Instructor: Fei He

TA: Jianhui Chen
Fengmin Zhu

Due date: 03/31/2019

Assignment 3

Zheng Zeng
2016013263

Instructions: Consult the included Dafny file `hw3.dfy` as you read the assignment. Fill in the required proofs in the Dafny file, and make sure that the verifier passes your code. To receive full credits, you must also include a careful manual proof (in this file) for each of the lemmas.

When finished, compile `hw3.tex` to a pdf. Compress your pdf and source code (i.e. `hw3.dfy`) to an `.zip` archive and hand it to *Tsinghua Web Learning* by the due date. Be sure to add your **student ID** and **full name** in the `stuid` and `stuname` macros at the top of `hw3.tex`.

Acamedic Honesty: Any kind of plagiarism is strictly prohibited in the full semester for this course. Students who are suspected to copy other's work and is confirmed through investigation will receive no credits (i.e, zero) for this assignment. If you asked other students for help, or your referred to any material that is not provided by us (e.g. websites, blogs, articles, papers, etc., both online and offline), please mention them in your assignment (e.g. writing an acknowledgment, adding a reference).

1 Peano Arithmetic

Recall from lecture that we discussed the Peano arithmetic, a first-order theory with signature:

$$\Sigma_{PA} : \{0, 1, +, \times, =\}$$

where,

- 0 and 1 are constants
- $+$ and $\times$ are binary functions
- $=$ is a binary predicate

T_{PA} has the following axioms:

- All of the equality axioms: reflexivity, symmetry, transitivity, and congruence
- **Zero:** $\forall x.\neg(x + 1 = 0)$
- **Additive identity:** $\forall x.x + 0 = x$
- **Times zero:** $\forall x.x \times 0 = 0$
- **Successor:** $\forall x, y.(x + 1 = y + 1) \rightarrow x = y$
- **Plus successor:** $\forall x, y.x + (y + 1) = (x + y) + 1$
- **Times successor:** $\forall x, y.x \times (y + 1) = x \times y + x$

It also has an axiom schema for induction:

$$(F[0] \wedge (\forall x.F[x] \rightarrow F[x + 1])) \rightarrow \forall x.F[x]$$

The intended interpretation for this theory is the natural numbers with constant symbols 0 and 1, a predicate symbol $=$ taking equality over $\mathbb{N}$, and function symbols $+, \times$ taking the corresponding expected functions over $\mathbb{N}$.

We can also define the natural numbers inductively, as follows:

$$\frac{}{0 \in \mathbb{N}} \qquad \frac{n \in \mathbb{N}}{succ(n) \in \mathbb{N}}$$

The corresponding inductive Dafny type is:

```
datatype Nat = Zero | Succ(n: Nat)
```

We can define the constant 1, as well as the functions for addition and multiplication as follows:

```
function one(): Nat
{
  Succ(Zero)
}

function add(x: Nat, y: Nat): Nat
{
  match(x) {
      case Zero => y
      case Succ(n) => Succ(add(n, y))
    }
}

function mult(x: Nat, y: Nat): Nat
{
  match(x) {
    case Zero => Zero
    case Succ(n) => add(mult(n, y), y)
  }
}
```

In this problem, you will use Dafny to prove that the inductively-defined `Nat` type satisfies the axioms listed above.

The Calc Statement. Before getting started, you should read section 21.17 of the Dafny manual to understand the basic usage of Calc statements, started with the keyword `calc`.

Instructions. Fill in the lemma preconditions, postconditions, and bodies marked in `hw3.dfy` to prove the axioms. There are eight lemmas in total, we only complete six of them for this part of the assignment. The lemma for additive identity is provided as an example of how to do proof-by-contradiction in Dafny, so you only have five axioms to do. The lemma for the induction axiom schema is marked as a bonus exercise.

To receive full credit for this assignment, you must provide the correct pre- and post-conditions for each of the six required lemmas, and the bodies of your lemmas satisfy each postcondition. If you use an `assume` statement in any of your lemmas, you will receive a fraction of the total credits.

Besides, you are also required to write a *careful* manual proof for each of them. By saying *careful*, we mean that you proof must include all details and explain the reasons for each step. A manual proof for **Additive identity** is already provided as an example. Remember in this assignment, the

only thing you know are the definitions of the functions listed in `hw3.dfy`, and a couple of "built-in" proof strategies (supported by Dafny and we admit them) including the (structural) induction, proof-by-contradiction and all equality axioms.

