CSC236, Summer 2004, Assignment 5 — sample solutions

1. Prove or disprove the following claims, assuming R, S, and T are regular expressions.

   (a) If $L(R^*) = \text{Rev}(L(R^*))$ then $L(R) = \text{Rev}(L(R))$.

   CLAIM: It is false that if $L(R^*) = \text{Rev}(L(R^*))$ then $L(R) = \text{Rev}(L(R))$.

   PROOF: Let $R = (10 + 1 + 0)$. Then $L(1+0)^* \subseteq L(R)^*$, and $L(1+0)^*$ (the language of all binary strings) is equal to $\text{Rev}(L(1+0)^*)$ (the reverse of a binary string is a binary string). But $L(10+1+0) = \{10, 1, 0\}$ is not equal to $\text{Rev}(L(10+1+0)) = \{01, 1, 0\}$, since the first language contains 10, and the second does not. Thus the claim does not hold. QED.

   (b) If $L(R) = \text{Rev}(L(R))$ then $L(R^*) = \text{Rev}(L(R^*))$.

   CLAIM: Suppose $L(R) = \text{Rev}(L(R))$. Then for all $x \in L(R^*)$, $x \in \text{Rev}(L(R^*))$.

   PROOF: Suppose $x \in L(R^*) = L(R)^*$. Then (by the alternative characterization of $L(R)^*$), either $x = \epsilon$, or, for some $k > 0$, $x = x_1 \cdots x_k$, where $x_1, \ldots, x_k \in L(R)$. In the first case, $\epsilon = \text{Rev}(\epsilon)$, so $x = \epsilon \in \text{Rev}(L(R^*))$, as claimed. In the second case, $x = \text{Rev}(\text{Rev}(x_k) \cdots \text{Rev}(x_1))$ (by repeated application of Theorem 7.4, page 189 of the Course Notes), and $\text{Rev}(x_k), \ldots, \text{Rev}(x_1) \in L(R)$ by assumption , so $x \in \text{Rev}(L(R^*))$, as claimed. Since x was chosen arbitrarily, $L(R^*) \subseteq \text{Rev}(L(R^*))$.

   On the other hand, suppose x is an arbitrary element of $\text{Rev}(L(R^*))$. Then either $x = \text{Rev}(\epsilon) = \epsilon$ (so $x \in L(R^*)$), or $x = \text{Rev}(x_1 \cdots x_k)$, where $x_1, \ldots, x_k \in L(R)$. But (by repeated application of Theorem 7.4) $\text{Rev}(x_1 \cdots x_k) = \text{Rev}(x_k) \cdots \text{Rev}(x_1)$, and (by assumption that $L(R) = \text{Rev}(L(R))$) $\text{Rev}(x_k), \ldots, \text{Rev}(x_1) \in L(R)$, so $x \in L(R^*)$. Since x was chosen arbitrarily, this implies that $\text{Rev}(L(R^*)) \subseteq L(R^*)$.

   Since $L(R^*)$ and $\text{Rev}(L(R^*))$ include each other, they are equal. QED.

   (c) If $(RS)^* \equiv (R^*S^*)$ then $R \equiv S$.

   CLAIM: It is false that if $(RS)^* \equiv (R^*S^*)$ then $R \equiv S$.

   PROOF: Let $R = 1$ and $S = \epsilon$. Then $R \not\equiv S$, since $L(R) = \{1\}$ contains 1, and $L(S) = \{\epsilon\}$ does not, but

$$
\begin{aligned}
\text{[identity law]} \quad (RS)^* &= (1\epsilon)^* = 1^* \\
\text{[identity law]} \quad &= (1^*\epsilon) \\
\epsilon^* = \epsilon \quad &= (1^*\epsilon^*) = (R^*S^*)
\end{aligned}
$$

   This counter-example proves that the claim is false. QED.

   (d) If $R \equiv RR$ and $R \not\equiv \emptyset$, then $R \equiv R^*$.

   CLAIM: If $R \equiv RR$ and $R \not\equiv \emptyset$, then $R \equiv R^*$.

   PROOF: Since $R \neq \emptyset$, $\{|x| : x \in L(R)\}$ is a non-empty subset of $\mathbb{N}$, and so it has a least element. In other words, there is some $x' \in L(R)$ such that $\forall x \in L(R)$, $|x'| \leq |x|$. Since $L(R) = L(RR)$ we must have $x' = x_1x_2$, where $x_1, x_2 \in L(R)$, and by the choice of x', $|x_1|$, $|x_2| \geq |x'|$. But this means that

$$|x'| = |x_1| + |x_2| \geq |x'| + |x'| \Rightarrow 0 \geq |x'|.$$

Since $|x'|$ is a natural number, it must be 0, and $x' = \epsilon$, so $\epsilon \in L(R)$.

Now, let $L = L(R)$, and consider:

BASIS: $\epsilon \in L$ (just shown).

INDUCTIVE STEP: If $x \in L$ and $y \in L(R)$, then (since $L = L(R) = L(RR)$), $xy \in L$.

These two facts verify that L has an identical definition by structural induction to $L(R^*)$, so (since $L(R) = L$) $L(R) = L = L(R^*)$, in other words, $R \equiv R^*$, as wanted. QED.

2. Give a regular expressions that denotes L, and justify your answer.

(a) $L = \{x \in \{0,1\}^* : x \text{ contains at least four 0s}\}$.

SOLUTION: $L = L(1^*01^*01^*01^*0(0+1)^*)$. Indicate the first four 0s. The first one is preceded by a prefix in 1^* (zero free), the first and second are separated by a substring in 1^*, the second and third are separated by a substring in 1^*, and third and fourth are separated by a substring in 1^*, and the fourth zero is followed by any arbitrary binary string.

