1. The infamous Scottish computational arborist Seòras na Coille has a favorite 26-node binary tree, whose nodes are labeled with the letters of the English alphabet. Preorder and inorder traversals of his tree yield the following letter sequences:

Preorder: U X M Z I W J E O V N H R D T K G L Y A F S Q P C B
Inorder: Z M X E J V O W I N D R T H U G K F A Q P S Y C B L

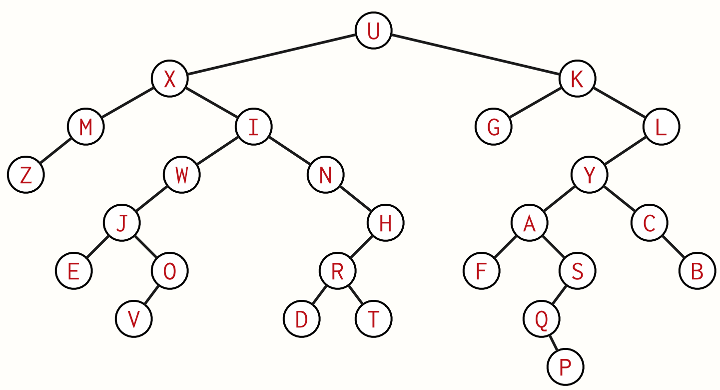
(a) List the nodes in Professor na Coille's tree according to a postorder traversal.

Solution: Z M E V O J W D T R H N I X G F P Q S A B C Y L K U ■

Rubric: 5 points. −1 for each misplaced, missing, or repeated letter, but no negative scores. No proof is required. The correct solution is unique.

(b) Draw Professor na Coille's tree.

Solution:



■

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. In particular, the distinction between left and right children matters, even for nodes with only one child. No proof is required. The correct solution is unique.

[Hint: It may be easier to write a short Python program than to figure this out by hand.]

Solution: The following code assumes that the input consists of two strings, containing the preorder and inorder traversals of the same binary tree, whose vertex labels are distinct characters. Invalid inputs will make the universe explode.

```
def GetPost(preorder, inorder):
    if len(preorder) <= 1:
        return preorder
    else:
        root = preorder[0]
        [inL, inR] = inorder.split(root)
        preL = preorder[1:len(inL)+1]
        preR = preorder[len(inL)+1:]
        return GetPost(preL, inL) + root + GetPost(preR, inR)

GetPost("UXMZIWJEOVNHRDTKGLYAFSQPCB", "ZMXEJVOWINDRTHUGKFAQPSYCBL")
```

■

Rubric: Not graded.

2. For any string $w \in \{0,1\}^*$, let $sort(w)$ denote the string obtained by sorting the characters in w. For example, $sort(010101) = 000111$. This function can be defined recursively as follows:

$$sort(w) := \begin{cases} \varepsilon & \text{if } w = \varepsilon \\ 0 \cdot sort(x) & \text{if } w = 0x \\ sort(x) \bullet 1 & \text{if } w = 1x \end{cases}$$

(a) Prove that $\#(0, sort(w)) = \#(0, w)$ for every string $w \in \{0,1\}^*$

Solution: Let w be an arbitrary binary string.
Assume $\#(0, sort(x)) = \#(0, x)$ for every string x that is shorter than w.
There are three cases to consider.

- If $w = \varepsilon$, then

$$\begin{aligned} \#(0, sort(w)) &= \#(0, sort(\varepsilon)) && \text{because } w = \varepsilon \\ &= \#(0, \varepsilon) && \text{by definition of } sort \\ &= \#(0, w) && \text{because } w = \varepsilon \end{aligned}$$

- If $w = 0x$ for some string x, then

$$\begin{aligned} \#(0, sort(w)) &= \#(0, sort(0x)) && \text{because } w = 0x \\ &= \#(0, 0 \cdot sort(x)) && \text{by definition of } sort \\ &= 1 + \#(0, sort(x)) && \text{by definition of } \# \\ &= 1 + \#(0, x) && \text{by the induction hypothesis} \\ &= \#(0, 0x) && \text{by definition of } \# \\ &= \#(0, w) && \text{because } w = 0x \end{aligned}$$

- If $w = 1x$ for some string x, then

$$\begin{aligned} \#(0, sort(w)) &= \#(0, sort(1x)) && \text{because } w = 1x \\ &= \#(0, sort(x) \bullet 1) && \text{by definition of } sort \\ &= \#(0, sort(x)) + \#(0, 1) && \#(a, u \bullet v) = \#(a, u) + \#(a, v) \\ &= \#(0, sort(x)) && \text{by definition of } \# \\ &= \#(0, x) && \text{by the induction hypothesis} \\ &= \#(0, 1x) && \text{by definition of } \# \\ &= \#(0, w) && \text{because } w = 1x \end{aligned}$$

In all cases, we have $\#(0, sort(w)) = \#(0, w)$. ■

Rubric: 3 points: Standard induction rubric (scaled). This is more detail than necessary for full credit.

