Week4 monday

Recap so far: In DFA, the only memory available is in the states. Automata can only "remember" finitely far in the past and finitely much information, because they can have only finitely many states. If a computation path of a DFA visits the same state more than once, the machine can't tell the difference between the first time and future times it visits this state. Thus, if a DFA accepts one long string, then it must accept (infinitely) many similar strings.

Definition A positive integer p is a **pumping length** of a language L over Σ means that, for each string $s \in \Sigma^*$, if $|s| \geq p$ and $s \in L$, then there are strings x, y, z such that

$$s = xyz$$

and

$$|y| > 0, \qquad \text{for each } i \geq 0,\ xy^iz \in L, \qquad \text{and} \qquad |xy| \leq p.$$

Negation: A positive integer p is **not a pumping length** of a language L over Σ iff

$$\exists s \left(\ |s| \geq p \land s \in L \land \forall x \forall y \forall z \left(\ (s = xyz \land |y| > 0 \land |xy| \leq p \ ) \rightarrow \exists i (i \geq 0 \land xy^iz \notin L) \right) \ \right)$$

Informally:

Restating **Pumping Lemma**: If L is a regular language, then it has a pumping length.

Contrapositive: If L has no pumping length, then it is nonregular.

The Pumping Lemma *cannot* be used to prove that a language *is* regular.

The Pumping Lemma **can** be used to prove that a language *is not* regular.

Extra practice: Exercise 1.49 in the book.

Proof strategy: To prove that a language L is **not** regular,

- Consider an arbitrary positive integer p
- Prove that p is not a pumping length for L
- Conclude that L does not have *any* pumping length, and therefore it is not regular.

Example: $\Sigma = \{0, 1\}$, $L = \{0^n 1^n \mid n \geq 0\}$.

Fix p an arbitrary positive integer. List strings that are in L and have length greater than or equal to p:

Pick $s =$

Suppose $s = xyz$ with $|xy| \leq p$ and $|y| > 0$.

Then when $i =$, $xy^i z =$

Example: $\Sigma = \{0, 1\}$, $L = \{ww^{\mathcal{R}} \mid w \in \{0,1\}^*\}$.

Fix p an arbitrary positive integer. List strings that are in L and have length greater than or equal to p:

Pick $s =$

Suppose $s = xyz$ with $|xy| \leq p$ and $|y| > 0$.

Then when $i =$, $xy^iz =$

Example: $\Sigma = \{0, 1\}$, $L = \{0^j1^k \mid j \geq k \geq 0\}$.

Fix p an arbitrary positive integer. List strings that are in L and have length greater than or equal to p:

Pick $s =$

Suppose $s = xyz$ with $|xy| \leq p$ and $|y| > 0$.

Then when $i =$, $xy^iz =$

Example: $\Sigma = \{0, 1\}$, $L = \{0^n1^t0^n \mid m, n \geq 0\}$.

Fix p an arbitrary positive integer. List strings that are in L and have length greater than or equal to p:

Pick $s =$

Suppose $s = xyz$ with $|xy| \leq p$ and $|y| > 0$.

Then when $i =$, $xy^iz =$

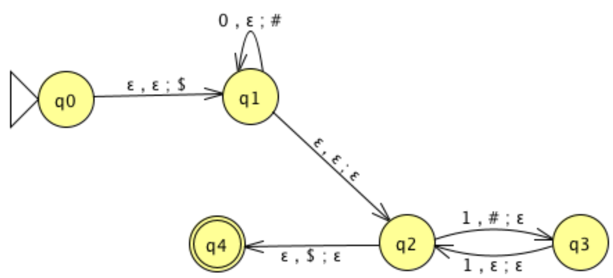
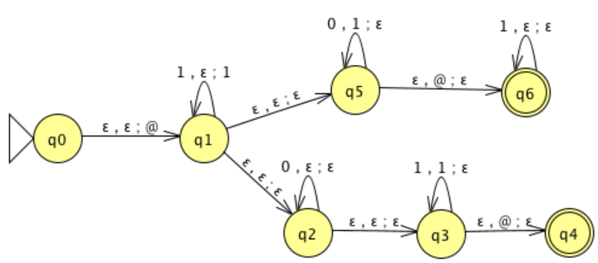
Week4 friday

Consider the state diagram of a PDA with input alphabet Σ and stack alphabet Γ.

Label	means
$a, b \to c$ when $a \in \Sigma$, $b \in \Gamma$, $c \in \Gamma$	
$a, \varepsilon \to c$ when $a \in \Sigma$, $c \in \Gamma$	
$a, b \to \varepsilon$ when $a \in \Sigma$, $b \in \Gamma$	
$a, \varepsilon \to \varepsilon$ when $a \in \Sigma$	

How does the meaning change if a is replaced by ε?

For the PDA state diagrams below, $\Sigma = \{0, 1\}$.

<table>
<tr><th>Mathematical description of language</th><th>State diagram of PDA recognizing language</th></tr>
<tr><td></td><td>
</td></tr>
<tr><td></td><td>
</td></tr>
<tr><td>$\{0^i 1^j 0^k \mid i, j, k \geq 0\}$</td><td></td></tr>
</table>

Assume $a \in \Sigma$ and L is a language over Σ. Recall that

$$aL = \{aw \mid w \in L\}$$

If $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ is a PDA with $L(M) = L$, a PDA M_1 that recognizes aL is

$$M_1 = (\qquad\qquad\qquad , \Sigma, \Gamma, \delta_1, \qquad\qquad , \qquad\qquad)$$

with

Week3 friday

Theorem: For an alphabet Σ, For each language L over Σ,

L is recognized by some DFA
iff
L is recognized by some NFA
iff
L is described by some regular expression

If (any, hence all) these conditions apply, L is called **regular**.

Prove or Disprove: There is some alphabet Σ for which there is some language recognized by an NFA but not by any DFA.

Prove or Disprove: There is some alphabet Σ for which there is some finite language not described by any regular expression over Σ.

Prove or Disprove: If a language is recognized by an NFA then the complement of this language is not recognized by any DFA.

Set	Cardinality
$\{0,1\}$	
$\{0,1\}^*$	
$\mathcal{P}(\{0,1\})$	
The set of all languages over $\{0,1\}$	
The set of all regular expressions over $\{0,1\}$	
The set of all regular languages over $\{0,1\}$	

Pumping Lemma (Sipser Theorem 1.70): If A is a regular language, then there is a number p (a *pumping length*) where, if s is any string in A of length at least p, then s may be divided into three pieces, $s = xyz$ such that

- $|y| > 0$
- for each $i \geq 0$, $xy^iz \in A$
- $|xy| \leq p$.

True or False: A pumping length for $A = \{0, 1\}^*$ is $p = 5$.

True or False: A pumping length for $A = \{1, 01, 001, 0001, 00001\}$ is $p = 4$.

True or False: A pumping length for $A = \{0^j1 \mid j \geq 0\}$ is $p = 3$.

True or False: For any language A, if p is a pumping length for A and $p' > p$, then p' is also a pumping length for A.