Formal Languages and Automata Theory
Exercises Languages and Formal Grammars
Unit 4 – Part 2

Authors:

Araceli Sanchis de Miguel
Agapito Ledezma Espino
Jose A. Iglesias Martínez
Beatriz García Jiménez
Juan Manuel Alonso Weber

* Several exercises are based on the ones proposed in the following books:

- Enrique Alfonseca Cubero, Manuel Alfonseca Cubero, Roberto Moriyón Salomón. *Teoría de autómatas y lenguajes formales.* McGraw-Hill (2007).
- Manuel Alfonseca, Justo Sancho, Miguel Martínez Orga. *Teoría de lenguajes, gramáticas y autómatas.* Publicaciones R.A.E.C. (1997).
- Pedro Isasi, Paloma Martínez y Daniel Borrajo. *Lenguajes, Gramáticas y Autómatas. Un enfoque práctico.* Addison-Wesley (1997).

Universidad Carlos III de Madrid
www.uc3m.es

1. Given the grammar G:

G = ({a,b,c,d,e,f,0,1,2,3,4}, {G,H,I,J}, G, P),
P = {abcG::=abcJ, Gdef::=Idef, J::=10H01, 2H4::=234, 2H4::=24, I::=1}

Determine its type in the Chomsky Hierarchy, and carry out the required modifications to obtain an equivalent grammar G' of the most restricted type in this hierarchy (i.e., from type-0 to type-1, from type-1 to type-2, or from type-2 to type-3).

2. Obtain an equivalent well-formed grammar for the following one:

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}

3. Given the following left-linear grammar G, obtain an equivalent right-linear G' grammar.

G=({0,1},{A,S},S,P)
P={ S ::= 1 | A1; A ::= S0}

4. Obtain an equivalent grammar in Chomsky Normal Form (CNF) equivalent to the following one:

G = ({a,b,c},{S, Q, R}, S, P)
P = { S ::= Rba | Q
Q ::= Qb | b
R ::= cRQb | cb
}

5. Given the grammar 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) Transform to CNF detailing the process followed.
   b) Determine whether the words 'abz' y 'azdbb' are included in the language generated by G. If this is the case, generate a parse tree for the included words. If not, justify the not inclusion in the language.

6. Obtain a grammar in CNF equivalent to the following grammar:

   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 }

7. [Exercise Exam September 1999] Given the grammar G, calculate an equivalent grammar in GNF.

   G = ({a,b}, {S}, S, P), where P = {S::=aSb | SS | λ}

8. Given the following grammar G, calculate an equivalent grammar in GNF.

   G = ({0,1,2}, {A,B,C,D,E}, A, P),
   where P = {A ::= CD | EB | λ
   B ::= BC | 1
   C ::= 2
   D ::= BC
   E ::= 1}

9. Given the following grammar G, calculate an equivalent grammar in GNF.

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 }

10. [Exercise Exam September 1998] Design a grammar in Greibach Normal Form to generate mathematical expressions having the form:

*number + (number / number) – number * (number + number)*

11. [Exercise Exam September 1997] Given an *if* sentence included in a high-level programming language with the following restrictions:

- There are only two instructions in each *if* sentence: the *if* part **only consists of one sentence**. The *then* part is the assignment of a number to a variable.
- Nested *if* sentences are allowed.
- The condition that is evaluated is a Boolean variable.
- Therefore, the set of terminal symbols set of the grammar must be:

Σ_T = {*if, then, else, ::=, var, num, cond*}

Transform the obtained grammar into GNF.