(b) Prove that $sort(w \bullet 1) = sort(w) \bullet 1$ for every string $w \in \{0, 1\}^*$

Solution: Let w be an arbitrary string.
Assume that $sort(x \bullet 1) = sort(x) \bullet 1$ for every string x that is shorter than w.
There are three cases to consider.

- If $w = \varepsilon$, then

$$\begin{aligned} sort(x \bullet 1) &= sort(\varepsilon \bullet 1) && \text{because } w = \varepsilon \\ &= sort(1) && \text{by definition of } \bullet \\ &= sort(\varepsilon) \bullet 1 && \text{by definition of } sort \\ &= sort(w) \bullet 1 && \text{because } w = \varepsilon \end{aligned}$$

- If $w = 0x$, then

$$\begin{aligned} sort(x \bullet 1) &= sort((0x) \bullet 1) && \text{because } w = 0x \\ &= sort(0(x \bullet 1)) && \text{by definition of } \bullet \\ &= 0 \cdot sort(x \bullet 1) && \text{by definition of } sort \\ &= 0 \cdot ((sort(x) \bullet 1) && \text{by the induction hypothesis} \\ &= (0 \cdot (sort(x)) \bullet 1 && \text{by definition of } \bullet \\ &= sort(0x) \bullet 1 && \text{by definition of } sort \\ &= sort(w) \bullet 1 && \text{because } w = 0x \end{aligned}$$

- if $w = 1x$, then

$$\begin{aligned} sort(x \bullet 1) &= sort((1x) \bullet 1) && \text{because } w = 1x \\ &= sort(1(x \bullet 1)) && \text{by definition of } \bullet \\ &= sort(x \bullet 1) \bullet 1 && \text{by definition of } sort \\ &= (sort(x) \bullet 1) \bullet 1 && \text{by the induction hypothesis} \\ &= sort(1x) \bullet 1 && \text{by definition of } sort \\ &= sort(w) \bullet 1 && \text{because } w = 1x \end{aligned}$$

In all cases, we have $sort(w \bullet 1) = sort(w) \bullet 1$. ■

Rubric: 3 points: Standard induction rubric (scaled). This is more detail than necessary for full credit.

(c) Prove that $sort(sort(w)) = sort(w)$ for every string $w \in \{\mathtt{0}, \mathtt{1}\}^*$.

Solution: Let w be an arbitrary string.
Assume $sort(sort(x)) = sort(x)$ for every string x that is shorter than w.
There are three cases to consider:

- If $w = \varepsilon$, then

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

- If $w = \mathtt{0}x$ for some string x, then

$$\begin{aligned} sort(sort(w)) &= sort(sort(\mathtt{0}x)) && \text{because } w = \mathtt{0}x \\ &= sort(\mathtt{0} \cdot sort(x)) && \text{by definition of } sort \\ &= \mathtt{0} \cdot sort(sort(x)) && \text{by definition of } sort \\ &= \mathtt{0} \cdot sort(x) && \text{by the induction hypothesis} \\ &= sort(\mathtt{0}x) && \text{by definition of } sort \\ &= sort(w) && \text{because } w = \mathtt{0}x \end{aligned}$$

- If $w = \mathtt{1}x$ for some string x, then

$$\begin{aligned} sort(sort(w)) &= sort(sort(\mathtt{1}x)) && \text{because } w = \mathtt{1}x \\ &= sort(sort(x) \bullet \mathtt{1}) && \text{by definition of } sort \\ &= sort(sort(x)) \bullet \mathtt{1} && \textbf{by part (b)} \\ &= sort(x) \bullet \mathtt{1} && \text{by the induction hypothesis} \\ &= sort(\mathtt{1}x) && \text{by definition of } sort \\ &= sort(w) && \text{because } w = \mathtt{1}x \end{aligned}$$

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

Rubric: 4 points: Standard induction rubric (scaled). This is more detail than necessary for full credit; however, any correct proof must explicitly invoke part (b) in the third case.

