Spring 2020 CIS 262

Automata, Computability and Complexity
Jean Gallier

Homework 5

February 13 2020; Due February 21, 2020, midnight

"B problems" must be turned in.

Do Problem B1 and B2.

Do Problem B3 or B4.

Problem B1 (30 pts). Consider the following NFA accepting the language $L = \{aa, aaa\}^*$ (over the alphabet $\Sigma = \{a\}$):

	ϵ	a
0	$\emptyset$	$\{1\}$
1	$\emptyset$	$\{2, 3\}$
2	$\emptyset$	$\{3\}$
3	$\{0\}$	$\emptyset$

State 0 is the start state and the only final state.

(1) Apply the subset construction algorithm presented in the notes to obtain a DFA for L (you should get a DFA with 5 states). You must show the transition table of the DFA obtained by applying the subset construction containing the states of the DFA as subsets of the original NFA, the symbolic names you are assigning to them (also called their indices), and the actual transition table, as shown in the notes. You don't need to show the steps that you went through to construct this table.

(2) The language $L = \{aa, aaa\}^*$ is of the form $L = \{a\}^* - S$, where S is a finite set of strings. What exactly is S?

(3) Give a DFA with 3 states accepting L.

Problem B2 (50 pts). Let R be any regular language over some alphabet Σ. Prove that the language

$$L = \{u \in \Sigma^* \mid \exists v \in \Sigma^*,\, uv \in R,\, |u| = |v|\}$$

is regular.

Hint. Think nondeterministically.

Problem B3 (50 pts). Let $D = (Q, \Sigma, \delta, q_0, F)$ be a DFA. Recall that a state $p \in Q$ is *accessible* or *reachable* iff there is some string $w \in \Sigma^*$, such that

$$\delta^*(q_0, w) = p,$$

i.e., there is some path from q_0 to p in D. Consider the following method for computing the set Q_r of reachable states (of D): define the sequence of sets $Q_r^i \subseteq Q$, where

$$Q_r^0 = \{q_0\},$$
$$Q_r^{i+1} = \{q \in Q \mid \exists p \in Q_r^i, \exists a \in \Sigma,\ q = \delta(p, a)\}.$$

(1) Prove by induction on i that Q_r^i is the set of all states reachable from q_0 using paths of length i (where i counts the number of edges).

Give an example of a DFA such that $Q_r^{i+1} \neq Q_r^i$ for all $i \geq 0$.

(2) Give an example of a DFA such that $Q_r^i \neq Q_r$ for all $i \geq 0$.

(3) Change the inductive definition of Q_r^i as follows:

$$Q_r^{i+1} = Q_r^i \cup \{q \in Q \mid \exists p \in Q_r^i, \exists a \in \Sigma,\ q = \delta(p, a)\}.$$

Prove that there is a smallest integer i_0 such that

$$Q_r^{i_0+1} = Q_r^{i_0} = Q_r.$$

Define the DFA D_r as follows: $D_r = (Q_r, \Sigma, \delta_r, q_0, F \cap Q_r)$, where $\delta_r \colon Q_r \times \Sigma \to Q_r$ is the restriction of δ to Q_r. Explain why D_r is indeed a DFA.

(4) Prove that $L(D_r) = L(D)$.

Problem B4 (50 pts). This problem illustrates the fact that languages can be used to encode complicated geometric shapes.

(1) Given the alphabet $\Sigma = \{0, 1, c\}$, construct a DFA accepting the following language:

$$L_S = \{u_1 c u_2 c \cdots c u_{n-1} c u_n \mid n \geq 1,\ u_i \in \{00, 01, 10\}\}.$$

The language L_S has a geometric interpretation as a certain subset of $\mathbb{R}^2$ (really $\mathbb{Q}^2$) as follows: Given any string $w = u_1 c u_2 c \cdots c u_n \in L_S$, if we denote the first character of u_i by x_i and the second character by y_i so that $u_i = x_i y_i$ ($x_i, y_i \in \{0, 1\}$), then we obtain two strings

$$w_x = x_1 \cdots x_n$$
$$w_y = y_1 \cdots y_n.$$

For example, if $w = 01c10c00c10$, then $w_x = 0101$ and $w_y = 1000$. Now, a string $z \in \{0,1\}^+$ can be interpreted as decimal number in $[0,1]$ written in base two! Indeed, if

$$z = z_1 z_2 \cdots z_k, \quad \text{with } z_i \in \{0,1\},\ 1 \leq i \leq k,$$

we interpret z as the number $n(z) = 0.z_1 z_2 \cdots z_k$, given by

$$n(z) = z_1 2^{-1} + z_2 2^{-2} + \cdots + z_k 2^{-k}.$$

Finally, a string $w = u_1 c u_2 c \cdots c u_n \in L_S$ is interpreted as the point in $\mathbb{R}^2$ of coordinates (x_w, y_w), with

$$x_w = n(w_x)$$
$$y_w = n(w_y).$$

(2) Prove that the set

$$S = \{(x_w, y_w) \in \mathbb{R}^2 \mid w \in L_S\}$$

is the subset of points $(x, y) \in \mathbb{R}^2$ in the triangle T determined by the vertices $A = (0,0)$, $B = (1,0)$, $C = (0,1)$ such that if the binary representations of x and y are

$$x = 0.x_1 x_2 \cdots x_n, \quad y = 0.y_1 y_2 \cdots y_n,$$

then $(x_i, y_i) \neq (1,1)$ for $i = 1, \ldots, n$.

