

# WHY IS $1 + 1 = 2$ ?

WUU YANG

ABSTRACT. Too often we take the *method* of decimal addition to be the *definition* of addition. We clarify this misunderstanding by first giving a definition of addition and then proving that the decimal addition method is both sound and complete with respect to the given definition. By showing the soundness and completeness of decimal addition with respect to the addition of natural numbers, we actually propose a new approach to algorithm correctness. We also discuss the differences between our logic approach and the traditional axiomatic approach.

## 1. INTRODUCTION

Too often we take the *method* of decimal addition to be the *definition* of addition. There are other kinds of addition methods, for instance, addition of binary numerals, addition of hexadecimal numerals, addition of Roman numerals, etc. All these are *methods* of addition, and they all implement the common concept of the *addition of natural numbers*.

In primary schools, students learn addition by means of such analogies as putting two piles of apples together. Then they learn decimal notation and the method of adding up two decimal numerals. Addition with decimal numerals is more convenient than addition with piles of apples, especially when thousands of apples are involved. After a little reflection, we should realize that adding up two decimal numerals is very different from putting two piles of apples together. This raises

---

*Date:* November 15, 2001.

*Key words and phrases.* addition, algorithm correctness, arithmetic, decimal addition, decimal notation, logic, foundation of mathematics.

The author wishes to express his sincere gratitude to the anonymous Reviewers for their very helpful comments. This work was supported, in part, by the National Science Council, Taiwan, Republic of China, under contracts NSC 89-2213-E-009-014, NSC 89-2213-E-009-068 and NSC 89-2213-E-009-146.

the question: Is the decimal addition method that we use almost every day correct? Does the decimal addition method always calculate the same sum as would be obtained if we did addition using apples?

Since counting apples is a more primitive notion than adding up decimal numerals, we will take putting-apples-together as the *definition* of the addition of natural numbers. There are several ways to put two piles of apples together. First, we may move the apples from the second pile to the first pile, one at a time. We may also take one apple from the second pile, then put the remaining apples in a single pile (recursively) and finally put back the apple that was taken out earlier. We will take the second method as the definition of addition in Section 3. (The two methods are equivalent. The second method was chosen to ease the task of proving properties.)

In Section 4, we will carefully present the rules of decimal addition, which comprise the *DECL* language. In defining *DECL*, we face an interesting question: What is the official definition of decimal addition? Surprisingly, students achieve only an intuitive understanding of decimal addition in elementary schools and use it for many years without learning a rigorous definition. On the other hand, students really do not care whether the *DECL* language exactly captures decimal addition. The *DECL* language might allow other models as well as the decimal addition. However, our study will show that every model of *DECL*, in particular, decimal addition, is *correct* with respect to the definition of addition.

Decimal addition is correct with respect to the definition of the addition of natural numbers in the sense that (1) the decimal addition method computes the same sum as would be obtained by counting apples and (2) all results that can be obtained by counting apples can also be obtained using decimal addition. These properties will be discussed in Section 5.

The main idea behind this paper is that decimal numerals are representations of natural numbers; other representations also exist, such as binary numerals, Roman numerals, or the Chinese abacus. In order to use a representation such as a decimal numeral, it is necessary to show that operations on the representation exactly reflect

operations on natural numbers. Here, *exactly* is manifested by the soundness and completeness properties.

This paper proposes a new approach to proving algorithm correctness. Decimal numerals and decimal addition may be viewed as a data type. From this viewpoint, our approach resembles Guttag and Horning's [1]. In our approach, a program (i.e., decimal addition) is specified as a first-order language, and its correctness criteria is specified in terms of another first-order language. Though we also make use of axioms and inference rules, our proof style is quite different from the traditional axiomatic approach [2, 3], in which logic serves as the inference mechanism. In contrast, a program is abstracted as logic axioms and inference rules in our approach. Furthermore, in our approach, the semantic domain is also modeled as a first-order language.

The last section concludes this paper by stating the significance of this study.

## 2. A FIRST-ORDER LANGUAGE WITH EQUALITY

Our discussion in this paper is based on a first-order language with equality. In this section, we will give a definition of such a language. Our notation is primarily based on [4].

**Definition.** A *first-order language with equality* consists of the following ingredients: zero or more constant symbols, zero or more variable symbols, zero or more function symbols, a binary predicate symbol  $=$ , the logical connectives  $\neg$  and  $\rightarrow$ , the quantifier  $\forall$ , and the punctuation symbols  $,$ ,  $($  and  $)$ . We sometimes use square brackets  $[$  and  $]$  instead of parentheses just to clarify the pairing of parentheses. A *term* is made up of constants, variables, and functions. A primitive *well-formed formula* (abbreviated as *formula*) has the form  $\alpha = \beta$ , where  $\alpha$  and  $\beta$  are two terms. More complicated formulae are made up of primitive formulae, logical connectives and the quantifier. The language includes the following axioms and inference rules: Let  $A$ ,  $B$ , and  $C$  be any well-formed formulae, and let  $f$  be any function symbol.

Axiom E1.  $A \rightarrow (B \rightarrow A)$ .

Axiom E2.  $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$ .

Axiom E3.  $(\neg A \rightarrow \neg B) \rightarrow (B \rightarrow A)$ .

Axiom E4.  $\forall(x)A(x) \rightarrow A(t)$ , where  $t$  is free for  $x$  in  $A$ .

Axiom E5.  $\forall(x)(A \rightarrow B) \rightarrow (A \rightarrow \forall(x)B)$ , where  $A$  contains no free occurrences of  $x$ .

Axiom E6.  $\forall(x)(x = x)$ .

Axiom E7.  $\forall(x)\forall(y)(x = y \rightarrow (A(x, x) \rightarrow A(x, y)))$ .

Axiom E8.  $\forall(x)\forall(y)(x = y \rightarrow f(x) = f(y))$ .

Inference rule I1. (*modus ponens*) From  $A$  and  $A \rightarrow B$ ,  $B$  may be derived, where  $A$  and  $B$  are any well-formed formulae.

Inference rule I2. (*generalization*) From  $A$ ,  $\forall(x)A$  may be derived, where  $A$  is any well-formed formula, and  $x$  is any variable.

The first five axioms and the two inference rules are the usual axioms and rules for a pure first-order language. The last three axioms define the equality predicate.

Note that the above definition is a generic definition for a first-order language with equality. Specific languages, such as *APPLE* and *DECL* (to be defined later in this paper), will include new constant symbols, function symbols or predicate symbols and new axioms or inference rules.

The semantics of addition will be defined using a first-order language, called *APPLE* (in Section 3), that includes the  $=$  predicate, a unary *successor* function, and a binary  $+$  function. The decimal addition algorithm will be defined by another first-order language, called *DECL* (in Section 4), that includes the  $=$  predicate and a decimal addition function  $\oplus$ . We will show that  $\oplus$  in *DECL* is *correct* with respect to the  $+$  function in *APPLE*.

We will need a few basic results for all first-order languages with equality.

**Theorem** (Transitivity Theorem). From  $A \rightarrow B$  and  $B \rightarrow C$ , we may conclude that  $A \rightarrow C$ .

The proof of the transitivity theorem can be found in most logic textbooks, such as [4, 5]. We will not include the proof here.

**Lemma 2.1.**  $\forall(x)\forall(y)(x = y \rightarrow y = x)$ .

**Proof.** Let  $A(\alpha, \beta)$  in Axiom E7 be  $\beta = \alpha$ .

1.  $\forall(x)\forall(y)(x = y \rightarrow (x = x \rightarrow y = x))$ , Axiom E7.
2.  $x = y \rightarrow (x = x \rightarrow y = x)$ , 1 and Axiom E4.
3.  $(x = y \rightarrow (x = x \rightarrow y = x)) \rightarrow ((x = y \rightarrow x = x) \rightarrow (x = y \rightarrow y = x))$ ,

Axiom E2.

4.  $(x = y \rightarrow x = x) \rightarrow (x = y \rightarrow y = x)$ , 2 and 3 by *modus ponens*.
5.  $x = x \rightarrow (x = y \rightarrow x = x)$ , Axiom E1.
6.  $\forall(x)(x = x)$ , Axiom E6.
7.  $x = x$ , 6 and Axiom E4.
8.  $x = y \rightarrow x = x$ , 5 and 7 by *modus ponens*.
9.  $x = y \rightarrow y = x$ , 4 and 8 by *modus ponens*.
10.  $\forall(x)\forall(y)(x = y \rightarrow y = x)$ , 9 by generalization.

**qed**

**Lemma 2.2.**  $\forall(x)\forall(y)\forall(z)(x = y \rightarrow (y = z \rightarrow x = z))$ .

**Proof.** Let  $A(\alpha, \beta)$  in Axiom E7 be  $\beta = z$ .

1.  $\forall(x)\forall(y)(x = y \rightarrow (x = z \rightarrow y = z))$ , Axiom E7.
2.  $x = y \rightarrow (x = z \rightarrow y = z)$ , 1 and Axiom E4.
3.  $\forall(y)\forall(x)(y = x \rightarrow x = y)$ , Lemma 2.1.
4.  $y = x \rightarrow x = y$ , 3 and Axiom E4.
5.  $y = x \rightarrow (x = z \rightarrow y = z)$ , 2 and 4 by the Transitivity Theorem.
6.  $\forall(y)\forall(x)\forall(z)(y = x \rightarrow (x = z \rightarrow y = z))$ , 5 by generalization.
7.  $\forall(x)\forall(y)\forall(z)(x = y \rightarrow (y = z \rightarrow x = z))$ , rename the variables.

**qed**

**Lemma 2.3.** Let  $P$  be a formula. We have  $\forall(x)\forall(y)\forall(z)[y = z \rightarrow ((P \rightarrow x = y) \rightarrow (P \rightarrow x = z))]$ .

**Proof.** Let  $A(\alpha, \beta)$  be the formula  $P \rightarrow x = \beta$ .

1.  $\forall(y)\forall(z)[y = z \rightarrow (A(y, y) \rightarrow A(y, z))]$ , Axiom E7.

2.  $y = z \rightarrow (A(y, y) \rightarrow A(y, z))$ , 1 and Axiom E4. This formula can be written as  $y = z \rightarrow ((P \rightarrow x = y) \rightarrow (P \rightarrow x = z))$ .

3.  $\forall(x)\forall(y)\forall(z)[y = z \rightarrow ((P \rightarrow x = y) \rightarrow (P \rightarrow x = z))]$ , 2 by generalization.

**qed**

**Lemma 2.4.** Let  $P$  be a formula. We have  $P \rightarrow P$ .

**Proof.**

1.  $P \rightarrow (x = x \rightarrow P)$ , Axiom E1.
2.  $[P \rightarrow (x = x \rightarrow P)] \rightarrow [(P \rightarrow x = x) \rightarrow (P \rightarrow p)]$ , Axiom E2.
3.  $(P \rightarrow x = x) \rightarrow (P \rightarrow P)$ , 1 and 2 by *modus ponens*.
4.  $x = x$ , Axiom E6.
5.  $x = x \rightarrow (P \rightarrow x = x)$ , Axiom E1.
6.  $P \rightarrow x = x$ , 4 and 5 by *modus ponens*.
7.  $P \rightarrow P$ , 3 and 6 by *modus ponens*.

**qed**

### 3. THE *APPLE* LANGUAGE

The most intuitive definition of addition should be based on our common experience. In this section, we will base our definition of the addition of natural numbers on piles of apples. When we attempt to put two piles of apples together, we simply put all the apples in a single pile, which should be the *sum* of the two piles. One way to put two piles into a single pile is to take one apple out from the second pile, add the remaining apples in the second pile to the first pile, and finally add the apple that was taken out earlier to the first pile. Finally, the first pile is considered the *sum* of the two piles.

The *APPLE* language defined below is adapted from the well-known Peano arithmetic system [5]. Axioms related to multiplication are omitted.

**Definition.** The language *APPLE* is a first-order language with equality that includes a constant symbol  $\emptyset$  (which is intended to stand for the constant 0), a unary function symbol  $s$  (which is intended to stand for the *successor* function)

and a binary function symbol  $+$  (which is intended to stand for generic addition). Axiom E8 should be stated as the following axiom in *APPLE*:

$$\text{Axiom A1. } \forall(x)\forall(y)(x = y \rightarrow s(x) = s(y)).$$

We may think of a term  $x$  in *APPLE* as a pile of apples. When the successor function is applied to the term, the result  $s(x)$  can be viewed as *adding one apple to the pile*  $x$ . Under this interpretation, the addition function can be defined by the following two axioms:

$$\text{Axiom A2. } \forall(x)(x + \emptyset = x).$$

$$\text{Axiom A3. } \forall(x)\forall(y)(x + s(y) = s(x + y)).$$

We need Axiom 4 to prevent a trivial interpretation. Axiom 5 is the usual mathematical induction. Note that induction, though very helpful in many proofs, is a serious constraint on the set of natural numbers. The induction axiom essentially says that all natural numbers can be represented as  $\emptyset$  or  $s(x)$ , where  $x$  is a natural number. No other kinds of natural numbers exist.

$$\text{Axiom A4. } \forall(x)(\neg(s(x) = \emptyset)).$$

$$\text{Axiom A5. } A(\emptyset) \rightarrow [\forall(x)(A(x) \rightarrow A(s(x))) \rightarrow \forall(x)A(x)], \text{ where } A$$

is a formula.

