

# CS46 practice problems 6

These practice problems are an opportunity for discussion and trying many different solutions. They are **not counted towards your grade**, and **you do not have to submit your solutions**. The purpose of these problems is to get more comfortable with reasoning and writing about Turing machines. You should be *practicing writing out descriptions and proofs* for your solutions to these problems.

1. **The power of two stacks.** We know that PDAs with zero stacks are just NFAs. We also know that PDAs with 1 stack are more powerful than NFAs, because they can recognize  $\{a^n b^n\}$  which is not a regular language. We *also* know that PDAs with 2 stacks are more powerful than 1-stack PDAs, because they can recognize  $\{a^n b^n c^n\}$  which is not context-free. How do 2-stack PDAs compare with Turing machines?
  - (a) Show that every Turing machine has an equivalent 2-stack PDA. (Every *standard* Turing machine: one tape, one read/write head. Don't make this more complicated than it needs to be.)
  - (b) Show that every 2-stack PDA has an equivalent Turing machine.
  - (c) What about a 3-stack PDA? Will it be more powerful, less powerful, or equivalent to a Turing machine? Support your answer with a proof.
  - (d) What about a 4-stack PDA? What about 5 stacks? 6 stacks?  $n$  stacks?
2. (Sipser 3.8a) Give an **implementation-level** description of a Turing machine that decides the following language over the alphabet  $\{0,1\}$ :

$$\{w \mid w \text{ contains an equal number of 0s and 1s}\}$$

- (a) First, give an **implementation-level** description this Turing machine.
  - (b) Next, give a **formal** description of this Turing machine. You can either give the full 7-tuple or draw the state diagram. (It should not be *too* complicated.)
4. **Closure properties for decidable languages.** (Sipser 3.15)

Show that the collection of Turing-decidable languages is closed under the operations:

- (a) union
- (b) concatenation
- (c) Kleene star
- (d) complementation
- (e) intersection