Problem Set 6

Spring 10

Due: Thursday 2pm, 18th March, in class before the lecture begins.
Please follow the homework format guidelines posted on the class web page:
`http://www.cs.uiuc.edu/class/sp10/cs373/`

1. [**Category**: Closure properties, **Points**: 40]

   Prove that the following languages are not regular using only closure properties. You may assume that regular languages are closed under union, intersection, concatenation, complement, reverse and homomorphism. You may also assume that $\{0^n1^n \mid n \geq 0\}$ is not regular.

   Closure under homomorphisms: We define homomorphisms as follows. Let Σ and Π be two finite alphabets. A homomorphism is a function $h : \Sigma \to \Pi^*$, that maps a symbol from the alphabet Σ to a word in Π^*. We can extend this function to strings: $h(x_1x_2 \ldots x_n) = h(x_1) \cdot h(x_2) \cdot \ldots \cdot h(x_n)$, where $x_1 \ldots x_n \in \Sigma^*$. Intuitively, we just transform each symbol of the string and then concatenate everything together. We can now extend this function to languages: $h(L) = \{h(w) \mid w \in L\}$. It now turns out that regular languages are closed under homomorphisms.

   Theorem. Closure under homomorphisms: If L is a regular language over Σ and $h : \Sigma \to \Pi^*$ is a homomorphism, then $h(L)$ is also regular.

   For example, if $\Sigma = \{0, 1, 2\}$ and $\Pi = \{a, b\}$, and h is a homomorphism that maps $h(0) = ab$, $h(1) = \epsilon$ and $h(2) = bb$, then since $L = (01)^*12^*$ is regular, $h(L) = (ab)^*(bb)^*$ is also regular. You may assume the above theorem to solve the problems below.

   Prove that these languages are not regular:

   (a) $L_k = \{0^{kn}1^{kn} \mid n >= 0\}$ for any fixed k (Hint: do this one first $\{0^{2n}1^{2n} \mid n \geq 0\}$)

   (b) $\{0^{2n}1^{3n} \mid n >= 0\}$ (Hint: use homomorphism)

   (c) $\{0^n\$1^n\$\$^* \mid n >= 0\}$

   (d) $\{w_1\$w_2\$w_3 \mid w_1, w_2, w_3 \in \{0, 1\}^*$ number of 0s in w_1 is equal to the number of 1s in $w_2\}$

   (e) $\{0^n\$0^m \mid 0 \leq n \leq m\}$

2. 2D Tape TM [**Category**: Simulation., **Points**: 20]

   A *two-dimensional Turing machine* is like a Turing machine except that instead of a one-dimensional tape it has a two-dimensional tape that is like the upper right quadrant of the plane, infinite in *up* and *right* directions. It has a finite input alphabet Σ and a finite tape alphabet Γ. If $x \in \Sigma^*$ is the input, $|x| = n$, the machine starts in its start state s with x written in tape cells $(0, 0), (0, 1), \ldots, (0, n-1)$. It has a read/write head initially pointing to the origin $(0, 0)$. In each step, it reads the symbol of Γ currently occupying the cell it is scanning. The transition of the Turing machine tells the TM, when reading a symbol and when at the current state of the finite control,

to rewrite the symbol to another one in Γ, and moves the head either *up, down, left* or *right*, and move to a new state. Prove that two-dimensional TMs and normal TMs are equivalent. Describe the simulations *informally* but in sufficient detail that the simulation is understandable, clear and easy to see it's correct.

3. Enumeration [**Category**: Comprehension, **Points**: 20]

   Fix an alphabet Σ. Assume we fix some encoding to represent regular expressions over Σ and DFAs as binary strings. (so as usual $\langle R \rangle$ represents the binary encoding of the regular expression X). Assume also that there is some encoding of DFAs as binary strings.

   Consider the following language:

   $L = \{\langle R \rangle \mid R$ is a regular expression such that there is some DFA D with language $L(R)$ and $\langle D \rangle$ is a prime number written in binary.$\}$

   In other words, L is the set of all encodings of regular expressions R such that some DFA D, whose binary representation represents a prime number, accepts the same language as that of R.

   Show that L is TM-recognizable by constructing a TM recognizing L. Your construction could be pseudo-code or an exact explanation of how it works. You don't need to do a formal low-level construction, but just intuitively describe a high-level construction.