Solution to Assignment 4

Jeremy Johnson

December 2, 2017

Question 1a

Theorem 1

$$\sum_{i=1}^{n} ba_i = b\sum_{i=1}^{n} a_i$$

Proof. The proof is by induction on n.

Base Case. The base case, $n = 1$, is proved by evaluating the left and right sides of the statement when $n = 1$.

$$\sum_{i=1}^{1} ba_i = ba_1 = b\sum_{i=1}^{1} a_i$$

Inductive Case. Assume the inductive hypothesis

$$\sum_{i=1}^{n} ba_i = b\sum_{i=1}^{n} a_i$$

and show

$$\sum_{i=1}^{n+1} ba_i = b\sum_{i=1}^{n+1} a_i$$

$$\begin{aligned}
\sum_{i=1}^{n+1} ba_i &= \sum_{i=1}^{n} ba_i + ba_{n+1} \text{ [By definition of } \sum \text{]} \\
&= b\sum_{i=1}^{n} a_i + ba_{n+1} \text{[By inductive hypothesis]} \\
&= b\left(\sum_{i=1}^{n} a_i + a_{n+1}\right) \text{ [By disbributive law]} \\
&= b\sum_{i=1}^{n+1} a_i \text{ [By definition of } \sum \text{]}
\end{aligned}$$

Question 1b

Theorem 2
$$\sum_{i=1}^{n}(a_i + b_i) = \sum_{i=1}^{n} a_i + \sum_{i=1}^{n} b_i$$

Proof. The proof is by induction on n.

Base Case. The base case, $n = 1$, is proved by evaluating the left and right sides of the statement when $n = 1$.

$$\sum_{i=1}^{1}(a_i + b_i) = a_1 + b_1 = \sum_{i=1}^{1} a_i + \sum_{i=1}^{1} b_i$$

Inductive Case. Assume the inductive hypothesis

$$\sum_{i=1}^{n}(a_i + b_i) = \sum_{i=1}^{n} a_i + \sum_{i=1}^{n} b_i$$

and show

$$\sum_{i=1}^{n+1}(a_i + b_i) = \sum_{i=1}^{n+1} a_i + \sum_{i=1}^{n+1} b_i$$

$$\begin{aligned}
\sum_{i=1}^{n+1}(a_i + b_i) &= \sum_{i=1}^{n}(a_i + b_i) + (a_{n+1} + b_{n+1}) \text{ [By definition of } \textstyle\sum \text{]} \\
&= \left(\sum_{i=1}^{n} a_i + \sum_{i=1}^{n} b_i\right) + (a_{n+1} + b_{n+1}) \text{[By inductive hypothesis]} \\
&= \left(\sum_{i=1}^{n} a_i + a_{n+1}\right) + \left(\sum_{i=1}^{n} b_i + b_{n+1}\right) \text{ [By associative law and commutative law]} \\
&= \sum_{i=1}^{n+1} a_i + \sum_{i=1}^{n+1} b_i \text{ [By definition of } \textstyle\sum \text{]}
\end{aligned}$$

Technically we use a lemma that states $(x + y) + (w + z) = (x + w) + (y + z)$, which can be proved by repeated application of the associative and commutative laws.

Question 2

Theorem 3 *Prove by induction that*

$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}.$$

You should compute an example to make sure you understand the statement.

Proof. The proof is by induction on n.

Base Case. The base case, $n = 1$,

$$\sum_{i=1}^{1} i^2 = 1 = \frac{1(1+1)(2+1)}{6}$$

Inductive Case. Assume the inductive hypothesis

$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6},$$

and show

$$\sum_{i=1}^{n+1} i^2 = \frac{(n+1)((n+1)+1)(2(n+1)+1)}{6} = \frac{(n+1)(n+2)(2n+3)}{6}.$$

$$\begin{aligned}
\sum_{i=1}^{n+1} i^2 &= \sum_{i=1}^{n} i^2 + (n+1)^2 \text{ [By property of } \textstyle\sum \text{]} \\
&= \frac{n(n+1)(2n+1)}{6} + (n+1)^2 \text{ [By inductive hypothesis]} \\
&= \frac{n(n+1)(2n+1) + 6(n+1)^2}{6} \text{ [By addition of fractions]} \\
&= \frac{(n+1)\left[n(2n+1) + 6(n+1)\right]}{6} \text{ [By factoring]} \\
&= \frac{(n+1)\left[2n^2 + 7n + 6\right]}{6} \text{ [By expanding]} \\
&= \frac{(n+1)(n+2)(2n+3)}{6} \text{ [By factoring]}
\end{aligned}$$

Question 3

Theorem 4 *A truth table with n variables has 2^n rows.*

Proof. The proof is by induction on n the number of variables.

Base Case. The base case, when $n = 1$ is proved by constructing a truth table with one variable, which has two rows, one with F and one with T.

Inductive Case. Assume the inductive hypothesis that a truth table with n variables has 2^n rows.

To construct a truth table with $n+1$ variables construct a truth table with all but the first variable and then set the first variable first to F and then to T. By induction the truth with all but the first variable has 2^n rows and there are two copies of this, one when the first variable is F and one when it is T. Thus the number of rows is $2 \cdot 2^n = 2^{n+1}$.

Question 4

Theorem 5 *Let $x_1, \ldots, x_n$ be boolean variables. Then*

$$x_1 \oplus \cdots \oplus x_n = \mathit{parity}(x_1, \ldots, x_n),$$

where $x_1 \oplus \cdots \oplus x_n = (x_1 \oplus \cdots \oplus x_{n-1}) \oplus x_n$ and in the base case when $n = 1$ is equal to x_1.

Proof. The proof is by induction on n the number of variables.

Base Case. The base case, when $n = 1$ follows from $\text{parity}(x) = x$.

Inductive Case. Assume the inductive hypothesis

$$x_1 \oplus \cdots \oplus x_n = \text{parity}(x_1, \ldots, x_n).$$

Then

$$x_1 \oplus \cdots \oplus x_{n+1} = (x_1 \oplus \cdots \oplus x_n) \oplus x_{n+1}$$

which by the inductive hypothesis is equal to

$$\text{parity}(x_1, \ldots, x_n) \oplus x_{n+1}.$$

When $x_{n+1} = 0$, $\text{parity}(x_1, \ldots, x_n) \oplus x_{n+1} = \mathit{parity}(x_1, \ldots, x_n, x_{n+1})$, since $x \oplus 0 = x$, and $\text{parity}(x_1, \ldots, x_n, 0) = \text{parity}(x_1, \ldots, x_n)$.

When $x_{n+1} = 1$, $\text{parity}(x_1, \ldots, x_n) \oplus 1 = \overline{\text{parity}(x_1, \ldots, x_n)}$ which is equal to $\text{parity}(x_1, \ldots, x_n, 1)$, since $x \oplus 1 = \overline{x}$, where $\overline{x}$ is the complement of x (x inverted).

Question 5

Theorem 6 *The binary reflected Gray code G_n is a gray code, i.e. exactly one bit changes from one entry to the next, including the first and last entries, and all n-bit binary numbers are included.*

Proof. The proof is by induction on n the number of bits.

Base Case. In the base case, when $n = 1$ $G_1 = [0, 1]$ which is a Gray code.

Inductive Case. Assume the inductive hypothesis that G_{n-1} is a Gray code and show that the recursive construction $G_n = [0G_{n-1}, 1G'_{n-1}]$ is a Gray code.

Observe that if G is a Gray code, then $0G$ and $1G$ satisfy the property that exactly one bit changes from one entry to the next. Also observe that if G is a Gray code then G' is a Gray code. Therefore since by induction G_{n-1} is a Gray code $0G_{n-1}$ and $1G'_{n-1}$ satisfy the property that exactly one bit changes from one entry to the next, and since all n-bit numbers are obtained from $(n-1)$-bit numbers by inserting zero and one bits in front of each $(n-1)$-bit number all n-bit numbers are contained in the concatenation of $0G_{n-1}$ and $1G'_{n-1}$. Finally, since the first element of G'_{n-1} is equal to the last element of G_{n-1} the last element of $0G_{n-1}$ differs by exactly one bit from the first element of $1G'_{n-1}$. Similarly, since the first element of G_{n-1} is equal to the last element of G'_{n-1}, the first element of $0G_{n-1}$ differs by exactly one bit from the last element of $1G'_{n-1}$. Thus we conclude that G_n is a Gray code.

Question 6

Let L be a list of length $n > 0$ with $L = (L_1 \ldots L_n)$. Prove using induction that the i-th element of (reverse L) is the $(n+1-i)$-th element of L. What is the base case?

```
(define (reverse l)
  (if (null? l)
    null
    (append (reverse (rest l)) (cons (first l) null))))
```

You may assume the following property of append

- $0 < i \leq$ (length x) implies the i-th element of (append x y)) = i-th element of x.
- (length x) $< i \leq$ (length x) + (length y) implies i-th element of (append x y) = (i – (length x)-th element of y.

Proof. The proof is by induction on n the length of L. Let R = (reverse L).

Base Case. In the base case, when $n = 1$ (reverse '(L_1)) = (append '() (cons (first '(L_1)) null) = '(L_1) and $L_{1+1-1} = L_1$.

Inductive Case. For the inductive hypothesis, assume that the property holds for lists of length less than n. In particular that it holds for L' = (rest L) which has length $n - 1$. Let R' = (reverseL'). By property 2 of reverse, the length of R' is equal to the length of L'. The inductive hypothesis becomes $R'_i = L'_{n-i}$.

Show that this implies $R_i = L_{n+1-i}$.

- (reverse L) = (append (reverse L') (cons L_1 null)) [By def of reverse]

By the above property of append there are two cases.

- $0 < i \leq n-1$. $R_i = R'_i = L'_{n-i} = L_{n+1-i}$ [By inductive hypothesis and since the i-th element of L' is the $(i+1)$-st element of L.]
- $i = n$. $R_n = L_1 = L_{n+1-n}$ [By def of reverse.]

Extra Credit

Theorem 7 *Let $G_n(i) = i$ ^ $(i >> 1)$ be a function from $[0, \ldots, 2^n - 1]$ to $[0, \ldots, 2^n - 1]$. Then $[G_n(0), \ldots, G_n(2^n - 1)]$ is the binary reflected gray code.*

Proof. The proof is by induction on n the number of bits.

Base Case. In the base case, when $n = 1$ $[G_1(0), G_1(1)] = [0, 1]$.

General Case. Assume the inductive hypothesis that $[G_n(0), \ldots, G_n(2^n - 1)]$ is the binary reflected gray code G_n on n bits and show that $[G_{n+1}(0), \ldots, G_{n+1}(2^{n+1} - 1)]$ is equal to the binary reflected gray code on $n + 1$ bits.

Let $b_n b_{n-1} \cdots b_1 b_0$ be the binary representation of i. Then $G_{n+1}(i)$ is the bitwise exclusive or of $b_n b_{n-1} \cdots b_1 b_0$ and $0 b_{n-1} \cdots b_1$. When $b_n = 0$ this is equal to $G_n(b_{n-1} \cdots b_0)$ with a zero prepended, and by induction $G_n(0), \ldots, G_n(2^n - 1)]$ is a binary reflected gray code.

When $b_n = 1$ this is equal to $G_n(b_{n-1} \cdots b_0)$ with the leading bit complemented by induction and the fact that $b_{n-1} \oplus 1 = \overline{b_{n-1}}$.

Since complementing the leading bit of the binary reflected gray code is the reflected binary gray code we have shown that $[G_{n+1}(0), \ldots, G_{n+1}(2^{n+1} - 1)] = [0G_n, 1G'_n] = G_{n+1}$.

The last remark follows from the construction of the binary reflected gray code. Complementing the leading bit of the n-bit binary reflected gray code is equal to $[1G_n, 0G'_n]$ which is equal to $[0G_n, 1G'_n]' = [1G''_n, 0G'_n]$ since $G''_n = G_n$.