(d) **Not for submission:** Prove that $x \bullet \mathtt{0} \neq y \bullet \mathtt{1}$ for all strings $x, y \in \{\mathtt{0}, \mathtt{1}\}^*$.

Solution (induction): Let x and y be arbitrary binary strings.
Assume $x' \bullet \mathtt{0} \neq y' \bullet \mathtt{1}$ for all strings x' and y' such that x' is shorter than x.
There are several cases to consider.

- If $x = y = \varepsilon$, then $x \bullet \mathtt{0} = \varepsilon \bullet \mathtt{0} = \mathtt{0} \neq \mathtt{1} = \varepsilon \bullet \mathtt{1} = y \bullet \mathtt{1}$, by definition of $\bullet$ (twice).
- If $x = \varepsilon$ and $y = \mathtt{0}y'$ for some string y', then

$$\begin{aligned}
x \bullet \mathtt{0} &= \varepsilon \bullet \mathtt{0} && \text{because } x = \varepsilon \\
&= \mathtt{0} && \text{by definition of } \bullet \\
&\neq \mathtt{0} \cdot (y' \bullet \mathtt{1}) && \text{because } (y' \bullet \mathtt{1}) \neq \varepsilon \\
&= (\mathtt{0}y') \bullet \mathtt{1} && \text{by definition of } \bullet \\
&= y \bullet \mathtt{1} && \text{because } y = \mathtt{0}y'
\end{aligned}$$

- If $x = \varepsilon$ and $y = \mathtt{1}y'$ for some string y', then

$$\begin{aligned}
x \bullet \mathtt{0} = \mathtt{0} && \text{by definition of } \bullet \\
\neq \mathtt{1} \cdot (y' \bullet \mathtt{1}) && \text{because } \mathtt{0} \neq \mathtt{1} \\
= (\mathtt{1}y') \bullet \mathtt{1} && \text{by definition of } \bullet \\
= y \bullet \mathtt{1} && \text{because } y = \mathtt{0}y'
\end{aligned}$$

- Symmetric arguments apply if $x \neq \varepsilon$ and $y = \varepsilon$.
- If $x = ax'$ and $y = ay'$ for some symbol a and strings x' and y', then

$$\begin{aligned}
x \bullet \mathtt{0} &= (ax') \bullet \mathtt{0} && \text{because } x = a \cdot x' \\
&= a \cdot (x' \bullet \mathtt{0}) && \text{by definition of } \bullet \\
&\neq a \cdot (y' \bullet \mathtt{1}) && \text{by definition of } = \text{ and } \textbf{IH} \\
&= (ay') \bullet \mathtt{1} && \text{because } y = a \cdot y' \\
&= y \bullet \mathtt{1} && \text{because } y = ay'
\end{aligned}$$

- Finally, if $x = ax'$ and $y = by'$ for some symbols $a \neq b$ and strings x' and y', then

$$\begin{aligned}
x \bullet \mathtt{0} &= (ax') \bullet \mathtt{0} && \text{because } x = a \cdot x' \\
&= a \cdot (x' \bullet \mathtt{0}) && \text{by definition of } \bullet \\
&\neq b \cdot (y' \bullet \mathtt{1}) && \text{by definition of } = \text{ and } a \neq b \\
&= (by') \bullet \mathtt{1} && \text{because } y = a \cdot y' \\
&= y \bullet \mathtt{1} && \text{because } y = by'
\end{aligned}$$

In all cases, we conclude $x \bullet \mathtt{0} \neq y \bullet \mathtt{1}$. ■

Rubric: Standard induction rubric. Yes, that is enough detail in the fourth case.

Solution (via reversal): We use properties of the reversal function w^R that we proved in Lab 1 on Wednesday.

Let x and y be arbitrary strings. We have

$$\begin{aligned}
(x \bullet \mathtt{0})^R &= \mathtt{0}^R \bullet x^R && \text{because } (u \bullet v)^R = v^R \bullet u^R \\
&= \mathtt{0} \bullet x^R && \text{by definition} \\
&= \mathtt{0} \cdot x^R && \text{by definition of } \bullet
\end{aligned}$$

and by similar reasoning, $(y \bullet \mathtt{1})^R = \mathtt{1} \cdot y^R$.

