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 cómo si es una gramática ambigua.

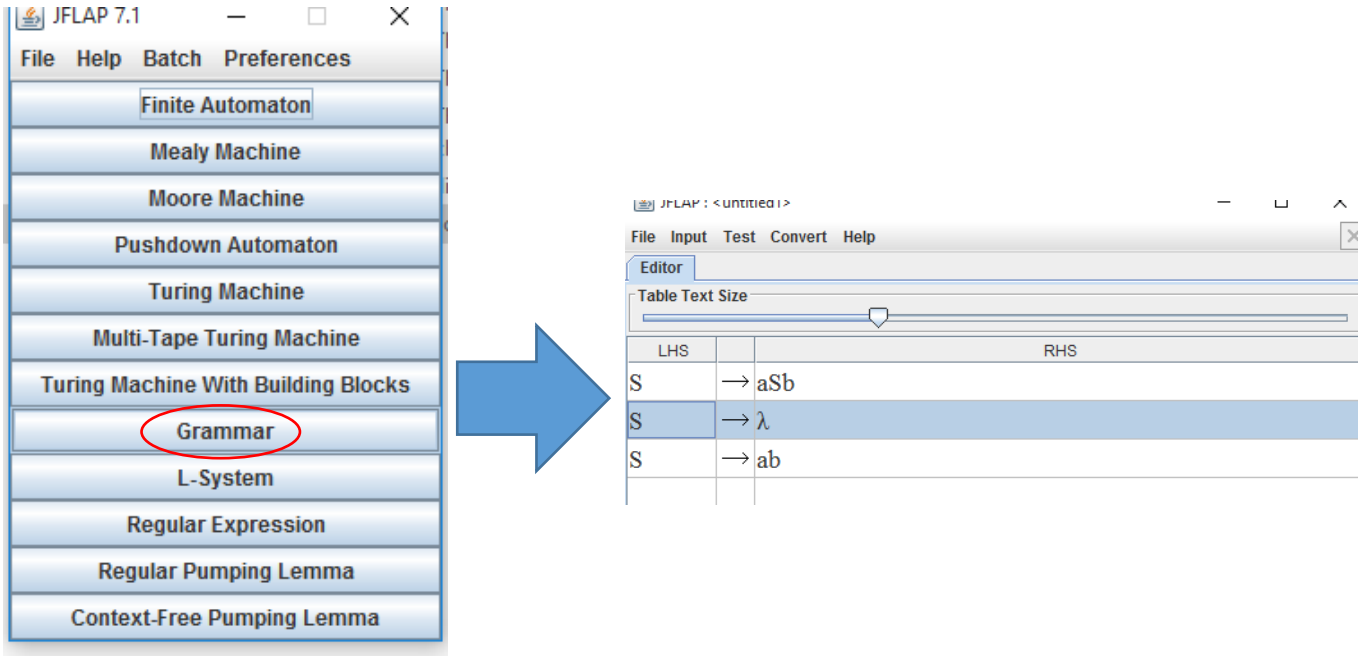
PLANTEAMIENTO DEL PROBLEMA

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.

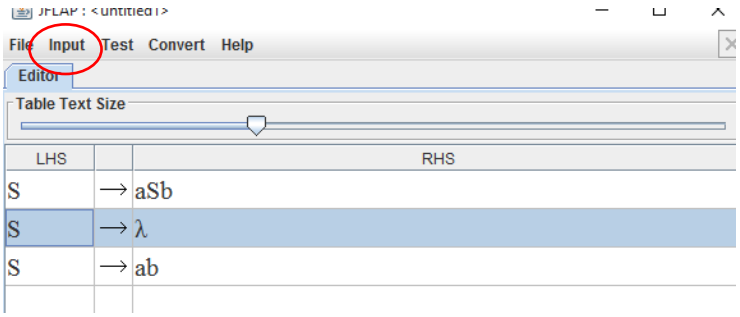
Para darle solución se optó por usar la herramienta de JFLAP ya que ofrece la opción para generar la gramática e ingresar las cadenas válidas y no válidas y posteriormente generar el árbol de derivación mostrando cómo se genera al final la cadena.

DISEÑO DE LA SOLUCIÓN

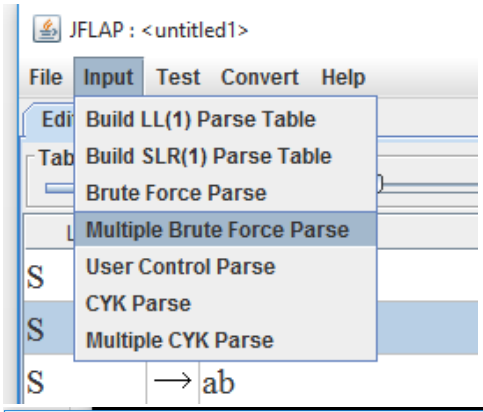
Como ya se comentó anteriormente se utilizó la herramienta JFLAP en la interfaz de gramática y posteriormente se hace el diseño de las reglas de la gramática.



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

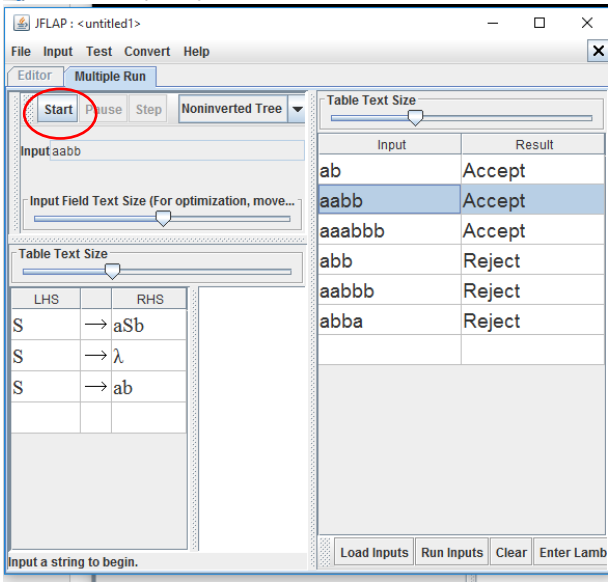


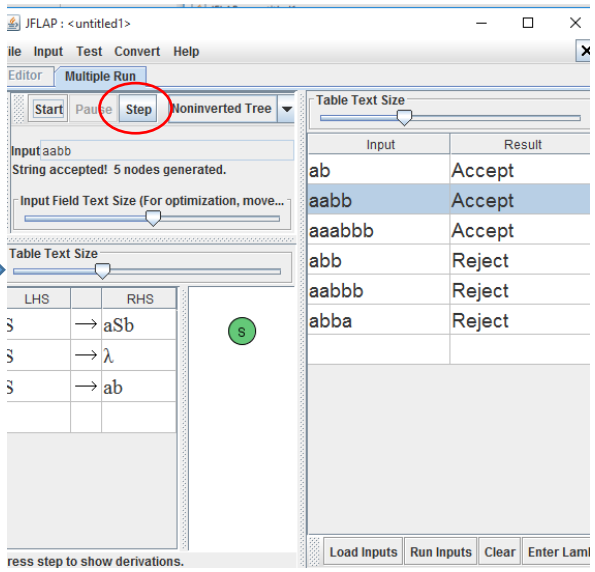
Una vez ahí se selecciona la opción de Multiple Brute Force Parse para agregar múltiples cadenas y verificar si son válidas o no válidas para la gramática declarada.



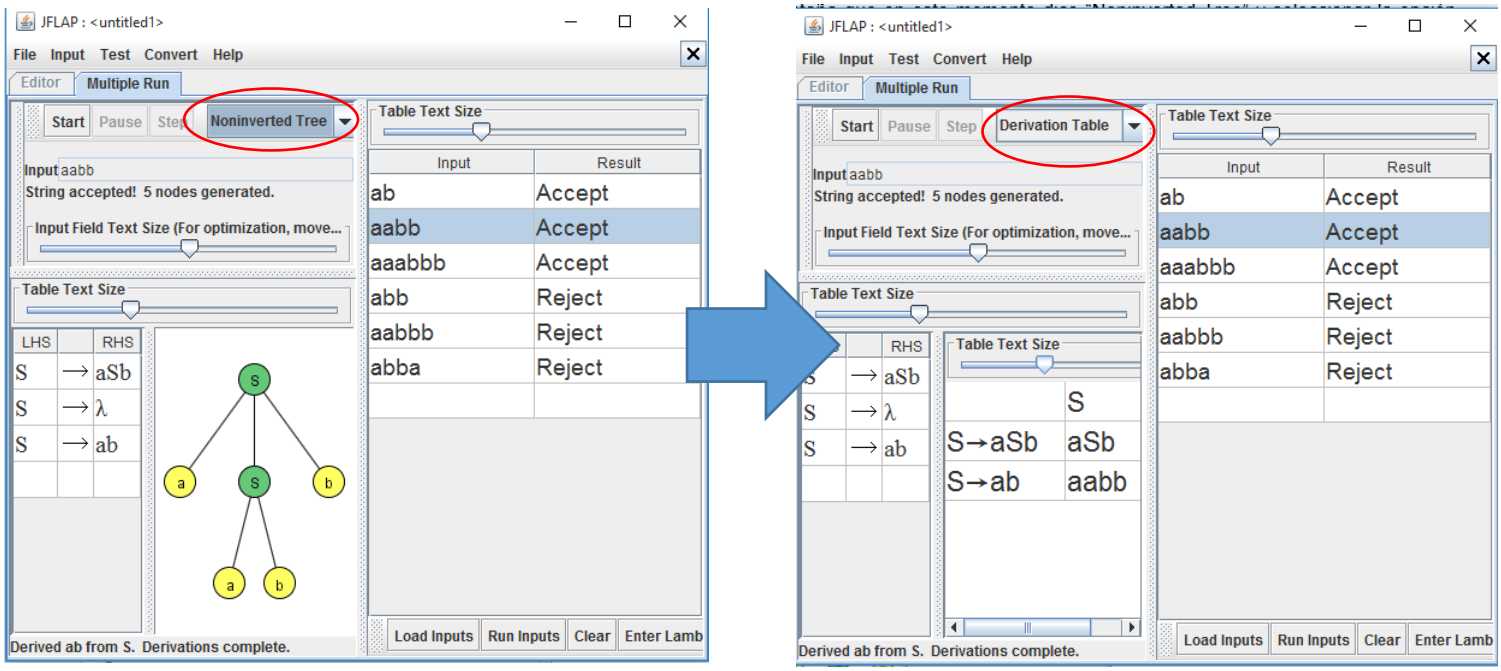
La pestaña generará una nueva ventana dentro de la aplicación para que se puedan ingresar las cadenas deseadas.

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”



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.

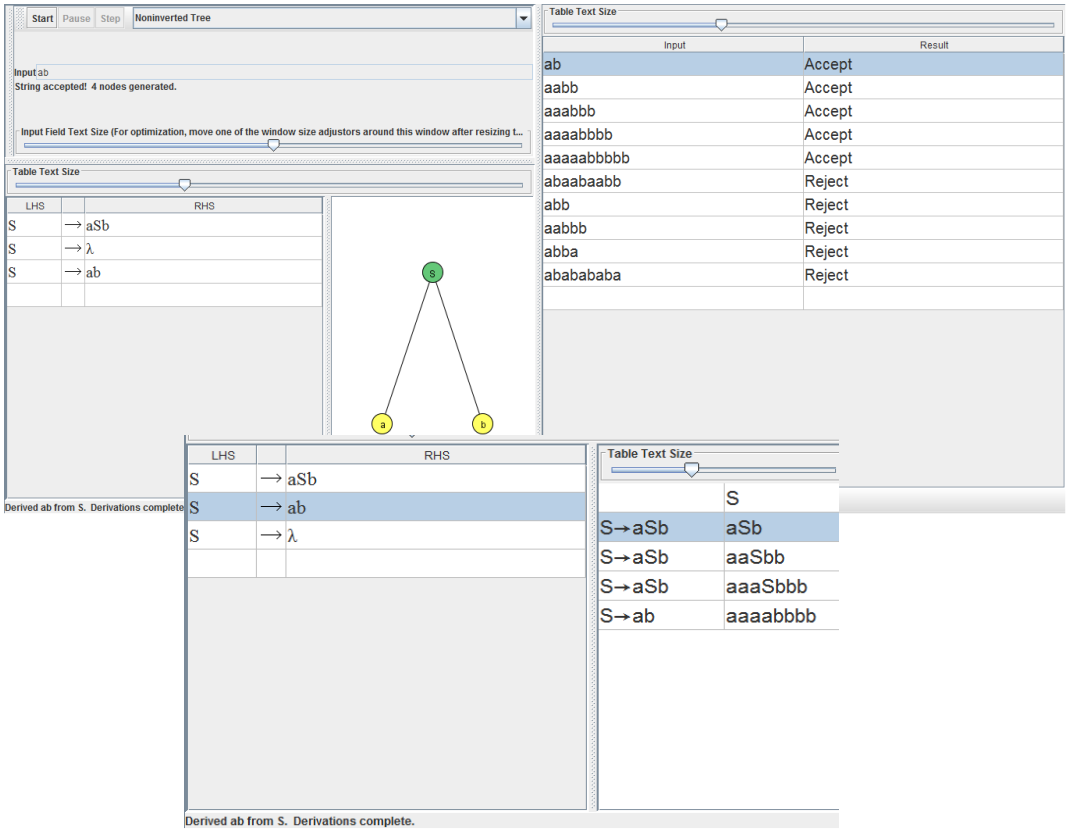
IMPLEMENTACIÓN DE LA SOLUCIÓN

1. S->aSb | ab | e
2. S->SS+ | SS* |A
   A->0 | 1
3. S->(S)S | e
4. S->Aa | b
   A->Ac | Sd | e
5. S->aSbS | bSaS | e
6. S->0 | 1 |S+S | S* |SS | (S)
7. S->a | (L)
   L->SL | e
8. S->aSbb | abb | e
9. S->AB
   A->aAb | e
   B->cBd | e

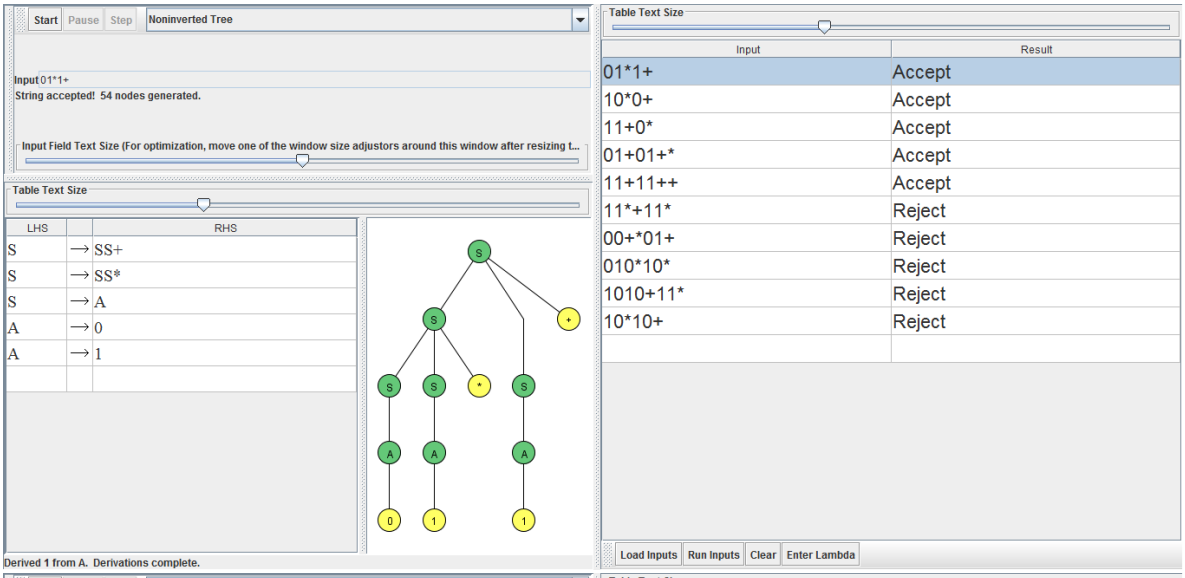
10. A->a
    E->b
    A->azb
    A->aX
    E->E
    G->g
    X->XE
    D->eI
    X->z
    Y->b
    I->fG
    x->Xb
    E->d
11. S->abS | Sab | aSb | SS | e
12. S->aSbS | bSaS | e

