CS/ECE 374 A ✧ Spring 2026

Homework 1

Due Tuesday, January 27, 2026 at 9pm Central Time

- **Submit your solutions electronically on Gradescope as PDF files.**
  - Submit a separate PDF file for each numbered problem.
  - Groups of up to three people can submit joint solutions. ***Exactly one*** student in each group should upload the solution to Gradescope and indicate their other group members.
  - You can find a LaTeX solution template on the course web site, which we encourage you to use to typeset your homework.
  - If you plan to submit scanned handwritten solutions, please use dark ink (not pencil) on white unlined paper (not notebook or graph paper), and use a scanner or a scanning app to create a high-quality PDF for submission (not a raw photo). We reserve the right to reject submissions that are difficult to read.
  - If you plan to use a tablet and a note-taking app, please make sure your submitted PDF is broken into standard US-letter sized pages (not a long scroll).
- **You may use any source at your disposal**—paper, human, or electronic—but you ***must*** cite every source that you use, and you ***must*** write everything yourself in your own words. You are responsible for all errors in your submissions. In particular:
  - Every lettered ***part*** of every submitted solution ***must*** include a list of all sources and collaborators (or the whole thing if it doesn't have lettered parts.) If you didn't consult any sources or collaborators, write "Sources and collaborators: None".
  - If you use any large language model for any purpose, you ***must*** include a brief explanation what you used the LLM to do.
- ***Standard grading rubrics*** for many problem types can be found on the course web page. For example, most problems in Homework 1 will be graded using our standard induction rubric. Please familiarize yourself with these rubrics *before* you submit your solutions.
- Each homework will include at least one fully ***solved problem***, similar to that week's assigned problems. These model solutions show the structure, presentation, and level of detail that we recommend in your homework solutions. (So do the lab solutions.) ***We strongly recommend reading them*** **before** ***submitting your homework solutions.***

See the course web site for more information.

If you have any questions about these policies,
please don't hesitate to ask in lecture, in labs, in office hours, or online.

---

1. For any string $w \in \{0,1\}^*$, let $sink(w)$ be the function defined recursively as follows:

$$sink(w) := \begin{cases} w & \text{if } |w| \le 1 \\ a \cdot sink(1x) & \text{if } w = 1ax \text{ for some } a \in \{0,1\} \text{ and } x \in \{0,1\}^* \\ 0 \cdot sink(ax) & \text{if } w = 0ax \text{ for some } a \in \{0,1\} \text{ and } x \in \{0,1\}^* \end{cases}$$

Let $\#(a, w)$ denote the number of times symbol a appears in string w; for example, $\#(0, 01000110111001) = \#(1, 01000110111001) = 7$.

(a) Prove that $|sink(w)| = |w|$.

(b) Prove that if $\#(1, w) \ge 1$, then $sink(w) = x \bullet 1$ for some string x (in other words, if w contains a 1, then $sink(w)$ ends with a 1.)

(c) Prove that if $sink(w) = x \bullet 1$ for some string x, then $\#(1, w) \ge 1$.

You may assume without proof that $\#(a, xy) = \#(a, x) + \#(a, y)$ for any symbol a and any strings x and y, or any other result proved in lecture, in lab, or in the lecture notes. In particular, you may also use the fact that concatenation is associative. Otherwise, your proofs must be formal and self-contained; in particular, they must invoke the recursive definitions of concatenation $\bullet$, length $|\cdot|$, and the *sink* function. Do not appeal to intuition!

2. Consider the following 3 sets of strings $L_0, L_1, L_2 \subseteq \{0,1\}^*$ defined (mutually) recursively as follows:

- The empty string ε is in L_0.
- For any $i \in \{0, 1, 2\}$ and any string $x \in L_i$, the string $0x$ is in $L_{i-1 \pmod 3}$ and the string $1x$ is in $L_{i+1 \pmod 3}$.
- These are the only strings in L_0, L_1, and L_2.

Again, let $\#(a, w)$ denote the number of times symbol a appears in string w.

