Instituto Politecnico Nacional

ESCOM “ESCUELA SUPERIOR DE CÓMPUTO”

TEORÍA COMPUTACIONAL

PRÁCTICA 8: TRANSORMACIÓN 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 unos ejercicios propuestos por la maestra, y por medio de la aplicación de JFLAP fuéramos limpiando paso a paso dicha gramática y probáramos algunas cadenas para esa gramática así cómo describir el lenguaje que genera cada una de las gramáticas, para posteriormente convertir la gramática a a su forma normal de Chomsky (FNC).

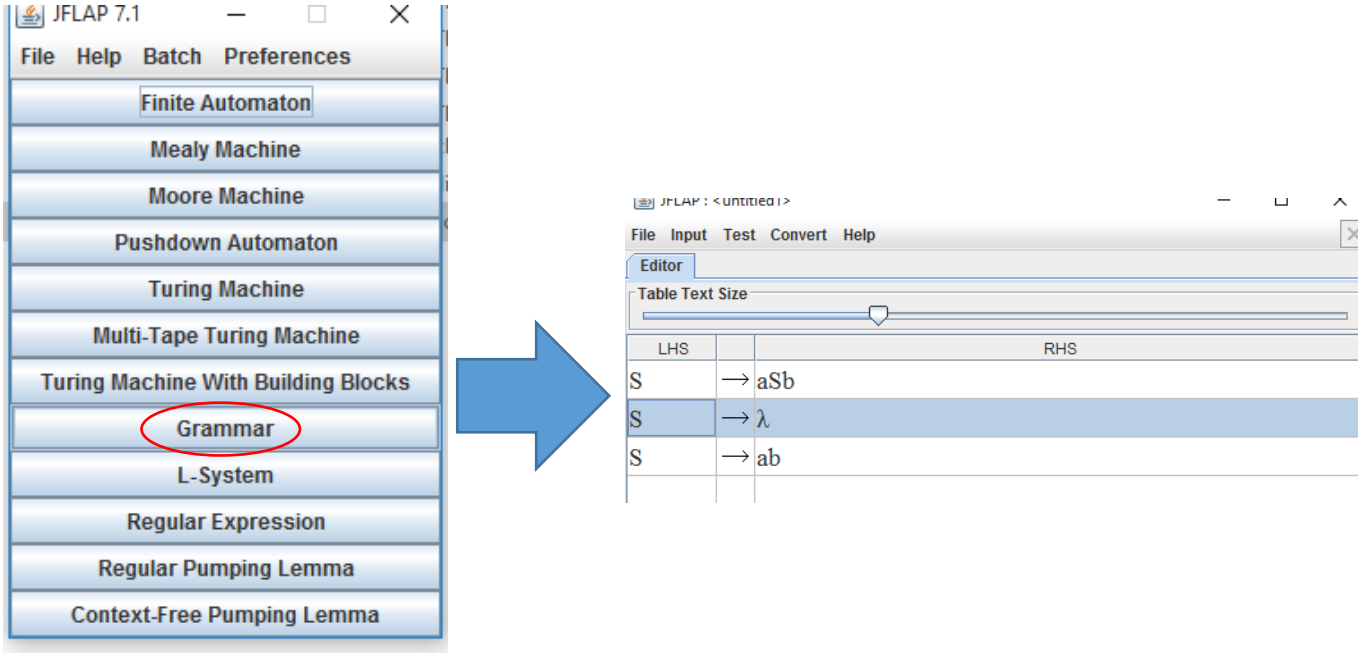
PLANTEAMIENTO DEL PROBLEMA

El problema que se presenta es que por medio de una gramática dada se tienen que limpiar y posteriormente convertir la gramática a su forma normal de Chomsky y probar cadenas que genera las derivaciones de las reglas de la gramática en su forma limpia y en su FNC.

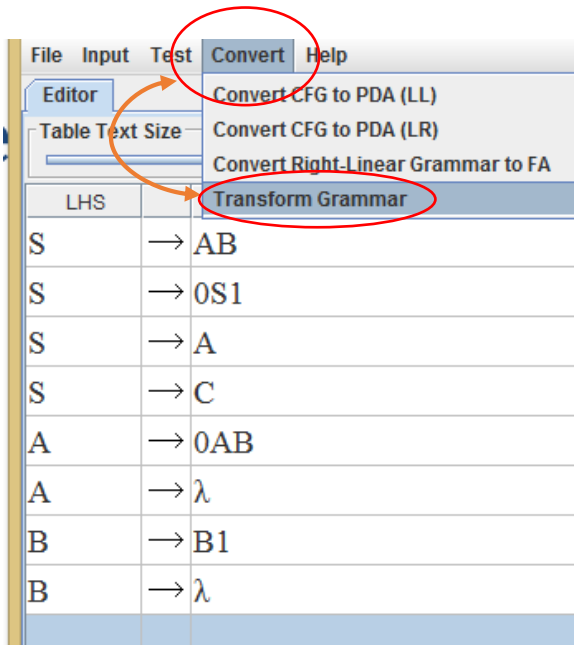
Para darle solución se optó por usar la herramienta de JFLAP ya que ofrece la opción para generar la gramática y paso a paso te dará la solución hasta que la gramática quede limpia y posteriormente se convierta a la forma normal de Chomsky

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.

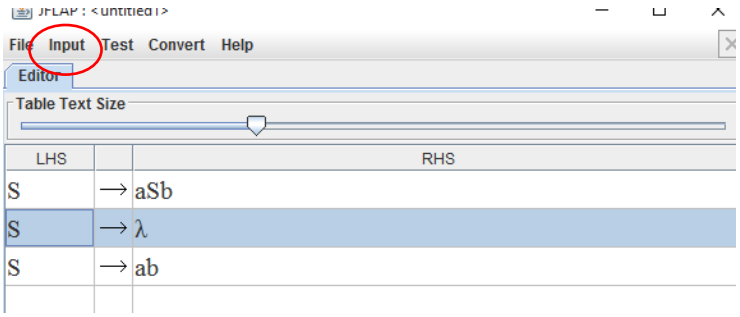


Una vez ingresada la gramática tenemos que ir a la pestaña de convert y posteriormente seleccionar la opción de convert to gramar para que empiece a limpiar la gramática.

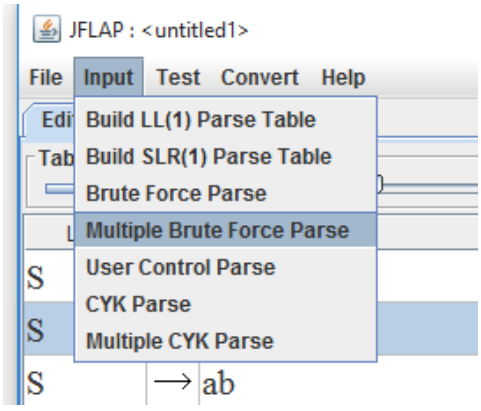


Una vez ahí, se selecciona la opción de do step para ir viendo paso a paso cómo se limpia la gramática ya que en la pestaña de arriba se dice que paso es el que estáoms realizando.

Posteriormente se selecciona la opción de export junto Do step para que podamos probar las cadenas pueden ser validas o no válidas. Para eso tenemos que ir a la parte de input.

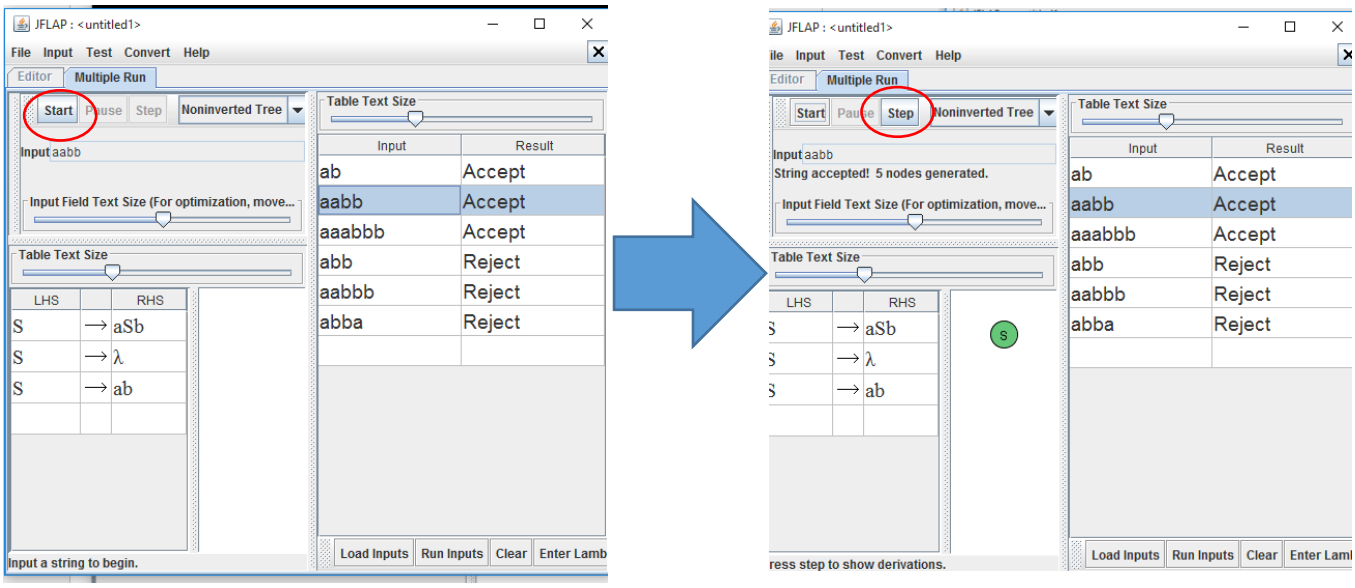


Una vez ahí se selecciona la opción de Multiple Brute Force Parse para agregar múltiples cadenas y verificar si son válidas o no válidas para la gramática declarada.

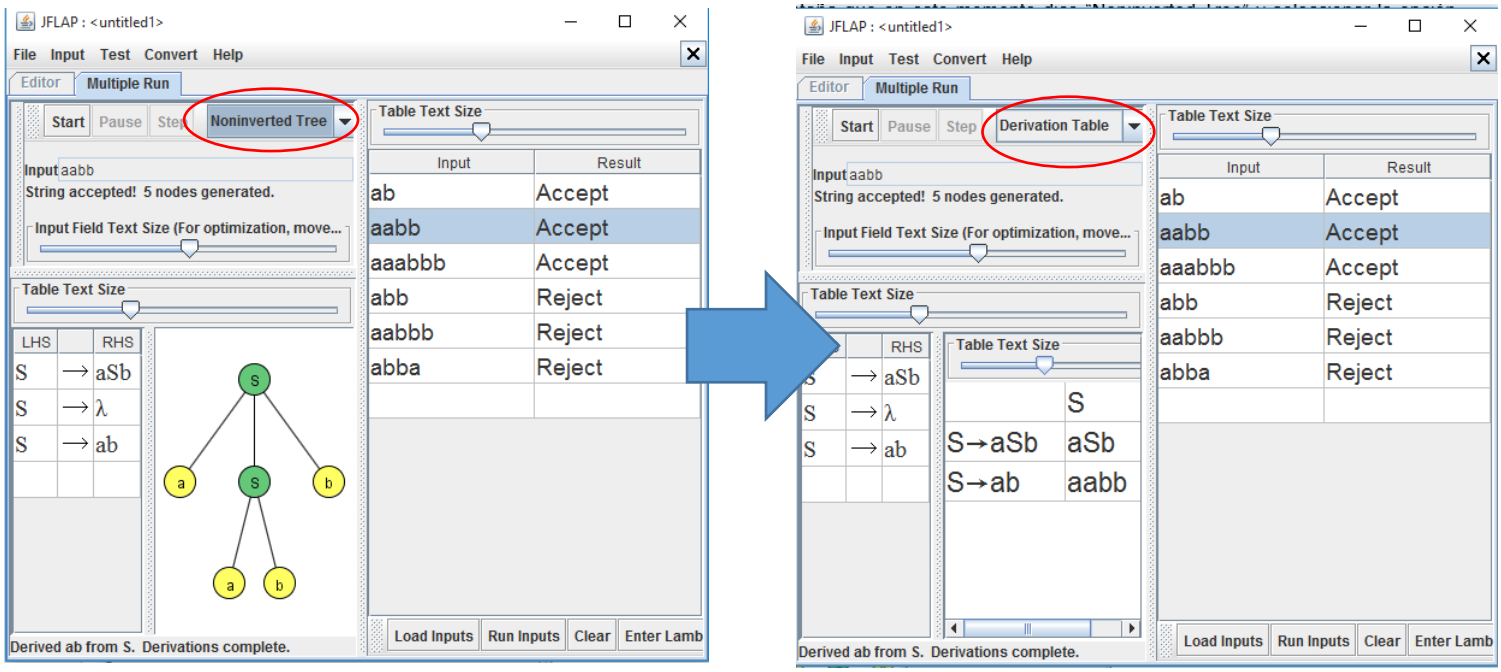


La pestaña generará una nueva ventana dentro de la aplicación para que se puedan ingresar las cadenas deseadas.

