INFG-TALF Teoría de Autómatas y Lenguajes Formales

Ejercicios de Lenguajes y Gramáticas (T4-P2)

3. Realizar las transformaciones necesarias del proceso de limpieza de gramáticas, para obtener una gramática limpia G' equivalente a la G dada.
   G = ({a,b,c,d}, {X,Y,Z,O,P,Q,A}, Z, P),
   P = {Z::=Z, Q::=OP, X::=aa, Z::=aX, Y::=aa, Z::=Ya, O::=b, Z::=aaa, P::=QO,
   Q::=d, P::=c, O::=PQ}

6. Dada la gramática G LD, obtener una G' LI equivalente.
   G=({0,1},{A,S},S,P)
   P={ S ::= 1 | A1; A ::= S0}

7. Obtener una gramática en Forma Normal de Chomsky (FNC) equivalente a la siguiente:
   G = ({a,b,c},{S, Q, R}, S, P)
   P = { S ::= Rba | Q
   Q ::= Qb | b
   R ::= cRQb | cb
   }

8. Dada la gramática G:
   G = ({e,f,g,z,a,b,d}, {Y, X, E, A, D, I, G}, A, P),
   P = { 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 }

   a) Transformar a FNC, explicando cada paso realizado.
   b) Determinar si las palabras 'abz' y 'azdbb' pertenecen al lenguaje generado por G. En caso afirmativo, generar un árbol de derivación para dicha palabra. En caso negativo, justificar la no pertenencia.

9. Obtener una gramática en FNC equivalente a la siguiente:
   G = ({a, b, c},{S, A, B, C, D, E), S, P)
   P = { S::=AaB | Cbb | B
   A::=Aa | cD
   B::=a | Ba | λ
   C::=Sa | a | abB
   D::=aaA
   E::=aa }

12. Sea la gramática G, hallar su FNG equivalente.
G = ({1,2}, {A,B,C,D,E}, A, P),
donde P = {A ::= CD | EB | λ
B ::= BC | 1
C ::= 2
D ::= BC
E ::= 1}

13. Sea la gramática G, hallar su FNG equivalente.
G = ({a,b,c,d,e,0,1}, {S,A,B}, S, P),
P = { S → AaB | AaC
A → Ab | Ac | b | c
B → BdC | 0
C → CeB | 1 }

Ejercicios de Lenguajes y Gramáticas (T4-P2) SOLUCIONES

3. Realizar las transformaciones necesarias del proceso de limpieza de gramáticas, para obtener una gramática limpia G' equivalente a la G dada.
G = ({a,b,c,d}, {X,Y,Z,O,P,Q,A}, Z, P),
P = { Z::=Z, Q::=OP, X::=aa, Z::=aX, Y::=aa, Z::=Ya, O::=b, Z::=aaa, P::=QO, Q::=d, P::=c, O::=PQ}

Solución:

G' = ({a}, {X,Y, Z}, Z, P'), P' = {Z::=aaa, Z::=aX, Z::=Ya, X::=aa, Y::=aa}

La gramática de entrada tiene una regla innecesaria asociada al axioma (Z::=Z).
Los símbolos inaccesibles son:

- No Terminales: O, P, Q
- Terminales: b, c, d (Estos símbolos forman parte de las producciones asociadas a los NT que se eliminan).

No hay reglas supérfluas.

6. Dada la gramática G LD, obtener una G' LI equivalente.
G=({0,1},{A,S},S,P)
P={ S ::= 1 | A1; A ::= S0}

Solución:

Primero, como hay axioma inducido (axioma en la parte derecha) en la tercera regla, por lo que antes de aplicar el algoritmo de conversión G3LI en G3LD hay que eliminar el axioma inducido, obteniendo su gramática equivalente.

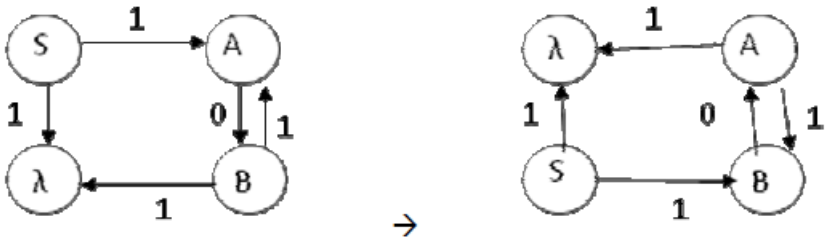
Se añade un nuevo NT => B ∈ Σ_N.