(b) $L = \{x \in \{0,1\}^* : x \text{ contains at least two 0s and at most one 1}\}$

SOLUTION: $L = L(000^* + 1000^* + 000^*1 + 0^*0100^*)$. A string in L may have zero 1s and at least two 0s, or it may have a single 1 followed by two or more 0s, or it may have a single 1 preceded by two or more 0s, or it may have a single 1 with at least one 0 before and at least one 0 after it. The union of these possibilities is $L(000^* + 1000^* + 000^*1 + 0^*0100^*)$.

(c) $L = \{x \in \{0,1\}^* : x \text{ contains an odd number of 0s, or exactly two 1s}\}$

SOLUTION: $L = L(1^*01^*(01^*01^*)^* + 0^*10^*10^*)$. The term $1^*01^*(01^*01^*)^*$ denotes the set of strings whose prefix 1^*01^* contains a single 0, followed by zero or more 1s, followed by 0 or more strings that contain two 0s each, so $L(1^*01^*(01^*01^*)^*)$ is the language of strings that contain an odd number of 0s. The term $0^*10^*10^*$ denotes any string that contains two ones surrounded (and separated) by zero or more 0s, so $L(0^*10^*10^*)$ is the language of strings that contain exactly two 1s. Thus $L(1^*01^*(01^*01^*)^* + 0^*10^*10^*)$ denotes the union of the set of strings with an odd number of zeros with the set of strings with exactly two 1s, as wanted.

(d) $L = \{x \in \{0,1\}^* : x \text{ doesn't contain the substring 101}\}$

SOLUTION: $L = L(0^*(1 + 1000^*)^*10^* + 0^*)$. Any string that doesn't contain 101, but does contain at least one 1 can be expressed as the concatenation:

- a prefix preceding the first 1 denoted by 0^*
- zero or more blocks starting with 1 and followed by either no 0s, or at least two 0s. These are denoted by $(1 + 1000^*)^*$.
- the final 1
- a suffix following the last 1, denoted by 0^*

The only other possibility for a string that doesn't contains any 1s. The expression $0^*(1+1000^*)^* + 0^*$ denotes the union of these two possibilities.

(e) $L = \{x \in \{0,1\}^* : x \text{ is neither 11 nor 111}\}$

SOLUTION: $L = L(1 + (1^*01^*)^* + 11111^*)$. Consider the following cases[1]

- Any binary string that is not comprised of one or more 1s is a member of $L((1^*01^*)^*)$, since it can be decomposed into the prefix before the first 0, the substring starting with the ith 0 until just before the $(i+1)$th 0, and so on.
- The binary string comprised of one or more 1s that aren't either 11 or 111 are either in $L(1)$ or $L(11111^*)$, since they have either one character or more than three characters.

The solution is the union of these cases, so $L \subseteq L(1 + (1^*01^*)^* + 11111^*)$. On the other hand, it is clear by inspection that neither 11 nor 111 match the regular expression, so the reverse inclusion is also true.

[1]Thanks to Carrie Chan for this solution. It is shorter (and nicer) than mine.

3. For each of the following languages, L, construct a DFSA that accepts L and a regular expression that denotes L. Prove your automata and regular expressions are correct.

(a) $L = \{x \in \{0,1\}^* : |x| > 2 \text{ or } x \text{ contains suffix } 1\}$

CLAIM 3(A)1: $L = L((0+1)^*1 + (0+1)(0+1)(0+1)(0+1)^*)$.

PROOF: Let x be an arbitrary string in L. There are two cases to consider

CASE 1, $|x| > 2$: If $|x| > 2$, then x can be expressed as the concatenation uv, where $|u| = 3$ and v is any binary string. Thus $u \in L((0+1)(0+1)(0+1))$, and $v \in (0+1)^*$, so $x = uv \in L((0+1)(0+1)(0+1)(0+1)^*)$.

CASE 2, x CONTAINS SUFFIX 1: If x contains suffix 1, then x can be expressed as the concatenation uv, where u is any binary string and $v = 1$, so $x \in L((0+1)^*1)$.

These two cases exhaust the possibilities, so x is in their union, that is $x \in L((0+1)^*1 + (0+1)(0+1)(0+1)(0+1)^*)$. Since x is an arbitrary element of L, this shows that $L \subseteq L((0+1)^*1 + (0+1)(0+1)(0+1)(0+1)^*)$.

On the other hand, let x be an arbitrary string in $L((0+1)^*1 + (0+1)(0+1)(0+1)(0+1)^*)$. There are two cases to consider:

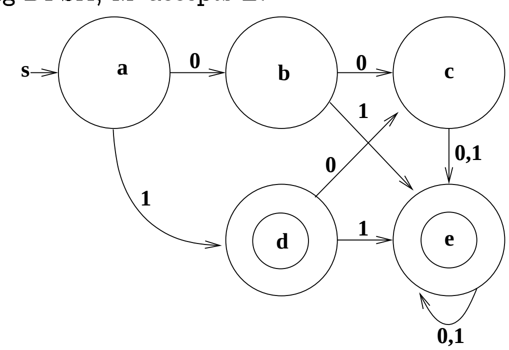
CASE 1, $x \in L((0+1)^*1)$: If $x \in L((0+1)^*1)$, then x can be expressed as the concatenation uv, where $u \in L((0+1)^*)$ is an arbitrary binary string, and $v \in L(1)$. In this case x has the suffix 1, so $x \in L$.

CASE 2, $x \in L((0+1)(0+1)(0+1)(0+1)^*)$: In this case, x can be expressed as the concatenation uv, where $v \in L((0+1)^*)$ is an arbitrary binary string, and u contains exactly three characters. In this case $|x| > 2$, so $x \in L$.