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

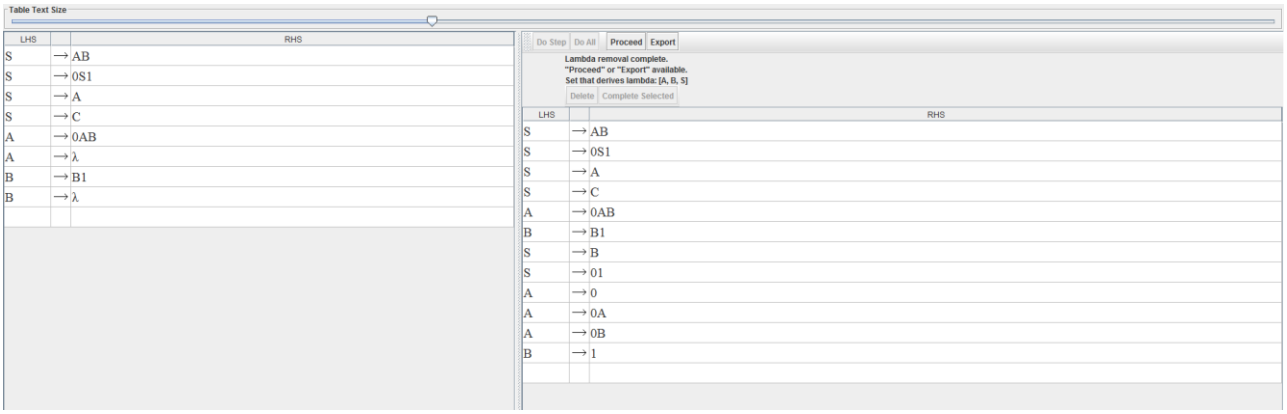
1. G=({0,1},{S,A,B,C},S,P)
   -S->AB | 0S1 | A | C
   -A->0AB | λ
   -B->B1 | λ
2. G=({i,+},{Z,E,F,G,P,Q,S,T},Z,P)
   -Z->E + T
   -E->E | S + F | T
   -F->F | FP | P
   -P->G
   -G->G | GG | F
   -T->T * i | i
   -Q->E | E + F | T | S
   S->i
3. S->Aab | B | CSa | b
   A->aA | Cb | a | aBAE
   B->bB | aBC | F | λ
   C->CG | DC
   D->aCb | a
   E->aaE | bB
   f->aF | ab
   G->F
4. S->aSb | aAb
   A->BA | λ | aCB
   B->ASb | abB
   C->aDF | aDb
   D->abC | aCB | aF
   F->Fb | aCb
5. S->A | AA | AAA
   A-> ABa | ACa | a
   B->ABa | Ab| λ
   C->Cab | CC
   D->CD | Cd | CEa
   E->b
6. S->aAb | cHB |CH
   A->dBH | eeC
   B->ff | D

C->gFB | ah
D->i
E->jF
F->deGGG | eF
G->kF
H->Hlm

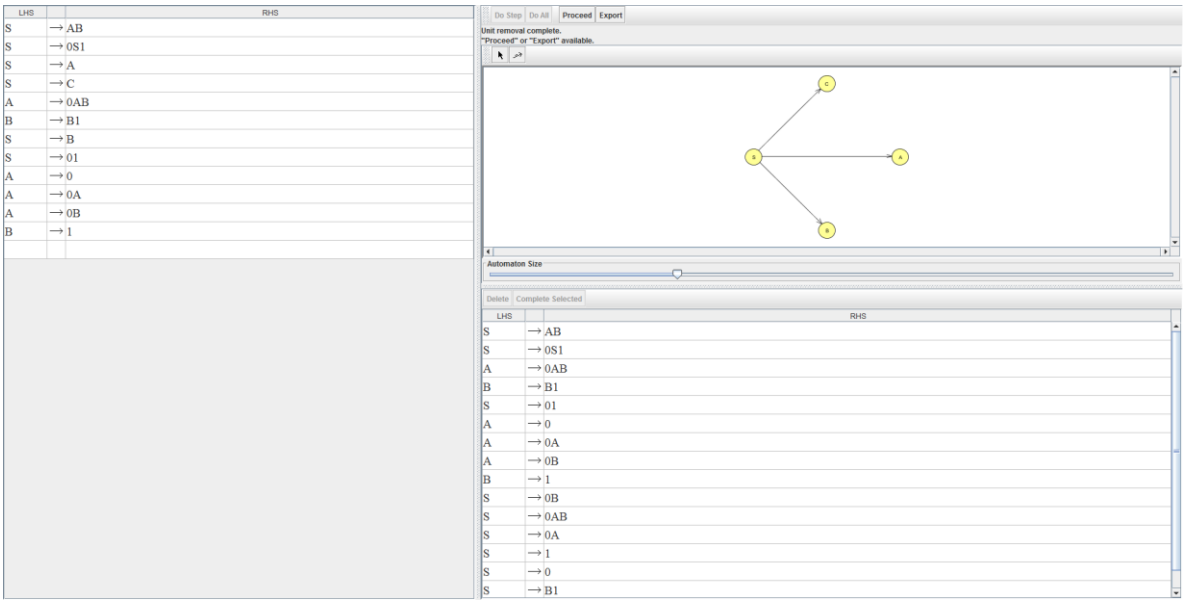
7. S->aAB | bBA | ABb | SS
   A->aAb | CCA | BB
   B->λ | bC
   C->aCS | SCS
   D->ab | SABC
8. S->CBa | D
   A->bbC
   B->Sc | ddd
   C->eA | f | C
   D->E | SABC
   E->gh

FUNCIONAMIENTO

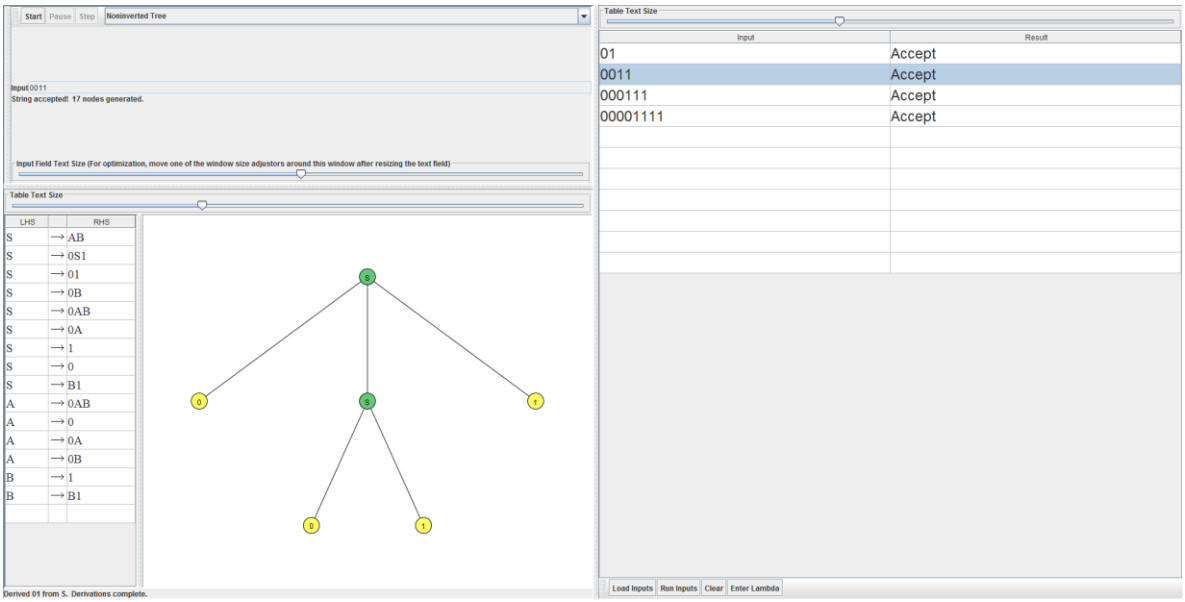
1. G= ({0,1}, {S, A, B, C}, S, P)
   -S->AB | 0S1 | A | C
   -A->0AB | λ
   -B->B1 | λ



