COMP2270/6270 – Theory of Computation
Twelfth Week

School of Electrical Engineering & Computing
The University of Newcastle

Exercise 1) Show each of the following grammars are ambiguous:

a) ({S,A,B,T,a,c}, {a,c}, R, S)
R = {S → AB, S → BA, A → aA, A → ac, B → Tc, T → aT, T → a}

b) ({S,a,b}, {a,b}, R, S)
R = {S → ε, S → aSa, S → bSb, S → aSb, S → bSa, S → SS}

Exercise 2) What is meant by *inherent ambiguity* for context-free languages? Is it a concept that applies to grammars or languages?

Exercise 3) Convert the following CFG to Chomsky Normal Form:

$$S \to \mathrm{aSa}$$
$$S \to \mathrm{B}$$
$$B \to \mathrm{bbC}$$
$$B \to \mathrm{bb}$$
$$C \to \epsilon$$
$$C \to \mathrm{cC}$$

Exercise 4) Show the PDA that cfgtoPDAtopdown will produce on G.

$$G = (\{S, A, B, a, b, c\}, \{a, b, c\}, R, S)$$

where:

$$R = \{S \to \mathrm{AB}, \mathrm{S} \to \mathrm{B}, \mathrm{A} \to \mathrm{aaA}, \mathrm{A} \to \mathrm{aa}, \mathrm{B} \to \mathrm{bbB}, \mathrm{B} \to \mathrm{b}\}$$

Exercise 5) Use closure properties to prove the following true or false (you can assume membership of other well-known languages (i.e. $a^n b^n \in CFL$).

a. $\{w \in \{a,b\}^* : \#_a(w) = \#_b(w)\}$ is regular.
b. $\{w \in a^i b^k : i \neq k\}$ is regular.
c. $\{w \in a^n b^n : n \neq 10000\}$ is context-free.
d. $\{w \in \{a,b,c\}^* : \#_a(w) = \#_b(w) = \#_c(w)\}$ is context-free.

Exercise 6) What is the difference between a deterministic and a non-deterministic turing machine? Is one more powerful than the other (in terms of the languages they can accept or functions they can compute)? What about single tape and multiple tape TMs?

Exercise 7) What is the difference between a deciding and a semi-deciding turing machine?

Exercise 8) Encode the following TMs as strings as their string representations <M>.

a. $M = (\{s, q, h\}, \{a, b, c\}, \{\square, a, b, c\}, \delta, s, \{h\})$
Where δ is given by:

state	**Symbol on tape**	$\boldsymbol{\delta}$
s	$\square$	$(q, \square, \rightarrow)$
s	a	$(s, b, \rightarrow)$
s	b	$(q, a, \leftarrow)$
s	c	$(q, b, \leftarrow)$
q	$\square$	$(s, a, \rightarrow)$
q	a	$(q, b, \rightarrow)$
q	b	$(q, b, \leftarrow)$
q	c	$(h, a, \leftarrow)$

a. $M = (\{p, q, y, n\}, \{v, x\}, \{\square, v, x, z\}, \delta, p, \{y, n\})$
Where δ is given by:

State	**Symbol on tape**	$\boldsymbol{\delta}$
p	$\square$	$(q, \square, \rightarrow)$
p	v	$(y, x, \rightarrow)$
p	x	$(y, v, \leftarrow)$
p	Y	$(y, z, \leftarrow)$
q	$\square$	$(n, z, \rightarrow)$
q	v	$(n, z, \rightarrow)$
q	x	$(p, z, \leftarrow)$
q	y	$(q, v, \leftarrow)$

Exercise 9) Describe in clear English a Turing Machine M to compute the following functions.

a. A single tape TM to compute addition of two unary numbers. Given the string $\langle x \rangle \# \langle y \rangle$ where $\langle x \rangle$ and $\langle y \rangle$ are the unary encoding of the values x and y the M should output halt with $\underline{\square}\langle z \rangle$ on the tape, where $\langle z \rangle$ is the unary encoding of $z = x + y$.
b. A multiple tape TM to compute **binary** addition in the same specification as above.

REFERENCES

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