Unit C4 Supplement, 3/25/05

Pumping Lemma (for regular languages)

Since I was not able to explain this well in class, I will add a few notes. In addition, the slides are in an abbreviated form. So, I will try to give more details here.

First, the Pumping Lemma says the following (informally). Given an infinite regular language, we can always find a (sufficiently long) string (w) that has a substring (y) contributing to (uncoordinated) repetition, and that the substring can be pumped zero or more times (y^*). Although we don't prove this lemma, we can intuitively tell this is true.

As discussed in class, this lemma is used only for showing that certain languages are not regular. To do this, we take the role of a falsifier and face a competing verifier.

- As a falsifier, choose an infinite (supposedly) regular language $L = 0^n1^n$ (shorthand for $\{0^n1^n \mid n \geq 0\}$). If this was indeed regular, the falsifier could never win.
- The opponent, a verifier, chooses a positive integer n_0, say, 4. This could be any other positive integer, though.
- We choose $w = 0^21^2 \in L$, which satisfies $|0^21^2| = 4 \geq n_0 = 4$.
- The opponent can decompose w. The opponent only needs to choose the decomposition most damaging to us. However, since we don't know what it is, we consider all the possibilities of decomposing $w = xyz$, where $|xy| \leq n_0 = 4$ and $|y| \geq 1$
  - Case 1 (y stays within 0^2)
    - Subcase A: $x = 0$, $y = 0$, z = 11
    - Subcase B: $x = \varepsilon$, $y = 00$, z = 11
  - Case 2 (y stays within 1^2)
    - Subcase A: $x = 00$, $y = 1$, z = 1
    - Subcase B: $x = 00$, $y = 11$, z = ε
  - Case 3 (y crosses 0^21^2)
    - Subcase A: $x = 0$, $y = 01$, z = 1
    - Subcase B: $x = \varepsilon$, $y = 001$, z = ε
    - Subcase C: $x = 0$, $y = 011$, z = ε
    - Subcase D: $x = \varepsilon$, $y = 0011$, z = ε
  - Note: Depending on the choice made in the previous steps, there may be more cases/subcases. But the idea is analogous.
- We choose $i = 2 \geq 0$. Then, for Cases 1 and 2, the balance between 0's and 1's will be destroyed. For Case 3, the 0-to-1 ordering will be destroyed. In any case, we defeated the opponent. This means that we have rejected the Pumping Lemma. However, since the lemma holds for any infinite regular languages, we conclude that the language is not regular. [Strictly speaking, this is a proof by contradiction.]

The problem $w\$w^R$ can be analyzed in an analogous way. In particular, you might consider different cases: whether or not y includes \$.

For ww^R, a similar technique may not work well. Recall the class discussion, where you are forced to pump the middle section of, say, 00<u>11</u>00. You *can* pump and will still get strings that are in the language. To see how to deal with this problem, you may want to re-visit 0^n1^n and re-examine the choices made by the players. There are ways for the falsifier to defeat the verifier more easily. For example, what would happen if you consider w twice as long as n_0 (instead of $|w| = n_0$ as in the earlier example)? If it leads to a simpler strategy, what makes it happen? Such an idea may be useful for showing that ww^R is not regular.

If you have questions, try to post on the discussion board, although you can always send them to me.

// End