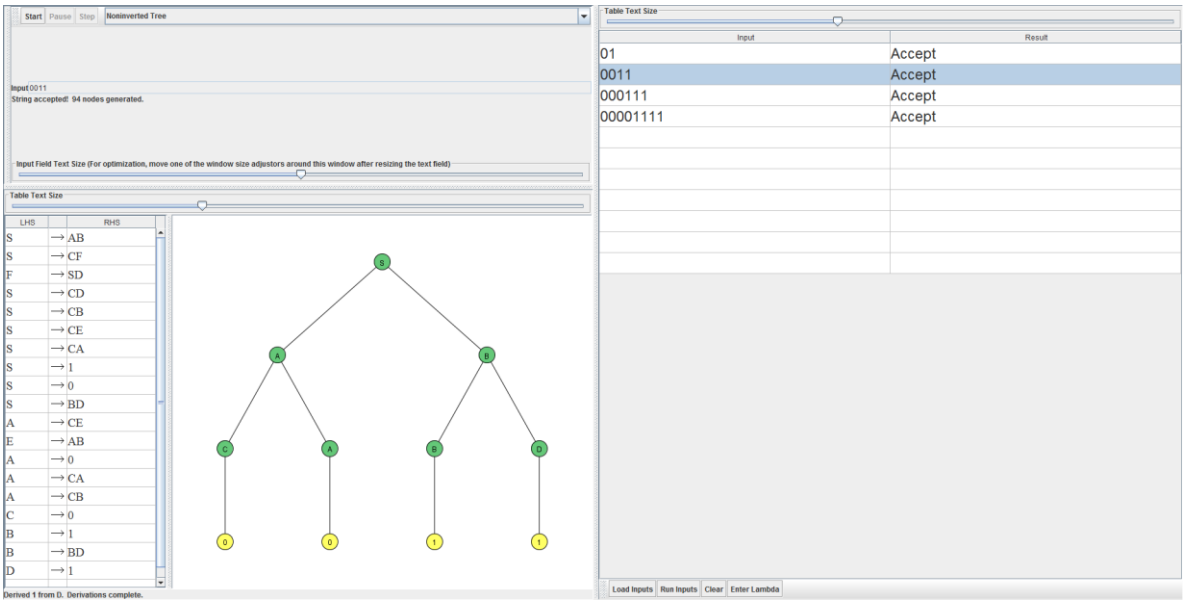
Remover lamda



Remover unitarios



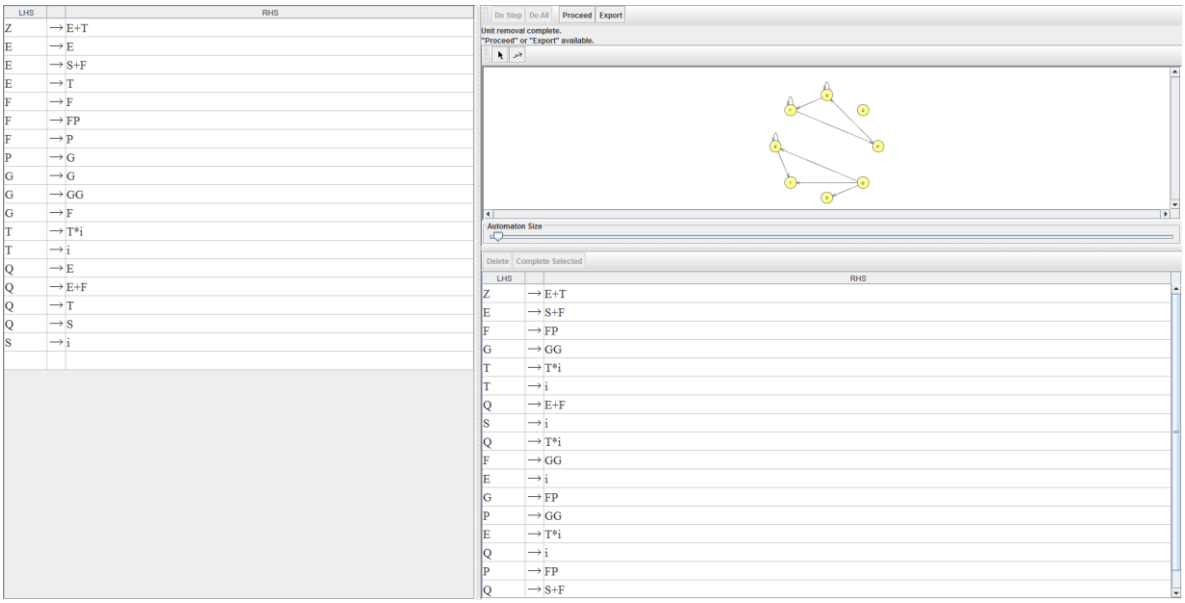
Gramática limpia



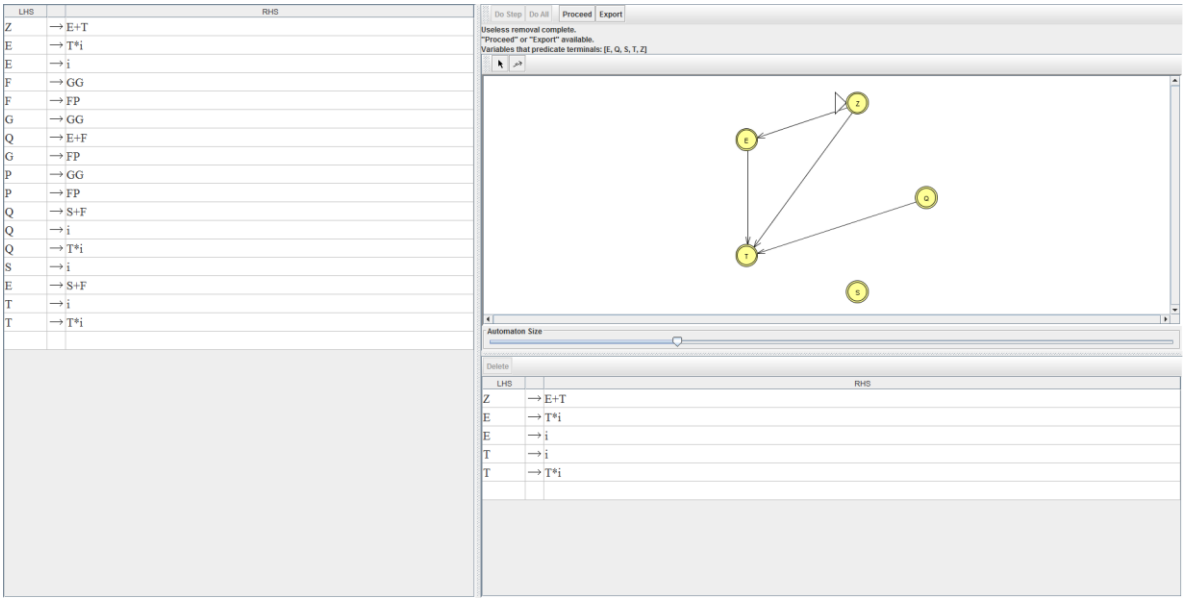
Forma normal de Chomsky

L= $\{0^n1^m \mid n=m=0,1,2,3,\ldots\}$

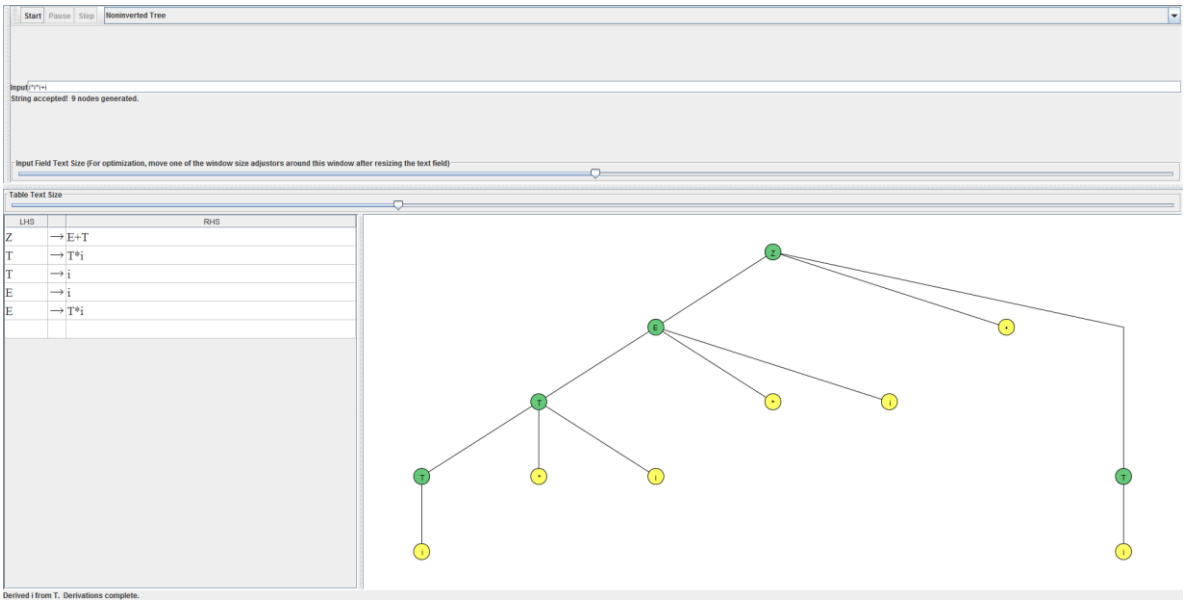
2. G= ({i, +}, {Z, E, F, G, P, Q, S, T}, Z, P)
   -Z->E + T
   -E->E | S + F | T
   -F->F | FP | P
   -P->G
   -G->G | GG | F
   -T->T * i | i
   -Q->E | E + F | T | S
   S->i



