# CSI3104 Midterm Solution and Marking Scheme

## Summer 2011

---

Instructions for Markers:

- This midterm exam has 5 questions, each of which is worth 6 marks, resulting in 30 marks in total.
- Detailed marking scheme for each question is shown below the solution provided for that question.
- Follow the marking scheme and use your own judgement on the merit of the students' answers, but be consistent in your marking of ALL the submitted papers.

---

**1.** (a) Let $S$ be a set of words. Define $S^*$.

(b) Consider the language $S^*$, where $S = \{a, b\}$. How many words does this language have of length 2? of length 3? of length $n$ with $n$ being any non-negative integer?

Solution:

(a) If $S$ be a set of words, then $S^*$ is defined as the set of all **finite strings** formed by **concatenating words from** $S$, where any word may be used **as often as we like**, and where the null string $\mathbf{\Lambda}$ **is also included.**.

(b) $S^*$ has **four** words of length 2, **eight** words of length 3, and $\mathbf{2^n}$ words of length $n$ in general.

Marking Scheme:

- 2 marks for part (a).
- 4 marks for part (b): 1 mark for "four words of length 2", 1 mark for "eight words of length 3", and 2 marks for "$2^n$ words of length $n$".

**2.** Show whether or not the following two regular expressions define the same language:

$$(\mathbf{a}+\mathbf{b})^*\mathbf{ba}(\mathbf{a}+\mathbf{b})^* + \mathbf{ab}^* \quad \text{and} \quad (\mathbf{a}+\mathbf{b})(\mathbf{a}+\mathbf{b})^*$$

Solution:

First, note that $(\mathbf{a}+\mathbf{b})(\mathbf{a}+\mathbf{b})^*$ represents all possible non-empty words of $a$'s and $b$'s.

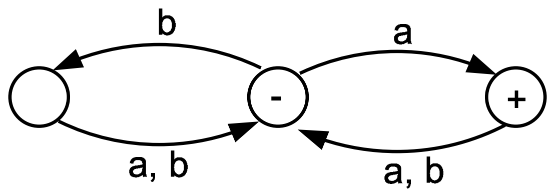
The regular expression $(\mathbf{a}+\mathbf{b})^*\mathbf{ba}(\mathbf{a}+\mathbf{b})^*$ represents all the words containing the substring $ba$. Compared with all possible non-empty words of $a$'s and $b$'s, there are three types of words missing: (i) words of all $a$'s, (ii) words of all $b$'s, and (iii) words in which all $a$'s precedes all $b$'s. However, not all of these can be generated by $\mathbf{ab}^*$. For example, the word $aabb$ **is not in** the language defined by $(\mathbf{a}+\mathbf{b})^*\mathbf{ba}(\mathbf{a}+\mathbf{b})^* + \mathbf{ab}^*$, but it **is in** the language defined by $(\mathbf{a}+\mathbf{b})(\mathbf{a}+\mathbf{b})^*$.

Hence, these two regular expressions do NOT defined the same language.

Marking Scheme:

- 6 marks for a clear, convincing and correct proof.

**3.** (a) Describe in English the language $L$ accepted by the following FA:




(b) Use your description in part (a), write a regular expression for the language $L$. Justify the correctness of your regular expression.

Solution:

(a) The above FA accepts the language $L$ of all **odd length words** over $\{a, b\}$ **that end with an** $a$.

(b) A regular expression for the language $L$ is

$$(\mathbf{aa} + \mathbf{ab} + \mathbf{ba} + \mathbf{bb})^*\mathbf{a}$$

Justification:

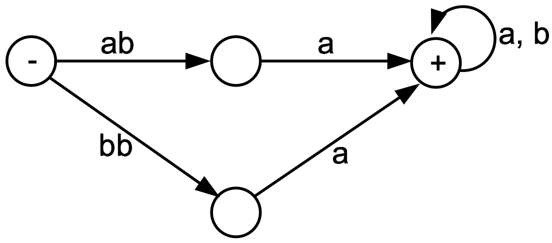
- The last part of the above regular expression, namely **a**, clearly indicates that any word of the language defined by this regular expression must end with an $a$.
- The first part of this regular expression, namely $(\mathbf{aa} + \mathbf{ab} + \mathbf{ba} + \mathbf{bb})^*$, represents all possible **even** clumps of $a$ and $b$ (including $\boldsymbol{\Lambda}$), which, when followed by an $a$, will form the language of all **odd length words that end with an** $a$.

Marking Scheme:

- 3 marks for part (a).
- 3 marks for part (b): 1.5 marks for correct regular expression, 1.5 marks for justification.

