

Harvard University  
Computer Science 121

Problem Set 0

Due September 29, 2015 11:59pm

Problem set by Kevin Zhang

Collaboration Statement: I collaborated on this assignment with Tomoya Hasegawa, Emily Wang, and Hugo Yen, got help from no one else, and referred to nothing else.

PART A (Graded by Juan, Sam)

PROBLEM 1 (1+1+1+1+1+1+1 points, suggested length of 3/4 page)

Determine the cardinality of each of the following sets (countably infinite, uncountably infinite, finite). If finite determine number of elements. Briefly justify each of your answers. You can assume  $\Sigma = \{a, b\}$ .

- (A)  $\{D : D \text{ is a DFA that accepts the string } ab\}$
- (B) Set of all languages over the alphabet  $\{a\}$
- (C)  $\{P(\Sigma^*)\}$
- (D)  $\emptyset \times \mathbb{N}$
- (E) Set of all syntactically correct OCaml programs.
- (F)  $P(\Sigma^*) \setminus REG$  where  $REG = \{L : L \text{ is a regular language}\}$
- (G)  $REG \setminus P(\Sigma^*)$

**Solution.**

(A) **Countably infinite**; this set is the set of DFAs that recognize the set of languages  $\{L \cup \{ab\} \mid L \in P(\Sigma^*)\}$ . Every DFA corresponds to a regular expression; hence, this set of DFAs is at most countably infinite, since the set of regular expressions is countably infinite as well.

(B) **Uncountably infinite**; the alphabet  $\{a\}$  yields a set of strings defined by  $\{a^*\}$  (which is countably infinite; simply map the length of each string to the corresponding natural number), and therefore the set of all languages over the given alphabet is a power set of a countably infinite set, resulting in an uncountably infinite set.

(C) **Finite**; this is just a set with one element in it.

(D) **Finite**; the cross product of the empty set with any other set yields the empty set. Hence, the number of elements is 0.

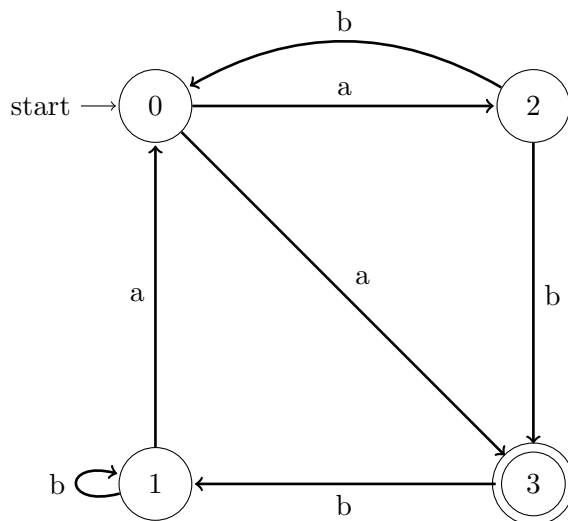
(E) **Countably infinite**; note that any OCaml program is just a string, where the alphabet is the set of all characters used to write an OCaml program (which is obviously finite). Hence, the set of valid OCaml programs is just a subset of the language given by  $\{\Sigma^*\}$ , where  $\Sigma$  is the alphabet above. But this language is a countably infinite set of strings (as all languages given by the Kleene star of a finite alphabet are). The set of valid OCaml programs is also obviously not finite (consider an arbitrary amount of whitespace, for example); hence, it must be countably infinite.

(F) **Uncountably infinite**; we know that  $P(\Sigma^*)$  is uncountably infinite (this is from lecture, and the proof is too long to go through here). Moreover, the set of regular languages is countably finite (after all, every regular language is just a string constructed from the alphabet of characters used in writing a regular language). We will prove later that subtracting a countably infinite set from an uncountably infinite set results in an uncountably infinite set; we use it here to prove that the given set is uncountably infinite.

