CS/ECE 374 A ✧ Fall 2023

Homework 1

Due Tuesday, August 29, 2023 at 9pm Central Time

- **Submit your written solutions electronically to Gradescope as PDF files.** Submit a separate PDF file for each numbered problem. If you plan to typeset your solutions, you are welcome to use the LaTeX solution template on the course web site. If you must submit scanned handwritten solutions, please use a black pen on blank white paper and a high-quality scanner app (or an actual scanner).
- Groups of up to three people can submit joint solutions on Gradescope. ***Exactly*** one student in each group should upload the solution and indicate their other group members.
- **You may use any source at your disposal**—paper, electronic,[1] or human—but you ***must*** cite ***every*** source that you use,[2] you ***must*** write everything yourself in your own words, and you are responsible for any errors in the sources you use.[3] See the academic integrity policies on the course web site for more details.
- Written homework is normally due every Tuesday at 9pm. In addition, guided problem sets on PrairieLearn are normally due every **Monday** at 9pm; each student must do these individually. In particular, Guided Problem Set 1 is due Monday, August 28!
- Both guided problem sets and homework may be submitted up to 24 hours late for 50% partial credit, or for full credit with an approved extension. See the grading policies on the course web site for more details.
- Each homework will include at least one fully solved problem, similar to that week's assigned problems, together with the rubric we would use to grade this problem if it appeared in an actual homework or exam. These model solutions show our recommendations for structure, presentation, and level of detail in your homework solutions. (Obviously, the actual *content* of your solutions won't match the model solutions, because your problems are different!) Homeworks may also include additional practice problems.
- Standard grading rubrics for many problem types can be found on the course web page. For example, the problems in this week's homework will be graded using the standard induction rubric. (Weak induction makes the baby Jesus cry.)

See the course web site for more information.

If you have any questions about these policies,
please don't hesitate to ask in class, in office hours, or on Piazza.

---

[1]Yes, including ChatGPT.
[2]Yes, including ChatGPT.
[3]Yes, including ChatGPT.

1. Consider the following recursively defined function:

$$stutter(w) := \begin{cases} \varepsilon & \text{if } w = \varepsilon \\ aa \bullet stutter(x) & \text{if } w = ax \end{cases}$$

For example, *stutter*(MISSISSIPPI) = MMIISSSSIISSSSIIPPPPII.

(a) Prove that $|stutter(w)| = 2|w|$ for every string w.

(b) Prove that $stutter(x \bullet y) = stutter(x) \bullet stutter(y)$ for all strings x and y.

(c) ***Practice only. Do not submit solutions.***
The ***reversal*** $\boldsymbol{w^R}$ of a string w is defined recursively as follows:

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

For example, $\texttt{MISSIPPIPPI}^R = \texttt{IPPIPPISSIM}$.
Prove that $stutter(w)^R = stutter(w^R)$ for every string w.

You may freely use any result proved in lecture, in lab, or in the lecture notes. Otherwise your proofs must be formal and self-contained. In particular, your proofs *must* invoke the formal recursive definitions of string length and concatenation (and for part (c), reversal).

2. For each positive integer n, we define two strings p_n and v_n, respectively called the nth *Piṅgala string* and the nth *Virahāṇka string*. Piṅgala strings are defined by the following recurrence:

$$p_n = \begin{cases} \texttt{1} & \text{if } n = 1 \\ \texttt{0} & \text{if } n = 2 \\ p_{n-2} \bullet p_{n-1} & \text{otherwise} \end{cases}$$

For example:

$$p_7 = \underbrace{\texttt{10010}}_{p_5}\overbrace{\underbrace{\texttt{010}}_{p_4}\underbrace{\texttt{10010}}_{p_5}}^{p_6}.$$

Virahāṇka strings are defined more indirectly as

$$v_n = \begin{cases} \texttt{1} & \text{if } n = 1 \\ \mathsf{grow}(v_{n-1}) & \text{otherwise} \end{cases}$$

where the string function grow is defined as follows:

$$\mathsf{grow}(w) = \begin{cases} \varepsilon & \text{if } w = \varepsilon \\ \texttt{0} \cdot \mathsf{grow}(x) & \text{if } w = \texttt{1}x \\ \texttt{10} \bullet \mathsf{grow}(x) & \text{if } w = \texttt{0}x \end{cases}$$

For example:

$$\mathsf{grow}(\texttt{01010010}) = \texttt{10} \bullet \texttt{0} \bullet \texttt{10} \bullet \texttt{0} \bullet \texttt{10} \bullet \texttt{10} \bullet \texttt{0} \bullet \texttt{10} = \texttt{1001001010010}$$

Finally, 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 $|p_n| = F_n$ for all $n \geq 1$.

(b) Prove that $\mathsf{grow}(w \bullet z) = \mathsf{grow}(w) \bullet \mathsf{grow}(z)$ for all strings w and z.

(c) Prove that $p_n = v_n$ for all $n \geq 1$. *[Hint: Careful!]*

(d) ***Practice only. Do not submit solutions.***
Prove that $|v_n| = F_n$ for all $n \geq 1$.

