9.5

Software Analysis & Verification **Due: Mar 24, 2020**

Homework 2

Instructor: Fei He
TA: Jianhui Chen, Fengmin Zhu

Chenhao Li (2017011466)

Read the instructions below carefully before you start working on the assignment:

- In this assignment, you are asked to both typeset your answers in the attached LaTeX source file, and also complete the missing code in the Dafny file `PA.dfy`. Make sure that the Dafny complier can type check your code! When done, compile this file to a PDF. Compress the PDF and `PA.dfy` to an `.zip` archive and hand it to Tsinghua Web Learning *before the due date*.
- Make sure you fill in your *name* and *Tsinghua ID*, and replace all "`TODO`"s with your solutions.
- Any kind of dishonesty is *strictly prohibited* in the full semester. If you refer to any material that is not provided by us, you *must cite* it.

Problem 1: Short-Answered Questions

1-1 Underline all free variables (to be precise, their occurrences) in the following first-order formula:

$$\forall x.(f(x) \wedge (\exists y.g(x,y,z))) \wedge (\exists z.g(x,y,z))$$

Solution

$$\forall x.(f(x) \wedge (\exists y.g(x,y,\underline{z}))) \wedge (\exists z.g(\underline{x},\underline{y},z))$$

-0.5

■

1-2 Which of the following problems or theories are decidable?

(a) Deciding validity for propositional logic.
(b) Deciding validity for first-order logic.
(c) T_E.
(d) $T_{\mathbb{N}}$.
(e) The quantifier-free fragment of T_A.

Solution (a), (d), (e) ■

1-3 Find an equivalence relation that is not a congruence relation.

Solution Let S be the set of natural number pairs, $=_0$ be a binary relation on S, such that $a =_0 b$ iff $fst(a) = fst(b)$. It is easy to show that $=_0$ is an equivalence relation. However, $=_0$ is not a congruence relation: let $f(a) = fst(a) + snd(a)$, then $a =_0 b \not\Rightarrow f(a) = f(b)$ ■

1-4 Find two distinct equivalence relations such that one refines the other.

Solution Let $S = \{0,1\}$, $R_1 = \{(0,0),(1,1)\}$, $R_2 = \{(0,0),(0,1),(1,0),(1,1)\}$, then both R_1 and R_2 are equivalence relations and R_1 refines R_2. ■

1-5 In the congruence closure algorithm, subterms of a formula are represented by DAGs. Which term does Figure 1 represents? Write it out in a formulaic way.

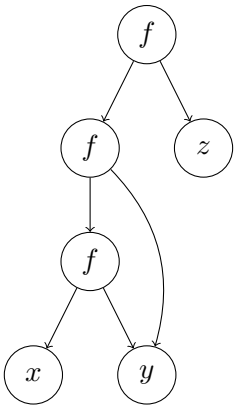

Figure 1: A subterm.

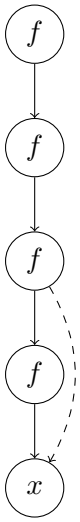

Figure 2: A DAG.

Solution The figure represents term $f(f(f(x,y),y),z)$. It also indicates $S_F = \{x, y, z, f(x,y), f(f(x,y),y), f(f(f(x,y),y),z)\}$ ■

1-6 Figure 2 presents a DAG in an execution of the congruence closure algorithm. The dashed edge was inserted via a union operation. Which congruences classes can you infer from this figure?

Solution

$$[x]_\sim = \{x, f(f(x)), f(f(f(f(x))))\}$$
$$[f(x)]_\sim = \{f(x), f(f(f(x)))\}$$

■

Problem 2: Peano Arithmetic

In this problem, we will show that two ways of defining natural numbers are, to some extent, the same – one is using the axioms of Peano Arithmetic, and another is using an inductive set whose elements are what we mean "natural numbers".

To receive full credit of this problem, you must both:

- complete the proofs in `PA.dfy`, and
- fill in the missing manual proofs in this file.

There is an example that has been done for you. You should read it carefully before you start.

PA Recall that *Peano Arithmetic* (PA) is 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

It 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:* $\forall x, y.\ (x + 1 = y + 1) \to x = y$
- *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:

$$(F[0] \land (\forall x.F[x] \to F[x + 1])) \to \forall x.F[x]$$

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 $\mathbb{N}$.

Natural Numbers as an Inductive Set On the other hand, we could define natural numbers as an *inductive* set S, i.e. a *minimum* set whose elements are generated by using only the following two rules:

- $0 \in S$;
- if $n \in S$, then $\mathsf{Succ}(n) \in S$.

In fact, our familiar $\mathbb{N} = S$ as defined above. Using the above notion, 1 is represented by $\mathsf{Succ}(0)$; 2 is represented by $\mathsf{Succ}(\mathsf{Succ}(0))$; and so on.

The above definition can be easily expressed in Dafny, as an inductive type:

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

We then 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 of PA.