(a) Prove that for each $i \in \{0, 1, 2\}$ and string $w \in L_i$, we have $\#(1, w) \equiv \#(0, w) + i \pmod 3$. (In other words, for every string $w \in L_i$, the number of 1s in w minus the number 0s has a remainder of i when divided by 3.)

(b) Prove that for each $i \in \{0, 1, 2\}$, the set L_i contains every string $w \in \{0,1\}^*$ such that $\#(1, w) \equiv \#(0, w) + i \pmod 3$.

Again, you may assume without proof that $\#(a, xy) = \#(a, x) + \#(a, y)$ for any symbol a and any strings x and y, or any other result proved in lecture, in lab, or in the lecture notes. Otherwise, your proofs must be formal and self-contained; in particular, they must invoke the recursive definition of the sets L_i. Do not appeal to intuition!

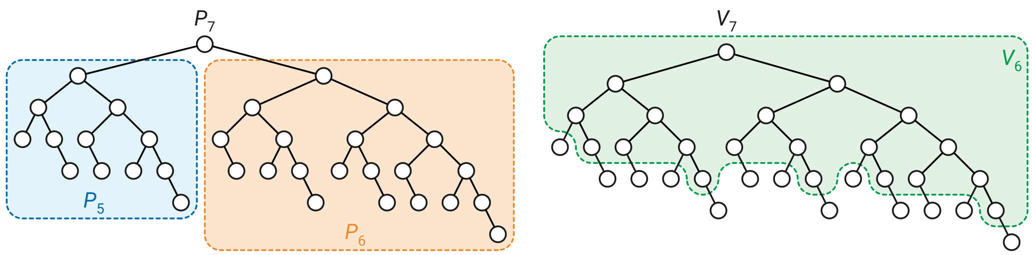
[Hint: Each part can be solved using a single inductive hypothesis that appeals to all three sets at once.]

*3. ***Practice only. Do not submit solutions.***

For each non-negative integer n, we recursively define two binary trees P_n and V_n, called the nth *Piṅgala tree* and the nth *Virahāṇka tree*, respectively.

- P_0 and V_0 are empty trees, with no nodes.
- P_1 and V_1 each consist of a single node.
- For any integer $n \geq 2$, the tree P_n consists of a root with two subtrees; the left subtree is a copy of P_{n-2}, and the right subtree is a copy of P_{n-1}.
- For any integer $n \geq 2$, the tree V_n is obtained from V_{n-1} by attaching a new right child to every leaf and attaching a new left child to every node that has only a right child.

The following figure shows the recursive construction of these two trees when $n = 7$.



Recall that the Fibonacci numbers are defined recursively as follows:

$$F_n = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ F_{n-1} + F_{n-2} & \text{otherwise} \end{cases}$$

(a) Prove that the tree P_n has exactly F_n leaves.

(b) Prove that the tree V_n has exactly F_n leaves. *[Hint: You need to prove a stronger result.]*

(c) Prove that the trees P_n and V_n are identical, for all $n \geq 0$.

[Hint: The hardest part of these proofs is developing the right language and notation.]

As in earlier problems, you may freely use any result that proved in lecture, in lab, or in the lecture notes. Otherwise your proofs must be formal and self-contained; in particular, they must invoke the recursive definitions of the trees P_n and V_n and the Fibonacci numbers F_n.

Solved Problems

4. For any string $w \in \{0,1\}^*$, let $swap(w)$ denote the string obtained from w by swapping the first and second symbols, the third and fourth symbols, and so on. For example:

$$swap(10\,11\,00\,01\,10\,1) = 01\,11\,00\,10\,01\,1.$$

The *swap* function can be formally defined as follows:

$$swap(w) := \begin{cases} \varepsilon & \text{if } w = \varepsilon \\ w & \text{if } w = 0 \text{ or } w = 1 \\ ba \bullet swap(x) & \text{if } w = abx \text{ for some } a, b \in \{0,1\} \text{ and } x \in \{0,1\}^* \end{cases}$$

(a) Prove that $|swap(w)| = |w|$ for every string w.