**Definition.** Let  $F$  be a well-formed formula in *APPLE*. We say that  $\vdash_A F$  if and only if  $F$  is provable from Axioms A1 through A5, E1 through E7 and the two inference rules I1 and I2.

Based on the above axioms, we can prove the following lemmas concerning *APPLE*.

$$\text{Lemma 3.1. } \vdash_A \forall(y)(\emptyset + y = y).$$

**Proof.** Let  $A(y)$  be the formula  $\emptyset + y = y$ . Then  $A(\emptyset)$  is  $\emptyset + \emptyset = \emptyset$ .

1.  $\emptyset + \emptyset = \emptyset$ , Axiom A2 (let  $x$  be  $\emptyset$ ).
2.  $A(\emptyset) \rightarrow (\forall(x)(A(x) \rightarrow A(s(x))) \rightarrow \forall(x)(\emptyset + x = x))$ , Axiom A5.
3.  $\forall(x)(A(x) \rightarrow A(s(x))) \rightarrow \forall(x)(\emptyset + x = x)$ , 1 and 2 by *modus ponens*.
4.  $\emptyset + s(y) = s(\emptyset + y)$ , Axiom A3 (let  $x$  be  $\emptyset$ ).

5.  $\emptyset + y = y \rightarrow (s(\emptyset + y) = s(\emptyset + y) \rightarrow s(\emptyset + y) = s(y))$ , Axiom E7.
6.  $[\emptyset + y = y \rightarrow (s(\emptyset + y) = s(\emptyset + y) \rightarrow s(\emptyset + y) = s(y))] \rightarrow [(\emptyset + y = y \rightarrow s(\emptyset + y) = s(\emptyset + y)) \rightarrow (\emptyset + y = y \rightarrow s(\emptyset + y) = s(y))]$ , Axiom E2.
7.  $(\emptyset + y = y \rightarrow s(\emptyset + y) = s(\emptyset + y)) \rightarrow (\emptyset + y = y \rightarrow s(\emptyset + y) = s(y))$ , 5 and 6 by *modus ponens*.
8.  $s(\emptyset + y) = s(\emptyset + y) \rightarrow (\emptyset + y = y \rightarrow s(\emptyset + y) = s(\emptyset + y))$ , Axiom E1.
9.  $s(\emptyset + y) = s(\emptyset + y)$ , Axiom E6.
10.  $\emptyset + y = y \rightarrow s(\emptyset + y) = s(\emptyset + y)$ , 8 and 9 by *modus ponens*.
11.  $\emptyset + y = y \rightarrow s(\emptyset + y) = s(y)$ , 7 and 10 by *modus ponens*.
12.  $\emptyset + s(y) = s(\emptyset + y) \rightarrow (s(\emptyset + y) = s(y) \rightarrow \emptyset + s(y) = s(y))$ , Lemma 2.2.
13.  $s(\emptyset + y) = s(y) \rightarrow \emptyset + s(y) = s(y)$ , 4 and 12 by *modus ponens*.
14.  $\emptyset + y = y \rightarrow \emptyset + s(y) = s(y)$ , 11 and 13 by the transitive theorem.
15.  $\forall(y)(\emptyset + y = y \rightarrow \emptyset + s(y) = s(y))$ , 14 by generalization. This formula can be written as  $\forall(x)(A(x) \rightarrow A(s(x)))$ .
16.  $\forall(x)(\emptyset + x = x)$ , 3 and 15 by *modus ponens*.

**qed**

**Lemma 3.2.**  $\vdash_A \forall(y)(s(x) + y = s(x + y))$ .

**Proof.** Let  $A(\alpha)$  be  $s(x) + \alpha = s(x + \alpha)$ .

1.  $A(\emptyset) \rightarrow (\forall(y)(A(y) \rightarrow A(s(y))) \rightarrow \forall(y)A(y))$ , Axiom A5.
2.  $s(x) + \emptyset = s(x)$ , Axiom A2.
3.  $x + \emptyset = x$ , Axiom A2.
4.  $x + \emptyset = x \rightarrow s(x + \emptyset) = s(x)$ , Axiom A1.
5.  $s(x + \emptyset) = s(x)$ , 3 and 4 by *modus ponens*.
6.  $s(x + \emptyset) = s(x) \rightarrow s(x) = s(x + \emptyset)$ , Lemma 2.1.
7.  $s(x) = s(x + \emptyset)$ , 5 and 6 by *modus ponens*.
8.  $s(x) + \emptyset = s(x) \rightarrow (s(x) = s(x + \emptyset) \rightarrow s(x) + \emptyset = s(x + \emptyset))$ , Lemma 2.2.
9.  $s(x) + \emptyset = s(x + \emptyset)$ , 2, 7, and 8 by *modus ponens*. This formula can be written as  $A(\emptyset)$ .

10.  $\forall(y)(A(y) \rightarrow A(s(y))) \rightarrow \forall(y)A(y)$ , 1 and 9 by *modus ponens*.
11.  $x + s(y) = s(x + y)$ , Axiom A3.
12.  $x + s(y) = s(x + y) \rightarrow s(x + s(y)) = s(s(x + y))$ , Axiom A1.
13.  $s(x + s(y)) = s(s(x + y))$ , 11 and 12 by *modus ponens*.
14.  $s(x) + s(y) = s(s(x) + y)$ , Axiom A3.
15.  $s(x) + y = s(x + y) \rightarrow s(s(x) + y) = s(s(x + y))$ , Axiom A1.
16.  $s(x) + y = s(x + y) \rightarrow s(x) + s(y) = s(s(x + y))$ , 14 and 15 by Lemma 2.4.
17.  $s(x) + y = s(x + y) \rightarrow s(x) + s(y) = s(x + s(y))$ , 13 and 16 by Lemma 2.4.

This formula can be written as  $A(y) \rightarrow A(s(y))$ .

18.  $\forall(y)(A(y) \rightarrow A(s(y)))$ . 17 by generalization.
19.  $\forall(y)A(y)$ , 10 and 18 by *modus ponens*.

