
CPSC 351

Problem Set 9

Decidability and PDA

Submission Details:

1. I encourage you to work in groups of up to three people
2. Submit only one copy of the solutions even though more than one person worked on them
3. The problems are due December 10 at 5:00 in my office, Herak 230.

The Problems

10 points

1. At the very beginning of our discussion of chapter 4, we talked about framing computational problems as languages. Here's your chance to do just that. Suppose you want to show that a DFA and a regular expression are equivalent. Express this problem as a language and show that it is decidable. The constructions associated with Theorem 1.54 and the result from Theorem 4.5 are relevant. To show that a language is decidable, you first need to express the language formally then you need to show that there is a Turing machine that decides it. Use Sipser's notation for your Turing machine. Do not reducibility to solve this problem. That is a matter for chapter 5.


10 points

2. Suppose S is the set of all infinite sequences of 0s and 1s. $001010101\dots$ is a member of S , for example. Show that S is uncountable.

20 points

3. Let $T = \{ 0^n 1^n \mid n \geq 0 \}$, a set that we have considered many times. You'll find the state diagram for a PDA that recognizes T on p. 115. Call that PDA, P . Write a python program that simulates P . The program is invoked from the Linux command line.

The program must have a function that accepts a formal definition of P (p. 113) as a tuple $(Q, \Sigma, \Gamma, \delta, q_0, F)$ along with its input, w . This could either be a simple function, called, say, "simulate," or the constructor for a class called, say, "PDA." The PDA will accept continuous input, halting when the user types CTRL-c. It prints 'accept' for each string that is an element of T , 'reject' if it is not. The PDA also rejects if a symbol is not an element of Σ or if a state is not an element of Q .



The PDA is an extension of the DFA you wrote for Project 7. This time, instead of δ being a simple function from a state and symbol to a new state, it is a function from a state, a symbol, and stack top to a new state and new stack top.

Because you have to manipulate a stack as well as input, the PDA does not lend itself quite so easily to the use of a dictionary. An approach, since the δ transitions are functions, is to define functions for each of the states. If you choose this approach, rather than the dictionary approach, you can relax the formal definition of the δ function.