CS 361: Theory of Computation

Assignment 6 (due 10/22/2025)

Instructor: Shikha Singh

LaTeX Source for Solutions: https://www.overleaf.com/read/zqppwbsfcfdb#fc791a

Problem 1. Let a k-PDA be a push-down automata with k stacks. Thus a 0-PDA is an NFA, a 1-PDA is a conventional PDA. We know that 1-PDAs are more powerful than 0-PDAs. In a single-transition, a 2-PDA can read an input symbol, change state, and push/pop the top of each stack.

(a) Justify that a 2-PDA is more powerful than a 1-PDA by arguing that it is possible to recognize a non-context-free language (such as $\{a^n b^n c^n \mid n \geq 0\}$) using two stacks.

Solution.

□

(b) Show that a 3-PDA is not more powerful than a 2-PDA. (*Remark.* Using the Church-Turing thesis, it is sufficient to argue that a a 2-PDA can simulate a TM.)

For this part of the answer, state (informally) how a 2-PDA can simulate a single-tape TM by mimicking its transition function and input tape through the use of two stacks.

Solution.

□

Problem 2. A Turing machine with stay put instead of left is similar to an ordinary Turing machine, but the transition function has the form $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{R, S\}$.

At each point, such a Turing machine can move its head right or stay in the same position. A transition that is of the form $(p, a) \rightarrow (q, b, S)$ means that the TM has changed its state from p to q and replaced the symbol on its current head position from a to b but the position of the head has not changed.

How powerful is such a TM? We approach this question in two parts.

(a) Show that for each TM, M, with stay put instead of left, there exists a TM, M', that always moves right on each transition such that $L(M) = L(M')$. *Hint. Show that a maximal contiguous sequence of stay-put transitions can be replaced by a single right transition without affecting the language of the machine.*

Solution.

□

(b) Now show that the Turing machines with stay put instead of left recognize exactly the class of regular languages by constructing an equivalent DFA. You must construct a DFA that recognizes the same language as such a TM. *You do not need to prove correctness formally, but you must provide the 5-tuple of the DFA including a description of its transition function.*

Solution. □

Problem 3. (a) Show that the class of decidable languages is closed under complement and intersection. That is, show that if a language A and B are decidable, then so are $\overline{A}$ and $A \cap B$.

Solution.

$\square$

(b) Does the same construction work for Turing-recognizable languages? In particular, are Turing-recognizable languages closed under (i) complement and (ii) intersection? Provide a brief justification.

Solution. $\square$

Decidable Properties of DFAs. Below, is a solved example for Problem 4. The goal is to convey the format and level of detail you should include in such questions.

Solved Example. Let $A = \{\langle R, S\rangle \mid R$ and S are regular expressions and $L(R) \subseteq L(S)\}$. Show that A is decidable.

Solution. In class, we showed that $\mathsf{E}_{\mathsf{DFA}}$ is decidable, so let S be a TM that decides $\mathsf{E}_{\mathsf{DFA}}$. The following TM decides A.

$Y =$ "On input $\langle R, S\rangle$:

1. Since regular languages are closed under intersection and complementation, construct a DFA M that recognizes the language $L(R) \cap \overline{L(S)}$.
2. Run S on input $\langle M\rangle$.
3. If S accepts, then *accept*. Otherwise, *reject*."

□

Problem 4. Consider the problem of determining whether or not a given DFA accepts a string with more 1s than 0s. Express this problem as a language and show that it is decidable.

Solution. □