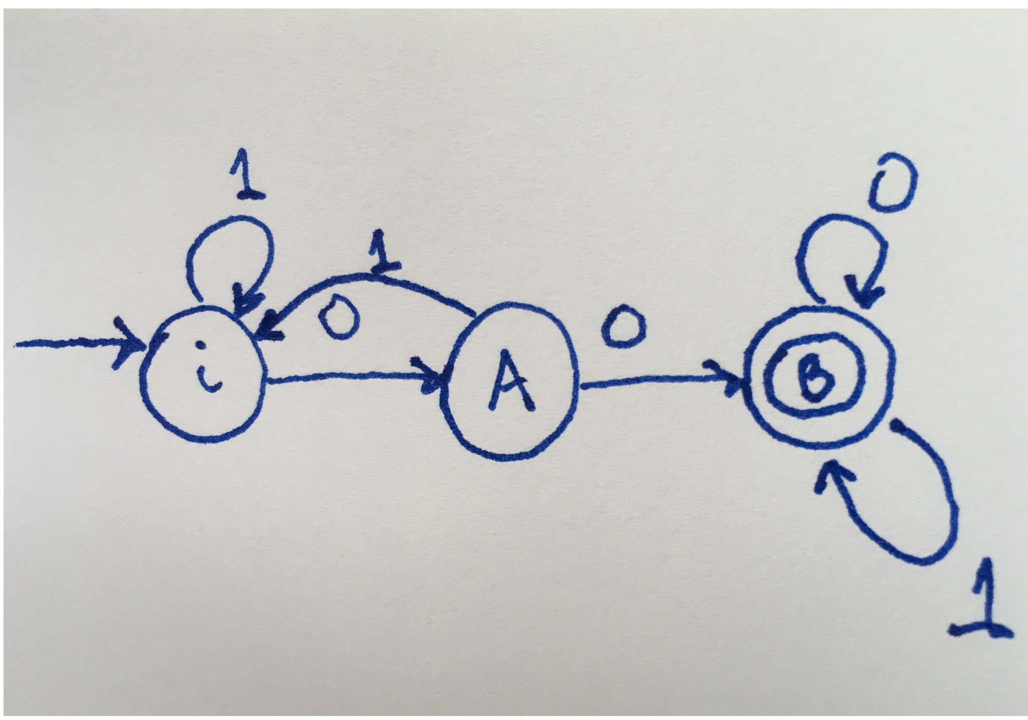
MAU22C00: TUTORIAL 10 SOLUTIONS
FORMAL LANGUAGES AND GRAMMARS

1) Let L be the language over the alphabet $A = \{0, 1\}$ consisting of all words where the string 00 occurs as a substring.

(a) Devise a regular grammar in normal form that generates the language L. Be sure to specify the start symbol, the non-terminals, and all the production rules.

(b) Write down a regular expression that gives the language L and justify your answer.

Solution: (a) We shall use the algorithm discussed in lecture in order to generate the regular grammar in normal form corresponding to the finite state acceptor constructed in last week's tutorial. Here is the finite state acceptor again:



The finite state acceptor has three states $\{i, A, B\}$, where i is the initial state. Correspondingly, we use three non-terminals in our regular grammar: the start symbol $\langle S\rangle$ corresponding to the initial state i, $\langle A\rangle$

corresponding to state A, and $\langle B\rangle$ corresponding to state B. We first write the production rules corresponding to the transitions out of the initial state i :

(1) $\langle S\rangle \to 1\langle S\rangle$.
(2) $\langle S\rangle \to 0\langle A\rangle$.

Next, we write the production rules corresponding to the transitions out of state A :

(3) $\langle A\rangle \to 1\langle S\rangle$.
(4) $\langle A\rangle \to 0\langle B\rangle$.

Finally, we write the production rules corresponding to the transitions out of state B :

(5) $\langle B\rangle \to 1\langle B\rangle$.
(6) $\langle B\rangle \to 0\langle B\rangle$.

Rules (1)-(6) are of type (i). For each accepting state, we will write down a rule of type (iii). Since there is only one accepting state, B, we have only one such rule:

(7) $\langle B\rangle \to \epsilon$.

(b) Recall from last week's tutorial that

$$L = \{w \in A^* \mid w = u \circ 00 \circ v \quad u, v \in A^*\}.$$

Therefore, $L = A^* \circ 00 \circ A^*$, and we have obtained the regular expression giving us the language L. Compare this solution to last week's tutorial when we proved this language was regular by applying the definition of a regular language.

2) Let M be the language

$$\{0101,\ 001001,\ 00010001,\ 0000100001,\ \ldots\}$$

whose words consist of some positive number n of occurrences of the digit 0, followed by the digit 1, followed by n further occurrences of the digit 0, and followed by the digit 1. (In particular, the number of occurrences of 0 preceding the first 1 is equal to the number of occurrences of 0 preceding the second 1.)

(a) Use the Pumping Lemma to show this language is not regular.
(b) Write down the production rules of a context-free grammar that generates exactly M. Justify your answer.

Solution: (a) If M is regular, then it has a pumping length p. Consider $w = 0^p10^p1 \in M$ and the decomposition $w = xuy$ with $|u| \geq 1$ and $|xu| \leq p$. Since $|xu| \leq p$, u can only consist of zeroes. Let $u = 0^{n_1}$, for some $n_1 \geq 1$. Clearly, $xu^2y \notin M$ as $xu^2y = 0^{p+n_1}10^p1$, so the length of the first sequence of zeroes is greater than that of the second sequence of zeroes violating the pattern of the language.

(b) Consider the following production rules:

(1) $\langle S\rangle \to 0\,\langle A\rangle\,01$,
(2) $\langle A\rangle \to 0\,\langle A\rangle\,0$,
(3) $\langle A\rangle \to 1$.

We can show by induction that a string w generated by these production rules is of one of the following forms:

- $w = \langle S\rangle$,
- $w = 0^n\,\langle A\rangle\,0^n 1$,
- $w = 0^n 1 0^n 1$.

Here $n \geq 1$. These rules will then generate exactly M. Note how these rules differ from the production rules of a regular grammar as non-terminals occur on both sides of the non-terminal in the first two production rules.