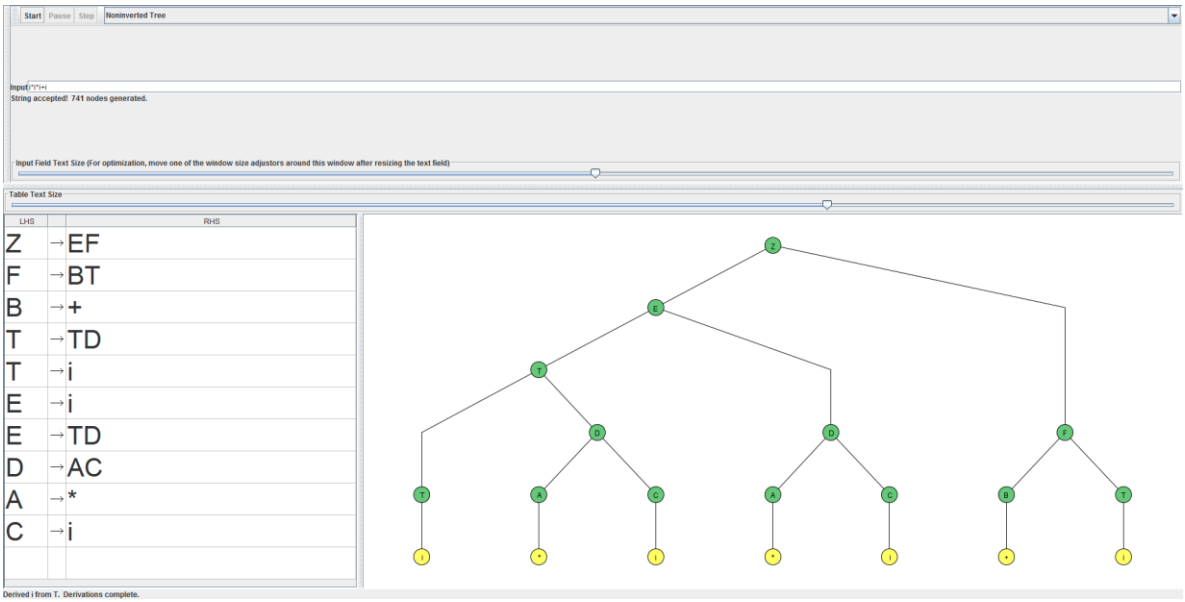
Elimina unitarios



Elimina los inaccesibles



Gramática Limpia

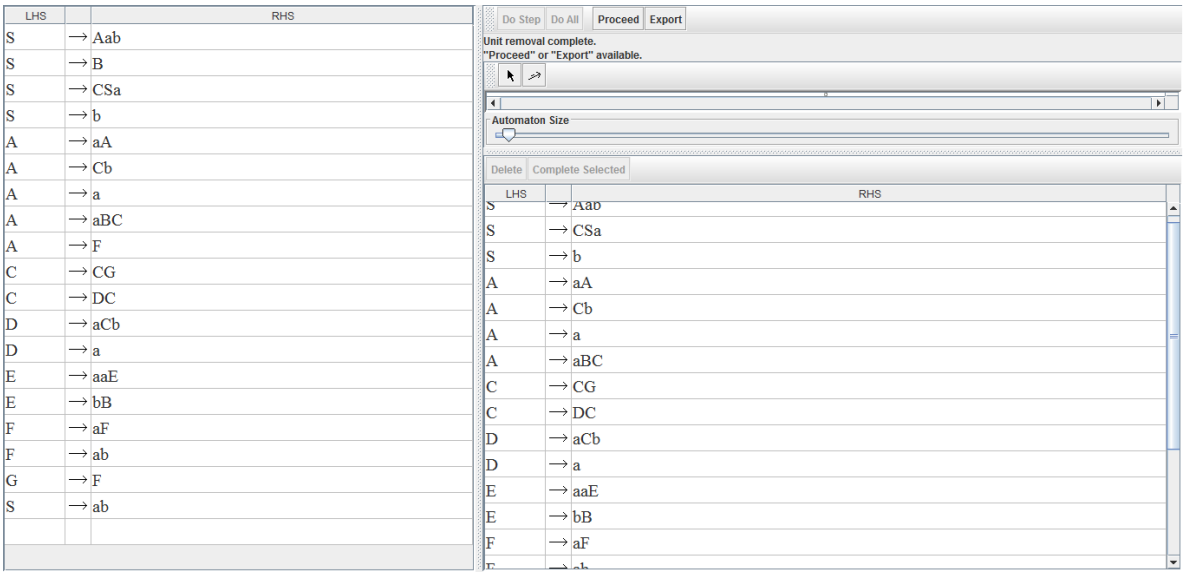


FNC

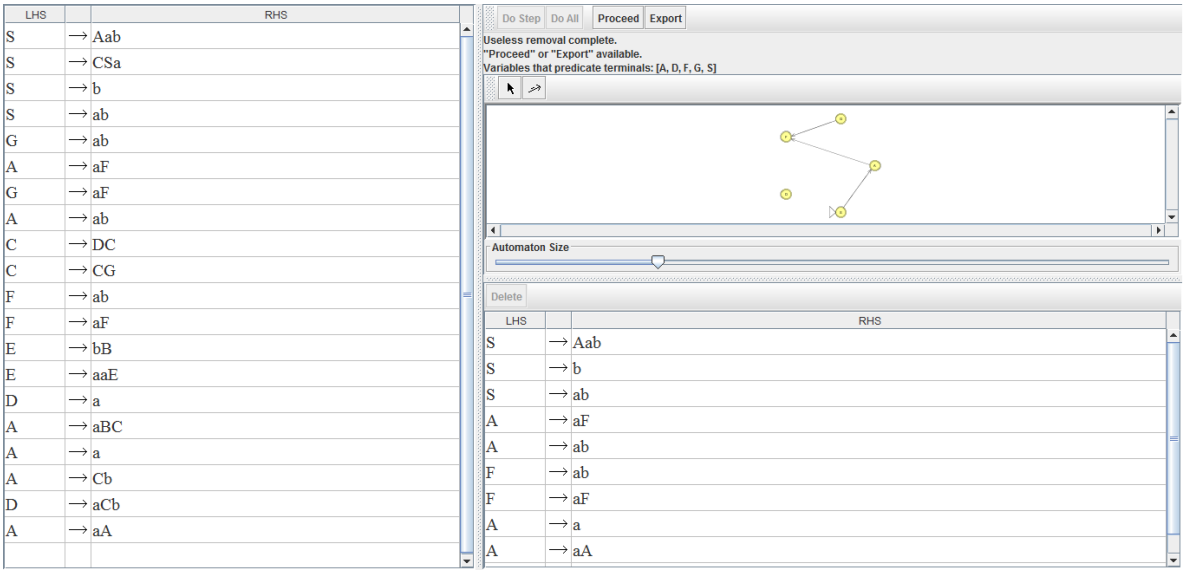
L= {i*i,i+i,i+*i,…}

3. S->Aab | B | CSa | b
   A->aA | Cb | a | aBAE
   B->bB | aBC | F | λ
   C->CG | DC
   D->aCb | a
   E->aaE | bB

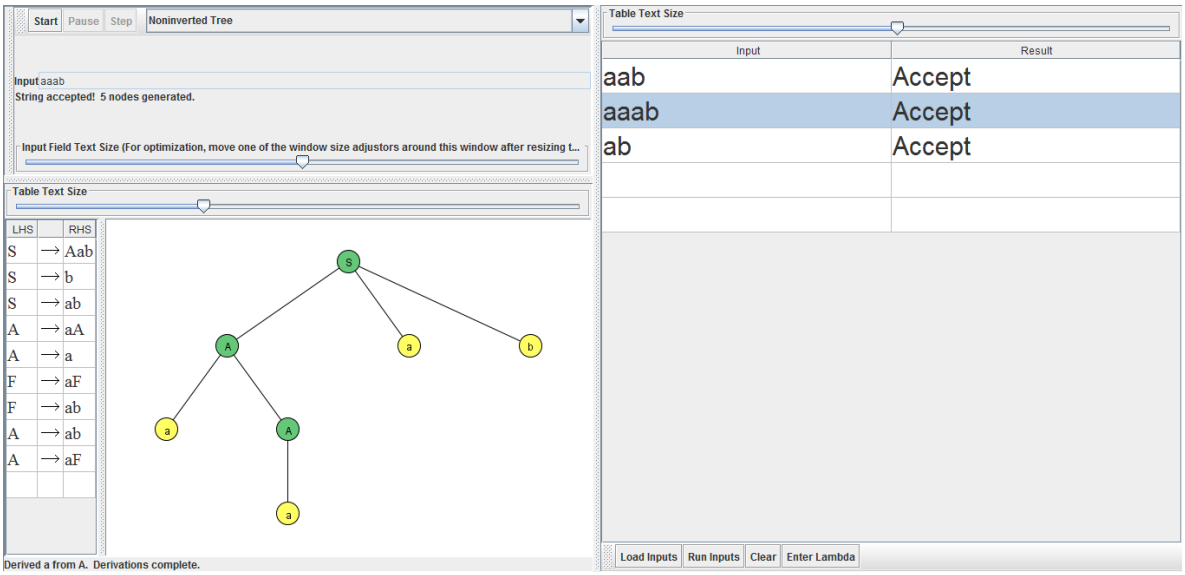
f->aF | ab
G->F



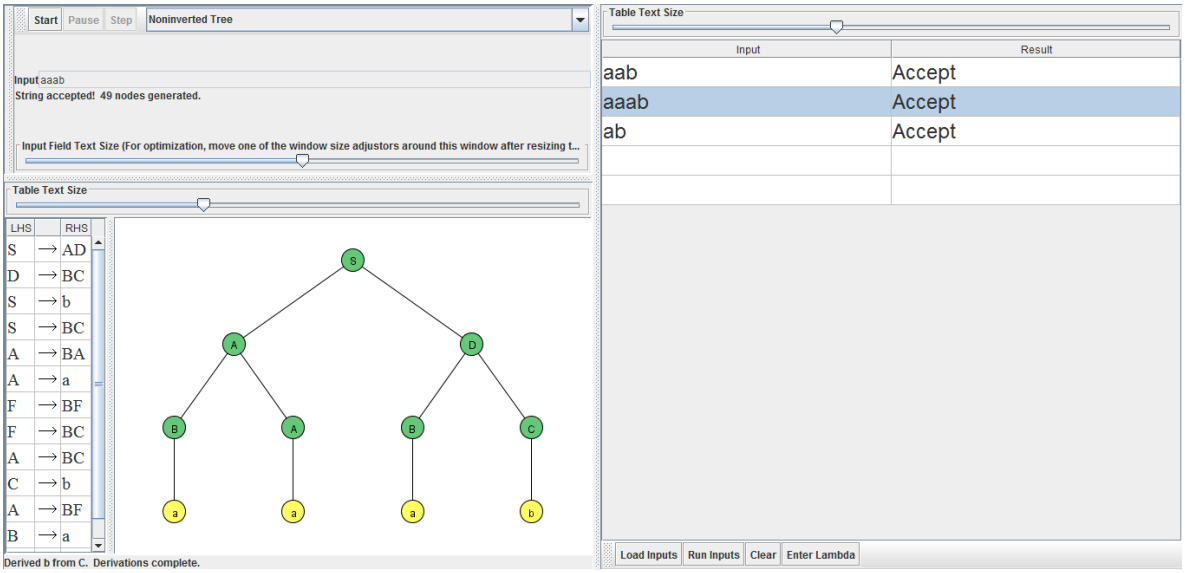
Elimina unitarios



Elimina inaccesibles



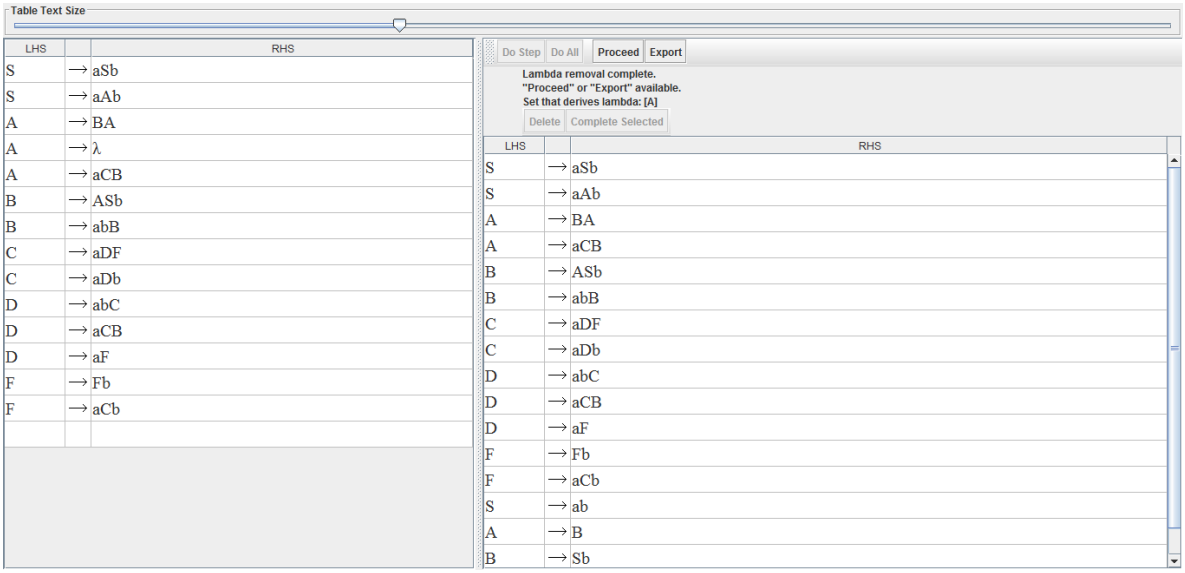
Gramática Limpia



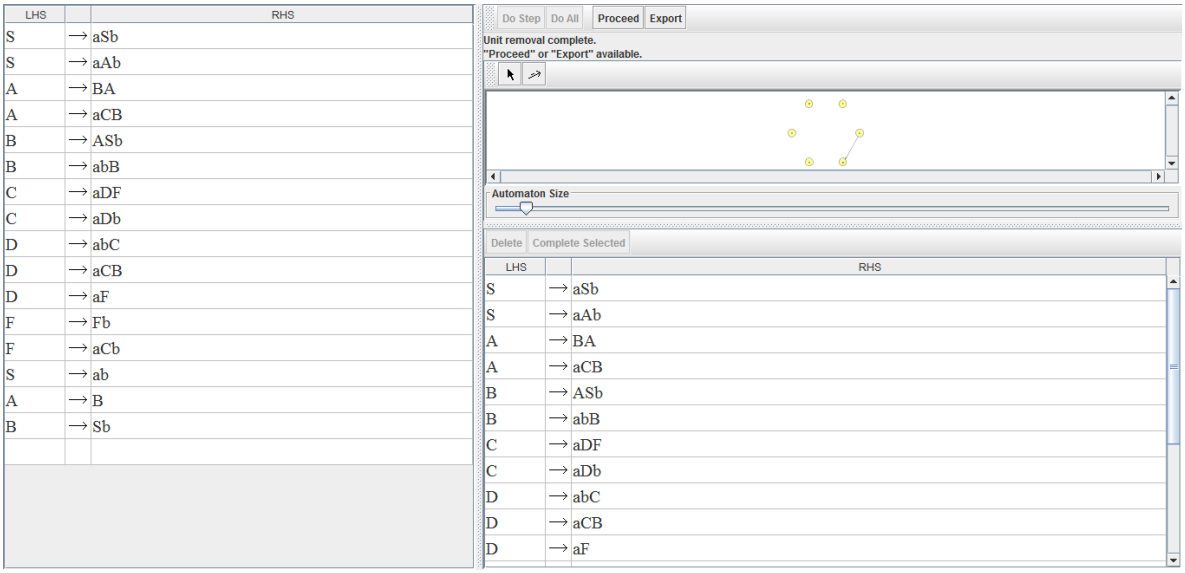
FNC

L= $\{a^n b \mid n=0,1,2,\ldots\}$

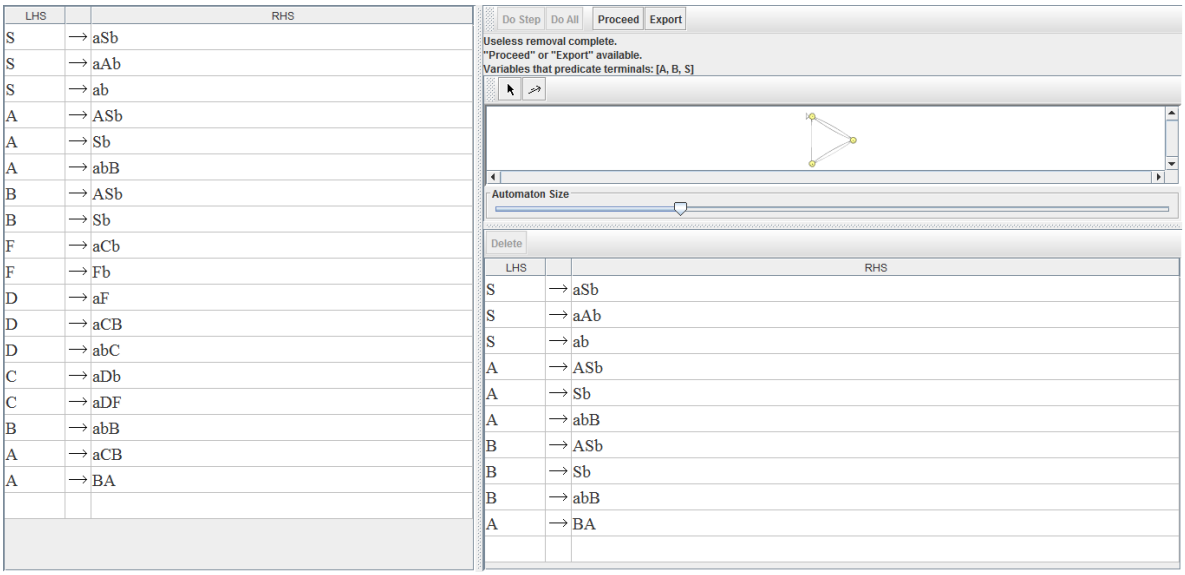
4. S->aSb | aAb
   A->BA | λ | aCB
   B->ASb | abB
   C->aDF | aDb
   D->abC | aCB | aF
   F->Fb | aCb



