CS/ECE 374 A ✧ Spring 2026

Midterm 1 Practice 2

February 20, 2026

Name:	Sarah Dowden
NetID:	sdowden2

- ***Don't panic!***
- You have 120 minutes to answer five questions. The questions are described in more detail in a separate handout.
- If you brought anything except your writing implements, your **hand-written** double-sided 8½" × 11" cheat sheet, and your university ID, please put it away for the duration of the exam. In particular, please turn off and put away *all* medically unnecessary electronic devices.
- Please clearly print your name and your NetID in the boxes above.
- Please also print your name at the top of every page of the answer booklet, except this cover page. We want to make sure that if a staple falls out, we can reassemble your answer booklet. (It doesn't happen often, but it does happen.)
- Proofs or other justifications are required for full credit if and only if we explicitly ask for them, using the word ***prove*** or ***justify*** in bold italics.
- **Do not write outside the black boxes on each page**. These indicate the area of the page that our scanners will actually scan. If the scanner can't see your work, we can't grade it.
- If you run out of space for an answer, please use the overflow/scratch pages at the back of the answer booklet, but **please clearly indicate where we should look**. If we can't find your work, we can't grade it.
- **Only work that is written into the stapled answer booklet will be graded**. In particular, you are welcome to detach scratch pages from the answer booklet, but any work on those detached pages will not be graded. Please let us know if you detach a page accidentally. We will provide additional scratch paper on request.
- Please return ***all*** paper with your answer booklet: your question sheet, your cheat sheet, and all scratch paper. **Please put all loose paper *inside* your answer booklet.**

Is between
between is and between
or between between and is
or between and and or
or between or and between
or between between and and
or between or and and
or between or and or?

CS/ECE 374 A ✧ Spring 2026 Midterm 1 Practice 2 Problem 1	Name: Sarah Dowden (sdowden2)

For each statement below, check "Yes" if the statement is always true and check "No" otherwise, and give a brief (one short sentence) explanation of your answer. Read these statements very carefully—small details matter!

(a) Every infinite language is regular.

☐ Yes ☑ No — False because $L = \{0^i : i \text{ is prime}\}$

(b) The language $(0 + 1(01^*0)^*1)^*$ is not context-free.

☐ Yes ☑ No — False because it is represented as a regex ⇒ regular, and all regular languages are context-free

(c) Every subset of an irregular language is irregular.

☐ Yes ☑ No — False. $\{01\} \subseteq \{0^i 1^i : i \in \mathbb{N}\}$ (regular ⊆ irregular)

(d) The language $\{0^a 1^b \mid a - b \text{ is divisible by } 374\}$ is regular.

$\delta(q,1) = q+1 \bmod 374$, $\delta(q,0) = q-1 \bmod 374$

☑ Yes ☐ No — True: $M = (Q, \Sigma, s, \delta, A)$, $Q = \{i : i \in \{-374, -373, \ldots, 1, \ldots, 374\}\}$, $s = 0$, $A = 0$ ⇒ proves we can easily track $a-b \bmod 374$

(e) If language L is not regular, then L has a finite fooling set.

☑ Yes ☒ No — $\emptyset$ is a fooling set for every language

(f) If there is a DFA that rejects every string in language L, then L is regular.

☐ Yes ☑ No —  rejects everything, which also then rejects the entire language L, but says nothing about what L is

(g) If language L is accepted by an DFA with n states, then its complement $\Sigma^* \setminus L$ is also accepted by a DFA with n states.

☑ Yes ☐ No — True, can just make $M' = (Q, \Sigma, \delta, s, A')$ where $A' = Q \setminus A$ from original DFA $M = (Q, \Sigma, \delta, s, A)$

(h) 1^*0^* is a fooling set for the language $\{1^i 0^{i+j} 1^j \mid i, j \geq 0\}$.

☒ Yes ☑ No — ~~$F = \{1^m 0^n : m \neq n\}$~~ $10 + 1100$ have no distinguishing suffix: ~~$x = 1^a 0^b$, $y = 1^c 0^d$, $b < d$~~ $x2 = 1^a 0^b 0^a 1^b$, $y2 = 1^c 0^d 0^a 1^b$

(i) Every regular language is accepted by a DFA with an odd number of accepting states.

