Supporting material for weeks 14 and 15

Overall aim of these workbooks and quizzes (all non-assessed)
The aim of these supporting materials is to provide, and work through, multiple examples of the concepts taught in the associated (more theoretical) lectures. Note that some examples may be left as supplementary exercises for you to complete in your own time. The materials will allow me to assess progress and problems overall with understanding of the key concepts. They also allow you to check your own progress against the learning objectives, receive feedback and feed forward to help you with future understanding of the module content.

Learning Objectives from Week 14

1. recognise context sensitive and unrestricted phrase structure grammars
2. design and understand the behaviour of Turing machines (TM)
3. understand the connection between Turing machines and unrestricted and context sensitive grammars
4. understand the order and relations of phrase structure grammars in the Chomsky Hierarchy

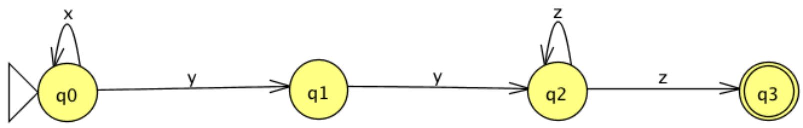
Learning Objectives from Week 15

5. appreciate the connection between formal grammars their computational implications
6. appreciate Turing's Thesis
7. understand the concept of a Universal Turing Machine
8. understand that there are some (well-specified) problems that cannot be solved (for example the halting problem)

You should use JFLAP in this week's workbook

JFLAP is downloadable *free* from the web, see https://www.jflap.org/ and is already installed in the SAT Building labs and on https://mylab.lancaster.ac.uk/ in the "SCC Lab" Virtual Machine.

1. a) Write down the regular grammar production rules associated with this finite state recogniser (FSR) shown in JFlap style. In your answer, you should also state which is the start symbol, and then separately list the non-terminals and terminals you are using in this phrase structure grammar.



b) Give two examples of valid sentences for this grammar and two examples of sentences that are not valid.

c) Using set definition notation, describe the set of valid sentences accepted by this FSR.

d) Using regular expression notation, describe the language recognised by the FSR.

2. In the following languages, lower case letters ‘a’, ‘b’, ‘c’ and ‘d’ and left ‘(’ and right round brackets ‘)’ are terminal symbols. Upper case letters ‘S’, ‘A’ and ‘B’ are non-terminal symbol symbols. ‘S’ is the start symbol. For each of the two languages below, the production rules are shown in context-free grammar format. However, it may be the case that these languages can also be expressed as regular grammars or alternatively they cannot be converted.

For each language below, describe the format of valid sentences using a set definition and write down whether it is only context-free, or whether it can be converted to a regular grammar. If it can be converted to regular format, then draw a *deterministic* FSR to recognise sentences from the language. If the grammar cannot be converted then state why not, and say how you could place extra restrictions on your set definition in order to make it possible to do so.

a) $S \rightarrow aAa$
$A \rightarrow bB \mid cB$
$B \rightarrow dB \mid d$

b) $S \rightarrow (S)$
$S \rightarrow (\,)$

3. Design a new Turing Machine (TM) to recognise palindromic sentences of *even* length in a language that contains only three symbols “a”, “b” and “c”. Your machine should reject *odd* length palindromes. For example, “abba” and “cbbc” will be accepted, but “aca” and “c” will be rejected. You can assume that your TM will start with its read/write head directly over the left-most non-blank symbol. You need not consider transitions to deal with invalid sentences and you can assume that the input tape will only consist of the three symbols in the language. For this question, you must use JFlap version 7. Handmade or drawn machines should not be submitted.

a) Describe in words the step-by-step algorithm for your machine. It must be clear how this corresponds to the states that you will create and show in part (b).

b) Create the machine in JFlap and paste a screenshot of it into your answer.

c) Third, use the “Multiple Run (Transducer)” item in the JFlap “Input” menu, and click “Load inputs” at the bottom to load in the “week_14_15_tm_input.txt” file that is provided on Moodle. Once this is loaded, click “Run inputs” to show which of the potential inputs are accepted, and to see the outputs. Paste in a screenshot of the whole multiple run list (showing inputs, outputs and results) in JFlap into your answer.

4. Consider the following complex-looking grammar where upper case letters are non-terminals and lower case letters are terminals, and "S" is the start symbol:

```
S → AS | AB
B → BB | C
AB → HXNB
NB → BN
BM → MB
NC → Mc
Nc → Mcc
XMBB → BXNB
XBMc → Bc
AH → HA
H → a
B → b
```

a) Categorise this grammar according to the Chomsky Hierarchy and explain why it is in this category.

b) This grammar generates sentences of the form specified in the set definition:
$\{a^i b^j c^{i*j} : i, j \geq 1\}$
The shortest possible valid sentence is "abc". Write down the step by step derivation of "abc" starting from the start symbol "S".

c) What property of this derivation indicates that a Linear Bounded Turing Machine is not sufficient to recognise strings of the language generated by this grammar?

5. Consider a Turing Machine which carries out an unspecified operation on its input tape. The Turing Machine has been coded as a set of quintuples shown in table form:

Current State	Read Symbol	Write Symbol	Move	New State
q0	1	1	R	q0
q0	+	1	R	q1
q1	1	1	R	q1
q1	B	B	L	q2
q2	1	B	L	q3
q3	1	1	L	q3
q3	B	B	R	q4

Draw the Turing Machine which corresponds to these quintuples. The start state is q0 and the halt state is q4.

6. Consider this context-free grammar for recognising sentences of Boolean logic, where symbols “S”, “A” and “T” are non-terminals and symbols “or”, “and”, “not”, “a”, “b” and “c” are terminals:

```
S → T | A | T or S | A or S
A → T and T | T and A | not A
T → a | b | c | not T
```

a) Write down two sentences that would be accepted by this grammar, and two sentences that would not be accepted.

b) This grammar is *ambiguous*. Discuss possible problems that the ambiguity might cause when interpreting the sentence “a or not b and c” (assuming the usual Boolean login interpretation of the symbols “and”, “or” and “not”). You should draw parse trees to illustrate your answer.

7. Symbols “S” and “A” are non-terminals, and “g”, “l”, “e” and “o” are terminals in a language where “S” is the start symbol. Given the following grammar describing the language expressed with these context free production rules:

$$S \rightarrow gAgle$$

$$A \rightarrow oAo \mid oo$$

a) draw a *non-deterministic* PDR with the usual three states 1, 2 and H

b) now generate an equivalent *deterministic* PDR:

c) Some context free grammars can be rewritten as regular grammars and some are genuinely context-free. In this case, it is possible to convert this into a regular grammar. Draw the finite state recogniser corresponding to this language. Again, it must be deterministic.