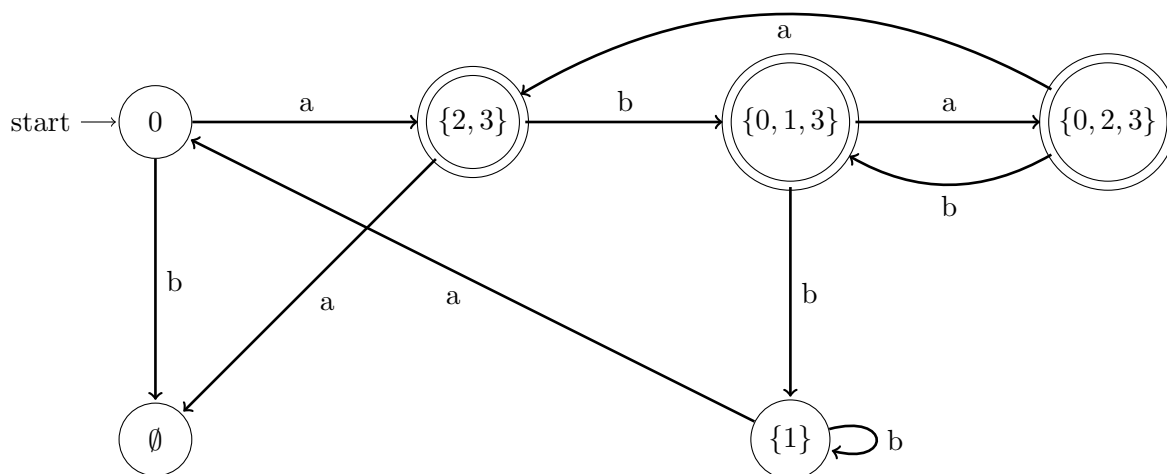
(G) **Finite**; note that every regular language is in the power set of the set of strings given by  $\Sigma^*$ ; hence, the given subtraction results in the empty set. Hence, the number of elements is 0.

PROBLEM 2 (6 points, suggested length of 1/4 a page)

Convert the following NFA into a DFA using subset construction. Provide a formal description (5-tuple) and diagram for full credit.



**Solution.**



The above DFA  $= (M', \Sigma', \delta', q'_0, F')$  is defined as follows:

- $M' = \{\{0\}, \{2, 3\}, \{0, 1, 3\}, \{0, 2, 3\}, \{1\}, \emptyset\}$
- $\Sigma' = \{a, b\}$
- $\delta' :=$

$q_0$	a	b
$\{0\}$	$\{2, 3\}$	$\emptyset$
$\{2, 3\}$	$\emptyset$	$\{0, 1, 3\}$
$\{0, 1, 3\}$	$\{0, 2, 3\}$	$\{1\}$
$\{0, 2, 3\}$	$\{2, 3\}$	$\{0, 1, 3\}$
$\{1\}$	$\{0\}$	$\{1\}$

- $q'_0 = \{0\}$
- $F = \{\{2, 3\}, \{0, 1, 3\}, \{0, 2, 3\}\}$

The reject state and transitions to rejection are omitted.

### PROBLEM 3 (3+2+3 points, suggested length of 3/4 page)

- (A) Prove that if  $A$  is an uncountable set and  $B$  is a countable set, then  $A \setminus B$  is uncountable.
- (B) Find 2 examples: one where the intersection of two countably infinite sets is finite and another where it is countably infinite.
- (C) Find 3 examples: one where the intersection of two uncountably infinite sets is finite, another where it is countably infinite, and another where it is uncountable.

**Solution.**

(A) Suppose that  $A \setminus B$  is countable. Then since the union of two countable sets is countable,  $(A \setminus B) \cup B$  is countable. But  $(A \setminus B) \cup B$  is just  $A$ , which is uncountable; a contradiction! Thus,  $A \setminus B$  cannot be countable.

(B) For the first example, let  $A$  be the positive integers and  $B$  be the positive integers. Their intersection is the empty set, a finite set.