☑ Yes ☐ No — True, if even # accepting states, we can always add an isolated accepting state to make a new DFA with even # of accepting states

(j) The context-free grammar $S \to \varepsilon \mid 0S1S \mid 1S0S$ generates all strings in which the number of 0s equals the number of 1s.

☑ Yes ☐ No — True, can only add a 0 when you add a 1, so #0s = #1s.

CS/ECE 374 A ✧ Spring 2026 Midterm 1 Practice 2 Problem 2	Name: Sarah Dowden (sdowden2)

For any non-empty string $w \in \{0,1\}^*$, let Delete1st(w) denote the string obtained by deleting the first run in w. Let L be an arbitrary regular language over the alphabet $\Sigma = \{0,1\}$. ***Prove*** that the following languages are also regular.

Delete(00010) = 10

(a) $\textsc{Insert1st}(L) = \{w \in \Sigma^* \mid w \neq \varepsilon \text{ and } \text{Delete1st}(w) \in L\}$

(b) $\textsc{Delete1st}(L) = \{\text{Delete1st}(w) \mid w \neq \varepsilon \text{ and } w \in L\}$ On Page 6

---

a) Want to apply delete

Create NFA M' to prove regularity. Given arbitrary DFA $M = (Q, \Sigma, \delta, s, A)$ which accepts the language L, which we know is regular.

$M' = (Q', \Sigma, \delta', s', A')$

$Q' = Q \times \{\varepsilon, 0, 1\} \times \{y, n\}$

what first char is; yes delete, no don't delete

$s' = (s, \varepsilon, y)$

start state transition $\delta'((s,\varepsilon,y), a) = \{(s, a, y)\}$

deleting transitions $\delta'((q,a,y), a) = \{(q, a, y)\}$

$\delta'((q,a,y), b) = \{(\delta(q,b), a, n\}$

non-deleting transitions $\delta'((q,a,n), c) = \{(\delta(q,c), a, n\}$

$A' = A$

M' deletes first run of characters with the deleting transitions. Once first run is deleted then it continues on like normal.

CS/ECE 374 A ✧ Spring 2026 Midterm 1 Practice 2 Problem 3	Name: Sarah Dowden (sdowden2)

For any string $w \in \{0,1\}^*$, let squish(w) denote the string obtained by dividing w into pairs of symbols, replacing each pair with 0 if the symbols are equal and 1 otherwise, and keeping the last symbol if w has odd length. (See the question handout for a formal recursive definition.)

01|10|11|00|0 → 1 1 0 0 0

assume squish works like this

(a) ***Prove*** that $\#(1, \text{squish}(w)) \le \#(1, w)$ for every string w.

(b) ***Prove*** that $\#(1, \text{squish}(w))$ is even if and only if $\#(1, w)$ is even, for every string w.

---

a) Let w be an arbitrary string, $w \in \{0,1\}^*$.

IH: For every arbitrary string $x \in \{0,1\}^*$, such that $|x| < |w|$, $\#(1, \text{squish}(x)) \le \#(1, x)$.

IS: We want to prove the statement in the question true for arbitrary string $w \in \{0,1\}^*$.

- $w = a$ for some character a
  - squish(w) = squish(a) = a (by def of squish)
  - $\#(1, \text{squish}(w)) = \#(1, a) \le \#(1, w)$ as $a = w$
- $w = \varepsilon$
  - squish(w) = squish(ε) = ε (by def of squish)
  - $\#(1, \text{squish}(w)) = \#(1, \varepsilon) \le \#(1, w)$ as $w = \varepsilon$
- $w = ab$, for some chars a, b s.t. $a \neq b$
  - squish(w) = squish(ab) = 1 (by def of squish)
  - $\#(1, \text{squish}(w)) = \#\text{squish}(a, b) = 1$
  - $\#(1, w) = 1$, since $a \neq b$, $a, b \in \{0,1\}$ So either a or b must be 1
  - Therefore $\#(1, \text{squish}(w)) \le \#(1, w)$

For part a continue to page 7

b) Let w be an arbitrary string, $w \in \{0,1\}^*$

IH: For every arbitrary string $x \in \{0,1\}^*$, such that $|x| < |w|$, $\#(1, \text{squish}(w)) \bmod 2 = \#(1, w) \bmod 2$.

IS: We want to prove question statement true for arbitrary $w \in \{0,1\}^*$.