Se **añade** una regla B::=x, para cada regla S::=x (pero no se eliminan las reglas del axioma):
S ::= 1 | A1 → B::=1 | A1

Se sustituye el axioma en la parte derecha (en todas las reglas que aparezca) por el nuevo símbolo NT añadido:
~~A ::= S0~~ → A::=B0

La gramática resultante equivalente sin axioma inducido es:
G' = ({a,b}, {A,S,B}, S, P'), P'={S::=1 | A1, A::=B0, B::=1 | A1}

Ahora se puede pasar a convertir en LD, construyendo el grafo correspondiente, e invirtiendo la orientación de las conexiones:



Reescribiendo las producciones leyendo el grafo de la derecha, la gramática G LD equivalente resultante es:
G LI = ({a,b}, {A,S,B}, S, P), P={S::=1|1B, B::=0A, A::=1|1B}

7. Obtener una gramática en Forma Normal de Chomsky (FNC) equivalente a la siguiente:

G = ({a,b,c},{S, Q, R}, S, P)

P = { S ::= Rba | Q
Q ::= Qb | b
R ::= cRQb | cb
}

Solución:

G no es bien formada pues tiene una regla de redenominación. La gramática bien formada tendría las siguientes producciones:

P ::= Rba | Qb | b
Q ::= Qb | b
R ::= cRQb | cb

Una posible solución en FNC sería:

G = ({a,b,c},{A,B,C,D,E,F, S, Q, R}, S, P)
P = { A ::= a
B ::= b
C ::= c
S ::= RD | QB | b
Q ::= QB | b
R ::= EF | CB
D ::= BA
E ::= CR
F ::= QB
}

---

8. Dada la gramática G:

G = ({e,f,g,z,a,b,d}, {Y, X, E, A, D, I, G}, A, P),

P = { 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 }

a) Transformar a FNC, explicando cada paso realizado.
b) Determinar si las palabras 'abz' y 'azdbb' pertenecen al lenguaje generado por G. En caso afirmativo, generar un árbol de derivación para dicha palabra. En caso negativo, justificar la no pertenencia.

Solución:

a) **Transformación a FNC, limpiándola previamente:**

Limpieza

- Reglas Innecesarias:
  E::=E
- Símbolos inaccesibles:
  - No aparecen en parte derecha: D, Y
    D::=eI, Y::=b
  - Tras varios ciclos (indirectos): G, I
    G::=g, I::=fG

Gramática limpia:

G' = ({a,b,d,z}, {X, E, A}, A, P'),

P' = { A::=a
A::=aX
A::=azb
E::=b
E::=d
X::=Xb
X::=XE
X::=z
}

Transformación a FNC (tabuladas en FNC)

- Tratamiento de A::=aX
  A::=BX
  B::=a

- Tratamiento de A::=azb
  A::=BC
  C::=zb
  C::=DF
  D::=z
  F::=b

- Tratamiento de X::=Xb
  X::=XF

Gramática en FNC:

G' '= ({a,b,d,z}, {A, B, C, D, E, F, X}, A, P''),

P'' = { A::=a
A::=BC
A::=BX
B::=a
C::=DF
D::=z
E::=b
E::=d
F::=b
X::=XE
X::=XF
X::=z
}

b) Palabras:

- abz: No pertenece, porque no existe ningún árbol de derivación que la genere.
- azdbb: Sí pertenece:

A → **BX** → a**X** → a**XF** → a**XE**FF → azEFF → azdFF → azdbF → azdbb

9. Obtener una gramática en FNC equivalente a la siguiente:

G = ({a, b, c},{S, A, B, C, D, E), S, P)

P = { S::=AaB | Cbb | B
A::=Aa | cD
B::=a | Ba | λ
C::=Sa | a | abB
D::=aaA
E::=aa }

Solución:

a) Gramática Bien Formada

Símbolos superfluos o no generativos = {A,D}

P = { S ::= Cbb | B
B ::= a | Ba | λ
C ::= Sa | a | abB
E ::= aa }

Símbolos inaccesibles = {E}

P = { S ::= Cbb | B
B ::= a | Ba | λ
C ::= Sa | a | abB}

Eliminación de reglas no generativas: B::= λ

P = { S ::= Cbb | B | λ
B ::= a | Ba
C ::= Sa | a | abB | ab}

Eliminación de reglas de redenominación: S::=B

P = { S ::= Cbb | a | Ba | λ
B ::= a | Ba
C ::= Sa | a | abB | ab}

La gramática bien formada equivalente es la siguiente (obsérvese que "c" no pertenece al alfabeto de símbolos terminales de la gramática, puesto que no hay ninguna regla que lo genere):