Proofs.

1. **Zero**

```
// Add an ensures annotation sufficient
// to prove the statement:
//   forall x . ~(x + 1 = 0)
// Then prove the lemma by providing a body
// sufficient to establish your ensures annotation.
// Note: your ensures annotation should make use
// of the argument passed to the lemma.
lemma {:induction false} axm_zero(x: Nat)
  ensures add(x, one()) != Zero
{
  if(add(x, one()) == Zero){
    match(x) {
      case Zero =>
        calc == {
          add(x, one());
          add(Zero, one());
          one(); !=
          Zero;
        }
      case Succ(n) =>
        calc == {
          add(x, one());
          Succ(add(n, one()));
            { add_lemmatwo(n); }
          Succ(Succ(n)); !=
          Zero;
        }
    }
  }
}
```

Before we prove the axiom, we prove this lemma first:
Lemma 1: $\forall x, x + 1 = succ(x)$

Proof. There are two cases to consider:

- x is zero, i.e. $x = 0$. By definition of $+$, we know that $x + 1 = 0 + 1 = 1$. By definition of 1, we have $x + 1 = 1 = succ(0) = succ(x)$.
- x is the successor of n. By definiton of $+$, we know that $x+1 = succ(n)+1 = succ(n+1)$. By inductive hypothesis, $n+1 = succ(n) = x$. Therefore, $x+1 = succ(n+1) = succ(x)$.

Therefore, $x + 1 = succ(x)$ holds for all x. □

Now we can Proof the original axiom:

Proof. By contradiction. Suppose that $x + 1 = 0$ does held for some x. Then, there are only

two possible choices:

- x is zero, i.e. $x = 0$. By definition of $+$, we know that $x+1 = 0+1 = 1$. Since $x+1 = 0$, we conclude that $1 = 0$, contradiction!
- x is the successor of n, i.e. $x = succ(n)$. By Lemma 1, we know that $x + 1 = succ(n) + 1 = succ(succ(n))$. Since $x + 1 = 0$, we conclude that 0 is the successor of some x, contradiction!

Therefore, $\neg(x + 1 = 0)$ holds for every x.
(This proof must based on the assumption(or axiom) that 0 is not the successor of any x.) □

2. **Additive identity** (We have done this as an example.)

```
// This lemma proves the statement:
//   forall x . x + 0 == x
// Note that it uses proof by contradiction,
// by assuming that its postcondition is false
// using an if statement, and deriving the
// negation of this assumption on all branches
// within the body.
lemma {:induction false} axm_pluszero(x: Nat)
  ensures add(x, Zero) == x
{
  if (add(x, Zero) != x) {
    match(x) {
      case Zero =>
        calc == {
          add(x, Zero);
          add(Zero, Zero);
          Zero;
          x;
        }
      case Succ(n) =>
        calc == {
          add(x, Zero);
          add(Succ(n), Zero);
          Succ(add(n, Zero));
            { axm_pluszero(n); } // inductive step
          Succ(n);
          x;
        }
    }
  }
}
```

Proof. By contradiction. Suppose that $x + 0 = x$ does not held for some x. Then, there are only two possible choices:

- x is zero, i.e. $x = 0$. By definition of $+$, we know that $x + 0 = 0 + 0 = 0$. Since $0 = x$, we conclude that $x + 0 = x$, contradiction!
- x is the successor of n, i.e. $x = succ(n)$. By definition of $+$, we know that $x + 0 = succ(n) + 0 = succ(n + 0)$. By inductive hypothesis, $n + 0 = n$. Thus, $succ(n + 0) = succ(n) = x$. We again conclude that $x + 0 = x$, contradiction!

Therefore, $x + 0 = x$ holds for every x. □

3. **Times zero**

```
// Add an ensures annotation sufficient
// to prove the statement:
//   forall x . x * 0 = 0
// Then prove the lemma by providing a body
// sufficient to establish your ensures annotation.
// Note: your ensures annotation should make use
// of the arguments passed to the lemma.
// Note: do not remove the {:induction false} attribute.
// Solutions that do not verify with this attribute
// will receive no credit!
lemma {:induction false} axm_timeszero(x: Nat)
  ensures mult(x, Zero) == Zero
{
  match(x) {
    case Zero => {
      //trivial
    }
    case Succ(n) => {
      calc == {
        mult(x, Zero);
        add(mult(n, Zero), Zero);
          { axm_pluszero(mult(n, Zero)); }
        mult(n, Zero);
          { axm_timeszero(n); }
        Zero;
      }
    }
  }
}
```

Proof. There are two cases to consider:

- x is zero, i.e. $x = 0$. By definition of $\times$, we know that $x \times 0 = 0 \times 0 = 0$.
- x is the successor of n. By definiton of $\times$, we know that $x \times 0 = (n \times 0) + 0$. By induction hypothesis, $n \times 0 = 0$. Therefore $x \times 0 = 0 + 0 = 0$.

