Assignment 1 **Due: Oct 9** **CAS705 Fall 2013**

Problem 1. Suppose that $x, y, z \in \Sigma^+$, i.e., they are three non-empty strings over some alphabet Σ. Show the following:

$xy = yz \iff$ there exists a $u \in \Sigma^+$, and a $v \in \Sigma^*$, and an integer $i \geq 0$, such that $x = uv$, $z = vu$ and $y = (uv)^i u = u(vu)^i$.

Solution: This problem is actually one of the "Lyndon-Schützenberger" theorems; it is a way of characterizing under what conditions can a string have non-trivial proper prefix and suffix that are indentical? For example `reader` which begins and ends with an `r` or `alfalfa` which begins and ends with `alfa`.

"$\Longleftarrow$" is the easy direction:

$$xy = (uv)(uv)^i u = u(vu)^i(vu)$$
$$yz = u(vu)^i(vu)$$

note that we assumed that $i > 0$, but it is trivial when $i = 0$. So, $xy = yz$. Now the direction "$\Longrightarrow$". We prove it by induction on $|y|$. If $|y| = 1$, then $y = a \in \Sigma$, so $xa = az$, and so x begins with a and z ends with a, and so $x = ax'$ and $z = z'a$. Thus, $ax'a = xa = az = az'a$, and so $x' = z'$. Thus we can take $u = a, v = x' = z', i = 0$.

Suppose now that $|y| > 1$. We break down the reasoning into two cases: suppose first that $|x| \geq |y|$. Then, there exists a $w \in \Sigma^*$ such that $x = yw$ and $z = wy$. So take $u = y, v = w, i = 0$, and we are done. Now suppose $|x| < |y|$, so $y = xw = wz$. We can apply the IH to w, since $0 < |w| < |y|$, so we know that $\exists u \in \Sigma^+, v \in \Sigma^*, i \geq 0$, such that $x = uv, z = vu, w = (uv)^i u = u(vu)^i$. It follows that $y = xw = uv(uv)^i u = (uv)^{i+1} u$.

Problem 2. Let $C_n = \{1^k : \exists \ell, k = n \cdot \ell\}$, that is, C_n consists of all those unary strings whose length is a multiple of n — for a given fixed n. Show that for every such fixed n, C_n is a regular language.

Solution: The point is that for every n, we want to build a (different) DFA A_n such that $L(A_n) = C_n$. Let A_n consist of states $Q = \{q_0, q_1, \ldots, q_{n-1}\}$ and let the transition function δ be defined as follows $\delta(q_i, 1) = q_{(i+1) \pmod n}$, and let $F = \{q_0\}$.

Problem 3. Argue that an algorithm that decides the following language:

$$L = \{0^{2^n} : n \in \mathbb{N}\}$$

must use some working space.

Solution: The point is that if we had an algorithm for L that does not require work-space, then this L would be regular; but it is not, as can be shown easily with the Myhill-Nerode theorem (or, alternatively, with the Pumping Lemma): if i is even and j is odd, then $[0^{2^i - i}] \neq [0^{2^j - j}]$ because $0^{2^i - i} \cdot 0^i \in L$ while $0^{2^j - j} \cdot 0^i \notin L$. Thus, for any fixed even i, we have infinitely many distinct classes corresponding to each odd j.