(3) Given any point $p \in S$ where p is given by its coordinates in binary as

$$p = (0.a_1 a_2 \cdots a_n,\ 0.b_1 b_2 \cdots b_n), \quad (n \geq 0)$$

with the understanding that $p = (0,0) = A$ if $n = 0$, show that any other point $q \in \mathbb{R}^2$ given by

$$q = (0.a_1 a_2 \cdots a_n x_1 \cdots x_m,\ 0.b_1 b_2 \cdots b_n y_1 \cdots y_m), \quad (n \geq 0, m \geq 0)$$

(with the understanding that $q = p$ if $m = 0$) belongs the small square whose lower left corner is p and whose sides have length 2^{-n}, and that any other point $q \in S$ given by

$$q = (0.a_1 a_2 \cdots a_n x_1 \cdots x_m,\ 0.b_1 b_2 \cdots b_n y_1 \cdots y_m), \quad (n \geq 0, m \geq 0)$$

belongs to the small lower triangle similar to (A, B, C) contained in the small square that we just defined. For example, if $p = (0.01, 0.10)$ is the point shown in red, then the points $q \in S$ belong to the triangle showed in blue in Figure 1.

Use this fact to prove that S is obtained by using the following recursive procedure. Starting with the triangle $S_0 = T$, form the four smaller triangles obtained by joining the midpoints of the sides of T_0, and delete the middle triangle. We obtain a subset S_1 of S_0 consisting of the union of three triangles. Repeat the procedure on the remaining three triangles. After n steps we obtain a shape S_n consisting of 3^n triangles. The first two steps of this procedure are illustrated in Figure 2 below.

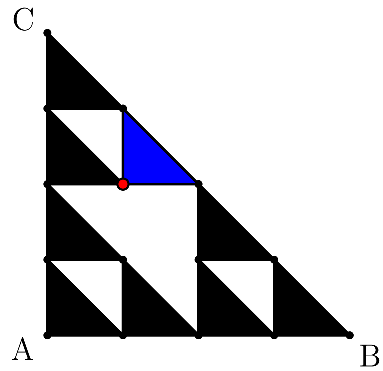

Figure 1: The small triangle containing the points q.

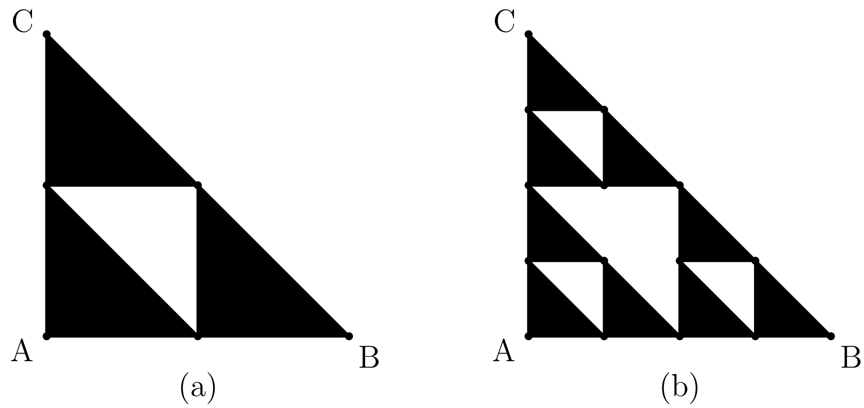

Figure 2: (a) The shape S_1 after one iteration. (b) The shape S_2 after two iterations.

Draw a picture by iterating the above procedure 3 or 4 times. What sort of shape do you get?

Remark: The set of points in S defined above the set of rational points of a fractal set known as the *Sierpinski gasket*. The S_n are approximations of S and S is the limit (in some suitable sense) of the sequence of S_n as n goes to infinity.

Extra Credit (30 pts). Write a computer program to display the Sierpinski gasket. You may use any language available but you must comment your code and explain any parts that are not obvious to understand. Please, submit your source and a printout of the output of your code.

No cheating: I want the version using a *right triangle*, as shown above, **not** the version using an *equilateral triangle*!

What happens to the length of the "perimeter" of S_n, that is, the sum of the lengths of all its boundary edges?

TOTAL: 130 + 30 points