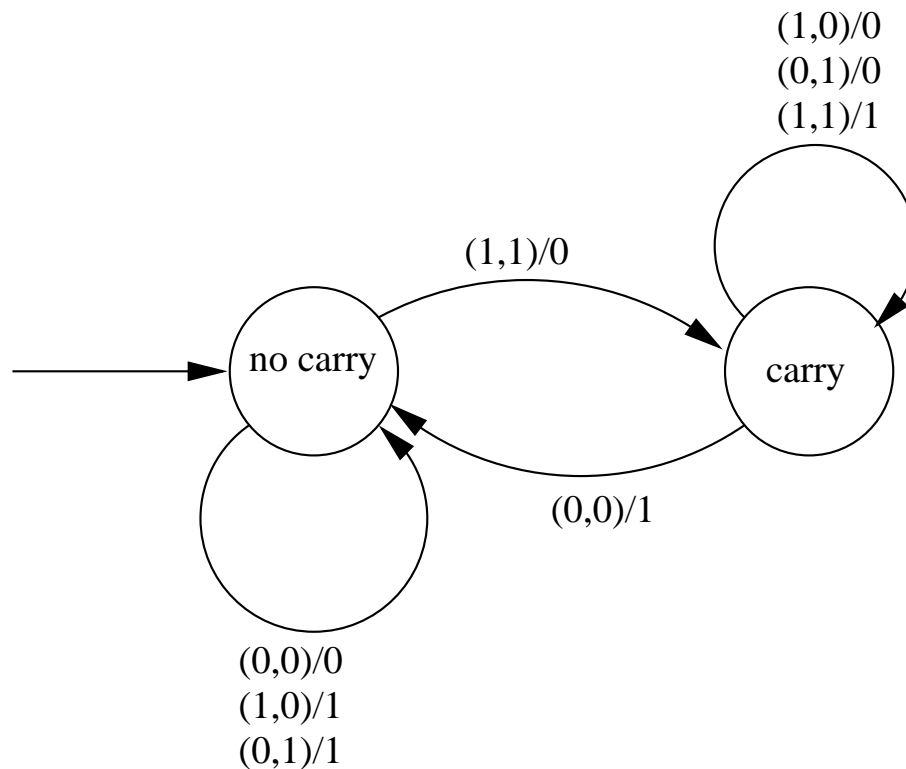


## CS20a HW2 Solutions

**Exercise 1.** The mealy machine is:



**Exercise 2.**

To prove that a language is not regular, we need to apply the first pumping lemma. Let us start by defining the first pumping property of length  $k$ :

*Let  $k \in \mathbb{N}$  and  $L$  be a language. Then a string  $w \in L$  has the first pumping property of length  $k$  iff  $\exists x, y, z$  (not necessarily in  $L$ ) such that  $w = xyz$ ,  $|xy| \leq k$ ,  $|y| \geq 1$  and  $xy^n z \in L$  for all  $n \geq 0$ .*

Using this definition, we can state the first pumping lemma as:

*Let  $L$  be a regular language. Then for some  $k \in \mathbb{N}$ , every  $w \in L$  with  $|w| \geq k$  has the first pumping property of length  $k$ .*

We can write the equivalent contrapositive form as:

*If for all  $k \in \mathbb{N}$ ,  $\exists w \in L$  that fails the pumping property of length  $k$ , then  $L$  is not regular.*

-

$\{0^m 1^n 0^{m+n} \mid n \geq 1 \wedge m \geq 1\}$  is not regular. For any  $k$  and  $|w| \geq k$  we split  $w \in L$  into  $xyz$  s.t.  $x = \epsilon$ ,  $y = 0^m$ , and  $z = 1^n 0^{m+n}$ . This clearly satisfies  $|xy| \leq k$ ,  $|y| \geq 1$ . However, if we "pump"  $y$  (e.g.  $xy^2z$ ), then  $w = xy^2z \notin L$  since the number of zeros in  $z$  will be less than what it should

be. Therefore,  $L$  fails the contrapositive of the first pumping lemma and we conclude that it is not regular.

The set of all strings that do not have 3 consecutive 0's is regular. If we let  $a = \Sigma - \{0\}$ , then the regular expression for the language is:

$$(0 + 00 + a^*) + (a(0 + 00)a^*) + (a^*(0 + 00)a)^*$$

$\{xwx^R \mid x, w \in (0 + 1)^+\}$  is regular. Because we can make  $w$  arbitrarily long, we can WLOG set it equal to all the letters in  $xwx^R$  except for the first and last. Thus the language is equivalent to  $\{0w0 + 1w1 \mid w \in (0 + 1)^+\}$  which is the language of all strings of length 3 or greater whose first and last letters are the same. A regular expression for this language is:

$$0(0 + 1)^+0 + 1(0 + 1)^+1$$

$\{xx^Rw \mid x, w \in (0 + 1)^+\}$  is not regular. To prove we split  $w \in L, |w| \geq k$  for any  $k \in \mathbb{N}$  into  $xyz$  such that  $|xy| \leq k$ , and  $|y| \geq 1$ . We let  $|xy| = 2$  and  $|y| = 1$ . Thus,  $y$  must be a letter in either  $x$  or  $x^R$ . If we now pump  $y$  to give  $xy^nz$ , then clearly the substring  $xx^R$  becomes unbalanced for any  $n \neq 2i, i \in \mathbb{N}$  if  $y$  is the first letter of  $x^R$  or the last of  $x$  and for  $n \neq 1$  if  $y$  is not on the boundary. Thus the contrapositive of the first pumping lemma is not satisfied and  $L$  is not regular.