G = ({a, b},{S, B,C), S, P)
P = { S ::= Cbb | a | Ba | λ
B ::= a | Ba
C ::= Sa | a | abB | ab}

b) Transformación a FNC

La transformación de dicha gramática en una gramática en forma formal de Chomsky conlleva la substitución de las siguientes reglas:

S::=Cbb
S::=CD
D::=EE
E::=b
S::=Ba
S::=BF
F::=a
B::=Ba
B::=BF
C::=Sa
C::=SF
C::=abB
C::=FG
G::=EB
C::=ab
C::=FE

La gramática en forma normal de Chomsky resultante será, por tanto:

G = ({a, b},{S, B,C,D,E,F,G), S, P)
P = { S ::= CD | a | BF | λ
B ::= a | BF
C ::= SF | a | FG | FE
D ::=EE
E ::= b
F ::= a
G ::= EB}

12. Sea la gramática G, hallar su FNG equivalente.

G = ({1,2}, {A,B,C,D,E}, A, P),
donde P = {A ::= CD | EB | λ
B ::= BC | 1
C ::= 2
D ::= BC
E ::= 1}

Se convierte a Gramática limpia → Ya lo está.

Se elimina la recursividad a la izquierda:

B ::= BC → B ::= 1 | 1X
X ::= C | CX → Podemos eliminar la redenominación → X ::= 2

G' = ({0,1,2}, {A,B,C,D,E}, A, P),
donde P = {**A ::= CD | EB | λ**
B ::= 1 | 1X
X ::= 2 | CX
C ::= 2
D ::= BC
E ::= 1}

Ordenamos los no terminales:

{A, D, B, X, C, E} (Por ejemplo).

Clasificamos las reglas según este orden: *=Reglas que deben ser transformadas.

A ::= CD (G2) *
A ::= EB (G2) *
A ::= λ (G1)
B ::= 1 (G1)
B ::= 1X (G1)
X ::= 2 (G1)
X ::= CX (G2) *
C ::= 2 (G1)
D ::= BC (G2) *
E ::= 1 (G1)

Transformamos las de grupo 2 en grupo 1:

A ::= CD (G2)
A ::= 2D
A ::= EB (G2)
A ::= 1B
X ::= CX (G2)
X ::= 2X
D ::= BC (G2)
D ::= 1C | 1XC

Gramática FNG:

A ::= 2D
A ::= 1B
A ::= λ
B ::= 1
B ::= 1X
X ::= 2
X ::= 2X
C ::= 2
D ::= 1C | 1XC
E ::= 1

13. Sea la gramática G, hallar su FNG equivalente.

G = ({a,b,c,d,e,0,1}, {S,A,B}, S, P),

P = { S → AaB | AaC
A → Ab | Ac | b | c
B → BdC | 0
C → CeB | 1 }

Primero, eliminamos la recursividad por la izquierda:

G = ({a,b,c,d,e,0,1}, {S,A,B}, S, P)
P = { S → AaB | AaC
A → b | c | bD | cD
D → b | c | bD | cD
B → 0 | 0E
E → dCE | dC
C → 1F | 1
F → eBF | eB}

Pasamos a FNG:

Ordenamos los NT: {S, A, D, B, E, C, F}

S → AaB (g2) *
S → AaC (g2) *
A → b (g1)
A → c (g1)
A → bD (g1)
A → cD (g1)
D → b (g1)
D → c (g1)
D → bD (g1)
D → cD (g1)
B → 0 (g1)
B → 0E (g1)
E → dCE (g1)

B → dC (g1)
C → 1F (g1)
C →1 (g1)
F → eBF (g1)
F →eB (g3)
}

Transformamos de g2 a g1:

S → baB | caB | bDaB | cDaB
S → baC | caC | bDaC | cDaC

Crearemos los siguientes 2 NT:
G → aB
H → aC
[o sólo G → a]

Con lo cual, la G en FNG final será:

	Con sólo G→a
S → bG | cG | bDG | cDG	S → bGB | cGB | bDGB | cDGB
S → bH | cH | bDH | cDH	S → bGC | cGC | bDGC | cDGC
G → aB	G → a
H → aC	----
A → b | c| bD |cD	Resto igual…
D → b | c | bD | cD	
B → 0 | 0E	
E → dCE	
B → dC	
C → 1F | 1	
F → eBF | eB	