Lecture #8: Modelling Computation of Languages
Exercises and Review

Questions for Review

1. What is a ***rewrite rule***? What other names for this can be found in the literature?
2. How is a rewrite rule applied, to produce one string from another?
3. What is a ***String Rewriting System***? What is another name for *this* that can be found in the literature?
4. What is a ***derivation***? What is the ***length*** of a derivation?
5. How are the languages of nondeterministic Turing machines, and String Rewriting Systems, related? *Briefly* describe how this can be proved.
6. What else can be said if a ***deterministic*** Turing machine is being "simulated" by a rewriting system in this way?
7. Suppose $\mathcal{M}$ is a (nondeterministic or deterministic) Turing machine.
   (a) What is $\Sigma(\mathcal{M})$?
   (b) What is $\Omega(\mathcal{M})$?
8. What is a ***phrase-structure grammar***? Please describe each of the *components* of such a grammar when answering this question.
9. What is the ***language*** of a phrase-structure grammar $G = (V, \Sigma, \Pi, S)$?
10. What is a ***Markov algorithm***? Discuss how the differences between the use of string rewriting rules in Markov algorithms, and in phrase structure grammars or string rewriting systems, help to ensure that computations by Markov algorithms are deterministic.
11. Describe the relationships between "computations using phrase-structure grammars", "computation by Markov algorithms", and "computations by Turing machines" as precisely as you can.

Practice Problems

These problems — which continue activities in the lecture presentation — will not be discussed during the lecture, and solutions for these problems will not generally be made available. They can be used as “practice” problems that can help you to practice skill considered in the lecture presentation for Lecture #8.

1. Let $\Sigma = \{\mathtt{a}, \mathtt{b}\}$. Describe a ***string rewriting system*** Π for strings over the alphabet Σ such that, for all $\mu, \nu \in \Sigma^\star$. $\mu \Rightarrow_\Pi^\star \nu$ if and only if the length of ν is less than or equal to the length of μ.

   Then sketch a proof that your answer is correct.

2. Once again, let $\Sigma = \{\mathtt{a}, \mathtt{b}\}$. Design a phrase-structure grammar $G = (V, \Sigma, \Pi, S)$ with language
$$L = \{\mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}\}.$$

   Then sketch a proof that your answer is correct.

3. Let $\Sigma_D = \{\mathtt{0}, \mathtt{1}, \mathtt{2}, \mathtt{3}, \mathtt{4}, \mathtt{5}, \mathtt{6}, \mathtt{7}, \mathtt{8}, \mathtt{9}\}$. As in the lecture presentation, let $L_{\text{unpad}} \subseteq \Sigma_D^\star$ be the set of strings that are *unpadded* decimal representations of non-negative integers, and let $f_{-1} : \Sigma_D^\star \to \Sigma_D^\star$ be a function satisfying the following properties, for all strings $\omega \in \Sigma_D^\star$.
   - If ω is the unpadded decimal representation of a *positive* integer n, then $f_{-1}(\omega)$ is the unpadded decimal representation of $n - 1$.
   - If $\omega = \mathtt{0}$ (the unpadded decimal representation of zero) or $\omega \notin L_{\text{unpad}}$, then $f_{-1}(\omega) = \lambda$, the empty string.

   Design a Markov algorithm that computes the function f_{-1} and sketch a proof that your algorithm is correct.