**qed**

**Lemma 3.3.**  $\vdash_A \forall(x)\forall(y)(s(x) + y = s(x + y))$ .

**Proof.** Lemma 3.3 can be obtained directly from Lemma 3.2 by generalization.

**qed**

**Lemma 3.4.**  $\vdash_A \forall(x)\forall(y)(x + s(y) = s(x) + y)$ .

**Proof.** Lemma 3.4 can be obtained directly from Axiom A3 and Lemma 3.3.

**qed**

**Lemma 3.5.**  $+$  is commutative, that is,  $\vdash_A \forall(x)\forall(y)(x + y = y + x)$ .

**Proof.** Let  $A(x)$  be the formula  $\forall(y)(x + y = y + x)$ .

1.  $\emptyset + y = y$ , Lemma 3.1.
2.  $y + \emptyset = y$ , Axiom A2.
3.  $y = y + \emptyset$ , 2 by Lemma 2.1.
4.  $\emptyset + y = y \rightarrow (y = y + \emptyset \rightarrow \emptyset + y = y + \emptyset)$ , Lemma 2.2.
5.  $\emptyset + y = y + \emptyset$ , 1, 3, and 4 by *modus ponens*.
6.  $\forall(y)(\emptyset + y = y + \emptyset)$ , 5 by generalization. This formula can be written as  $A(\emptyset)$ .
7.  $A(\emptyset) \rightarrow (\forall(x)(A(x) \rightarrow A(s(x))) \rightarrow \forall(x)A(x))$ , Axiom A5.
8.  $\forall(x)(A(x) \rightarrow A(s(x))) \rightarrow \forall(x)A(x)$ , 6 and 7 by *modus ponens*.

We will need to prove  $\forall(x)(A(x) \rightarrow A(s(x)))$ , which can be written as  $\forall(x)(\forall(y)(x + y = y + x) \rightarrow \forall(y)(s(x) + y = y + s(x)))$ . This formula is obtained in line 20 below.

9.  $s(x) + y = s(x + y)$ , Lemma 3.3.

10.  $y + s(x) = s(y + x)$ , Axiom A4.

11.  $s(y + x) = y + s(x)$ , 10 by Lemma 2.1.

12.  $x + y = y + x \rightarrow s(x + y) = s(y + x)$ , Axiom A1.

13.  $x + y = y + x \rightarrow s(x + y) = y + s(x)$ , 11 and 12 by Lemma 2.3.

14.  $x + y = y + x \rightarrow s(x) + y = y + s(x)$ , 9 and 13 by Lemma 2.3.

15.  $\forall(y)(x + y = y + x) \rightarrow x + y = y + x$ , Axiom E4.

16.  $\forall(y)(x + y = y + x) \rightarrow s(x) + y = y + s(x)$ , 14 and 15 by the transitivity theorem.

17.  $\forall(y)[\forall(y)(x + y = y + x) \rightarrow s(x) + y = y + s(x)]$ , 16 by generalization.

18.  $\forall(y)[\forall(y)(x + y = y + x) \rightarrow s(x) + y = y + s(x)] \rightarrow [\forall(y)(x + y = y + x) \rightarrow \forall(y)(s(x) + y = y + s(x))]$ , Axiom E5.

19.  $\forall(y)(x + y = y + x) \rightarrow \forall(y)(s(x) + y = y + s(x))$ , 17 and 18 by *modus ponens*.

20.  $\forall(x)(\forall(y)(x + y = y + x) \rightarrow \forall(y)(s(x) + y = y + s(x)))$ , 19 by generalization.

This formula can be written as  $\forall(x)(A(x) \rightarrow A(s(x)))$ .

21.  $\forall(x)A(x)$ , 8 and 20 by *modus ponens*. This formula can be written as  $\forall(x)\forall(y)(x + y = y + x)$ .

**qed**

**Lemma 3.6.**  $\vdash_A \forall(x)\forall(y)\forall(z)(x = y \rightarrow x + z = y + z)$ .

**Proof.** Let  $A(z)$  be the formula  $x = y \rightarrow x + z = y + z$ . We will use the induction axiom to prove that  $\forall(z)[x = y \rightarrow x + z = y + z]$  (at line 16 below).

1.  $x + \emptyset = x$ , Axiom A2.

2.  $y + \emptyset = y$ , Axiom A2.

3.  $x = y \rightarrow x = y$ , Lemma 2.4.

4.  $x = y \rightarrow x + \emptyset = y + \emptyset$ , 1, 2, and 3 by Lemmas 2.3 and 3.5. This formula can be written as  $A(\emptyset)$ .

5.  $A(\emptyset) \rightarrow (\forall(z)(A(z) \rightarrow A(s(z)))) \rightarrow \forall(z)A(z)$ , Axiom A5.

6.  $\forall(z)(A(z) \rightarrow A(s(z))) \rightarrow \forall(z)A(z)$ , 4 and 5 by *modus ponens*.

Next we will prove that  $\forall(z)(A(z) \rightarrow A(s(z)))$  (in line 15 below).

7.  $x + s(z) = s(x + z)$ , Axiom A3.

8.  $y + s(z) = s(y + z)$ , Axiom A3.

9.  $x + z = y + z \rightarrow s(x + z) = s(y + z)$ , Axiom A1.

10.  $x + z = y + z \rightarrow x + s(z) = y + s(z)$ , 7, 8, and 9 by Lemma 2.3. We will use  $P$  to represent this formula in this proof.

11.  $P \rightarrow (x = y \rightarrow P)$ , Axiom E1.

12.  $x = y \rightarrow P$ , 10 and 11 by *modus ponens*. This formula can be written as  $x = y \rightarrow (x + z = y + z \rightarrow x + s(z) = y + s(z))$ .

13.  $[x = y \rightarrow (x + z = y + z \rightarrow x + s(z) = y + s(z))] \rightarrow [(x = y \rightarrow x + z = y + z) \rightarrow (x = y \rightarrow x + s(z) = y + s(z))]$ , Axiom E2.

14.  $(x = y \rightarrow x + z = y + z) \rightarrow (x = y \rightarrow x + s(z) = y + s(z))$ , 12 and 13 by *modus ponens*. This formula can be written as  $A(z) \rightarrow A(s(z))$ .

15.  $\forall(z)(A(z) \rightarrow A(s(z)))$ , 14 by generalization.

16.  $\forall(z)A(z)$ , 5 and 15 by *modus ponens*.

17.  $\forall(x)\forall(y)\forall(z)(x = y \rightarrow x + z = y + z)$ , 16 by generalization.