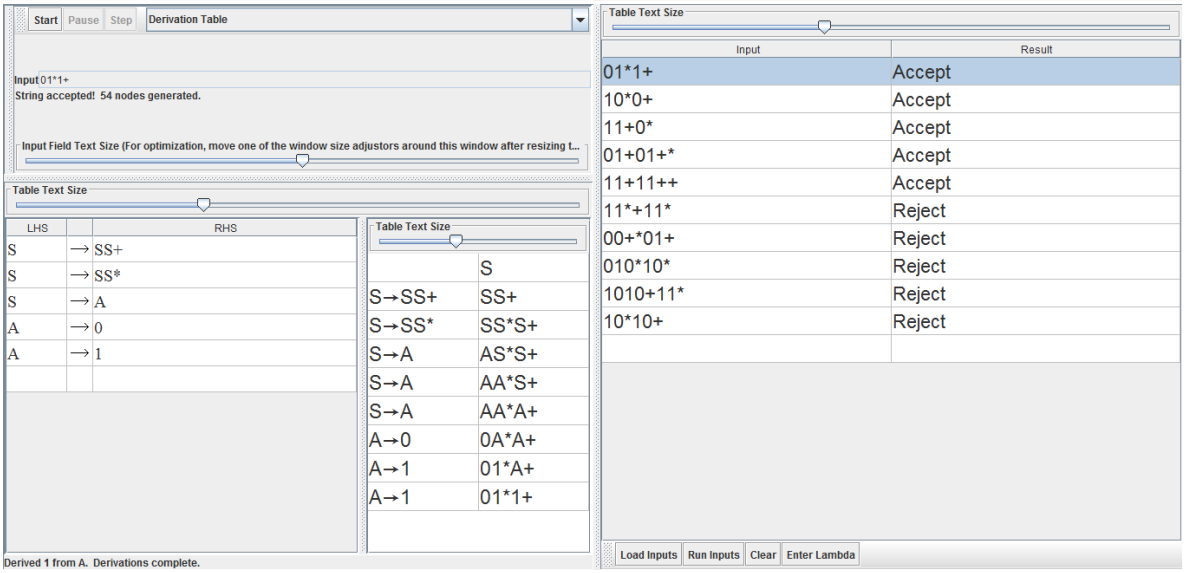
FUNCIONAMIENTO

1) L(G1)={$a^n b^n$ | n=m=0,1,2,…} Tipo 2, No ambigua

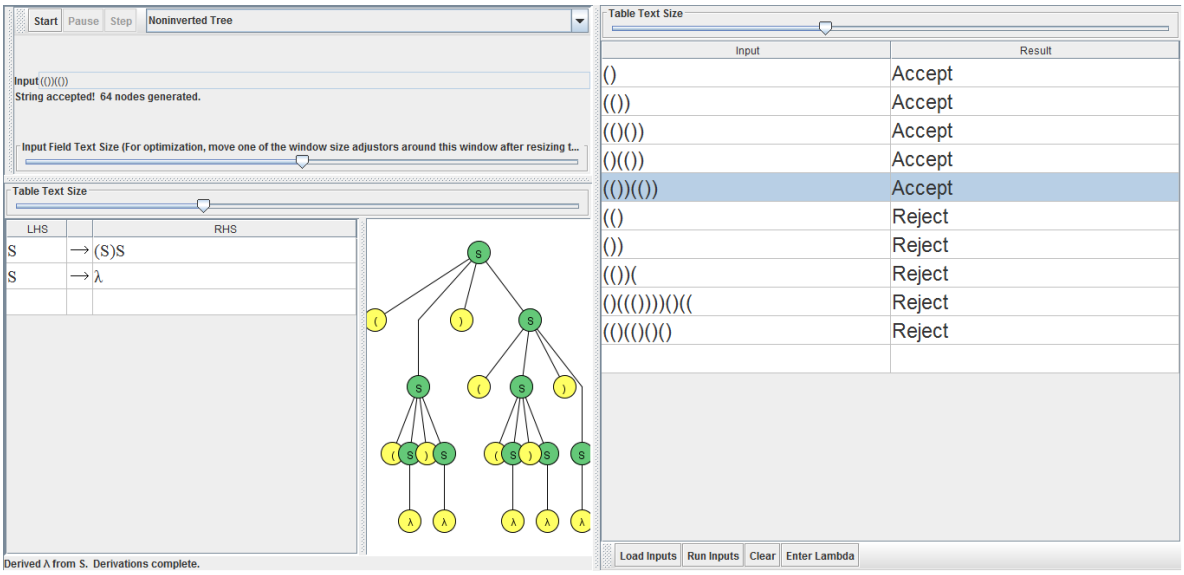


2) L(G2)={01*,10+,01*+,10+*,01+01*} Tipo 2, No ambigua





3) L(G3)={(),(()),()(),(())(()),…} Tipo 2, No ambigua



Start | Pause | Step | Derivation Table

Input ()(())

String accepted! 64 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	→	(S)S
S	→	λ

Table Text Size

	S
S→(S)S	(S)S
S→(S)S	(S)(S)S
S→(S)S	((S)S)(S)S
S→(S)S	((S)S)((S)S)S
S→λ	(()S)((S)S)S
S→λ	(())((S)S)S
S→λ	(())(()S)S
S→λ	(())(())S
S→λ	(())(())

Derived λ from S. Derivations complete.

Table Text Size

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

Load Inputs | Run Inputs | Clear | Enter Lambda

Start | Pause | Step | Noninverted Tree

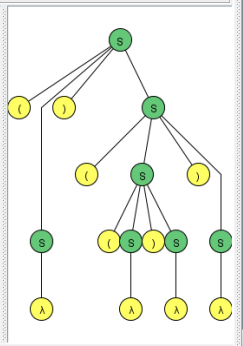
Input ()(())

String accepted! 26 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	→	(S)S
S	→	λ



Derived λ from S. Derivations complete.

Table Text Size

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

Load Inputs | Run Inputs | Clear | Enter Lambda

Start | Pause | Step | Derivation Table

Input ()(())

String accepted! 26 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	→	(S)S
S	→	λ

Table Text Size

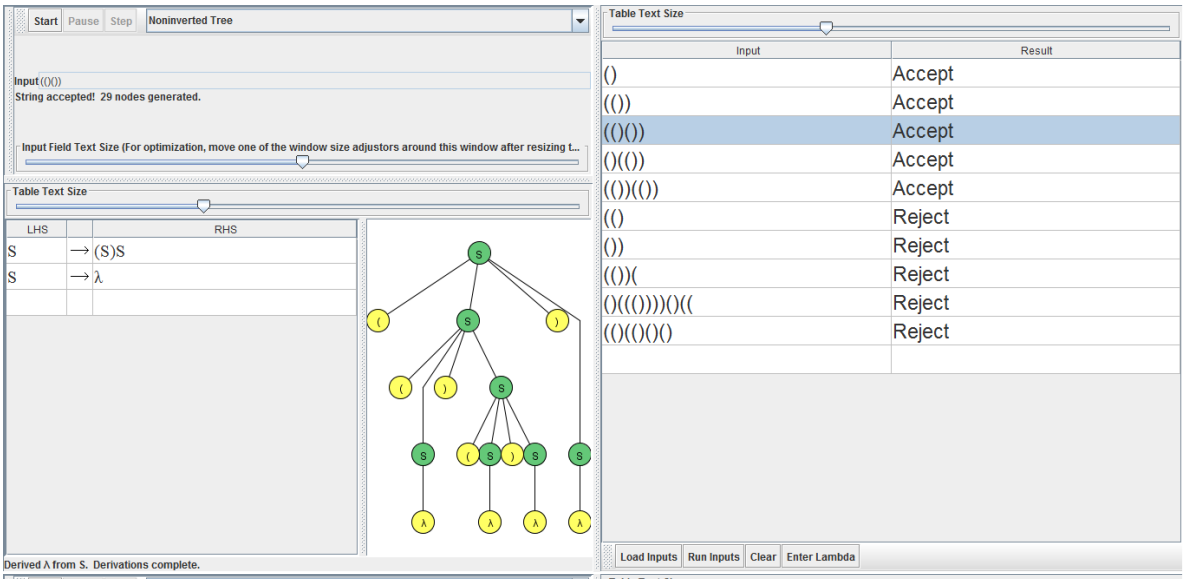
	S
S→(S)S	(S)S
S→(S)S	(S)(S)S
S→(S)S	(S)((S)S)S
S→λ	()((S)S)S
S→λ	()(()S)S
S→λ	()(())S
S→λ	()(())

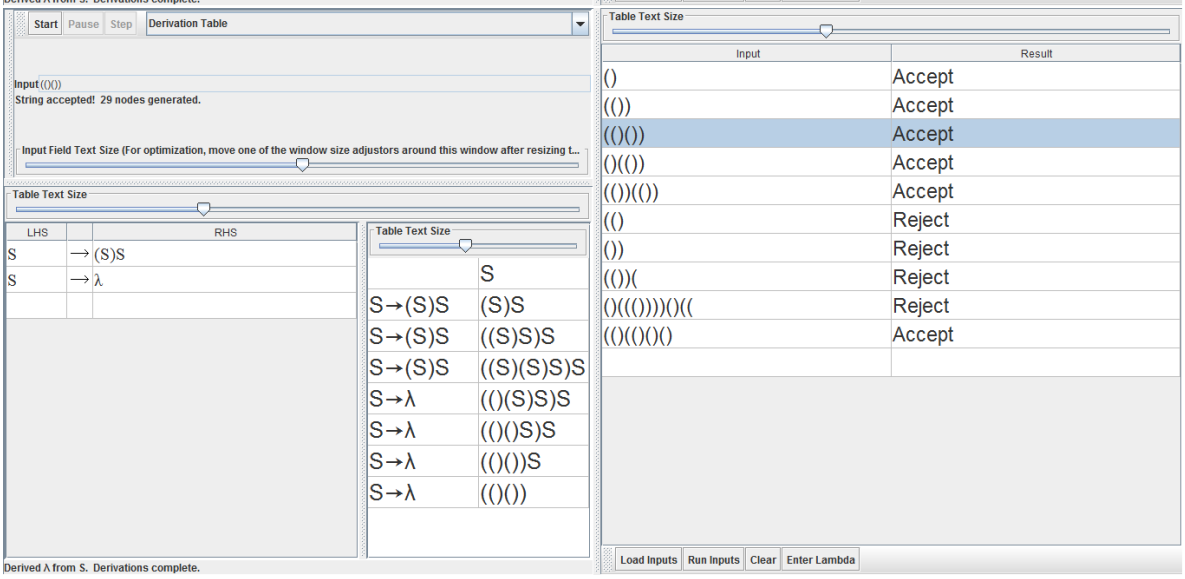
Derived λ from S. Derivations complete.

Table Text Size

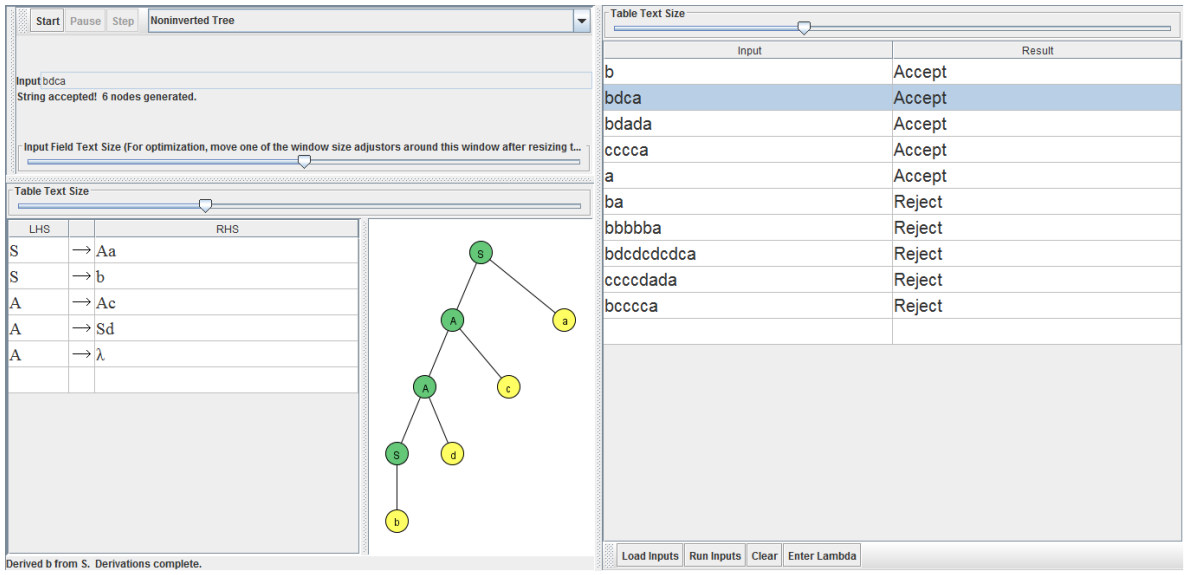
Input	Result
()	Accept
(())	Accept
(()())	Accept
()(())	Accept
(())(())	Accept
(()	Reject
())	Reject
(())(	Reject
()(((((((()((	Reject
(()(()()()	Reject

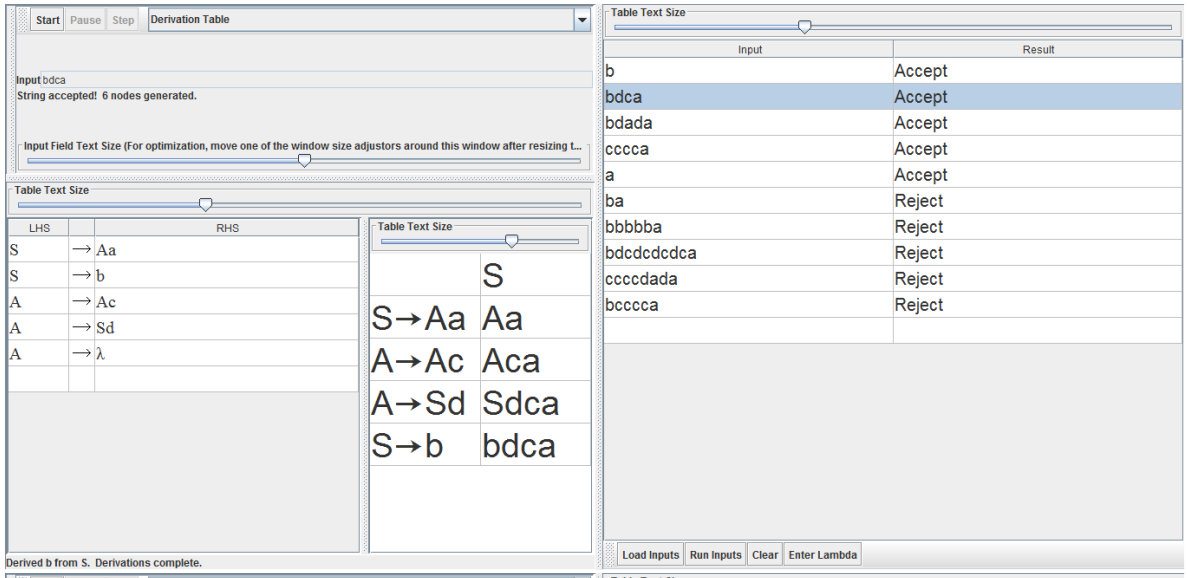
Load Inputs | Run Inputs | Clear | Enter Lambda

