Spring, 2003 CIS 511

Introduction to the Theory of Computation

Solutions to the Final Exam

April 30

Problem 1. Let Σ be an alphabet and let $L_1, L_2 \subseteq \Sigma^*$ be languages so that L_1 is *not* regular but L_2 *is* regular.

(i) Assume $L_1 \cap L_2$ is finite. Since every finite set is regular, $L_1 \cap L_2$ is regular. Observe that

$$L_1 = ((L_1 \cup L_2) - L_2) \cup (L_1 \cap L_2).$$

If $L_1 \cup L_2$ were regular, since the regular languages are closed under the boolean operations and since L_2 and $L_1 \cap L_2$ are regular, L_1 would be regular, a contradiction. Therefore, $L_1 \cup L_2$ is not regular.

(ii) Now, allow $L_1 \cap L_2$ to be infinite. If $L_1 \subseteq L_2$, then $L_1 \cup L_2 = L_2$ is regular. For example, let $L_1 = \{a^n b^n \mid n \geq 1\}$ and $L_2 = a^* b^*$ (or $L_2 = \{a, b\}^*$).

Problem 2. (i) If the congruence $\sim$ has n equivalence classes, then the congruence $\sim_R$ defined by

$$u \sim_R v \quad \text{iff} \quad u^R \sim v^R$$

also has n equivalence classes, because the equivalence class $[u]_{\sim_R}$ is given by

$$[u]_{\sim_R} = [u^R]_{\sim}.$$

It follows that the congruence $\approx \, = \, \sim \cap \sim_R$ has at most n^2 equivalence classes. This is because the partition associated with $\approx$ consists of all nonempty pairwise intersections of some equivalence class of $\sim$ and some equivalence class of $\sim_R$. Therefore, $\approx$ is also a congruence with a finite number of classes. Obviously, if $u \approx v$, then $u \sim v$ and $u^R \sim v^R$.

(ii) Let L be a regular language (over Σ^*) and let

$$L' = \{w \in \Sigma^* \mid ww^R \in L\}.$$

As L is regular, it is the union of equivalence classes of some congruence with a finite number of equivalence classes, $\sim$. We claim that L' is the union of equivalence classes of $\approx$. Since $\approx$ is a right-invariant equivalence relation (in fact, a congruence) with a finite number of classes, by the usual Myhill-Nerode theorem, L' is regular.

All we need to prove is that for every $u \in L'$, the equivalence class $[u]_{\approx}$ is contained in L'. So, let $u \in L'$ and assume $u \approx v$. We need to prove that $v \in L'$. As $u \in L'$, we have $uu^R \in L$. Now, $u \approx v$ implies that $u \sim v$ and $u^R \sim v^R$. Since $\sim$ is a congruence, we get

$$uu^R \sim vv^R.$$

As $uu^R \in L$, $uu^R \sim vv^R$ and L is the union of equivalence classes of $\sim$, we deduce that $vv^R \in L$. Therefore, $v \in L'$, as claimed.

Problem 3. (i) Give context-free grammars for the languages

$$L_3 = \{a^m b^n c^p \mid m \neq n,\ m, n, p \geq 1\}$$

and

$$L_4 = \{a^m b^n c^p \mid n \neq p,\ m, n, p \geq 1\}.$$

Let G_1 be the grammar whose productions are

$$\begin{array}{rcl} S & \longrightarrow & AXC \mid XBC \\ X & \longrightarrow & aXb \mid ab \\ A & \longrightarrow & aA \mid a \\ B & \longrightarrow & bB \mid b \\ C & \longrightarrow & cC \mid c \end{array}$$

and let G_2 be the grammar whose productions are

$$\begin{array}{rcl} S & \longrightarrow & ABY \mid AYC \\ Y & \longrightarrow & bYc \mid bc \\ A & \longrightarrow & aA \mid a \\ B & \longrightarrow & bB \mid b \\ C & \longrightarrow & cC \mid c. \end{array}$$

It is easy to check (by induction on the length of derivations) that $L(G_1) = L_3$ and $L(G_2) = L_4$.

(ii) The grammar obtained by taking the union of the rules of G_1 and G_2 yields a grammar for $L_5 = L_3 \cup L_4$. Consider $L_6 = \{a, b, c\}^* - L_5$. If L_6 were context-free, as the context-free languages are closed under intersection with the regular languages, then $L_6 \cap a^*b^*c^*$ would be context-free. But,

$$L_6 \cap a^*b^*c^* = \overline{L_3} \cap \overline{L_4} \cap a^*b^*c^* = \{a^n b^n c^n \mid n \geq 0\},$$

which is well-known *not* to be context-free. Therefore, L_6 is not context-free.

Problem 4. Let $f\colon \mathbb{N} \to \mathbb{N}$ be any function such that for some (large) $n_0 \in \mathbb{N}$, we have $f(n+1) - f(n) \geq n + 1$, for all $n \geq n_0$.