Solution: Let w be an arbitrary string.
Assume $|swap(x)| = |x|$ for every string x that is shorter than w.
There are three cases to consider (mirroring the definition of *swap*):

- If $w = \varepsilon$, then

$$\begin{aligned} |swap(w)| &= |swap(\varepsilon)| && \text{because } w = \varepsilon \\ &= |\varepsilon| && \text{by definition of } swap \\ &= |w| && \text{because } w = \varepsilon \end{aligned}$$

- If $w = 0$ or $w = 1$, then

$$|swap(w)| = |w| \qquad \text{by definition of } swap$$

- Finally, if $w = abx$ for some $a, b \in \{0,1\}$ and $x \in \{0,1\}^*$, then

$$\begin{aligned} |swap(w)| &= |swap(abx)| && \text{because } w = abx \\ &= |ba \bullet swap(x)| && \text{by definition of } swap \\ &= |ba| + |swap(x)| && \text{because } |y \bullet z| = |y| + |z| \\ &= |ba| + |x| && \text{by the induction hypothesis} \\ &= 2 + |x| && \text{by definition of } |\cdot| \\ &= |ab| + |x| && \text{by definition of } |\cdot| \\ &= |ab \bullet x| && \text{because } |y \bullet z| = |y| + |z| \\ &= |abx| && \text{by definition of } \bullet \\ &= |w| && \text{because } w = abx \end{aligned}$$

In all cases, we conclude that $|swap(w)| = |w|$. ■

Rubric: 5 points: Standard induction rubric (scaled). This is more detail than necessary for full credit.

(b) Prove that $swap(swap(w)) = w$ for every string w.

Solution: Let w be an arbitrary string.
Assume $swap(swap(x)) = x$ for every string x that is shorter than w.
There are three cases to consider (mirroring the definition of *swap*):

- If $w = \varepsilon$, then

$$\begin{aligned} swap(swap(w)) &= swap(swap(\varepsilon)) && \text{because } w = \varepsilon \\ &= swap(\varepsilon) && \text{by definition of } swap \\ &= \varepsilon && \text{by definition of } swap \\ &= w && \text{because } w = \varepsilon \end{aligned}$$

- If $w = 0$ or $w = 1$, then

$$\begin{aligned} swap(swap(w)) &= swap(w) && \text{by definition of } swap \\ &= w && \text{by definition of } swap \end{aligned}$$

- Finally, if $w = abx$ for some $a, b \in \{0, 1\}$ and $x \in \{0, 1\}^*$, then

$$\begin{aligned} swap(swap(w)) &= swap(swap(abx)) && \text{because } w = abx \\ &= swap(ba \bullet swap(x)) && \text{by definition of } swap \\ &= swap(ba \bullet z) && \text{where } z = swap(x) \\ &= swap(baz) && \text{by definition of } \bullet \\ &= ab \bullet swap(z) && \text{by definition of } swap \\ &= ab \bullet swap(swap(x)) && \text{because } z = swap(x) \\ &= ab \bullet x && \text{by the induction hypothesis} \\ &= abx && \text{by definition of } \bullet \\ &= w && \text{because } w = abx \end{aligned}$$

In all cases, we conclude that $swap(swap(w)) = w$. ■

Rubric: 5 points: Standard induction rubric (scaled). This is more detail than necessary for full credit.

5. The ***reversal*** w^R of a string w is defined recursively as follows:

$$w^R := \begin{cases} \varepsilon & \text{if } w = \varepsilon \\ x^R \bullet a & \text{if } w = a \cdot x \end{cases}$$

A ***palindrome*** is any string that is equal to its reversal, like AMANAPLANACANALPANAMA, RACECAR, POOP, I, and the empty string.

(a) Give a recursive definition of a palindrome over the alphabet Σ.

Solution: A string $w \in \Sigma^*$ is a palindrome if and only if either

- $w = \varepsilon$, or
- $w = a$ for some symbol $a \in \Sigma$, or
- $w = axa$ for some symbol $a \in \Sigma$ and some *palindrome* $x \in \Sigma^*$.

