CSE 311 - HW 6

Eric Boris,
Samek Mulepati, Wendy Jiang, Yolin Tsai, William Nguyen, Brittan Robinett, Sam Vanderlinda, Alexander Ayres

November 2019

1 Leaps and Bounds

1. Let P(n) be the claim that $T(n) \leq 10n$ for $T(n) = T(\lfloor \frac{n}{2} \rfloor) + 9\lfloor \frac{n}{3} \rfloor + 4$. Prove P(n) for $n \in \mathbb{N}$ and $n > 1$ by strong induction.

2. Base Case: P(1) holds because $T(1) = 8 \leq 10 = 10(1)$. P(2) holds because $T(2) = T(\lfloor \frac{2}{2} \rfloor) + 9\lfloor \frac{2}{3} \rfloor + 4 = 12 \leq 20 = 10(2)$.

3. Induction Hypothesis: Suppose that for an arbitrary natural number $k > 2$, P(j) is true for every integer j between 1 and k.

4. Inductive Step: Goal: Show P(k + 1); i.e. $T(k+1) = T(\lfloor \frac{k+1}{2} \rfloor) + 9\lfloor \frac{n+1}{3} \rfloor + 4 \leq 10(k+1)$

$$
\begin{aligned}
T(k+1) &= T(\lfloor \frac{k+1}{2} \rfloor) + 9\lfloor \frac{k+1}{3} \rfloor + 4 \\
&\leq 10(\lfloor \frac{k+1}{2} \rfloor) + 9\lfloor \frac{k+1}{3} \rfloor + 4 && \text{By IH, } \lfloor \tfrac{k+1}{2} \rfloor \leq k \text{ and } \lfloor \tfrac{k+1}{2} \rfloor \in \mathbb{Z} \\
&\leq 10(\frac{k+1}{2}) + 9(\frac{k+1}{3}) + 4 && \text{By def of } \lfloor \cdot \rfloor \\
&= 5(k+1) + 3(k+1) + 4 \\
&= 8k + 12 \\
&\leq 10(k+1)
\end{aligned}
$$

   We make this last claim, that $8k + 12 \leq 10(k + 1)$ on the basis that the difference between the two sides is always greater than or equal to 0 for a k greater than 1. In other words, $10(k + 1)$ is always greater than $8k + 12$ by a positive factor $2(k - 1)$. Namely,

$$
\begin{aligned}
8k + 12 &\leq 10(k+1) \\
8k + 12 &\leq 10k + 10 \\
0 &\leq 2k - 2 \\
0 &\leq 2(k-1)
\end{aligned}
$$

5. Thus, P(n) holds because we can show by strong induction that for an arbitrary natural number k+1, $T(k + 1) \leq 10(k + 1)$. $\square$

2 And Then Sum

1. Let P(n) be the claim that any integer $n \geq 18$ can be recursively represented as the sum of multiples of 4 and 7, i.e. $4x + 7y = n$. We prove P(n) by strong induction.

2. Base Case: If $n = 18$, we can make that with two 7's and one 4, i.e. $4(1)+7(2) = 18$. Similarly, if $n = 19$, that can be made with one 7 and three 4's, $4(3)+7(1) = 19$. If $n = 20 = 4(5)+7(0)$ and if $n = 21 = 4(0) + 7(3)$. Therefore, the base case P(n) holds for $18 \leq n \leq 21$.

3. Inductive Hypothesis: Assume that P(j) holds for arbitrary integers j and k such that $18 \leq j \leq k$ and $k \geq 21$.

4. Inductive Step: Goal, show that P(k+1) such that $4x + 7y = k + 1$ holds.
Because we've already proved base cases up through 21, we can assume that $k + 1 \geq 22$.

$$\begin{aligned} k+1 &\geq 22 \\ (k+1)-4 &\geq 18 \\ (k+1)-4 &= 4x+7y && \text{Inductive Hypothesis} \\ k+1 &= 4(x+1)+7y \end{aligned}$$

5. Conclusion: Because k was arbitrary, we've shown that we can represent $k + 1 \geq 18$ as the sum of multiples of 4 and 7. Therefore, P(n) is true for all $n \in \mathbb{N}$ by strong induction. $\square$

3 Some Strings Attached

3(a)

Basis Step: $x \in \{00, 01, 10, 11\}$
Recursive Step: If $x \in S$ then $x000, x001, x010, x011, x100, x101, x110, x111 \in S$
Exclusion Rule: Each element of S is obtained from the basis and a finite number of applications of the recursive step
Justification: We show that $x \in S$ iff the length $|x|$ satisfies $|x| \equiv 2(\text{mod } 3)$.