For the second example, let  $A$  be the positive integers and  $B$  be the integers. Their intersection is just  $A$ , which is countably infinite.

(C) For the first example, let  $A$  be the positive reals and  $B$  be the negative reals. Their intersection is the empty set, a finite set.

For the second example, let  $A$  be the positive reals and  $B$  be the negative reals joined with the integers. Their intersection is the positive integers, which is countably infinite.

For the last example, let  $A$  be the positive reals and  $B$  be the reals. Their intersection is the positive reals, which is uncountably infinite.

**PART B (Graded by Cecilia and Charles)**

**PROBLEM 4** (each part 1 points, suggested length of 1/2 page)

Translate the following languages over  $\Sigma = \{a, b\}$  from English description to regular expressions, or vice versa.

- (A)  $L = \{w \in \Sigma^* : \text{the second letter of } w \text{ is the same as the last letter, } |w| \geq 2\}$
- (B)  $L = \{w \in \Sigma^* : \text{every } a \text{ is followed by an odd number of } b\text{'s}\}$
- (C)  $(a(a \cup b)^*b) \cup (b(a \cup b)^*a)$
- (D)  $L = \{w \in \Sigma^* : w \text{ has at least two } b\text{'s and at most one } a\}$
- (E)  $L = \{w \in \Sigma^* : \text{every } b \text{ in } w \text{ has a } b \text{ directly before it}\}$
- (F)  $(b^* \cup ba)^*$

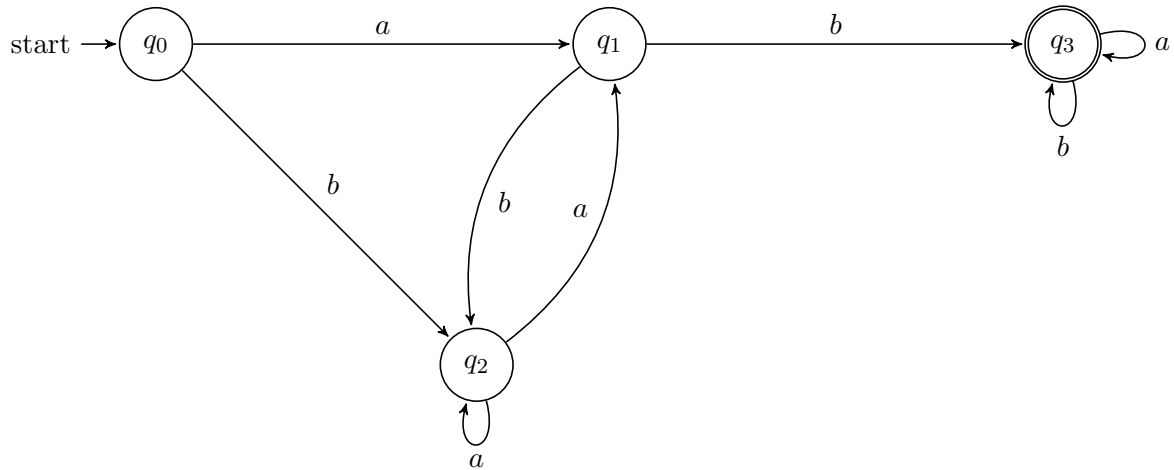
**Solution.**

- (A)  $\Sigma a(\Sigma^*)a \cup \Sigma b(\Sigma^*)b \cup \Sigma \Sigma$
- (B)  $b^*(ab(bb)^*)^*$
- (C)  $L = \{w \in \Sigma^* : \text{start and end with different letters, } |w| \geq 2\}$
- (D)  $bb(b^*)a \cup b(b^*)ab(b^*) \cup abb(b^*) \cup bb(b^*)$
- (E)  $a^*$
- (F)  $L = \{w \in \Sigma^* : \text{every } a \text{ has at least one } b \text{ before it}\}$