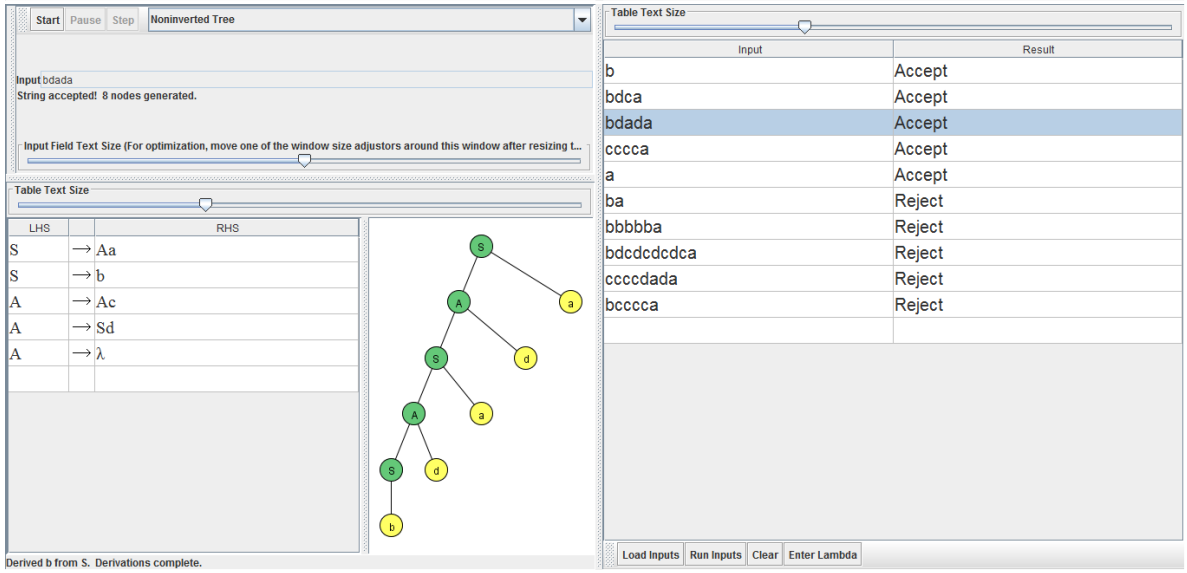


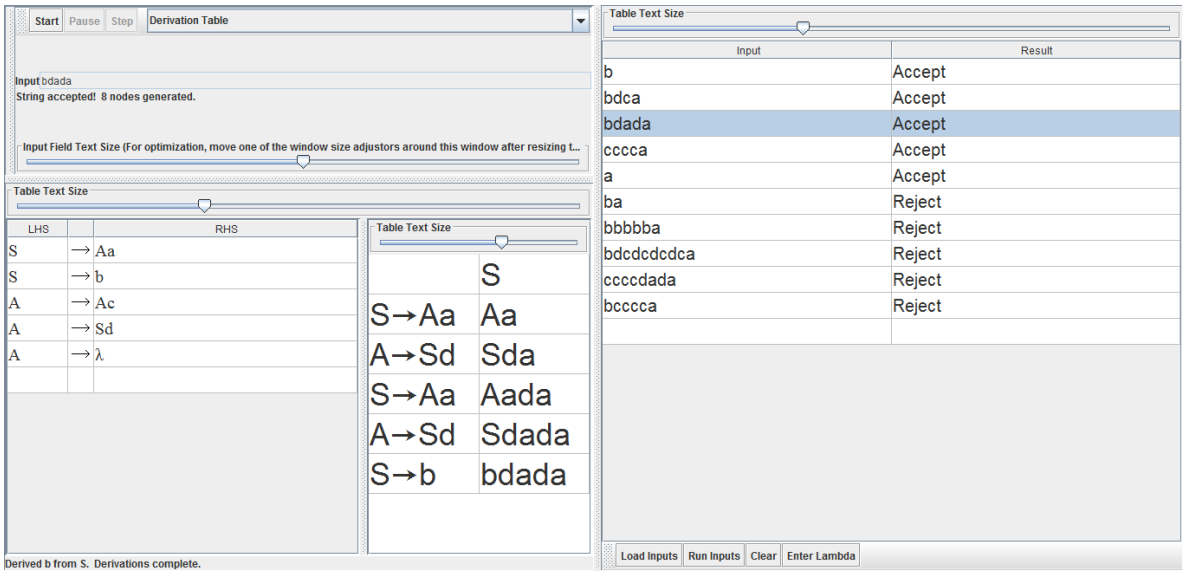


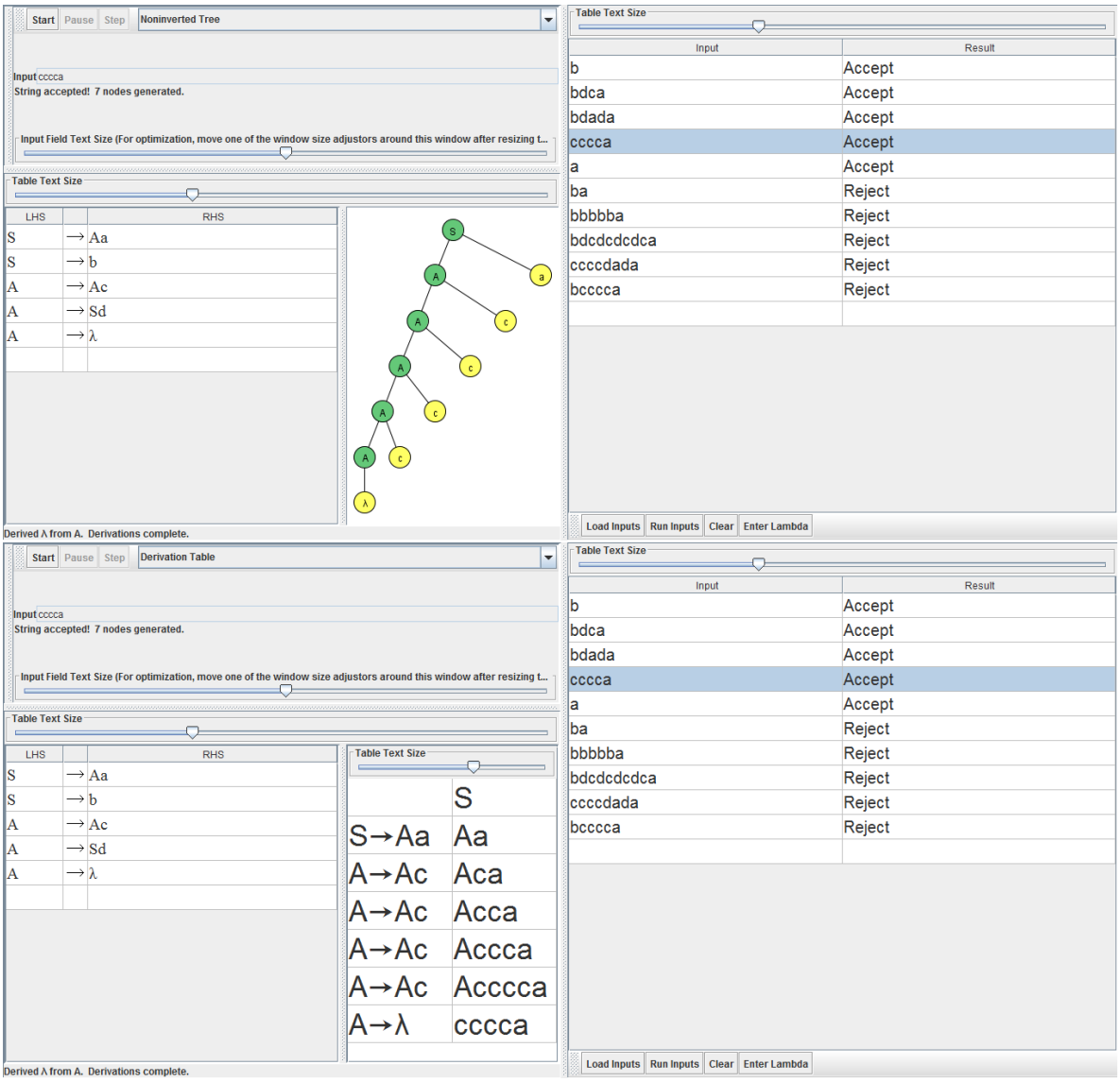
4) L(G4)={ b,a,bdca,cccc…a,bdad…a } Tipo 2, No ambigua



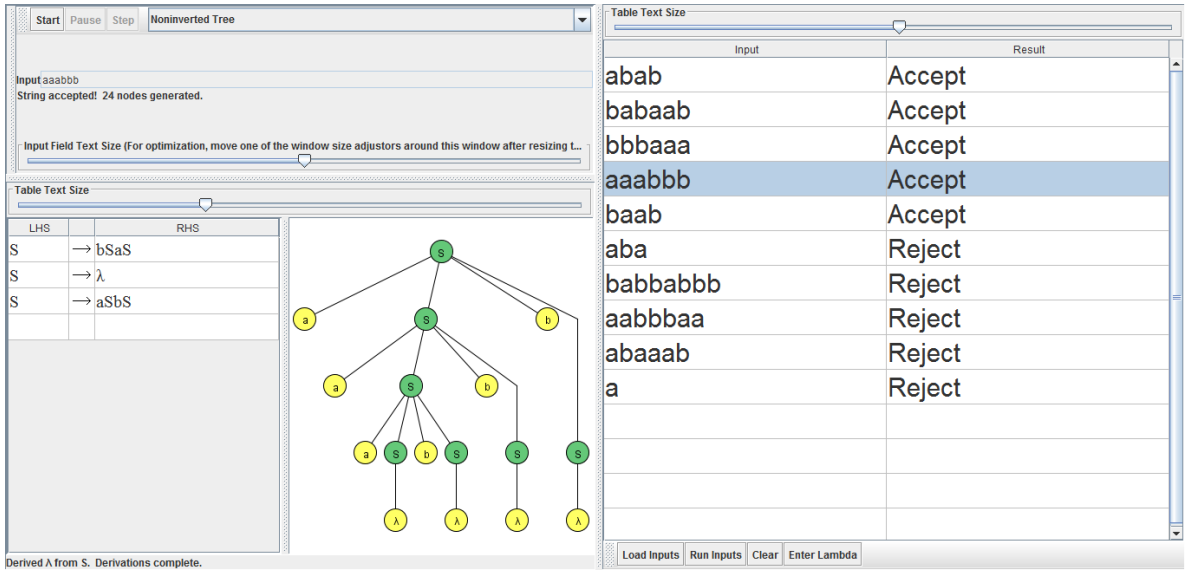

Start
Pause
Step
Derivation Table
Input bdca
String accepted! 6 nodes generated.
Input Field Text Size (For optimization, move one of the window size adjustors around this window after resizing t...
Table Text Size
LHS
RHS
S → Aa
S → b
A → Ac
A → Sd
A → λ
Table Text Size
S
S→Aa Aa
A→Ac Aca
A→Sd Sdca
S→b bdca
Derived b from S. Derivations complete.
Table Text Size
Input
Result
b Accept
bdca Accept
bdada Accept
cccca Accept
a Accept
ba Reject
bbbbba Reject
bdcdcdcdca Reject
ccccdada Reject
bcccca Reject
Load Inputs
Run Inputs
Clear
Enter Lambda


Start
Pause
Step
Noninverted Tree
Input bdada
String accepted! 8 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 → Aa
S → b
A → Ac
A → Sd
A → λ
S
A
a
S
d
A
a
S
d
b
Derived b from S. Derivations complete.
Table Text Size
Input
Result
b Accept
bdca Accept
bdada Accept
cccca Accept
a Accept
ba Reject
bbbbba Reject
bdcdcdcdca Reject
ccccdada Reject
bcccca Reject
Load Inputs
Run Inputs
Clear
Enter Lambda


Start
Pause
Step
Derivation Table
Input bdada
String accepted! 8 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 → Aa
S → b
A → Ac
A → Sd
A → λ
Table Text Size
S
S→Aa Aa
A→Sd Sda
S→Aa Aada
A→Sd Sdada
S→b bdada
Derived b from S. Derivations complete.
Table Text Size
Input
Result
b Accept
bdca Accept
bdada Accept
cccca Accept
a Accept
ba Reject
bbbbba Reject
bdcdcdcdca Reject
ccccdada Reject
bcccca Reject
Load Inputs
Run Inputs
Clear
Enter Lambda



5) L(G5)={ab,abab,ababa,bbaa,…} Tipo 2, No ambigua



Start | Pause | Step | Derivation Table

Input aaabbb

String accepted! 24 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	→	bSaS
S	→	λ
S	→	aSbS

Table Text Size

	S
S→aSbS	aSbS
S→aSbS	aaSbSbS
S→aSbS	aaaSbSbSbS
S→λ	aaabSbSbS
S→λ	aaabbSbS
S→λ	aaabbbS
S→λ	aaabbb

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
aabbbaa	Reject
abaaab	Reject
a	Reject

Load Inputs | Run Inputs | Clear | Enter Lambda

Start | Pause | Step | Noninverted Tree

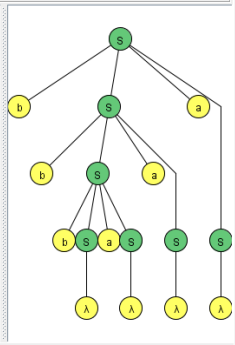
Input bbbaaa

String accepted! 24 nodes generated.

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

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
aabbbaa	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 adjustors around this window after resizing t...

Table Text Size

LHS		RHS
S	→	bSaS
S	→	λ
S	→	aSbS

Table Text Size

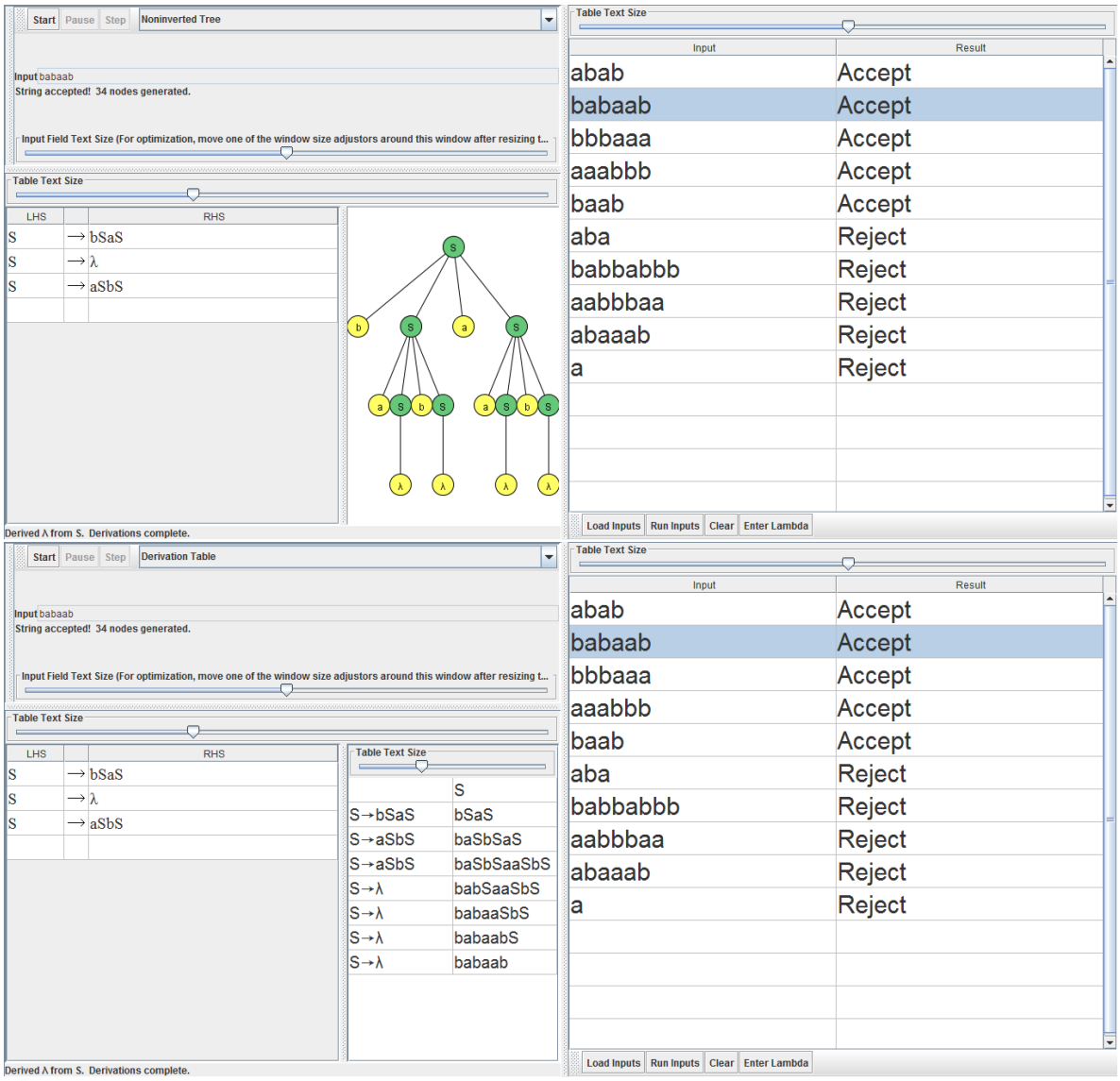
	S
S→bSaS	bSaS
S→bSaS	bbSaSaS
S→bSaS	bbbSaSaSaS
S→λ	bbbaSaSaS
S→λ	bbbaaSaS
S→λ	bbbaaaS
S→λ	bbbaaa

Derived λ from S. Derivations complete.

