

Dustin Newman

CS181, Disc. 1C

Campbell, Wong

# Assignment One

---

## 0

---

Sipser starts his numbering system for chapters from zero. Sections within chapters are numbered `<chapter number>.<section number>`. Note that the chapter numbers start from zero, but all other numbering systems start at one. So, there is no section `0.0` or exercise `1.0`. Exercises, which test definitions and concepts, are numbered as `<chapter number>.<exercise number>`. Problems, which test creativity and deeper understandings, follow exercises and use the same numbering system, starting consecutively after exercises (e.g. if the last exercise was `1.30`, then the first problem will be `1.31`). Figures, examples, and theorems all comprise one sequence within their chapters. So, if there is a figure `1.20`, we know that that figure is in chapter one and the next example, theorem, or figure will be `1.21`, regardless of which one it is.

If  $G$  is a connected acyclic (undirected) graph, then adding exactly one edge will result in graph  $G'$  which contains a cycle. Since  $G$  is connected, there exists a path  $P$  between any two nodes  $u$  and  $v$ . If we add the edge  $(v, u)$  to  $G$  to create  $G'$ , we have now created a cycle  $\{P, (v, u)\}$  which - by definition - forms a path which starts and ends with  $u$ . This same argument applies to any node  $w$  which lies along  $P$ . If we separate  $P$  into two disjoint sets  $P_u$  and  $P_v$  where  $P_u$  is the path from  $u$  to  $w$  and  $P_v$  is the path from  $w$  to  $v$ , then we cannot create any edges  $(u, w)$  or  $(w, v)$  because that would create cycles  $\{P_u, (w, u)\}$  or  $\{P_v, (v, w)\}$ , respectively. Therefore, for any path  $P$  between any two nodes  $u$  and  $v$  (and there must exist a  $P$  for **any** two  $u$  and  $v$  since  $G$  is connected), we cannot add an edge without creating a cycle.

To prove by contradiction that  $G'$  (the resulting graph after one edge addition) must always have *exactly* one cycle, suppose that we  $G'$  has two cycles. This either means that  $G$  had one cycle and the addition of our edge created the second in  $G'$  or that  $G$  had no cycles and the addition of our edge created both cycles. The first one derives a contradiction as  $G$  is acyclic. Thus, our edge  $e$  must have created two cycles. To have created two cycles from one edge, we had to have two paths  $q = u \rightarrow v$  and  $r = x \rightarrow y$  where  $u \neq x$ . After  $e$ , we now have  $\{q, (v, u)\}$  and  $\{r, (y, x)\}$ . Thus,  $e$  must have both  $u$  and  $x$  in its second position or else  $u = x$ . The former is a contradiction of the definition of an edge which is a relationship between **two** nodes, not three. The latter is a contradiction of our earlier statement  $u \neq x$ . By contradiction,  $G'$  must have exactly one cycle.

## 2

---

**a**

$$\{(0, x, 0), (0, x, 1), (0, y, 0), (0, y, 1), (0, z, 0), (0, z, 1), \\ (1, x, 0), (1, x, 1), (1, y, 0), (1, y, 1), (1, z, 0), (1, z, 1)\}$$

**b**

$$\{(0, x, 0), (0, y, 0), (0, z, 0), (1, x, 0), (1, y, 0), (1, z, 0), \\ (0, x, 1), (0, y, 1), (0, z, 1), (1, x, 1), (1, y, 1), (1, z, 1)\}$$

**c**

$$\{(0, x, 0), (0, x, 1), (0, y, 0), (0, y, 1), (0, z, 0), (0, z, 1) \\ (1, x, 0), (1, x, 1), (1, y, 0), (1, y, 1), (1, z, 0), (1, z, 1)\}$$

**d**

$$|B \times X| = |B| \times |X| = 6 \\ |\mathcal{P}(B \times X)| = 2^6 = 64$$

### 3

---

**a**

$$\begin{aligned} L_3 \circ \{a, c, aa\} &= \{aa, a, ad\} \circ \{a, c, aa\} \\ &= \{aaa, aac, aaaa, aa, ac, aaa, ada, adc, adaa\} \end{aligned}$$

**b**

$$\begin{aligned} L_3^+ \circ \{\} &= \{xy \mid x \in L_3^+ \wedge y \in \{\}\} \\ &= \{\} \end{aligned}$$

The concatenation of the empty set is always the empty set as there exists no such  $y$  within the empty set to satisfy the AND within the set definition of the concatenation result (i.e.  $\neg \exists y, y \in \{\}$ ).

