

LHS	R...
S	→ a
S	→ (...)
L	→ SL
L	→ λ

Table Text Size

LHS	RHS
S	→ (L)
L	→ SL
L	→ SL
S	→ (L)
S	→ (L)
L	→ SL
L	→ SL
S	→ a
L	→ λ
S	→ a
L	→ λ
L	→ λ
L	→ λ

Table Text Size

Input	Result
a	Accept
(a)	Accept
()	Accept
((a)(a))	Accept
((()))	Accept
(a())	Reject
()(a())	Reject
a(a)	Reject
(a())a	Reject
a(a)a()	Reject

Load Inputs Run Inputs Clear Enter Lamb

Derived λ from L. Derivations complete.

Start Pause Step Noninverted Tree

Input (a)
String accepted! 6 nodes generated.

Input Field Text Size (For optimization, move...)

Table Text Size

LHS	RHS
S	→ a
S	→ (L)
L	→ SL
L	→ λ

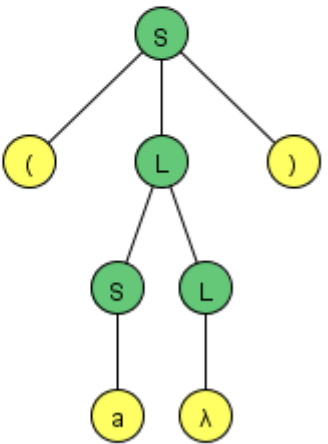


Table Text Size

Input	Result
a	Accept
(a)	Accept
()	Accept
((a)(a))	Accept
((()))	Accept
(a())	Reject
()(a())	Reject
a(a)	Reject
(a())a	Reject
a(a)a()	Reject

Load Inputs Run Inputs Clear Enter Lamb

Derived λ from L. Derivations complete.

Start Pause Step Derivation Table

Input (a)
String accepted! 6 nodes generated.

Input Field Text Size (For optimization, move...)

Table Text Size

LHS	RHS
S	→ a
S	→ (L)
L	→ SL
L	→ λ

Table Text Size

LHS	RHS
S	→ (L)
L	→ SL
S	→ a
L	→ λ

Table Text Size

Input	Result
a	Accept
(a)	Accept
()	Accept
((a)(a))	Accept
((0))	Accept
(a)()	Reject
()(a())	Reject
a(a)	Reject
(a())a	Reject
a(a)a()	Reject

Derived λ from L. Derivations complete.

Load Inputs Run Inputs Clear Enter Lamb

8) $L(G8) = \{a^n b^m \mid m=2n, m,n=0,1,2,\dots\}$ Tipo 2, No ambigua

Start Pause Step Noninverted Tree

Input abb
String accepted! 3 nodes generated.

Input Field Text Size (For optimization, move...)

Table Text Size

LHS	RHS
S	→ aSbb
S	→ abb
S	→ λ

Table Text Size

Input	Result
abb	Accept
aabbbb	Accept
aaabbbbb	Accept
aaaabbbbbbb	Accept
aaaaabbbbbbb	Accept
ab	Reject
abba	Reject
aaabbbbba	Reject
ababbb	Reject
abababbb	Reject

Derived abb from S. Derivations complete.

Load Inputs Run Inputs Clear Enter Lamb

```

graph TD
    S((S)) --- a((a))
    S --- b1((b))
    S --- b2((b))
  
```


Start Pause Step Derivation Table

Input **abb**
String accepted! 3 nodes generated.

Input Field Text Size (For optimization, move...

Table Text Size

LHS		RHS
S	→	aSbb
S	→	abb
S	→	λ

Table Text Size

S → abb	S
abb	

Table Text Size

Input	Result
abb	Accept
aabbbb	Accept
aaabbbbb	Accept
aaaabbbbbbb	Accept
aaaaabbbbbbbbb	Accept
ab	Reject
abba	Reject
aaabbbbbba	Reject
ababbb	Reject
abababbbb	Reject

Load Inputs

Run Inputs

Clear

Enter Lamb

Derived abb from S. Derivations complete.