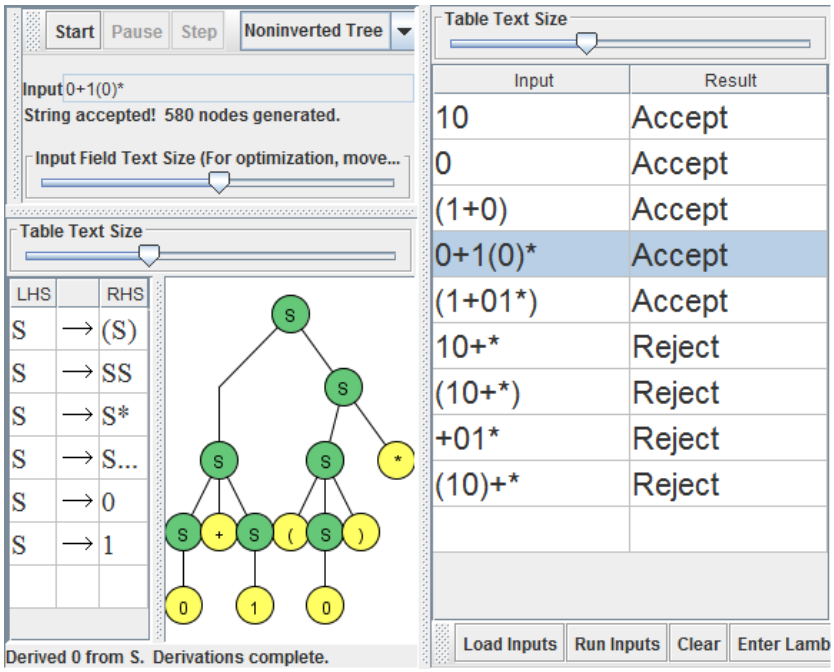
Table Text Size

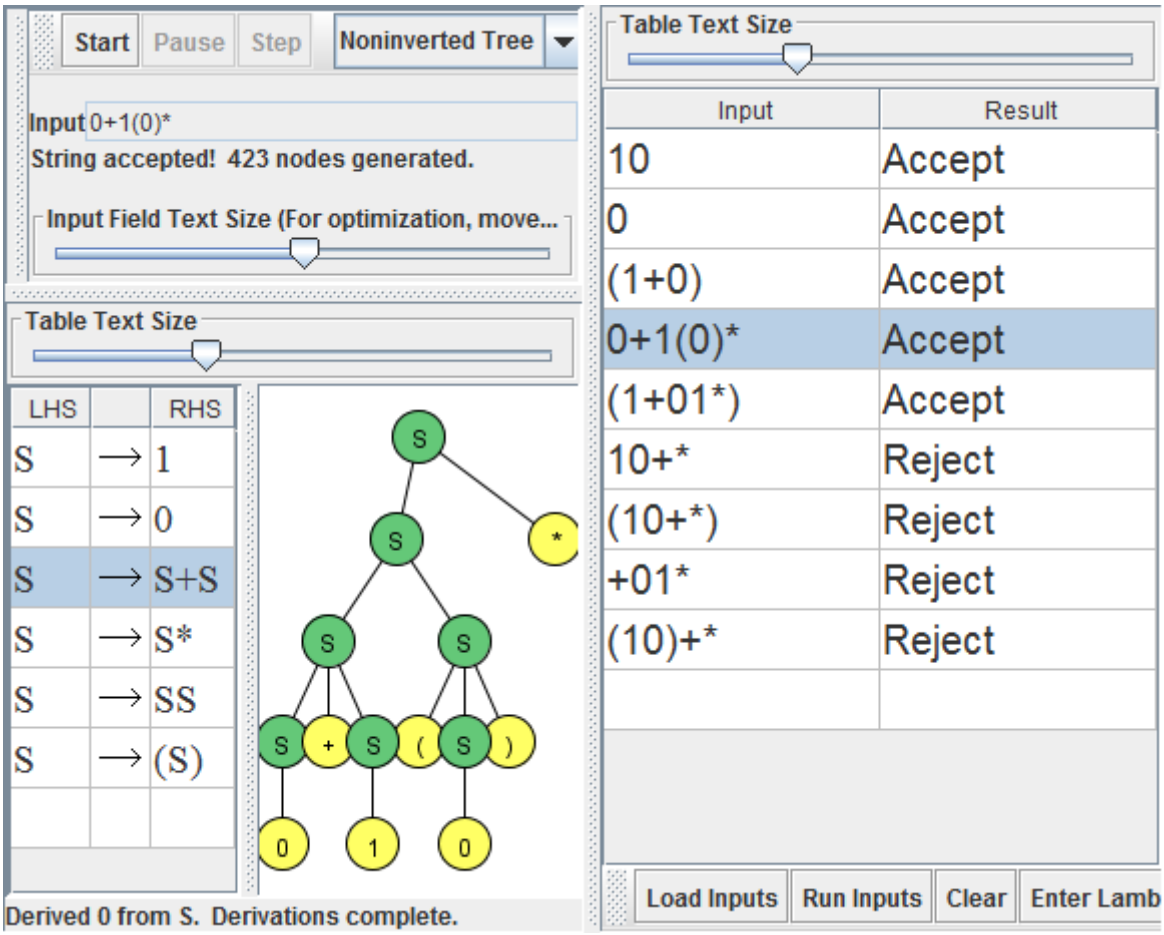
Input	Result
abab	Accept
babaab	Accept
bbbaaa	Accept
aaabbb	Accept
baab	Accept
aba	Reject
babbabbb	Reject
aabbbaa	Reject
abaaab	Reject
a	Reject

Load Inputs | Run Inputs | Clear | Enter Lambda

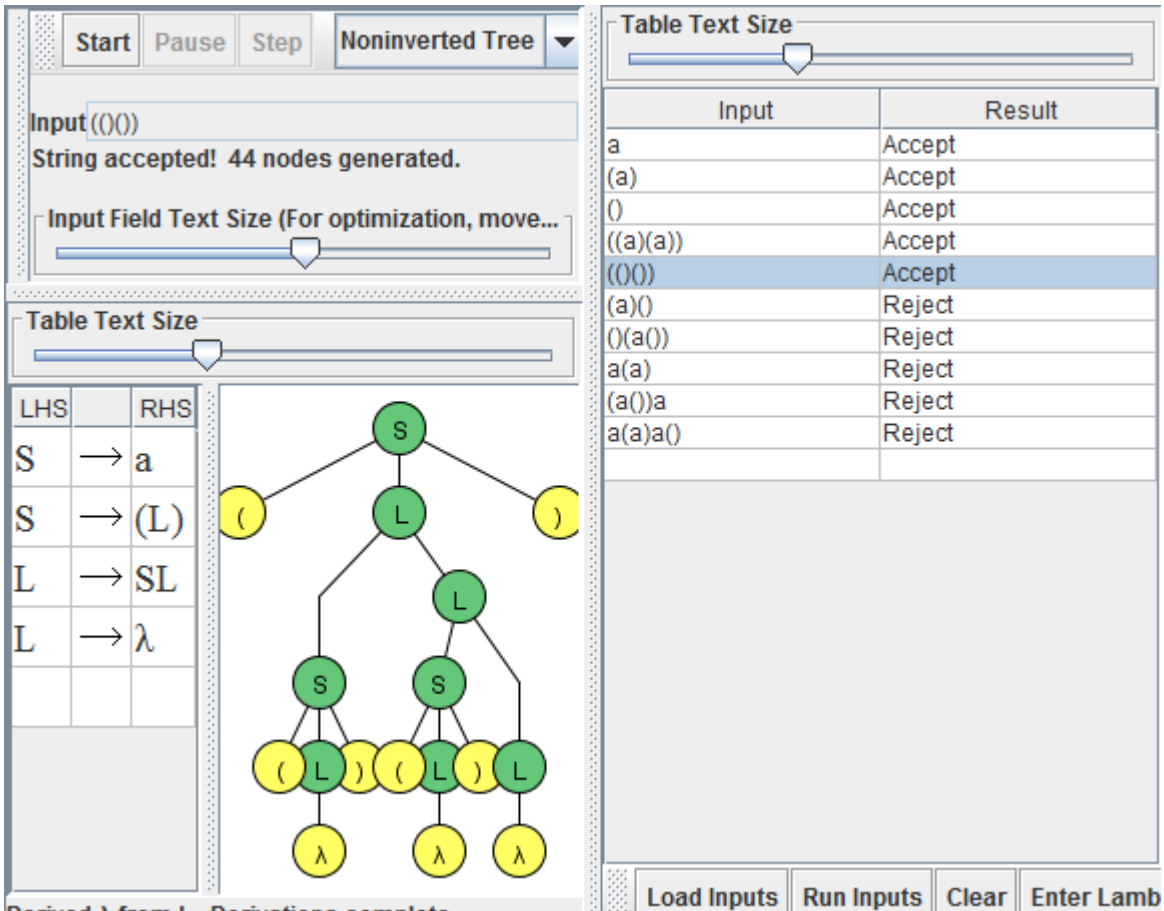


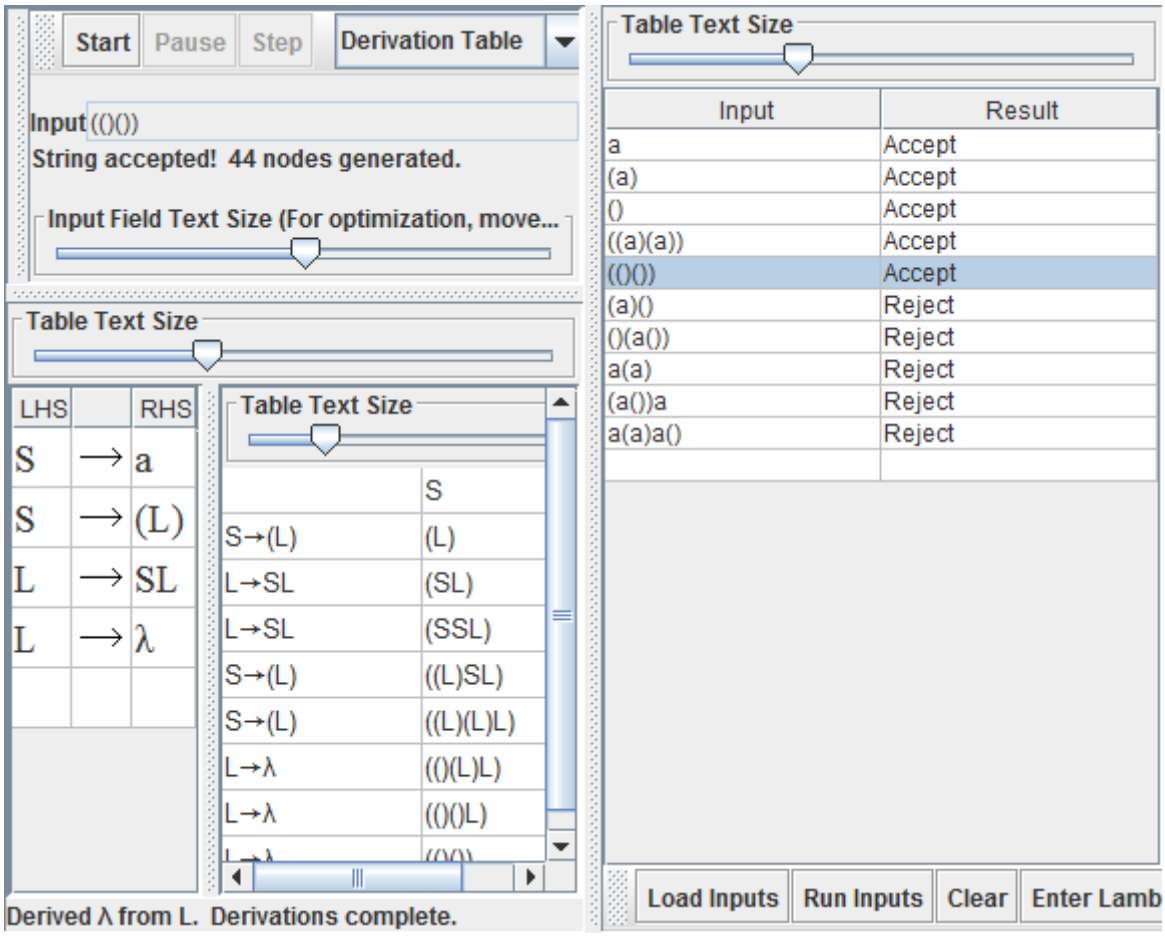
6) L(G6)={0,1,01,011,10,1+0,(0+1),(1+0),(1+01*)} Tipo 2, Ambigua

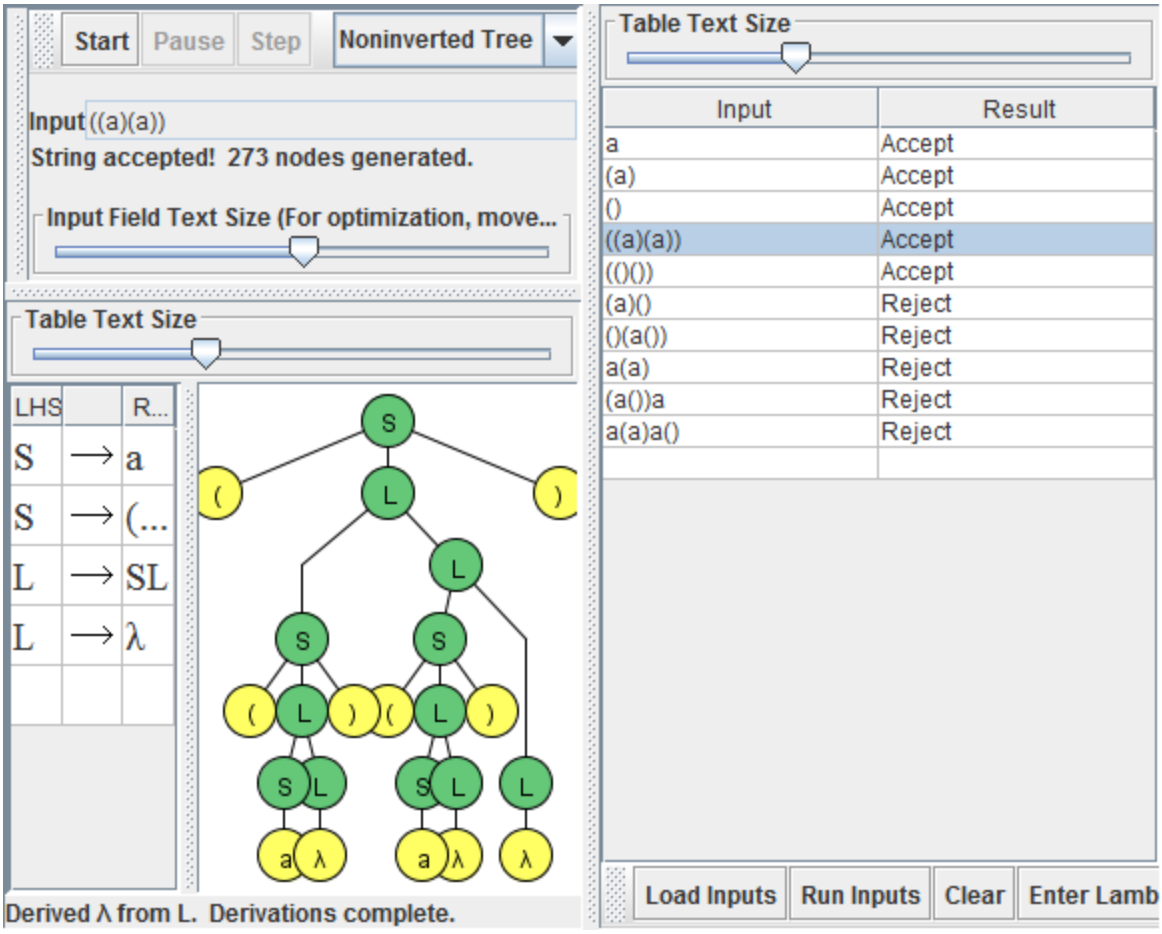


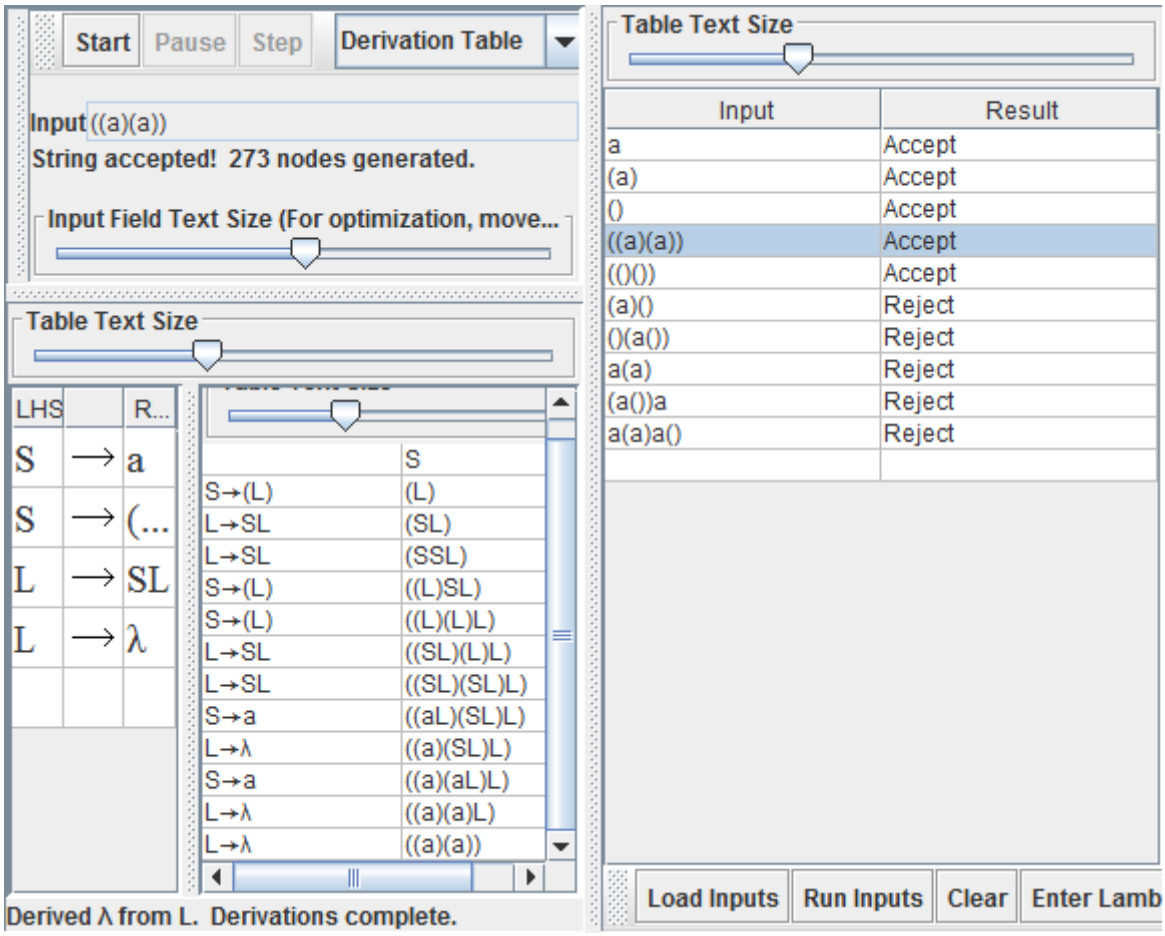


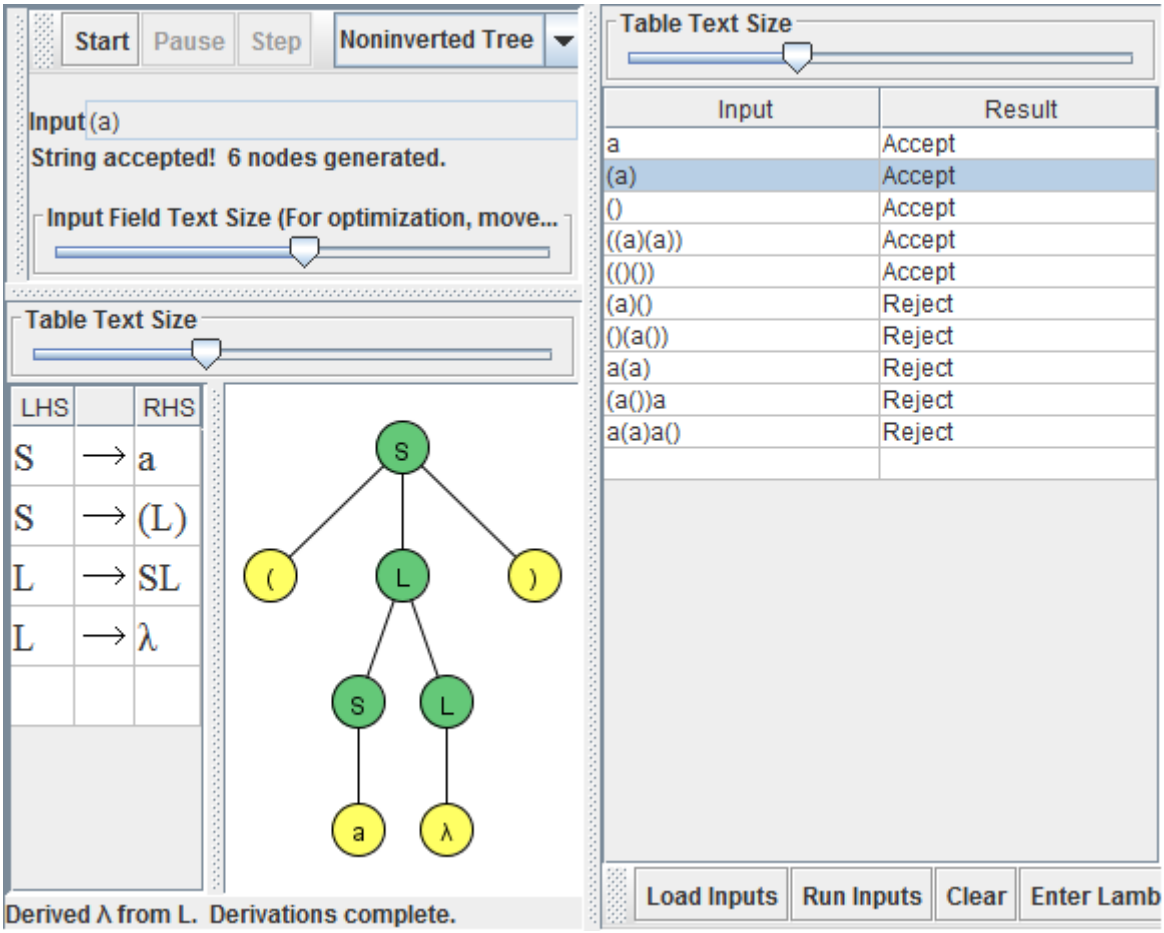
7) L(G7)={(),a,(a),((a)),((a)(a)),… (),(()),…} Tipo 2, No ambigua

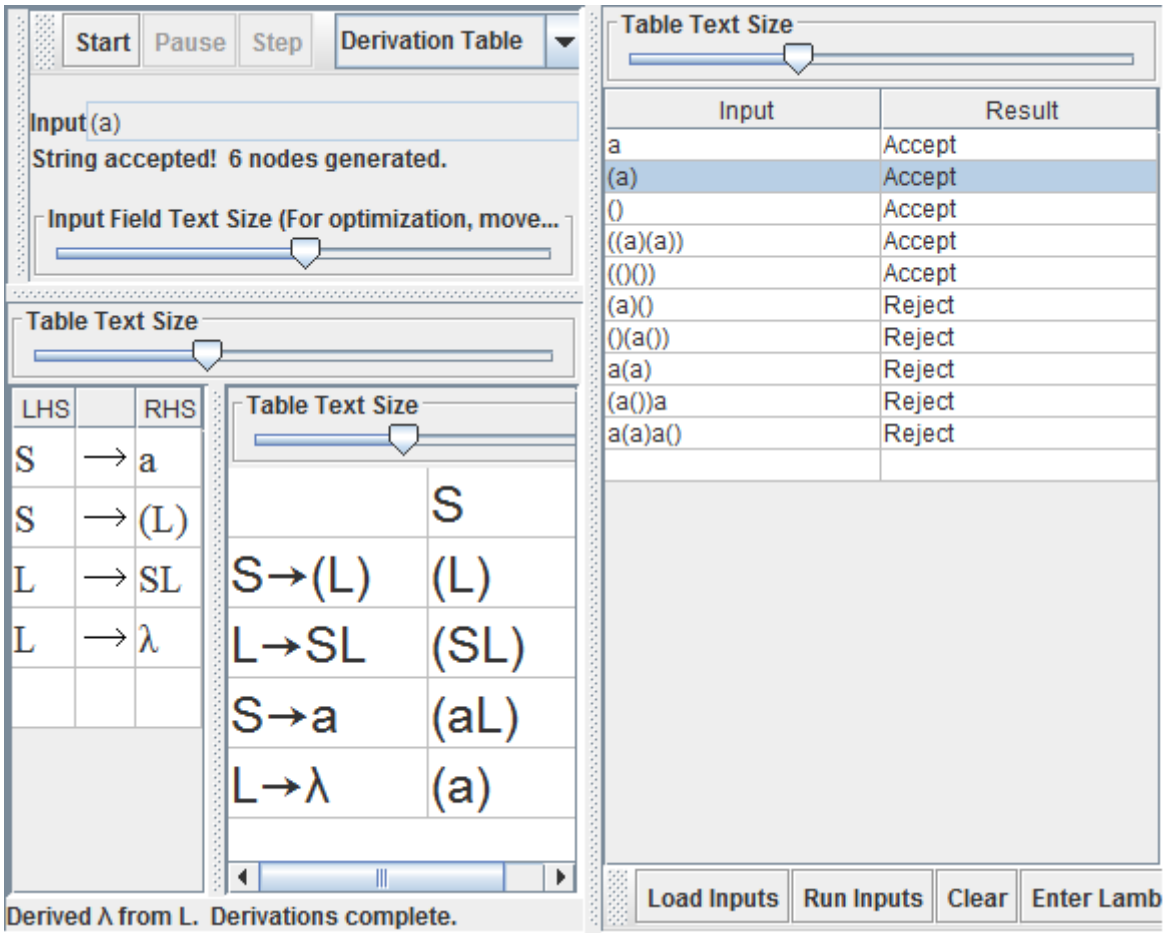



Start
Pause
Step
Derivation Table
Input (()())
String accepted! 44 nodes generated.
Input Field Text Size (For optimization, move...
Table Text Size
LHS
RHS
S → a
S → (L)
L → SL
L → λ
Table Text Size
S
S→(L) (L)
L→SL (SL)
L→SL (SSL)
S→(L) ((L)SL)
S→(L) ((L)(L)L)
L→λ (()(L)L)
L→λ (()()L)
Derived λ from L. Derivations complete.
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

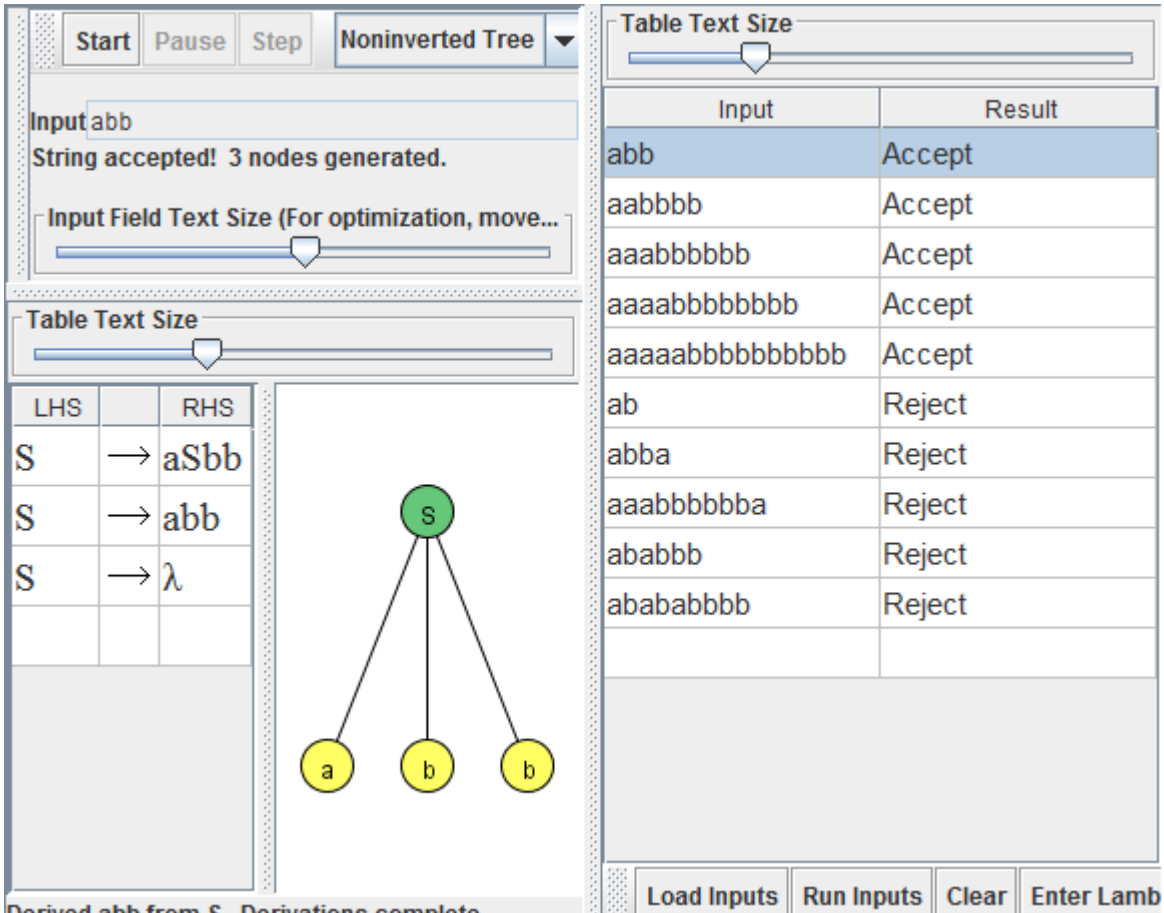

Start
Pause
Step
Noninverted Tree
Input ((a)(a))
String accepted! 273 nodes generated.
Input Field Text Size (For optimization, move...
Table Text Size
LHS
R...
S → a
S → (...
L → SL
L → λ
Derived λ from L. Derivations complete.
Table Text Size
Input
Result
a Accept
(a) Accept
() Accept
((a)(a)) Accept
(()()) Accept
(a)() Reject
()(a()) Reject
a(a) Reject
(a())a Reject
a(a)a() Reject
Load Inputs
Run Inputs
Clear
Enter Lamb

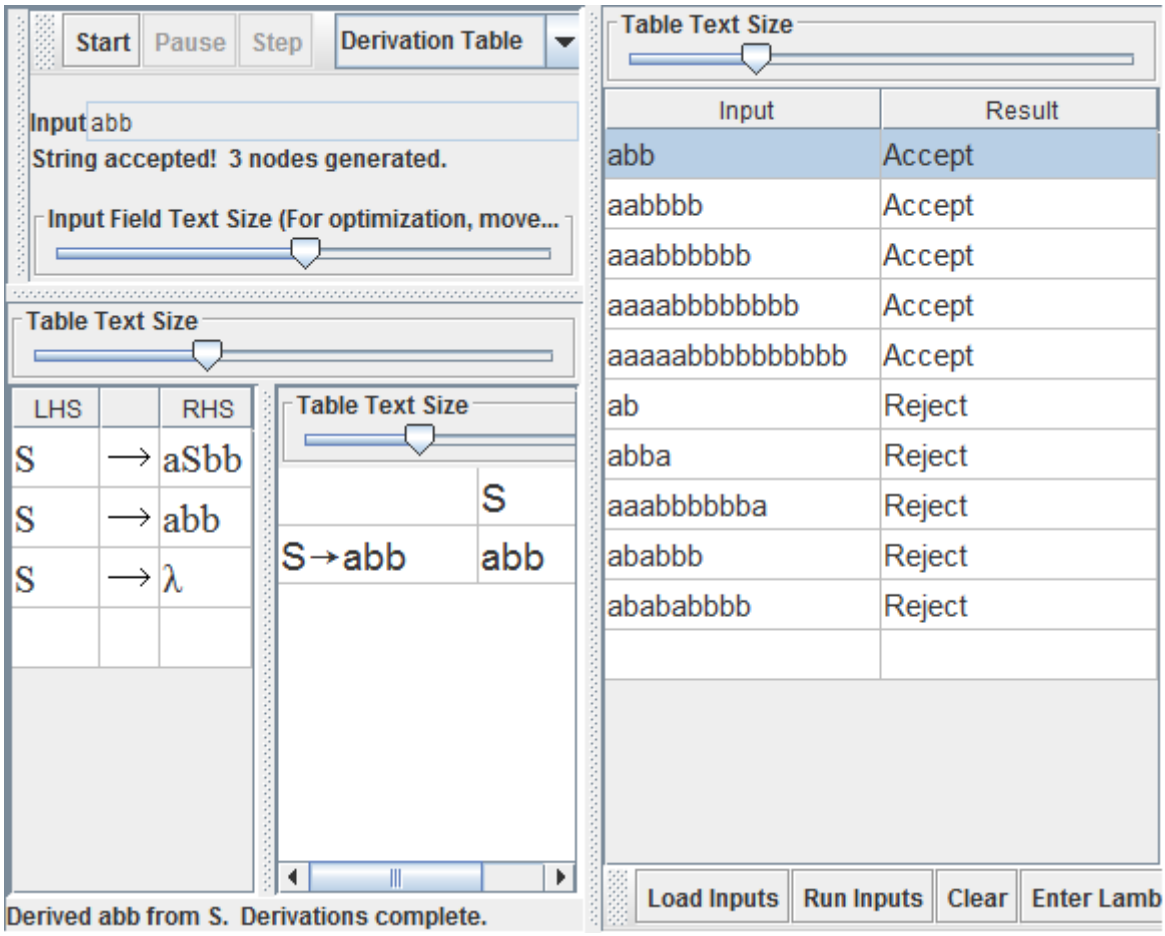

Start
Pause
Step
Derivation Table
Input ((a)(a))
String accepted! 273 nodes generated.
Input Field Text Size (For optimization, move...
Table Text Size
LHS
R...
S → a
S → (...
L → SL
L → λ
S
S→(L) (L)
L→SL (SL)
L→SL (SSL)
S→(L) ((L)SL)
S→(L) ((L)(L)L)
L→SL ((SL)(L)L)
L→SL ((SL)(SL)L)
S→a ((aL)(SL)L)
L→λ ((a)(SL)L)
S→a ((a)(aL)L)
L→λ ((a)(a)L)
L→λ ((a)(a))
Derived λ from L. Derivations complete.
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

