The Austalian National University
Research School of Computer Science
Dirk Pattinson
Jinbo Huang

Semester 1, 2013
Tutorial 2

Theory of Computation

Questions marked (S) are self-test questions with solutions provided at

`http://infolab.stanford.edu/~ullman/ialcsols/sol2.html`

Questions marked (A) are assignment questions.

Exercise 1 **Construction of Regular Expressions** (S)

Write regular expressions for the followign languages:

1. The set of strings over the alphabet $\{a, b, c\}$ that contain at least one a and at least one b.
2. The set of strings of 0's and 1's such that ever pair of adjacent 0's appears before any pair of adjacent 1's.

(These are exercises 3.1.1 (a) and 3.1.2 (a) in the textbook.)

Exercise 2 **Decoding Regular Expressions** (S)

Give an english language description of the language of the regular expression $(1 + \epsilon)(00^*1) * 0^*$. (This is exercise 3.1.4. (a) in the textbook.)

Exercise 3 **From Automata to Regular Expressions** (S)

Consider the automaton with the following transition table:

	0	1
$\to q_0$	q_2	q_1
q_1	q_3	q_1
$*q_2$	q_3	q_2

You should think of state q_i as having number i, and recall the definition of the regular expression $R_{ij}^{(k)}$ that was used in the construction of a regular expression from a DFA.

1. Give all regular expressions $R_{ij}^{(0)}$.
2. Give all regular expressions $R_{ij}^{(1)}$ and simplify them as much as possible.
3. Construct the transition diagram for the DFA and give a regular expression for tis language by eliminating state q_2 (and labelling remaining arcs with regular expressions).

(This is exercise 3.2.4 in the textbook.)

Exercise 4 **From Regular Expressions to Automata** (S)

Convert the regular expression 01^* to an NFA with ϵ-transitions. (This is exercise 3.2.4 in the textbook.)

Exercise 5 **Identities between Regular Expressions** (S)

Prove or disprove the following identities about regular expressions:

1. $(R + S)^* = R^* + S^*$
2. $(RS + R)^*RS = (RR^*S)^*$

(These are exercises 3.4.2. (a) and (c) in the textbook.)

Exercise 6 **Higman's Lemma** (A)

Suppose that Σ is a finite alphabet and $L \subseteq \Sigma^*$ is a (not necessarily regular) language over Σ.

The language $S(L)$, called the *subsequence language* of L, contains all strings over Σ that can be constructed from strings in L by deleting characters from Σ.

Formally:

$$S(L) = \{a_1 \dots a_n \in \Sigma^* \mid w_0 a_1 w_1 a_2 w_2 \dots a_n w_n \in L \text{ for some } w_0, \dots, w_n \in \Sigma^*\}$$

Higman's Lemma states that $S(L)$ is regular as long as Σ is finite (in particular without requiring that L is regular!). You are asked to prove the special case of Higman's lemma for the case of alphabets consisting of just one letter. That is:

Suppose that Σ contains a single letter and let $L \subseteq \Sigma^*$ be a not necessarily regular language. Give a proof of the fact that $S(L)$ *is* a regular language.

Exercise 7 **Relationships between Regular Languages** (A)

Consider the regular languages:

$$L_1 = L((ab + ac^*)^*) \qquad L_2 = L(a(b + c^*)^*)$$

Decide whether or not each of the following statements is true.

1. $L_1 = L_2$
2. $L_1 \cap L_2 = \emptyset$
3. $L_1 \subseteq L_2$
4. $L_2 \subseteq L_1$.

In each case, provide a formal proof that justifies your answer.