The two cases exhaust the possibilities, so $x \in L$. Since x was chosen as an arbitrary element in $L((0+1)^*1 + (0+1)(0+1)(0+1)(0+1)^*)$, This means that $L((0+1)^*1 + (0+1)(0+1)(0+1)(0+1)^*) \subseteq L$.

This shows that the two languages contain each other, and thus are equal. QED.

CLAIM 3(A)2: The following DFSA, M accepts L:



Before proving Claim 3(a)2, I need to prove the following state invariant:

CLAIM 3(A)2I: Define $P(x)$ by

$$P(x): \quad \delta^*(s,x) = \begin{cases} a, & \text{if } x = \epsilon \\ b, & \text{if } x = 0 \\ c, & \text{if } x \in \{00, 10\} \\ d, & \text{if } x = 1 \\ e, & \text{if } x \in \{11, 01\} \text{ or } |x| > 2 \end{cases}$$

PROOF (INDUCTION ON $|x|$): Suppose $|x| = 0$, that is, $x = \epsilon$. Then $\delta^*(s,x) = s = a$, and $P(\epsilon)$ claims that $x = \epsilon$, which is certainly true. Thus the base case ($P(\epsilon)$) holds.

INDUCTION STEP: For some arbitrary non-empty string x, assume that $P(y)$ holds for every y such that $|y| = |x| - 1$. There are two possibilities to consider:

CASE $x = y0$ FOR SOME $y \in \{0,1\}^*$: Since you've assumed $P(y)$, you can substitute it into the state invariant:

$$\delta^*(s, y0) = \begin{cases} \delta(a, 0), & \text{if } y = \epsilon \\ \delta(b, 0), & \text{if } y = 0 \\ \delta(c, 0), & \text{if } y \in \{00, 10\} \\ \delta(d, 0), & \text{if } y = 1 \\ \delta(e, 0), & \text{if } y \in \{11, 01\} \text{ or } |y| > 2 \end{cases}$$

Now evaluate the transition function, and take into account that you have appended a 0:

$$\delta^*(s, x) = \begin{cases} b, & \text{if } x = 0 \\ c, & \text{if } x = 00 \\ e, & \text{if } x \in \{000, 100\} \\ c, & \text{if } y = 10 \\ e, & \text{if } x \in \{110, 010\} \text{ or } |x| > 3 \end{cases}$$

The two claims for state c combine to "if $x \in \{00, 10\}$ then $\delta^*(s, x) = c$." The two claims for state e, together with the fact that in Case 1 x ends in 0, combine to "if $|x| > 2$ then $\delta^*(s, x) = e$." The claim for b is identical to that in $P(x)$, and the claims for a and d hold vacuously (false antecedents). Thus $P(x)$ holds in the case where $x = y0$.

CASE $x = y1$ FOR SOME $y \in \{0,1\}^*$: You've already assumed $P(y)$, so substitute it into the state invariant:

$$\delta^*(s, y1) = \begin{cases} \delta(a, 1), & \text{if } y = \epsilon \\ \delta(b, 1), & \text{if } y = 0 \\ \delta(c, 1), & \text{if } y \in \{00, 10\} \\ \delta(d, 1), & \text{if } y = 1 \\ \delta(e, 1), & \text{if } y \in \{11, 01\} \text{ or } |y| > 2 \end{cases}$$

Now evaluate the transition function and take into account that you have appended a 1:

$$\delta^*(s, y1) = \begin{cases} d, & \text{if } x = 1 \\ e, & \text{if } x = 01 \\ e, & \text{if } x \in \{001, 101\} \\ e, & \text{if } x = 11 \\ e, & \text{if } x \in \{111, 011\} \text{ or } |x| > 3 \end{cases}$$

The claim about state d is identical to that in $P(x)$, and the claims about a, b, c hold vacuously (false antecedents). The claims about e, together with the fact that x ends in 1 in Case 2, combine to "if $x \in \{01, 11\}$ or $|x| > 2$, then $\delta^*(s, x) = e$. Thus $P(x)$ holds in the case where $x = y1$.

In either case $P(y)$ implies $P(x)$, as wanted.

I conclude that $P(x)$ is true for all $x \in \{0,1\}^*$. QED.

To prove Claim 3(a)2, first assume that x is an arbitrary string in L. If x has prefix 1, then by $P(x)$ either $\delta^*(s, x) = d$, or $\delta^*(s, x) = e$, both accepting states. If $|x| > 2$, then by $P(x)$ $\delta^*(s, x) = e$, and x is accepted. So $x \in L(M)$, and (since x was chosen to be an arbitrary string in L) this means that $L \subseteq L(M)$.

On the other hand, assume that x is an arbitrary string in $L(M)$, but not a string in L. Thus x does not end in 1, and has 2 or fewer digits, that is $x \in \{\epsilon, 0, 00, 10\}$. However, by $P(x)$

then $\delta^*(s,x) \in \{a,b,c\}$, contradicting the assumption that x is accepted by M. Thus the assumption that $x \notin L$ is false, and $x \in L$. Since x was chosen arbitrarily, $L(M) \subseteq L$.
Since L and $L(M)$ include each other, they are equal. QED.

(b) $L = \{x \in \{0,1\}^* : x \text{ contains substring } 11 \text{ and } x \text{ has an even number of 0s}\}$

CLAIM 3(B)1: $L = L(1^*(01^*01^*)^*111^*(01^*01^*)^* + 1^*01^*(01^*01^*)^*111^*01^*(01^*01^*)^*)$.

PROOF: Let x be an arbitrary element of L. Fix an instance of the substring 11 and there are two possibilities

CASE 1: There are an even number of 0s preceding the instance of 11, and hence an even number of 0s following it. Thus x can be expressed as the concatenation uvw, where $u, w \in L(1^*(01^*01^*)^*$ (proved in Course Notes), and $v \in L(11)$, so $x = uvw$ is a member of $L(1^*(01^*01^*)^*111^*(01^*01^*)^*$.