**c**

$$\begin{aligned} \{\epsilon\} \times L_3 &= \{\epsilon\} \times \{aa, a, ad\} \\ &= \{(\epsilon, aa), (\epsilon, a), (\epsilon, ad)\} \end{aligned}$$

**d**

$$\{\} \times L_3^* = \{\}$$

Similar to the process in (b) above, there exists no element within the empty set that can be formed into a tuple with  $L_3^*$  and thus the result is also the empty set.

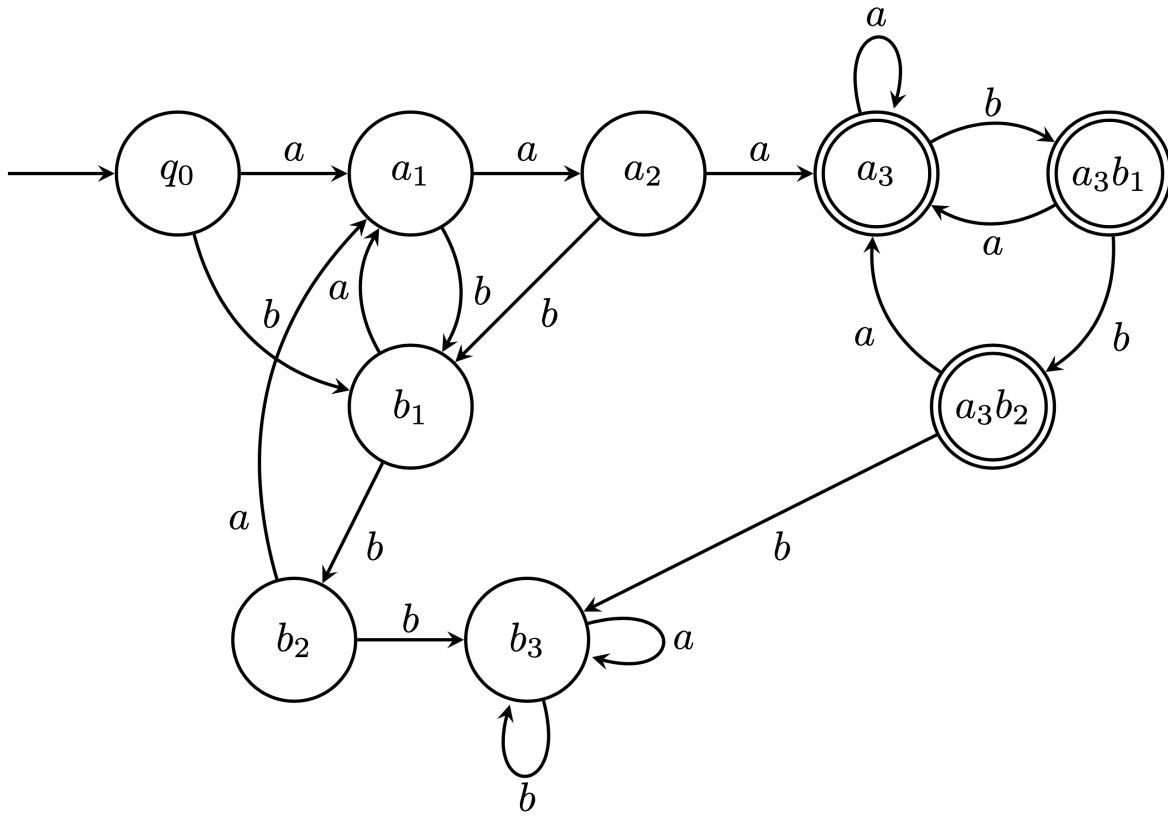
**e**

$$\begin{aligned} \{\epsilon\} \circ \{L_3^+\} &= \{\epsilon x \mid x \in L_3^+\} \\ &= \{x \mid x \in L_3^+\} \\ &= L_3^+ \end{aligned}$$

The concatenation of the set containing the empty string with the  $\Sigma^+$  of  $L_3$  is  $L_3^+$ . This is because the concatenation of the two sets is the concatenation of the empty string with each string in  $L_3^+$ .

However, since  $\epsilon x = x$  (i.e. the empty string is the identity operation), the resulting set is just  $L_3^+$ .

