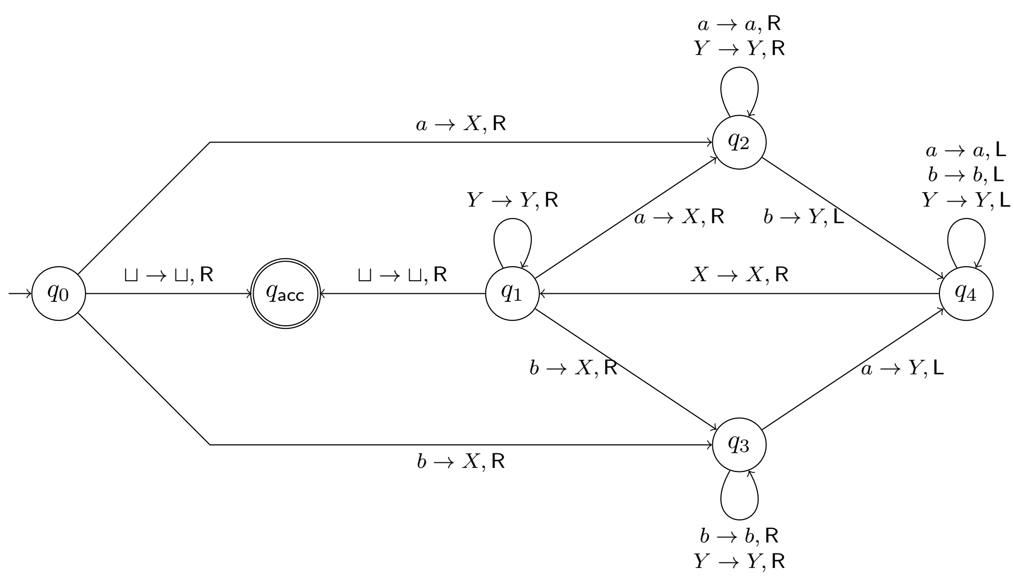

Homework 7
CIS 770: Formal Language Theory

Assigned: April 21, 2016   Due on: April 28, 2016

Instructions: This homework has 2 problems that can be solved individually. Please follow the homework guidelines given on the class website; submission not following these guidelines will not be graded.

Recommended Reading: Lecture 20 and 21.

Problem 1. [Category: Comprehension] Consider the following Turing Machine M with input alphabet $\Sigma = \{a, b\}$. The reject state q_{rej} is not shown, and if from a state there is no transition on some symbol then



as per our convention, we assume it goes to the reject state.

1. Give the formal definition of M as a tuple. **[5 points]**
2. Describe each step of the computation of M on the input *baabab* as a sequence of instantaneous descriptions. **[3 points]**
3. Describe the language recognized by M. Give an informal argument that outlines the intuition behind the algorithm used by M justifies your answer. **[2 points]**.

Problem 2. [Category: Comprehension+Design+Proof] A *2-PDA* is a pushdown automaton that has *two stacks*. Thus, like a PDA, it is a nondeterministic machine. In each step, the current control state, the current input symbol read (which could be ϵ meaning nothing is read from the input), the symbol popped from the first stack (which could be ϵ, meaning that no symbol is popped), and the symbol popped from the second stack (which again could be ϵ), determine what the possible next state is, and the symbols to be pushed onto each of the two stacks (which could be ϵ meaning that no symbol is pushed) are. Additionally, like a PDA, the 2-PDA accepts an input if it reaches a accepting/final state after reading the entire input, irrespective of the contents of either of its two stacks.

1. Give the formal definition of a 2-PDA as a tuple, giving the domain and range of the transition function. **[2 points]**
2. Give the formal definitions of the instantaneous description of the 2-PDA, computation on a word, and the language accepted by the machine. **[3 points]**
3. Prove that if M is a (deterministic, single-tape) Turing machine then there is a 2-PDA P such that $\mathbf{L}(P) = \mathbf{L}(M)$. You only need give the construction of the 2-PDA; you do not have to prove that your construction is correct. **[5 points]**

Aside: Any recursive program, without dynamic allocation, using only variables that are Boolean valued (or variables taking values in some finite set) can be modelled using a pushdown automaton: the control state gives value to each program variable and the program counter (resulting in finitely many states since all these take values in some finite set), and the stack is used to model the call stack of the program. Thus a 2-threaded program (or multi-threaded program, in general) can be modelled by a 2-PDA (or a k-PDA, for some value of k), where the different stacks model the call stack of each of the threads. This result suggests that multi-threaded programs (with only Boolean variables) can compute anything that a general program with variables of arbitrary type can. This is also to contrast with the limited computational power of a PDA with a single stack.