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

   (a) $\{0^a 1 0^b 1 0^c \mid a + b = c\}$

Solution: Consider the set $F = 10^*1$.
Let x and y be arbitrary distinct strings in F.
Then $x = 10^i1$ and $y = 10^j1$ for some integers $i \neq j$.
Let $z = 0^i$.

- Then $xz = 10^i10^i = 0^010^i10^i \in L$ because $0 + i = i$.
- But $yz = 10^j10^i = 0^010^j10^i \notin L$ because $0 + j \neq i$.

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. ■

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 $i \neq j$.
Let $z = 110^i$.

- Then $xz = 0^i110^i = 0^i10^010^i \in L$ because $i + 0 = i$.
- But $yz = 0^j110^i = 0^j10^010^i \in L$ because $j + 0 \neq i$.

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. ■

Solution: Consider the set $F = \{0^n10^n1 \mid n \geq 0\}$.
Let x and y be arbitrary distinct strings in F.
Then $x = 0^i10^i1$ and $y = 0^j10^j1$ for some integers $i \neq j$.
Let $z = 0^{2i}$.

- Then $xz = 0^i10^i10^{2i} \in L$ because $i + i = 2i$.
- But $yz = 0^j10^j10^{2i} \notin L$ because $j + j \neq 2i$.

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). These are not the only correct solutions.

(b) $\{w \in (0+1)^* \mid \#(0,w) \le 2 \cdot \#(1,w)\}$

Solution: Consider the set $F = 1^+ = 11^*$.
Let x and y be arbitrary distinct strings in F.
Then $x = 1^i$ and $y = 1^j$ for some $i \neq j$.
Without loss of generality, assume $i < j$.
Let $z = 0^{2i+1}$.

- $xz = 1^i 0^{2i+1} \notin L$ because $\#(0,xz) = 2i+1 \not\le 2i = 2 \cdot \#(1,xz)$.
- $yz = 1^j 0^{2i-1} \in L$ because $\#(0,yz) = 2i+1 \le 2j-1 < 2j = 2 \cdot \#(1,yz)$.

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 $F = 0(00)^*$.
Let x and y be arbitrary distinct strings in F.
Then $x = 0^{2i+1}$ and $y = 0^{2j+1}$ for some non-negative integers $i \neq j$.
Without loss of generality, assume $i < j$ and therefore $i+1 \le j$.
Let $z = 1^j$.

- $xz = 0^{2i+1}1^j \in L$ because $\#(0,xz) = 2i+1 < 2i+2 \le 2j = 2 \cdot \#(1,xz)$.
- $yz = 0^{2j+1}1^j \notin L$ because $\#(0,yz) = 2j+1 \not\le 2j = 2 \cdot \#(1,yz)$.

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 $F = 0^*$.
Let x and y be arbitrary distinct strings in F.
Then $x = 0^i$ and $y = 0^j$ for some non-negative integers $i \neq j$.
Without loss of generality, assume $i < j$ and therefore $i+1 \le j$.
Let $z = 1^i 0^{i+1}$.

- Then $xz = 0^i 1^i 0^{i+1} \notin L$ because $\#(0,xz) = 2i+1 \not\le 2i = 2 \cdot \#(1,xz)$.
- But $yz = 0^j 1^i 0^{i+1} \in L$ because $\#(0,yz) = i+j+1 \le 2j = 2 \cdot \#(1,yz)$.

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). These are not the only correct solutions. Watch out for boundary conditions (like allowing $i = 0$ in the first proof), inequality reversals (proving $\ge$ instead of $\le$), and off-by-one errors!

(c) $\left\{0^m 1^n \;\middle|\; m+n>0 \text{ and } \gcd(m,n)=1\right\}$

Solution: Consider the set $F = \{0^p \mid p \text{ is prime}\}$.
Let x and y be arbitrary distinct strings in F.
Then $x = 0^p$ and $y = 0^q$ for some distinct primes p and q.
Let $z = 1^q$.

- Then $xz = 0^p 1^q \in L$ because $\gcd(p,q) = 1$.
- But $yz = 0^q 1^q \notin L$ because $\gcd(q,q) = q \neq 1$.

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 $F = 0^*$.
Let x and y be arbitrary distinct strings in F.
Then $x = 0^i$ and $y = 0^i$ for some non-negative integers i and j.
Without loss of generality, assume $i < j$.
Let $z = 0^{(k-1)i} 1^k$, where k is any positive integer such that $\gcd(j-i,k) = 1$. (For example, let k be any prime larger than j.)

