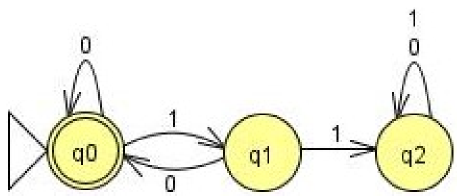
Theory of Computation

CS-312 **Fall 2023**
Exam 1 **Solution Set**

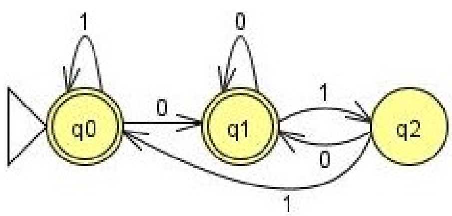
---

1. *10 points each. Give* deterministic *finite automata which accept the following languages:*

   (a) *Strings in which every* 1 *is immediately followed by a* 0*.*



   (b) *Strings which do not end in* 01*.*



2. *10 points each. Give regular expressions for the following languages:*

   (a) *Strings with an odd number of* 1*s.*

$$0^*10^*(10^*10^*)^*$$

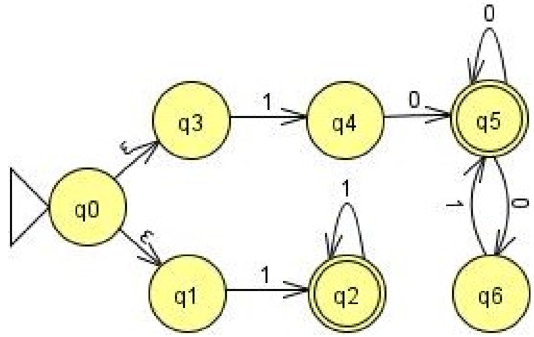
   (b) *Strings with exactly one occurrence of* 00 *as a substring. (Note that* 000 *has two occurrences of* 00 *as a substring and should thus be excluded from the language.)*

$$(1+01)^*00(1+10)^*$$

3. *10 points. Convert the following regular expression to a* nondeterministic finite automaton*:*

$$11^* + 10(0+01)^*$$

Answers will vary. Here is one solution:



4. *10 points. The POSIX standard for regular expressions includes many extensions to regular expressions. One of these is the* Kleene plus operator*.*

   Let r be a regular expression. The expression r^+ matches any concatenation of one or more strings *which match r. Note that this is different from the classical Kleene star operator r^*, which matches any concatenation of* zero or more strings *which match r.*

   Show that the Kleene plus operator does not change the class of languages accepted by regular expressions. That is: show how to transform any regular expression containing a Kleene plus operator into an equivalent regular expression without a Kleene plus operator.

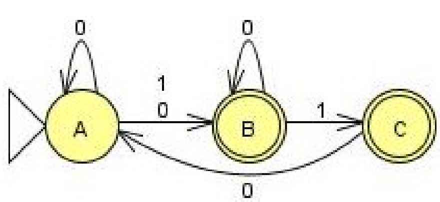
   The expression r^+ is equivalent to the regular expression rr^*.

5. *10 points. Let L_1 and L_2 be regular languages. Describe an algorithm to decide if $L_1 \cap L_2 \neq \emptyset$. That is, is there any string w such that $w \in L_1$ and $w \in L_2$?*

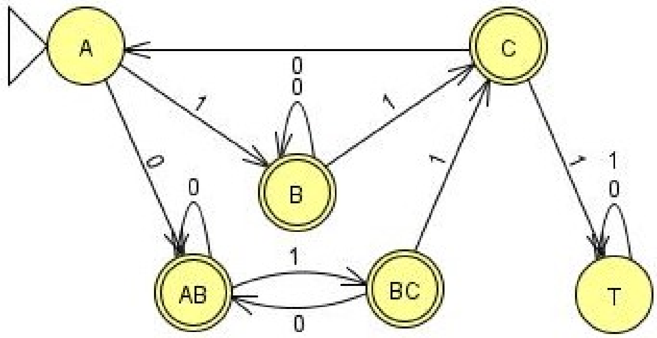
   Since L_1 and L_2 are regular languages, their intersection $L_1 \cap L_2$ is also a regular language. As a regular language, there is a DFA M that accepts $L_1 \cap L_2$.

   Construct M (using techniques discussed in class). Perform a graph search (BFS or DFS) on M, starting at the start state of M. If a final state is reachable from the start state, then there is a string that will drive M from its start state to a final state, which will then be accepted, and thus $L_1 \cap L_2 \neq \emptyset$. Conversely, if no final state is reachable from the start state, then there is no way for M to accept any strings, and $L_1 \cap L_2 = \emptyset$.

6. *15 points. Convert the* nondeterministic *finite automaton shown below to a* deterministic *finite automaton, using the "powerset" construction presented in class (and the textbook).*



   The answer is shown below.



7. *15 points. Let L be the set of binary strings with even length whose middle symbols are* 00. *Prove that L is not regular.*

   Let M be a DFA with k states that accepts L. By the Pumping Lemma, for any $z \in L$ with $|z| \geq k$, there are strings u, v, w such that:

   - $z = uvw$
   - $|uv| \leq k$
   - $|v| > 0$ ($i.e.$ $v \neq \epsilon$)
   - $uv^i w \in L$ for any $i \geq 0$

   Choose $z = 1^k 001^k$. Since $|z| = 2k + 2 > k$, the Pumping Lemma applies to z. That is, $1^k 001^k = z = uvw$, where the conditions in the Pumping Lemma apply.

   Since $|uv| \leq k$ and $uvw = 1^k 001^k$, we must have $v = 1^\ell$ for some $\ell > 0$, with v being a substring of the first block of 1s.

   The Pumping Lemma states that $uv^2 w = uvvw \in L$. Observe that $uvvw = 1^{k+\ell} 001^k$. Since $k + \ell > k$, and there are only two 0s in the string, the 0s are not the middle symbols of the string, so this string is not in L.

   Thus, $uvvw$ is both in L and not in L, our desired contradiction.