CSE596, Fall 2012 **Problem Set 3** **Due Wed. Oct. 3**

The **First Prelim Exam** will definitely be on **Wed. Oct. 10**, in class period. This is the last problem set before it; hence it is also longer.

Office Hours (final): Regan Mon. 2–4pm and Tue. 1–3pm, 326 Davis; Hughes Thursdays 11am–noon, Fridays 11:30am–class time, plus the Mondays 11–11:50am **Review Sessions** in **19 Clemens**.

Reading. For Friday next week, read Chapter 3 through Section 3.5, although Section 3.5 may come in the following week. One stylistic difference is that I won't emphasize pairing functions per-se, since the current material only requires that the pairing be one-to-one. Instead I will suppose that the original alphabet Σ is enlarged to an alphabet Σ' that includes an extra "separator symbol" such as '#' that is not in Σ. Then strings $x, y \in \Sigma^*$ are paired simply by writing $x\#y$ as a string over Σ'. If needed or desired, we can re-code Σ' back to strings over Σ by something like $0 \mapsto 00$, $1 \mapsto 11$, and $\# \mapsto 01$. Unlike with a pairing function this encoding isn't *onto* Σ^* (or onto $\mathbf{N}$), because it doesn't have strings like 10 in its range, but its being 1-1 is good enough to define languages such as the language of the Halting Problem. Besides simplicity, another advantage is that arbitrary tuples can be encoded as $x_1\#x_2\#\cdots\#x_n$ without having to "cascade" the pairing function.

In the latter two problems, the alphabet called "Σ" is unspecified. You should give your answers in such a way that the particular identity of Σ does not matter. Generally you may assume Σ includes $\{0, 1\}$, and you may speak as though Σ is no larger than the set ASCII of (printable) ASCII characters. Your answers may lead you to enlarge Σ, or rather enlarge the work alphabets "Γ" of machines being described. This is fine. Ultimately ASCII or larger alphabets such as UNICODE can be re-coded as fixed-size words over $\{0, 1\}$. At various points in the course (especially Section 5.1) we will probe whether this matters.

(1) Given any fixed language L over an alphabet Σ, define its "Myhill-Nerode dimension" *at strings of length* m by $f_L(m)$ = the maximum cardinality of a distinctive set $S \subseteq \Sigma^m$. That is, the maximum number of strings of length m that are all inequivalent under the relation

$$x \sim_L y \longleftrightarrow (\forall z \in \Sigma^*)[L(xz) = L(yz)].$$

For example with $L = \{a^n b^n : n \geq 0\}$, the dimension is $m + 2$, because all of the strings $a^m, a^{m-1}b, a^{m-2}bb, \ldots, b^m$ are distinctive, plus one more equivalence class for "dead prefixes" like $ba\ldots$.

Prove that when L is the language of palindromes over $\Sigma = \{a, b\}$, $f_L(m) = 2^m$. (18 pts.)

(2) Now let $\Sigma = \{(,)\}$, and let L be the language of balanced-parenthesis strings. Show that now $f_L(m) = O(m)$. Can you give an exact value?

There is a general theorem that to decide a language L on a *single-tape* Turing machine requires time $\Omega(n \log f_L(n))$. The idea is that $\log_2 f_L(n)$ bits of information need to be "ferried" along the tape, though the proof itself is advanced. Instead, however, **sketch** how the "ferry" idea can be used on a virtual second "track" to create a 1-tape TM that decides this language in time $O(n \log n)$. (18 + 24 = 42 pts.)

(3) Text, "Homework 3.5" on page 50. That is, given programs M_1 and M_2, show how to design programs M_3 and M_4 so that $L(M_3) = L(M_1) \cup L(M_2)$ and $L(M_4) = L(M_1) \cap L(M_2)$. (Although you can technically use any of the equivalent definitions of "c.e." to come in the Friday 9/29 lecture, the intent is to use the original one: a language A is c.e. if there is a program M (whether Turing machine or C or Pascal or Java or other high-level language program) such that $L(M) = A$. Use as a primitive the idea of simulating one next step of a program. You need not give Turing machine arc-node level detail—flowcharts or pseudocode sketches are fine. 12 pts.)

(4) Suppose A is a c.e. language, f and g are computable total functions, and B is a language such that for all $x \in \Sigma^*$, $x \in B \iff f(x) \in A$, and $x \notin B \iff g(x) \in A$. Show that B is decidable. (12 pts., for 84 on the problem set)