Recursion and Induction

Problem 1
Prove by induction that

$$1 \cdot 1! + 2 \cdot 2! + \ldots + n \cdot n! = (n+1)! - 1 \quad \text{for } n \geq 1$$

Solution

Let $P(n)$ be the proposition that $1 \cdot 1! + 2 \cdot 2! + \ldots + n \cdot n! = (n+1)! - 1$. We will prove that $P(n)$ holds for all $n \geq 1$ by induction on n.

Base case $n = 1$. $1.1! = 1 = 2! - 1 = (1+1)! - 1$ so $P(1)$ holds.

Inductive case. Assume $P(k)$ holds for some $k \in \mathbb{N}_{>0}$. That is $1 \cdot 1! + 2 \cdot 2! + \ldots + k \cdot k! = (k+1)! - 1$. Then

$$\begin{aligned} 1 \cdot 1! + 2 \cdot 2! + \ldots + k \cdot k! + (k+1) \cdot (k+1)! &= (k+1)! - 1 + (k+1) \cdot (k+1)! \quad \text{(Induction hypothesis)} \\ &= (1+k+1)(k+1)! - 1 \\ &= ((k+1)+1)(k+1)! - 1 \end{aligned}$$

so $P(k+1)$ holds.
Therefore, by the Principle of Induction, $P(n)$ holds for all $n \geq 1$.

Problem 2
Let $\Sigma = \{1, 2, 3\}$.

(a) Give a recursive definition for the function $\mathsf{sum} : \Sigma^* \to \mathbb{N}$ which, when given a word over Σ returns the sum of the digits. For example $\mathsf{sum}(1232) = 8$, $\mathsf{sum}(222) = 6$, and $\mathsf{sum}(1) = 1$. You should assume $\mathsf{sum}(\lambda) = 0$.

(b) For $w \in \Sigma^*$, let $P(w)$ be the proposition that for all words $v \in \Sigma^*$, $\mathsf{sum}(wv) = \mathsf{sum}(w) + \mathsf{sum}(v)$. Prove that $P(w)$ holds for all $w \in \Sigma^*$.

(c) Consder the function $\mathsf{rev} : \Sigma^* \to \Sigma^*$ defined recursively as follows:

- $\mathsf{rev}(\lambda) = \lambda$
- For $w \in \Sigma^*$ and $a \in \Sigma$, $\mathsf{rev}(aw) = \mathsf{rev}(w)a$

Prove that for all words $w \in \Sigma^*$, $\mathsf{sum}(\mathsf{rev}(w)) = \mathsf{sum}(w)$

Solution

(a) We give a definition using the recursive nature of Σ^*:

$$\begin{array}{rcl} \mathsf{sum}(\lambda) & = & 0 \\ \mathsf{sum}(a.w) & = & a + \sum(w). \end{array}$$

(b) We first need the recursive definition of concatenation:

$$\begin{array}{rcl} \lambda.v & = & v \\ (aw).v & = & a(w.v) \end{array}$$

We will now prove $P(w)$ for all $w \in \Sigma^*$ by structural induction on w.

Base case ($w = \lambda$).

$$\begin{array}{rll} \mathsf{sum}(wv) & = \mathsf{sum}(\lambda.v) & \\ & = \mathsf{sum}(v) & \text{Definition of concatenation} \\ & = 0 + \mathsf{sum}(v) & \\ & = \mathsf{sum}(\lambda) + \mathsf{sum}(v) & \text{Definition of sum} \\ & = \mathsf{sum}(w) + \mathsf{sum}(v) & \end{array}$$

So $P(\lambda)$ holds.

Inductive case ($w = aw'$). Assume $P(w')$ holds, that is for all $v \in \Sigma^*$, $\mathsf{sum}(w'v) = \mathsf{sum}(w') + \mathsf{sum}(v)$. Then for all $v \in \Sigma^*$ and all $a \in \Sigma$:

$$\begin{array}{rll} \mathsf{sum}((aw')v) & = \mathsf{sum}(a(w'v)) & \text{Definition of concatenation} \\ & = a + \mathsf{sum}(w'v) & \text{Definition of sum} \\ & = a + \mathsf{sum}(w') + \mathsf{sum}(v) & \text{Inductive hypothesis} \\ & = \mathsf{sum}(aw') + \mathsf{sum}(v) & \text{Definition of sum} \end{array}$$

So $P(w')$ implies $P(aw')$ for all $a \in \Sigma$.