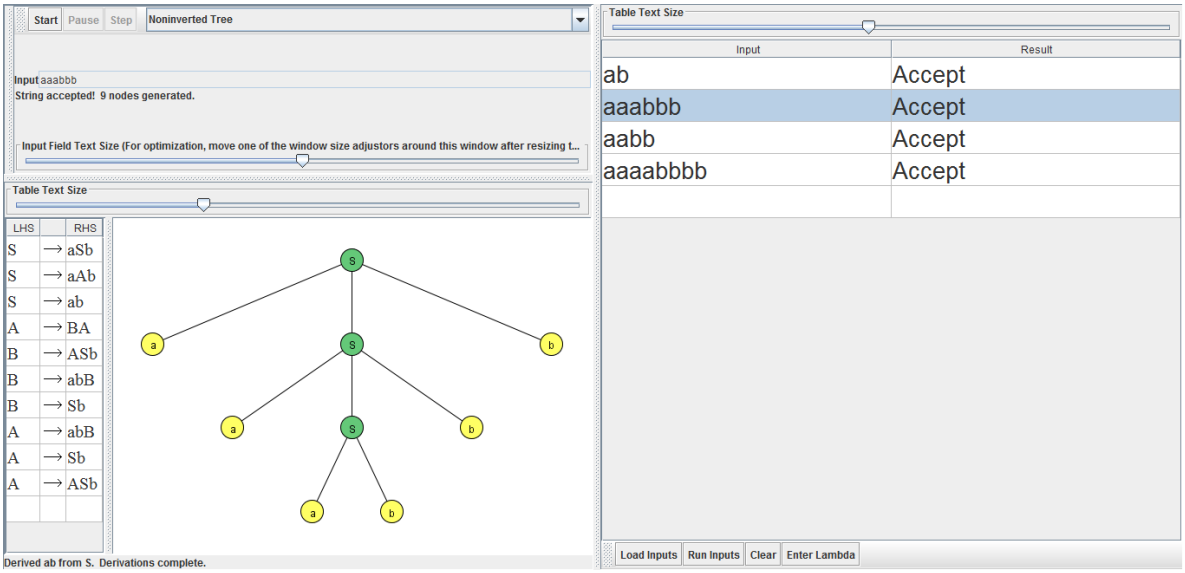
Elimina cadenas vacías



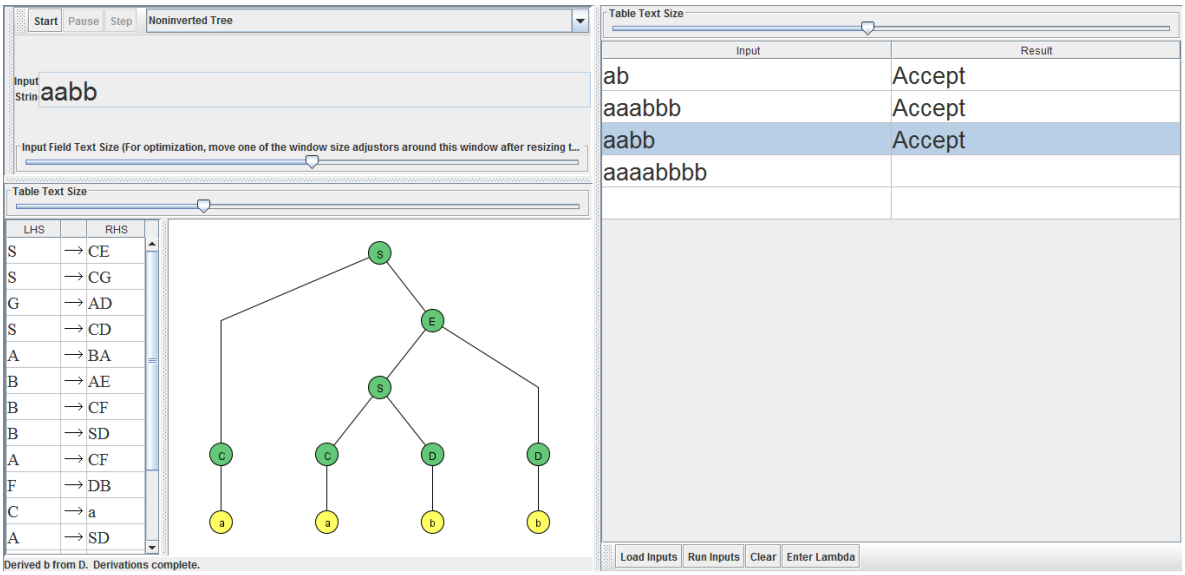
Elimina Unitarios



Elimina inaccesibles



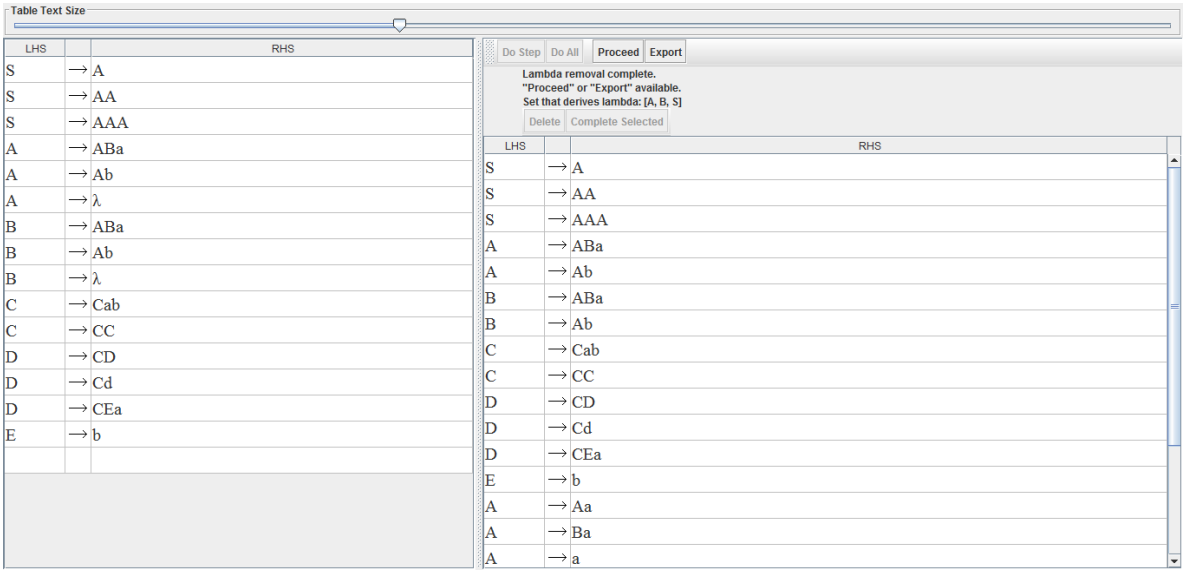
Gramática Limpia



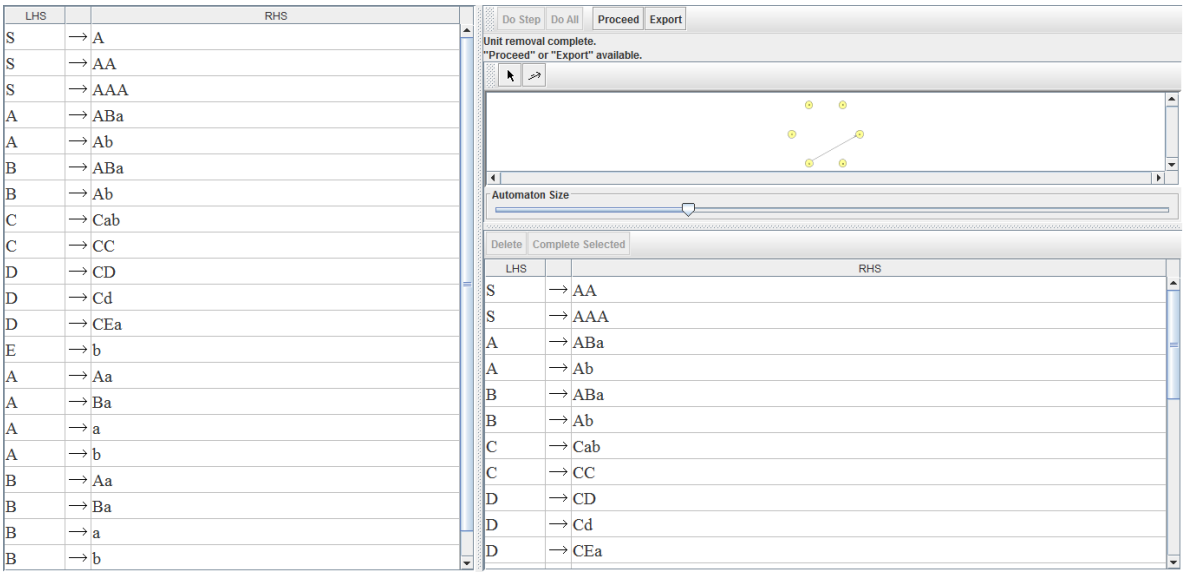
FNC

L= {$a^n b^m$ | n=m=1,2,3,…}

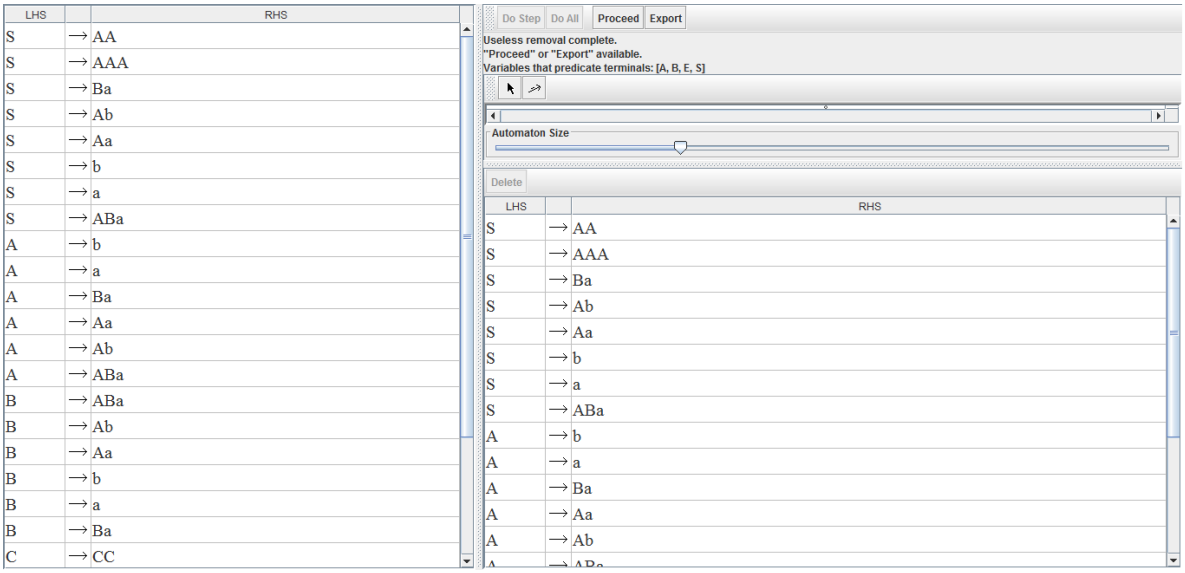
5. S->A | AA | AAA
   A-> ABa | ACa | a
   B->ABa | Ab| λ
   C->Cab | CC
   D->CD | Cd | CEa
   E->b



