CS 682 (Spring 2001) - Solutions to Assignment 3

Let H_i denote a two-head read-only finite automaton whose heads move independently from left to right.

(1) Prove that

$$\{H_i | L(H_i) \text{ is cofinite } \}$$

is a Σ_2-complete set.

Proof. Note:

(a) A given H_i, when running on a given input, has finitely many configurations, and therefore its halting can be determined recursively. As a result, $L(H_i)$ is recursive.

(b) The H_i's can check for valid computations of Turing Machines: The two heads can scan each pair of successive configurations, going over both configurations simultaneously, and checking whether the transition between them is legal according to the TM's finite transition function which can be programmed into the automaton.

First,

$$L(H_i) \text{ is cofinite} \iff (\exists x)(\forall y > x) y \in L(H_i)$$

which is Σ_2 by (a).

Now, let f be recursive such that $L(H_{f(i)}) = \overline{\text{VALCOM}(M_i)}$. This can be done by (b). This gives

$$L(M_i) \text{ is finite} \iff L(H_{f(i)}) \text{ is cofinite}$$

which shows that f is a many-one reduction of a Σ_2-complete set to $\{H_i | L(H_i) \text{ is cofinite } \}$.

Therefore $\{H_i | L(H_i) \text{ is cofinite } \}$ is Σ_2-complete. ∎

(2) Place the following sets in the lowest possible class of the Kleene Hierarchy. Give a short justification for your answer, no formal proof of the hardness is required. As usual G_i, L_i and A_i denote context-free grammars, linearly-bounded automata and finite automata, respectively.

Proof.

(a1) $A = \{H_i | L(H_i) \text{ is infinite } \}$

First,

$$H_i \in A \iff (\forall x)(\exists y > x) y \in L(H_i)$$

and therefore A is Π_2 (see note (a), problem 1). To show hardness, let f be recursive such that $L(H_{f(i)}) = \text{VALCOM}(M_i)$ (see note (b), problem 1). Then

$$M_i \in \text{INF} = \{M_i | L(M_i) \text{ is infinite } \} \iff H_{f(i)} \in A$$

and hardness follows from Π_2-completeness of INF.

(a2) $A = \{G_i | L(G_i) \text{ is infinite }\}$

This is recursive (see Theorem 6.6(c) in Hopcroft and Ullman).

(b) $A = \{(M_i, H_j) | L(M_i) \cap L(H_j) \text{ is regular }\}$

First,

$$(M_i, H_j) \in A \qquad \Longleftrightarrow \qquad (\exists k)(\forall x)(x \in L(A_k) \leftrightarrow x \in L(M_i) \wedge x \in L(H_j)).$$

The matrix of this definition is a conjunction of a Σ_1 relation ($\rightarrow$) and a Π_1 relation ($\leftarrow$), which is also a conjunction of two Π_2 relations, which is Π_2. With the two quantifiers, it is a Σ_3 relation. To show hardness, let H_{Σ^*} be a two-head read-only finite automaton which accepts all strings (one clearly exists), and let $f(M_i) = (M_i, H_{\Sigma^*})$. f is clearly recursive, and

$$M_i \in \text{REG} = \{M_i | L(M_i) \text{ is regular }\} \qquad \Longleftrightarrow \qquad L(M_i) \cap L(H_{\Sigma^*}) \text{ is regular}$$
$$\Longleftrightarrow \qquad f(M_i) \in A.$$

Hardness then follows from Σ_3-completeness of REG.

(c) $A = \{(A_i, G_j) | L(A_i) \subseteq L(G_j)\}$

First,

$$(A_i, G_j) \in A \qquad \Longleftrightarrow \qquad (\forall x)(x \in L(A_i) \rightarrow x \in L(G_j))$$

and therefore A is Π_1 (regular sets and context free languages are all recursive). To show hardness, let A_{Σ^*} be a DFA which accepts all strings, and let $f(G_i) = (A_{\Sigma^*}, G_i)$. f is clearly recursive, and

$$G_i \in B = \{G_i | L(G_i) = \Sigma^*\} \qquad \Longleftrightarrow \qquad L(A_{\Sigma^*}) \subseteq L(G_i) \qquad \Longleftrightarrow \qquad f(G_i) \in A.$$

Hardness then follows from Π_1-completeness of B.

(d) $A = \{(A_i, G_j) | L(G_j) \subseteq L(A_i)\}$

Note:

(i) Complements of regular sets are regular, in a uniform way: there is a recursive f_1 such that $L(A_{f_1(i)}) = \overline{L(A_i)}$. This follows from the existence of an algorithm that, given a DFA A_i, constructs a DFA which accepts $\overline{L(A_i)}$ (see proof of Theorem 3.2 in Hopcroft and Ullman).

(ii) Intersections of regular sets and context free languages are context free, again in a uniform way: there is a recursive f_2 such that $L(G_{f_2(i,j)}) = L(A_i) \cap L(G_j)$. This follows from the existence of an algorithm that, given a DFA A_i and a PDA G_j, constructs a PDA which accepts $L(A_i) \cap L(G_j)$ (see proof of Theorem 6.5 in Hopcroft and Ullman).

