A GIRL HAS NO NAME
Computer Science 81
Homework 10
4/11/17

1, 2, 3, 4, 5, 6, 7

1. [7 points] This concerns the PCP language as described in the lecture. Show that the candidate strings (strings that are well-formed pairs, but which might not have a solution) can be mapped to first-order predicate logic clauses in such a way that there is a solution iff the set of clauses is unsatisfiable.

Hint: Treat the symbols a and b as function symbols. Define a predicate that simulates the concatenation of strings.

■

2. [3 points] Demonstrate your technique in the previous problem using Prover9, on the following three examples:

a. (a, ab)(ba,a)

b. (ba,baa)(ab,ba)(aa,a)

c. (a,ab)(ba,ab)(bb,ba)

3. [5 points] As we know, strings and natural numbers are equivalent, in that strings can be encoded as numbers and numbers as strings. There are multiple one-to-one correspondences[a] between the set of natural numbers and the set of strings over any finite alphabet. For example, if the alphabet is $\{1\}$, then the number n corresponds to 1^n (a string of n 1's). In this sense, we can treat a language as a set of numbers. We can then speak of a set of numbers as being decidable, recognizable, etc. the same as we would a language.

Suppose that $f : N \to N$ is a computable function that enumerates a language $L \subseteq N$. From the Enumeration Theorem, we know that L is therefore recognizable. Suppose further that f has the **monotone** property:

$$\text{If } i < j \text{ then } f(i) < f(j).$$

Prove that an infinite language is decidable iff it is enumerable by a monotone computable function. (Use the Church-Turing thesis in an informal proof.)

[a] A one-to-one correspondence between two sets is a function that is both one-to-one and onto, that is, a function $f : A \to B$ such that (a) $\forall x \in A \forall y \in A (f(x) = f(y) \to x = y)$ and (b) $\forall z \in A \exists x \in A f(x) = z$

■

4. [5 points] Give an informal (but nonetheless convincing) algorithm for **deciding** the following language:

$$\{< M > \mid M \text{ is a Turing machine that, when started on an all-blank tape, eventually prints something other than a blank}\}$$

As usual, $< M >$ means a string encoding the rules for M.

5. [3 points] Show that if L and M are arbitrary recognizable languages, then $L \cup M$ is also recognizable. Use the Church-Turing thesis. (Before answering, review the distinction between recognizable and decidable.) In your proof, you must address the issue that neither L nor M necessarily halt on a given input, i.e. they don't have to have deciders.

■

6. [2 points] Show that the following language is recognizable:

$$\{< M > \mid M \text{ is a Turing machine that accepts the empty string } \epsilon\}.$$

Here ¡M¿ represents the encoding of M, as usual.

■

7. [5 points] Give a convincing proof that the following language of equations over alphabet $\{1, +, =\}$ using 1-adic numerals is not decidable by any finite-state machine:

$$L = \left\{ 1^m + 1^n = 1^{m+n} \mid m > 0 n > 0 \right\}$$

Here 1^m means a sequence of m 1's. The first few strings in L are:

$$\begin{aligned} &1 + 1 = 11 \\ &1 + 11 = 111 \\ &11 + 1 = 111 \\ &11 + 11 = 1111 \\ &11 + 111 = 11111 \\ &111 + 11 = 11111 \end{aligned}$$

etc. (In this language, the + symbol is used exactly once on the left-hand side and not used at all on the right-hand side. The = symbol is used once in each string.)

■