Solutions to Quiz 2
Nov 14, 2006

Prof. Amit Chakrabarti
Computer Science Department
Dartmouth College

1. Let $\Sigma = \{0, 1\}$. Consider the following classes of languages over the alphabet Σ:

$$\begin{aligned}
A &= \{L \subseteq \Sigma^* : L \text{ is decidable}\} \\
B &= \{L \subseteq \Sigma^* : L \text{ can be recognized by an NFA}\} \\
C &= \{L \subseteq \Sigma^* : L \text{ can be generated by a regular expression}\} \\
D &= \{L \subseteq \Sigma^* : L \text{ can be generated by a CFG in which every rule is of the form } A \to BC \\
&\qquad \text{or else } A \to a \text{ or else } S \to \varepsilon, \text{ where } A, B, C \text{ are variables}, S \text{ is the start variable and} \\
&\qquad a \text{ is a terminal of the CFG}\} \\
E &= \{L \subseteq \Sigma^* : L \text{ can be generated by a CFG in which every rule is of the form } A \to aB \\
&\qquad \text{or else } A \to \varepsilon, \text{ where } A, B \text{ are variables and } a \text{ is a terminal of the CFG}\} \\
F &= \{L \subseteq \Sigma^* : L \text{ can be recognized by a PDA}\} \\
G &= \{L \subseteq \Sigma^* : \text{both } L \text{ and } \Sigma^* - L \text{ can be recognized by PDAs}\}
\end{aligned}$$

1.1. Describe, in a one-line chain, all the proper subset and equality relationships between the classes A, B, C, D, E, F and G. For example, you might write something like $A \subset F \subset B = G \subset C = D = E$. Here "$\subset$" means "is a *proper* subset of."

[5 points]

Solution: $B = C = E \subset G \subset D = F \subset A$.
$B = C$ was proved in class; $C = E$ follows from HW5, #4.3; $E \subseteq G$ follows because regular languages are closed under complement and regular languages are context-free; $G \subseteq F$ is trivial; $D = F$ follows from our theorem about Chomsky normal form CFGs; $F \subseteq A$ follows because an NDTM can simulate a PDA.

1.2. For every proper subset relationship you indicated above, give an example of a language that proves the inequality of the corresponding classes. For example, if you wrote "$B \subset C$" then give an example of a language in C but not in B, and so on. Keep in mind that $\Sigma = \{0, 1\}$.

[10 points]

$E \subset G$: $\{0^n 1^n : n \geq 0\}$ is in G but not in E.
$G \subset D$: $\Sigma^* - \{ww : w \in \Sigma^*\}$ is in D but not in G.
$F \subset A$: $\{ww : w \in \Sigma^*\}$ is in A but not in F.

2. In class, we gave a procedure for simulating the computation of a multitape Turing Machine on a usual (single-tape) Turing Machine. Describe, briefly, how we represented the multiple tapes of the former on the single tape of the latter. Make sure you specify the connection between the tape alphabets of the two machines. (You do not have to explain anything else about the simulation.)

[10 points]

Solution: Let M be a k-tape TM with tape alphabet Γ. We simulate it on a single-tape TM M' whose tape alphabet consists of all symbols in Γ, "marked" versions of all symbols in Γ, and a delimiter symbol $\# \notin \Gamma$. If, at any point of time, the k tapes of M' hold the strings $w_1, \ldots, w_k$, each followed by infinitely many blanks, then the tape of M' will hold

$$\#w_1' \sqcup \#w_2' \sqcup \cdots \#w_k' \sqcup \# \quad ,$$

where w_i' is w_i with the symbol under the i^{th} head marked.

3. Suppose A is a Turing-recognizable language over the alphabet Σ and B is a decidable language over Σ. Give an informal high-level description of a Turing machine that recognizes $A - B$. You do not have to prove that your construction is correct.

[10 points]

Solution: Let M_A be a recognizer TM for A and M_B a decider TM for B. We construct a recognizer 2-tape TM M for $A - B$ as follows:

$M =$ "On input x:
1. Copy x onto tape 2 and reset both heads to their leftmost positions.
2. Run M_B using tape 2; if it accepts, then REJECT.
3. Run M_A using tape 1; if it accepts, then ACCEPT.
4. If we reach here, then REJECT."

M accepts a string iff M_B rejects it and then M_A accepts it, as required.

An interesting observation is that it is perfectly okay to run M_A first!

4. Let (V_1, Σ, R_1, S_1) be a CFG for a language $L_1 \subseteq \Sigma^*$ and let (V_2, Σ, R_2, S_2) be a CFG for $L_2 \subseteq \Sigma^*$. Assume that V_1 and V_2 are disjoint. Specify, *formally*, a CFG for the language $L_1^* \cup L_2$. No proof of correctness required.

[10 points]

Solution: Let $G = (V, \Sigma, R, S)$, where

$$V \;=\; V_1 \cup V_2 \cup \{S, T\}, \quad \text{where } S, T \notin V_1 \cup V_2\,,$$

and R is obtained by starting with $R_1 \cup R_2$ and adding the following new rules:

$$\begin{array}{rcl} S & \longrightarrow & T \mid S_2 \\ T & \longrightarrow & S_1 T \mid \varepsilon\,. \end{array}$$

5. For a language L over alphabet Σ, define

$$\textsc{Half}(L) \quad = \quad \{x \in \Sigma^* : \exists\, y \in \Sigma^* \,(|x| = |y| \text{ and } xy \in L)\}\,.$$

Throughout this section, let $A = \{a^m b^m c^n \#\# d^{3n} \,:\, m, n \geq 0\}$; thus, A is a language over the alphabet $\{a, b, c, d, \#\}$. Also, throughout this section, you may use without proof any facts proved in class, provided you clearly state what fact(s) you are using.

5.1. Specify a CFG for A. No explanation is necessary.

[9 points]

Solution: The following CFG generates A:

$$\begin{array}{rcl} S & \longrightarrow & TU \\ T & \longrightarrow & aTb \mid \varepsilon \\ U & \longrightarrow & cUddd \mid \#\# \end{array}$$

5.2. Specify the following language as simply as possible in set notation:

$$\textsc{Half}(A) \cap a^*b^*c^*\# \,.$$

No explanation is necessary. For an example of how a language is specified in set notation, see the definition of A above.

[9 points]

Solution: $\{a^n b^n c^n \# : \, n \geq 0\}$.

5.3. Is the language $\textsc{Half}(A)$ context-free? Prove your answer.

[10 points]

Solution: No, $\textsc{Half}(A)$ is not context-free. Here's a proof.

Suppose $\textsc{Half}(A)$ is context-free. Then $\textsc{Half}(A) \cap a^*b^*c^*\#$ must also be context-free because, as proved in class, the intersection of a CFL and a regular language is a CFL. By the result of #5.2, this means the language $B = \{a^n b^n c^n \# \, : \, n \geq 0\}$ must be context-free.

At this point you could use the pumping lemma to get a contradiction, but there is an even simpler solution! Suppose $G = (V, \{a, b, c, \#\}, R, S)$ is a CFG that generates B. Then, the CFG $G' = (V \cup \{\#\}, \{a, b, c\}, R', S)$, where $R' = R \cup \{`\# \to \varepsilon'\}$, clearly generates $\{a^n b^n c^n \, : \, n \geq 0\}$. However, as proved in class, this latter language is not context-free, so we have a contradiction.

5.4. What can you conclude about the closure of the class of context-free languages under the operation $\textsc{Half}$?

[2 points]

Solution: From #5.1 and #5.3, we infer that context-free languages are *not* closed under the operation $\textsc{Half}$.