We will show that A is recursive, by reducing it to a recursive set: Let $g(A_i, G_j) = G_{f_2(f_1(i),j)}$. Then g is clearly recursive, and

$$(A_i, G_j) \in A \qquad \Longleftrightarrow \qquad \overline{L(A_i)} \cap L(G_j) = \emptyset \qquad \Longleftrightarrow \qquad g(A_i, G_j) \in C = \{G_i | L(G_i) = \emptyset\}.$$

Note that C is recursive (see Theorem 6.6(a) in Hopcroft and Ullman).

(e) $A = \{M_i | L(M_i) \text{ is an infinite context-free language }\}$

First,

$$M_i \in A \qquad \Longleftrightarrow \qquad (\exists j)\left[(L(G_j) \text{ is infinite}) \wedge (\forall x)(x \in L(M_i) \leftrightarrow x \in L(G_j))\right].$$

Since "$L(G_j)$ is infinite" is a recursive relation, the conjunction is Π_2 (similar to (b)), and the whole relation is Σ_3. To show hardness, let f be recursive such that $L(M_{f(i)}) = L(M_i) \cup \{\sharp^n | n \in \omega\}$, where $\sharp$ is a new symbol. Notice that $L(M_{f(i)})$ is always infinite, and is context-free iff $L(M_i)$ is. Then

$$M_i \in \text{CFL} = \{M_i | L(M_i) \text{ is context-free}\} \qquad \Longleftrightarrow \qquad M_{f(i)} \in A$$

and hardness follows from Σ_3-completeness of CFL.

(f) $A = \{M_i | L(M_i)$ accepts an infinite number of TM-names of TMs that accept cofinite sets $\}$

First,

$$M_i \in A \qquad \Longleftrightarrow \qquad (\forall j)(\exists k > j)\,[M_k \in L(M_i) \wedge (\exists x)(\forall y > x) y \in L(M_k)]$$

The conjunction is of a Σ_1 relation and a Σ_3 relation, and is Σ_3. The whole relation is therefore Π_4. To show hardness, suppose P is a Π_4 set, that is

$$x \in P \qquad \Longleftrightarrow \qquad (\forall y) R(x, y)$$

where R is a Σ_3 relation. Since COF$= \{M_i | L(M_i)$ is cofinite $\}$ is Σ_3-complete, there is a recursive f that reduces R to COF, that is

$$(x, y) \in R \qquad \Longleftrightarrow \qquad M_{f(x,y)} \in \text{COF}.$$

Now, let g be recursive such that $L(M_{g(x)}) = \{M^1, M^2, M^3, \dots\}$ where

$$L(M^y) = L(M_{f(x,1)}) \cap L(M_{f(x,2)}) \cap \cdots \cap L(M_{f(x,y)}).$$

Think of $M_{g(x)}$ as an enumerator, which at stage y computes $f(x,1), \dots, f(x,y)$ and prints a TM-name of a TM which accepts the intersection of the languages of the $M_{f(x,i)}$s (i.e. a TM which simulates all y machines, and accepts iff all of them accept). By construction, $M^1, M^2, \dots$ are all different (and if not, $M_{g(x)}$ can always make them so by adding dummy states).

Now, if $x \in P$ then $(\forall y) M_{f(x,y)} \in$COF, so the M^is will all have cofinite languages (the intersection of a finite number of cofinite sets is cofinite), and then $M_{g(x)} \in A$.

Conversely, if $x \notin P$, then $M_{f(x,y_0)} \notin$COF for some y_0. Then $L(M_{f(x,y_0)})$ will "ruin" the cofiniteness of $L(M^y)$ for all $y \geq y_0$ (any subset of a co-infinite set is co-infinite), and only finitely many of the M^is will have confinite languages, implying $M_{g(x)} \notin A$.

Therefore g is a many-one reduction of P to A. Since P was an arbitrary Π_4 set, A is Π_4-hard.

■

(3) Let P be a non-trivial property on the r.e. sets that is false for context-free languages. Prove that

$$\{M_i | P(M_i) = \text{ True }\}$$

is Π_2-hard.

Proof. Let $P = \{M_i | P(M_i) = \text{ True }\}$ (standard notation). Since P is non-trivial, there is an r.e. set $A \in P$. We will use A to construct a many-one reduction of $\Delta = \{M_i | L(M_i)$ is infinite$\}$ to P. Since Δ is Π_2-hard, this will prove that P is.

Let f be recursive such that $M_{f(i)}$ accepts an input x iff $x \in A$ and there exists some $y > x$ in $L(M_i)$. To accomplish this $M_{f(i)}$ enumerates A and $L(M_i)$, both r.e., and accepts x once x has appeared in the first enumeration and some $y > x$ has appeared in the second enumeration.

Now, if $L(M_i)$ is infinite, $M_{f(i)}$ will accept x iff $x \in A$, since the second condition will always hold, so $L(M_{f(i)}) = A$. Therefore

$$M_i \in \Delta \quad \Longrightarrow \quad M_{f(i)} \in P.$$

On the other hand, if $L(M_i)$ is finite, so is $L(M_{f(i)})$, since no $x > \max(L(M_i))$ will be accepted. Therefore

$$M_i \notin \Delta \quad \Longrightarrow \quad M_{f(i)} \notin P$$

since all finite sets are context-free. ■