CS/ECE 374 A ✧ Fall 2021

Homework 1

Due Tuesday, August 31, 2021 at 8pm 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, please 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. All group members must be already registered on Gradescope.

- You are ***not*** required to sign up on Gradescope or Piazza with your real name and your illinois.edu email address; you may use any email address and alias of your choice. However, to give you credit for the homework, we need to know who Gradescope thinks you are. **Please fill out the web form linked from the course web page.**

- **You may use any source at your disposal**—paper, electronic, or human—but you ***must*** cite ***every*** source that you use, and you ***must*** write everything yourself in your own words. See the academic integrity policies on the course web site for more details.

- Written homework will be due every Tuesday at 8pm, except in weeks with exams. In addition, guided problems sets on PrairieLearn are due every **Monday** at 8pm; each student must do these individually. In particular, Guided Problem Set 1 is due Monday, August 30! Each Guided Problem Set has the same weight as one numbered homework problem.

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. Consider the following pair of mutually recursive functions on strings:

$$odds(w) := \begin{cases} \varepsilon & \text{if } w = \varepsilon \\ a \cdot evens(x) & \text{if } w = ax \end{cases} \qquad evens(w) := \begin{cases} \varepsilon & \text{if } w = \varepsilon \\ odds(x) & \text{if } w = ax \end{cases}$$

For example, the following derivation shows that $evens(\texttt{PARITY}) = \texttt{AIY}$:

$$\begin{aligned} evens(\texttt{PARITY}) &= odds(\texttt{ARITY}) \\ &= \texttt{A} \cdot evens(\texttt{RITY}) \\ &= \texttt{A} \cdot odds(\texttt{ITY}) \\ &= \texttt{A} \cdot (\texttt{I} \cdot evens(\texttt{TY})) \\ &= \texttt{A} \cdot (\texttt{I} \cdot odds(\texttt{Y})) \\ &= \texttt{A} \cdot (\texttt{I} \cdot (\texttt{Y} \cdot evens(\varepsilon))) \\ &= \texttt{A} \cdot (\texttt{I} \cdot (\texttt{Y} \cdot \varepsilon)) \\ &= \texttt{AIY} \end{aligned}$$

A similar derivation implies that $odds(\texttt{PARITY}) = \texttt{PRT}$.

(a) Give a self-contained recursive definition for the function *evens* that does not involve the function *odds*.

(b) Prove the following identity for all strings w and x:

$$evens(w \bullet x) = \begin{cases} evens(w) \bullet evens(x) & \text{if } |w| \text{ is even,} \\ evens(w) \bullet odds(x) & \text{if } |w| \text{ is odd.} \end{cases}$$

You may assume without proof any result proved in class, in lab, or in the lecture notes. Otherwise, your proofs must be formal and self-contained, and they must invoke the *formal* definitions of concatenation $\bullet$, length $|\cdot|$, and the *evens* and *odds* functions. Do not appeal to intuition!

2. Consider the following recursive function that perfectly shuffles two strings together:

$$shuffle(w,z) := \begin{cases} z & \text{if } w = \varepsilon \\ a \cdot shuffle(z,x) & \text{if } w = ax \end{cases}$$

For example, the following derivation shows that $shuffle(\texttt{PRT},\texttt{AIY}) = \texttt{PARITY}$:

$$\begin{aligned} shuffle(\texttt{PRT},\texttt{AIY}) &= \texttt{P} \cdot shuffle(\texttt{AIY},\texttt{RT}) \\ &= \texttt{P} \cdot (\texttt{A} \cdot shuffle(\texttt{RT},\texttt{IY})) \\ &= \texttt{P} \cdot (\texttt{A} \cdot (\texttt{R} \cdot shuffle(\texttt{IY},\texttt{T}))) \\ &= \texttt{P} \cdot (\texttt{A} \cdot (\texttt{R} \cdot (\texttt{I} \cdot shuffle(\texttt{T},\texttt{Y})))) \\ &= \texttt{P} \cdot (\texttt{A} \cdot (\texttt{R} \cdot (\texttt{I} \cdot (\texttt{T} \cdot shuffle(\texttt{Y},\varepsilon))))) \\ &= \texttt{P} \cdot (\texttt{A} \cdot (\texttt{R} \cdot (\texttt{I} \cdot (\texttt{T} \cdot (\texttt{Y} \cdot shuffle(\varepsilon,\varepsilon)))))) \\ &= \texttt{P} \cdot (\texttt{A} \cdot (\texttt{R} \cdot (\texttt{I} \cdot (\texttt{T} \cdot (\texttt{Y} \cdot \varepsilon))))) \\ &= \texttt{PARITY} \end{aligned}$$

(a) Prove that $shuffle(odds(w), evens(w)) = w$ for every string w.

(b) Prove $evens(shuffle(w,z)) = z$ for all strings w and z such that $|w| = |z|$.

You may assume without proof any result proved in class, in lab, or in the lecture notes. Otherwise, your proofs must be formal and self-contained, and they must invoke the *formal* definitions of concatenation • and the functions *shuffle*, *evens*, and *odds*. Do not appeal to intuition!

Rubrics

We will announce standard grading rubrics for common question types, which we will apply on all homeworks and exams. However, please remember that some homework and exam questions may fall outside the scope of these standard rubrics.

Standard induction rubric. For problems worth 10 points:

- \+ 1 for explicitly considering an *arbitrary* object.
- \+ 2 for a valid ***strong*** induction hypothesis
  - **Deadly Sin!** No credit here for stating a weak induction hypothesis, unless the rest of the proof is *absolutely perfect*.
- \+ 2 for explicit exhaustive case analysis
  - No credit here if the case analysis omits an infinite number of objects. (For example: all odd-length palindromes.)
  - −1 if the case analysis omits an finite number of objects. (For example: the empty string.)
  - −1 for making the reader infer the case conditions. Spell them out!
  - No penalty if the cases overlap (for example: even length at least 2, odd length at least 3, and length at most 5.)
- \+ 1 for cases that do not invoke the inductive hypothesis ("base cases")
  - No credit here if one or more "base cases" are missing.
- \+ 2 for correctly applying the ***stated*** inductive hypothesis
  - No credit here for applying a ***different*** inductive hypothesis, even if that different inductive hypothesis would be valid.
- \+ 2 for other details in cases that invoke the inductive hypothesis ("inductive cases")
  - No credit here if one or more "inductive cases" are missing.

For (sub)problems worth less than 10 points, scale and round to the nearest half-integer.

Solved Problems

Each homework assignment will include at least one fully solved problem, similar to the problems assigned in that homework, together with the grading rubric we would apply if *this problem appeared on a homework or exam. These model solutions illustrate our recommendations for structure, presentation, and level of detail in your homework solutions. Of course, the actual* **content** *of your solutions won't match the model solutions, because your problems are different!*

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 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:

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

(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 $w = \texttt{0}$ or $w = \texttt{1}$, then

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

- Finally, if $w = abx$ for some $a, b \in \{\texttt{0}, \texttt{1}\}$ and $x \in \{\texttt{0}, \texttt{1}\}^*$, 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 $w = \mathtt{0}$ or $w = \mathtt{1}$, 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 $a, b \in \{\mathtt{0}, \mathtt{1}\}$ and $x \in \{\mathtt{0}, \mathtt{1}\}^*$, 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*** $\boldsymbol{w^R}$ of a string w is defined recursively as follows:

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

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

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

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 that $y^R = y$.
  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).