1. Prove that the following languages are *not* regular.

(a) $\{0^m 1^n \mid m > n\}$

Solution: Consider the set $F = 0^*$.
Let x and y be arbitrary distinct strings in F.
Then $x = 0^i$ and $y = 0^j$ for some integers $i \neq j$.
Without loss of generality, assume $i < j$.
Let $z = 1^i$.

- Then $xz = 0^i 1^i \notin L$, because $i \not> i$.
- But $yz = 0^j 1^i \in L$, because $j > i$.

Thus, z is a distinguishing suffix for x and y.
We conclude that F is a fooling set for L.
Because F is infinite, L cannot be regular. ■

Rubric: 3 points: standard fooling set rubric (scaled). This is not the only correct solution.

- Watch carefully for inequality reversal (proving $i > j$ instead of $i < j$).
- We really do need to assume $i < j$, because otherwise $0^j 1^i \notin L$.

(b) $\left\{w \in (0+1)^* \;\middle|\; \#(0,w)/\#(1,w) \text{ is an integer}\right\}$

Solution: Consider the set $F = 0^+$.
Let x and y be arbitrary distinct strings in F.
Then $x = 0^i$ and $y = 0^j$ for some *positive* integers $i \neq j$.
Without loss of generality, assume $i < j$.
Let $z = 1^j$.

- $xz = 0^i 1^j \notin L$, because $\#(0,xz)/\#(1,xz) = i/j$ lies strictly between 0 and 1 and thus is not an integer.
- $yz = 0^j 1^j \in L$, because $\#(1,yz) = j > 0$ and $\#(0,yz)/\#(1,yz) = j/j = 1$ is an integer.

Thus, z is a distinguishing suffix for x and y.
We conclude that F is a fooling set for L.
Because F is infinite, L cannot be regular. ■

Rubric: 3 points: standard fooling set rubric (scaled). This is not the only correct solution.

- We are really reasoning about the language $L \cap 0^+1^+$.
- The main idea is to force $\#(0,w) \leq \#(1,w)$, so that the ratio lies between 0 and 1.
- We really do need i to be positive, because $xz \in L$ when $i = 0$ and $j > 0$.
- We really do need to assume $i < j$, because $xz \in L$ when $i = 2j$, for example.
- Watch carefully for reversals ("$\#(1,w)/\#(0,w)$ is an integer").

(c) The set of all palindromes in $(0+1)^*$ whose length is divisible by 7.

Solution: Consider the set $F = (0000000)^*1111111 = \{0^{7n}1^7 \mid n \geq 0\}$.
Let x and y be arbitrary distinct strings in F.
Then $x = 0^{7i}1^7$ and $y = 0^{7j}1^7$ for some integers $i \neq j$.
Let $z = 0^{7i}$.

- Then $xz = 0^{7i}1^70^{7i}$ is a palindrome of length $7(2i+1)$, so $xz \in L$.
- But $yz = 0^{7j}1^70^{7i}$ is not a palindrome (because $i \neq j$), so $yz \notin L$.

Thus, z is a distinguishing suffix for x and y.
We conclude that F is a fooling set for L.
Because F is infinite, L cannot be regular. ■

Solution: Consider the set $F = (0000000)^*000 = \{0^{7n+3} \mid n \geq 0\}$.
Let x and y be arbitrary distinct strings in F.
Then $x = 0^{7i+3}$ and $y = 0^{7j+3}$ for some integers $i \neq j$.
Let $z = 10^{7i+3}$.

- Then $xz = 0^{7i+3}10^{7i+3}$ is a palindrome of length $2(7i+3)+1 = 7(2i+1)$, so $xz \in L$.
- But $yz = 0^{7j+3}10^{7i+3}$ is not a palindrome (because $i \neq j$), so $yz \notin L$.

Thus, z is a distinguishing suffix for x and y.
We conclude that F is a fooling set for L.
Because F is infinite, L cannot be regular. ■

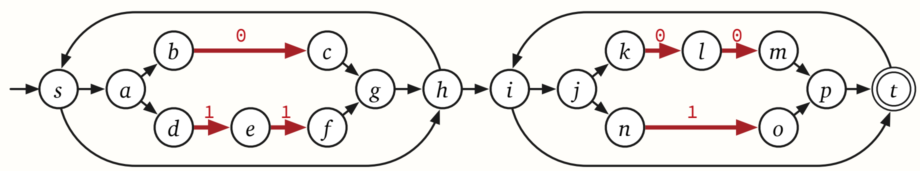
Rubric: 4 points: standard fooling set rubric (scaled). These are not the only correct solutions.