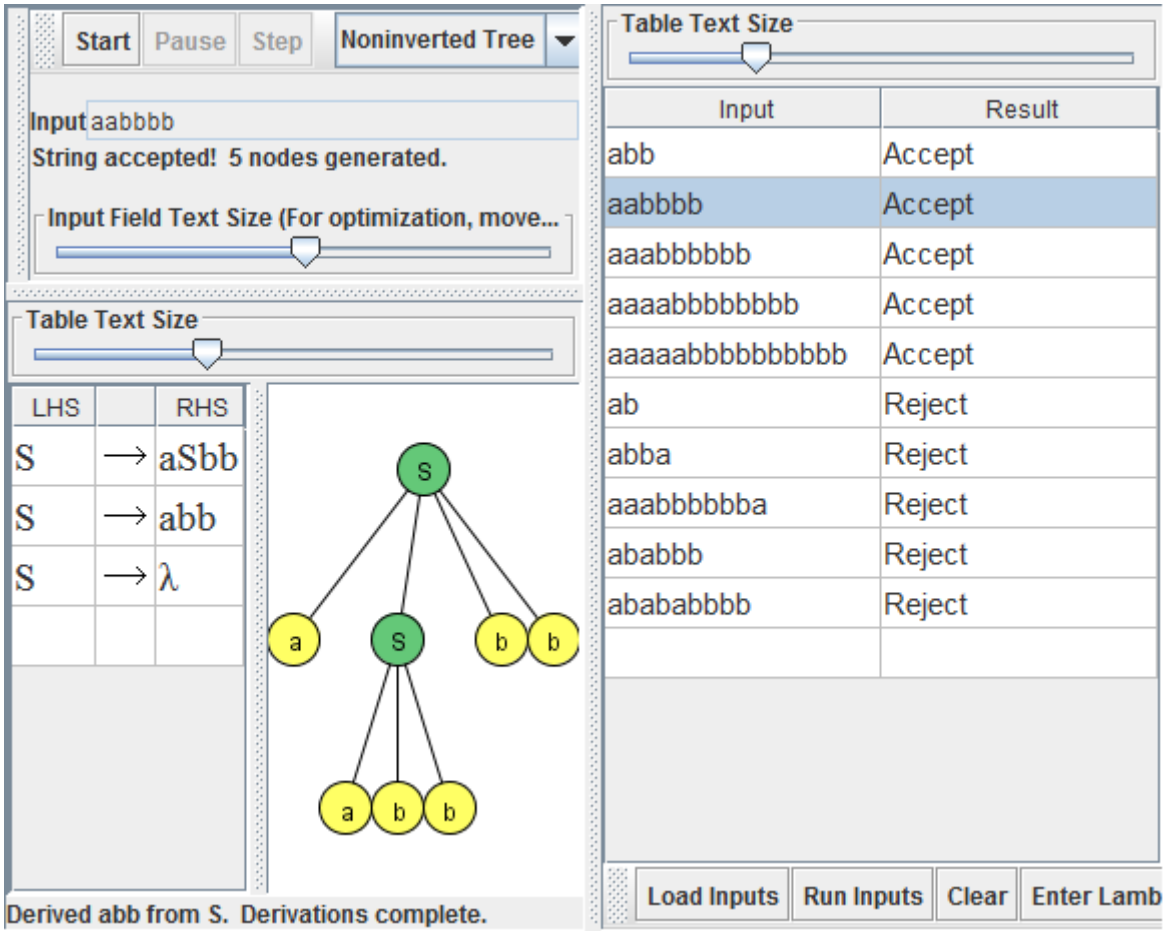

Start
Pause
Step
Noninverted Tree
Input (a)
String accepted! 6 nodes generated.
Input Field Text Size (For optimization, move...
Table Text Size
LHS
RHS
S → a
S → (L)
L → SL
L → λ
S
(
L
)
S
L
a
λ
Derived λ from L. Derivations complete.
Table Text Size
Input
Result
a Accept
(a) Accept
() Accept
((a)(a)) Accept
(()()) Accept
(a)() Reject
()(a()) Reject
a(a) Reject
(a())a Reject
a(a)a() Reject
Load Inputs
Run Inputs
Clear
Enter Lamb

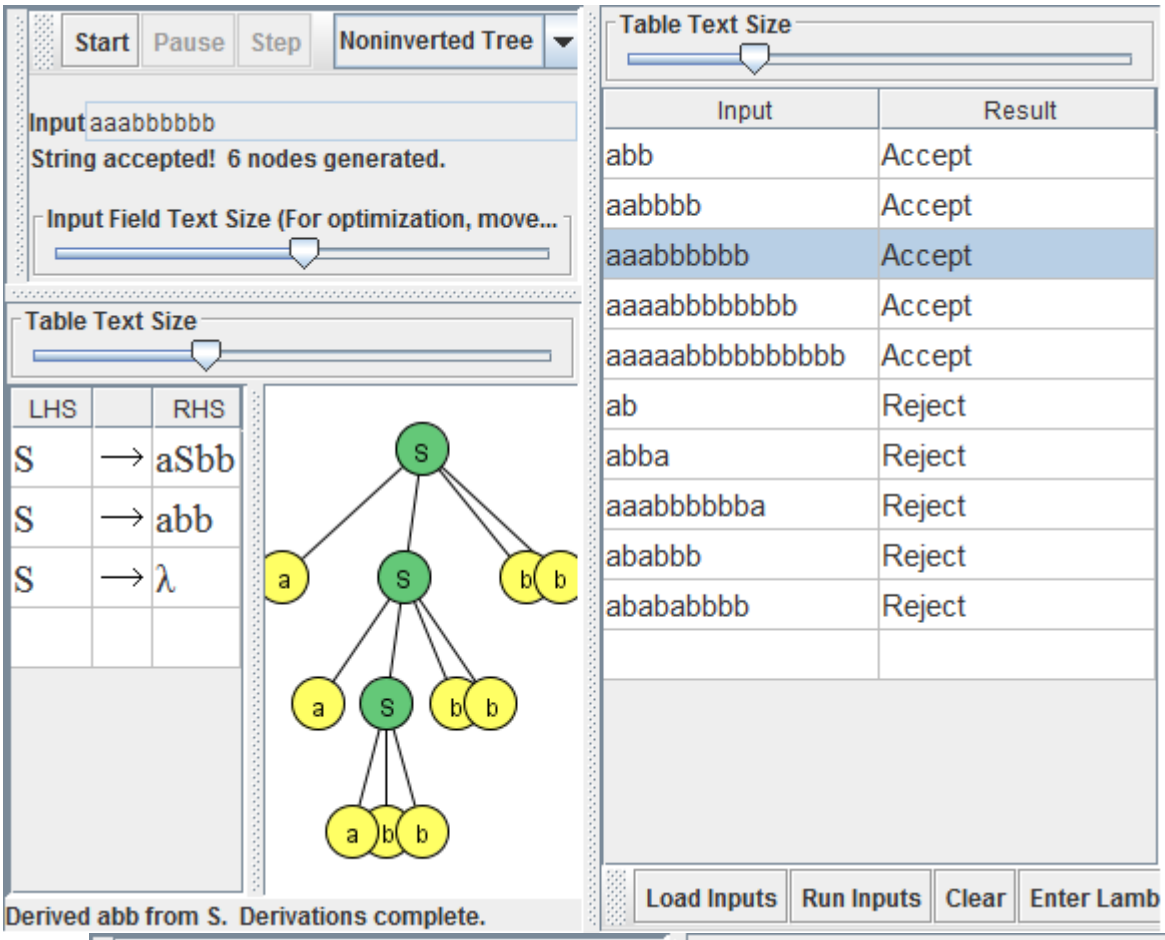


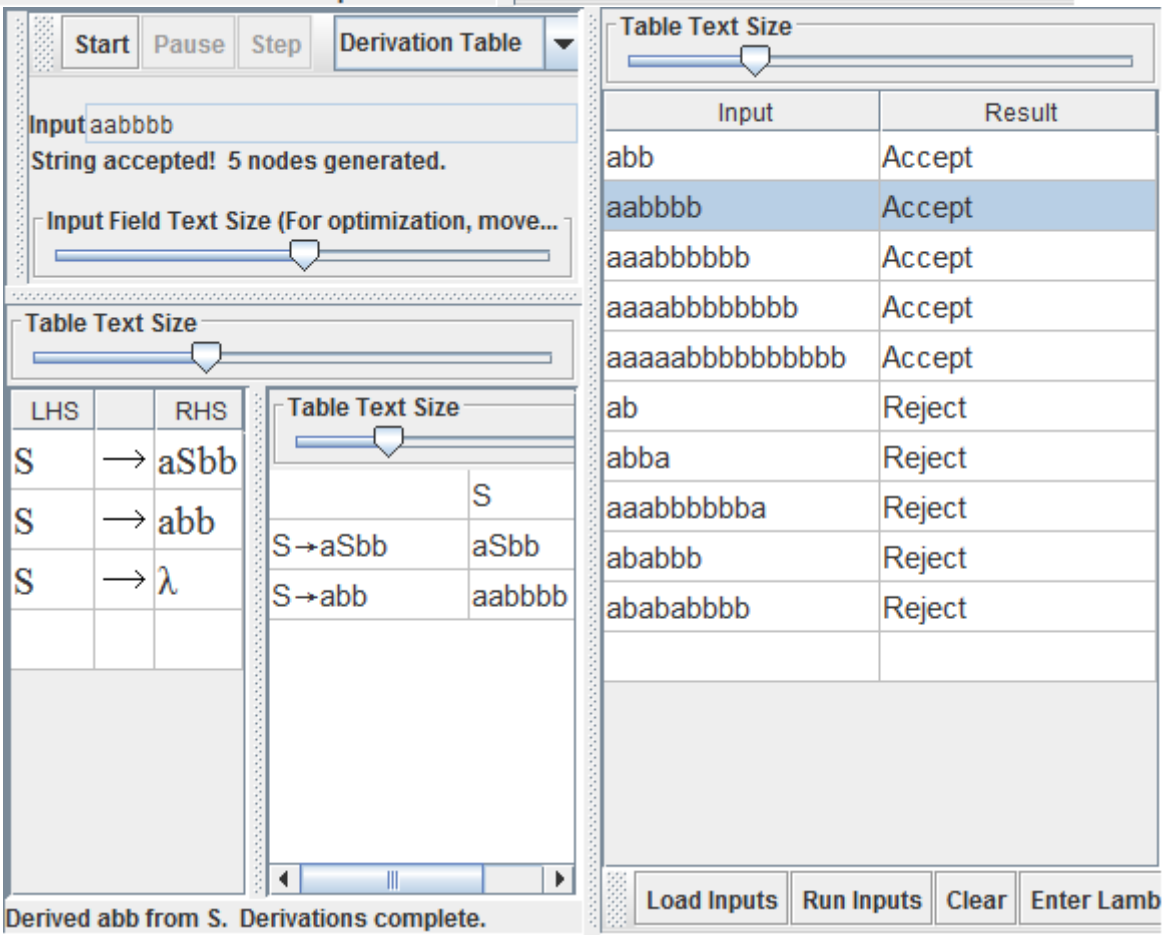
8) L(G8)={$a^n b^m$ | m=2n, m,n=0,1,2,…} Tipo 2, No ambigua

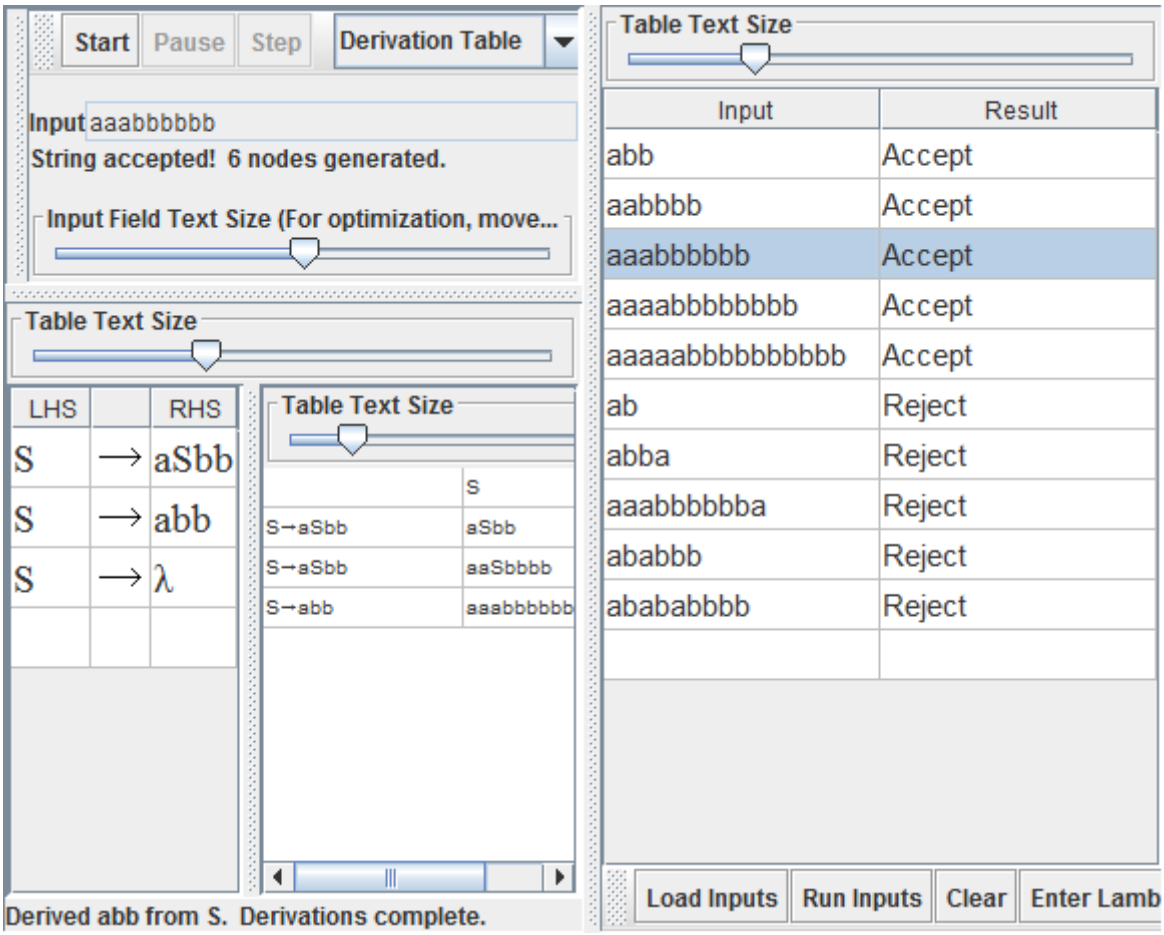



Start
Pause
Step
Derivation Table
Input abb
String accepted! 3 nodes generated.
Input Field Text Size (For optimization, move...
Table Text Size
LHS
RHS
S → aSbb
S → abb
S → λ
Table Text Size
S
S→abb
abb
Derived abb from S. Derivations complete.
Table Text Size
Input
Result
abb
Accept
aabbbb
Accept
aaabbbbbb
Accept
aaaabbbbbbb
Accept
aaaaabbbbbbbbb
Accept
ab
Reject
abba
Reject
aaabbbbbba
Reject
ababb
Reject
abababbb
Reject
Load Inputs
Run Inputs
Clear
Enter Lamb