Therefore, by the Principle of Structural Induction, $P(w)$ holds for all $w \in \Sigma^*$.

(c) Let $P(w)$ be the proposition that $\mathsf{sum}(\mathsf{rev}(w)) = \mathsf{sum}(w)$. We will show that $P(w)$ holds for all words $w \in \Sigma^*$ by structural induction on w.

Base case ($w = \lambda$). From the definition of rev we have: $\mathsf{sum}(\mathsf{rev}(\lambda)) = \mathsf{sum}(\lambda)$. So $P(\lambda)$ holds.

Inductive case ($w = aw'$). Suppose $P(w')$ holds, that is $\mathsf{sum}(\mathsf{rev}(w')) = \mathsf{sum}(w')$. For any $a \in \Sigma$ we have:

$$\begin{array}{rll} \mathsf{sum}(\mathsf{rev}(aw')) & = \mathsf{sum}(w'a) & \text{Definition of rev} \\ & = \mathsf{sum}(w') + \mathsf{sum}(a) & \text{From (b)} \\ & = \mathsf{sum}(w') + a + \mathsf{sum}(\lambda) & \text{Definition of sum} \\ & = a + \mathsf{sum}(w') + 0 & \text{Definition of sum} \\ & = \mathsf{sum}(aw') & \text{Definition of sum} \end{array}$$

So $P(w')$ implies $P(aw')$ for all $a \in \Sigma$.

Therefore, by the Principle of Structural Induction, $P(w)$ holds for all $w \in \Sigma^*$.

Problem 3

Define $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ recursively as follows: $f(m, 0) = 0$ for all $m \in \mathbb{N}$ and $f(m, n+1) = m + f(m, n)$.

(a) Let $P(n)$ be the proposition that $f(0, n) = f(n, 0)$. Prove that $P(n)$ holds for all $n \in \mathbb{N}$.

*(b) Let $Q(m)$ be the proposition $\forall n,\ f(m, n) = f(n, m)$. Prove that $Q(m)$ holds for all $m \in \mathbb{N}$.

Solution

1. We show that $P(n)$ holds for all $n \in \mathbb{N}$ by induction.

   Base case: $n = 0$**.** Since $f(0, 0) = f(0, 0)$, $P(0)$ holds.

   Inductive case. Now suppose $P(n)$ holds. Then

   $$\begin{aligned} f(0, n+1) &= 0 + f(0, n) && \text{(Def)} \\ &= 0 + f(n, 0) && \text{(IH)} \\ &= 0 && \text{(Def)} \\ &= f(n+1, 0). && \text{(Def)} \end{aligned}$$

   So $P(n) \to P(n+1)$, and thus $P(n)$ holds for all $n \in \mathbb{N}$.

2. We will prove by induction that $f(m, n) = mn$, from which it follows that $f(m, n) = mn = nm = f(n, m)$. Let $R(n)$ be the proposition that: for all m, $f(m, n) = mn$.

   Base case: $n = 0$**.** From the definition of f, $f(m, 0) = 0 = 0.m$ for all m. So $R(0)$ holds.

   Inductive case. Suppose that $R(n)$ holds. That is, for all m, $f(m, n) = mn$. Then, for all m,

   $$\begin{aligned} f(m, n+1) &= m + f(m, n) && \text{Definition of } f \\ &= m + mn && \text{Induction hypothesis} \\ &= m(n+1). \end{aligned}$$

   So $R(n+1)$ holds. Thus, $R(n)$ implies $R(n+1)$, so by the Principle of Induction $f(m, n) = mn$ for all m and n. Therefore $f(m, n) = f(n, m)$.

Problem 4† (20T2)

Let $\Sigma = \{a, b\}$ and define $f : \Sigma^* \to \mathbb{R}$ recursively as follows:

- $f(\lambda) = 0$,
- $f(aw) = \frac{1}{2} + \frac{1}{2} f(w)$ for $w \in \Sigma^*$, and
- $f(bw) = -\frac{1}{2} + \frac{1}{2} f(w)$ for $w \in \Sigma^*$.

(a) What is $f(abba)$?