CASE 2: There are an odd number of 0s preceding the instance of 11, and hence an odd number of 0s following it. Thus x can be expressed as the concatenation $u_1u_2vw_1w_2$, where u_1 is the prefix of x up to and including the first 0, u_2 is the substring of x following the first 0 and preceding the instance of 11 (and hence containing an even number of 0s), v is 11, w_1 is the substring of x following the instance of 11 and including the next 0, and w_2 is the suffix of x following that 0 (and hence containing an even number of 0s. Hence $u_2, w_2 \in L(1^*(01^*01^*)^*$, u_1 and w_1 consist of zero or more 1s with a 0 suffix, and are in $L(1^*0)$, and $v = 11$. This means that $x = u_1u_2vw_1w_2 \in L(1^*01^*(01^*01^*)^*111^*01^*(01^*01^*)^*)$.

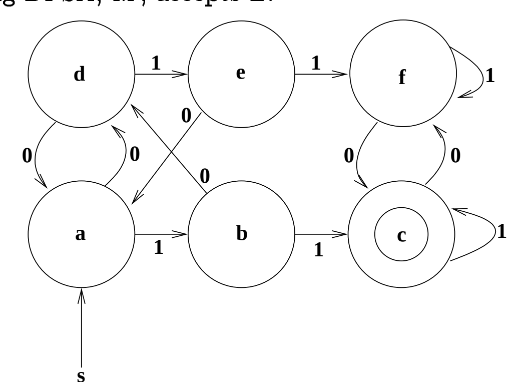
The two cases exhaust the possibilities, so $x \in$ of the union $L(1^*(01^*01^*)^*111^*(01^*01^*)^* + 1^*01^*(01^*01^*)^*111^*01^*(01^*01^*)^*)$. Since x was chosen as an arbitrary element of L, this shows that $L \subseteq L(1^*(01^*01^*)^*111^*(01^*01^*)^* + 1^*01^*(01^*01^*)^*111^*01^*(01^*01^*)^*)$.
On the other hand, suppose x is an arbitrary element of $L(1^*(01^*01^*)^*111^*(01^*01^*)^* + 1^*01^*(01^*01^*)^*111^*01^*(01^*01^*)^*)$. Then there are two possibilities:

CASE 1: x is in $L(1^*(01^*01^*)^*111^*(01^*01^*)^*$, so we can express x as uvw, where $u, w \in L(1^*(01^*01^*)^*)$ have an even number of 0s (proof in Course Notes), $v = 11$, so uvw has an even number of zeros and contains 11.

CASE 2: x is in $L(1^*01^*(01^*01^*)^*111^*01^*(01^*01^*)^*)$, so we can express as $u_1u_2vw_1w_2$, where $u_2, w_2 \in L(1^*(01^*01^*)^*)$ have an even number of 0s, u_1, w_1 have exactly one 0 each, and $v = 11$, so uvw has an even number of zeros and contains 11.

The two cases exhaust the possibilities, and in both cases $x \in L$. Since x was chosen to be an arbitrary element of $L(1^*(01^*01^*)^*111^*(01^*01^*)^* + 1^*01^*(01^*01^*)^*111^*01^*(01^*01^*)^*)$, this shows that $L(1^*(01^*01^*)^*111^*(01^*01^*)^* + 1^*01^*(01^*01^*)^*111^*01^*(01^*01^*)^*) \subseteq L$.
The two languages have been shown to contain each other, and are hence equal. QED.

CLAIM 3(B)2: The following DFSA, M, accepts L:



Before proving claim 3(b)2, I need to prove the following invariant:

CLAIM 3(B)2I: Define $P(x)$ as

$$P(x): \quad \delta^*(s,x) = \begin{cases} a, & \text{if } x \text{ doesn't contain 11, has an even number of 0s, and doesn't end in 1} \\ b, & \text{if } x \text{ doesn't contain 11, has an even number of 0s, and ends in 1} \\ c, & \text{if } x \text{ contains 11 and has an even number of 0s} \\ d, & \text{if } x \text{ doesn't contain 11, has an odd numb er of 0s, and doesn't end in 1} \\ e, & \text{if } x \text{doesn't contain 11, has an odd number of 0s, and ends in 1} \\ f, & \text{if } x \text{ contains 11 and has an odd number of 0s} \end{cases}$$

Then $P(x)$ is true for all $x \in \{0,1\}^*$.

PROOF (INDUCTION ON $|x|$): If $|x| = 0$, that is $x = \epsilon$, then $\delta^*(s,x) = a$, and $P(\epsilon)$ claims that ϵ doesn't contain 11 nor end in 1, which is certainly true. So $P(\epsilon)$ holds (basis).

INDUCTION STEP: Assume that $P(y)$ holds for all y such that $|y| = |x| - 1$, for some arbitrary x. There are two possibilities

CASE $x = y0$, FOR SOME $y \in \{0,1\}^*$: We have assumed $P(y)$, so we can substitute y into our invariant:

$$\delta^*(s,y0) = \begin{cases} \delta(a,0), & \text{if } y \text{ doesn't contain 11, has an even number of 0s, and doesn't end in 1} \\ \delta(b,0), & \text{if } y \text{ doesn't contain 11, has an even number of 0s, and ends in 1} \\ \delta(c,0), & \text{if } y \text{ contains 11 and has an even number of 0s} \\ \delta(d,0), & \text{if } y \text{ doesn't contain 11, has an odd numb er of 0s, and doesn't end in 1} \\ \delta(e,0), & \text{if } y \text{ doesn't contain 11, has an odd number of 0s, and ends in 1} \\ \delta(f,0), & \text{if } y \text{ contains 11 and has an odd number of 0s} \end{cases}$$