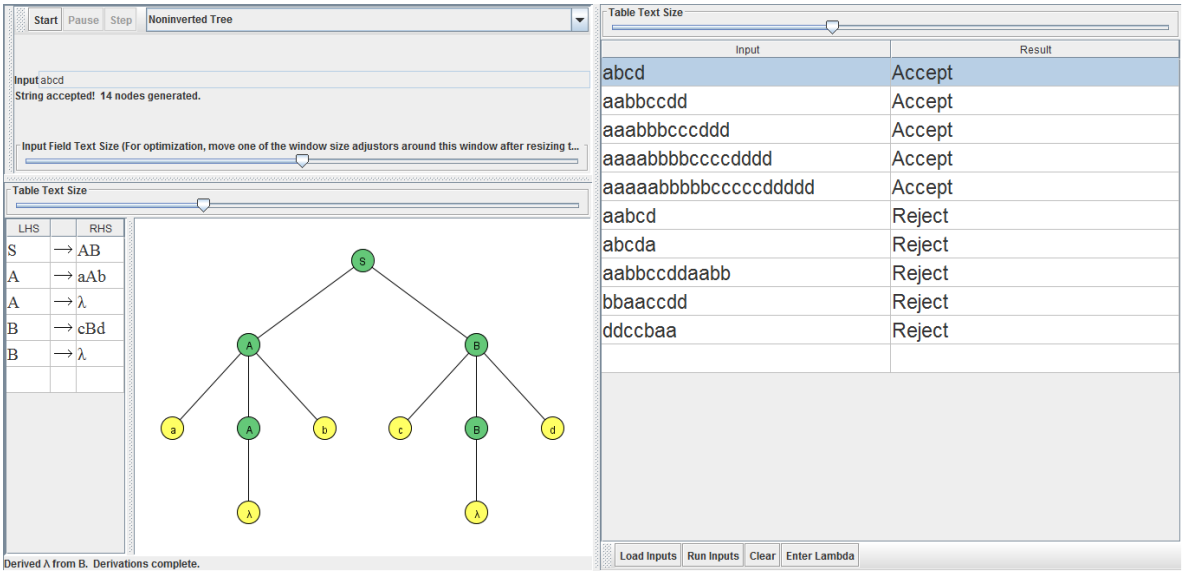

Start
Pause
Step
Noninverted Tree
Input aabbbb
String accepted! 5 nodes generated.
Input Field Text Size (For optimization, move...
Table Text Size
LHS
RHS
S → aSbb
S → abb
S → λ
S
a
S
b
b
a
b
b
Derived abb from S. Derivations complete.
Table Text Size
Input
Result
abb
Accept
aabbbb
Accept
aaabbbbbb
Accept
aaaabbbbbbb
Accept
aaaaabbbbbbbbb
Accept
ab
Reject
abba
Reject
aaabbbbbba
Reject
ababb
Reject
abababbb
Reject
Load Inputs
Run Inputs
Clear
Enter Lamb


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
S
a
b
Derived abb from S. Derivations complete.
Input
Result
abb Accept
aabbbb Accept
aaabbbbbb Accept
aaaabbbbbbbb Accept
aaaaabbbbbbbbbb Accept
ab Reject
abba Reject
aaabbbbbba Reject
ababbb Reject
abababbbb Reject
Load Inputs
Run Inputs
Clear
Enter Lamb


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
S
S→aSbb
aSbb
S→abb
aabbbb
Derived abb from S. Derivations complete.
Input
Result
abb Accept
aabbbb Accept
aaabbbbbb Accept
aaaabbbbbbbb Accept
aaaaabbbbbbbbbb Accept
ab Reject
abba Reject
aaabbbbbba Reject
ababbb Reject
abababbbb Reject
Load Inputs
Run Inputs
Clear
Enter Lamb



9) L(G9)={ $a^n b^m c^i d^j$ | n=m=i=j=0,1.2,…} Tipo 2, No ambigua



Start | Pause | Step | Derivation Table

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

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

Derived λ from B. Derivations complete.

Table Text Size

Input	Result
abcd	Accept
aabbccdd	Accept
aaabbbcccddd	Accept
aaaabbbbccccdddd	Accept
aaaaabbbbbccccddddd	Accept
aabcd	Reject
abcda	Reject
aabbccddaabb	Reject
bbaaccdd	Reject
ddccbaa	Reject

Load Inputs | Run Inputs | Clear | Enter Lambda

Start | Pause | Step | Noninverted Tree

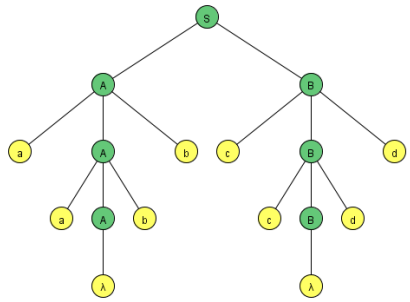
Input aabbccdd

String accepted! 21 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



Derived λ from B. Derivations complete.

Table Text Size

Input	Result
abcd	Accept
aabbccdd	Accept
aaabbbcccddd	Accept
aaaabbbbccccdddd	Accept
aaaaabbbbbccccddddd	Accept
aabcd	Reject
abcda	Reject
aabbccddaabb	Reject
bbaaccdd	Reject
ddccbaa	Reject

Load Inputs | Run Inputs | Clear | Enter Lambda

Start | Pause | Step | Derivation Table

Input aabbccdd

String accepted! 21 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

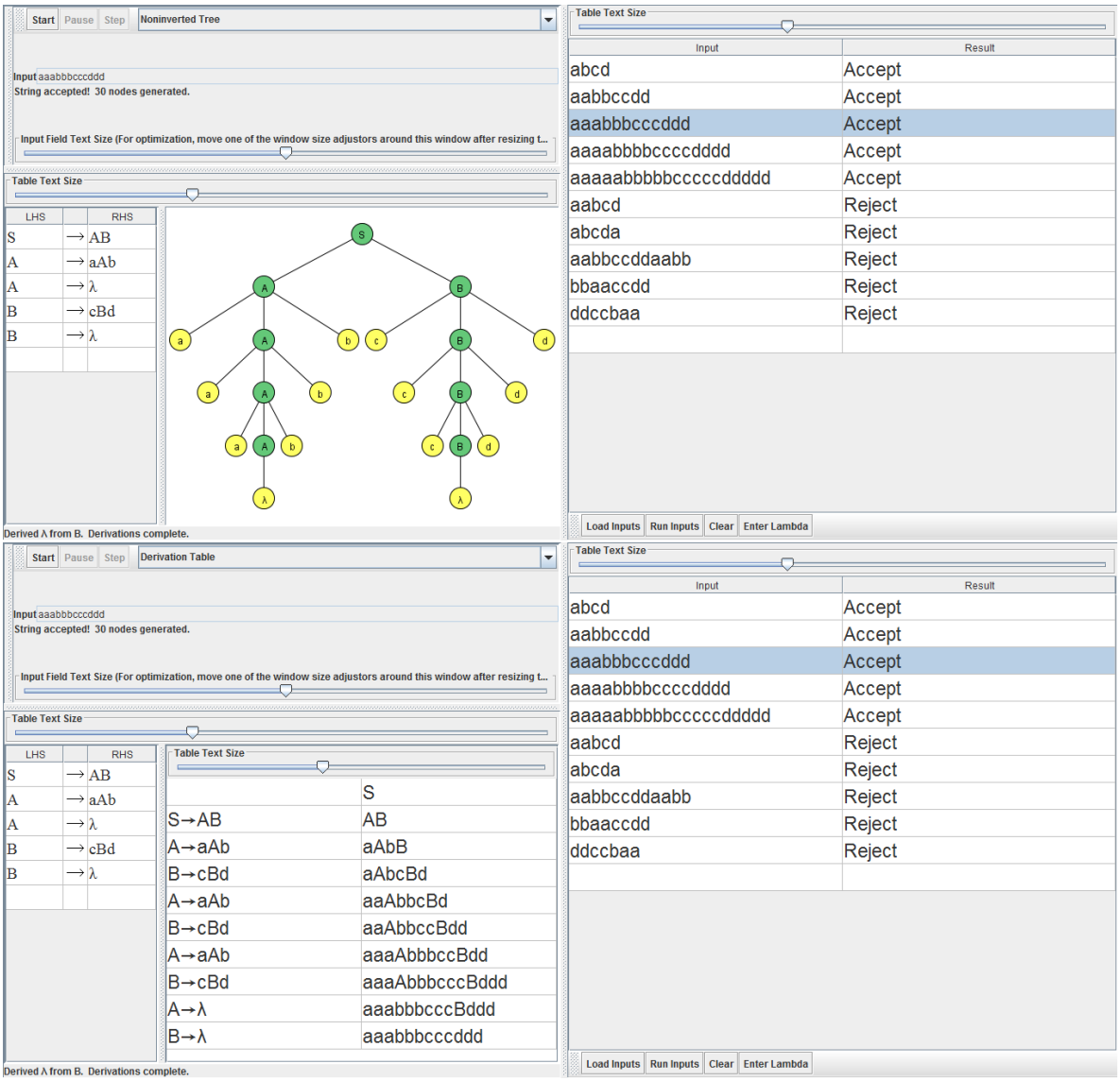
	S
S→AB	AB
A→aAb	aAbB
B→cBd	aAbcBd
A→aAb	aaAbbcBd
B→cBd	aaAbbccBdd
A→λ	aabbccBdd
B→λ	aabbccdd

Derived λ from B. Derivations complete.

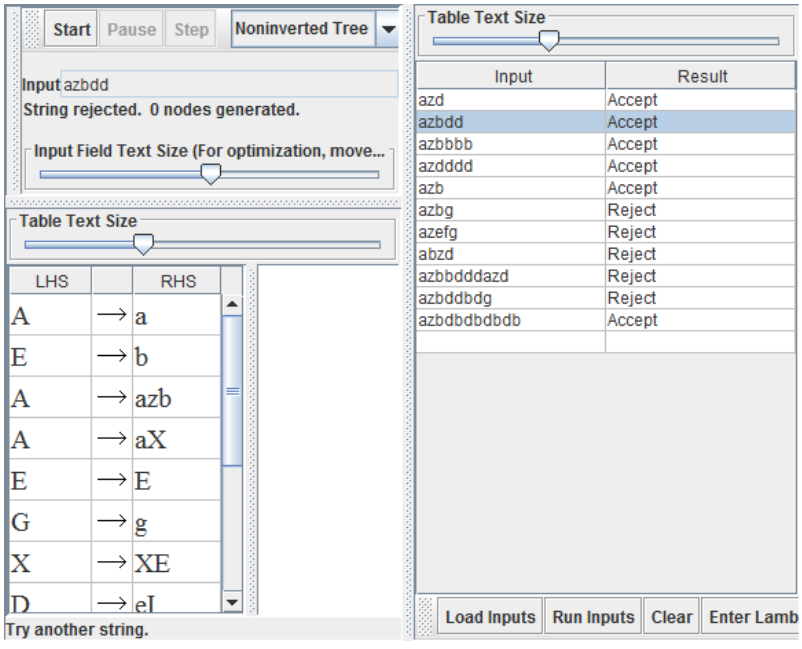
Table Text Size

Input	Result
abcd	Accept
aabbccdd	Accept
aaabbbcccddd	Accept
aaaabbbbccccdddd	Accept
aaaaabbbbbccccddddd	Accept
aabcd	Reject
abcda	Reject
aabbccddaabb	Reject
bbaaccdd	Reject
ddccbaa	Reject

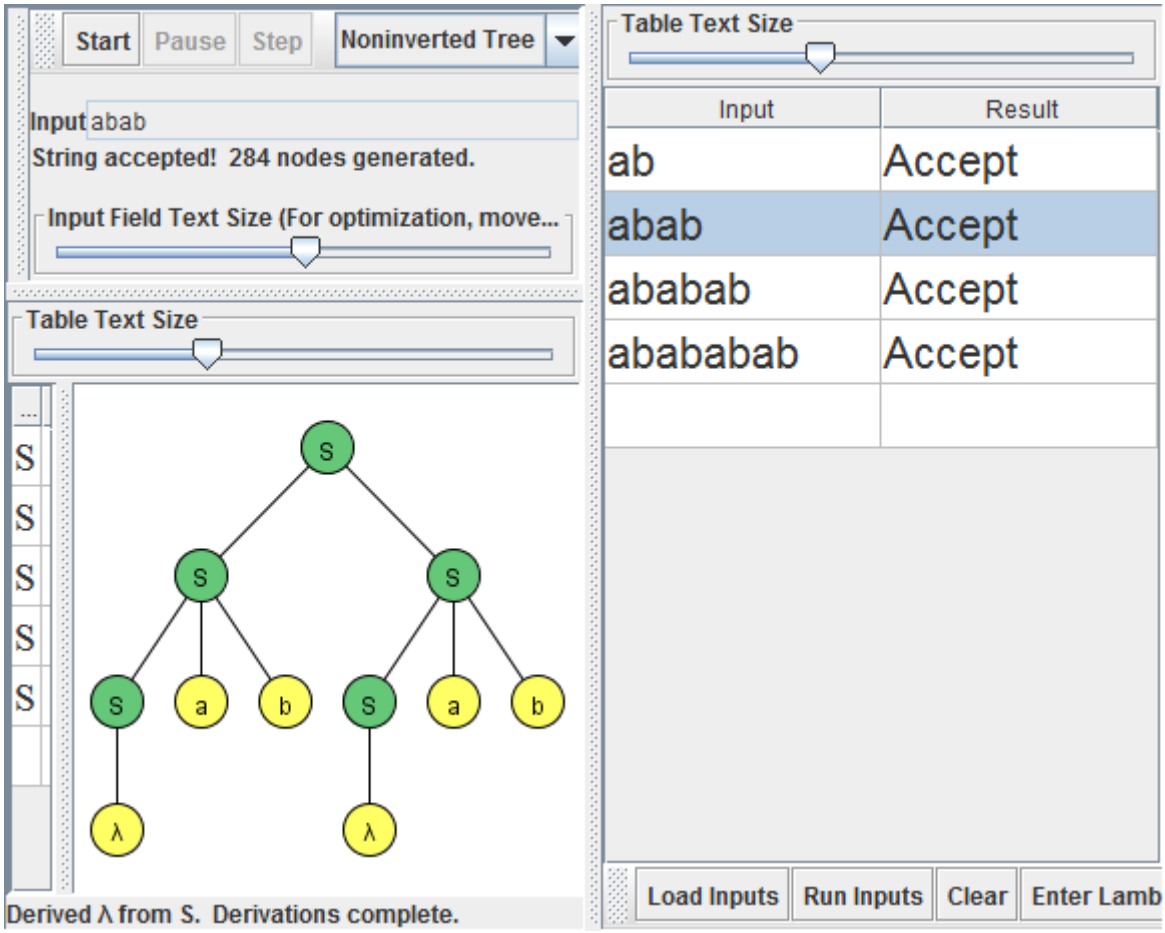
Load Inputs | Run Inputs | Clear | Enter Lambda