■

Rubric: 2 points = ½ for each base case + 1 for the recursive case. No credit for the rest of the problem unless this part is correct.

(b) Prove $w = w^R$ for every palindrome w (according to your recursive definition).

You may assume the following facts about all strings x, y, and z:

- Reversal reversal: $(x^R)^R = x$
- Concatenation reversal: $(x \bullet y)^R = y^R \bullet x^R$
- Right cancellation: If $x \bullet z = y \bullet z$, then $x = y$.

Solution: Let w be an arbitrary palindrome.

Assume that $x = x^R$ for every palindrome x such that $|x| < |w|$.

There are three cases to consider (mirroring the definition of "palindrome"):

- If $w = \varepsilon$, then $w^R = \varepsilon$ by definition, so $w = w^R$.
- If $w = a$ for some symbol $a \in \Sigma$, then $w^R = a$ by definition, so $w = w^R$.
- Finally, if $w = axa$ for some symbol $a \in \Sigma$ and some palindrome $x \in P$, then

$$\begin{aligned} w^R &= (a \cdot x \bullet a)^R && \text{because w = axa} \\ &= (x \bullet a)^R \bullet a && \text{by definition of reversal} \\ &= a^R \bullet x^R \bullet a && \text{by concatenation reversal} \\ &= a \bullet x^R \bullet a && \text{by definition of reversal} \\ &= a \bullet x \bullet a && \text{by the inductive hypothesis} \\ &= w && \text{because w = axa} \end{aligned}$$

In all three cases, we conclude that $w = w^R$. ■

Rubric: 4 points: standard induction rubric (scaled)

(c) Prove that every string w such that $w = w^R$ is a palindrome (according to your recursive definition).

Again, you may assume the following facts about all strings x, y, and z:

- Reversal reversal: $(x^R)^R = x$
- Concatenation reversal: $(x \bullet y)^R = y^R \bullet x^R$
- Right cancellation: If $x \bullet z = y \bullet z$, then $x = y$.

Solution: Let w be an arbitrary string such that $w = w^R$.

Assume that every string x such that $|x| < |w|$ and $x = x^R$ is a palindrome.

There are three cases to consider (mirroring the definition of "palindrome"):

- If $w = \varepsilon$, then w is a palindrome by definition.
- If $w = a$ for some symbol $a \in \Sigma$, then w is a palindrome by definition.
- Otherwise, we have $w = ax$ for some symbol a and some *non-empty* string x.
  The definition of reversal implies that $w^R = (ax)^R = x^R a$.
  Because x is non-empty, its reversal x^R is also non-empty.
  Thus, $x^R = by$ for some symbol b and some string y.
  It follows that $w^R = bya$, and therefore $w = (w^R)^R = (bya)^R = ay^R b$.

  ⟨⟨At this point, we need to prove that $a = b$ and that y is a palindrome.⟩⟩

  Our assumption that $w = w^R$ implies that $bya = ay^R b$.
  The recursive definition of string equality immediately implies $a = b$.

  Because $a = b$, we have $w = ay^R a$ and $w^R = aya$.
  The recursive definition of string equality implies $y^R a = ya$.
  Right cancellation implies $y^R = y$.
  The inductive hypothesis now implies that y is a palindrome.

  We conclude that w is a palindrome by definition.

In all three cases, we conclude that w is a palindrome. ■

Rubric: 4 points: standard induction rubric (scaled).

6. Let $L \subseteq \{0, 1\}^*$ be the language defined recursively as follows:

- The empty string ε is in L.
- For any string $x \in L$, the strings $0101x$ and $1010x$ are also in L.
- For all strings x and y such that $xy \in L$, the strings $x00y$ and $x11y$ are also in L. (In other words, inserting two consecutive 0s or two consecutive 1s anywhere in a string in L yields another string in L.)
- These are the only strings in L.

Let EE denote the set of all strings $w \in \{0, 1\}^*$ such that $\#(0, w)$ and $\#(1, w)$ are both even.

(a) Prove that $L \subseteq EE$.