(b) Prove that $f(w) \in (-1, 1)$ for all $w \in \Sigma^*$

(c) Prove, or give a counterexample to disprove:

  (i) f is injective

  (ii) $Im(f) = (-1, 1)$

Solution

(a) We have

$$
\begin{aligned}
f(\lambda) &= 0 \\
f(a) &= \tfrac{1}{2} + \tfrac{1}{2} f(\lambda) && = \tfrac{1}{2} \\
f(ba) &= -\tfrac{1}{2} + \tfrac{1}{2} f(a) && = -\tfrac{1}{2} + \tfrac{1}{4} && = -\tfrac{1}{4} \\
f(bba) &= -\tfrac{1}{2} + \tfrac{1}{2} f(ba) && = -\tfrac{1}{2} - \tfrac{1}{8} && = -\tfrac{5}{8} \\
f(abba) &= \tfrac{1}{2} + \tfrac{1}{2} f(bba) && = \tfrac{1}{2} - \tfrac{5}{16} && = \tfrac{3}{16}
\end{aligned}
$$

(b) For $w \in \Sigma^*$, let $P(w)$ be the proposition that $f(w) \in (-1, 1)$. We will show $P(w)$ is true for all $w \in \Sigma^*$, by structural induction on w.

- Base case: $w = \lambda$.
  We have $f(\lambda) = 0 \in (-1, 1)$, so $P(\lambda)$ is true.
- Inductive case: Suppose $P(w')$ holds for $w' \in \Sigma^*$. That is $f(w') \in (-1, 1)$. Then $\frac{1}{2} f(w') \in (-\frac{1}{2}, \frac{1}{2})$.
  Consider $w = aw'$. We have

  $$f(w) = f(aw') = \frac{1}{2} + \frac{1}{2} f(w') \in (\frac{1}{2} - \frac{1}{2}, \frac{1}{2} + \frac{1}{2}) = (0, 1) \subseteq (-1, 1)$$

  So $P(aw')$ holds.
  Now consider $w = bw'$. We have

  $$f(w) = f(bw') = -\frac{1}{2} + \frac{1}{2} f(w') \in (-\frac{1}{2} - \frac{1}{2}, -\frac{1}{2} + \frac{1}{2}) = (-1, 0) \subseteq (-1, 1)$$

  So $P(bw')$ holds.
  Therefore, if $P(w')$ holds, $P(xw')$ holds for all $x \in \Sigma$.

It follows, by structural induction, that $P(w)$ holds for all $w \in \Sigma^*$.

(c) (i) This is true. Suppose f is not injective. Let $f(x) = f(y)$ with $x \neq y$, and choose x so that $\text{length}(()x)$ is minimal. That is, if $f(x') = f(y')$ and $\text{length}(()x') < \text{length}(()x)$ then $x' = y'$.

From (b), we have the following observation:

- $f(\lambda) = 0$
- $f(aw) \in (0, 1)$, so $f(aw) > 0$, and
- $f(bw) \in (-1, 0)$, so $f(bw) < 0$.

We also observe, from algebraic manipulation, that for all $w \in \Sigma^*$:

$$f(w) = 2f(aw) - 1 = 2f(bw) + 1.$$

Now, if $x = \lambda$ then $f(y) = 0$, so $y = \lambda$ which is a contradiction.
If $x = ax'$ then $f(x) > 0$, so f(y)>o, so $y = ay'$. But then

$$f(x') = 2f(x) - 1 = 2f(y) - 1 = f(y')$$

so $x' = y'$ (by the minimality of x). But then $x = ax' = ay' = y$, which is a contradiction.
Finally, if $x = bx'$ then $f(x) < 0$, so f(y)<o, so $y = by'$. But then

$$f(x') = 2f(x) + 1 = 2f(y) + 1 = f(y')$$

so $x' = y'$ (by the minimality of x). But then $x = bx' = by' = y$, which is a contradiction.
Therefore f is injective.

(ii) This is false. We observe the following:

- $f(\lambda) = 0$ is rational.
- If $f(w)$ is rational, then $f(aw) = \frac{1}{2} + \frac{1}{2}f(w)$ and $f(bw) = -\frac{1}{2} + \frac{1}{2}f(w)$ are both rational.

It follows, by structural induction, that $f(w)$ is rational for all $w \in \Sigma^*$.
But then there is no $w \in \Sigma^*$ such that $f(w) = \frac{1}{\sqrt{2}} \in (-1, 1)$. Therefore $Im(f) \neq (-1, 1)$.

