CS 682 (Spring 2001) - Solutions to Assignment 4

For $A, B \subseteq \Sigma^*$, let

$$A/B = \{x|(\exists y)(xy \in A \text{ and } y \in B)\}.$$

(1) Prove that for any M_i there exist two context-free grammars, G_j and G_k, such that

$$L(G_j)/L(G_k) = L(M_i).$$

Proof. Fix M_i. We know that $L_1 = \{y\#z^R|y \rightarrow_{M_i} z\}$ and $L_2 = \{y^R\#z|y \rightarrow_{M_i} z\}$ are context free languages (see proof of lemma 8.6, Hopcroft and Ullman). Clearly $L_0 = \{x\#\#x^R q_0|x \in \Sigma^*\}$ is also a CFL (Σ here denotes the alphabet used by M_i, and does not contain the symbols for the states of M_i, nor the symbol $\#$).

Let $L_3 = L_0\#(L_1\#)^*(\{\epsilon\}\cup\Sigma^*F\Sigma^*\#)$ and $L_4 = \#\#(L_2\#)^*(\{\epsilon\}\cup\Sigma^*F\Sigma^*\#)$. Then both are context free languages, since they are built from CFLs by union, concatenation and Kleene closure. Suppose $xy \in L_3$ and $y \in L_4$. Then the $\#\#$ prefix of y must match the $\#\#$ in the L_0 part of xy. Further matchings of $\#$ show that xy has to be of the form

$$x\#\#y_0^R\#y_1\#y_2^R\#\ldots\#y_n^{(R)}\#$$

where $y_0 = q_0x$ and $y_j \rightarrow_{M_i} y_{j+1}$ for all j, and y_n is an accepting configuration of M_i. Therefore $x \in L(M_i)$.

Conversely, for any $x \in L(M_i)$ let $y = \#\#y_0^R\#\ldots\#y_n^{(R)}\#$, and then $xy \in L_3$ and $y \in L_4$.

Therefore $L(M_i) = L_3/L_4 = L(G_j)/L(G_k)$, where G_j, G_k are the CFGs which produce L_3, L_4 respectively.

Note that the above can be formalized into a recursive construction of G_j, G_k, uniform in M_i. ∎

(2) Let $I_1, I_2, \ldots$ be a recursive list of FA with a two-way head that can scan the input symbols and either erase or leave them unaltered.

Prove that

$$\{I_j|L(I_j) \text{ is finite }\} \text{ is } \Sigma_2\text{-complete.}$$

Proof. Let $\mathcal{I}_{fin} := \{I_i \mid L(I_i) \text{ is finite}\}$. To see that $\mathcal{I}_{fin}$ is in Σ_2, notice that

$$I_i \in \mathcal{I}_{fin} \qquad \Longleftrightarrow \qquad (\exists x)(\forall y \geq x) y \notin L(I_i).$$

The matrix is decidable, since on any given input y, I_i can only go through a bounded number of states (after all, it's even weaker than an LBA).

To show Σ_2-hardness, we will reduce FIN=$\{M_i|L(M_i) \text{ is finite}\}$ to $\mathcal{I}_{fin}$. To do that, we will use a modified version of VALCOMs. The format is the same as the one described in class (or that you might have defined in your homework), except each symbol is written twice (i.e. `abcb` becomes `aabbccbb`). We claim that these modified VALCOMs can be decided by a no-ink machine.

The no-ink machine first erases one copy of each symbol of the first and last instantaneous description (ID), say the first copy for the first ID, and the second copy for the last ID. It then compares, symbol by symbol, every pair of adjacent IDs, from left to right. We always compare the "even" copies of the left ID with the "odd" copies of the right ID, erasing them when they are equal. Finding the currently compared symbols is easy (as it is the leftmost non-blank even/odd symbol in the current ID), and we can of course find out which ID we are in by looking at the separating #.

Now let f be recursive such that $I_{f(i)}$ accepts the (modified) VALCOMs of M_i. Then

$$M_i \in \text{FIN} \qquad \Longleftrightarrow \qquad I_{f(i)} \in \mathcal{I}_{fin}.$$

and therefore $\mathcal{I}_{fin}$ is Σ_2-complete. ■

(3) (a) Prove that there exists a LBA language $L(\mathcal{L}_i)$ that is immune to all regular sets, i.e., $L(\mathcal{L}_i)$ is infinite, and $L(A_j) \subseteq L(\mathcal{L}_i)$ implies that $L(A_j)$ is finite.

(b) Is there a recursive set immune to all LBA languages? Prove the correctness of your answer.

Proof.

(a) Let $L = \{a^n b^n | n \in \omega\}$. We saw that an LBA can count its own steps (up to the number of different configurations it can have), so it certainly can count the number of a's and b's, and compare the two numbers. Therefore L is an LBA language.

On the other hand, L is infinite, and we know that any infinite subset of L is not regular (easily shown with the pumping lemma for regular sets), so L is immune to regular sets.

(b) We demonstrate that there is such a set R by constructing a TM M that enumerates R in increasing order (thus ensuring R is recursive). M works in stages, keeping track of a finite list S, as follows:

At first $S = \emptyset$. At the beginning of even stages $n = 2m$, M adds A_m to S. At any stage n, M checks for each $A_i \in S$ if it accepts n. If such A_i are found, they are removed from S, and the stage ends. Let such a stage be denoted as Type A. If no $A_i \in S$ accepts n, M prints n (in the enumeration), and the stage ends. This is a Type B stage.

First, notice that there are infinitely many stages of Type B. To show this, it suffices to show that for every even n there is some stage $m > n$ of Type B. Let k be the size of S at the end of stage n. If all stages after n are Type A stages, then by stage $n+2k+2$ M will have added to S $k+1$ elements (one for each even stage), and removed at least $2k+2$ elements (at least one for each stage), which is clearly impossible. This proves the claim, which immediately implies that $L(M)$ is infinite, since at every stage n of Type B, a new number n is enumerated.

The construction also guarantees that no infinite LBL is contained in $L(M)$: suppose $L(A_i)$ is infinite. A_i is added to S at stage $2i$. Let k be the least element of $L(A_i)$ which is not less than $2i$. Then at stage k M removes A_i from S and does not enumerate k. Since k can only be enumerated at stage k, it is not enumerated at all and we have

$$k \in L(A_i) \setminus L(M) \quad \Longrightarrow \quad L(A_i) \not\subseteq L(M)$$

Therefore $L(M)$ is a recursive set, immune to all LBLs.

■