1. The famous Czech professor Jiřina Z. Džunglová has a favorite 23-node binary tree, in which each node is labeled with a unique letter of the alphabet. Preorder and inorder traversals of the tree visit the nodes in the following order:

- Preorder: \( Y \ G \ E \ P \ V \ U \ B \ N \ X \ I \ Z \ L \ O \ F \ J \ A \ H \ R \ C \ D \ S \ M \ T \)
- Inorder: \( P \ E \ U \ V \ B \ G \ X \ N \ I \ Y \ F \ O \ J \ L \ R \ H \ D \ C \ S \ A \ M \ Z \ T \)

(a) List the nodes in Professor Džunglová’s tree in post-order.

**Solution:** \( P \ U \ B \ V \ E \ X \ I \ N \ G \ F \ J \ O \ R \ D \ S \ C \ H \ M \ A \ L \ T \ Z \ Y \).

In his classic recreational linguistics book *Language on Vacation*, Dmitri Borgmann defined the word \( pubvexingfjordschmaltzy \) as “As if in the manner of the extreme sentimentalism generated in some individuals by the sight of a majestic fjord, which sentimentalism is annoying to the clientele of an English inn.”

Yes, there is such a thing as recreational linguistics.

**Rubric:** 5 points. —1 for each misplaced, missing, or repeated letter, but no negative scores. No proof is required.

(b) Draw Professor Džunglová’s tree.

**Solution:**

![Binary Tree Diagram]

**Rubric:** 5 points. —1 for each misplaced, missing, or repeated node, but no negative scores. No credit if the submission is not a binary tree. No proof is required.

1
2. The **complement** $w^c$ of a string $w \in \{0, 1\}^*$ is obtained from $w$ by replacing every 0 in $w$ with a 1 and vice versa; for example, $111011000100^c = 000100111011$. The complement function is formally defined as follows:

$$w^c :=
\begin{cases}
\varepsilon & \text{if } w = \varepsilon \\
1 \cdot x^c & \text{if } w = 0x \\
0 \cdot x^c & \text{if } w = 1x
\end{cases}$$

(a) Prove by induction that $|w| = |w^c|$ for every string $w$.

**Solution:** Let $w$ be an arbitrary string.

Assume for all strings $x$ where $|x| < |w|$ that $|x| = |x^c|$.

There are two cases to consider:

- If $w = \varepsilon$, then $w^c = \varepsilon$ by definition, so $w = w^c$, which trivially implies $|w| = |w^c|$.
- If $w = 0x$ for some string $x$, then

\[
|w^c| = |1x^c| \quad \text{by definition of } w^c \\
= 1 + |x^c| \quad \text{by definition of } || \\
= 1 + |x| \quad \text{by the inductive hypothesis} \\
= |0x| \quad \text{by definition of } || \\
= |w|.
\]

- Similarly, if $w = 1x$ for some string $x$, then

\[
|w^c| = |0x^c| \quad \text{by definition of } w^c \\
= 1 + |x^c| \quad \text{by definition of } || \\
= 1 + |x| \quad \text{by the inductive hypothesis} \\
= |1x| \quad \text{by definition of } || \\
= |w|.
\]

In all cases, we conclude that $|w| = |w^c|$.

**Rubric:** 5 points: standard induction rubric (scaled)
(b) Prove by induction that \((x \cdot y)^c = x^c \cdot y^c\) for all strings \(x\) and \(y\).

**Solution:** Let \(x\) and \(y\) be arbitrary strings. Assume for all strings \(z\) where \(|z| < |x|\) that \((z \cdot y)^c = z^c \cdot y^c\).

There are three cases to consider:

- If \(x = \varepsilon\), then

  \[
  (x \cdot y)^c = y^c \\
  = \varepsilon \cdot y^c \\
  = x^c \cdot y^c
  \]

  by definition of \(\cdot\),

  by definition of \(\cdot\),

  by definition of \(x^c\).

- If \(x = \theta z\) for some string \(z\), then

  \[
  (x \cdot y)^c = (\theta \cdot (z \cdot y))^c \\
  = 1 \cdot (z \cdot y)^c \\
  = 1 \cdot (z^c \cdot y^c) \\
  = (1 \cdot z^c) \cdot y^c \\
  = x^c \cdot y^c
  \]

  by definition of \(\cdot\),

  by definition of \(\cdot\),

  by the induction hypothesis,

  by definition of \(\cdot\),

  by definition of \(x^c\).