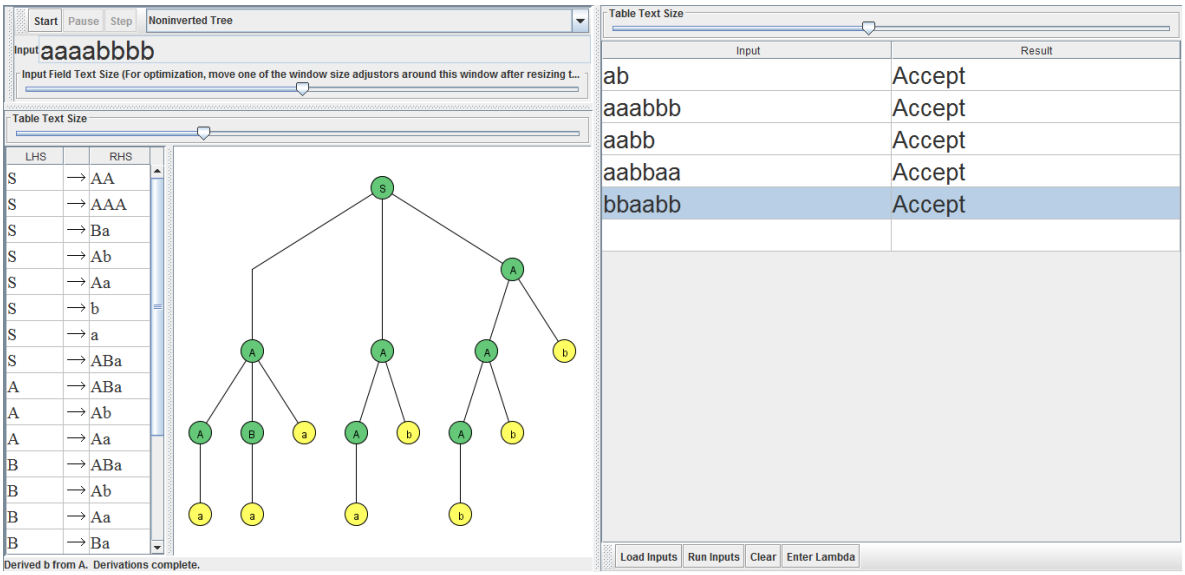
Elimina Cadenas Vacías



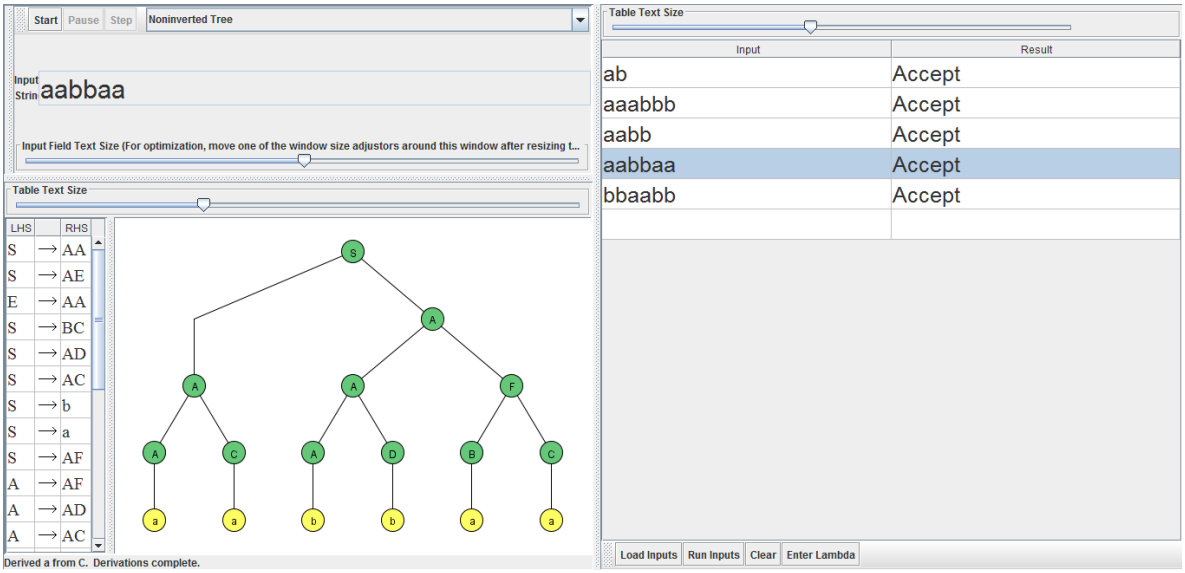
Elimina Unitarios



Elimina Inaccesibles



Gramática limpia

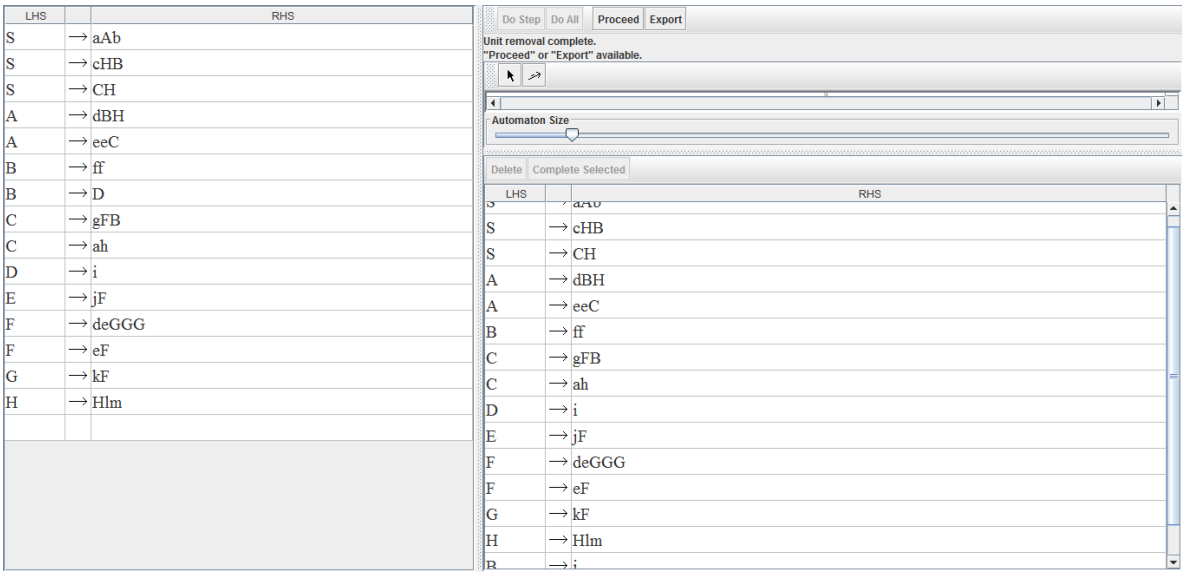


FNC

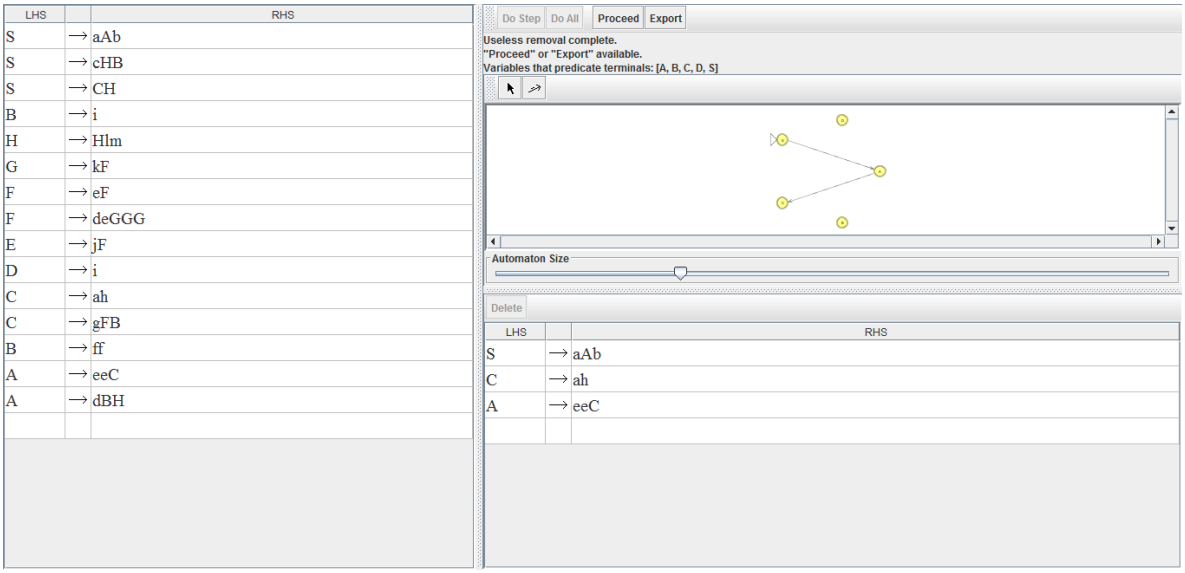
L= {$a^n b^m$, $b^n a^m$, $a^n b^m a^n$, $b^n a^m, b^n$ | n=m=1,2,3,…}