Start Pause Step Noninverted Tree

Input **aabbbb**
String accepted! 5 nodes generated.

Input Field Text Size (For optimization, move...

Table Text Size

LHS		RHS
S	→	aSbb
S	→	abb
S	→	λ

Table Text Size

Input	Result
abb	Accept
aabbbb	Accept
aaabbbbb	Accept
aaaabbbbbbb	Accept
aaaaabbbbbbbbb	Accept
ab	Reject
abba	Reject
aaabbbbbba	Reject
ababbb	Reject
abababbbb	Reject

Load Inputs

Run Inputs

Clear

Enter Lamb

Derived abb from S. Derivations complete.

Start Pause Step Noninverted Tree

Input aaabbbbbb

String accepted! 6 nodes generated.

Input Field Text Size (For optimization, move...)

Table Text Size

LHS		RHS
S	→	aSbb
S	→	abb
S	→	λ

```

graph TD
    S1((S)) --- a1((a))
    S1 --- S2((S))
    S1 --- b1((b))
    S1 --- b2((b))
    S2 --- a2((a))
    S2 --- S3((S))
    S2 --- b3((b))
    S2 --- b4((b))
    S3 --- a3((a))
    S3 --- b5((b))
    S3 --- b6((b))
          
```

Input	Result
abb	Accept
aabbbb	Accept
aaabbbbbb	Accept
aaaabbbbbbbb	Accept
aaaaabbbbbbbbb	Accept
ab	Reject
abba	Reject
aaabbbbba	Reject
ababbb	Reject
abababbbb	Reject

Load Inputs

Run Inputs

Clear

Enter Lamb

Derived abb from S. Derivations complete.

Start Pause Step Derivation Table

Input aabbbb

String accepted! 5 nodes generated.

Input Field Text Size (For optimization, move...)

Table Text Size

LHS		RHS
S	→	aSbb
S	→	abb
S	→	λ

Table Text Size

	S
S → aSbb	aSbb
S → abb	aabbbb

Input	Result
abb	Accept
aabbbb	Accept
aaabbbbbb	Accept
aaaabbbbbbbb	Accept
aaaaabbbbbbbbb	Accept
ab	Reject
abba	Reject
aaabbbbba	Reject
ababbb	Reject
abababbbb	Reject

Load Inputs

Run Inputs

Clear

Enter Lamb

Derived abb from S. Derivations complete.

Start Pause Step Derivation Table

Input aaabbbbbb
String accepted! 6 nodes generated.

Input Field Text Size (For optimization, move...

Table Text Size

LHS		RHS
S	→	aSbb
S	→	abb
S	→	λ

Table Text Size

S→aSbb	S
S→aSbb	aSbb
S→abb	aaSbbbb
	aaSbbbbbb

Table Text Size

Input	Result
abb	Accept
aabbbb	Accept
aaabbbbbb	Accept
aaaabbbbbbbb	Accept
aaaaabbbbbbbbb	Accept
ab	Reject
abba	Reject
aaabbbbbba	Reject
ababbb	Reject
abababbbb	Reject

Load Inputs

Run Inputs

Clear

Enter Lamb

Derived abb from S. Derivations complete.

9) $L(G_9) = \{ a^n b^m c^i d^j \mid n=m=i=j=0,1,2,\dots \}$ Tipo 2, No ambigua

Start Pause Step Noninverted Tree

Input abcd
String accepted! 14 nodes generated.

Input Field Text Size (For optimization, move one of the window size adjustors around this window after resizing t...

Table Text Size

LHS		RHS
S	→	AB
A	→	aAb
A	→	λ
B	→	cBd
B	→	λ

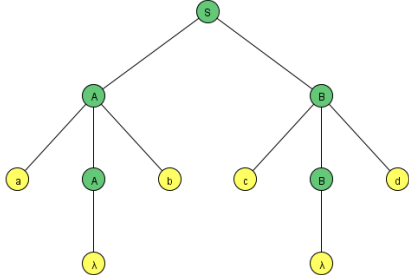


Table Text Size

Input	Result
abcd	Accept
aabbccdd	Accept
aaabbbccdd	Accept
aaaabbbccdd	Accept
aaaaabbbccdd	Accept
abcb	Reject
abcb	Reject
aabbccdaabb	Reject
bbaaccdd	Reject
ddccbaa	Reject

Load Inputs

Run Inputs

Clear

Enter Lambda

Derived A from B. Derivations complete.