- Similarly, if \(x = 1z\) for some string \(z\), then

  \[
  (x \cdot y)^c = (1 \cdot (z \cdot y))^c \\
  = \theta \cdot (z \cdot y)^c \\
  = \theta \cdot (z^c \cdot y^c) \\
  = (\theta \cdot z^c) \cdot y^c \\
  = x^c \cdot y^c
  \]

  by definition of \(\cdot\),

  by definition of \(\cdot\),

  by the induction hypothesis,

  by definition of \(\cdot\),

  by definition of \(x^c\).

In all cases, we conclude that \((x \cdot y)^c = x^c \cdot y^c\). □

**Rubric:** 5 points: standard induction rubric (scaled)
3. Recursively define a set $L$ of strings over the alphabet $\{0, 1\}$ as follows:

- The empty string $\epsilon$ is in $L$.
- For all strings $x$ and $y$ in $L$, the string $0x1y$ is also in $L$.
- For all strings $x$ and $y$ in $L$, the string $1x0y$ is also in $L$.
- These are the only strings in $L$.

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

(a) Prove that the string $01000110111001$ is in $L$.

**Solution:**

- $\epsilon \in L$ by definition.
- $01 = 0 \cdot \epsilon \cdot 1 \cdot \epsilon \in L$ because $\epsilon \in L$ and $\epsilon \in L$.
- $1001 = 1 \cdot \epsilon \cdot 0 \cdot 01$ because $\epsilon \in L$ and $01 \in L$.
- $001101 = 0 \cdot 01 \cdot 1 \cdot 01 \in L$ because $01 \in L$ and $01 \in L$.
- $10001101 = 1 \cdot \epsilon \cdot 0 \cdot 001101 \in L$ because $\epsilon \in L$ and $001101 \in L$.
- $01000110111001 = 0 \cdot 10001101 \cdot 1 \cdot 1001 \in L$ because $1001 \in L$ and $10001101 \in L$.

**Solution (clever):** By part (c), it suffices to observe that $\#(0, 01000110111001) = \#(1, 01000110111001) = 7$.

**Rubric:** 2 points. The first two solutions describe the only two correct derivations of the string, but these are not the only valid proof structures. The first proof is more detailed than necessary for full credit, but any proof must separately justify each of the component substrings (10001101, 001101, 1001, and 01 in the first proof). The clever proof is worth full credit even without a solution to part (c).
(b) Prove by induction that every string in \( L \) has exactly the same number of 0s and 1s. (You may assume without proof that \( \#(a,x) = \#(a,x) + \#(a,y) \) for any symbol \( a \) and any strings \( x \) and \( y \).)

**Solution:** Let \( w \) be an arbitrary string in \( L \).

Assume, for any string \( x \in L \) where \( |x| < |w| \), that \( \#(\emptyset, x) + \#(1, x) \).

There are three cases to consider.

- If \( w = \epsilon \), then \( \#(\emptyset, w) = 0 = \#(1, \epsilon) \) by definition of \( \# \).
- Suppose \( w = \theta x 1 y \) for some strings \( x, y \in L \).

\[
\begin{align*}
\#(\emptyset, w) &= \#(\emptyset, x 1 y) & \text{by definition of \#} \\
&= 1 + \#(\emptyset, x) + \#(\emptyset, 1 y) & \text{\( (a, x y) = \#(a, x) + \#(a, y) \)} \\
&= 1 + \#(\emptyset, x) + \#(\emptyset, y) & \text{by definition of \#} \\
&= 1 + \#(1, x) + \#(1, y) & \text{by the inductive hypothesis} \\
&= \#(1, x) + \#(1, 1 y) & \text{by definition of \#} \\
&= \#(1, x 1 y) & \text{\( (a, x y) = \#(a, x) + \#(a, y) \)} \\
&= \#(1, \emptyset x 1 y) & \text{by definition of \#} \\
&= \#(1, w)
\end{align*}
\]

- Suppose \( w = 1 x \emptyset y \) for some strings \( x, y \in L \).

\[
\begin{align*}
\#(\emptyset, w) &= \#(\emptyset, x \emptyset y) & \text{by definition of \#} \\
&= \#(\emptyset, x) + \#(\emptyset, \emptyset y) & \text{\( (a, x y) = \#(a, x) + \#(a, y) \)} \\
&= \#(\emptyset, x) + 1 + \#(\emptyset, y) & \text{by definition of \#} \\
&= \#(1, x) + 1 + \#(1, y) & \text{by the inductive hypothesis} \\
&= \#(1, x) + 1 + \#(1, \emptyset y) & \text{by definition of \#} \\
&= 1 + \#(1, x \emptyset y) & \text{\( (a, x y) = \#(a, x) + \#(a, y) \)} \\
&= \#(1, 1 x \emptyset y) & \text{by definition of \#} \\
&= \#(1, w)
\end{align*}
\]

**Rubric:** 4 points: standard induction rubric (scaled)
(c) Prove by induction that \( L \) contains every string with the same number of \( \emptyset \)s and \( 1 \)s.

