

We may grade a **subset of the assigned questions**, to be determined after the deadline, so that we can provide better feedback on the graded questions.

Unless otherwise stated, each question requires sufficient justification to convince the reader of the correctness of your answer.

For bonus questions, we will not provide any insight during office hours or Piazza, and we do not guarantee anything about the difficulty of these questions.

We strongly encourage you to typeset your solutions in L^AT_EX.

If you collaborated with someone, you must state their name(s). You must write your own solution for all problems and may not look at any other student's write-up.

0. If applicable, state the name(s) and username(s) of your collaborator(s).

Solution:

1. Define a language L which corresponds to each decision problem stated in parts (a) through (c).

Example: Given integers x, y , determine if x is divisible by y .

$$L = \{\langle x, y \rangle : x, y \in \mathbb{Z} \text{ and } x \bmod y \equiv 0\}$$

The language L consists of all *encodings* of pairs of integers $\langle x, y \rangle$ such that $x \bmod y \equiv 0$, i.e., y divides x . (You do not need to explain how the encoding works. In this case, we could say it is a string in $\{-, 0, 1, \#\}^*$ where x, y are written in binary, negated with '-', and separated by a #, e.g., $\langle -5, 3 \rangle = -101\#11$.)

- (a) Given an integer x , is x a perfect cube?

Solution:

- (b) Given an integer m , is m odd?

Solution:

- (c) Given natural numbers p and q , do p and q share a common prime factor?

Solution:

2. Draw a DFA which decides the following languages:

- (a) $\{x \in \{0, 1\}^* \mid x \text{ has an even number of 0s and the last character of } x \text{ is } 1\}$.

- (b) $b(aaa)^*b$.

Reminder: the alphabet is $\{a, b\}$.

- (c) Let $x \in \{0, 1\}^*$. In this problem, we will consider the *reversal* of x as defined in lecture slides, where we read least significant digits first.

Formally, if we have read n bits so far, let $x = x_0x_1 \dots x_n$. The *unsigned numeric value* of x is notated as $\langle x \rangle_*$, where $\langle x_0x_1 \dots x_n \rangle_* := \sum_{i=0}^n 2^i x_i$. (Note the reversed binary representation, where the leftmost bit x_0 corresponds to 2^0 , x_1 corresponds to 2^1 , etc.) We would read this bitstring from left to right.

Devise a DFA that decides

$$L = \{x \in \{0, 1\}^* \mid \langle x \rangle_* \bmod 3 = 0\}.$$

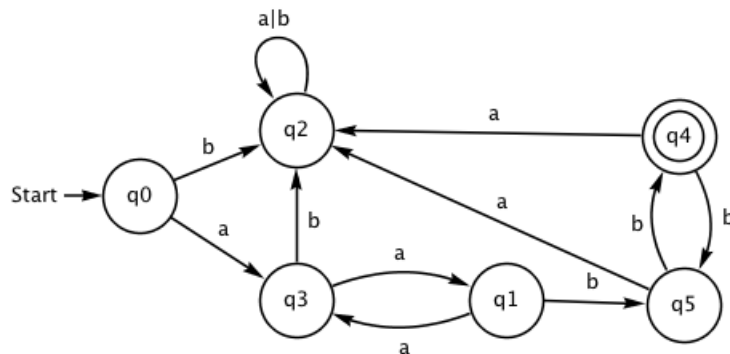
Note:

- The empty string has value zero. $\langle \varepsilon \rangle_* = 0$.
- There may be extra zeros. For example, $\langle 11010000 \rangle_* = \langle 1101 \rangle_* = 11$.
- You may represent your DFA as a table of state transitions or as a diagram, whichever is easier.

Hint: Try to characterize $\langle x_0x_1 \dots x_k \rangle_*$ in terms of the previously read value $\langle x_0x_1 \dots x_{k-1} \rangle_*$. Consider the value of $(2^k \bmod 3)$ for even and odd k .

Solution:

3. What language does the following DFA decide? Give your answer as a regular expression.



Solution:

4. A robot is walking on a 2D-grid. It receives instructions from the set $\Sigma = \{N, W, S, E\}$, which indicate a move 1 step to the north, west, south and east, respectively. The robot starts at the origin, and receives an instruction string $x \in \Sigma^*$. The robot wants to know whether it is at the origin after executing x . Prove that if the robot uses a DFA, it cannot answer this question.

Solution:

5. Define the “fractional knapsack problem” as a variant of the original knapsack problem which allows one to take any *partial* amount of an item (that is, for an item of value v and weight w , one may select some $t \in [0, 1]$ and add $t \cdot v$ and $t \cdot w$ as weight and value to the knapsack.)
- (a) For this fractional variant, describe a greedy algorithm which returns the maximum value given the capacity of the knapsack. Explain (1-2 sentences should be sufficient) why your algorithm gives the optimal value.
 - (b) Prove that the optimal value for the fractional knapsack is at least as large as that for the original 0-1 knapsack problem (for the same weights, values, and knapsack capacity).
 - (c) Prove that the greedy approach you proposed in part (a) may not find an optimal solution to the original 0-1 knapsack problem.

Solution: