1 Finite automata

Here is a picture of a (deterministic) finite automaton:

![Finite Automaton Diagram]

Figure 1: A simple finite automaton

The two circles represent the states of the machine. Here, the states are labeled $q_0$ and $q_1$. You input to the machine a string of $a$’s and $b$’s, e.g. $aba$. The machine then begins a “computation”, starting in the state which has the unlabeled arrow pointing to it, the initial state, then transitions from state to state by following the arrows that correspond to the letters in the string. So, in our example, upon input $aba$, here is the “computation”:

- The machine starts in state $q_0$.
- It follows the $a$ arrow from state $q_0$ to state $q_1$.
- It then follows the $a$ arrow from state $q_1$ to state $q_0$.
- It then follows the $b$ arrow from state $q_0$ to state $q_1$.
- It finally follows the $a$ arrow from state $q_0$ to state $q_1$.

Once the computation has concluded, the machine accepts if the machine ended up in an accept state (that is, a state with a double circle) and rejects otherwise.

**Problem 1**: Does the machine in Figure 1 accept upon input string $aba$?

---

*This work was based off a Math Circle written by Dillon Zhi at UCLA several years ago.*
Problem 2: Consider the following machine:

![Diagram of a machine](image)

Figure 2: A slightly more complicated machine

(a) What sequence of states does the machine go through if you input 11001?
(b) Does the machine accept 11001?
(c) If you input the empty string $\varepsilon$, does the machine accept?

The name finite automaton refers to the fact that there are only finitely many states and the machine is self-operating. Each finite automaton has an alphabet, which is the set of letters it takes. In Figure 1, the alphabet was $\{a, b\}$ while the machine from Figure 2 had alphabet $\{0, 1\}$. Each state of a finite automaton has exactly one arrow leading out of it for each letter of its alphabet. (That is what makes these automata deterministic; by relaxing this condition, one arrives at nondeterministic automata.)
Problem 3: For the alphabet \{a, b\}, which of the machines below are *not* deterministic finite automata?

(a)

(b)

(c)
Problem 4: Suppose you are designing a vending machine which accepts only Quarters and Dollar Bills. The machine should only dispense a snack if at least $1.75 has been inserted during a transaction, otherwise the machine rejects the transaction and does not dispense. Your job is to design a finite automaton to control this machine.

(a) Think about what alphabet this automaton should work with. How can you represent a transaction as a string?

(b) Draw a finite automaton which can determine when the vending machine should dispense a snack.
Problem 5: Draw a finite automata that accepts only those strings over the alphabet \( \{\oplus, \otimes\} \) for which the number of \( \oplus \)'s minus the number of \( \otimes \)'s in the string is a multiple of 5.

2 Regular languages

Given a finite alphabet \( A \), we let \( A^* \) denote the set of strings from \( A \). A language over \( A \) is a subset of \( A^* \), that is, a collection of strings from the alphabet \( A \). The collection strings accepted by a finite automaton is called the language recognized by the automaton and a language is called regular if it is the language recognized by some finite automaton.

Problem 6: Give a concrete description of the language recognized by the finite automaton from Figure 1.
**Problem 7:** Draw a finite automaton with 6 states recognizing the language over the alphabet \{a, b\} that contains \textit{abbaa} as a substring at least once.

**Problem 8:** Draw a finite automaton recognizing the language of strings over the alphabet \{0, 1\} that end with either 00 or 11.
The next problems establish some closure properties of regular languages.

**Problem 9:** Suppose that $L$ is a regular language. Show that the complement of $L$ (consisting of those strings that do *not* belong to $L$) is also regular. (Hint: This is actually incredibly easy!)

**Problem 10:** Suppose that $L_1$ and $L_2$ are regular languages. Prove that the union $L_1 \cup L_2$ (consisting of all strings that belong to either $L_1$ or $L_2$, perhaps both) and the concatenation $L_1 L_2$ (consisting of all strings formed by placing a string from $L_1$ next to a string from $L_2$) are also regular languages.
3 The pumping lemma

How do you prove that a language is not regular? The main technique is the **Pumping Lemma**, which states that if $L$ is a regular language, then there is some integer $n$ such that, for all strings $w$ that belong to the language whose length is at least $n$, there are substrings $x$, $y$, and $z$ of $w$ such that:

- $w = xyz$
- the length of $xy$ is no more than $n$
- the length of $y$ is at least 1, and
- for all $k \geq 0$, $xy^kz$ also belongs to $L$.

Here, $y^k$ is $y$ stringed together $k$ times. So for strings in regular languages, as long as a word is long enough, some “interior” portion of the word can be “pumped up” as many times as you wish and the resulting word still belongs to the language.

**Problem 11:** Use the pumping lemma to prove that the following languages are *not* regular:

(a) The language $L$ over the alphabet $\{a, b\}$ consisting of all strings $a^kb^k$ for $k \geq 0$.

(b) The language $L$ over the alphabet $\{a, b\}$ consisting of all those strings that have an equal number of $a$’s and $b$’s.

(c) The language $L$ over alphabet $\{a, b\}$ consisting of all *palindromes*, that is, strings which are the same when reversed.
**Problem 12:** Prove the pumping lemma. Here are some steps to help out. First, suppose that $L$ is recognized by a machine with $n$ states. Suppose $w$ is a word in $L$ of length bigger than $n$.

(a) Conclude that the machine must repeat a state $q$ when processing the first $n$ symbols from $w$.

(b) Let $x$ denote the substring of $w$ processed before state $q$ was reached and let $y$ denote the substring of $w$ processed in between the first and second occurrences of $q$. Finally, let $z$ be the “rest” of $w$. (Draw a picture!) Show that these $x$, $y$, and $z$ are as desired.