Alternative proof. We will show that there is no w such that $f(w) = \frac{1}{3}$.
Suppose $f(w) = \frac{1}{3}$. As observed in (c)(i), since $f(w) > 0$, we have $w = aw'$ for some word w'. Therefore,

$$f(w') = 2f(w) - 1 = \frac{2}{3} - 1 = -\frac{1}{3}.$$

Now, as observed in (c)(ii), since $f(w') < 0$, we have $w' = bw''$ for some word w''. But then,

$$f(w'') = 2f(w') + 1 = -\frac{2}{3} + 1 = \frac{1}{3} = f(w)$$

Since f is injective (as shown in (b)), we have $w = w''$. However, $w = aw' = abw''$, so this is a contradiction.
Therefore, there is no w such that $f(w) = \frac{1}{3}$.

Problem 5
Let $\Sigma = \{0, 1\}$

(a) Recursively define a function $\mathsf{str2num} : \Sigma^+ \to \mathbb{N}$ that converts a non-empty word over Σ to the number that one obtains by viewing the word as a binary number. For example $\mathsf{str2num}(1100) = 12$, $\mathsf{str2num}(0111) = 7$, $\mathsf{str2num}(0000) = 0$.

(b) Recursively define a function $\mathsf{num2str} : \mathbb{N} \to \Sigma^+$ that converts a number to its (shortest) binary representation. *Hint: you may want to use* div *and* %.

(c) Writing your functions as code in the natural way,

(i) Give an asymptotic upper bound in terms of $\mathsf{length}(()w)$ on the running time to compute $\mathsf{str2num}(w)$.

(ii) Give an asymptotic upper bound in terms of n on the running time to compute $\mathsf{num2str}(n)$.

Solution

(a) One approach:

- $\mathsf{str2num}(0) = 0$
- $\mathsf{str2num}(1) = 1$
- $\mathsf{str2num}(0w) = \mathsf{str2num}(w)$ for $w \in \Sigma^+$
- $\mathsf{str2num}(1w) = 2^{\text{length}(w)} + \mathsf{str2num}(w)$ for $w \in \Sigma^+$

(b) One approach:

- $\mathsf{num2str}(0) = 0$
- $\mathsf{num2str}(1) = 1$
- $\mathsf{num2str}(n) = \mathsf{num2str}(n \text{ div } 2).\mathsf{num2str}(n \ \% \ 2)$ where . is string concatenation, for $n \geq 2$.

Solution (ctd)

(c) (i) The "natural" code is:

```
str2num(w) :
  if w = 0:
    return 0
  else if w = 1:
    return 1
  else if w = 0w':
    return str2num(w')
  else if w = 1w':
    return 2^length(w') + str2num(w')
```

The running time for each of these lines, excluding the recursive calls, is $O(1)$. Computing $\mathsf{length}(w)$ takes $O(\mathsf{length}(w))$ time unless we store w "smartly" using a complex data structure that keeps track of the length of w. If we let $T(n)$ denote the running time of $\mathsf{str2num}(w)$ when $\mathsf{length}(w) = n$ we see that the first recursive call (line 6) will take $O(1) + T(n-1)$ time; whereas the second call (line 8) will take $O(1) + O(n) + T(n-1)$ time. In the worst case, we will always execute the statement that takes the longest time giving us the following recurrence for $T(n)$:

$$T(1) \in O(1) \qquad T(n) \leq O(1) + O(n) + T(n-1).$$

Therefore, using the linear form of the Master Theorem, $T(n) \in O(n^2)$.

(ii) The "natural" code is:

```
num2str(n) :
  if n = 0:
    return 0
  else if n = 1:
    return 1
  else:
    return num2str(n div 2).num2str(n % 2)
```

Let $T(n)$ denote the running time of $\mathsf{num2str}(n)$. We observe that $\mathsf{num2str}(n \;\%\; 2)$ will execute in $O(1)$ time, and with a suitable method of storing words, concatenating the single symbol will also take $O(1)$ time. So the final line will take $T(n/2) + O(1)$ time to execute. For $n \geq 2$ this line will always get executed, giving us the following recurrence for $T(n)$:

$$T(0), T(1) \in O(1) \qquad T(n) \leq O(1) + T(n/2)$$

Using the Master Theorem, we have $a = 1$, $b = 2$, $c = d = 0$, so we are in Case 2, and $T(n) \in O(\log n)$.