CSCI 432/532, Spring 2025

Homework 1

Due Monday, January 28, 2025 at 9am Mountain Time

Submission Requirements

- Type or clearly hand-write your solutions into a PDF format so that they are legible and professional. Submit your PDF to the appropriate Canvas dropbox.
- Do not submit your first draft. Type or clearly re-write your solutions for your final submission.
- You may work with a group of up to three students and submit ***one single document*** for the group. Just be sure to list all group members at the top of the document.
- When possible, the 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!)

Academic Integrity

Remember, you may access ***any*** resource in preparing your solution to the homework. However, you must

- write your solutions in your own words, and
- credit every resource you use (for example: "Bob Smith helped me on problem 2. He took this course at UM in Fall 2020"; "I found a solution to a problem similar to this one in the lecture notes for a different course, found at this link: www.profzeno.com/agreatclass/lecture10"; "I asked ChatGPT how to solve problem 1 part (c); "I put my solution for problem 1 part (c) into ChatGPT to check that it was correct and it caught a missing case and suggested some grammer fixes.") If you use the provided LaTeX template, you can use the `sources` environment for this. Ask if you need help!

Grading Rubrics

For the recursive function definition:

Definition rubric. 2 points =

+ 1 For all correct base cases

+ 1 For all correct recursive cases.

No credit for the rest of the problem unless this part is correct.

For each induction proof (there are two on this homework):

Induction rubric. 10 points =

+ 1 for explicitly considering an arbitrary object
+ 2 for an explicit valid induction hypothesis
  - Yes, you need to write it down. Yes, even if it's "obvious". Remember that the goal of the homework is to communicate with people who aren't as clever as you.
+ 2 for explicit exhaustive case analysis
  - No credit here if the case analysis omits an infinite number of objects. (For example: all oddlength palindromes.)
  - −1 if the case analysis omits a 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 proof of 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 the regular expression problems:

Regular expression rubric. 10 points =

+ 2 for a syntactically correct regular expression.
+ 4 for a brief English explanation of your regular expression. This is how you argue that your regular expression is correct.
  - For longer expressions, you should explain each of the major components of your expression, and separately explain how those components fit together.
  - We do not want a transcription; don't just translate the regular-expression notation into English.
  - Yes, you need to write it down. Yes, even if it's "obvious". Remember that the goal of the homework is to communicate with people who aren't as clever as you.
+ 4 for correctness.
  - −4 for incorrectly answering $\emptyset$ or Σ^*.
  - −1 for a single mistake: one typo, excluding exactly one string in the target language, or including exactly one string not in the target language. (The incorrectly handled string is almost always the empty string ε.)
  - −2 for incorrectly including/excluding more than one but a finite number of strings.
  - −4 for incorrectly including/excluding an infinite number of strings.

- Regular expressions that are more complex than necessary may be penalized. Regular expressions that are significantly too complex may get no credit at all. On the other hand, minimal regular expressions are not required for full credit.

1. Given a string w and a symbol a, let $delete(a, w)$ be the string w with all instances of a removed. For example, $delete$(`z`, `jazzy`) is `jay` and $delete$(`1`,`00101110`) is `0000`.

   (a) (2 points) Write a recursive function that computes $delete(a, w)$.

   (b) (4 points) For strings x and y and symbol a, prove that $delete(a, x \bullet y) = delete(a, x) \bullet delete(a, y)$.

   (c) (4 points) Recall the function $\#(a, w)$ from problem session 1, which returns the number of occurrences of the symbol a in string w. For string $w \in \{0, 1\}^*$, prove that $|delete(1, w)| = \#(0, w)$.

2. For each of the following languages over the alphabet $\Sigma = \{0, 1\}$, give an equivalent regular expression, and briefly explain why your regular expression is correct. Note that there are infinitely many correct answers for each language.

   (a) (3 points) All strings that begin with the prefix `001`, end with the suffix `100`, and contain an odd number of `1`'s.

   (b) (3 points) All strings that contain both `0011` and `1100` as substrings.

   (c) (4 points) All strings that contain the substring `01` an odd number of times.

Solved Problems

1. 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 $w = \mathtt{0}$ 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.

2. 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 = + 1 for base cases and + 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).