- Then $xz = 0^{ki} 1^k \notin L$ because $\gcd(ki,k) = k \neq 1$.
- But $yz = 0^{ki+j-i} 1^p \notin L$ because $\gcd(ki+j-i,k) = \gcd(k,j-i) = 1$, by Aryabhata's Pulverizer:[a]

$$\gcd(x,y) = \begin{cases} x & \text{if } y = 0 \\ \gcd(y, x \bmod y) & \text{if } y > 0 \end{cases}$$

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. ■

[a] known in the West as "Euclid's algorithm"

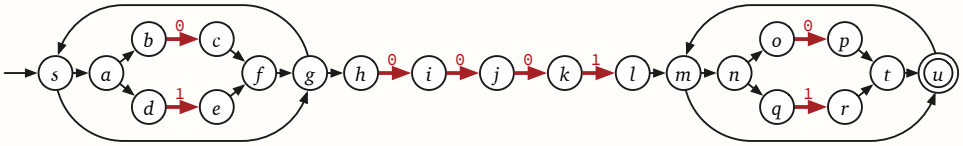
Rubric: 4 points: standard fooling set rubric (scaled). These are not the only correct solutions. Watch out for boundary conditions and off-by-one errors!

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) $(0+1)^* \cdot 0001 \cdot (0+1)^*$

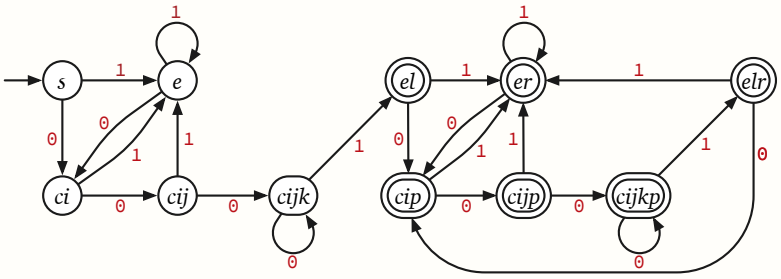
Solution: Thompson's algorithm yields the following 21-state NFA. (Unlabeled arrows indicate ε-transitions.)



The incremental subset algorithm builds the following table:

q'	ε-reach	0	1	A'?
s	$sabdgh$	ci	e	
ci	$sabdcfghi$	cij	e	
e	$sabdefgh$	ci	e	
cij	$sabdcfghij$	$cijk$	e	
$cijk$	$sabdcfghijk$	$cijk$	el	
el	$sabdefghlmnoqu$	cip	er	✓
cip	$sabdcfghimnoqptu$	$cijp$	er	✓
er	$sabdefghmnoqrtu$	cip	er	✓
$cijp$	$sabdcfghijmnoqptu$	$cijkp$	er	✓
$cijkp$	$sabdcfghijkmnoqptu$	$cijkp$	elr	✓
elr	$sabdefghlmnoqrtu$	cip	er	✓

We obtain the following 11-state DFA:[a]



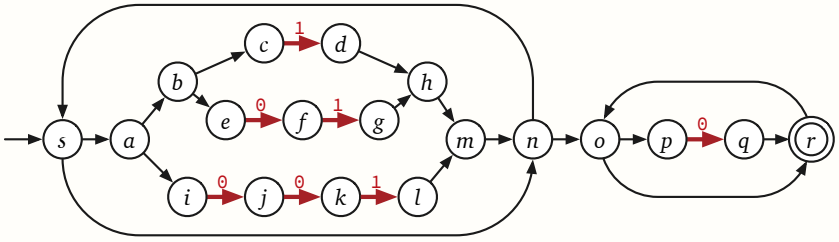
■

[a] Obviously the six accept states can be merged together; states s and e can also be merged, because they have the same outgoing transitions. Applying both simplifications leaves us with a minimal five-state DFA. But these simplifications are beyond the scope of the homework.

Rubric: 5 points = 2½ points for NFA + 2½ points for DFA; see page 6 for details.

(b) $(1 + 01 + 001)^*0^*$

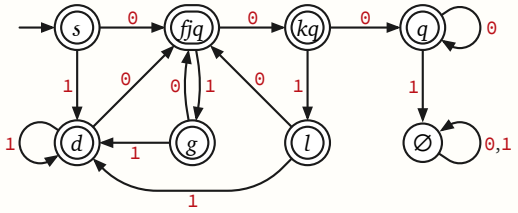
Solution: Thompson's algorithm yields the following 19-state NFA. (Unlabeled arrows indicate ε-transitions.)



The incremental subset algorithm builds the following table:

q'	ε-reach	0	1	A'?
s	$sabceinopr$	fjq	d	✓
fjq	$fjopqr$	kq	g	✓
d	$sabcdehimnopr$	fjq	d	✓
kq	$kopqr$	q	l	✓
g	$sabceghimnopr$	fjq	d	✓
q	$opqr$	q	$\emptyset$	✓
l	$sabceilmnopr$	fjq	d	✓
$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	

We obtain the following 8-state DFA:[a]



■

[a] States s, d, g, and l can be merged together, because they have the same outgoing transitions, leaving us with a minimal five-state DFA. In fact, it's the same minimal DFA as in part (a), but with complementary accepting states!

