18.404/6.5400 Recitation 1

Recitations supply opportunities to review difficult concepts and think through new examples of ideas from lecture. You're welcome to come to any or none of the six sections: it's your choice. Our sections all differ a bit because we spend time on what the students who show up ask about. These notes attempt to encapsulate all the sections in terms of important material.

1 Basic Vocabulary

Some terms that sometimes get muddled but should be distinct and used precisely:

- Symbol: a basic, atomic building block.

  What can be a *symbol* in this class? Anything, really; we'll encounter familiar symbols like $0, 1, a, b, c$, but there will also be more exotic ones like $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $\langle START \rangle$, etc.

- String: an ordered sequence of symbols.

- Language: a set of strings.

2 Finite Automata (FA)

A *finite automaton* (plural automata) is our first (very simple) of computation. Finite automata can be seen as detecting strings which answer a yes/no question.

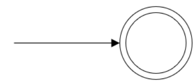
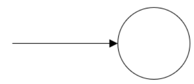
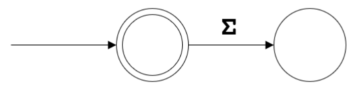
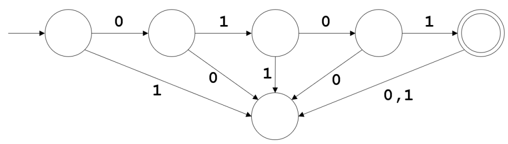
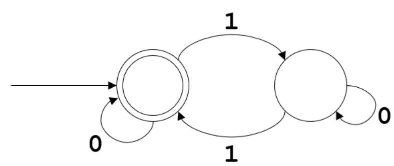
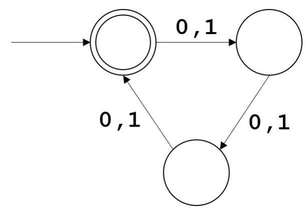
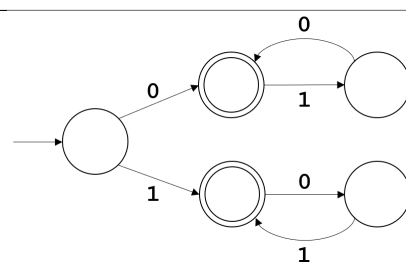
2.1 Clarification on "accept" vs. "recognize"

For any string s and a finite automaton M, M either ***accepts*** or ***rejects*** s.
M ***recognizes*** exactly one language: $\{s|M \text{ accepts } s\}$.

Exercise. For each language below, describe a FA which recognizes it:

1. Σ^*
2. $\emptyset$
3. ε
4. $\{0101\}$
5. $\{w|w \text{ has an even number of 1s}\}$
6. $\{w|w \text{ has length a multiple of 3}\}$
7. $\{w|w \text{ has length a multiple of } n\}$
8. $\{w|w \text{ starts and ends with the same symbol}\}$

Solutions on the next page.

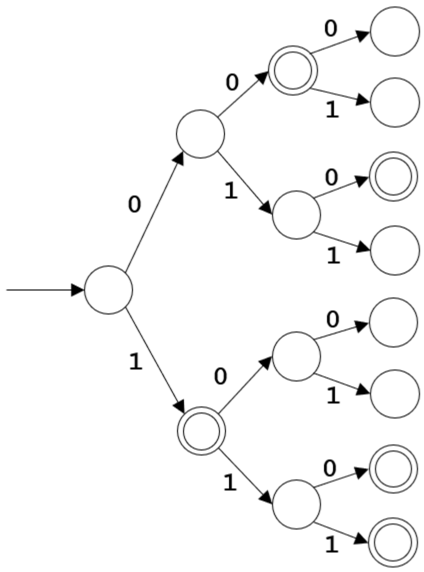
Σ^*	
$\emptyset$	
ε	
$\{0101\}$	
$\{w \mid w$ has an even number of 1s$\}$	
$\{w \mid w$ has length a multiple of 3$\}$	
$\{w \mid w$ starts and ends with the same symbol$\}$	

3 Regular Languages

A language is *regular* if and only if some finite automaton recognizes it. When a problem gives you a regular language (call it A), often your proof will start with something like "Let M_A be a DFA/NFA recognizing A..."

Theorem. Every finite language is regular.

Proof 1: by construction. Start with a concrete example: Assume for sake of illustration that our alphabet is $\Sigma = \{0, 1\}$ and the finite language contains strings of up to length 3: $L = \{1, 00, 010, 110, 111\}$.



The DFA above (where any symbol from the rightmost states goes to a dead state, not pictured) recognizes the language by expressing *all possible strings of length up to 3* in a binary tree form, and accepting when the input so far is in the language. This idea can be extended to any alphabet, and any finite language. As an exercise, how do $|\Sigma|$ and the length of the longest string in L affect this construction? $\square$

Food for thought: How would you do this with an NFA?

Proof 2: using closure. We know (for example, using Exercise 4 above) that any language with only one string is regular. But then, any finite language can be expressed as a union of languages with one string each. Finally, since the regular languages are *closed under union*, we know that any finite language is regular. $\square$

3.1 Note on Regular Languages vs. Regular Expressions

Regular expressions are built using regular operations on certain atomic building blocks, while regular languages are defined as languages recognized by some finite automaton. See the textbook definitions to solidify the distinction.

Although regular languages and regular expressions turn out to have the same expressive power, they were not constructed in the same way; *their equivalence is not by definition,* rather we had to do quite a bit of work to prove it.