

**Policy on homeworks**

- *Collaboration:* You may discuss a question with any other student currently taking CS81 provided: (i) both of you contribute equally; (ii) you come away from any discussion with an understanding in your mind (and no archived solution of any form is retained); (iii) your submission is your own work prepared by yourself on a separate occasion.
- *Reference materials:* You should only refer to materials from this semester of CS81 (class notes, handouts, textbooks, grutors, instructor, etc).
- *Submission:* Your submission should be legible or is prepared using TeX.

**Turing Machines** Recall our conventions regarding Turing machines:

- The tape is *one-way infinite*. The machine hangs if the tape reader attempts to move left off the leftmost tape square.
- There is a single halt state  $h$ .
- If a Turing machine  $M$  computes a function  $f$ , then  $(s, \square \alpha \square) \vdash_M^* (h, \square f(\alpha) \square)$ .

1. Consider the following Turing machine (which we call LEFTSHIFTER).

- (a) (True or False) LEFTSHIFTER removes the leftmost blank on its tape; that is, it transforms its tape from  $\square \alpha \square$  to  $\alpha \square$ . If True, describe in English how this Turing machine works; if False, describe a way to fix and remove the bugs.

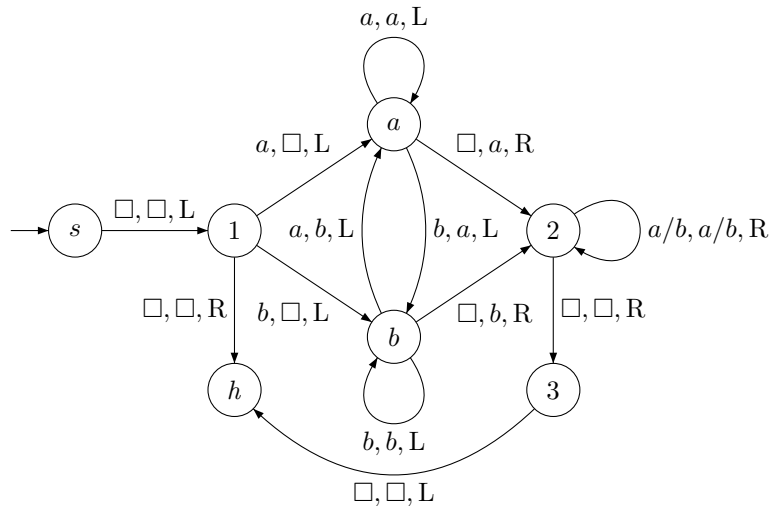


Figure 1: A Turing machine called LEFTSHIFTER.

- (b) Can you find a smaller Turing machine that makes its leftmost blank disappear?
- (c) (True or False) LEFTSHIFTER will also work if there is more than one blank to the left of  $\alpha$ . If True, explain why; if not, determine if there is a way to fix this so that the leftmost (and only the leftmost) blank is removed.

2. Describe a Turing machine  $M$  that concatenates two strings  $\alpha$  and  $\beta$ :

$$(s, \square \alpha \square \beta \square) \vdash_M^* (h, \square \alpha \beta \square)$$

3. Describe a Turing machine  $M$  that computes unary addition over the alphabet  $\Sigma = \{0\}$ .

**Computability** For each of the following languages over  $\Sigma$ , state whether the language is decidable, semi-decidable, or not even semi-decidable. Justify your answers. We assume a suitable encoding  $\langle M \rangle$  for each of the models.

1.  $L = \{\langle M \rangle : M \text{ is a DFA where } L(M) = \Sigma^*\}$
2.  $L = \{\langle M \rangle : M \text{ is a CFG where } \epsilon \notin L(M)\}$
3.  $L = \{\langle M \rangle : M \text{ is a DTM where } L(M) = \emptyset\}$