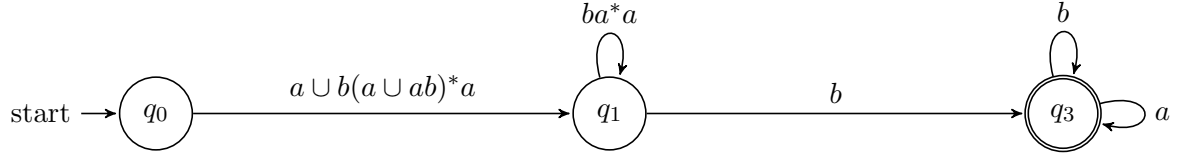
PROBLEM 5 (9 points, suggested length of 1 page)

Using the formal construction from the lecture notes or Sipser, convert the shown NFA to a regular expression. Please remove only one state per step. Show all steps or give explanation for steps not shown.

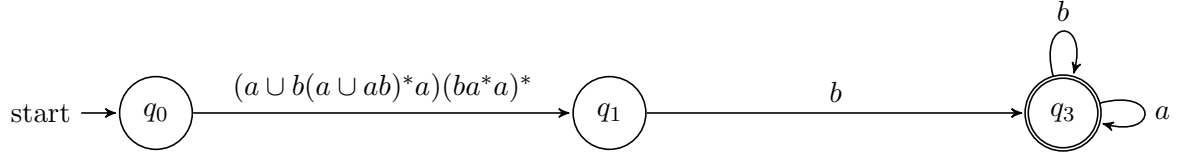
Note: There is no need to add an additional final state and add epsilon transitions to that state. If you are texing your solutions, you can just copy paste the NFA from the problem statement below and modify it. You can also use <http://madebyevan.com/fsm/>.



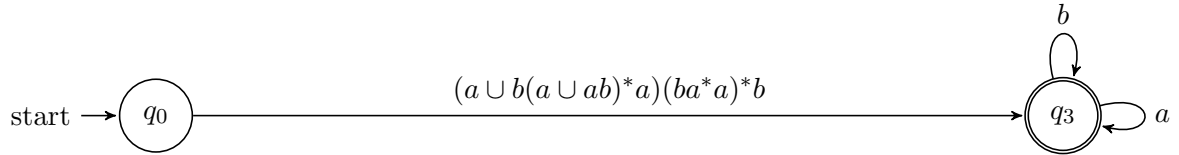
**Solution.**



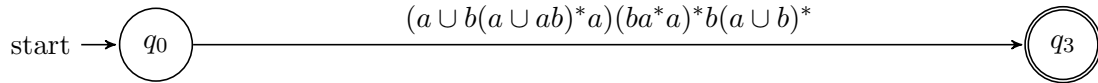
We try to remove  $q_2$  here; note that every path from  $q_0$  to  $q_1$  through  $q_2$  starts with an  $b$  and ends with an  $a$ , and intermediate steps from  $q_2$  to itself are either  $a$  or  $ab$ . Thus, any route that goes from  $q_0$  to  $q_1$  through  $q_2$  takes the form  $b(a \cup ab)^*a$ ; join this with the other paths (just  $a$ ) and we get all paths from  $q_0$  to  $q_1$ . Now, we also account for paths from  $q_1$  to itself through  $q_2$ ; these loops are of the form  $ba^*a$ .



Now we combine the loop into the left path.



Now remove  $q_1$ ; just join  $b$  with the previous path to create the path from the start to final state.



The last step is easy; we just add an arbitrary string to the end of our expression to remove the self-loops from  $q_3$ . The result is  $(a \cup b(a \cup ab)^*a)(ba^*a)^*b(a \cup b)^*$ . This can be simplified as follows:

$$(a \cup b(a \cup ab)^*a)(ba^*a)^*b(a \cup b)^* = (a \cup (ba^*a)^*)(ba^*a)^*b(a \cup b)^* \quad (1)$$

$$= (a(ba^*a)^* \cup (ba^*a)^*)b(a \cup b)^* \quad (2)$$

$$= (a \cup \varepsilon)(ba^*a)^*b(a \cup b)^* \quad (3)$$