Suppose that $x \in S$ then by the base case, it is one of the three possible values of binary string that has length 2 mod 3, which is clearly congruent to 2 mod 3. Otherwise, x is built up from a base case string but repeated applications of the recursive step, each application of which appends a string of length 3 to x. These applications follow the pattern of $5 \equiv 8 \equiv 11 \equiv x+3n \equiv 2(\text{mod } 3)$. Because the basis step holds and each repeated application holds we can see that the definition holds for 2(mod 3).

Now, suppose that $|x|$ is congruent to 2(mod 3). If it's length is 2 then by definition it is congruent to 2 regardless of it's modulo value. Otherwise, it has length $|x + 3n|$ for some $n > 0$ which has a length congruent to 2(mod 3) as well. Hence, we can see that x can be build from a string that begins as being congruent to 2(mod 3) and continues to remain so following repeated applications of the recursive step which shows that $x \in S$.

3(b)

Basis Step: $\epsilon \in S$
Recursive Step: If $x \in S$ then $x0, x11 \in S$
Exclusion Rule: Each element of S is obtained from the basis and a finite number of applications of the recursive step

Justification: We show that $x \in S$ iff x does not contain a substring of the form "11...1" with an odd length.

Suppose that $x \in S$. If x is the empty string, then x does not contain the substring. Otherwise, x is built from zeroes or pairs of ones. In the case of appending pairs of ones, 2n is always even so there will never be a substring of ones of even length.

Now, suppose that x does not contain a substring of the form "11..1" that is an odd length. If it's length is zero, then it's the empty string which doesn't contain the substring and is in S. Otherwise, x can be composed of any number of zeroes or pairs of ones. Appending pairs of ones cannot make the length of any substring of ones odd. Hence, we can see that x can be built from the empty string by applying the recursive step from x_1 to x_2 to x_n, each of which maintains the condiditon that $x \in S$.

3(c)

Basis Step: $x \in \{1\}$
Recursive Step: If $x \in S$ then $x1, x00, 00x \in S$ Exclusion Rule: Each element of S is obtained from the basis and a finite number of applications of the recursive step
Justification: We show that $x \in S$ iff x contains at least one 1 and an even number of 0s.

Suppose $x \in S$. If x is the basis case, then x contains at least one 1 and, because it has no 0s, an even number of 0s. Otherwise, the recursive step can only append 0s in pairs, and from above, $2n$ will always be even. Hence, every string in S satisfies the conditions.

Now, suppose that x contains at least one 1 and an even number of 0s. For some string it will always contain a 1 by the basis case and because we can build up the string by recursively applying 2n 0s or some number of 1s, it will remain in S. Hence, $x \in S$.

3(d)

Basis Step: $\epsilon \in S$
Recursive Step: If $x \in S$ then $(x), x(), ()x \in S$
Exclusion Rule: Each element of S is obtained from the basis and a finite number of applications of the recursive step
Justification: We show that $x \in S$ iff x is contains only substrings of properly nested and paired parentheses.

Suppose $x \in S$. If x is the empty string then $x \in S$. Otherwise, x can only be built by appending correctly formed substring pairs of parentheses. That is, there will never be a single parenthesis, nor two facing the wrong direction, and because the string is recursively built, well formed pair on top of well formed pair, it cannot be poorly formed. Hence, every string x in S satisfies the property.

Now, suppose x contains only substrings of properly nested and paired parentheses. If it contains no parentheses then it is the empty string which is in S. Otherwise, because we can build up the

string x from the empty string by applying the rules $x \to (x), ()x, ()x$ and then produce the string by applying that rule some number of times, the string must be in S.

4 Tied Up With Strings (Submitted Online)

5 Wish List

5(a)

