Assignment 11: Bringing CS 81 to a Logical Conclusion

Due: Wednesday, December 12

- Emails about this assignment should be directed to `cs81help@cs.hmc.edu`.
- The usual collaboration rules apply. You may *discuss* an exercise with any other student(s) currently taking CS 81 as long as:
  - You contribute equally;
  - You come away from this discussion only with *understanding in your head* — no written materials or computer notes may be retained;
  - Your submission is authored solely by you, on a separate occasion.
- You should refer only to materials from this semester of CS 81 (lecture notes, handouts, textbooks, grutors, profs, etc.).
- Bring a writeup/printout to class on the due date. Illegible answers will get no credit.
- Make sure your submission includes your name!

1 From a Logical Point of View

For each of the following statements,

a. Express the statement as a formula of predicate logic. You may use any standard mathematical or CS 81 notation, and may define new predicates or functions as needed, but your formula must contain at least two logical symbols ($\forall$, $\exists$, $\rightarrow$, $\wedge$, etc.).

b. Based on the logical form of your formula, in no more than one sentence suggest the most direct strategy for proving it. Identify some natural deduction rule(s) involved.

For example, if the statement were "the language L_1 is not regular," we might say

a. $\neg\exists r.\ \big(\text{Regex}(r)\ \wedge\ L(r) = L_1\big)$
   where $\text{Regex}(r) \iff$ "r is a regular expression"
   and $L(r)$ is the language of r (as usual).

b. Assume that we have a regular expression r whose language is L_1, and derive a contradiction. ($\neg i$ and $\exists e$)

Of course, answers are not unique. Another possibility might be $\neg\exists r.\ P(r, L_1)$ where $P(a, B) \iff$ "a is a regular expression whose language is B." Or we could define predicates or functions involving finite-state machines, in which case the proof strategy would change.

Notice, though, that the free variables of the English statement (in the above example, L_1 is the only free variable) are the same as the free variables of the logical formula. (The variables in the formula are L_1 and r, but r is not free because it occurs inside the quantifier $\exists r$). All your translations should have the same free variable(s) as the original statements.

1. For every finite state machine, there is a regular expression with the same language.
2. The state machine M_1 is deterministic. [Don't use "is deterministic" or "is a DFA" as one of your defined predicates, because your formula won't have enough logical symbols, and it won't give you any guidance for a direct proof.]
3. For every n-state NFA, there is a DFA with no more than 2^n states that accepts the same language.
4. It is possible that $L_1 \cup L_2$ is regular, but neither L_1 nor L_2 is regular.
5. The function $f : A \to B$ is injective (a.k.a. one-to-one). [You may or may not want to use the notation B^A to denote the set of all functions from A to B.]
6. The set S is countable.
7. Context-Free Grammar G_1 is ambiguous. [I used the predicate $PT(G, s, P) \iff$ "String s is produced by the CFG G, as shown by parse tree P".]
8. The formal language L_1 is decidable.
9. The Halting Problem for Turing Machines is undecidable.
10. The Halting Problem for Turing Machines reduces to the Accepts-ε problem for TMs.
11. (The statement of the Pumping Lemma for Regular Languages.)
12. (The statement of Rice's Theorem.)
13. (The statement of Church's Thesis.)

2 Logical Intuition

In physicist Richard Feynman's autobiography *Surely You're Joking, Mr. Feynman!*, he describes a series of bets that he won against his mathematician friends.

> I challenged them: "I bet there isn't a single theorem that you can tell me—what the assumptions are and what the theorem is in terms I can understand—where I can't tell you right away whether it's true or false." …
>
> I had a scheme, which I still use today when somebody is explaining something that I'm trying to understand: I keep making up examples. For instance, the mathematicians would come in with a terrific theorem, and they're all excited. As they're telling me the conditions of the theorem, I construct something that fits all the conditions. You know, you have a set (one ball)—disjoint (two balls). Then the balls turn colors, grow hairs, or whatever, in my head as they put some conditions on. Finally they state the theorem, which is some dumb thing about the ball which isn't true for my hairy green ball thing, so I say, "False!" …Then I point out my counterexample.

Using CS 81 vocabulary, explain what Feynman is doing and why it worked.