6. S->aAb | cHB |CH
   A->dBH | eeC
   B->ff | D
   C->gFB | ah
   D->i
   E->jF
   F->deGGG | eF

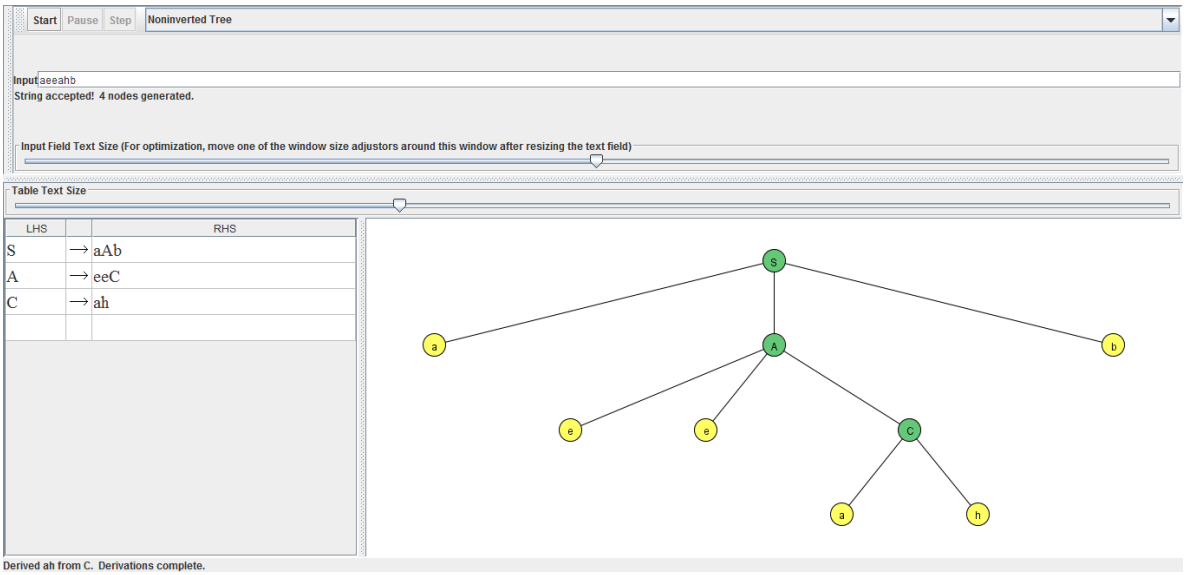
G->kF
H->Hlm



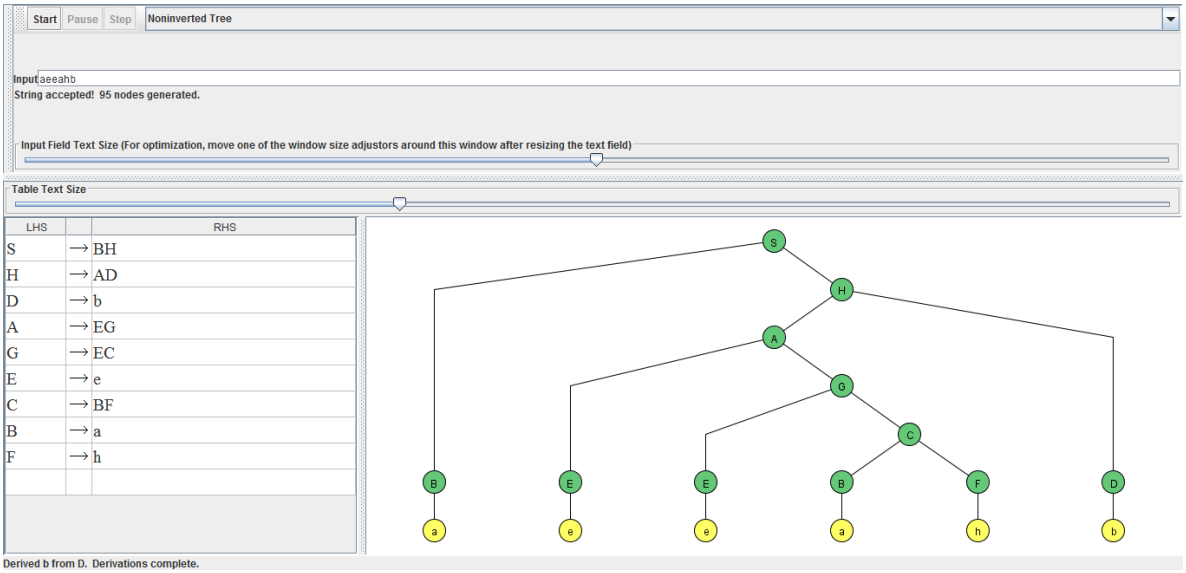
Elimina Unitarios



Elimina inaccesibles



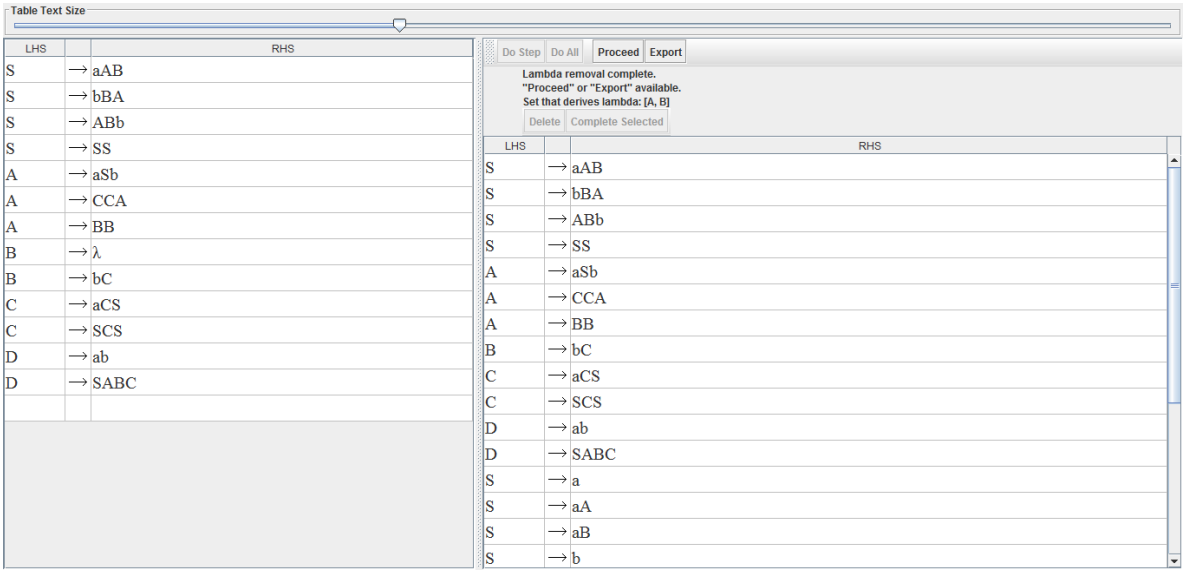
Gramática limpia



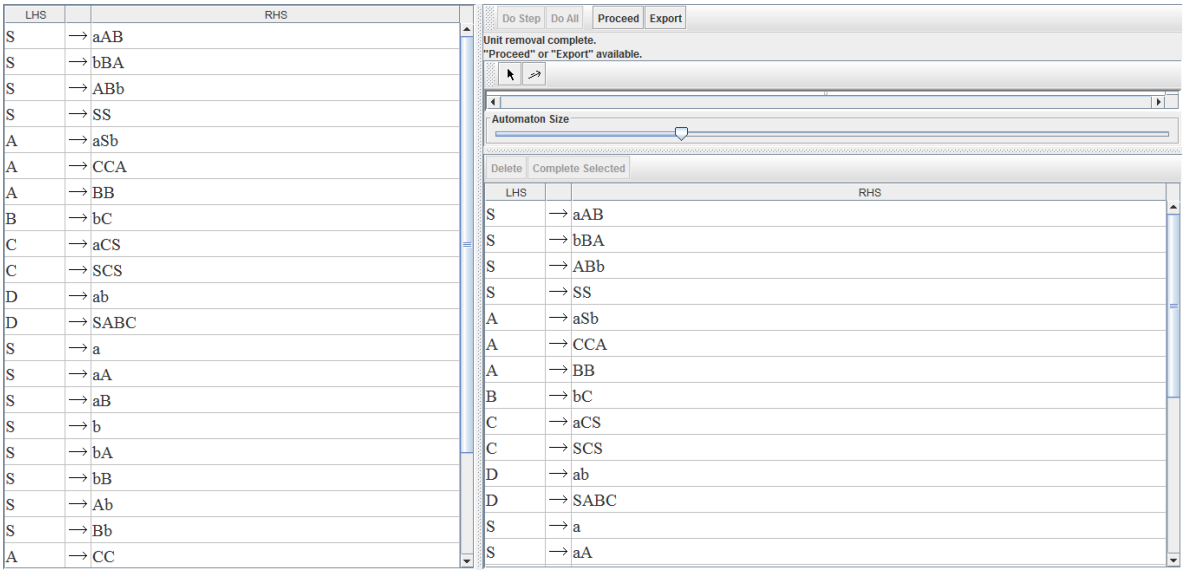
FNC

L= {aeeahb} (Sólo acepta una cadena)

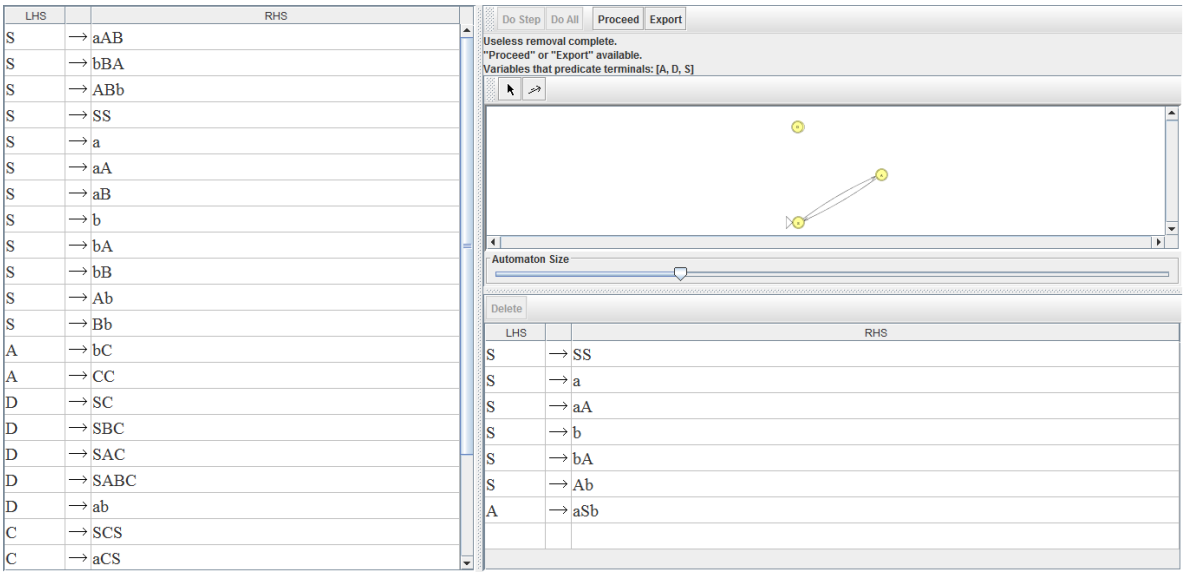
7. S->aAB | bBA | ABb | SS
   A->aAb | CCA | BB
   B->λ | bC
   C->aCS | SCS
   D->ab | SABC



