CS 3100 – Models of Computation – Fall 2011

Assignment 8, Posted on: 10/20, Due 10/27 midnight
This assignment is worth 8% of the total points for assignments
100 points total

1 Logic, Regular Languages, Ultimately Periodic Sets

1.1 Implication (if-then, $\Rightarrow$) and Bi-implication (iff, $\Leftrightarrow$)

$a \rightarrow b$ is equivalent to $\neg a \vee b$ or "if a then b."
Example: For any language L, $Finite(L) \Rightarrow Regular(L)$.

This also means that
$\neg Regular(L) \Rightarrow \neg Finite(L)$.
But it **does not** mean $\neg Finite(L) \Rightarrow \neg Regular(L)$.
IF IT WERE THE CASE THAT $Finite(L) \Leftrightarrow Regular(L)$, then we could have drawn two conclusions:

- $Finite(L) \Rightarrow Regular(L)$.
- $\neg Finite(L) \Rightarrow \neg Regular(L)$.

But we only have $Finite(L) \Rightarrow Regular(L)$.
Try this:

- Convert $Finite(L) \Leftrightarrow Regular(L)$ to contain $\wedge$, $\vee$, and $\neg$ only.

1.2 What all this means

This means there can be *infinite languages that are regular*, as well as *infinite languages that are non-regular.* Two examples:

$$L_{zebraPattern} = \{0^i \mid i \geq 0\}$$

is infinite and regular.

However,

$$L_{warpedSlinky} = \{0^{i^2} \mid i \geq 0\}$$

is infinite and non-regular.

1.3 Quantification Operators ($\forall$, $\exists$) and Quantified Propositions

$\forall x$ is a compact way to write a repeated conjunction.
$\exists x$ is a compact way to write a repeated disjunction.
Examples:

- $\forall x \in Nat : prime(x) \Rightarrow (x = 2) \vee odd(x)$ means

  $(prime(0) \Rightarrow (0 = 2) \vee odd(0)$

$\wedge$

$(prime(1) \Rightarrow (1 = 2) \vee odd(1)$

$\wedge$

$(prime(2) \Rightarrow (2 = 2) \vee odd(2)$

$\wedge$

$(prime(3) \Rightarrow (3 = 2) \vee odd(3)$

$\wedge$

$(prime(4) \Rightarrow (4 = 2) \vee odd(4)$

$\wedge$

$\ldots$

Try this:

- Simplify $\neg(\forall x \in Nat : prime(x) \Rightarrow (x = 2) \vee odd(x))$ and express it in the above form.

2 (Incomplete) Pumping Lemmas – what we will study

Use our Pumping Lemma versions to *Prove that something is NOT regular*. To see a complete version, see the chapter excerpt from my book that I have placed online.

2.1 Basics of OUR (incomplete) Pumping Lemmas

Let us first consider singleton alphabets. One immediate observation we can make is this:

- All infinite regular languages over a singleton alphabet have a DFA that is *lasso shaped*.

The lengths of strings form an ultimately periodic set (if you look far enough beyond some point p, each element after p in the set is some multiple of k steps away from p through $p - k$).

This is not true of $L_{warpedSlinky}$ and so we at least get one fact established: the *lengths* of strings in a regular language is ultimately periodic.

We have two incomplete Pumping Lemmas that we can use: the one in Section 12.1 of my book, and the one in 12.1.1 of my book.

Here is the proof structure:

- Suspect that L is not regular. Look for periodicity, finiteness, etc. Be virtually sure it is not regular. Then go to the next step.

- *Lengths of strings is not a certain criterion.* If the lengths are not periodic, then not regular. But not the other way. Example:
$$\{0^n 1^n \mid n \geq 0\}$$

- Decide to attack straight using the incomplete PL or apply a closure property to simplify the problem (see my hand-written notes of L15 for examples; also my book chapter).
- Then proceed to apply the incomplete PL to show that L is not regular.
- Ask the adversary (who claims L is regular) for a number m representing the number of states in the minimal DFA of L.
- Pick a string w of length $\geq m$.
- Split w in an interesting way.
- Show that there are pumps that go outside of L.
- But if L were regular, all pumps stay inside L.
- Hence, L can't be regular!

2.2 Jflap Pumping Lemma Tutor

Read a full description at `http://www.cs.duke.edu/csed/jflap/tutorial/pumpinglemma/regular/index.html`. This follows the 12.1 version of my Pumping lemma. Here, I provide some extra details.

If the user (U) goes first and wins (a "YOU WIN" is shown), then, assuming that their "computer" moves are designed well, the language is regular. That means, we can 'pump'. That means, we can pump and stay in the language for all pumps.

Thus, if U goes first and gets a "try again," after many tries, the user assumes he/she lost—the language is then likely *not regular*. Sit down and prove it at that point (also hit the "Explain" button).

If the computer (C) goes first and the *user wins* (a "YOU WIN" is shown), then it means the computer loses. That is, the user was smart in picking a partition such that the pump went outside the language. That means, the computer could not, *in all cases*, pick a pump and stay in the language (that would be the case for a regular language). So we have evidence (in the form of the user chosen partition) that we can go outside the language at least in one case. The language is *not* regular.

But if C goes first and the user keeps getting "try again," the language is likely regular.

3 Your Assignment

Submit your answers as simple writeups in English. Be brief, yet complete. To make grading easier, please provide (or summarize) the question and then your answer. Submit as PDF or TXT into asg8 via handin.

1. **(10 points) for each language** Let the computer go first. Select these languages:

$$L = \{a^n b^k c^{(n+k)} \mid n \geq 0,\ k \geq 0\}$$

$$L = \{a^n b^l a^k \mid n > 5,\ l > 3,\ k \leq l\}$$

Show how you won (which means you could prove that the language is not regular). Then understand the explanation provided, and copy it down in your answer.

2. **(10 points) for each language** Let the computer go first. Select two languages from the JFLAP Pumping Lemma tutor for which you are guaranteed to get "Please try again." Show that to be the case for at least three tries for each language. Then understand the explanation provided, and copy it down in your answer.

3. **(20 points)** Do 12.6 from my book chapter.

4. **(3 points) for each part** Do 12.7, parts 1, 2, 3, 4, 5, 8, 9, 10, 11, 12 from my book chapter.

5. **(10 points)** Write a summary of the proof of "U" versus "Y" given in my book. What would have been a simpler proof for that problem? Explain in a few sentences.

4 Extra Credit Problem - 15 points extra

Prove that the following language in the Pumping Lemma tutor is regular (draw a DFA for it). (Note: this is not as hard as it sounds—simplify the language first.)

$$L = \{b^5 w \ \mid \ w \in \{a,b\}^*, \ \ (2n_a(w) + 5n_b(w)) \ mod\ 3 = 0\}$$

Here, $n_a(w)$ denotes the number of as in string w, and so on.