Solutions to Quiz 2
Nov 20, 2007

Prof. Amit Chakrabarti
Computer Science Department
Dartmouth College

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

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

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

[5 points]

Solution: $A = B = F \subset E \subset D = G \subset C$.
$A = B$ was proved in class; $B = F$ follows from HW5, #4.2; $F \subseteq E$ follows because regular languages are closed under complement and regular languages are context-free; $E \subseteq D$ is trivial; $D = G$ follows from our theorem about Chomsky normal form CFGs; $G \subseteq C$ 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]

Solution: Here are some examples that work:
$F \subset E :$ $\{0^n 1^n : n \geq 0\}$ is in E but not in F.
$E \subset D :$ $\Sigma^* - \{ww : w \in \Sigma^*\}$ is in D but not in E.
$G \subset C :$ $\{ww : w \in \Sigma^*\}$ is in C but not in G.

2. Recall that a *configuration* of a Turing Machine $(Q, \Sigma, \Gamma, \delta, q_0, q_{\text{acc}}, q_{\text{rej}})$ is a string in $\Gamma^* Q \Gamma^*$ (or equivalently, a string in $(Q \cup \Gamma)^*$ that contains exactly one symbol from Q). The string encodes certain pieces of information about the TM in the middle of a run.

2.1. Suppose $\Sigma = \{0, 1\}$ and $q_5 \in Q$. Exactly what information is encoded by the configuration "$1011q_5010$"?

[5 points]

Solution: The given configuration tells us that

- the TM is in state q_5,
- the tape contents are "1011010" followed by an infinite number of blanks, and
- the head is currently over the second '0'.

2.2. Give a complete formal definition of what it means for a configuration C_1 of a TM to *yield* a configuration C_2. Make sure you cover all cases.

[10 points]

Solution: We say that C_1 yields C_2 if at least one of the following conditions holds:

- $C_1 = uaqbv$, $C_2 = uacrv$ and $\delta(q, b) = (r, c, R)$
- $C_1 = uaqbv$, $C_2 = uracv$ and $\delta(q, b) = (r, c, L)$
- $C_1 = qbv$, $C_2 = crv$ and $\delta(q, b) = (r, c, R)$
- $C_1 = qbv$, $C_2 = rcv$ and $\delta(q, b) = (r, c, L)$

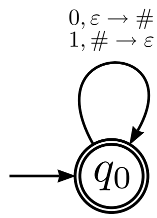
for some $a, b \in \Gamma$, $u, v \in \Gamma^*$ and $q, r \in Q$.

3. For a string $x \in \{0,1\}^*$, let $N_0(x)$ and $N_1(x)$ denote the number of 0s and 1s in x, respectively. Draw a PDA for the following language (keep it simple!):

$$\{x \in \{0,1\}^* : \text{ every prefix } w \text{ of } x \text{ satisfies } N_0(w) \geq N_1(w)\}\,.$$

[10 points]

Solution:



4. 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!

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 $A = \{a^m b^m c^n \#\# d^{3n} \,:\, m, n \ge 0\}$; 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.

[5 points]

Solution: The following CFG generates A:

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

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.

[5 points]

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

5.3. Your answer above should look very similar to a language we have studied in class. Examine it carefully. Based on it, what can you conclude about the closure of the class of context-free languages under the operation $\textsc{Half}$? Prove your answer.

[10 points]

Solution: Context-free languages are *not* closed under $\textsc{Half}$. The language A provides a counterexample. We gave a CFG for it in #5.1, so A is context-free. However, $\textsc{Half}(A)$ 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 $B = \{a^n b^n c^n \# \,:\, n \ge 0\}$ 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 $R' = R \cup \{\text{‘}\# \to \varepsilon\text{’}\}$, clearly generates $\{a^n b^n c^n \,:\, n \ge 0\}$. However, as proved in class, this latter language is not context-free, so we have a contradiction.