### Exercise 3.

If  $L$  is finite then  $L^*$  must be regular since we can construct a regular expression to match it. To do so, just build a RE for each element of  $L$  and take the union of their Kleene closures.

If  $L$  is infinite, then our task is harder. We begin by noting that, since the alphabet of  $L$  has only one letter, there is an injection from elements of  $L^*$  to  $\mathbb{N}$ . Let  $M = \{|x|, x \in L\}$ , and let  $M^* = \{|x|, x \in L^*\}$ . Concatenation of two strings in  $L$  corresponds to addition of their images in  $M$ , and thus  $M^*$  is the set of all integers  $m$  that can be written as:

$$m = \sum_{i=1}^{\infty} c_i m_i, c_i \in \mathbb{N}, m_i \in M$$

Assuming the greatest common divisor (gcd)  $g$  of  $M$  is well-defined, it is clear that all elements of  $M^*$  are multiples of the gcd. Also, any multiple of the gcd is a linear combination of numbers in  $M$  with (possibly negative) integer coefficients. We would like to prove that for large enough multiples of the gcd we can always find *non-negative* coefficients. Doing so will tell us that for some integer  $f$ , we can write a regular expression for  $L^*$  as  $0^f(0^g)^*$  unioned with a finite number of strings of length less than  $f$ .

First we prove that the GCD of an infinite set always exists: Let  $S = \{s_1, s_2, s_3, \dots\}$  and consider a finite  $S' \subseteq S = \{s'_1, s'_2, s'_3, \dots, s'_k\}$ . Let  $g = \text{GCD}(S')$ . If we start increasing the size of  $S'$ , then either the  $g$  will remain the same or it will go down. Since  $g$  is bounded below by 1, the GCD of an infinite set always exists.

Let  $M = \{m_1, m_2, \dots\}$  with  $m_i < m_j$  whenever  $i < j$ . Next we need to find  $f$ , such that

$$\forall k \in \mathbb{N}, f + kg = \sum_{i=1}^n c_i m_i$$

, with all  $c_i \geq 0$ .

Let  $S = \{m_1, m_2, \dots, m_n\}$  be a subset of  $M$  such that  $\text{GCD}(S) = \text{GCD}(M)$ , and let  $g$  be that GCD. We know that  $g = \sum_{i=1}^n c_i m_i$ ,  $c_i \in \mathbb{Z}$ . Let  $f = m_1 \sum_{i=1}^n |c_i| m_i$ . Clearly all coefficients can be non-negative in  $f + kg$  if  $k < m_1$ , since in this case  $\forall i, m_1 |c_i| + kc_i > 0$ . Furthermore, every number  $f + kg$  with  $k$  any element in  $\mathbb{N}$  can be written as  $f + kg = (f + (k - pm_1)g) + pm_1g$ , where  $k - pm_1 < m_1$  and  $p \in \mathbb{N}$ . We just showed that the first term can be written as a non-negative linear combination of elements in  $S$ , and the second term is clearly a non-negative multiple of  $a_1$ , so we have proven that  $f$  satisfies our requirement, and that every multiple of  $g$  greater than  $f$  is in  $M^*$ . Consequently, we have shown that  $L^*$  is regular.

#### Exercise 4.

Let  $M = (Q, \Sigma, \delta, q_0, F)$  be a DFA that implements  $L$ .

#### HALF(L)

Construct a new machine  $M'$  for  $HALF(L)$ . Essentially, within  $HALF(L)$ , we record the state that the input string leads us to in  $|w|$  moves, as well as the set of states that can be reached by taking  $|w|$  backward moves from any final state. Given a set of states  $S$ , the function  $B(S)$  computes the states that can move to any state in  $S$  in one move.

Let  $M' = (Q \times 2^Q, \Sigma, \delta', [q_0, F], F')$ , where

$$\begin{aligned} B(S) &= \{q \mid \exists a \in \Sigma, t \in S. \delta(q, a) = t\} \\ F' &= \{[q, S] \mid q \in S\} \\ \delta'([q, S], a) &= [\delta(q, a), B(S)] \end{aligned}$$

#### SQUARE(L)

#### LOG(L)

For the log machine, we produce a similar construction, but we have to keep track of more states. Given a DFA  $M = (Q, \Sigma, \delta, s, F)$  that implements  $L$ , let  $Q = \{q_1, \dots, q_n\}$ .