We immediately have $\mathtt{0} \cdot x^R \neq \mathtt{1} \cdot y^R$, because $\mathtt{0} \neq \mathtt{1}$.

It follows that $(x \bullet \mathtt{0})^R \neq (y \bullet \mathtt{1})^R$, and therefore $x \bullet \mathtt{0} \neq y \bullet \mathtt{1}$. ■

Rubric: Yes, this would be worth full credit. Labs are fair game for prior results.

(e) **Not for submission:** Prove that $sort(w) \neq x \bullet \mathtt{10} \bullet y$, for all strings $w, x, y \in \{\mathtt{0}, \mathtt{1}\}^*$.

Solution: Let w, x, and y be arbitrary strings.

Assume $sort(w') \neq x' \bullet \mathtt{10} \bullet y'$, for all strings w', x', and y' such that w' is shorter than w.

There are three cases to consider.

- Suppose $w = \varepsilon$.
  Then $sort(w) = \varepsilon$ by definition.
  There are two subcases to consider, depending on whether x is empty.
  - If $x = \varepsilon$, then

$$\begin{aligned}
x \bullet \mathtt{10} \bullet y &= \varepsilon \bullet \mathtt{10} \bullet y && \text{because } x = \varepsilon \\
&= \mathtt{10} \bullet y && \text{by definition of } \bullet \\
&= \mathtt{1} \cdot (\mathtt{0} \cdot y) && \text{by definition of } \bullet \\
&\neq \varepsilon && \text{by definition of } =
\end{aligned}$$

  - If $x = a \cdot x'$ for some $a \in \{\mathtt{0}, \mathtt{1}\}$ and some string x', then

$$\begin{aligned}
x \bullet \mathtt{10} \bullet y &= (a \cdot x') \bullet \mathtt{10} \bullet y && \text{because } x = \varepsilon \\
&= a \cdot (x' \bullet \mathtt{10} \bullet y) && \text{by definition of } \bullet \\
&\neq \varepsilon && \text{by definition of } =
\end{aligned}$$

- Suppose $w = \mathtt{0}w'$ for some string w'.
  Then $sort(w) = \mathtt{0} \cdot sort(w')$ by definition.
  There are three subcases to consider, depending on the first symbol of x.
  - If $x = \varepsilon$, then

$$\begin{aligned}
x \bullet \mathtt{10} \bullet y &= \mathtt{1} \cdot (\mathtt{0} \cdot y) && \text{by previous analysis} \\
&\neq \mathtt{0} \cdot sort(w') && \text{by definition of } = \text{ and } \mathtt{0} \neq \mathtt{1}
\end{aligned}$$

– If $x = \mathtt{0} \cdot x'$ for some string x', then

$$\begin{aligned} x \bullet \mathtt{10} \bullet y &= \mathtt{0} \cdot (x' \bullet \mathtt{10} \bullet y) && \text{by previous analysis} \\ &\neq \mathtt{0} \cdot sort(w') && \text{by the induction hypothesis} \end{aligned}$$

– If $x = \mathtt{1} \cdot x'$ for some string x', then

$$\begin{aligned} x \bullet \mathtt{10} \bullet y &= \mathtt{1} \cdot (x' \bullet \mathtt{10} \bullet y) && \text{by previous analysis} \\ &\neq \mathtt{0} \cdot sort(w') && \text{by definition of } = \text{ and } \mathtt{0} \neq \mathtt{1} \end{aligned}$$

- Suppose $w = \mathtt{1}w'$ for some string w'.
  Then $sort(w) = sort(w') \bullet \mathtt{1}$ by definition.
  There are three subcases to consider, depending on the *last* symbol of y.