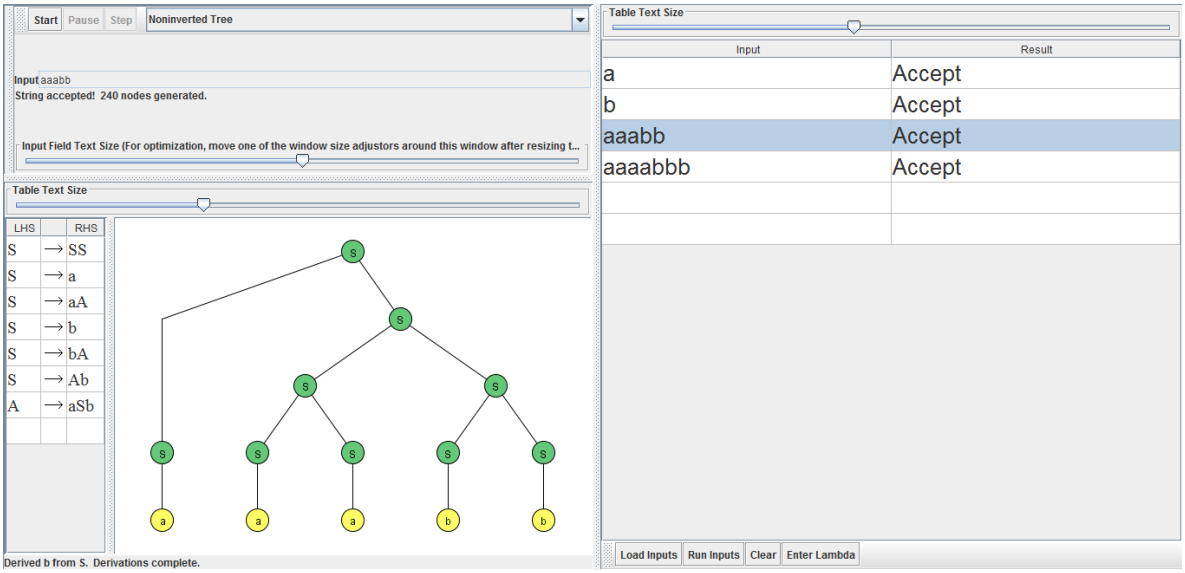
Elimina cadenas vacías



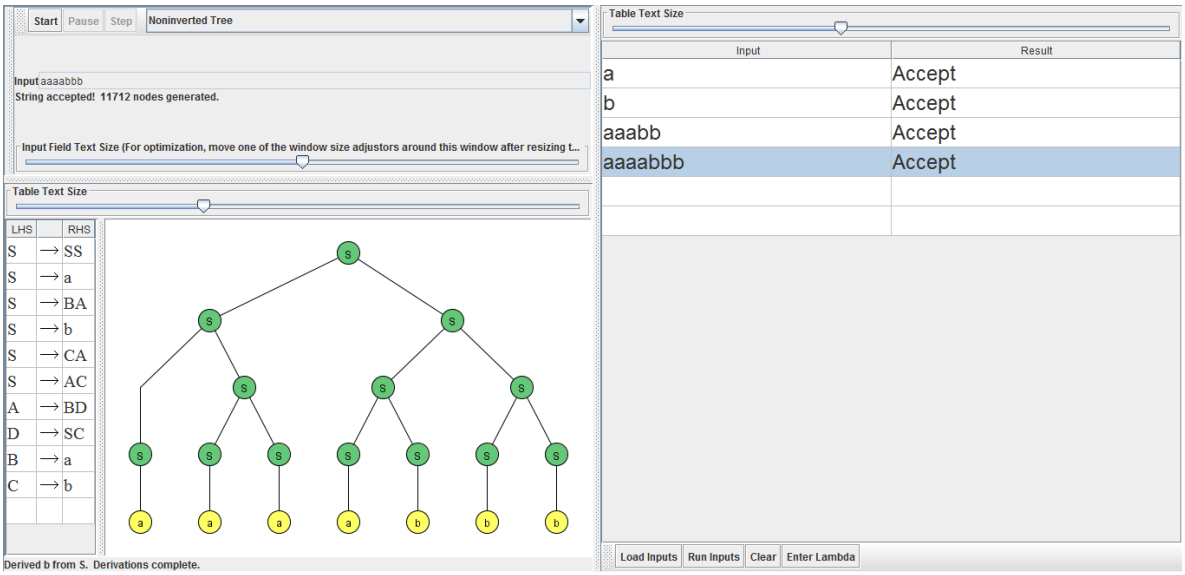
Elimina Unitarios



Elimina Inaccesibles



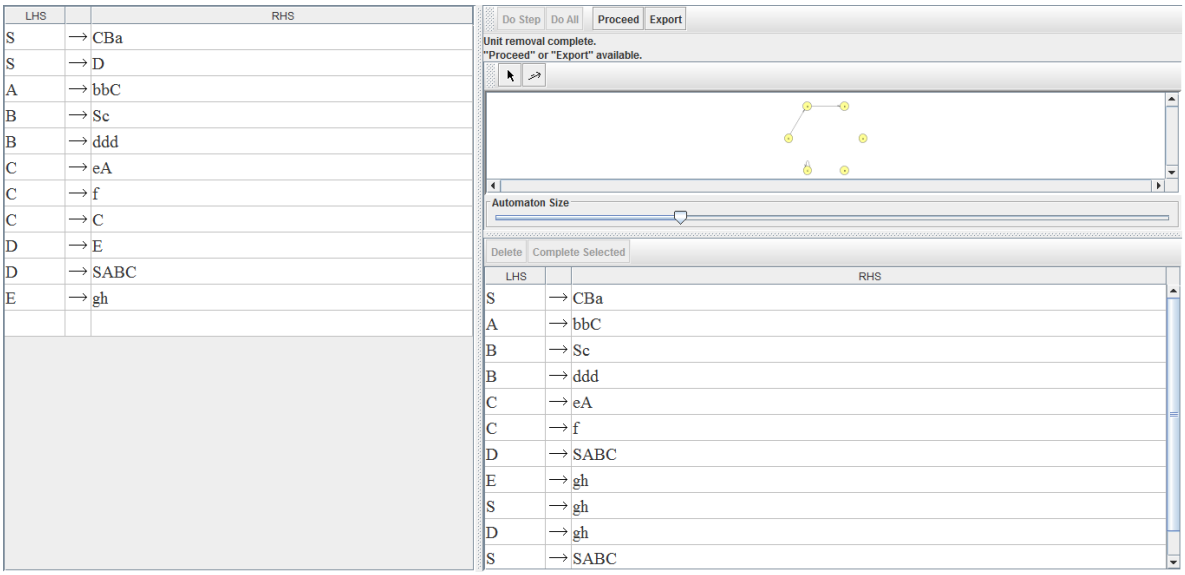
Gramática limpia



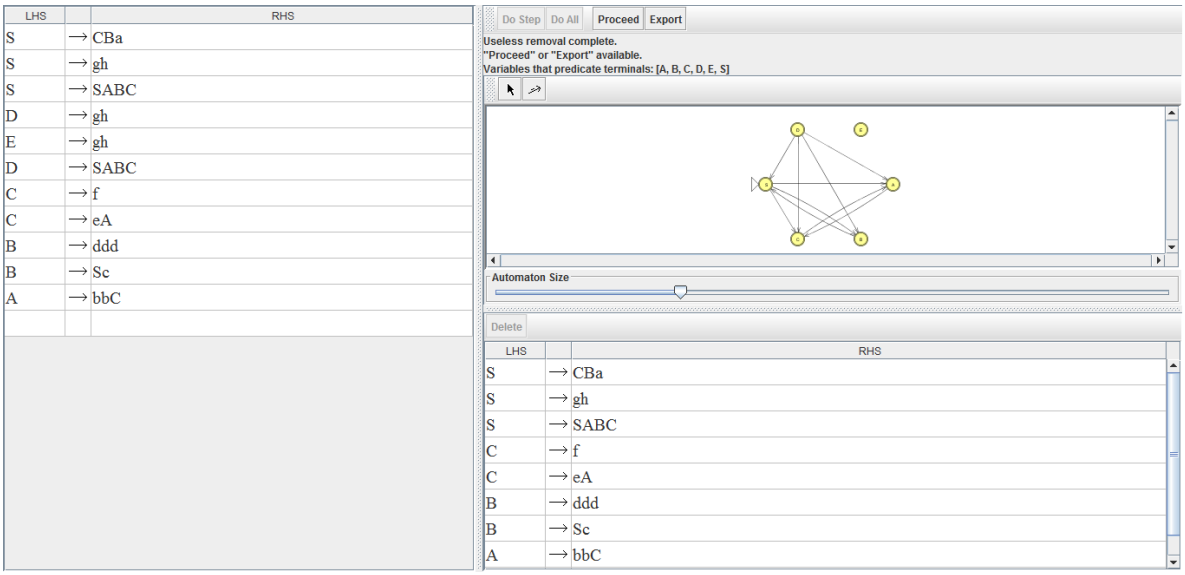
FNC

L= {$a^n b^m$ | n>m, n=1,2,3,…,m=1,2,3,…}

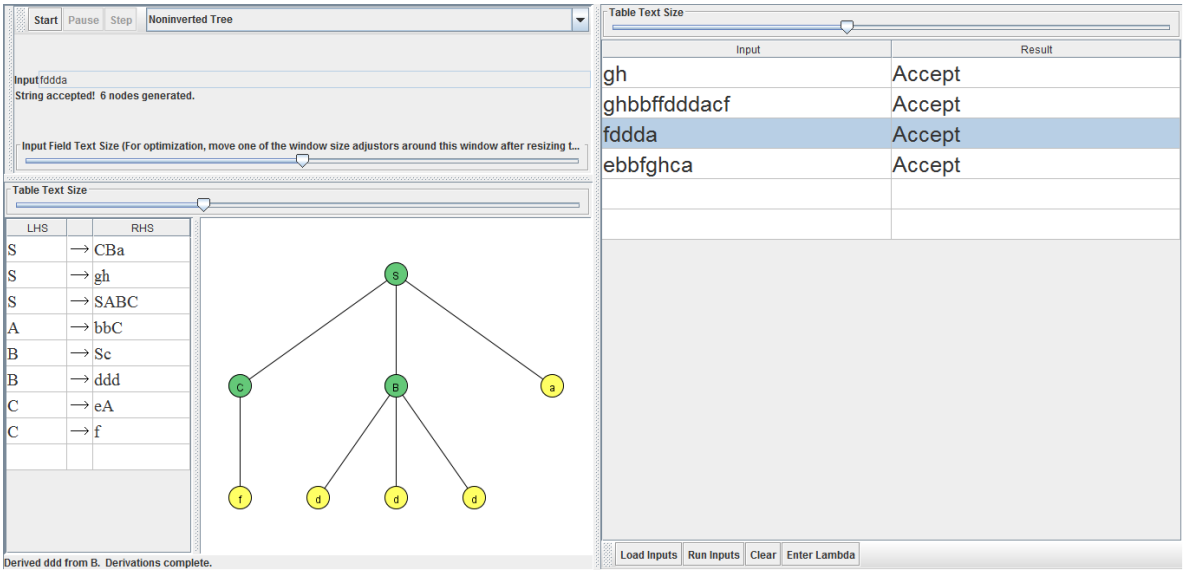
8. S->CBa | D
   A->bbC
   B->Sc | ddd
   C->eA | f | C
   D->E | SABC
   E->gh



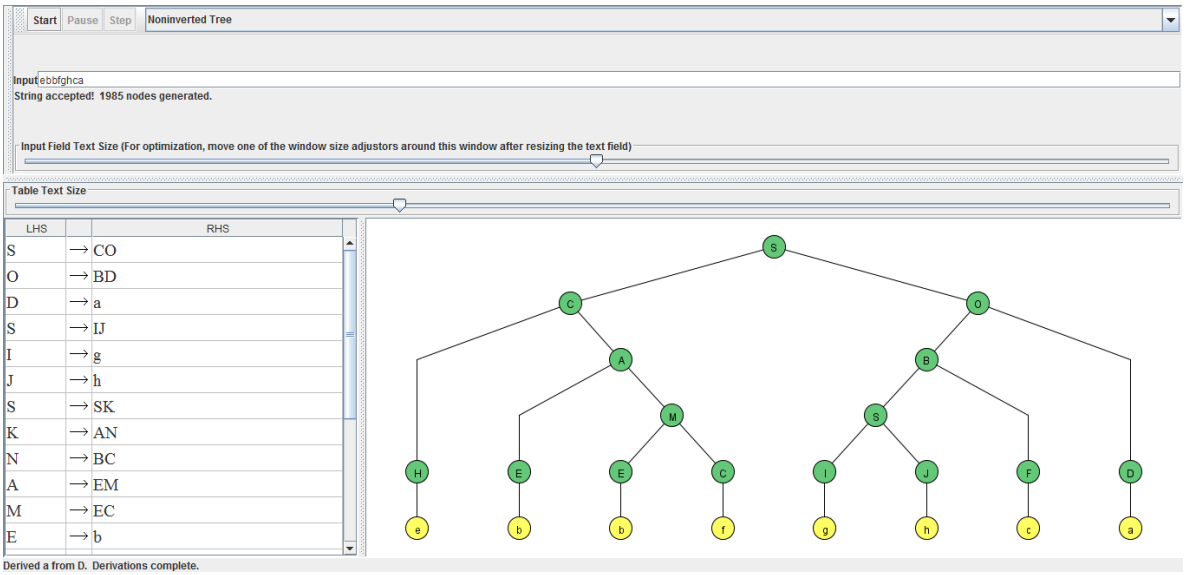
Elimina Unitarios



Elimina inaccesibles



Gramática limpia



FNC

L) {ghbbffdddacf,fddda,ebbfghca,…}

CONCLUSIÓN

En ésta práctica se realizó la conversión paso a paso para limpiar una gramática, y posteriormente convertirla en su forma normal de Chomsky, me di cuenta que visualizarlo paso a paso abre muchas posibilidades a cómo lo harías a papel, ya que aunque el procedimiento es el mismo, los pasos que se hace en el programa a como se hacen analíticamente son algo distintos, pero siempre se va a llegar a la misma conclusión o lo que es lo mismo a la misma gramática limpia.

BIBLIOGRAFÍA

1) http://repositori.uji.es/xmlui/bitstream/handle/10234/5875/bolAuto5.pdf;jsessionid=44C764C197A3310BFCA622A5D5BF5B54?sequence=6
2) http://teodelacomp.blogspot.com/2011/03/forma-normal-de-chomsky.html