StartPauseStepDerivation Table

Input abcd
String accepted! 14 nodes generated.

Input Field Text Size (For optimization, move one of the window size adjusters around this window after resizing L...

Table Text Size

LHS	RHS
S	→ AB
A	→ aAb
A	→ λ
B	→ cBd
B	→ λ

Table Text Size	
S→AB	S
A→aAb	AB
B→cBd	aAbB
A→λ	aAbcBd
B→λ	abcBd
	abcd

Derived A from B. Derivations complete.

StartPauseStepNoninverted Tree

Input aabbccdd
String accepted! 21 nodes generated.

Input Field Text Size (For optimization, move one of the window size adjusters around this window after resizing L...

Table Text Size

LHS	RHS
S	→ AB
A	→ aAb
A	→ λ
B	→ cBd
B	→ λ

```
graph TD; S((S)) --> A1((A)); S --> B1((B)); A1 --> a1((a)); A1 --> A2((A)); A2 --> a2((a)); A2 --> lambda1((λ)); B1 --> c1((c)); B1 --> B2((B)); B2 --> c2((c)); B2 --> lambda2((λ));
```

Derived A from B. Derivations complete.

StartPauseStepDerivation Table

Input aabbccdd
String accepted! 21 nodes generated.

Input Field Text Size (For optimization, move one of the window size adjusters around this window after resizing L...

Table Text Size

LHS	RHS
S	→ AB
A	→ aAb
A	→ λ
B	→ cBd
B	→ λ

Table Text Size	
S→AB	S
A→aAb	AB
B→cBd	aAbB
A→aAb	aaAbcBd
B→cBd	aaAbbccBdd
A→λ	aabbccBdd
B→λ	aabbccdd

Derived A from B. Derivations complete.

Table Text Size

Input	Result
abcd	Accept
aabbccdd	Accept
aaabbbccdd	Accept
aaaabbbcccccdd	Accept
aaaaabbbccccccdd	Accept
aabcd	Reject
abcda	Reject
aabbccddaabb	Reject
bbaaccdd	Reject
ddccbaa	Reject

Load InputsRun InputsClearEnter Lambda

Table Text Size

Input	Result
abcd	Accept
aabbccdd	Accept
aaabbbccdd	Accept
aaaabbbcccccdd	Accept
aaaaabbbccccccdd	Accept
aabcd	Reject
abcda	Reject
aabbccddaabb	Reject
bbaaccdd	Reject
ddccbaa	Reject

Load InputsRun InputsClearEnter Lambda

Table Text Size

Input	Result
abcd	Accept
aabbccdd	Accept
aaabbbccdd	Accept
aaaabbbcccccdd	Accept
aaaaabbbccccccdd	Accept
aabcd	Reject
abcda	Reject
aabbccddaabb	Reject
bbaaccdd	Reject
ddccbaa	Reject

Load InputsRun InputsClearEnter Lambda

Start Pause Step Noninverted Tree

Input aaabbbccdd

String accepted! 30 nodes generated.

Input Field Text Size (For optimization, move one of the window size adjusters around this window after resizing L...

Table Text Size

LHS	RHS
S	→ AB
A	→ aAb
A	→ λ
B	→ cBd
B	→ λ

Derived A from B. Derivations complete.

Start Pause Step Derivation Table

Input aaabbbccdd

String accepted! 30 nodes generated.

Input Field Text Size (For optimization, move one of the window size adjusters around this window after resizing L...

Table Text Size

LHS	RHS
S	→ AB
A	→ aAb
A	→ λ
B	→ cBd
B	→ λ

Table Text Size	S
S→AB	AB
A→aAb	aAbB
B→cBd	aAbcBd
A→aAb	aaAbbcBd
B→cBd	aaAbbccBdd
A→aAb	aaaAbbccBdd
B→cBd	aaaAbbccBddd
A→λ	aaabbbcccBddd
B→λ	aaabbbcccddd

Derived A from B. Derivations complete.