2. For each of the following regular expressions, describe or draw two finite-state machines:

   - An NFA that accepts the same language, constructed from the given regular expression using Thompson's algorithm (described in class and in the notes).
   - An equivalent DFA, constructed from your NFA using the incremental subset algorithm (described in class and in the notes). For each state in your DFA, identify the corresponding subset of states in your NFA. Your DFA should have no unreachable states.

(a) $(0+11)^*(00+1)^*$

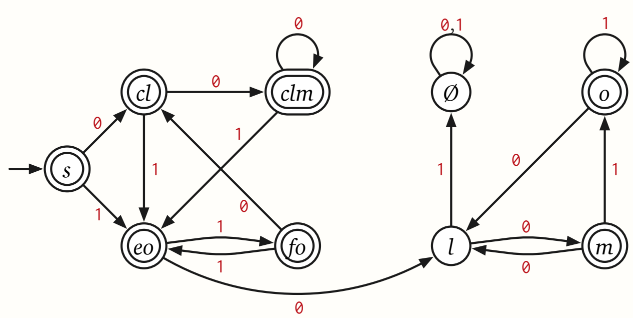
Solution: Thompson's algorithm yields the following 18-state NFA. (Unlabeled arrows indicate ε-transitions.)



The incremental subset algorithm builds the following table.

q'	ε-reach	0	1	A'?
s	$sabdhijknt$	cl	eo	✓
cl	$sabcdghijklnt$	clm	eo	✓
eo	$eijknopt$	l	fo	✓
clm	$sabcdghijklmnt$	clm	eo	✓
l	l	m	$\varnothing$	
fo	$sabdfhijknot$	cl	eo	✓
m	$ijkmnpt$	l	o	✓
o	$ijknopt$	l	o	✓
$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	

We obtain the following 9-state DFA:

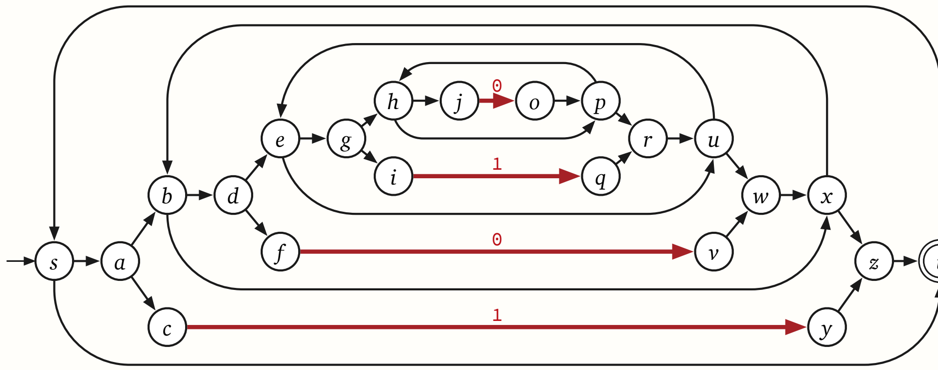


(States s, cl, and clm are equivalent, but simplifying the DFA is beyond the scope of the homework.) ■

Rubric: 5 points = 2½ points for NFA + 2½ points for DFA; see page 5 for details.

(b) $(((0^* + 1)^* + 0)^* + 1)^*$

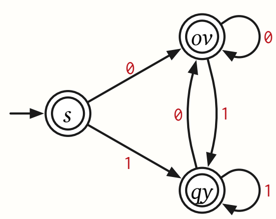
Solution: Thompson's algorithm yields the following 22-state NFA. (As usual, unlabeled arrows indicate ε-transitions.)



The incremental subset algorithm builds the following table. I'm explicitly listing only the *complement* of the ε-reach of each state, both to save space and to protect against errors.

q'	ε-reach	0	1	A'?
s	$Q \setminus oqvy$	ov	qy	✓
ov	$Q \setminus qy$	ov	qy	✓
qy	$Q \setminus ov$	ov	qy	✓

We obtain the following DFA with three(!) states:



(All states are accepting and therefore equivalent, but simplifying the DFA is beyond the scope of the homework.) ∎

Rubric: 5 points = 2½ points for NFA + 2½ points for DFA; see next page for details.