Because it is  $\Sigma^+$  and not  $\Sigma^*$ , **no**, the result does **not contain**  $\epsilon$ .



The requirement of consecutive a's and b's makes the construction of the DFA slightly simpler. My DFA works informally by checking for "b" at every step, as three consecutive b's will disqualify the string. The only time that having a "b" will not move the machine back to the  $b_1$  state is when we already have three consecutive a's, in which case we must remain in an accepting state until the third occurrence of "b." If we have encountered one or two consecutive b's (i.e. we are in  $b_1$  or  $b_2$ ), then we can still accept the string if we encounter an "a" and thus move to  $a_1$ . However, once we have encountered three consecutive b's, we have no choice but to move to  $b_3$ , where we never return. The requirement that we must have three consecutive a's means that our initial state is rejecting.

My DFA is defined formally below:

$$\begin{aligned}
 DFA &= (Q, \Sigma, \delta, q_0, F) \\
 Q &= \{q_0, b_1, b_2, b_3, a_1, a_2, a_3, a_3b_1, a_3b_2\} \\
 \Sigma &= \{a, b\} \\
 q_0 &= q_0 \\
 F &= \{a_3, a_3b_1, a_3b_2\}
 \end{aligned}$$

$\delta$  is defined as:

	<b>a</b>	<b>b</b>
$q_0$	$a_1$	$b_1$
$b_1$	$a_1$	$b_2$
$b_2$	$a_1$	$b_3$
$b_3$	$b_3$	$b_3$
$a_1$	$a_2$	$b_1$
$a_2$	$a_3$	$b_1$
$a_3$	$a_3$	$a_3 b_1$
$a_3 b_1$	$a_3$	$a_3 b_2$
$a_3 b_2$	$a_3$	$b_3$

# 5

---

## a

$$S_1 = \underbrace{cccc}_{x^4} \underbrace{dece}_{y} \underbrace{c}_{x}$$

$$S_2 = \underbrace{e}_{y} \underbrace{eeee}_{x^4 = x \circ x^3}$$

$S_1$  is an example string of  $L_5$  as it contains the substring  $xxxx$  i.e.  $x^4$  where  $x = c$ .  $S_1$  is an example of a string where the substring  $x^4$  is not continuous with the substring  $xy$ . Also note that in  $S_1$ ,  $y = dece \in \Sigma^+$  and of course  $S_1 = w \in \Sigma^+$  as well.

$S_2$  is an example string of  $L_5$  where the substring  $x^4$  is also continuous with the substring  $xy$ . The definition of  $L_5$  does not specify that the two substrings must be non-overlapping, so  $S_2 \in L_5$ . For  $S_2$ , I noted this fact by stating that  $x^4 = x \circ x^3$ , that the repetition of four  $x$ 's is equal to one  $x$  concatenated with the repetition of three  $x$ 's. This way, we can see the  $yx$  substring, while also recognizing the  $x^4$  substring.

## b

$$S_3 = \underbrace{c}_{x} \underbrace{e}_{y} \underbrace{ccc}_{x^3}$$

$$S_4 = \underbrace{eeee}_{x^4}$$

$S_3$  is not an example string of  $L_5$  because substrings must be continuous and both conditions of the set definition must be satisfied. So, even though there is a  $yx$  substring and four occurrences of  $x$ , the lack of continuity disqualifies  $S_3$  from being in  $L_5$ .

$S_4$  is not an example string of  $L_5$  because  $y \in \Sigma^+$  and thus it cannot be the empty string  $\epsilon$ . If instead  $y \in \Sigma^*$  (Kleene-star), then  $y$  could be  $\epsilon$  and  $S_4$  would be a valid string since  $eeee = \epsilon \circ eeee = y \circ x \equiv yx$ .

## c

$L_5$  is all the strings containing "c", "e", or "d" which are longer than four characters *and* have four identical contiguous characters.

## 6

---

$$L = \{w \mid w \text{ contains a multiple of 3 a's and an even number of b's}\}$$

This DFA recognizes the language  $L$  which contains a multiple of three amount of a's (i.e. if  $L(a)$  is the number of occurrences of "a" in the string  $w$ , then  $L(a) \bmod 3 = 0$ ) and an even number of b's (i.e.  $L(b) \bmod 2 = 0$ ). It also recognizes the empty string  $\epsilon$  as it also satisfies these conditions (e.g.  $0 \bmod 3 = 0$ ).