– If $y = \varepsilon$, then

$$\begin{aligned} x \bullet \mathtt{10} \bullet y &= x \bullet \mathtt{10} && \text{because } u \bullet \varepsilon = u \\ &= (x \bullet \mathtt{1}) \bullet \mathtt{0} && \text{because } \bullet \text{ is associative} \\ &\neq sort(w') \bullet \mathtt{1} && \text{by part (d)} \end{aligned}$$

– If $y = y' \bullet \mathtt{0}$ for some string y', then

$$\begin{aligned} x \bullet \mathtt{10} \bullet y &= x \bullet \mathtt{10} \bullet (y' \bullet \mathtt{0}) && \text{because } y = y' \bullet \mathtt{0} \\ &= (x \bullet \mathtt{10} \bullet y') \bullet \mathtt{0} && \text{because } \bullet \text{ is associative} \\ &\neq sort(w') \bullet \mathtt{1} && \text{by part (d)} \end{aligned}$$

– If $y = y' \bullet \mathtt{1}$ for some string y', then

$$\begin{aligned} x \bullet \mathtt{10} \bullet y &= x \bullet \mathtt{10} \bullet (y' \bullet \mathtt{0}) && \text{because } y = y' \bullet \mathtt{0} \\ &= (x \bullet \mathtt{10} \bullet y') \bullet \mathtt{1} && \text{because } \bullet \text{ is associative} \\ &\neq sort(w') \bullet \mathtt{1} && \text{by the induction hypothesis} \end{aligned}$$

In all eight(!) cases, we conclude that $sort(w) \neq x \bullet \mathtt{10} \bullet y$. ■

Rubric: Standard induction rubric. The only thing *difficult* about this proof is having the patience to enumerate all the cases.

3. Consider the set of strings $L \subseteq \{\texttt{0}, \texttt{1}\}^*$ defined recursively as follows:

- The empty string ε is in L.
- For any strings x in L, the strings $\texttt{0}x\texttt{1}$ and $\texttt{1}x\texttt{0}$ are also in L.
- For any two *non-empty* strings x and y in L, the string $x \bullet y$ is also in L.
- These are the only strings in L.

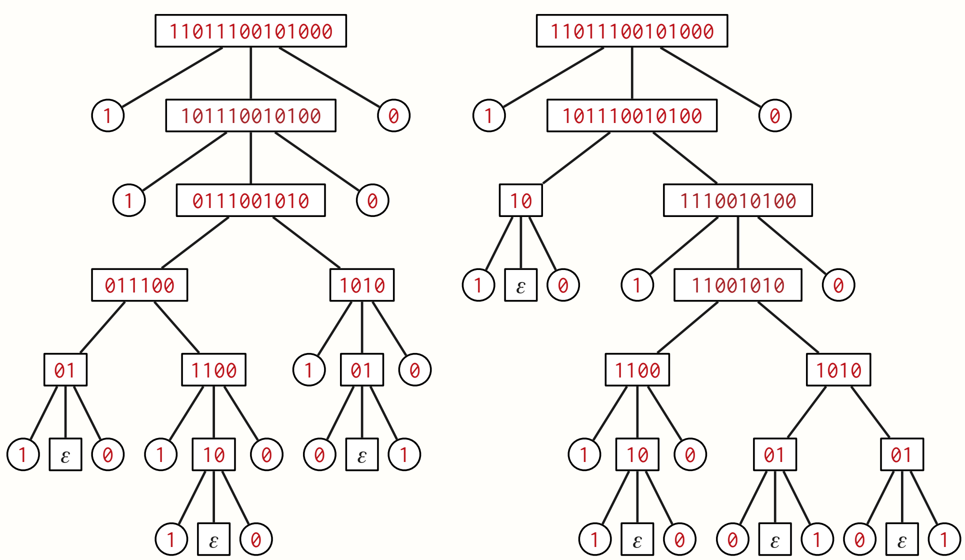
(a) Prove that the string $\texttt{11011100101000}$ is in L.

Solution (derivation): We derive a series of strings in L as follows:

$$
\begin{aligned}
s &= \texttt{0}\varepsilon\texttt{1} = \texttt{0 1}\\
t &= \texttt{1}\varepsilon\texttt{0} = \texttt{1 0}\\
u &= s \bullet s = \texttt{01 01}\\
v &= t \bullet u = \texttt{10 0101}\\
w &= \texttt{1}v\texttt{0} = \texttt{1 100101 0}\\
x &= s \bullet w = \texttt{01 11001010}\\
y &= \texttt{1}x\texttt{0} = \texttt{1 0111001010 0}\\
z &= \texttt{1}y\texttt{0} = \texttt{1 101110010100 0}
\end{aligned}
$$

■

Solution (parse tree): Either of the following parse trees provides a proof. Each string in a rectangle is in L, either because it's empty, or because it can be broken into smaller strings (the children) as described in the definition of L. Circles represent individual $\texttt{0}$s and $\texttt{1}$s.