We construct  $M' = (Q \times (2^Q)^n, \Sigma, \delta', s', F')$ , where

$$\begin{aligned} B(S) &= \{q \mid \exists a \in \Sigma. \delta(q, a) \in S\} \\ s' &= [s, B(\{q_1\}), \dots, B(\{q_n\})] \\ F' &= \{[q, S_1, \dots, S_n] \mid \exists i. q_i \in F \wedge q \in S_i\} \\ \delta'([q, S_1, \dots, S_n], a) &= [\delta(q, a), \bigcup_{q_i \in S_1} S_i, \dots, \bigcup_{q_i \in S_n} S_n] \end{aligned}$$

We claim that  $L(M') = LOG(L)$ . Next, we prove the following by induction on  $k$ :

$$\delta'([s, B(\{q_1\}), \dots, B(\{q_n\})], x) = [\delta(s, x), B^{2^k}(\{q_1\}), \dots, B^{2^k}(\{q_n\})]$$

**Base** This is clearly true for  $k = 0$ .

**Step** Next, assume the equation is true for  $k$ , and show it is true for  $k + 1$ .

$$\begin{aligned}
& \delta'([s, B(\{q_1\}), \dots, B(\{q_n\})], xa) \\
= & \delta'(\delta'([s, B(\{q_1\}), \dots, B(\{q_n\})], x), a) \\
= & \delta'([\delta(s, x), B^{2^k}(\{q_1\}), \dots, B^{2^k}(\{q_n\})], a)
\end{aligned}$$

Now we have to show that for  $j \in \{1, \dots, n\}$ ,

$$A = \bigcup_{q_i \in B^{2^k}(\{q_j\})} B^{2^k}(\{q_i\}) = B^{2^{k+1}}(\{q_j\})$$

Well, if  $q \in A$ , then  $\exists i$  with  $q_i \in S_j$  and  $q \in B^{2^k}(\{q_i\})$ . So,  $\exists \alpha, \beta \in \Sigma^*$  with  $|\alpha| = |\beta| = 2^k$ ,  $\delta(q, \alpha) \in \{q_i\}$ , and  $\delta(q_i, \beta) \in \{q_j\}$ , Hence  $\delta(q, \alpha\beta) = q_j$ ,

For the reverse direction, if  $q \in B^{2^{k+1}}(\{q_j\})$ , then  $\exists \varphi$  with  $|\varphi| = 2^{k+1}$  and  $\delta(q, \varphi) = q_j$ , Let  $\alpha, \beta$  be of length  $2^k$  where  $\alpha\beta = \varphi$ . Then  $q_i = \delta(q, \alpha) \in B^{2^k}(\{q_j\})$ , So  $B^{2^k}(\{q_i\}) \subset A$ .

Finally, suppose  $x \in LOG(L)$ , and choose  $y$  so that  $xy \in L$ . Then  $\delta(s, xy) = q_f$  for some  $q_f \in F$ , and  $\delta(s, x) \in B^{2^{|x|}}(\{q_f\})$  when  $M'$  accepts  $x$ . Conversely, the  $x \in L(M')$ , then  $\delta(s, x) \in B^{2^{|x|}}(\{q_f\})$  for some  $q_f \in F$ , and so  $y$  exists.

### Exercise 5.

We know that:  $xRy \Rightarrow wxzRwyz$  for all  $w$  and  $z$ . Define  $[x] = \{y \mid xRy\}$ . If  $R$  is a congruence relation of a finite index, then there are a finite number of congruence classes  $[x_1], [x_2], \dots, [x_n]$ . If we let  $S = [x_1] \cup [x_2] \cup \dots \cup [x_k]$  where  $k \leq n$  (we can arbitrarily select which  $[x]$ 's are in the first  $k$ , then all we need to show is that  $S$  is regular  $\Leftrightarrow S = L(M_S)$  for some DFA  $M_S$ .

Define  $M_S$  as:

$$\begin{aligned}
Q &= \{[x] \mid x \in \Sigma^*\} \\
\Sigma &= \text{any finite alphabet} \\
\delta([x], a) &= [xa] \\
S &= \{[\epsilon]\} \\
F &= \{[x] \mid x \in S\}
\end{aligned}$$

Since  $xRy \Rightarrow \epsilon xaR\epsilon ya \Rightarrow [xa] = [ya] \Rightarrow \delta(x, a) = \delta(y, a)$ ,  $\delta$  is well defined.

$\hat{\delta}([x], y) = [xy]$  can be proved by induction on the length of  $y$ :

**Base case:**  $\hat{\delta}([x], \epsilon) = [x] = [x\epsilon]$

**Inductive Hypothesis:**  $\hat{\delta}([x], y) = [xy]$

**Inductive Step:**  $\hat{\delta}([x], ya) = \delta(\hat{\delta}([x], y), a) = \delta([xy], a) = [xya]$ .

Using the last property we can write:

$$x \in L(M_S) \Leftrightarrow \hat{\delta}([\epsilon], x) \in F \Leftrightarrow [\epsilon x] \in F \Leftrightarrow x \in S.$$

Since  $L(M_S)$  is regular, all  $x \in L(M_S)$  are regular so all  $x \in S$  are regular.