Table Text Size

Input	Result
abcd	Accept
aabbccdd	Accept
aaabbbccdd	Accept
aaaabbbcccccddd	Accept
aaaaabbbcccccddd	Accept
aabcd	Reject
abcda	Reject
aabbccddaabb	Reject
bbaaccdd	Reject
ddccbaa	Reject

Load Inputs Run Inputs Clear Enter Lambda

10) $L(G_{10}) = \{a, azb^n, azd^n, azb^n d^m, azb^n d^m b^n, \dots \mid n \text{ diferente de } m > 0\}$ Tipo 2, Ambigua

Start Pause Step Noninverted Tree

Input azbdd

String rejected. 0 nodes generated.

Input Field Text Size (For optimization, move...

Table Text Size

LHS	RHS
A	→ a
E	→ b
A	→ azb
A	→ aX
E	→ E
G	→ g
X	→ XE
D	→ eI

Try another string.

Table Text Size

Input	Result
azd	Accept
azbdd	Accept
azbbb	Accept
azddd	Accept
azb	Accept
azbg	Reject
azefg	Reject
abzd	Reject
azbdddzd	Reject
azbddbdg	Reject
azbdbdbdbb	Accept

Load Inputs Run Inputs Clear Enter Lamb

11) $L(G_{11}) = \{(ab)^n, a^n b^m\}$ Tipo 2, Ambigua

Start Pause Step Noninverted Tree

Input: abab
String accepted! 284 nodes generated.

Input Field Text Size (For optimization, move...)

Table Text Size

Input	Result
ab	Accept
abab	Accept
ababab	Accept
abababab	Accept

Load Inputs Run Inputs Clear Enter Lamb

Derived λ from S. Derivations complete.

Start Pause Step Noninverted Tree

Input: ababab
String accepted! 4334 nodes generated.

Input Field Text Size (For optimization, move...)

Table Text Size

Input	Result
ab	Accept
abab	Accept
ababab	Accept
abababab	Accept

Load Inputs Run Inputs Clear Enter Lamb

Derived λ from S. Derivations complete.

Start Pause Step Noninverted Tree

Input ababab

String accepted! 657 nodes generated.

Input Field Text Size (For optimization, move...)

Table Text Size

S

S

S

S

S

Derived λ from S. Derivations complete.

Table Text Size

Input	Result
ab	Accept
abab	Accept
ababab	Accept
abababab	Accept

Load Inputs

Run Inputs

Clear

Enter Lamb

Start Pause Step Noninverted Tree

Input abab

String accepted! 37 nodes generated.

Input Field Text Size (For optimization, move...)

Table Text Size

S

S

S

S

S

Derived λ from S. Derivations complete.

Table Text Size

Input	Result
ab	Accept
abab	Accept
ababab	Accept
abababab	Accept

Load Inputs

Run Inputs

Clear

Enter Lamb

12) $L(G_{12}) = \{a^n b^b, (ab)^n\}$ Tipo 2, No ambigua

Start Pause Step Noninverted Tree

Input abab
String accepted! 16 nodes generated.

Input Field Text Size (For optimization, move one of the window size adjustors around this window after resizing L...

Table Text Size

LHS	RHS
S	$\rightarrow aSbS$
S	$\rightarrow bSaS$
S	$\rightarrow \lambda$

Derived λ from S. Derivations complete.

Start Pause Step Derivation Table

Input abab
String accepted! 16 nodes generated.

Input Field Text Size (For optimization, move one of the window size adjustors around this window after resizing L...

Table Text Size

LHS	RHS
S	$\rightarrow aSbS$
S	$\rightarrow bSaS$
S	$\rightarrow \lambda$

Table Text Size	
S	
S $\rightarrow aSbS$	aSbS
S $\rightarrow aSbS$	aSbaSbS
S $\rightarrow \lambda$	abaSbS
S $\rightarrow \lambda$	ababS
S $\rightarrow \lambda$	abab

Derived λ from S. Derivations complete.