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 **in this homework** for omitting ε-transitions from Kleene star or concatenation gadgets — see the lab solutions for an explanation. (For this language, the non-standard gadgets don't break the NFA, but that's not always true.)
  - No penalty for "safe" simplifications, like using a simple path without ε-transitions for individual strings (as shown here), or generalizing the $+$ gadget to more than two subexpressions (merging a with b and h with m in the solution for part (b)), provided these simplifications are applied consistently.
  - **Careful:** Thompson's algorithm yields a unique NFA for every regular expression tree, but not for every regular expression. In particular, subexpressions of the form $A+B+C$ can be parsed as either $(A+B)+C$ or $A+(B+C)$, and each of these leads to a correct NFA.
- 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) Strings in which the substrings 00 and 11 do not appear the same number of times.

Solution: Not regular. Consider the set $F = 0^+ = 00^*$.

Let x and y be arbitrary distinct strings in F.

Then $x = 0^i$ and $y = 0^j$ for some *positive* integers $i \neq j$.

Let $z = 1^i$.

- Then $xz = 0^i 1^i \notin L$ because 00 and 11 each appear $i - 1$ times (because $i > 0$).
- But $yz = 0^j 1^i \in L$ because 00 appears $j - 1$ times (because $j > 0$), 11 appears $i - 1$ times (because $i > 0$), and $i \neq j$.

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 answer. Careful about $i = 0$!

   (b) Strings in which the substrings 01 and 10 do not appear the same number of times.

Solution: Regular. This is the set of all non-empty binary strings whose first and last symbols are different; thus, the language is described by the regular expression

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

Every string can be partitioned into an alternating sequence of substrings in 0^+ and substrings in 1^+; for example:

$$0000000111100011111011111 = 0^7 \bullet 1^4 \bullet 0^3 \bullet 1^5 \bullet 0^1 \bullet 1^5$$

The substrings 01 and 10 appear precisely at the boundaries of these blocks:

$$0000000\underline{01}\underline{11}\underline{00}\underline{01}\underline{11}\underline{10}\underline{1}\underline{1}1111$$

Thus, the number of 01 substrings and the number of 10 substrings are equal if and only if the first and last symbols are equal (or the string is empty). ■

Rubric: 2½ points: ½ for "regular" + 1 for regular expression + 1 for justification

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

Solution: Regular. Let z be an arbitrary string $z \in \Sigma^*$. Then we have $z = \varepsilon \bullet z \bullet \varepsilon$. Since $\varepsilon \in \Sigma^*$ by definition, and $z \in \Sigma^*$ by assumption, we have $z \in L$. We conclude that $L = \Sigma^*$! ■

Rubric: 2½ points: ½ for "regular" + 1 for regular expression + 1 for justification

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

Solution: Not regular. Consider the set $F = 0^*1$.

Let x and y be arbitrary distinct strings in F.

Then $x = 0^i1$ and $y = 0^j1$ for some non-negative integers $i \neq j$.

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

Let $z = 10^i1$.

- Then $xz = 0^i110^i1 = 0^i1 \bullet 1 \bullet 0^i1 \in L$.
- But I claim that $yz = 0^j110^i1 \notin L$. Suppose to the contrary that $0^j110^i1 = wvw$ for some non-empty strings w and v. Then w must end with 1, which implies that w must begin with the prefix 0^j. But the substring 0^j appears only once in yz, because $i < j$, so we have a contradiction.

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 inequality $i < j$ is crucial here: If $i > j$, then $0^j110^i1 = 0^j1 \bullet 10^{i-j} \bullet 0^j1 \in L$!

Standard fooling set rubric. For problems worth 5 points:

- 2 points for the fooling set:
  - + 1 for explicitly describing the proposed fooling set F.
  - + 1 if the proposed set F is actually a fooling set for the target language.
  - — No credit for the proof if the proposed set is not a fooling set.
  - — No credit for the *problem* if the proposed set is finite.
- 3 points for the proof:
  - ◦ The proof must correctly consider *arbitrary* strings $x, y \in F$.
    - — No credit for the proof unless both x and y are *always* in F.
    - — No credit for the proof unless x and y can be *any* strings in F.
  - + 1 for correctly describing a suffix z that distinguishes x and y.
  - + 1 for proving either $xz \in L$ or $yz \in L$.
  - + 1 for proving either $yz \notin L$ or $xz \notin L$, respectively.

As usual, scale partial credit (rounded to nearest ½) for problems worth fewer points.