Now I evaluate the transition function, and toggle the parity of the number of 0s:

$$\delta^*(s,x) = \begin{cases} d, & \text{if } x \text{ doesn't contain 11, has an odd number of 0s, and doesn't end in 10} \\ d, & \text{if } x \text{ doesn't contain 11, has an odd number of 0s, and ends in 10} \\ f, & \text{if } x \text{ contains 11 and has an odd number of 0s, ends in 0} \\ a, & \text{if } x \text{ doesn't contain 11, has an even number of 0s, and doesn't end in 10} \\ a, & \text{if } x \text{ doesn't contain 11, has an even number of 0s, and ends in 10} \\ c, & \text{if } x \text{ contains 11 and has an even number of 0s, ends in 0} \end{cases}$$

Combining the two implications about state d with the fact that (for Case 1) x doesn't end in 1, yields "if x doesn't contain 11, has an odd number of 0s, and doesn't end in 1, then $\delta^*(s,x) = d$." Similarly, combining the two implications about state a yields "if x doesn't contain 11, has an even number of 0s, and doesn't end in 1, then $\delta^*(s,x) = a$." The implications about b and e are vacuously true (there are no 0-transitions into these states), and the implications about states c and f are what $P(x)$ claims. So $P(x)$ is true in this case.

CASE $x = y1$, FOR SOME $y \in \{0,1\}^*$: I have assumed $P(y)$, so I can substitute it into the invariant

$$\delta^*(s,y1) = \begin{cases} \delta(a,1), & \text{if } y \text{ doesn't contain 11, has an even number of 0s, and doesn't end in 1} \\ \delta(b,1), & \text{if } y \text{ doesn't contain 11, has an even number of 0s, and ends in 1} \\ \delta(c,1), & \text{if } y \text{ contains 11 and has an even number of 0s} \\ \delta(d,1), & \text{if } y \text{ doesn't contain 11, has an odd numb er of 0s, and doesn't end in 1} \\ \delta(e,1), & \text{if } y \text{doesn't contain 11, has an odd number of 0s, and ends in 1} \\ \delta(f,1), & \text{if } y \text{ contains 11 and has an odd number of 0s} \end{cases}$$

Now I evaluate the transition function, noting that the number of 0s is unchanged, and x ends in an extra 1:

$$\delta^*(s,x) = \begin{cases} b, & \text{if } x \text{ doesn't contain 11, has an even number of 0s, and ends in 1} \\ c, & \text{if } x \text{ has an even number of 0s, and ends in 11} \\ c, & \text{if } x \text{ contains 11 and has an even number of 0s, ends in 1} \\ e, & \text{if } x \text{ doesn't contain 11, has an odd number of 0s, ends in 1} \\ f, & \text{if } f \text{has an odd number of 0s, and ends in 11} \\ f, & \text{if } x \text{ contains 11 and has an odd number of 0s, and ends in 1} \end{cases}$$

The two implications about state c, together with the fact that in Case 2 x ends with a 1, combine to "if x contains 11 and an even number of zeros, then $\delta^*(s,x) = c$." The two implications about state f, combine to "if x contains 11 and an odd number of zeros, then $\delta^*(s,x) = f$." The claims about states b and e are verified, and the claims about a and d are vacuously true (there are no 1-transitions into those states). So $P(x)$ is true in this case as well.

In each case, $P(y)$ implies $P(x)$, as wanted.

I conclude that $P(x)$ is true for all $x \in \{0,1\}^*$. QED.

To prove Claim 3(b)2, let x be an arbitrary string in L. Then, by $P(x)$, $\delta^*(s,x) = c$, an accepting state, and my machine accepts x, so $x \in L(M)$. Since x was chosen arbitrarily, $L \subseteq L(M)$.

On the other hand, suppose $x \in L(M)$. Then (again by $P(x)$), if x were not in L, $\delta^*(s,x) \in \{a,b,d,e,f\}$, contradicting the assumption that $x \in L(M)$. Thus $x \in L$, and (since x was chosen arbitrarily) $L(M) \subseteq L$. By mutual inclusion, $L = L(M)$, as claimed.

(c) Let $(x)_2$ denote the value of x as a binary number.

$$L = \{x \in \{0,1\}^* : \text{ for some } n \in \mathbb{N}, (x)_2 = n \text{ and } \text{ for some } i,j \in \mathbb{N}, (n \operatorname{div} 2^i) \bmod 2^j = 5\}$$

Solution: Most of the work here is translating what L means. Using Proposition 1.7 (Division Algorithm), $n \operatorname{div} 2^i$ is defined as q_1 where

$$\begin{aligned} n &= q_1 2^i + r \quad (0 \leq r < 2^i) \\ \text{and} \quad q_1 &= q_2 2^j + 5 \quad\quad (0 \leq 5 < 2^j) \\ \text{so} \quad n &= (q_2 2^j + 5)2^i + r = q_2 2^{i+j} + 5 \times 2^i + r. \end{aligned}$$

This means that, in binary, n is the sum of a binary number ending with $i+j$ 0s, where $j \geq 3$ (since $5 < 2^j$), plus 101 followed by i 0s, plus the binary representation of r, which has i or fewer digits (since $r < 2^i$). These are exactly the binary numbers that contain the substring 101 (binary 5).

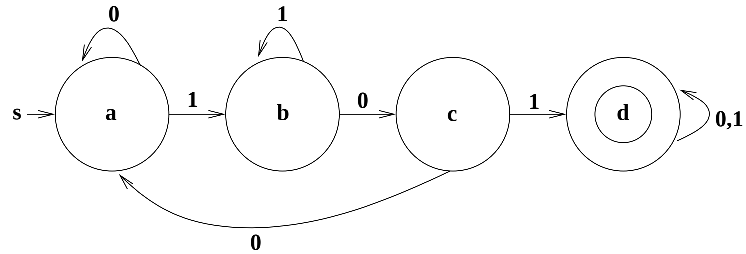
Claim 3(c)1: $L = L((0+1)^*101(0+1)^*)$.

