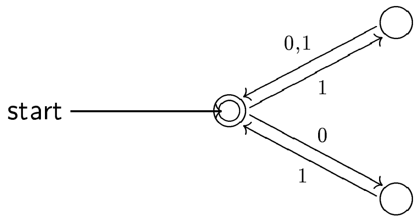
**CS154: Introduction to Automata and Complexity Theory** **Fall 1997**

# Midterm Solutions

PROBLEM 1 **a. Construct an NFA accepting the language defined by the regular expression $((01 \cup 10)^* \cup 11)^*$. b. Construct a CFG accepting the language $L = \{0^m 1^n \mid m \leq 2n\}$.**

**(a)** The RE is equivalent to $(01 \cup 10 \cup 11)^*$. (You can prove this using the identity $(\alpha \cup \beta)^* = (\alpha^* \beta^*)^*$ — though this identity itself needs proof.) Does this look familiar? From PS 2:




**(b)** The point here is that each 0 must be associated with at most 2 1s. This problem caused the most loss of points in the entire midterm, mostly because people enforced $n \leq m \leq 2n$ rather than $0 \leq m \leq 2n$. The grammar is

$$S \rightarrow 00S1 \mid 0S1 \mid S1 \mid \epsilon$$

PROBLEM 2 **Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA where $m = |Q|$. Prove that if $M$ accepts a word of length greater than $m$ then $L(M)$ must be infinite.**

Here's a handwavy proof that was pretty much all we expected under the timed pressure of the midterm: Let $s_i$ be the state $M$ is in after reading $w_i$. If $|w| > |Q|$, by the pigeonhole principle two of the $s_i$ must be the same. Thus there is a "loop" starting and ending in $s_i$ which we can pump an arbitrary number of times, yielding an infinite number of strings accepted by $M$.

To make this formal, we have to prove that you really can pump the loop. This is similar to the proof of the pumping lemma. Suppose $s_i = s_j (i \neq j)$. Then $w^{(k)} = w_1 \cdots w_{i-1}(w_i \cdots w_{j-1})^k w_j \cdots w_n$ is the string pumped $k$ times. We claim:

> Let $w_a$ be the $a^{th}$ character in $w$. Then $M$ is in the same state after reading $w_a$ (which may happen many times if $w_a$ is in the repeated part of $w^{(k)}$) in $w^{(k)}$ as it is after reading $w_a$ in $w$.

This claim implies that after reading $w_n$, $M$ will be in the same state for $w^{(k)}$ as it is for $w$. Since $M$ accepts $w$, this same state must be a final state, so $M$ accepts $w^{(k)}$ as well.

Proof of the claim is by induction. Obviously, after reading $w_1$, $M$ will be in state $\delta(q_0, w_1)$ for all strings. Now, suppose the $w$ and $w^{(k)}$ are all in the same state $q$ after reading $w_a$. Then what state will they be in after reading $w_{a+1}$? Well, $w$ will be in state $\delta(q, w_{a+1})$. $w^{(k)}$ will be as well, unless $a = j - 1$ and we are not on the last repetition of the loop. In this case $w$ will be in state $\delta(q, w_j) = s_j$, while $w^{(k)}$ will be in state $\delta(q, w_i) = s_i$. But $s_i = s_j$ (remember, that's how $i$ and $j$ were chosen), so everyone is in the same state in this case too.

This argument falls apart for $k = 0$ (though a slightly more sophisticated argument works) because there's no concept of "last loop" in the $k = 0$ case. So we'll limit $k$ to be greater than 1. We still get an infinite number of strings $w^{(k)}$ accepted by $M$, so $L(M)$ is infinite.

**Problem 3** **Consider the alphabet $\Sigma = \{(0,0)\ ,\ (0,1)\ ,\ (1,0)\ ,\ (1,1)\}$. Prove that the language**

$$L = \left\{ (a_1, b_1)(a_2, b_2) \ldots (a_k, b_k) \mid k > 0 \textbf{ and } \sum_{i=1}^{k} a_i = \sum_{i=1}^{k} b_i \right\}$$

**is not regular. Give an example of a regular, non-empty subset of $L$. Justify your answer.**

The point is that the $(0,1)$s must each have a matching $(1,0)$. Really this is $a^n b^n$ in a different guise. Thus, I won't even bother to pump but will use closure properties (which I love).

Consider $L \cap (0,1)^*(1,0)^*$. If the result is not regular, neither is $L$, since the intersection of two regular languages is supposed to be regular (and $(0,1)^*(1,0)^*$ is certainly regular). Any time you intersect with $(0,1)^*(1,0)^*$, you get a language consisting only of strings of the form $(0,1)^i(1,0)^j$. But in this case $i$ must equal $j$, or else the sum constraint of $L$ won't be satisfied. Thus the intersection language is $(0,1)^n(1,0)^n$. We know this language isn't regular, so neither is $L$.

As for the regular, non-empty subset, try the tiny language $\{(0,0)\}$. It's finite and hence regular.

**Problem 4** **Prove that the language $L = \{0^p \mid p \textbf{ prime}\}$ is not context free. Give an example of a non-empty subset of $L$ that is context free. Justify your answer.**

This is the easiest context-free pumping you'll ever see, because there's only one way to divvy up $w$. As is normal in pumping, the adversary gives us a minimum string size $m$. We pick $w = 0^p$ for some prime $p > m$. The adversary divides $w$ into $w = uvxyz$. We then pump $p+1$ times to get $uv^{p+1}xy^{p+1}z$. The length of the pumped string is $|uvxyz| + |vy| \times p$. But $|uvxyz| = |w| = p$, so the total length is $(1 + |vy|)p$, which is certainly not prime. Since we pumped to get a string not in $L$, $L$ is not context free.

Again, any finite subset of $L$, say $\{00\}$, is context-free by dint of being finite.

**Problem 5** **Answer the following True/False questions. Justify your answer by giving counterexamples or proof sketches where necessary.**

1. **All finite languages are regular.**
2. $L \cdot L = \{ww : w \in L\}$**.**
3. **If $L$ is a regular language, so is every subset of $L$.**
4. **If $L$ is not context free then neither is any subset of $L$.**
5. **If $L$ is context free and $F$ is a finite language then $L \cap F$ is context free.**

1. **True.** It's easy to write a regular expression for a finite language: $(w_1) \cup (w_2) \cup \cdots \cup (w_n)$. The RE is finite since $n$ is, so we get a perfectly legal regular expression for $L$.

2. **False.** Consider $L = \{a, b\}$. Then $L \cdot L = \{aa, ab, ba, bb\}$, but $\{ww : w \in L\} = \{aa, bb\}$.

3. **False.** $L = a^*$ is regular, but $L' = \{a^p : p \text{ prime}\}$ is a subset of $L$ that isn't regular. In general, note that for any alphabet $\Sigma$, $\Sigma^*$ is regular, but *everything* is a subset of $\Sigma^*$, and most things aren't regular.

4. **False.** The contrapostive of 3. Any finite subset of $L$ is context free, because it's regular. In particular, the empty language is a subset of all context-free languages, and it's certainly context-free. (3) and (4) together imply that you can't say anything about a language just based on what it's a subset of.

5. **True.** You can appeal to the problem set, where you proved a CFL intersected with a RL is context-free, but it's easier just to note that any language intersected with a finite language is finite, and therefore regular, and therefore context free.