**COMP2270/6270 – Theory of Computation**
**Seventh week**

**School of Electrical Engineering & Computing**
**The University of Newcastle**

*Note: Some exercises belong to Chapters 11 and 12 of Ref [1]*

**Exercise 1)** Give an example of a language that is not regular but that it can be generated by a context-free grammar. Can you give another three examples? Justify your answers.

**Exercise 2)** Let *G* be the ambiguous expression grammar given below (Example 11.14 of Ref. [1]). Show at least three different parse trees that can be generated from *G* for the string `id+id*id*id`.

G = {{E, id, +, *, (, )}, { id, +, *, (, ) }, R, E }, where:

R = { E → E + E
E → E * E
E → ( E )
E → id
}

**Exercise 3)** Consider the expression grammar G′ given below (Example 11.19 of Ref. [1]).

G′ = {{E, T, F, id, +, *, (, )}, { id, +, *, (, ) }, R, E }, where:

R = { E → E + T
E → T
T → T * F
T → F
F → ( E )
F → id
}

a) Trace a derivation of the string `id+id*id*id` in *G′*.
b) Add exponentiation (**) and unary minus (-) to *G′*, assigning the highest precedence to unary minus, followed by exponentiation, multiplication, and addition, in that order.

**Exercise 4)** Let *G* be the grammar given below (Example 11.12 of Ref [1]). Show a third parse tree that *G* can produce for the string (())().

G = {{S, ), (}, { ), ( }, R, S }, where:

R = { S → ( S )
S → SS
S → ε
}

**Exercise 5)** Consider the following grammar *G*: $S \rightarrow 0S1 \mid SS \mid 10$
Show a parse tree produced by *G* for each of the following strings:

a) 010110
b) 00101101

**Exercise 6)** Convert each of the following grammars to Chomsky Normal Form:

a) $S \rightarrow ABC$
$A \rightarrow \mathtt{a}C \mid D$
$B \rightarrow \mathtt{b}B \mid \varepsilon \mid A$
$C \rightarrow A\mathtt{c} \mid \varepsilon \mid C\mathtt{c}$
$D \rightarrow \mathtt{aa}$

b) $S \rightarrow \mathtt{a}TV\mathtt{a}$
$T \rightarrow \mathtt{a}T\mathtt{a} \mid \mathtt{b}T\mathtt{b} \mid \varepsilon \mid V$
$V \rightarrow \mathtt{c}V\mathtt{c} \mid \varepsilon$

**Exercise 7)** Build a PDA to accept each of the following languages $L$:

a) BalDelim = {$w$ : where $w$ is a string of delimeters: (, ), [, ], {, }, that are properly balanced}.
b) $\{\mathtt{a}^i\mathtt{b}^j : 2i = 3j + 1\}$.
c) $\{w \in \{\mathtt{a}, \mathtt{b}\}^* : \#_{\mathtt{a}}(w) = 2 \cdot \#_{\mathtt{b}}(w)\}$.
d) $\{\mathtt{a}^n\mathtt{b}^m : m \leq n \leq 2m\}$.
e) $\{w \in \{\mathtt{a}, \mathtt{b}\}^* : w = w^R\}$.

REFERENCES

[1] Elaine Rich, Automata Computatibility and Complexity: Theory and Applications, Pearson, Prentice Hall, 2008.