Notes: Pumping Lemma

Thursday, 31 January

Upcoming Schedule

Friday, 1 February (10-11:30am): I will have extra Office Hours (Olsson 236A) tomorrow. This is intended primarily for answering questions about Problem Set 1. Please read through the comments handed out today first, and ask any questions you have about them or your answers on Problem Set 1.

Monday, 4 February (2-3pm): Office Hours (Olsson 236A)

Monday, 4 February (5:30-6:30pm): Problem-Solving Session **(Olsson 228E - note room change)**

Wednesday, 6 February (9:30-10:30am): Theory Coffee Hours (Wilsdorf Coffee Shop, I may be at one of the tables upstairs)

Wednesday, 6 February (6-7pm): Problem-Solving Session (Olsson 226D)

Thursday, 7 February: Problem Set 2 is due at the beginning of class.

Proofs

A *proof* is a convincing argument. An argument is a sequence of clearly stated claims. For an argument to be convincing, the first claim must follow from the given assumptions, each subsequent claim must follow from the previous claims, and the final claim must be equivalent to the statement you are proving. Please read through the comments on PS1 carefully to understand what makes a convincing proof. If you are writing a long (more than 3 lines) proof, you should include a sentence or two at the beginning explaining your overall proof idea, and enough prose in your proof to make it clear to a reader how the steps in your argument are connected, and how they show the statement you are proving is true. The proofs we do in cs302 will involve only a few main types of argument:

Proof by Construction — If you can express the statement you are proving in the form, *An X exists*, then you can prove the statement by showing how to make an X. An example is the proof we did in the previous class that NFAs and DFAs are equivalent. We can state this as two parts: (1) For any DFA, an NFA exists that recognizes the same language. (2) For any NFA, a DFA exists that recognizes the same language. For the first part, our premise is a DFA M exists that recognizes some language A. To prove 1, we need to show that *an X exists*, where X is an NFA that recognizes the same language A. This direction is easy: we can construct the NFA directly from M. For direction (2), we needed to show how to construct a DFA that recognizes the same language as any given NFA. This was tougher, but we were able to show it could be done by making DFA states to represent each set of possible states in the NFA.

Many of the constructions we will do in this class are of the form, *for any Y, there is an X* (e.g., for any *NFA*, there is *a DFA that recognizes the same language*). For a construction proof to be convincing, it needs to explain that the result of the construction is an X (often, the way it is constructed is enough to be convincing for this already), and the construction method has to be general enough to be clear that it works for all possible Y. In the PS1 comments, we used proof by construction for problems 8 and 9.

Proof by Contradiction — To prove X, start by assuming X is not true, and show that it leads to a conclusion that is obviously untrue. In the PS1 comments, we used proof by contradiction for

Problem 4b. In this class, we will often combine the proof by construction and proof by contradiction strategies by showing that if we had an X, we could use it to build a Y, but we know Y cannot exist. This proves that X cannot exist.

Proof by Induction — We use proof by induction when we need to prove something is true for all elements of some infinite set that can be generated inductively. To prove something is true for all elements in the infinite set we need to (1) *basis step* prove it is true for some finite number of elements of the set (usually just one); and (2) *induction step* prove that if it is true for some element of the set, it is also true for all elements of the set that can be generated from that element.

For example, the natural numbers can be generated by: 1 is a natural number; if n is a natural number, so is $n + 1$. So, if we want to prove $P(n)$ is true for all natural numbers, we could prove $P(1)$, and then prove that if $P(n)$ is true, it implies $P(n + 1)$. Since we can generate all the natural numbers by starting with $N = 1$ and with the generation rule, if $x \in N$ then $x + 1 \in N$. All sets can be generated by: $\emptyset$ is a set; if s is a set, adding one element to s produces a set. So, if we want to prove $P(S)$ for all sets, we start by proving $P(\emptyset)$, and then prove that if $P(S)$ is true for any set S where $|S| = n$, it implies $P(S \cup x)$ is true for any element x.

Pumping Lemma

> If A is a regular language, then there is a number p (the pumping length) where for any string $s \in A$ and $|s| \geq p$, s may be divided into three pieces, $s = xyz$, such that $|y| > 0$, $|xy| \leq p$, and for any $i \geq 0$, $xy^iz \in A$.

Informal argument: if $s \in A$, some part of s that appears within the first p symbols must correspond to a loop in a DFA that recognizes A. We call y the string along that loop. Since it is a loop in the DFA, we can go around the loop any number of times without changing the acceptance result for A.

Game view: Suppose A is some language. Player one picks $n \in N$, the maximum number of states, and attempts to design a DFA M that recognizes A using n or fewer states. Player two picks a string s. Player two wins if she can find a string s that is not processed correctly by M (that is, either $s \in A$ and M rejects, or $s \notin A$ and M accepts). If there is a strategy player two can use to always win no matter what value player one picks for n, then A is not a regular language. Otherwise, A is regular.

We will use the pumping lemma to prove a language is non-regular using proof by contradiction. The proof steps will always be similar to:

1. Assume A is regular.
2. Then, the pumping lemma is true for A.
3. This means there is some number p, such that or any string $s \in A$ and $|s| \geq p$, s may be divided into three pieces, $s = xyz$, such that $|y| > 0$, $|xy| \leq p$, and for any $i \geq 0$, $xy^iz \in A$.
4. **The creative part:** Identify a string s that can be built using p and show that there is no way to divide $s = xyz$ such that $|y| > 0$ and $xy^iz \in A$ for any $i \geq 0$.

We can condence the first 3 steps into the statement, "Assume A is regular and p is the pumping length for A."

Example. Prove the language $\{0^n1^n | n \geq 0\}$ is not regular.