(i) First, we note that the condition on f implies that $L_7 = \{a^{f(n)} \mid n \geq 0\}$ is infinite. Also, for all $n \geq n_0 + 1$, we have $f(n) \geq n$. We use the pumping lemma (for the context-free languages) to derive a contradiction. If $K \geq 2$ is the constant of the pumping lemma, let $N = \max\{K, n_0 + 1\}$, and pick $w = a^{f(N)} \in L_7$. Since $N \geq n_0 + 1$, we have $|w| = f(N) \geq N \geq K$ and the pumping lemma applies. So, we can write $w = uvxyz$, with $u = a^i$, $v = a^j$, $x = a^k$, $y = a^l$ and $z = a^{f(N)-(i+j+k+l)}$, and where $j + l \geq 1$ (since $vy \neq \epsilon$) and $j + k + l \leq K$ (since $|vxy| \leq K$). We also have

$$uv^m xy^m z \in L_7 \quad \text{for all } m \geq 0.$$

This means that

$$a^{f(N)-(i+j+k+l)+i+k+m(j+l)} = a^{f(N)+(m-1)(j+l)} \in L_7 \quad \text{for all } m \geq 0.$$

Let $m = 2$. Then, we have $a^{f(N)+j+l} \in L_7$. However, $1 \leq j + l \leq K \leq N$, so $f(N) < f(N) + j + l \leq f(N) + N$, and yet for all $n \geq N \geq n_0 + 1$, we have $f(n + 1) \geq f(n) + n + 1$, a contradiction. Therefore, L_7 is not regular.

(ii) The languages

$$\begin{aligned} L_8 &= \{a^{n(n+1)/2} \mid n \geq 0\} \\ L_9 &= \{a^{n!} \mid n \geq 0\} \end{aligned}$$

are not context-free. Indeed,

$$(n + 1)(n + 2)/2 - n(n + 1)/2 = n + 1$$

and

$$(n + 1)! - n! = (n + 1)n! - n! = nn! \geq n + 1,$$

for all $n \geq 2$, and we apply part (i).

Problem 5. (i) Since every r.e. set is the domain of some partial recursive function, we have $A = \mathrm{dom}(\varphi_i)$ and $B = \mathrm{dom}(\varphi_j)$, for some indices i and j (where, as in Problem 6, we let $\varphi_0, \varphi_1, \varphi_2, \ldots$ be any fixed acceptable indexing of the partial recursive functions). Let f and g be the partial recursive functions defined as follows:

$$\begin{aligned} f(x) &= \min z[T(i, x, \Pi_1(z)) \wedge T(j, x, \Pi_2(z))] \\ g(x) &= \min z[T(i, x, \Pi_1(z)) \vee T(j, x, \Pi_2(z))], \end{aligned}$$

where T is Kleene's T-predicate. Recall that $T(i, x, z)$ is true iff there is a halting computation coded by z of the RAM program coded by i on input y. The RAM program coded by x computes the partial function φ_x. Thus, $\min z\, T(i, x, z)$ is defined iff $\varphi_i(x)$ is defined. Note that g can also be defined by

$$g(x) = \min z[T(i, x, z) \vee T(j, x, z)],$$

i.e., the projections Π_1 and Π_2 are not needed. However, they are for f; otherwise, we would be requiring that the RAM programs computing $\varphi_i(x)$ and $\varphi_j(x)$ have the same computation (z), and that's too strong. Clearly, $x \in \text{dom}(f)$ iff $x \in \text{dom}(\varphi_i) \cap \text{dom}(\varphi_j)$ and $x \in \text{dom}(g)$ iff $x \in \text{dom}(\varphi_i) \cup \text{dom}(\varphi_j)$. Thus, $A \cap B = \text{dom}(f)$ and $A \cup B = \text{dom}(g)$, which proves that $A \cap B$ and $A \cup B$ are r.e.

(ii) Consider $K = \{x \in \mathbb{N} \mid \varphi_x(x) \quad \text{is defined}\}$. We know that K is nonrecursive, yet r.e. So, its complement, $\overline{K}$, is not r.e., since otherwise K would be recursive (Recall: if a set A and its complement $\overline{A}$ are both r.e., then A is recursive).

Problem 6. Let $\varphi_0, \varphi_1, \varphi_2, \ldots$ be any fixed acceptable indexing of the partial recursive functions.

(i) Let $a \in \mathbb{N}$ be any fixed natural number and consider the set

$$S_a = \{i \in \mathbb{N} \mid \varphi_i(i) = a\}.$$

We use Rice's theorem to prove that S_a is not recursive. For this, we just have to prove that S_a is neither empty nor $\mathbb{N}$. Since the constant function with value a is a recursive function, this function occurs as some φ_i, and $\varphi_i(i) = a$. Thus, $S_a \neq \emptyset$. On the other hand, the partial function undefined everywhere also occurs as some φ_i such that $i \notin S_a$; thus, $S_a \neq \mathbb{N}$. Since S_a is nontrivial, by Rice's theorem, it is not recursive.

Since the equality predicate is primitive recursive, the set S_a is r.e. because it is the domain of the partial recursive function

$$f(x) = \begin{cases} 1 & \text{if } \varphi_x(x) = a \\ \text{undefined} & \text{otherwise.} \end{cases}$$

(ii) Let $A = S_0$ and $B = S_1$. These two sets are clearly disjoint, r.e., and nonrecursive, by part (i). Assume that there is some recursive set, R, so that $S_0 \subseteq R$ and $S_1 \cap R = \emptyset$. Since R is recursive, there is a (total) recursive function, φ_i, so that $\varphi_i(x) = 1$ iff $x \in R$ and $\varphi_i(x) = 0$ iff $x \notin R$. Look at $\varphi_i(i)$. Either $\varphi_i(i) = 1$ or $\varphi_i(i) = 0$.

If $\varphi_i(i) = 1$, then $i \in R$ and also $i \in S_1$. So, $i \in S_1 \cap R$, contradicting the fact that $S_1 \cap R = \emptyset$.

If $\varphi_i(i) = 0$, then $i \notin R$ and also $i \in S_0$. But, this contradicts $S_0 \subseteq R$. Therefore, S_0 and S_1 are recursively inseparable.