Proof: Suppose x is an arbitrary string in L. Then (by the preceding discussion) x is a binary string that contains 101, and can be expressed as the concatenation uvw, where $u,w \in L(0+1)^*$ are arbitrary binary strings, and $v = 101$. Thus $x = uvw \in L((0+1)^*101(0+1)^*)$. Since x was chosen arbitrarily, $L \subseteq L((0+1)^*101(0+1)^*)$.

On the other hand, suppose x is an arbitrary string in $L((0+1)^*101(0+1)^*)$. Then x is the concatenation uvw, where $u,w \in L(0+1)^*$ and $v = 101$. In this case x contains the substring 101 (namely v), so $x \in L$. Since x was chosen arbitrarily, $L((0+1)^*101(0+1)^*) \subseteq L$.

We have shown that the two languages contain each other, hence they are equal. QED.

Claim 3(c)2: The following machine accepts L:



In order to prove this claim, I need to prove the following invariant:

CLAIM 3(C)2I: Let $P(x)$ be defined:

$$P(x) : \delta^*(s,x) = \begin{cases} a, & \text{if } x \text{ doesn't contain 101, and doesn't end in 1 or 10} \\ b, & \text{if } x \text{ doesn't contain 101 and ends in 1} \\ c, & \text{if } x \text{ doesn't contain 101 and ends in 10} \\ d, & \text{if } x \text{ contains substring 101} \end{cases}$$

Then $P(x)$ is true for all $x \in 0, 1^*$.

PROOF (INDUCTION ON $|x|$): Suppose $|x| = 0$, in other words, $x = \epsilon$. Then $\delta^*(s,x) = a$, and x doesn't contain 101 nor end in either 1 or 10. All the other branches of $P(x)$ have false antecedents, and thus hold vacuously, so $P(\epsilon)$ holds (base case).

INDUCTION STEP: Suppose $|x| > 0$, and assume $P(y)$ for all strings y with $|y| = |x| - 1$. There are two possibilities:

CASE $x = y0$, FOR SOME $y \in \{0,1\}^*$: By assumption, we have $P(y)$, so we can substitute y into our invariant:

$$\delta^*(s,y0) = \begin{cases} \delta^*(a,0), & \text{if } y \text{ doesn't contain 101, and doesn't end in 1 or 10} \\ \delta^*(b,0), & \text{if } y \text{ doesn't contain 101 and ends in 1} \\ \delta^*(c,0), & \text{if } y \text{ doesn't contain 101 and ends in 10} \\ \delta^*(d,0), & \text{if } y \text{ contains substring 101} \end{cases}$$

Now we evaluate the transition function to get:

$$\delta^*(s,x) = \begin{cases} a, & \text{if } x \text{ doesn't contain 101, and doesn't end in 10 or 100} \\ c, & \text{if } x \text{ doesn't contain 101 and ends in 10} \\ a, & \text{if } x \text{ doesn't contain 101 and ends in 100} \\ d, & \text{if } x \text{ contains substring 101 and ends in 0} \end{cases}$$

The two implications about state a, together with the fact that (in Case 1) x doesn't end in 1, combine into "if x doesn't contain 101 and doesn't end in 1 or 10, then $\delta^*(s,x) = a$." This verifies all of $P(x)$ except for strings that move the machine to state b. By construction, x doesn't end in 1, the implication "if x doesn't contain 101 and ends in 1, then $\delta^*(s,x) = b$" is vacuously true in this case. So $P(x)$ holds for the case where $x = y0$.

CASE $x = y1$ FOR SOME $y \in \{0,1\}^*$: Again we assume $P(y)$ and substitute it into our invariant:

$$\delta^*(s,y1) = \begin{cases} \delta^*(a,1), & \text{if } y \text{ doesn't contain 101, and doesn't end in 1 or 10} \\ \delta^*(b,1), & \text{if } y \text{ doesn't contain 101 and ends in 1} \\ \delta^*(c,1), & \text{if } y \text{ doesn't contain 101 and ends in 10} \\ \delta^*(d,1), & \text{if } y \text{ contains substring 101} \end{cases}$$

Evaluate the transition function to get

$$\delta^*(s,x) = \begin{cases} b, & \text{if } x \text{ doesn't contain 101, and ends in 1 but not 11} \\ b, & \text{if } x \text{ doesn't contain 101 and ends in 11} \\ d, & \text{if } x \text{ ends in 101, first occurrence of 101} \\ d, & \text{if } x \text{ contains substring 101 followed later by by suffix 1} \end{cases}$$

The two implications for state b combine into "if x doesn't contain 101 and ends in 1, then $\delta^*(s,x) = b$" and the two cases for state d combine into "if x contains 101 and ends in 1, then $\delta^*(s,x) = d$". The implications about the other two states are vacuously true, since there are no 1-transitions into a and c. Thus $P(x)$ is verified for the case where $x = y1$.

In both cases, $P(y) \Rightarrow P(x)$.

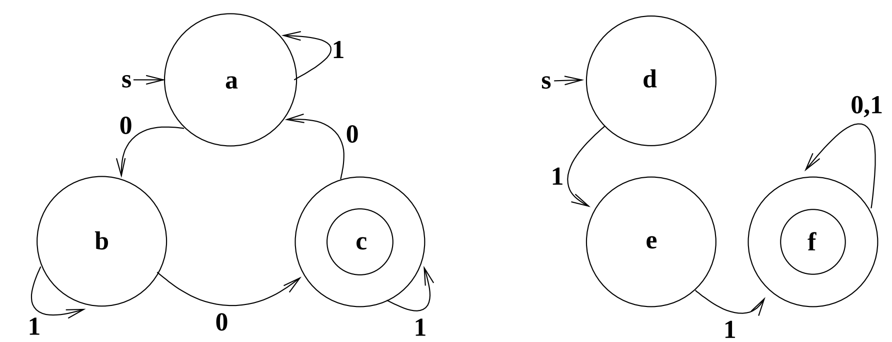
I conclude that $P(x)$ is true for all $x \in \{0,1\}^*$.