- $w = \varepsilon$, so squish(w) = ε (by def). Then $\#(0, \varepsilon) = 0$, so both have even parity.
- $w = a$, $a \in \{0,1\}$, squish(w) = squish(a) = a, so then w = squish(w). If $\#(1, w)$ is even, then $\#(1, \text{squish}(w))$ is even as $w = \text{squish}(w)$.
- $w = ab$, $a, b \in \{0,1\}$, $a \neq b$. Either a or b is 1, so $\#(1, w) = 1$. squish(w) = squish(ab) = 1, $\#(1,1) = 1$, so $\#(1, w) = \#(1, \text{squish}(w))$, so both are odd.
- $w = aa$, $a \in \{0,1\}$. If $a = 1$, then $\#(1, w) = \#(1, 11) = 2$ (even). squish(w) = squish(11) = 0 (even). If $a = 0$, then $\#(1, w) = \#(1, 00) = 0$ (even. squish(w) = squish(00) = 0 (even). In both cases, parity is the same which is what we want to prove.

For part b, continue to page 7

CS/ECE 374 A ✧ Spring 2026 Midterm 1 Practice 2 Problem 4	Name: Sarah Dowden (sdowden2)

Let L be the set of all strings in $\{0,1\}^*$ in which every run of 0s is followed immediately by a *shorter* run of 1s.

(a) ***Prove*** that L is *not* a regular language.

(b) Describe a context-free grammar for L.

---

a)

Consider infite fooling set $F = \{0^n : n \in \mathbb{N}\}$

Let x, y be arbitrary strings such that $x, y \in F$. $x = 0^i$, $y = 0^j$ where $i < j$.

Let $z = 1^i$.

$xz = 0^i 1^i$, and $xz \notin L$, since the run of 1s following run of zeros is the same length (and thus not a shorter run).

$yz = 0^j 1^i$, and $yz \in L$, since w.l.o.g. $i < j$, so the run of 1s following run of 0s is shorter.

Thus, z is a distinguishing suffix.

Hence, F is a fooling set for L.
Therefore, since F is infinite, L can't be regular. □

b) $S \to A \mid \varepsilon \mid B \mid C \mid AD$

$A \to 0A1 \mid 0 \mid 0B$

$B \to 0 \mid 0B \mid \varepsilon$

$C \to 1 \mid 1C \mid \varepsilon$

$D \to AD \mid \varepsilon$

A is where we build our uneven runs

B just builds 0s

C just builds 1s

D allows us to start more runs of 0s followed by 1s

CS/ECE 374 A ✧ Spring 2026 Midterm 1 Practice 2 Problem 5	Name: Sarah Dowden (sdowden2)

For each of the following languages L over the alphabet $\Sigma = \{0, 1\}$, describe a DFA that accepts L **and** give a regular expression that represents L. You do not need to justify your answers.

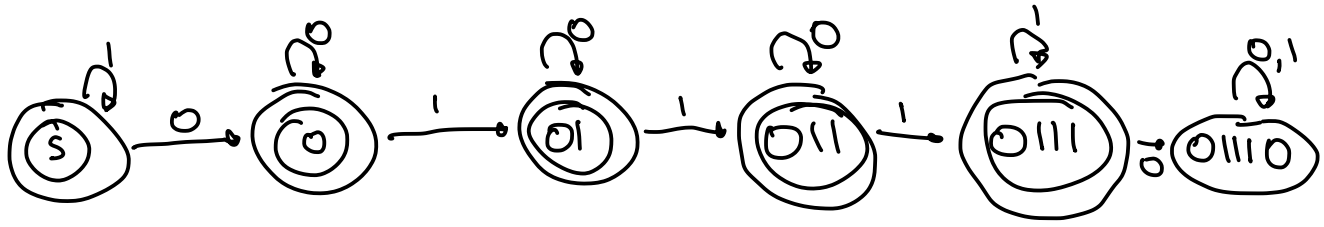
(a) Strings that do not contain the subsequence 01110.