■

Solution (clever): The string $w = \texttt{11011100101000}$ has exactly seven $\texttt{0}$s and seven $\texttt{1}$s, so $\Delta(w) = 0$, and so part (c) implies that $w \in L$. ■

Rubric: 2 points. These are not the only derivations of this string, and these are not the only valid proof structures. The first proof is more detailed than necessary for full credit, but any similar proof must separately justify each of the component substrings (`01`, `10`, `0101`, `100101`, and so on). You don't need both parse trees, obviously. The clever proof is worth full credit even without a solution to part (c), but it must explicitly invoke part (c).

(b) Prove that $\Delta(w) = 0$ for every string $w \in L$.

Solution: Let w be an arbitrary string in L.
Assume that $\Delta(x) = 0$ for every string $x \in L$ that is shorter than w.
There are four cases to consider.

- If $w = \varepsilon$, then $\Delta(w) = 0$ by definition of Δ.
- If $w = \texttt{0}x\texttt{1}$ for some string $x \in L$, then

$$\begin{aligned}
\Delta(w) &= \Delta(\texttt{0}x\texttt{1}) && \text{because } w = \texttt{0}x\texttt{1} \\
&= \Delta(\texttt{0}) + \Delta(x) + \Delta(\texttt{1}) && \text{addition property} \\
&= \Delta(x) && \text{by definition of } \Delta \\
&= 0 && \text{by the inductive hypothesis}
\end{aligned}$$

- If $w = \texttt{1}x\texttt{0}$ for some string $x \in L$, then

$$\begin{aligned}
\Delta(w) &= \Delta(\texttt{1}x\texttt{0}) && \text{because } w = \texttt{1}x\texttt{0} \\
&= \Delta(\texttt{1}) + \Delta(x) + \Delta(\texttt{0}) && \text{addition property} \\
&= \Delta(x) && \text{by definition of } \Delta \\
&= 0 && \text{by the inductive hypothesis}
\end{aligned}$$

- If $w = x \bullet y$ for some nonempty strings $x, y \in L$, then

$$\begin{aligned}
\Delta(w) &= \Delta(x \bullet y) && \text{because } w = x \bullet y \\
&= \Delta(x) + \Delta(y) && \text{addition property} \\
&= 0 && \text{by the inductive hypothesis (twice)}
\end{aligned}$$

In all cases, we conclude that $\Delta(w) = 0$. ■

Rubric: 4 points: Standard induction rubric (scaled). This is more detail than necessary for full credit.

(c) Prove that L contains every string $w \in \{0,1\}^*$ such that $\Delta(w) = 0$.

Solution: Let w be an arbitrary string such that $\Delta(w) = 0$.
Assume, for every string x that is shorter than w, that if $\Delta(x) = 0$ then $x \in L$.
There are three cases to consider.

- If $w = \varepsilon$, then $w \in L$ by definition.
- If $w = 0$ or $w = 1$, then $\Delta(w) \neq 0$, which is impossible.
- Suppose $w = 0x1$ or $w = 1x0$ for some string x.
  Addition implies $\Delta(x) = 0$.
  So the inductive hypothesis implies $x \in L$.
  So the definition of L implies $w \in L$.
- Suppose $w = 0x0$ for some string x.
  Addition implies $\Delta(x0) = +1$ and $\Delta(0) = -1$.
  Interpolation 1 implies there are strings y and z such that $x = yz$ and $\Delta(z0) = 0$.
  The identity $x = yz$ implies that $w = 0y \bullet z0$.
  Addition and $\Delta(z0) = 0$ imply $\Delta(0y) = 0$.
  So the inductive hypothesis implies $0y \in L$ and $z0 \in L$.
  So the definition of L implies that $w \in L$.
- The final case $w = 1x1$ follows from a similar argument (via Interpolation 2).

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

Rubric: 4 points: Standard induction rubric (scaled). This is more detail than necessary for full credit.

(d) **Not for submission:** Prove the addition property: $\Delta(xy) = \Delta(x) + \Delta(y)$ for all strings x and y.

Solution (induction): Let x and y be arbitrary strings.
Assume for all strings x' shorter than x that $\Delta(x'y) = \Delta(x') + \Delta(y)$.
There are three cases to consider.