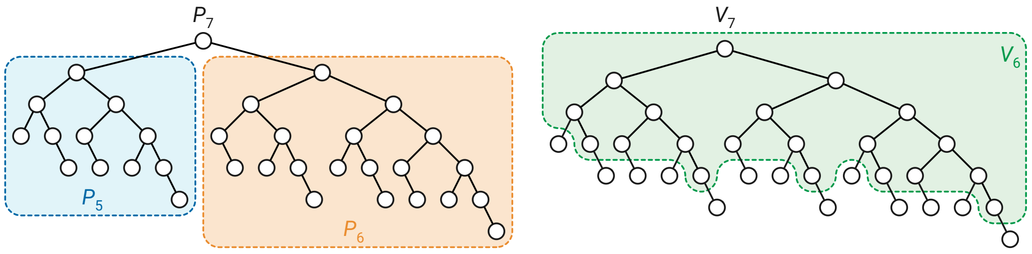
As in problem 1, you may freely use any result that proved in lecture, in lab, or in the lecture notes. Otherwise your proofs must be formal and self-contained. In particular, your proofs *must* invoke the formal recursive definitions of the strings p_n and v_n, the grow function, and the Fibonacci numbers F_n.

*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-1}, and the right subtree is a copy of P_{n-2}.
- For any integer $n \geq 2$, the tree L_n is obtained from L_{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 this proof is developing the right language/notation.]

As in problem 1, 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, your proofs *must* invoke the formal recursive definitions of the trees P_n and V_n and the Fibonacci numbers F_n.

Solved Problems

3. 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 $\mathit{swap}(\mathit{swap}(w)) = w$ for every string w.

Solution: Let w be an arbitrary string.

Assume $\mathit{swap}(\mathit{swap}(x)) = x$ for every string x that is shorter than w.

There are three cases to consider (mirroring the definition of *swap*):

- If $w = \varepsilon$, then

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

- If $w = \mathtt{0}$ or $w = \mathtt{1}$, then

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

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

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

In all cases, we conclude that $\mathit{swap}(\mathit{swap}(w)) = w$. ■

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

4. 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).

5. 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.

In the following proofs, you may freely use any result proved in lecture, in lab, in the lecture notes, or earlier in your homework. Otherwise your proofs must be formal and self-contained; in particular, they must invoke the formal recursive definitions of # and L.

(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 $\#(\mathtt{0}, w)$ is even. Symmetric arguments imply that $\#(\mathtt{1}, w)$ is even. We conclude that $w \in EE$. ■

Rubric: 5 points: standard induction rubric (scaled). Yes, this is enough detail for $\#(\mathtt{1}, w)$. If explicit proofs are given for both $\#(\mathtt{0}, w)$ and $\#(\mathtt{1}, w)$, 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, $\mathtt{00}$, or $\mathtt{11}$; brute force inspection implies that every other string of length at most 3 ($\mathtt{0}$, $\mathtt{1}$, $\mathtt{01}$, $\mathtt{10}$, $\mathtt{000}$, $\mathtt{001}$, $\mathtt{010}$, $\mathtt{011}$, $\mathtt{100}$, $\mathtt{101}$, $\mathtt{110}$, $\mathtt{111}$) has an odd number of $\mathtt{0}$s or an odd number of $\mathtt{1}$s (or both). All three strings ε, $\mathtt{00}$, and $\mathtt{11}$ 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 $\mathtt{00\underline{00}}$ or $\mathtt{0\underline{00}1}$ or $\mathtt{\underline{00}10}$ or $\mathtt{\underline{00}11}$ or $\mathtt{01\underline{00}}$ or $\mathtt{1\underline{00}0}$ or $\mathtt{1\underline{00}1}$ or $\mathtt{11\underline{00}}$. Then $w = x\mathtt{00}y$ for some (possibly empty) strings x and y. Arguments in part (a) imply that $\#(\mathtt{0}, xy) = \#(\mathtt{0}, w) - 2$ and $\#(\mathtt{1}, xy) = \#(\mathtt{1}, w)$ are both even. Thus $xy \in EE$ by definition of EE. So the induction hypothesis implies $xy \in L$. We conclude that $w = x\mathtt{00}y \in L$ by definition of L.
- Suppose the first four symbols of w are $\mathtt{00\underline{11}}$ or $\mathtt{0\underline{11}0}$ or $\mathtt{0\underline{11}1}$ or $\mathtt{10\underline{11}}$ or $\mathtt{\underline{11}00}$ or $\mathtt{\underline{11}01}$ or $\mathtt{1\underline{11}0}$ or $\mathtt{\underline{11}11}$.) After swapping $\mathtt{0}$s and $\mathtt{1}$s, the argument in the previous case implies that $w \in L$.
- Finally, suppose the first four symbols of w are $\mathtt{0101}$ or $\mathtt{1010}$; in other words, suppose $w = \mathtt{0101}x$ or $w = \mathtt{1010}x$ for some (possibly empty) string x. Then $\#(\mathtt{0}, x) = \#(\mathtt{0}, w) - 2$ and $\#(\mathtt{1}, x) = \#(\mathtt{1}, w) - 2$ 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.