Therefore, $x \times 0 = 0$ holds for all x. □

4. **Successor**

```
// Add an ensures annotation sufficient
// to prove the statement:
//   forall x, y . x + 1 = y + 1 ==> x = y
// Then prove the lemma by providing a body
// sufficient to establish your ensures annotation.
// Note: your ensures annotation should make use
// of the arguments passed to the lemma.
lemma axm_successor(x: Nat, y: Nat)
  requires add(x, one()) == add(y, one())
  ensures  x == y
{
  match(x) {
```

```
13      case Zero => {
14        match(y) {
15          case Zero => {
16
17          }
18          case Succ(n) => {
19            calc == {
20              add(y,one());
21              Succ(add(n, one()));
22                {add_lemmatwo(n);}
23              Succ(Succ(n)); !=
24              Succ(Zero);
25              Succ(x);
26                {add_lemmatwo(x);}
27              add(x, one());
28            }
29          }
30        }
31      }
32      case Succ(m) => {
33        match(y) {
34          case Zero => {
35            calc == {
36              add(x, one());
37              add(Succ(m), one());
38              Succ(add(m, one()));
39                { add_lemmatwo(m); }
40              Succ(Succ(m)); !=
41              Succ(Zero);
42              one();
43              add(y, one());
44            }
45          }
46          case Succ(n) => {
47            calc == {
48              add(m, one());
49              calc == {
50                Succ(add(m, one()));
51                add(Succ(m), one());
52                add(x, one());
53                add(y, one());
54                add(Succ(n), one());
55                Succ(add(n, one()));
56              }
57              add(n, one());
58            }
59            { axm_successor(m,n);}
60            calc == {
61              m;
62              n;
63            }
64            calc == {
65              x;
```

```
66                Succ(m);
67                Succ(n);
68                y;
69              }
70            }
71          }
72        }
73      }
74  }
```

Proof. There are two cases to consider for x:

- $x = 0$: By contradiction. Suppose that $y! = 0$, then there must be some n that $y = succ(n)$. so $y + 1 = succ(n + 1) = succ(succ(n))$, while $x + 1 = succ(x) = succ(0)$. Since$succ(succ(n))! = succ(0)$, we can conclude that $x + 1! = y + 1$, contradiction!
- $x = succ(n)$ for some n. Suppose that $y = 0$, then we can get contradiction similarly. Suppose that $y = succ(n)$, then $succ(m + 1) = succ(succ(m)) = succ(x) = x + 1$, $succ(n + 1) = y + 1$.Since $n + 1$ and $m + 1$ have equal successores, we can conclude that $m+1 = n+1$. By induction hypothesis, $m = n$. Therefore, $x = succ(m) = succ(n) = y$.

Therefore, $x + 1 = y + 1 \Rightarrow x = y$ holds for all x, y. □

5. **Plus successor**

```
// Add an ensures annotation sufficient
// to prove the statement:
//   forall x, y . x + (y + 1) = (x + y) + 1
// Then prove the lemma by providing a body
// sufficient to establish your ensures annotation.
// Note: your ensures annotation should make use
// of the arguments passed to the lemma.
lemma {:induction false} axm_plussuccessor(x: Nat, y: Nat)
  ensures add(x, add(y, one())) == add(add(x, y), one())
{
  match(x)
  {
    case Zero => {
      calc == {
        add(x, add(y, one()));
        add(Zero, add(y, one()));
        add(y, one());
        add(add(Zero, y), one());
        add(add(x, y), one());
      }
    }
    case Succ(n) => {
      calc == {
        add(x, add(y, one()));
        add(Succ(n), add(y, one()));
        Succ(add(n, add(y, one())));
          { axm_plussuccessor(n, y);}
        Succ(add(add(n, y), one()));
          { add_lemmatwo(add(n, y));}
        Succ(Succ(add(n, y)));
        Succ(add(Succ(n), y));
        Succ(add(x, y));
          { add_lemmatwo(add(x,y));}
        add(add(x, y), one());
      }
    }
  }
}
```

Proof. There are two cases to consider for x.