- If $x = \varepsilon$, then

$$\begin{aligned}
\Delta(xy) &= \Delta(\varepsilon \bullet y) && \text{because } x = \varepsilon \\
&= \Delta(y) && \text{by definition of } \bullet \\
&= \Delta(\varepsilon) + \Delta(y) && \text{by definition of } \Delta \\
&= \Delta(x) + \Delta(y) && \text{because } x = \varepsilon
\end{aligned}$$

- If $x = \mathtt{0}x'$ for some string x', then

$$\begin{aligned}
\Delta(xy) &= \Delta((\mathtt{0}x') \bullet y) && \text{because } x = \mathtt{0}x' \\
&= \Delta(\mathtt{0}(x' \bullet y)) && \text{by definition of } \bullet \\
&= \Delta(x' \bullet y) - 1 && \text{by definition of } \Delta \\
&= \Delta(x') + \Delta(y) - 1 && \text{by the induction hypothesis} \\
&= \Delta(\mathtt{0}x') + \Delta(y) && \text{by definition of } \Delta \\
&= \Delta(x) + \Delta(y) && \text{because } x = \mathtt{0}x'
\end{aligned}$$

- The last case $x = \mathtt{1}x'$ is similar to the previous case.

In all cases, we conclude that $\Delta(xy) = \Delta(x) + \Delta(y)$. ■

Rubric: Standard induction rubric.

Solution (definition-chasing):

$$\begin{aligned}
\Delta(xy) &= \#(\mathtt{1}, xy) - \#(\mathtt{0}, xy) && \text{by definition of } \Delta \\
&= \big(\#(\mathtt{1}, x) + \#(\mathtt{1}, y)\big) - \big(\#(\mathtt{0}, x) + \#(\mathtt{0}, y)\big) && (*) \\
&= \big(\#(\mathtt{1}, x) - \#(\mathtt{0}, x)\big) + \big(\#(\mathtt{1}, y) - \#(\mathtt{0}, y)\big) && \text{arithmetic} \\
&= \Delta(x) + \Delta(y) && \text{by definition of } \Delta
\end{aligned}$$

The second step $(*)$ uses the identity $\#(a, uv) = \#(a, u) + \#(a, v)$, which is proved in the solutions to Lab 1. ■

Rubric: Yes, this would be worth full credit. Lab solutions are fair game for prior results.

(e) **Not for submission:** Prove the downward interpolation property: For all strings w and z such that $\Delta(wz) > 0$ and $\Delta(z) < 0$, there are strings x and y such that $w = xy$ and $\Delta(yz) = 0$.

Solution: Let w and z be arbitrary strings such that $\Delta(wz) > 0$ and $\Delta(z) < 0$. Assume for all strings w' that are shorter than w, such that $\Delta(w'z) > 0$, there are strings x' and y' such that $w' = x'y'$ and $\Delta(y'z) = 0$.

There are three cases to consider.

- If $w = \varepsilon$, then $\Delta(wz) = \Delta(z)$, which is impossible.
- Suppose $w = \mathtt{0}w'$ for some string w'.
  Addition implies $\Delta(w'z) = \Delta(wz) + 1 > 0$.
  Thus, the inductive hypothesis implies there are strings x' and y' such that $w' = x'y'$ and $\Delta(y'z) = 0$.
  Let $x = \mathtt{0}x'$ and $y = y'$.
- Finally, suppose $w = \mathtt{1}w'$ for some string w'.
  Part (a) implies $\Delta(w'z) = \Delta(wz) - 1 \geq 0$.
  There are two subcases to consider.
  - If $\Delta(w'z) = 0$, let $x = \mathtt{1}$ and $y = w'$.
  - If $\Delta(w'z) > 0$, the inductive hypothesis implies there are strings x' and y' such that $w' = x'y'$ and $\Delta(y'z) = 0$.
    Let $x = \mathtt{1}x'$ and $y = y'$.

In all (possible) cases, we have found strings x and y such that $w = xy$ and $\Delta(yz) = 0$. ■

Rubric: Standard induction rubric.

(f) **Not for submission:** Prove the upward interpolation property: For all strings w and z such that $\Delta(wz) < 0$ and $\Delta(z) > 0$, there are strings x and y such that $w = xy$ and $\Delta(yz) = 0$.

Solution: Swap $\mathtt{0} \leftrightarrow \mathtt{1}$ and $+ \leftrightarrow -$ and $< \leftrightarrow >$and $\leq \leftrightarrow \geq$ in the previous proof. ■

Rubric: This would be worth exactly the same partial credit that you got for part (e).