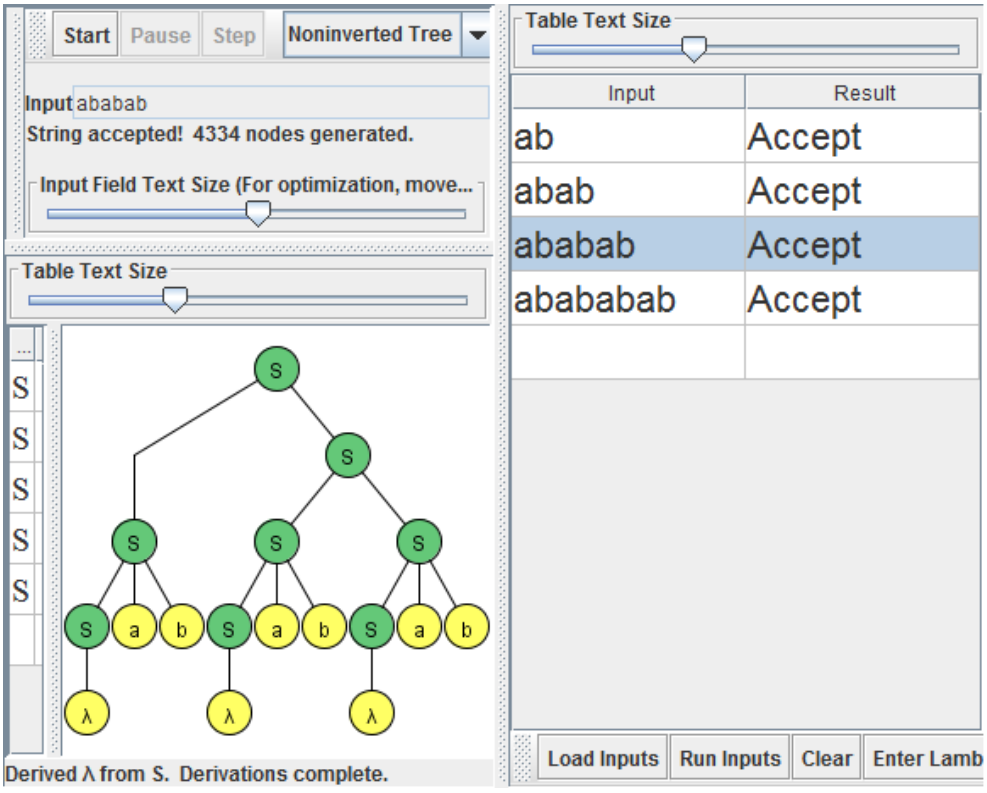


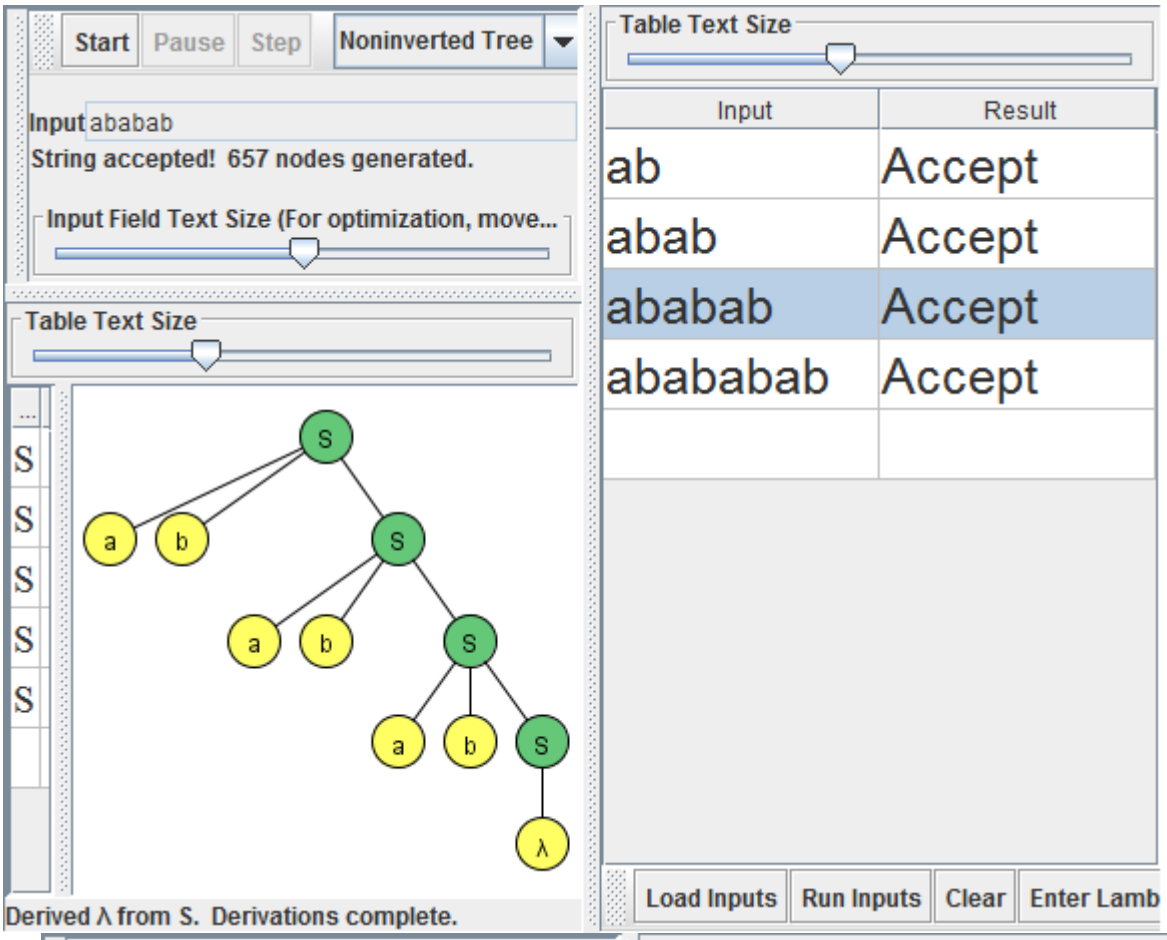
10) L(G10)={a,azb^n,azd^n,azb^{n}d^m,azb^{n}d^{m}b^n,… | n diferente de m>0} Tipo 2, Ambigua

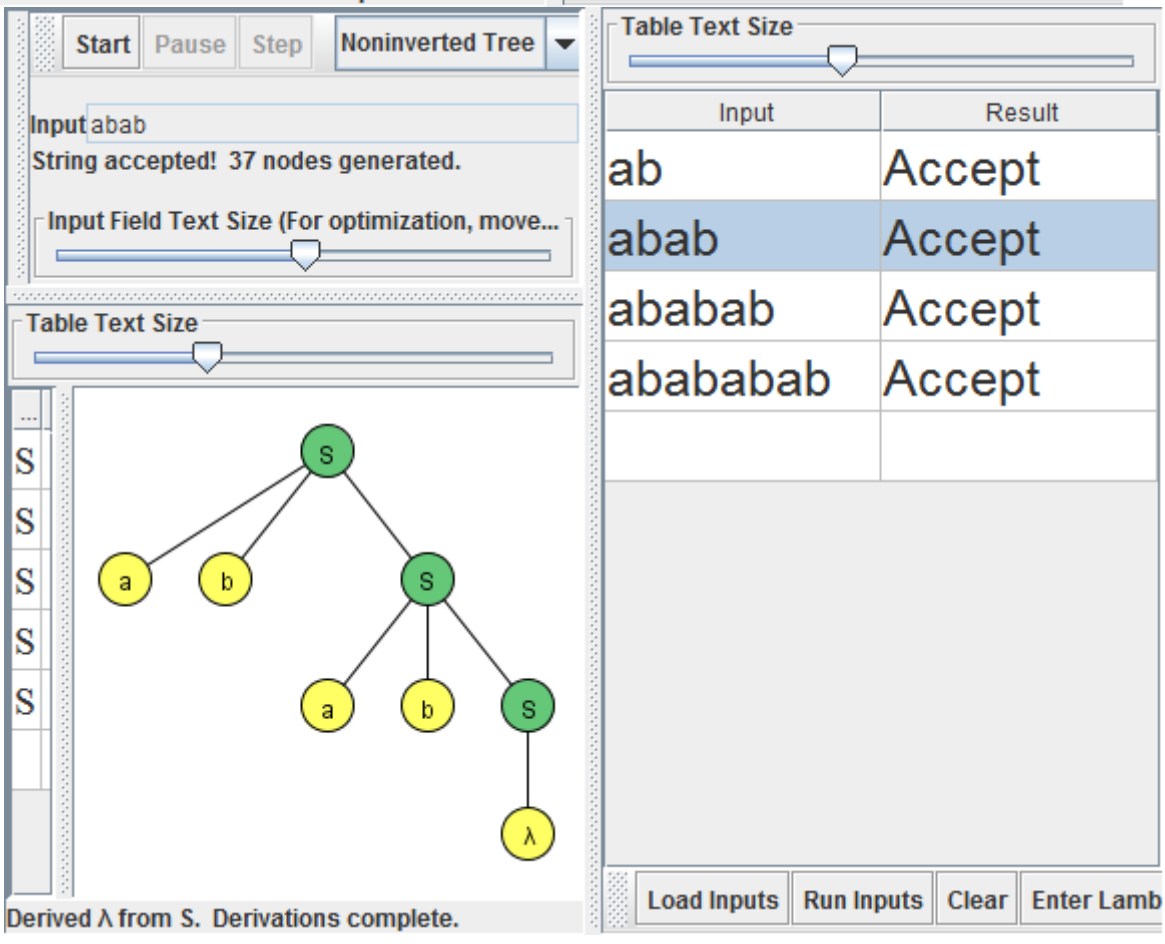


11) $L(G11)=\{(ab)^n,a^nb^m\}$ Tipo 2, Ambigua

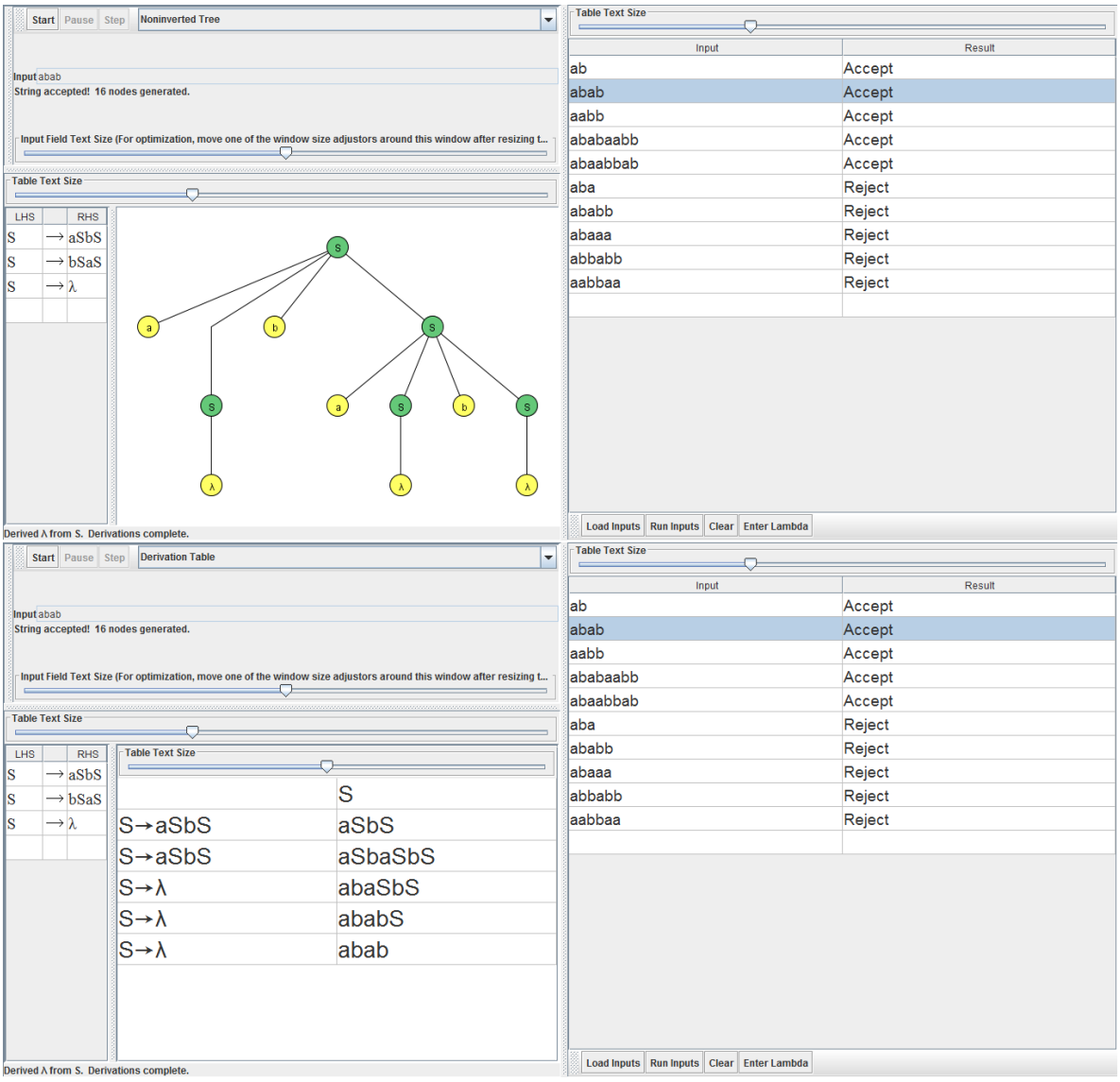





Start
Pause
Step
Noninverted Tree
Input ababab
String accepted! 657 nodes generated.
Input Field Text Size (For optimization, move...
Table Text Size
Input
Result
ab
Accept
abab
Accept
abababab
Accept
ababababab
Accept
Load Inputs
Run Inputs
Clear
Enter Lamb
S
a
b
λ
Derived λ from S. Derivations complete.


Start
Pause
Step
Noninverted Tree
Input abab
String accepted! 37 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
S
a
b
λ
Derived λ from S. Derivations complete.

12) $L(G12)=\{a^n b^b,(ab)^n\}$ Tipo 2, No ambigua



Start | Pause | Step | Noninverted Tree

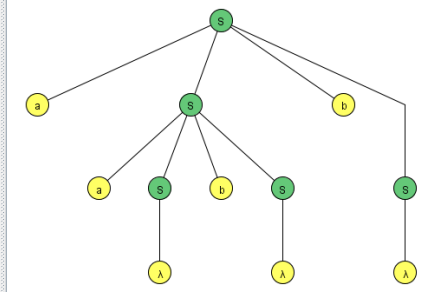
Input aabb

String accepted! 12 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	→	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
aabbaa	Reject

Load Inputs | Run Inputs | Clear | Enter Lambda

Start | Pause | Step | Derivation Table

Input aabb

String accepted! 12 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	→	aSbS
S	→	bSaS
S	→	λ

Table Text Size

	S
S→aSbS	aSbS
S→aSbS	aaSbSbS
S→λ	aabSbS
S→λ	aabbS
S→λ	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
aabbaa	Reject

Load Inputs | Run Inputs | Clear | Enter Lambda

Start | Pause | Step | Noninverted Tree

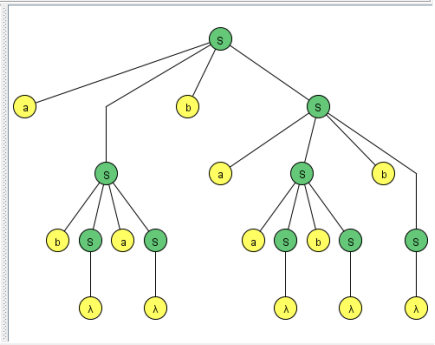
Input ababaabb

String accepted! 91 nodes generated.

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

Table Text Size

LHS		RHS
S	→	aSbS
S	→	bSaS
S	→	λ

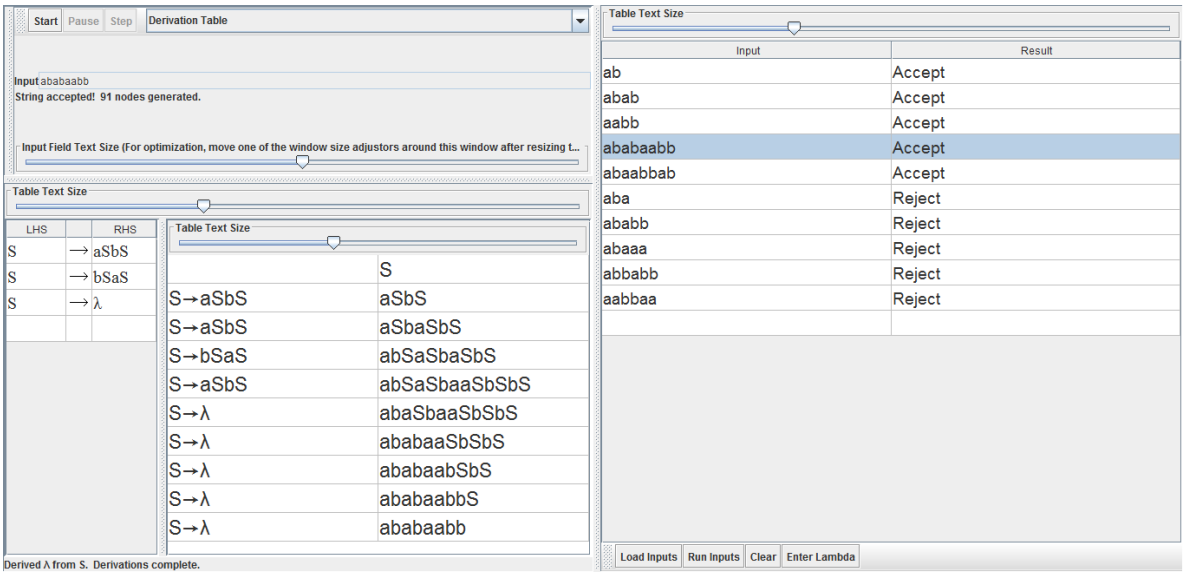


Derived λ from S. Derivations complete.

Table Text Size

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

Load Inputs | Run Inputs | Clear | Enter Lambda



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_ADPND.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