Rubric (for each part of problem 2): 5 points =

- 2½ points for Thompson's NFA:
  - No credit unless NFA is actually constructed using Thompson's algorithm.
  - −1 for one small mistake; otherwise, no credit unless the NFA accepts the target language.
  - No penalty for "safe" simplifications, like using a simple path without ε-transitions for individual strings (as shown in the part (a) solution), or generalizing the $+$ gadget to more than two subexpressions, provided these simplifications are applied correctly and consistently.
- 2½ points for incremental-subset DFA:
  - No credit unless the DFA is constructed from the NFA given in part (a) using the incremental subset algorithm, **without further simplifications**.
  - Removing ***all*** ε-transitions from the NFA in part (a) before applying the incremental subset algorithm is fine, as long as it's done systematically (using the algorithm described in the notes) and correctly. This will not change the final DFA.
  - −1 for one small mistake; otherwise, no credit unless the DFA accepts the target language.
  - **Either** the table **or** the drawing (with state labels) is sufficient for full credit; it is not necessary to provide both. The table (even without the ε-reach column) is a complete description of the DFA! **If you provide both, the grader is free to grade *either* the table *or* the drawing.**
  - −1 for a bare drawing without labels correctly indicating the set of NFA-states corresponding to each DFA-state.
  - No further explanation of the DFA is necessary.

3. For each of the following languages over the alphabet $\Sigma = \{0, 1\}$, either prove that the language is regular (by constructing an appropriate DFA, NFA, or regular expression) or prove that the language is not regular (by constructing an infinite fooling set).

(a) $\left\{0^a 1^b 0^c \;\middle|\; (a \le b + c \text{ and } b \le a + c) \text{ or } c \le a + b\right\}$

Solution: Regular. This is the language $\mathbf{0}^*\mathbf{1}^*\mathbf{0}^*$.

- Consider an arbitrary string $w \in L$.
  By definition of L, we have $w = 0^a 1^b 0^c$ for some integers a, b, c.
  It follows immediately that $w \in 0^* 1^* 0^*$.
- On the other hand, consider an arbitrary string $w \in 0^* 1^* 0^*$.
  We immediately have $w = 0^a 1^b 0^c$ for some integers a, b, c.
  There are two cases to consider.
  - If $c \le a + b$, then $w \in L$ by definition.
  - If $c > a + b$, then $c > a$ and $c > b$, which implies both $a < b + c$ and $b < a + c$, so again $w \in L$ by definition.

  In both cases, we conclude that $w \in L$.

■

Rubric: 2½ points: ½ for "regular" + 1 for regular expression + 1 for justification (= ½ for "if" + ½ for "only if"). This is more detail than necessary for full credit. This is not the only correct solution.

(b) $\left\{0^a 1^b 0^c \;\middle|\; a \le b + c \text{ and } (b \le a + c \text{ or } c \le a + b)\right\}$

Solution (fooling set): Not regular. Consider the set $F = 0^*$.

Let x and y be arbitrary distinct strings in F.

Then $x = 0^i$ and $y = 0^j$ for some integers $i \ne j$.

Without loss of generality, assume $i < j$.

Let $z = 1^i$.

- Then $xz = 0^i 1^i = 0^a 1^b 0^c$ where $(a, b, c) = (i, i, 0)$.
  We have $a = i \le i = b + c$ and $b = i \le i = a + c$, which implies $xz \in L$.
- And $yz = 0^j 1^i = 0^a 1^b 0^c$ where $(a, b, c) = (j, i, 0)$.
  We have $a = j > i = b + c$, which implies $yz \notin L$.

Thus, z is a distinguishing suffix for x and y.

We conclude that F is a fooling set for L.

Because F is infinite, L cannot be regular. ■

Rubric: 2½ points: standard fooling set rubric (scaled). This is not the only correct fooling set argument. We really do need to assume $i < j$; otherwise, $yz \in L$.

The main idea of this solution is to impose additional structure that forces $c = 0$. We are really reasoning about the language $L \cap 0^*1^* = \{0^a 1^b \mid a \le b\}$.

Solution (closure): Let a, b, c be arbitrary non-negative integers. If $b < c$, then $b \le a + c$; on the other hand, if $b \ge c$, then $c \le b + a$. Thus, the parenthesized "or" condition in the definition of L is redundant; we can express L more simply as $\left\{0^a 1^b 0^c \;\middle|\; a \le b + c\right\}$.