**qed**

**Lemma 3.7.**  $\vdash_A \forall(x)\forall(y)\forall(z)(x = y \rightarrow z + x = z + y)$ .

**Proof.** By Lemmas 2.3, 3.5 and 3.6. **qed**

**Lemma 3.8.**  $+$  is associative, that is,  $\vdash_A \forall(x)\forall(y)\forall(z)((x + y) + z = x + (y + z))$ .

**Proof.** Let  $A(z)$  be the formula  $(x + y) + z = x + (y + z)$ . We will use the induction axiom to prove that  $\forall(z)A(z)$  (in line 22 below). Then we will use the generalization rule to prove the lemma.

1.  $A(\emptyset) \rightarrow (\forall(z)(A(z) \rightarrow A(s(z)))) \rightarrow \forall(z)A(z)$ , Axiom A5.

2.  $(x + y) + \emptyset = x + y$ , Axiom A2.

3.  $y + \emptyset = y$ , Axiom A2.

4.  $y + \emptyset = y \rightarrow x + (y + \emptyset) = x + y$ , Lemma 3.7.
5.  $x + (y + \emptyset) = x + y$ , 3 and 4 by *modus ponens*.
6.  $x + (y + \emptyset) = x + y \rightarrow x + y = x + (y + \emptyset)$ , Lemma 2.1.
7.  $x + y = x + (y + \emptyset)$ , 5 and 6 by *modus ponens*.
8.  $(x + y) + \emptyset = x + y \rightarrow (x + y = x + (y + \emptyset) \rightarrow (x + y) + \emptyset = x + (y + \emptyset))$ ,

Lemma 2.2.

9.  $(x + y) + \emptyset = x + (y + \emptyset)$ , 2, 7, and 8 by *modus ponens*. This formula can be written as  $A(\emptyset)$ .

10.  $\forall(z)(A(z) \rightarrow A(s(z))) \rightarrow \forall(z)A(z)$ , 1 and 9 by *modus ponens*.

We will next prove that  $\forall(z)(A(z) \rightarrow A(s(z)))$  (in line 21 below).

11.  $(x + y) + s(z) = s((x + y) + z)$ , Axiom A3.
12.  $y + s(z) = s(y + z)$ , Axiom A3.
13.  $y + s(z) = s(y + z) \rightarrow x + (y + s(z)) = x + s(y + z)$ , Lemma 3.7.
14.  $x + (y + s(z)) = x + s(y + z)$ , 12 and 13 by *modus ponens*.
15.  $x + s(y + z) = s(x + (y + z))$ , Axiom A3.
16.  $x + (y + s(z)) = x + s(y + z) \rightarrow (x + s(y + z) = s(x + (y + z)) \rightarrow x + (y + s(z)) = s(x + (y + z)))$ , Lemma 2.2.
17.  $x + (y + s(z)) = s(x + (y + z))$ , 14, 15 and 16 by *modus ponens*.
18.  $(x + y) + z = x + (y + z) \rightarrow s((x + y) + z) = s(x + (y + z))$ , Axiom A1.
19.  $(x + y) + z = x + (y + z) \rightarrow (x + y) + s(z) = s(x + (y + z))$ , 11 and 18 by

Lemma 2.3.

20.  $(x + y) + z = x + (y + z) \rightarrow (x + y) + s(z) = x + (y + s(z))$ , 17 and 19 by Lemma 2.3. This formula can be written as  $A(z) \rightarrow A(s(z))$ .

21.  $\forall(z)[A(z) \rightarrow A(s(z))]$ , 20 by generalization.
22.  $\forall(z)A(z)$ , 10 and 21 by *modus ponens*.
23.  $\forall(x)\forall(y)\forall(z)((x + y) + z = x + (y + z))$ , 22 by generalization.

**qed**

Note that we have adopted the conventional notation in order to omit certain parentheses. For instance,  $x + y = y + x$  is an abbreviation for  $((x + y) = (y + x))$ .

#### 4. THE *DECL* LANGUAGE

Now we want to define the language *DECL* for decimal addition. This language is a first-order language with equality. First we need to include the ten constants  $\bar{0}, \bar{1}, \bar{2}, \bar{3}, \bar{4}, \bar{5}, \bar{6}, \bar{7}, \bar{8}, \bar{9}$ , which represent the ten decimal digits used in the decimal addition.

A decimal numeral, such as 538, is considered an abbreviation of the term  $((\bar{5} \triangleright \bar{3}) \triangleright \bar{8})$  in the *DECL* language, where  $\triangleright$  is a new binary function symbol. Intuitively, a term such as  $\bar{5} \triangleright \bar{3}$  is intended to denote the natural number  $5 \times 10 + 3$ . The function symbol  $\triangleright$  is the basis of the decimal notation.

**Definition.** A *decimal numeral* is either one of the ten decimal digits or has the form  $(\alpha \triangleright d)$ , where  $\alpha$  is a (simpler) decimal numeral and  $d$  is a decimal digit.

Obviously, the set of all decimal numerals is a proper subset of the set of all the terms in *DECL*. For instance, the term  $\bar{6} \triangleright (\bar{3} \triangleright \bar{5})$  is not a decimal numeral, though it is equivalent to one. Though we are only concerned with the arithmetic of decimal numerals, it is more convenient to consider all the terms of *DECL*.

In decimal notation, leading zeros are useless and may be omitted. We will use the following axiom to erase leading zeros:

$$\text{Axiom D1. } \forall(\alpha)((\bar{0} \triangleright \alpha) = \alpha).$$

We will use the function symbol  $\oplus$  to denote traditional addition of decimal numerals. Note that  $\oplus$  is not the same as  $+$  used in *APPLE* in that  $\oplus$  is applied to terms of *DECL* whereas  $+$  is applied to terms of *APPLE*.