**4.** (a) Give the definition of a **transition graph (TG)**.

(b) Use the **bypass algorithm** (presented in the proof of Kleene's theorem, Part 2) to convert the following TG into regular expression:

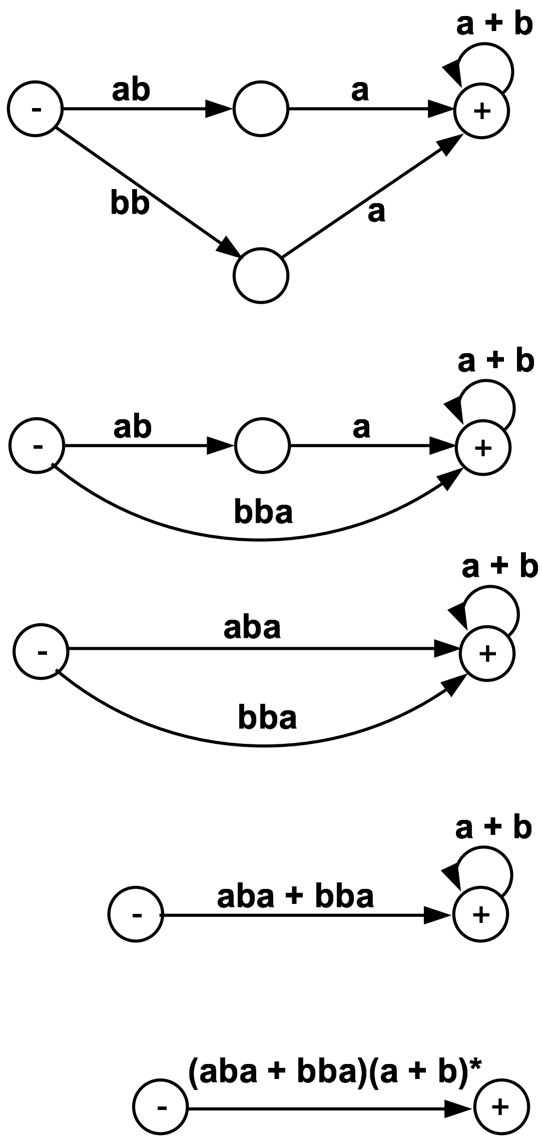



Solution:

(a) A **transition graph**, abbreviated **TG**, is a collection of three things:

- A **finite set** of states, **at least** one of which is designated as the start state (−), and some (maybe none) of which are designated as final states (+).
- An **alphabet** Σ of possible input letters from which input strings are formed.
- A **finite set** of transitions (edge labels) that show how to go from some states to some others, based on reading specified **substrings** of input letters (possibly even the **null string Λ**).

(b) Using the bypass algorithm, we convert the TG into regular expression as follows:



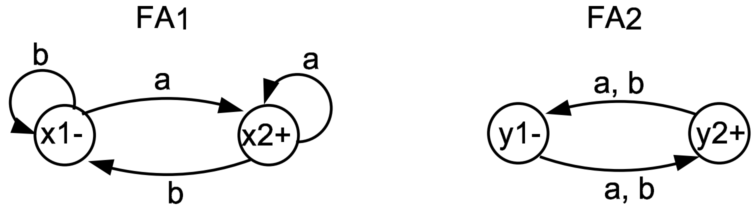

Hence, the regular expression is

$$(\mathbf{aba} + \mathbf{bba})(\mathbf{a} + \mathbf{b})^* = (\mathbf{a} + \mathbf{b})\mathbf{ba}(\mathbf{a} + \mathbf{b})^*$$

<u>Marking Scheme:</u>

- 2 marks for part (a).
- 4 marks for part (b).

**5.** Let $FA_1$ be the machine below that accepts all words that end in $a$. Let $FA_2$ be the machine below that accepts all words with an odd number of letters (i.e., odd length).

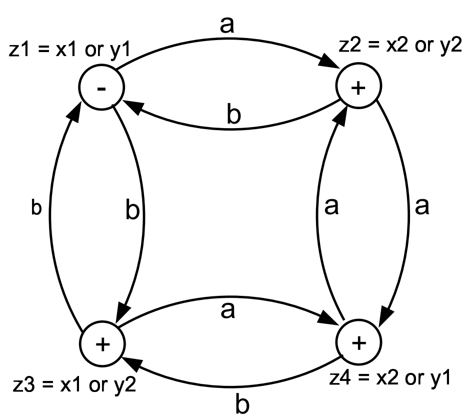



Use the algorithm presented in the proof of Kleene's theorem, Part 3, Rule 2, to construct an $FA$ for the union language $FA_1 + FA_2$ (i.e., the language of all words that either have an odd number of letters or end in $a$). Clearly show the transitions leading to the construction of the final diagram.

Solution:

- Let $z_1(-) = (x_1 \text{ or } y_1)$.
  In $z_1$, read an $a$, go to $(x_2(+) \text{ or } y_2(+)) = z_2(+)$.
  In $z_1$, read a $b$, go to $(x_1 \text{ or } y_2(+)) = z_3(+)$.
- In $z_2$, read an $a$, go to $(x_2(+) \text{ or } y_1) = z_4(+)$.
  In $z_2$, read a $b$, go to $(x_1 \text{ or } y_1) = z_1$.
- In $z_3$, read an $a$, go to $(x_2(+) \text{ or } y_1) = z_4(+)$.
  In $z_3$, read a $b$, go to $(x_1 \text{ or } y_1) = z_1$.
- In $z_4$, read an $a$, go to $(x_2(+) \text{ or } y_2(+)) = z_2(+)$
  In $z_4$, read a $b$, go to $(x_1 \text{ or } y_2(+)) = z_3(+)$.

Thus, the required $FA$ looks like this:



Marking Scheme:

- 6 marks for constructing the correct $FA$ that accepts the union language (using the required algorithm):
  - 3 marks for clearly showing the correct transitions leading to the construction of the final diagram
  - 3 marks for the correct final diagram