Introduction to Theoretical Computer Science

Exercise Sheet: Week 3. Answers

Here are a few routine exercises.

1. Consider the language $L = \{ww \mid w \in \{\mathtt{a}, \mathtt{b}\}^*\}$. Show that it is not regular.

   Consider the sequence $u_i = \mathtt{a}^i\mathtt{b}$ for $i \in \mathbb{N}$. Then for each i and j (where $i \neq j$), we have the word $w_{ij} = u_i$ such that $u_i w_{ij} \in L$ and $u_j w_{ij} \notin L$. Thus as the sequence u_i is infinite, there are infinite $\equiv_L$ equivalence classes, and therefore L is not regular by the Myhill-Nerode theorem.

   This question could also be done via pumping using $\mathtt{a}^p\mathtt{b}\mathtt{a}^p\mathtt{b}$ as your choice of string where p is the pumping length.

2. Is the language $L' = \{ww \mid w \in \{\mathtt{a}, \mathtt{b}\}^* \wedge |w| < 4\}$ regular? Why or why not?

   Yes. It is a finite language (there are finitely many strings of length < 4), and thus is regular, as all finite languages are regular.

3. We saw in lectures that the language L is not context-free. Show that, by contrast, its complement *is* context-free.

   It can be described by the following CFG:

   $$\begin{array}{lcl} S & \to & A \mid B \mid AB \mid BA \\ A & \to & \mathtt{a} \mid \mathtt{a}A\mathtt{a} \mid \mathtt{b}A\mathtt{a} \mid \mathtt{a}A\mathtt{b} \mid \mathtt{b}A\mathtt{b} \\ B & \to & \mathtt{b} \mid \mathtt{a}B\mathtt{a} \mid \mathtt{b}B\mathtt{a} \mid \mathtt{a}B\mathtt{b} \mid \mathtt{b}B\mathtt{b} \end{array}$$

   A and B are any odd-length string with $\mathtt{a}$ and $\mathtt{b}$ in the centre, respectively. The only words we must exclude are words that are even length, comprised of two identical odd-length strings. But the only even-length strings accepted here are comprised of two odd-length strings with different centre characters (AB and BA).

   This one may take a while for everyone to understand.

The following questions/comments are intended as prompts for discussion. Of course, you can ask/discuss about anything. Some of these topics we've touched on in discussion in lectures – this is an opportunity to think about them a bit more.

4. Suppose we augmented a finite automaton with an additional feature: a single mutable variable. On each transition, it may read from or write to this variable.

   a) If the variable is of type Σ (that is, it can contain one alphabet symbol), what class of languages can such automata recognise?

      Regular languages. The number of Σ symbols is finite, so we could convert these automata to regular automata by having $|\Sigma|$ copies of every state.

b) If the variable is of type $\mathbb{N}$, what class of languages can they recognise?

Assuming we can do basic arithmetic operations, this automaton could recognise any Turing-recognisable (recursively enumerable) language. This is because almost all reasonable domains can be encoded into a single natural number. Discuss how to do this. Also discuss what domains might not be so encodable (e.g. infinite binary strings).

5. A queue automaton is like a pushdown automaton, but with a queue (fifo) instead of a stack (lifo). These can recognise any Turing-recognisable language. Now suppose we instead defined a pushdown automaton with an additional stack. What class of languages can these 2-stack PDAs recognise? *By simulating a queue with two stacks, you can emulate a queue automaton. Hence they also recognise the Turing-recognisable languages.*