Table Text Size

Input	Result
ab	Accept
abab	Accept
aabb	Accept
ababaabb	Accept
abaabbab	Accept
aba	Reject
ababb	Reject
abaaa	Reject
abbabb	Reject
aabbba	Reject

Load Inputs Run Inputs Clear Enter Lambda

Start Pause Step Noninverted Tree

Input aabb
String accepted! 12 nodes generated.

Input Field Text Size (For optimization, move one of the window size adjusters around this window after resizing L...

Table Text Size

LHS	RHS
S	→ aSbS
S	→ bSaS
S	→ λ

Derived λ from S. Derivations complete.

Start Pause Step Derivation Table

Input aabb
String accepted! 12 nodes generated.

Input Field Text Size (For optimization, move one of the window size adjusters around this window after resizing L...

Table Text Size

LHS	RHS
S	→ aSbS
S	→ bSaS
S	→ λ

Table Text Size

S	aSbS
S	aaSbSbS
S	aabSbS
S	aabbS
S	aabb

Derived λ from S. Derivations complete.

Table Text Size

Input	Result
ab	Accept
abab	Accept
aabb	Accept
ababaabb	Accept
abaabbab	Accept
aba	Reject
ababb	Reject
abaaa	Reject
abbabb	Reject
aabbba	Reject

Load Inputs Run Inputs Clear Enter Lambda

Start Pause Step Noninverted Tree

Input ababaabb
String accepted! 91 nodes generated.

Input Field Text Size (For optimization, move one of the window size adjusters around this window after resizing L...

Table Text Size

LHS	RHS
S	→ aSbS
S	→ bSaS
S	→ λ

Derived λ from S. Derivations complete.

Table Text Size

Input	Result
ab	Accept
abab	Accept
aabb	Accept
ababaabb	Accept
abaabbab	Accept
aba	Reject
ababb	Reject
abaaa	Reject
abbabb	Reject
aabbba	Reject

Load Inputs Run Inputs Clear Enter Lambda

Start Pause Step Derivation Table

Input ababaabb
String accepted! 91 nodes generated.
Input Field Text Size (For optimization, move one of the window size adjusters around this window after resizing L...)

Table Text Size

LHS	RHS
S	→ aSbS
S	→ bSaS
S	→ λ

Table Text Size

S→aSbS	aSbS
S→aSbS	aSbaSbS
S→bSaS	abSaSbaSbS
S→aSbS	abSaSbaaSbSbS
S→λ	abaSbaaSbSbS
S→λ	ababaaSbSbS
S→λ	ababaabSbS
S→λ	ababaabbS
S→λ	ababaabb

Table Text Size

Input	Result
ab	Accept
abab	Accept
aabb	Accept
ababaabb	Accept
abaabbab	Accept
aba	Reject
ababb	Reject
abaaa	Reject
abbabb	Reject
aabbaa	Reject

Load Inputs Run Inputs Clear Enter Lambda

Derived λ from S. Derivations complete.

CONCLUSIÓN

En ésta práctica se realizaron ejercicios con diferentes gramáticas para probar su ambigüedad y definir un lenguaje para estas gramáticas, asignando cadenas válidas y no válidas y comprobar su árbol de derivación para dichas cadenas.

Personalmente ésta práctica fue muy laboriosa porque fueron muchos ejercicios y muchas cadenas a probar, y a mi parecer el propósito de esto es que al momento de hacer cualquier ejercicio nos demos cuenta con facilidad de cada uno de éstos elementos y saberlos identificar con más facilidad en cualquier ejercicio.

BIBLIOGRAFÍA

- 1) http://webdiis.unizar.es/asignaturas/LGA/material_2003_2004/4_LIC_GIC_A_DPND.pdf
- 2) <http://teodelacomp.blogspot.com/2011/03/arboles-de-derivacion.html>
- 3) <http://ri.uaemex.mx/oca/view/20.500.11799/34043/1/secme-16209.pdf>