- x is zero, i.e. $x = 0$. By definition of $+$, $x + (y + 1) = 0 + (y + 1) = y + 1 = (0 + y) + 1$, The equation holds for zero.
- x is the successor of n, i.e. $x = succ(n)$. By definition of $+$, $x + (y + 1) = succ(n) + (y + 1) = succ(n + (y + 1))$. By inductive hypothesis, $n + (y + 1) = (n + y) + 1$. And we can conclude that $x + (y + 1) = succ((n + y) + 1) = succ(succ(n + y))$ with Lemma 1. So $x + (y + 1) = succ(succ(n + y)) = succ(succ(n) + y) = succ(x + y) = x + y + 1$. The equation holds for x.

Therefore, $x + (y + 1) = (x + y) + 1$ holds for all x. □

6. **Times successor**

```
// Prove the following lemma by providing a body
// sufficient to establish the ensures annotation.
// Hint: if you get stuck, you should consider
// making use of existing lemmas, or defining a
// new one.
// Note: do not change the provided postcondition,
// you will not receive credit if you do
lemma {:induction false} axm_timessuccessor(x: Nat, y: Nat)
  ensures mult(x, add(y, one())) == add(mult(x, y), x)
{
  match(x) {
    case Zero => {
      calc == {
        add(mult(Zero, y), Zero);
        add(Zero, Zero);
        Zero;
        mult(Zero, add(y, one()));
      }
    }
    case Succ(n) => {
      calc == {
        mult(Succ(n), add(y, one()));
        add(mult(n, add(y, one())), add(y, one()));
          { axm_timessuccessor(n, y); }
        add(add(mult(n,y),n), add(y, one()));
          { axm_plussuccessor(add(mult(n,y),n), y);}
        add(add(add(mult(n, y),n), y), one());
          { add_lemmathree(mult(n, y), n, y);}
        add(add(add(mult(n, y),y), n), one());
          { axm_plussuccessor(add(mult(n,y),y),n);}
        add(mult(Succ(n),y),add(n,one()));
          { add_lemmatwo(n);}
        add(mult(Succ(n),y),Succ(n));
      }
    }
  }
}
```

Still, we need to prove a new lemma first:
Lemma 2: $\forall x, y, z, x + (y + z) = (x + z) + y$

Proof. There are two cases to consider for z.

- if $z = 0$, $x + (y + z) = x + (y + 0) = x + y = (x + 0) + y = (x + z) + y$, the lemma holds.

- if $z = succ(n)$ for some n,

$$\begin{aligned}(x+y)+z &= (x+y)+succ(n)\\ &= (x+y)+(n+1) \text{ , by Lemma 1}\\ &= ((x+y)+n)+1\\ &= ((x+n)+y)+1 \text{ , by induction hypothesis}\\ &= succ((x+n)+y)\\ &= succ(x+n)+y\\ &= ((x+n)+1)+y \text{ , by Lemma 1}\\ &= (x+(n+1))+y\\ &= (x+succ(n))+y\\ &= (x+z)+y\end{aligned}$$

Therefore, $x+(y+z) = (x+z)+y$ holds for all x, y, z. □

Now we can prove the axiom:

Proof. There are two cases to consider for x.

- if $x = 0$, $x \times y + x = 0 \times y + 0 = 0 = 0 \times (y+1) = x \times (y+1)$.
- if $x = succ(n)$ for some n,

$$\begin{aligned}x \times (y+1) &= succ(n) \times (y+1)\\ &= n \times (y+1) + (y+1)\\ &= (n \times y + n) + (y+1) \text{ , by induction hypothesis}\\ &= ((n \times y + n) + y) + 1\\ &= ((n \times y + y) + n) + 1\\ &= ((succ(n) \times y) + n) + 1 \text{ , by definition of } \times\\ &= ((succ(n) \times y) + (n+1)\\ &= ((succ(n) \times y) + succ(n)\\ &= x \times y + x. \text{ , assumption.}\end{aligned}$$

Therefore, $x \times (y+1) = x \times y + x$ for all x, y. □