To add two decimal numerals such as 538 and 217, the traditional algorithm essentially computes  $((\bar{5} \triangleright \bar{3}) \triangleright \bar{8}) \oplus ((\bar{2} \triangleright \bar{1}) \triangleright \bar{7})$ , where the new function symbol  $\oplus$  is defined by the following axiom. Let  $\alpha, \beta, \gamma,$  and  $\delta$  denote arbitrary terms of *DECL*:

$$\text{Axiom D2. } \forall(\alpha)\forall(\beta)\forall(\gamma)\forall(\delta)((\alpha \triangleright \beta) \oplus (\gamma \triangleright \delta) = (\alpha \oplus \gamma) \triangleright (\beta \oplus \delta)).$$

Because adding two large digits, such as  $\bar{7}$  and  $\bar{8}$ , may result in a carry, we need one axiom to process the carry:

$$\text{Axiom D3. } \forall(\alpha)\forall(\beta)\forall(\gamma)(\alpha \triangleright (\beta \triangleright \gamma) = (\alpha \oplus \beta) \triangleright \gamma).$$

We also need to define how to add two decimal digits. We need the following axiom, which is the addition table of decimal digits:

$$\begin{aligned} \text{Axiom D4. } & \bar{0} \oplus \bar{0} = \bar{0}. \bar{0} \oplus \bar{1} = \bar{1}. \bar{0} \oplus \bar{2} = \bar{2}. \bar{0} \oplus \bar{3} = \bar{3}. \bar{0} \oplus \bar{4} = \bar{4}. \\ & \bar{0} \oplus \bar{5} = \bar{5}. \bar{0} \oplus \bar{6} = \bar{6}. \bar{0} \oplus \bar{7} = \bar{7}. \bar{0} \oplus \bar{8} = \bar{8}. \bar{0} \oplus \bar{9} = \bar{9}. \bar{1} \oplus \bar{0} = \bar{1}. \\ & \bar{1} \oplus \bar{1} = \bar{2}. \bar{1} \oplus \bar{2} = \bar{3}. \bar{1} \oplus \bar{3} = \bar{4}. \bar{1} \oplus \bar{4} = \bar{5}. \bar{1} \oplus \bar{5} = \bar{6}. \bar{1} \oplus \bar{6} = \bar{7}. \\ & \bar{1} \oplus \bar{7} = \bar{8}. \bar{1} \oplus \bar{8} = \bar{9}. \bar{1} \oplus \bar{9} = \bar{1} \triangleright \bar{0}. \bar{2} \oplus \bar{0} = \bar{2}. \bar{2} \oplus \bar{1} = \bar{3}. \\ & \bar{2} \oplus \bar{2} = \bar{4}. \bar{2} \oplus \bar{3} = \bar{5}. \bar{2} \oplus \bar{4} = \bar{6}. \bar{2} \oplus \bar{5} = \bar{7}. \bar{2} \oplus \bar{6} = \bar{8}. \bar{2} \oplus \bar{7} = \bar{9}. \\ & \bar{2} \oplus \bar{8} = \bar{1} \triangleright \bar{0}. \bar{2} \oplus \bar{9} = \bar{1} \triangleright \bar{1}. \bar{3} \oplus \bar{0} = \bar{3}. \bar{3} \oplus \bar{1} = \bar{4}. \bar{3} \oplus \bar{2} = \bar{5}. \\ & \bar{3} \oplus \bar{3} = \bar{6}. \bar{3} \oplus \bar{4} = \bar{7}. \bar{3} \oplus \bar{5} = \bar{8}. \bar{3} \oplus \bar{6} = \bar{9}. \bar{3} \oplus \bar{7} = \bar{1} \triangleright \bar{0}. \\ & \bar{3} \oplus \bar{8} = \bar{1} \triangleright \bar{1}. \bar{3} \oplus \bar{9} = \bar{1} \triangleright \bar{2}. \bar{4} \oplus \bar{0} = \bar{4}. \bar{4} \oplus \bar{1} = \bar{5}. \bar{4} \oplus \bar{2} = \bar{6}. \\ & \bar{4} \oplus \bar{3} = \bar{7}. \bar{4} \oplus \bar{4} = \bar{8}. \bar{4} \oplus \bar{5} = \bar{9}. \bar{4} \oplus \bar{6} = \bar{1} \triangleright \bar{0}. \bar{4} \oplus \bar{7} = \bar{1} \triangleright \bar{1}. \\ & \bar{4} \oplus \bar{8} = \bar{1} \triangleright \bar{2}. \bar{4} \oplus \bar{9} = \bar{1} \triangleright \bar{3}. \bar{5} \oplus \bar{0} = \bar{5}. \bar{5} \oplus \bar{1} = \bar{6}. \bar{5} \oplus \bar{2} = \bar{7}. \\ & \bar{5} \oplus \bar{3} = \bar{8}. \bar{5} \oplus \bar{4} = \bar{9}. \bar{5} \oplus \bar{5} = \bar{1} \triangleright \bar{0}. \bar{5} \oplus \bar{6} = \bar{1} \triangleright \bar{1}. \bar{5} \oplus \bar{7} = \bar{1} \triangleright \bar{2}. \\ & \bar{5} \oplus \bar{8} = \bar{1} \triangleright \bar{3}. \bar{5} \oplus \bar{9} = \bar{1} \triangleright \bar{4}. \bar{6} \oplus \bar{0} = \bar{6}. \bar{6} \oplus \bar{1} = \bar{7}. \bar{6} \oplus \bar{2} = \bar{8}. \\ & \bar{6} \oplus \bar{3} = \bar{9}. \bar{6} \oplus \bar{4} = \bar{1} \triangleright \bar{0}. \bar{6} \oplus \bar{5} = \bar{1} \triangleright \bar{1}. \bar{6} \oplus \bar{6} = \bar{1} \triangleright \bar{2}. \bar{6} \oplus \bar{7} = \bar{1} \triangleright \bar{3}. \\ & \bar{6} \oplus \bar{8} = \bar{1} \triangleright \bar{4}. \bar{6} \oplus \bar{9} = \bar{1} \triangleright \bar{5}. \bar{7} \oplus \bar{0} = \bar{7}. \bar{7} \oplus \bar{1} = \bar{8}. \bar{7} \oplus \bar{2} = \bar{9}. \\ & \bar{7} \oplus \bar{3} = \bar{7} \triangleright \bar{0}. \bar{7} \oplus \bar{4} = \bar{1} \triangleright \bar{1}. \bar{7} \oplus \bar{5} = \bar{1} \triangleright \bar{2}. \bar{7} \oplus \bar{6} = \bar{1} \triangleright \bar{3}. \\ & \bar{7} \oplus \bar{7} = \bar{1} \triangleright \bar{4}. \bar{7} \oplus \bar{8} = \bar{1} \triangleright \bar{5}. \bar{7} \oplus \bar{9} = \bar{1} \triangleright \bar{6}. \bar{8} \oplus \bar{0} = \bar{8}. \bar{8} \oplus \bar{1} = \bar{9}. \\ & \bar{8} \oplus \bar{2} = \bar{1} \triangleright \bar{0}. \bar{8} \oplus \bar{3} = \bar{1} \triangleright \bar{1}. \bar{8} \oplus \bar{4} = \bar{1} \triangleright \bar{2}. \bar{8} \oplus \bar{5} = \bar{1} \triangleright \bar{3}. \\ & \bar{8} \oplus \bar{6} = \bar{1} \triangleright \bar{4}. \bar{8} \oplus \bar{7} = \bar{1} \triangleright \bar{5}. \bar{8} \oplus \bar{8} = \bar{1} \triangleright \bar{6}. \bar{8} \oplus \bar{9} = \bar{1} \triangleright \bar{7}. \\ & \bar{9} \oplus \bar{0} = \bar{9}. \bar{9} \oplus \bar{1} = \bar{1} \triangleright \bar{0}. \bar{9} \oplus \bar{2} = \bar{1} \triangleright \bar{1}. \bar{9} \oplus \bar{3} = \bar{1} \triangleright \bar{2}. \bar{9} \oplus \bar{4} = \bar{1} \triangleright \bar{3}. \\ & \bar{9} \oplus \bar{5} = \bar{1} \triangleright \bar{4}. \bar{9} \oplus \bar{6} = \bar{1} \triangleright \bar{5}. \bar{9} \oplus \bar{7} = \bar{1} \triangleright \bar{6}. \bar{9} \oplus \bar{8} = \bar{1} \triangleright \bar{7}. \\ & \bar{9} \oplus \bar{9} = \bar{1} \triangleright \bar{8}. \end{aligned}$$