**Solution:** To simplify notation, let \( \Delta(w) = \#(1, w) - \#(\emptyset, w) \) for any string \( w \). Because \( \#(a, x \cdot y) = \#(a, x) + \#(a, y) \), we have \( \Delta(x \cdot y) = \Delta(x) + \Delta(y) \) for all strings \( x \) and \( y \). In particular, we have \( \Delta(x \cdot \emptyset) = \Delta(x) - 1 \) and \( \Delta(x \cdot 1) = \Delta(x) + 1 \). We need to prove that \( L \) contains every string \( w \) where \( \Delta(w) = 0 \).

**Proof:** Let \( w \) be an arbitrary string such that \( \Delta(w) = 0 \).

Assume \( L \) contains every string \( x \) such that \( |x| < |w| \) and \( \Delta(x) = 0 \). There are three cases to consider.

- If \( w = \varepsilon \), then \( w \in L \) by definition.
- Suppose \( w = \emptyset y \) for some string \( y \).

  Write \( w = p \cdot z \), where \( p \) is the shortest non-empty prefix of \( w \) such that \( \Delta(p) \geq 0 \). We know that such a prefix exists, because \( w \) is a non-empty prefix of \( w \) with \( \Delta(w) \geq 0 \). (The suffix \( z \) might be empty.)

  Because \( p \) is non-empty, we can write \( p = qa \) for some string \( q \) and some symbol \( a \in \{\emptyset, 1\} \). The definition of \( p \) implies that \( \Delta(q) < 0 \) and therefore \( \Delta(q) \leq -1 \).

  If \( a = \emptyset \), then \( \Delta(p) = \Delta(q\emptyset) = \Delta(q) - 1 < 0 \), which is impossible because \( \Delta(p) \geq 0 \). So we must have \( a = 1 \) and therefore \( \Delta(p) = \Delta(q) + 1 \). It follows that \( \Delta(p) = 0 \) and \( \Delta(q) = -1 \).

  Because \( p \) starts with \( \emptyset \) and ends with \( 1 \), we must have \( p = \emptyset x 1 \) for some string \( x \) (which might be empty). It follows that \( \Delta(x) = \Delta(p) = 0 \), and therefore \( x \in L \) by the inductive hypothesis.

  We also have \( \Delta(w) = \Delta(p) + \Delta(z) = \Delta(z) \), and therefore \( \Delta(z) = 0 \). So the inductive hypothesis implies \( z \in L \).

  We conclude that \( w = \emptyset x 1 z \), where \( x \in L \) and \( z \in L \), and therefore \( w \in L \).

- A symmetric argument implies that if \( w = 1 y \) for some string \( y \), then \( w \in L \).

  Write \( w = p \cdot z \), where \( p \) is the shortest non-empty prefix of \( w \) such that \( \Delta(p) \leq 0 \).

  Because \( p \) is non-empty, \( p = qa \) for some string \( q \) and some symbol \( a \). The definition of \( p \) implies that \( \Delta(q) \geq 1 \).

  If \( a = 1 \), then \( \Delta(p) = \Delta(q) + 1 > 0 \), which is impossible. So \( a = \emptyset \) and \( \Delta(p) = \Delta(q) - 1 \), which implies that \( \Delta(p) = 0 \) and \( \Delta(q) = 1 \).

  Thus, \( p = 1 x \emptyset \) for some string \( x \). We have \( \Delta(x) = \Delta(p) = 0 \), and therefore \( x \in L \) by the inductive hypothesis.

  We also have \( \Delta(z) = \Delta(w) - \Delta(p) = 0 \), and therefore \( z \in L \) by the inductive hypothesis.

  We conclude that \( w = 1 x \emptyset z \), where \( x \in L \) and \( z \in L \), and thus \( w \in L \).

In all three cases, we conclude that \( w \in L \). ■

Rubric: 4 points = 1 for induction boilerplate (including strong induction hypothesis and exhaustive case analysis) + \( \frac{1}{2} \) for base case + 2 for first inductive case + \( \frac{1}{2} \) for second inductive case. “Essentially the same argument implies \( w \in L \) when \( w = 1 y \)” is enough for the last \( \frac{1}{2} \) point.
Rubric (induction): For problems worth 10 points:

+ 1 for explicitly considering an arbitrary object
+ 2 for a valid strong induction hypothesis
  – **Deadly Sin!** Automatic zero for stating a weak induction hypothesis, unless the rest of the proof is 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 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 cases overlap

+ 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.