CS 501: Formal Language Theory **Spring 2021**

Midterm Solutions

Note: LaTeX template courtesy of UC Berkeley EECS dept.

1. **(10 × 2 points) Unjustified True/False Questions.**
   (a) FALSE. L is not regular. The infinite set $\{\varepsilon, 0, 0^2, \ldots\}$ is a set of pairwise L-distinguishable strings.
   (b) FALSE. Take $L = (abc)^*$, but then L^{++} is the set of all strings over $\{a, b, c\}$ with equal numbers of a's, b's, and c's, which is irregular.
   (c) TRUE. Otherwise, if p is the pumping length, $a^p b^{2p} c^p$ cannot be pumped.
   (d) TRUE. Simulate M on all possible (finitely many) inputs of length at most k.
   (e) TRUE. Convert N to DFA by the subset construction, use graph search to reach any state containing two or more accept states of the original NFA.
   (f) TRUE. Modify the PDA to keep track of the maximum stack height so far.
   (g) TRUE. Graph search to find the set F of accept states reachable from the starting state, and then check reachability between every pair in F.
   (h) FALSE. Rice's Theorem.
   (i) FALSE. The TD languages $\varnothing$ and Σ^* are exceptions.
   (j) TRUE. Output YES iff there exist i, j with $|a_i| \leq |b_i|$ and $|a_j| \geq |b_j|$.

2. **(5 × 6 points) Justified True/False Questions.** For each of the following questions, indicate whether it is TRUE or FALSE, *and provide a brief justification (i.e. either a proof or a counterexample).*
   (a) TRUE. Replace all transitions in an L-DFA by $0, 1$ labels. This creates an NFA for L'.
   (b) FALSE. Take a 2020-state DFA for L and add a dummy state.
   (c) TRUE. We can generate it by the grammar $S \to TST|B$, $B \to TTB|T1$, $T \to 0|1$.
   (d) TRUE. Simulate an doubly-infinite TM by a two-tape ordinary TM, containing the contents of the left half of the tape reading backwards, and the contents of the right half reading forwards. The converse is trivial.
   (e) TRUE. Given a PDA for X, construct a PDA for $f^{-1}(X)$ by storing each state as an ordered pair of an X-state, together with a suffix string of $f(a)$ for any $a \in \Sigma$ (there are finitely many of them). On reading an input letter a, we transition through a sequence of states for $f(a)$ before moving onto the next letter.

3. **(5 points) Hanging TMs.** We reduce from A_{TM}. Given an instance $\langle M, w \rangle$ of A_{TM}, we construct a HTM N as follows. On input x, it ignores its input and writes w on its tape, starting from the second cell of the tape, after marking the leftmost cell with a special symbol. It then simulates the computation of M on w, ensuring every time it gets to the marked cell, it goes back ("resets") one step to the second cell. If M accepts, N also does. Then, note that $\langle M, w \rangle \in A_{TM}$ if and only if $\langle N, w \rangle \in A_{HTM}$, proving the reduction. Of course, the w in the RHS could be replaced by any string.

4. **(10 points) Replicating Nonterminals.**

   (a) Given a CFG, we can draw a graph whose vertices are the nonterminals, with an arc between two vertices if there is a grammar rule where the corresponding "tail" terminal derives a string containing the "head" terminal. Then, a specific non-terminal is replicating if and only if the corresponding vertex is part of a directed cycle. This is a well-known graph theory problem and has plenty of polynomial-time algorithms for it.

   (b) The CFLPL proof shows that if a string is sufficiently long (longer than its pumping length p), then its parse tree has a replicating terminal. It follows that if there are no replicating non-terminals, all strings in the language must be bounded above by a constant, and so in fact the language is then finite.

5. **(10 points) Complementary Check-In.** If L is decidable by M, we can build a complement-enumerator as follows: take all strings in Σ^* in lexicographical order, and for each string, run M on it and output it if and only if M rejects it.

   Conversely, if $\overline{L}$ is finite, then it is decidable, and so is L. Otherwise, a decider for L works as follows: on input w, run a complement-enumerator for $\overline{L}$ until it prints either w (in which case, reject), or a string lexicographically after w (in which case, accept).

6. **(10 points) Reversal in Fortunes.** We will reduce from $\mathsf{ALL}_{\mathsf{CFG}}$. Given an arbitrary instance $\langle L \rangle$ of $\mathsf{ALL}_{\mathsf{CFG}}$, extend its alphabet by the symbol $\#$, and modify the grammar (straightforwardly) to obtain the language $L' = L\# \cup \#\Sigma^*$. This is the union of two languages – $L\#$ is the set of strings in L concatenated with the new symbol $\#$, and $\#\Sigma^*$ is the set of strings over the original alphabet Σ with an additional $\#$ appended at the beginning. We claim that $\langle L' \rangle \in \mathsf{ReverseCFL}$ if and only if $\langle L \rangle \in \mathsf{ALL}_{\mathsf{CFG}}$. Of course, if L' is its own reverse, then the reversal of every string in $\#\Sigma^*$ must be in $L\#$, which is the same as saying $L = \Sigma^*$. Conversely, if $L = \Sigma^*$, then of course L' is its own reverse by design. This completes the reduction.

7. **(3 × 5 points) A Resolved Issue.**

   (a) Suppose x and x' go to the same state in D, and suppose WLOG that x is unresolved and x' is resolved. By definition, there are suffixes y and z such that $xy \in L$ and $xz \notin L$. Note that $x'y \in L$ and $x'z \notin L$, as the DFA processes the suffixes in the same way for both strings from the same state. So x' cannot be resolved, contradiction.

   (b) By part (c), all resolved strings in D end up in at most two terminal states, which must have all their transitions into themselves. Therefore, any state in D with at least one transition into a different state corresponds to an unresolved string.

   (c) By the Myhill-Nerode Theorem, we claim all resolved strings in L form a single class. Otherwise, say x and x' are both resolved strings in L, but $x \not\sim_L x'$. This means WLOG there is some suffix w such that $xw \in L$ and $x'w \notin L$. Also, $|w| \geq 1$, as $x, x' \in L$. But then, $x'w \notin L$, but $x'\varepsilon \in L$, so x' cannot be resolved, contradiction.

   By the same logic, all resolved strings not in L also form a single class.

   Therefore, the resolved strings are partitioned into at most two parts (those in L and those not in L), and this corresponds to 0, 1, or 2 states.