Axioms D1 through D4 capture our intuitive, but rather vague, understanding of decimal addition. For the *DECL* language to be useful in deriving formal proofs,

we also need an induction axiom. The induction axiom for *DECL*, which is stated in Axiom 5, is based on the number of digits of a decimal numeral.

$$\begin{aligned} \text{Axiom D5. } & A(\bar{0}) \rightarrow (A(\bar{1}) \rightarrow (A(\bar{2}) \rightarrow (A(\bar{3}) \rightarrow (A(\bar{4}) \rightarrow (A(\bar{5}) \rightarrow \\ & (A(\bar{6}) \rightarrow (A(\bar{7}) \rightarrow (A(\bar{8}) \rightarrow (A(\bar{9}) \rightarrow [\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \\ & \bar{0})) \rightarrow (\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{1})) \rightarrow (\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \\ & \bar{2})) \rightarrow (\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{3})) \rightarrow (\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \\ & \bar{4})) \rightarrow (\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{5})) \rightarrow (\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \\ & \bar{6})) \rightarrow (\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{7})) \rightarrow (\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{8})) \rightarrow \\ & (\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{9})) \rightarrow \forall(\alpha)A(\alpha)))))))))))). \end{aligned}$$

The induction axiom for *DECL* is also a constraint on the set of decimal numerals. It states that every term of *DECL* is equivalent to a decimal numeral, which is either a decimal digit or has the form  $\alpha \triangleright d$ , where  $\alpha$  is a (simpler) decimal numeral and  $d$  is a decimal digit.

Note that the function  $\triangleright$  is not associative; for example,  $(\bar{3} \triangleright \bar{4}) \triangleright \bar{5}$  (which is intended to denote the natural number 345) is not equivalent to  $\bar{3} \triangleright (\bar{4} \triangleright \bar{5})$  (which is intended to denote the natural number 75). The latter term is equivalent to  $\bar{7} \triangleright \bar{5}$  according to Axioms D3 and D4. Specifically, the following formula is not a theorem in *DECL*:  $\forall(\alpha)\forall(\beta)\forall(\gamma)((\bar{\alpha} \triangleright \bar{\beta}) \triangleright \bar{\gamma} = \bar{\alpha} \triangleright (\bar{\beta} \triangleright \bar{\gamma}))$ .

In summary, a term in *DECL* may be a decimal digit, a variable, or may have the form  $\alpha \triangleright \beta$  or  $\alpha \oplus \beta$ , where  $\alpha$  and  $\beta$  are terms of *DECL*.

**Definition.** Let  $F$  be a well-formed formula in *DECL*. We say that  $\vdash_D F$  if and only if  $F$  is provable from Axioms D1 through D5, E1 through E7 and the two inference rules I1 and I2.

We will need the following lemma in the proof of the completeness theorem in the next section.

**Lemma 4.1.**  $\vdash_D \forall(\alpha)(\alpha \oplus \bar{0} = \alpha)$ .

**Proof.** We can prove this lemma with the induction axiom. Let  $A(\alpha)$  be the formula  $\alpha \oplus \bar{0} = \alpha$ .

1.  $A(\bar{0}) \rightarrow (A(\bar{1}) \rightarrow (A(\bar{2}) \rightarrow (A(\bar{3}) \rightarrow (A(\bar{4}) \rightarrow (A(\bar{5}) \rightarrow (A(\bar{6}) \rightarrow (A(\bar{7}) \rightarrow (A(\bar{8}) \rightarrow (A(\bar{9}) \rightarrow [\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{0})) \rightarrow (\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{1})) \rightarrow (\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{2})) \rightarrow (\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{3})) \rightarrow (\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{4})) \rightarrow (\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{5})) \rightarrow (\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{6})) \rightarrow (\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{7})) \rightarrow (\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{8})) \rightarrow (\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{9})) \rightarrow \forall(\alpha)A(\alpha)))))))))))])))))$ ,

Axiom D5.

2.  $\bar{0} \oplus \bar{0} = \bar{0}$ , Axiom D4. This formula can be written as  $A(\bar{0})$ .

3.  $\bar{1} \oplus \bar{0} = \bar{1}$ , Axiom D4. This formula can be written as  $A(\bar{1})$ .

4.  $\bar{2} \oplus \bar{0} = \bar{2}$ , Axiom D4. This formula can be written as  $A(\bar{2})$ .

5.  $\bar{3} \oplus \bar{0} = \bar{3}$ , Axiom D4. This formula can be written as  $A(\bar{3})$ .

6.  $\bar{4} \oplus \bar{0} = \bar{4}$ , Axiom D4. This formula can be written as  $A(\bar{4})$ .

7.  $\bar{5} \oplus \bar{0} = \bar{5}$ , Axiom D4. This formula can be written as  $A(\bar{5})$ .

8.  $\bar{6} \oplus \bar{0} = \bar{6}$ , Axiom D4. This formula can be written as  $A(\bar{6})$ .

9.  $\bar{7} \oplus \bar{0} = \bar{7}$ , Axiom D4. This formula can be written as  $A(\bar{7})$ .

10.  $\bar{8} \oplus \bar{0} = \bar{8}$ , Axiom D4. This formula can be written as  $A(\bar{8})$ .

11.  $\bar{9} \oplus \bar{0} = \bar{9}$ , Axiom D4. This formula can be written as  $A(\bar{9})$ .