To prove Claim 3(c)2, let x be an arbitrary string in L. By $P(x)$, $\delta^*(s,x) = d$, an accepting state. On the other hand, suppose x is accepted by our machine. Then x must contain the substring 101, since otherwise, by $P(x)$, $\delta^*(s,x)$ would be in one of the non-accepting states a, b, or c. Thus our machine accepts exactly the strings of L. QED.

4. Let

$$\begin{aligned} L_1 &= \{x \in \{0,1\}^* : \text{ for some } k \in \mathbb{N}, x \text{ has } 3k+2 \text{ zeros}\} \\ L_2 &= L(11(0+1)^*). \end{aligned}$$

(a) Construct DFSAs M_1 and M_2 so that $L_1 = L(M_1)$ and $L_2 = L(M_2)$.

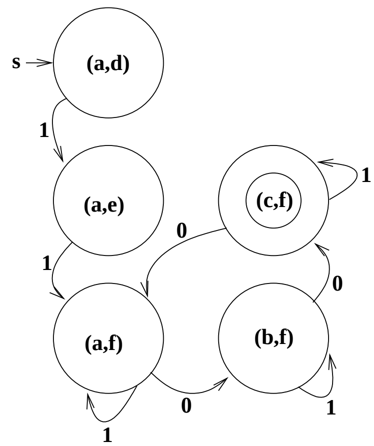


The machine on the left is M_1, and the one on the right is M_2.

(b) Use the Cartesian product construction (page 228) to create a DFSA M' that accepts $L_1 \cap L_2$. First I list the transition function, δ' in table form. In indicate missing (dead) transitions with a dash.

State (q_1, q_2)	$\delta'((q_1,q_2),0)$	$\delta'((q_1,q_2),1)$
(a,d)	—	(a,e)
(a,e)	—	(a,f)
(a,f)	(b,f)	(a,f)
(b,d)	—	(b,e)
(b,e)	—	(b,f)
(b,f)	(c,f)	(b,f)
(c,d)	—	(c,e)
(c,e)	—	(c,f)
(c,f)	(a,f)	(c,f)

By inspection, states (b, d) and (c, d) are never the targets of transition function δ', and states (b, e) and (c, e) are only ever reached from (c, d) and (c, d), so we can omit these states from our diagram



(c) Give a state invariant for M', and prove it correct.

Here is a state invariant for M' (any string not specified in the invariant implicitly takes M' to a dead state):

$$\delta'^*(s, x) = \begin{cases} (a, d), & \text{if } x = \epsilon \\ (a, e), & \text{if } x = 1 \\ (a, f), & \text{if } x \text{ has prefix 11 and } 3k + 0 \text{ 0s} \\ (b, f) & \text{if } x \text{ has prefix 11 and } 3k + 1 \text{ 0s} \\ (c, f) & \text{if } x \text{ has prefix 11 and } 3k + 2 \text{ 0s} \\ \text{dead}, & \text{otherwise} \end{cases}.$$

CLAIM: $P(x)$ "The state invariant above is true for x" holds for all $x \in \{0, 1\}^*$.

PROOF (INDUCTION ON $|x|$): If $|x| = 0$, then $x = \epsilon$, and (by definition of the starting state) $\delta'^*(s, x) = s = (a, d)$. All the other implications in the invariant are true by virtue of having false antecedents. This verifies the basis $P(\epsilon)$.

INDUCTION STEP: Assume that $P(y)$ holds for each $y \in \{0, 1\}^*$ where $|x| > |y| \geq 0$. I need to show that this implies $P(x)$. There are two cases to consider

CASE $x = y0$ AND $P(y)$ IS ASSUMED: Since $|x| > 0$, it is impossible that $x = \epsilon$, and since x has a suffix 0, it impossible that $x = 1$. Also, $\delta(\text{dead}, 0) = \text{dead}$, so by $P(y)$:

$$\delta'^*(s, y0) = \begin{cases} (a, d), & \text{if } y0 = \epsilon \text{ (false antecedent)} \\ (a, e), & \text{if } y0 = 1 \text{ (false antecedent)} \\ \delta((a, f), 0), & \text{if } y \text{ has prefix 11 and } 3k + 0 \text{ 0s} \\ \delta((b, f), 0), & \text{if } y \text{ has prefix 11 and } 3k + 1 \text{ 0s} \\ \delta((c, f), 0), & \text{if } y \text{ has prefix 11 and } 3k + 2 \text{ 0s} \\ \delta(\text{dead}, 0), & \text{otherwise} \end{cases}$$

Use the transition function, δ, and the fact that x has one more 0 than y does:

$$\delta'^*(s, y0) = \begin{cases} (a,d), & \text{if } y0 = \epsilon \text{ (false antecedent)} \\ (a,e), & \text{if } y0 = 1 \text{ (false antecedent)} \\ (b,f), & \text{if } x \text{ has prefix 11 and } 3k+1 \text{ 0s} \\ (c,f), & \text{if } x \text{ has prefix 11 and } 3k+2 \text{ 0s} \\ (a,f), & \text{if } x \text{ has prefix 11 and } 3k+0 \text{ 0s} \\ \text{dead}, & \text{otherwise} \end{cases}.$$

Thus the invariant is preserved in this case.