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 In `PA.dfy`: Provide the correct pre- and post-conditions for lemmas *Zero* (2-1), *Times zero* (2-3), *Successor* (2-4), *Plus successor* (2-5) and *Time successor* (2-6), and the bodies of your lemmas must satisfy each postcondition. The proof for *Additive identity* (2-2) is provided as an example. If you use an `assume` statement in any of your lemmas, you will receive *partial* (i.e. not full) credits.

Moreover, in this file: Write a *careful manual* proof for each of the lemmas above. Again, the manual proof for *Additive identity* (2-2) is provided as an example. By saying *careful*, we mean that your proof must include all details and you should explain the reasons for each step. Remember that the only thing you know are: (1) the definitions of the functions listed in `PA.dfy`, and (2) 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

2-1 *Zero*

Proof. $x+1$ is the successor of x, while 0 is not the successor of any natural number, so $x+1 \neq 0$. □

2-2 *Additive identity* (example)

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. $x = n + 1$. By definition of $+$, we know that $x + 0 = (n + 1) + 0 = (n + 0) + 1$. By inductive hypothesis, $n + 0 = n$. Thus, $(n + 0) + 1 = n + 1 = x$. We again conclude that $x + 0 = x$, contradiction!

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

2-3 *Times zero*

Proof. There are only two possible choices:

- 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, i.e., $x = n + 1$. By definition of $\times$, $x \times 0 = (n + 1) \times 0 = n \times 0 + 0$. By axiom *Additive identity*, $n \times 0 + 0 = n \times 0$. By inductive hypothesis, $n \times 0 = 0$.

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

2-4 *Successor*

Proof. $x + 1$ is the successor of x, and $y + 1$ is the successor of y. Since every natural number has one unique successor and $x + 1 = y + 1$, we conclude that $x = y$. □

2-5 *Plus successor*

Let us first prove a stronger lemma *Associative law of addition*: $x + (y + z) = (x + y) + z$.

Proof. There are only two possible choices:

- x is zero, i.e., $x = 0$. By definition of $+$, $x + (y + z) = 0 + (y + z) = y + z$, and $(x + y) + z = (0 + y) + z = y + z$.
- x is the successor of n, i.e., $x = n + 1$.

  For the left hand term, by definition of $+$, $x + (y + z) = (n + 1) + (y + z) = (n + (y + z)) + 1$. By inductive hypothesis, $(n + (y + z)) + 1 = ((n + y) + z) + 1$.

  For the right hand term, by definition of $+$, $(x + y) + z = ((n + 1) + y) + z = ((n + y) + 1) + z = ((n + y) + z) + 1$.

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

Now let us get back to axiom *Plus successor*.

Proof. Simply apply lemma *Associative law of addition*, and let $z = 1$, we conclude that $x + (y + 1) = (x + y) + 1$. □

2-6 *Times successor*

Let us first prove a lemma *Commutative law of addition*: $x + y = y + x$.

Proof. There are only two possible choices:

- x is zero, i.e., $x = 0$. By definition of $+$, $x + y = 0 + y = y$. By axiom *Additive identity*, $y + x = y + 0 = y$.
- x is the successor of n, i.e., $x = n + 1$.

  For the left hand term, by definition of $+$, $x + y = (n + 1) + y = (n + y) + 1$. By inductive hypothesis, $(n + y) + 1 = (y + n) + 1$.

  For the right hand term, by lemma *Associative law of addition* $y + x = y + (n + 1) = (y + n) + 1$.

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

Now let us get back to axiom *Times successor*.

Proof. There are only two possible choices:

- x is zero, i.e., $x = 0$. By definition of $\times$, $x \times (y + 1) = 0 + \times(y + 1) = 0$. By definition of $+$ and $\times$, $(x \times y) + x = (0 \times y) + 0 = 0 + 0 = 0$.
- x is the successor of n, i.e., $x = n + 1$.

  For the left hand term, by definition of $\times$, $x \times (y+1) = (n+1) \times (y+1) = n \times (y+1) + (y+1)$. By inductive hypothesis, $n \times (y + 1) + (y + 1) = (n \times y + n) + (y + 1)$. By lemma *Associative law of addition*, $(n \times y + n) + (y + 1) = (n \times y) + (n + (y + 1))$. By lemma *Associative law of addition*, $(n \times y) + (n + (y + 1)) = (n \times y) + ((n + y) + 1)$. By lemma *Commutative law of addition*, $(n \times y) + ((n + y) + 1) = (n \times y) + ((y + n) + 1)$.

  For the right hand term, by definition of $\times$, $x \times y + x = (n+1) \times y + (n+1) = (n \times y + y) + (n+1)$. By lemma *Associative law of addition*, $(n \times y + y) + (n + 1) = (n \times y) + (y + (n + 1))$. By lemma *Associative law of addition*, $(n \times y) + (y + (n + 1)) = (n \times y) + ((y + n) + 1)$.

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