Automata B-Exam—Solutions
January 1996

There are 4 questions worth 10 points each for a total of 40 points.

1. For any two integers p and q denote

$$A_{pq} = \{x \mid x = p + nq,\ x \geq 0,\ n - \text{integer}\}.$$

Show that the language

$$R = \{a^i \mid i \in A_{pq}\}$$

is regular.

Solution: Let $p_0 = \min\{p + nq \mid p + nq \geq 0,\ n \in \mathcal{N}\}$. Then

$$A_{pq} = \{x \mid x = p_0 + n|q|, n \geq 0, n \in \mathcal{N}\}$$

and

$$R = a^{p_0}(a^{|q|})^*.$$

Since R is generated by a regular expression, then R is regular.

2. Let

$$L = \{a^n(a+b)^{n+m}b^m \mid n\text{-odd},\ m\text{-even}\}.$$

Is L a CFL? Justify your answer.

Solution: The following non-deterministic push-down automata P accepts L by final state.

1. During any of the first n moves P reads an input symbol, verifies that the symbol is a, and stores it on the stack. (Note that P "guesses" the value of n.)
2. P checks that n is odd. (If n is even P has no next move.)
3. P reads the next n input symbols erasing a stack symbol per input symbol read. If there are less than n input symbols P has no next move.
4. During the next m moves P reads an input symbol and stores it on the stack.
5. P checks that m is even. (If m is odd P has no next move.)
6. P reads the next m input symbols erasing one stack symbol per input symbol. If there are exactly m symbols left and they are all b-s then P accepts. Otherwise P rejects.

3. Let the language L be recursively enumerable but not recursive. Show that for any Turing machine M that accepts L there are infinitely many strings on which M does not halt.

Solution: Let M be a Turing machine that accepts L and let L' be the set of all strings on which M does not halt. Then the language $L \setminus L'$ is recursive since M halts on all words of $L \setminus L'$. L' is also recursive language since it is finite. The union of two recursive languages is a recursive language and thus $L = (L \setminus L') \cup L'$ is recursive.

4. Let L_1 and L_2 be languages.

(a) Suppose that there is a 2^n time-bounded reduction of L_1 to L_2, and L_2 is in DTIME(2^n). What can you conclude about L_1?

(b) Suppose that there is a log-space reduction of L_1 to L_2, and L_2 is the *graph reachability problem*. What can you conclude about L_1?

Solution: (a) Since there is a 2^n time-bounded reduction of L_1 to L_2, then there exists a mapping $g(x)$ of L_1 to L_2 such that

(i) for any string x, $x \in L_1$ iff $g(x) \in L_2$;
(ii) there exists a TM M_1 that computes $g(x)$ in at most $2^{|x|}$ moves. This implies that if $|x| = n$, then the length of $g(x)$ is at most 2^n.

Since L_2 is in DTIME(2^n), there exists a TM M_2 such that for any string x, M_2 checks whether $x \in M_2$ in at most $2^{|x|}$ moves.

Let z be a string of length n. The following algorithm checks in $O(2^{2^n})$ time whether $z \in L_1$. (1) Compute $g(z)$ in time at most 2^n using M_1; (2) Check if $g(z) \in L_2$ in time at most 2^{2^n} using M_2. Total time is $O(2^{2^n})$. Thus L_1 is in DTIME(2^{2^n}).

(b) The graph reachability problem is in NSPACE($\log n$) (Theorem 13.13 in HU). By Lemma 13.2 in HU, L_1 is also in NSPACE($\log n$).