(b) Strings that contain at least two even-length runs of 1s.

a) $1^* 0^* (10^* + 10^*10^* + 10^*10^*10^* + 1^*)$ (based on DFA)

regex:

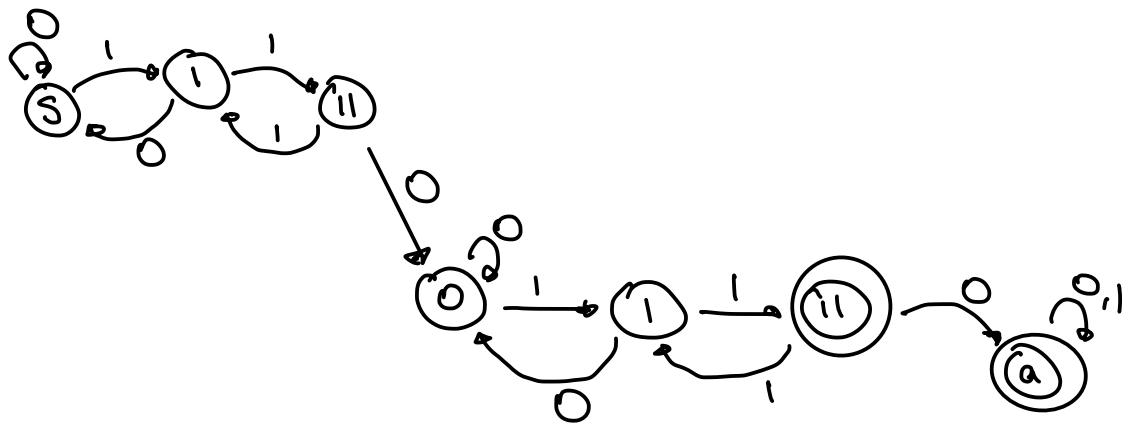
DFA:



---

b)

DFA:



regex: $0^* (0 + (11)^*10)^* (11)^* (0 + (11)^*10)^* 0 (11)^* (\varepsilon + 0(01)^*)$

- 0^*: can start with zeros
- $(0 + (11)^*10)^*$: build any string with only odd runs of 1, must end in 0
- $(11)^*$: first even run
- $(0 + (11)^*10)^*$: build any string with only odd runs of 1, must end in 0
- 0: must have 0 before second even run
- $(11)^*$: second even run
- $(\varepsilon + 0(01)^*)$: end there or must have 0 then any string

Sarah Dowden (sdowden2)

(scratch paper)

2b) undo the delete (essentially insert a run)

Given arbitrary DFA $M = (Q, \Sigma, \delta, s, A)$ which accepts L, I will create NFA $M' = (Q', \Sigma, \delta', s', A')$ to show regularity for language transformation.

guess which char to run with

$$Q' = Q \times \{y, n\} \times \{0, 1\} \cup \{fail\}$$

y: in initial run
n: not in initial run

$$s' = \{(s, y, 0), (s, y, 1)\}$$

initial run

$$\delta'((s, y, 0), \varepsilon) = \{(\delta(s, 0), y, 0)\}$$
$$\delta'((s, y, 1), \varepsilon) = \{(\delta(s, 1), y, 0)\}$$

$$\delta'((q, y, 0), \varepsilon) = \{(\delta(q, 0), y, 0)\}$$
$$\delta'((q, y, 1), \varepsilon) = \{(\delta(q, 1), y, 1)\}$$

switch to end run

$$\delta'((q, y, a), c) = \begin{cases} fail, & \text{if } a = c \\ (\delta(q, c), n, a) & \text{if } a \neq c \end{cases}$$

normally process rest of string

$$\delta'((q, n, a), c) = \{(\delta(q, c), n, a)\}$$

$$\delta'(fail, c) = \emptyset$$ (super clear fail is a dump state)

original not be in run; any character guess fine

$$A' = A \times \{n\} \times \{0, 1\}$$

M' nondeterministically guesses what character we should use for our initial (arbitrary length) run, and then makes sure it's not equal to first character read in, and then processes rest of string like normal.