CASE $x = y1$ AND $P(y)$ IS ASSUMED: Since $|x| > 0$, $x \neq \epsilon$, and $\delta(\text{dead}, 1) = \text{dead}$, so, by $P(y)$:

$$\delta'^*(s, y1) = \begin{cases} \delta((a,d),1), & \text{if } y = \epsilon \\ \delta((a,e),1), & \text{if } y = 1 \\ \delta((a,f),1), & \text{if } y \text{ has prefix 11 and } 3k+0 \text{ 0s} \\ \delta((b,f),1), & \text{if } y \text{ has prefix 11 and } 3k+1 \text{ 0s} \\ \delta((c,f),1), & \text{if } y \text{ has prefix 11 and } 3k+2 \text{ 0s} \\ \delta(\text{dead},1), & \text{otherwise} \end{cases}$$

Evaluating the transition function, and noting that a 1 has been appended:

$$\delta'^*(s, x) = \begin{cases} (a,d), & \text{if } x = \epsilon \text{ (false antecedent)} \\ (a,e), & \text{if } x = 1 \\ (a,f), & \text{if } x = 11 \\ (a,f), & \text{if } x \text{ has prefix 11 and } 3k+0 \text{ 0s} \\ (b,f), & \text{if } x \text{ has prefix 11 and } 3k+1 \text{ 0s} \\ (c,f), & \text{if } x \text{ has prefix 11 and } 3k+2 \text{ 0s} \\ \text{dead}, & \text{otherwise} \end{cases}$$

Thus the invariant is preserved in this case.

In both cases $P(y)$ implies $P(x)$.

I assume that $P(x)$ is true for all $x \in \{0,1\}^*$. QED.

(d) Use the previous part to prove that $L(M') = L_1 \cap L_2$.

CLAIM: $L(M') = L_1 \cap L_2$.

PROOF: Let $x \in L_1 \cap L_2$. Then x has $3k + 2$ zeros and begins with the prefix 11. According to $P(x)$ (proved above), $\delta'^*(s, x) = (c, f)$, the unique accepting state of M', so $x \in L(M')$. Since x was chosen as an arbitrary member of $L_1 \cap L_2$, you have $L_1 \cap L_2 \subseteq L(M')$.

Now suppose $x \in L(M')$. According to $P(x)$, if x doesn't have prefix 11 or $3k + 2$ zeroes, then $\delta'^*(s, x)$ is some state other than (c, f), which contradicts $x \in L(M')$. Thus x does have prefix 11 and $3k + 2$ zeroes, so $x \in L_1 \cap L_2$. Since x was chosen as an arbitrary member of $L(M')$, you have $L(M') \subseteq L_1 \cap L_2$.

By mutual inclusion, $L(M') = L_1 \cap L_2$, as wanted. QED.

5. Is L regular? Justify your claim.

(a) L is the language of first-order formulas with variables $\{x_1, x_2, \ldots\}$, predicate symbol S of arity 3, and constant symbol $\mathbf{c}$.

SOLUTION: L is not regular. One way to see this is to note that regular languages are defined over a finite alphabet, and hence cannot denote all of the formulas that use an infinite set of symbols.

Another way to see this is to use the pumping lemma, and consider a formula $(((\cdots(S(x_1,x_2,x_3) \wedge S(x_1,x_2,x_3)) \wedge \cdots)$ (a formula beginning with p left parentheses, where p is the pumping length). In this case uv^kw will destroy the parity of left and right parentheses for any $k \neq 1$.

(b) $L = \{x \in \{0,1\}^* : |x| \text{ is prime}\}$

SOLUTION: L is not regular. If L were regular, then by the pumping lemma there would be a pumping length p such that every $x \in L$ with $|x| \geq p$ would be expressible as $x = uvw$ with $|uv| \leq p$, $|v| > 0$, and $uv^kw \in L$ for every $k \in \mathbb{N}$. Let d be some prime greater than $p+1$, and let $x = 1^d$ (d 1s). Thus $x \in L$, and $|x| > p+1$, so $x = uvw$, as specified. Since $|uvw| = d > p+1$ and $|v| \leq p$, you know that $|uw| > 1$, so

$$|uv^{|uw|}w| = |uw| + |uw| \times |v| = |uw| \times (1 + |v|),$$

So $|uv^{|uw|}w|$ is not prime (it has two divisors greater than 1), and $uv^{|uw|}w \notin L$, contradicting the assumption that L is regular and the pumping lemma applies.

(c) $L = \{x \in \{0,1\}^* : x \text{ contains exactly one 1 and } x \text{ contains an even number of 0s }\}$.

SOLUTION: L is regular, since $L = L_1 \cap L_2$, where $L_1 = L(0^*10^*)$ (the language with exactly one 1), and $L_2 = L(1^*(01^*01^*)^*$ (the language with an even number of 0s. Both L_1 and L_2 are regular languages, and the regular languages are closed under intersection, so $L = L_1 \cap L_2$ is also regular.

(d) $L = \{x \in L(0^n10^n) : n \in \mathbb{N}, \Sigma = \{0,1\}\}$

SOLUTION: L is not regular. Suppose L were regular, then it would have an associated pumping length p, and whenever $x \in L$ has $|x| \geq p$, then $x = uvw$ with $|uv| << p$, $|v| > 0$, and $uv^kw \in L$ for all $k \in \mathbb{N}$. Let $x = 0^p10^p$. Then $v = 0^j$, for some $1 \leq j \leq p$, so $uv^0w = 0^{p-j}10^p$, which is not in L, since it has fewer 0s before the 1 than after it. This contradicts the assumption that L is regular.

(e) $L = \{x \in \{0,1\}^* : x \text{ contains an equal number of strings 01 and 10}\}$

SOLUTION: L is regular. Here is a DFSA that accepts L: