

COMPSCI/SFWRENG 2FA3  
Discrete Mathematics with Applications II  
Winter 2020

## Assignment 2

Dr. William M. Farmer  
McMaster University

Revised: January 25, 2020

Assignment 2 consists of two problems. You must write your solutions to the problems using LaTeX.

Please submit Assignment 2 as two files, `Assignment_2_YourMacID.tex` and `Assignment_2_YourMacID.pdf`, to the Assignment 2 folder on Avenue under Assessments/Assignments. *YourMacID* must be your personal MacID (written without capitalization). The `Assignment_2_YourMacID.tex` file is a copy of the LaTeX source file for this assignment (`Assignment_2.tex` found on Avenue under Contents/Assignments) with your solution entered after each problem. The `Assignment_2_YourMacID.pdf` is the PDF output produced by executing

```
pdflatex Assignment_2_YourMacID
```

This assignment is due **Sunday, February 2, 2020 before midnight**. You are allowed to submit the assignment multiple times, but only the last submission will be marked. **Late submissions and files that are not named exactly as specified above will not be accepted!** It is suggested that you submit your preliminary `Assignment_2_YourMacID.tex` and `Assignment_2_YourMacID.pdf` files well before the deadline so that your mark is not zero if, e.g., your computer fails at 11:50 PM on February 2.

**Although you are allowed to receive help from the instructional staff and other students, your submission must be your own work. Copying will be treated as academic dishonesty! If any of the ideas used in your submission were obtained from other students or sources outside of the lectures and tutorials, you must acknowledge where or from whom these ideas were obtained.**

## Problems

1. [10 points] Let  $C$  be a set of *coefficients* closed under addition and multiplication such as the integers  $\mathbb{Z}$ , the rational numbers  $\mathbb{Q}$ , or the real numbers  $\mathbb{R}$ . A *polynomial over  $C$*  is a mathematical expression that is constructed from an *indeterminant*  $x$  and members of  $C$  by applying addition (+) and multiplication (\*) operators. For example,  $x * ((2 * x) + 3)$  is a polynomial over  $\mathbb{Z}$ . Let  $P$  be the set of polynomials over some  $C$ . The *value of a polynomial  $p \in P$  at  $c \in C$*  is the result of replacing the indeterminant  $x$  with  $c$ . For example, the value of  $x * ((2 * x) + 3)$  at 5 is  $5 * ((2 * 5) + 3) = 65$ .

Let Poly be the inductive type defined by the following constructors:

$X : \text{Poly}.$

$\text{Coeff} : \mathbb{Q} \rightarrow \text{Poly}.$

$\text{Sum} : \text{Poly} \times \text{Poly} \rightarrow \text{Poly}.$

$\text{Prod} : \text{Poly} \times \text{Poly} \rightarrow \text{Poly}.$

The members of Poly represent the polynomials over  $\mathbb{Q}$ . Define

$\text{val} : \text{Poly} \times \mathbb{Q} \rightarrow \mathbb{Q}$

by structural recursion using pattern matching so that, for all  $p \in \text{Poly}$  and  $q \in \mathbb{Q}$ ,  $\text{val}(p, q)$  is the value of  $p$  at  $q$ .

**Name:** Hishmat Salehi

**MacID:** salehh6

**Date:** February 2, 2020

$\text{val}(X, q) = q$

$\text{val}(\text{Coeff}(a), q) = a$ , where  $a \in \mathbb{Q}$

$\text{val}(\text{Sum}(a, b), q) = \text{val}(a, q) + \text{val}(b, q)$ , where  $a, b \in \text{Poly}$

$\text{val}(\text{Prod}(a, b), q) = \text{val}(a, q) * \text{val}(b, q)$ , where  $a, b \in \text{Poly}$

2. [10 points] Let Bit be the inductive set defined by the following constructors:

$\text{Zero} : \text{Bit}.$

$\text{One} : \text{Bit}.$

The members of Bit represent 0 and 1.

Let BinNum be the inductive set defined by the following constructors:

$\text{Nil} : \text{BinNum}.$

$\text{Join} : \text{BinNum} \times \text{Bit} \rightarrow \text{BinNum}.$

The members of `BinNum` not equal to `Nil` represent binary numerals; `Nil` represents an empty numeral. For example,

$$\text{Join}(\text{Join}(\text{Join}(\text{Nil}, \text{One}), \text{Zero}), \text{One}),$$

represents the binary number 101.

The function

$$\text{len} : \text{BinNum} \rightarrow \mathbb{N}$$

maps a member of `BinNum` to its length. `len` is defined by the following equations using recursion and pattern matching:

$$\text{len}(\text{Nil}) = 0.$$

$$\text{len}(\text{Join}(u, b)) = \text{len}(u) + 1.$$

The function

$$\text{val} : \text{BinNum} \rightarrow \mathbb{N}$$

maps a member of `BinNum` to the value of the binary numeral it represents. For example,

$$\text{val}(\text{Join}(\text{Join}(\text{Join}(\text{Nil}, \text{One}), \text{Zero}), \text{One})) = (101)_2 = 5.$$

`val` is defined by the following equations using recursion and pattern matching:

$$\text{val}(\text{Nil}) = 0.$$

$$\text{val}(\text{Join}(u, \text{Zero})) = 2 * \text{val}(u).$$

$$\text{val}(\text{Join}(u, \text{One})) = (2 * \text{val}(u)) + 1.$$

The function

$$\text{add} : \text{BinNum} \times \text{BinNum} \rightarrow \text{BinNum}$$

is intended to implement addition on members of `BinNum`. It is defined by the following equations using recursion and pattern matching:

$$\text{add}(u, \text{Nil}) = u.$$

$$\text{add}(\text{Nil}, u) = u.$$

$$\text{add}(\text{Join}(u, \text{Zero}), \text{Join}(v, \text{Zero})) = \text{Join}(\text{add}(u, v), \text{Zero}).$$

$$\text{add}(\text{Join}(u, \text{One}), \text{Join}(v, \text{Zero})) = \text{Join}(\text{add}(u, v), \text{One}).$$

$$\text{add}(\text{Join}(u, \text{Zero}), \text{Join}(v, \text{One})) = \text{Join}(\text{add}(u, v), \text{One}).$$

$$\begin{aligned} \text{add}(\text{Join}(u, \text{One}), \text{Join}(v, \text{One})) = \\ \text{Join}(\text{add}(\text{add}(u, v), \text{Join}(\text{Nil}, \text{One})), \text{Zero}). \end{aligned}$$

Notice that the algorithm behind the definition is essentially the same algorithm that children learn to add numbers represented as decimal numerals. The last equation is a bit complicated because it involves a carry of 1.

**Lemma 1.** For all  $u, v \in \text{BinNum}$ ,

$$\text{len}(\text{add}(u, v)) \leq \text{len}(u) + \text{len}(v).$$

**Theorem 1.** For all  $u, v \in \text{BinNum}$ ,

$$\text{val}(\text{add}(u, v)) = \text{val}(u) + \text{val}(v).$$

Theorem 1 states that `add` correctly implements addition on the members of `BinNum`.

Prove Theorem 1 assuming Lemma 1. (You are not required to prove Lemma 1.) Hint: Use strong induction with  $P(n) \equiv \text{val}(\text{add}(u, v)) = \text{val}(u) + \text{val}(v)$  for all  $u, v \in \text{BinNum}$  such that  $n = \text{len}(u) + \text{len}(v)$ .

**Name:** Hishmat Salehi

**MacID:** salehh6

**Date:** February 2, 2020

## Problem 2

*Proof.* Assuming Lemma 1, let  $P(n) \equiv \text{val}(\text{add}(u, v)) = \text{val}(u) + \text{val}(v)$  for all  $u, v \in \text{BinNum}$  such that  $n = \text{len}(u) + \text{len}(v)$ . We will prove  $P(n)$  for all  $u, v \in \text{BinNum}$  by strong induction.

**Base case:**  $n = 0$ . We must show  $P(0)$ . Since  $n = 0$ , then  $0 = \text{len}(\text{Nil}) + \text{len}(\text{Nil})$  we have  $u = \text{Nil}$  and  $v = \text{Nil}$ .

$$\begin{aligned} & \text{val}(\text{add}(\text{Nil}, \text{Nil})) && \langle \text{LHS of } P(0) \rangle \\ = & \text{val}(\text{Nil}) && \langle \text{definition of } \text{add}(a, b) \text{ when } b = \text{Nil} \rangle \\ = & 0 && \langle \text{definition of } \text{val}(a) \text{ when } a = \text{Nil} \rangle \\ = & 0 + 0 && \langle \text{arithmetic; definition of } \text{val}(a) \text{ when } a = \text{Nil} \rangle \\ = & \text{val}(\text{Nil}) + \text{val}(\text{Nil}) && \langle \text{arithmetic; RHS of } P(0) \rangle \end{aligned}$$

So  $P(0)$  holds.

**Base case:**  $n = 1$ . We must show  $P(1)$ . Since  $n = 1$ , then  $1 = \text{len}(\text{Join}(\text{Nil}, \text{Bit})) + \text{len}(\text{Nil})$  we have  $u = \text{Join}(\text{Nil}, \text{Bit})$  and  $v = \text{Nil}$ .

$$\begin{aligned} & \text{val}(\text{add}(\text{Join}(\text{Nil}, \text{Bit}), \text{Nil})) && \langle \text{LHS of } P(1) \rangle \\ = & \text{val}(\text{Join}(\text{Nil}, \text{Bit})) && \langle \text{definition of } \text{add}(a, b) \text{ when } b = \text{Nil} \rangle \\ = & \text{val}(\text{Join}(\text{Nil}, \text{Bit})) + 0 && \langle \text{arithmetic} \rangle \\ = & \text{val}(\text{Join}(\text{Nil}, \text{Bit})) + \text{val}(\text{Nil}) && \langle \text{definition of } \text{val}(b), \text{ where } b = \text{Nil}; \text{RHS of } P(1) \rangle \end{aligned}$$

The same proof can be used to prove  $P(1)$  when  $v = \text{Join}(\text{Nil}, \text{Bit})$  and  $u = \text{Nil}$ , **without loss of generality**. So  $P(1)$  holds.

**Induction step:**  $n > 1$ . Assume  $P(2), P(3), \dots, P(n-2), P(n-1)$ . We must show  $P(n)$ .

$$\text{val}(\text{add}(u, \text{Nil})) = \text{val}(u).$$

$$\text{val}(\text{add}(\text{Nil}, u)) = \text{val}(u).$$

$$\text{val}(\text{add}(\text{Join}(u, \text{Zero}), \text{Join}(v, \text{Zero}))) = \text{val}(\text{Join}(\text{add}(u, v), \text{Zero})).$$

$$\text{val}(\text{add}(\text{Join}(u, \text{One}), \text{Join}(v, \text{Zero}))) = \text{val}(\text{Join}(\text{add}(u, v), \text{One})).$$

$$\text{val}(\text{add}(\text{Join}(u, \text{Zero}), \text{Join}(v, \text{One}))) = \text{val}(\text{Join}(\text{add}(u, v), \text{One})).$$

$$\text{val}(\text{add}(\text{Join}(u, \text{One}), \text{Join}(v, \text{One}))) = \text{val}(\text{Join}(\text{add}(\text{add}(u, v), \text{Join}(\text{Nil}, \text{One})), \text{Zero})).$$

With pattern matching recursively and using Inductive hypothesis  $P(0), P(1), \dots, P(n-2), P(n-1)$  we know that  $\text{len}(\text{add}(u, v)) \leq \text{len}(u) + \text{len}(v) = n$ , then  $P(n)$  takes 1 more step than  $P(n-1)$  therefore it is finite and  $P(n)$  holds.

Therefore,  $P(n)$  holds for all  $n \in \mathbb{N}$  by strong induction. □