If L is regular, then the language $0^*1^* \setminus L = \left\{0^a 1^b \;\middle|\; a > b\right\}$ must also be regular, because regular languages are closed under boolean operations.

But we proved that $\left\{0^a 1^b \;\middle|\; a > b\right\}$ is not regular in problem 1(a).

We conclude that L is **not regular**. ■

Rubric: 2½ points. Yes, even without a solution to problem 1(a).

(c) $\{wxw^R \mid w, x \in \Sigma^+\}$

Solution: Regular. This is the language $\mathtt{0}(\mathtt{0}+\mathtt{1})^+\mathtt{0} + \mathtt{1}(\mathtt{0}+\mathtt{1})^+\mathtt{1}$ of all strings of length at least 3 that start and end with the same symbol.

- Let z be an arbitrary string in L.
  By definition, $z = wxw^R$ for some non-empty strings w and x.
  Because $w \neq \varepsilon$, we have $w = ay$ for some symbol a and some string y.
  The definition of reversal implies $w^R = y^R a$.
  Thus, $z = ayxy^R a$ starts and ends with the same symbol a.
  The remaining substring yxy^R is non-empty, because x is nonempty.
  We conclude that $z \in \mathtt{0}(\mathtt{0}+\mathtt{1})^+\mathtt{0} + \mathtt{1}(\mathtt{0}+\mathtt{1})^+\mathtt{1}$.
- On the other hand, let z be an arbitrary string in $\mathtt{0}(\mathtt{0}+\mathtt{1})^+\mathtt{0} + \mathtt{1}(\mathtt{0}+\mathtt{1})^+\mathtt{1}$.
  Then $z = axa$ for some symbol a and some nonempty string x.
  Because $a = a^R$, we have $z = axa^R$, which implies $z \in L$.

■

Rubric: 2½ points: ½ for "regular" + 1 for regular expression + 1 for justification (= ½ for "if" + ½ for "only if"). This is more detail than necessary for full credit.

(d) $\{ww^Rx \mid w, x \in \Sigma^+\}$

Solution: Not regular. Consider the set $F = \mathtt{1}(\mathtt{00})^*\mathtt{01}$.

Let x and y be arbitrary distinct strings in F.

Then $x = \mathtt{10}^{2i+1}\mathtt{1}$ and $y = \mathtt{10}^{2j+1}\mathtt{1}$ for some non-negative integers $i \neq j$.

Let $z = \mathtt{10}^{2i+1}\mathtt{11}$.

- Then $xz = \mathtt{10}^{2i+1}\mathtt{110}^{2i+1}\mathtt{11} = ww^Rv$, where $w = x$ and $v = \mathtt{1}$, so $xz \in L$.
- For the sake of argument, suppose $yz = \mathtt{10}^{2j+1}\mathtt{110}^{2i+1}\mathtt{11} \in L$.
  Then $yz = ww^Rv$ for some non-empty strings w and v.
  The first two symbols of yz are different, so $|w| > 1$.
  The prefix w begins with $\mathtt{10}$ (the first two symbols of yz).
  So its reversal w^R must end with $\mathtt{01}$.
  The substring $\mathtt{01}$ appears exactly twice in yz.
  So there are only two possibilities for the substring ww^R.
  - $ww^R = \mathtt{10}^{2j+1}\mathtt{1}$ is impossible because $|ww^R|$ must be even.
  - $ww^R = \mathtt{10}^{2j+1}\mathtt{110}^{2i+1}\mathtt{1}$ is impossible because ww^R must be a palindrome, and $i \neq j$.

  We have derived a contradiction, which implies that $yz \notin L$.

Thus, z is a distinguishing suffix for x and y.

We conclude that F is a fooling set for L.

Because F is infinite, L cannot be regular. ■

Rubric: 2½ points: standard fooling set rubric (scaled). This is not the only correct solution.

The main idea here is to impose additional structure that forces the prefix w to be arbitrarily long. We are really reasoning about the language

$$L \cap \mathtt{1}(\mathtt{00})^*\mathtt{011}(\mathtt{00})^*\mathtt{011} = \{\mathtt{10}^n\mathtt{110}^n\mathtt{11} \mid n \text{ is odd}\}.$$