Solution: Let w be an arbitrary string in L. We need to prove that $\#(0, w)$ and $\#(1, w)$ are both even. Here I will prove only that $\#(0, w)$ is even; the proof that $\#(1, w)$ is even is symmetric.

Assume for every string $x \in L$ such that $|x| < |w|$ that $\#(0, x)$ is even.

There are several cases to consider, mirroring the definition of L.

- Suppose $w = \varepsilon$. Then $\#(0, w) = 0$, and 0 is even.
- Suppose $w = 0101x$ or $w = 1010x$ for some string $x \in L$. The definition of $\#$ (applied four times) implies $\#(0, w) = \#(0, x) + 2$. The inductive hypothesis implies $\#(0, x)$ is even. We conclude that $\#(0, w)$ is even.
- Suppose $w = x00y$ for some strings x and y such that $xy \in L$. Then

$$\begin{aligned}\#(0, w) &= \#(0, x00y)\\ &= \#(0, x) + \#(0, 00) + \#(0, y)\\ &= \#(0, x) + \#(0, y) + \#(0, 00)\\ &= \#(0, xy) + 2\end{aligned}$$

  The induction hypothesis implies $\#(0, xy)$ is even. We conclude that $\#(0, w) = \#(0, xy) + 2$ is also even.
- Finally, suppose $w = x11y$ for some strings x and y such that $xy \in L$. Then

$$\begin{aligned}\#(0, w) &= \#(0, x11y)\\ &= \#(0, x) + \#(0, 11) + \#(0, y)\\ &= \#(0, x) + \#(0, y)\\ &= \#(0, xy)\end{aligned}$$

  The induction hypothesis implies $\#(0, w) = \#(0, xy)$ is even.

In all cases, we have shown that $\#(0, w)$ is even. Symmetric arguments imply that $\#(1, w)$ is even. We conclude that $w \in EE$. ■

Rubric: 5 points: standard induction rubric (scaled). Yes, this is enough detail for $\#(1, w)$. If explicit proofs are given for both $\#(0, w)$ and $\#(1, w)$, grade them independently, each for 2½ points.

(b) Prove that $EE \subseteq L$.

Solution: Let w be an arbitrary string in EE. We need to prove that $w \in L$.

Assume that for every string $x \in EE$ such that $|x| < |w|$, we have $x \in L$.

There are four (overlapping) cases to consider, depending on the first four symbols in w.

- Suppose $|w| < 4$. Then w must be one of the strings ε, 00, or 11; brute force inspection implies that every other string of length at most 3 (0, 1, 01, 10, 000, 001, 010, 011, 100, 101, 110, 111) has an odd number of 0s or an odd number of 1s (or both). All three strings ε, 00, and 11 are in L. In all other cases, we can assume that $|w| \geq 4$, so the "first four symbols of w" are well-defined.
- Suppose the first four symbols of w are 0000 or 0001 or 0010 or 0011 or 0100 or 1000 or 1001 or 1100. Then $w = x00y$ for some (possibly empty) strings x and y. Arguments in part (a) imply that $\#(0, xy) = \#(0, w) - 2$ and $\#(1, xy) = \#(1, w)$ are both even. Thus $xy \in EE$ by definition of EE. So the induction hypothesis implies $xy \in L$. We conclude that $w = x00y \in L$ by definition of L.
- Suppose the first four symbols of w are 0011 or 0110 or 0111 or 1011 or 1100 or 1101 or 1110 or 1111.) After swapping 0s and 1s, the argument in the previous case implies that $w \in L$.
- Finally, suppose the first four symbols of w are 0101 or 1010; in other words, suppose $w = 0101x$ or $w = 1010x$ for some (possibly empty) string x. Then $\#(0, x) = \#(0, w) - 2$ and $\#(1, x) = \#(1, w) - 2$ are both even, so $x \in EE$ by definition. The induction hypothesis implies $x \in L$. We conclude that $w \in L$ by definition of L.

Each of the 16 possible choices for the first four symbols of w is considered in at least one of the last three cases.

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

Rubric: 5 points: standard induction rubric (scaled). This is not the only correct proof. This is not the only correct way to express this particular case analysis.