Sarah Dowden (sdowden2)

(scratch paper)

Continuation of part 2a

- $w = aa$ for some char $a \in \{0,1\}$
  - $squish(w) = squish(aa)$
    $= 0$ (by def of squish)
  - $\#(1,w) \leq 2$ if $a = 1$
  - $\#(1, squish(w)) = \#(1,0) = 0$
  - Therefore $\#(1,squish(w)) \leq \#(1,w)$
- $w = abx$ for some chars $a,b \in \{0,1\}$ s.t. $a \neq b$, and some arbitrary string $x \in \{0,1\}^*$
  - $squish(w) = squish(abx)$
    $= squish(ab) \cdot squish(x)$ (by def of squish)
    $= 1 \cdot squish(x)$
  - $\#(1,w) = \#(1,abx)$
    $= \#(1,ab) + \#(1,x)$
    $= 1 + \#(1,x)$
    (because $a \neq b$ and $a,b \in \{0,1\}$, so either a or b must be a 1)
  - $\#(1, 1 \cdot squish(x)) = \#(1,1) + \#(1, squish(x))$
    $= 1 + \#(1, squish(x))$
  - By IH, $1 + \#(1,x) \geq 1 + \#(1, squish(x))$ since $|x| < |w|$
  - That becomes $\#(1,w) \geq 1 + \#(1, squish(x))$ by above

Proof for part a continued on page 8

Continuation of 2b

- $w = 11x$, $x \in \{0,1\}^*$
  - $\#(1,w) = \#(1,11x)$
    $= \#(1,11) + \#(1,x)$
    $= 2 + \#(1,x)$
  - $\#(1, squish(w)) = \#(1, squish(11x))$
    $= \#(1,11) + \#(1, squish(x))$
    $= 2 + \#(1, squish(x))$
  - By IH since $|x| < w$, then
    $\#(1, squish(x)) \bmod 2 = \#(1,x) \bmod 2$
    $2 + \#(1, squish(x)) \bmod 2 = 2 + \#(1,w) \bmod 2$
    $\#(1, squish(w)) \bmod 2 = \#(1,w)$
- $w = 00x$, $x \in \{0,1\}^*$
  - $\#(1,w) = \#(1,00x)$
    $= \#(1,00) + \#(1,x)$
    $= 0 + \#(1,x)$
    $= \#(1,x)$
  - $\#(1, squish(w)) = \#(1, squish(00x))$
    $= \#(1,0) + \#(1, squish(x))$
    $= 0 + \#(1, squish(x))$
  - By IH since $|x| < w$, then
    $\#(1, squish(x)) \bmod 2 = \#(1,x) \bmod 2$
    $\#(1, squish(w)) \bmod 2 = \#(1,w)$

Proof for part b continued on page 8

Sarah Dowden (sdowden2)

(scratch paper)

- $w = aax$, $a \in \{0,1\}$, $x \in \{0,1\}^*$
  - $squish(w) = squish(aax)$
    $= squish(aa) \cdot squish(x)$ (by def of squish)
    $= 0 \cdot squish(x)$ (by def of squish)
  - $\#(1, squish(w)) = \#(1,0) + \#(1, squish(x))$
    $= 0 + \#(1, squish(x))$
    $= \#(1, squish(x))$
  - $\#(1, w) = \#(1, aax)$
    $= \#(1, aa) + \#(1, x)$
    $\leq 2 + \#(1, x)$
  - By IH since $|x| < w$, we know
    $\#(1, x) \geq \#(1, squish(x))$
  - Then, $2 + \#(1,x) \geq \#(1, squish(x))$
    (the $\#(1,x)$ and $\#(1, squish(x))$ are equal)
    so, $\#(1,w) \geq \#(1, squish(w))$

In all cases, we conclude
$\#(1,w) \geq \#(1, squish(w))$ □

- $w = abx$, $a,b \in \{0,1\}$, $x \in \{0,1\}^*$, $a \neq b$
  - $\#(1,w) = \#(1,ab) + \#(1,x)$
    $= 1 + \#(1,x)$ (as a or b must be 1)
  - $\#(1, squish(w)) = \#(1, squish(abx))$
    $= \#(1, squish(ab))$
    $+ \#(1, squish(x))$
    $= 1 + \#(1, squish(x))$
  - By IH, since $|x| < |w|$, then we know $\#(1,x) \bmod 2 = \#(1, squish(x)) \bmod 2$
  - Then $1 + \#(1,x) \bmod 2 = 1 + \#(1, squish(x)) \bmod 2$
  - so we can rewrite
    $\#(1,w) \bmod 2 = \#(1, squish(w)) \bmod 2$

In all cases, we conclude
$\#(1,w) \bmod 2 = \#(1, squish(w)) \bmod 2$
□

(scratch paper)