$$\begin{aligned}
\text{remove}(\text{map}_f(L)) &= \text{remove}(\text{map}_f(\text{Node}(1, \text{Node}(2, \text{Node}(3, \text{null}))))) && \text{Replace L} \\
&= \text{remove}(\text{Node}(f(1), \text{map}_f(\text{Node}(2, \text{Node}(3, \text{null}))))) && \text{Definition of map}_f \\
&= \text{remove}(\text{Node}(3, \text{Node}(f(2), \text{map}_f(\text{Node}(3, \text{null}))))) && \text{Definition of map}_f \\
&= \text{remove}(\text{Node}(3, \text{Node}(5, \text{Node}(f(3), \text{map}_f(\text{null}))))) && \text{Definition of map}_f \\
&= \text{remove}(\text{Node}(3, \text{Node}(5, \text{Node}(7, \text{null})))) && \text{Definition of map}_f \\
&= \text{shift}(\text{Node}(3, \text{Node}(5, \text{Node}(7, \text{null}))), \text{null}) && \text{Definition of remove()} \\
&= \text{shift}(\text{Node}(5, \text{Node}(7, \text{null})), \text{Node}(3, \text{null})) && \text{Definition of shift()} \\
&= \text{shift}(\text{Node}(7, \text{null}), \text{Node}(5, \text{Node}(3, \text{null}))) && \text{Definition of shift()} \\
&= \text{shift}(\text{null}, \text{Node}(7, \text{Node}(5, \text{Node}(3, \text{null})))) && \text{Definition of shift()} \\
&= \text{Node}(7, \text{Node}(5, \text{Node}(3, \text{null}))) && \text{Definition of shift()}
\end{aligned}$$

5(b)

1. Let $P(N)$ for some $N \in$ Lists, be the claim that $\text{map}_f(\text{shift}(N, R)) = \text{shift}(\text{map}_f(N), \text{map}_f(R))$. Prove $P(N)$ by structural induction.
2. Base Case: The base case $P(\text{null})$ holds because $\text{map}_f(\text{shift}(\text{null}), R) = \text{map}_f(R) = \text{shift}(\text{null}, \text{map}_f(R)) = \text{shift}(\text{map}_f(\text{null}), \text{map}_f(R))$.
3. Inductive Hypothesis: Assume $P(M)$ for an arbitrary $M \in$ Lists.
4. Inductive Step: Goal, show $P(\text{Node}(x, M))$ for any $x \in \mathbb{Z}$, i.e. $\text{map}_f(\text{shift}(\text{Node}(x, M))) = \text{shift}(\text{map}_f(\text{Node}(x, M)), \text{map}_f(R))$.

$$\begin{aligned}
\text{map}_f(\text{shift}(\text{Node}(x, M)), R) &= \text{map}_f(\text{shift}(M, \text{Node}(x, R))) && \text{Definition of shift()} \\
&= \text{shift}(\text{map}_f(M), \text{map}_f(\text{Node}(x, R))) && \text{Inductive Hypothesis} \\
&= \text{shift}(\text{map}_f(M), \text{Node}(f(x), \text{map}_f(R))) && \text{Definition of map}_f() \\
&= \text{shift}(\text{Node}(f(x), \text{map}_f(M)), \text{map}_f(R)) && \text{Definition of shift()} \\
&= \text{shift}(\text{map}_f(\text{Node}(x, M)), \text{map}_f(R)) && \text{Definition of map}_f()
\end{aligned}$$

5. Conclusion: Therefore, by structural induction we show that P(n) holds. $\square$

5(c)

1. Prove $P(N)$ for some $N \in$ Lists, be the claim that $\text{map}_f(\text{reverse}(N)) = \text{reverse}(\text{map}_f(N))$.
2. Base Case: The base case, $P(\text{null})$ holds because $\text{map}_f(\text{reverse}(\text{null})) = \text{map}_f(\text{null}) = \text{reverse}(\text{map}_f(\text{null}))$.
3. Inductive Hypothesis: Assume $P(M)$ holds for an arbitrary $M \in$ Lists.
4. Inductive Step: Goal, show $P(\text{Node}(x, M))$ for an arbitrary $x \in Z$, i.e. $\text{map}_f(\text{reverse}(\text{Node}(x, M))) = \text{reverse}(\text{map}_f(\text{Node}(x, M)))$.

$$\begin{aligned}
\text{map}_f(\text{reverse}(\text{Node}(x, M))) &= \text{map}_f(\text{shift}(\text{Node}(x, M), \text{null})) && \text{Definition of reverse()} \\
&= \text{shift}(\text{map}_f(\text{Node}(x, M)), \text{map}_f(\text{null})) && \text{5.b Proof} \\
&= \text{shift}(\text{map}_f(\text{Node}(x, M)), \text{null}) && \text{Definition of map}_f() \\
&= \text{reverse}(\text{map}_f(\text{Node}(x, M))) && \text{Definition of reverse()}
\end{aligned}$$

5. Conclusion: Therefore, we've shown that the claim $P(N)$ holds. $\square$