12.  $\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{0})) \rightarrow (\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{1})) \rightarrow (\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{2})) \rightarrow (\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{3})) \rightarrow (\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{4})) \rightarrow (\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{5})) \rightarrow (\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{6})) \rightarrow (\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{7})) \rightarrow (\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{8})) \rightarrow (\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{9})) \rightarrow \forall(\alpha)A(\alpha))))))))))$ , 1, 2, 3, 4, 5, 6, 7, 8, 9,

10 and 11 by *modus ponens*.

Next we want to prove that  $\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{0}))$  (in line 25 below).

13.  $\bar{0} \triangleright \bar{0} = \bar{0}$ , Axiom D1.

14.  $(\alpha \triangleright \bar{0}) \oplus \bar{0} = (\alpha \triangleright \bar{0}) \oplus \bar{0}$ , Axiom E6.

15.  $(\alpha \triangleright \bar{0}) \oplus \bar{0} = (\alpha \triangleright \bar{0}) \oplus (\bar{0} \triangleright \bar{0})$ , 13 and 14 by Axiom E7.

16.  $(\alpha \triangleright \bar{0}) \oplus (\bar{0} \triangleright \bar{0}) = (\alpha \oplus \bar{0}) \triangleright (\bar{0} \oplus \bar{0})$ , Axiom D2.

17.  $\bar{0} \oplus \bar{0} = \bar{0}$ , Axiom D4.

18.  $(\alpha \triangleright \bar{0}) \oplus (\bar{0} \triangleright \bar{0}) = (\alpha \oplus \bar{0}) \triangleright \bar{0}$ , 16 and 17 by Axiom E7.

19.  $(\alpha \triangleright \bar{0}) \oplus \bar{0} = (\alpha \oplus \bar{0}) \triangleright \bar{0}$ , 15 and 18 by Axiom E7.

20.  $(\alpha \triangleright \bar{0}) \oplus \bar{0} = (\alpha \oplus \bar{0}) \triangleright \bar{0} \rightarrow [\alpha \oplus \bar{0} = \alpha \rightarrow (\alpha \triangleright \bar{0}) \oplus \bar{0} = (\alpha \oplus \bar{0}) \triangleright \bar{0}]$ , Axiom E1.

21.  $\alpha \oplus \bar{0} = \alpha \rightarrow (\alpha \triangleright \bar{0}) \oplus \bar{0} = (\alpha \oplus \bar{0}) \triangleright \bar{0}$ , 19 and 20 by *modus ponens*.

22.  $\alpha \oplus \bar{0} = \alpha \rightarrow [(\alpha \triangleright \bar{0}) \oplus \bar{0} = (\alpha \oplus \bar{0}) \triangleright \bar{0} \rightarrow (\alpha \triangleright \bar{0}) \oplus \bar{0} = \alpha \triangleright \bar{0}]$ , Axiom E7.

23.  $[\alpha \oplus \bar{0} = \alpha \rightarrow (\alpha \triangleright \bar{0}) \oplus \bar{0} = (\alpha \oplus \bar{0}) \triangleright \bar{0}] \rightarrow [\alpha \oplus \bar{0} = \alpha \rightarrow (\alpha \triangleright \bar{0}) \oplus \bar{0} = \alpha \triangleright \bar{0}]$ , 22 by Axiom E2.

24.  $\alpha \oplus \bar{0} = \alpha \rightarrow (\alpha \triangleright \bar{0}) \oplus \bar{0} = \alpha \triangleright \bar{0}$ , 21 and 23 by *modus ponens*. This formula can be written as  $A(\alpha) \rightarrow A(\alpha \triangleright \bar{0})$ .

25.  $\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{0}))$ , 24 by generalization.

By a similar argument (from line 13 to line 25), we can derive the following formulae:

26.  $\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{1}))$ , similar to line 13 through line 25.

27.  $\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{2}))$ , similar to line 13 through line 25.

28.  $\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{3}))$ , similar to line 13 through line 25.

29.  $\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{4}))$ , similar to line 13 through line 25.

30.  $\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{5}))$ , similar to line 13 through line 25.

31.  $\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{6}))$ , similar to line 13 through line 25.

32.  $\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{7}))$ , similar to line 13 through line 25.

33.  $\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{8}))$ , similar to line 13 through line 25.

34.  $\forall(\alpha)(A(\alpha) \rightarrow A(\alpha \triangleright \bar{9}))$ , similar to line 13 through line 25.

35.  $\forall(\alpha)A(\alpha)$ , 12, 25-34 by *modus ponens*. This formula can be written as  $\forall(\alpha)(\alpha \oplus \bar{0} = \alpha)$ .

**qed**

## 5. SOUNDNESS AND COMPLETENESS OF THE *DECL* LANGUAGE

Now that we have finished our formalization of the decimal addition method, we need to show that the decimal addition method is *correct* in the sense that there

is a mapping  $\mathcal{M}$  from the terms of *DECL* to the terms of *APPLE* under which *DECL* is *sound* and *complete*.

**Definition.** Let  $\alpha$  and  $\beta$  be two terms of *DECL*. Let  $\mathcal{M}$  be a mapping from the terms of *DECL* to the terms of *APPLE*. We say that  $\models_{D,\mathcal{M}} \alpha = \beta$  if and only if  $\vdash_A \mathcal{M}[\alpha] = \mathcal{M}[\beta]$ . We say that  $\vdash_D \alpha = \beta$  if and only if the well-formed formula  $(\alpha = \beta)$  is provable from Axioms D1 through D5, E1 through E7 and the two inference rules I1 and I2.

The main result of this paper is the following theorem. The first conclusion of the theorem is concerned with the soundness property, while the second conclusion is concerned with the completeness property. Note that soundness and completeness is established through the same mapping,  $\mathcal{M}$ .

**Theorem** (Soundness and Completeness Theorem of *DECL*). There is a mapping  $\mathcal{M}$  from the terms of *DECL* to the terms of *APPLE* such that (1)  $\vdash_D (\alpha \oplus \beta) = \gamma$  implies  $\models_{D,\mathcal{M}} (\alpha \oplus \beta) = \gamma$ , (2)  $\models_{D,\mathcal{M}} (\alpha \oplus \beta) = \gamma$  implies  $\vdash_D (\alpha \oplus \beta) = \gamma$ , where  $\alpha$ ,  $\beta$ , and  $\gamma$  are decimal numerals.

In order to prove the completeness and soundness theorem, we will first define the mapping  $\mathcal{M}$ . Though we only need to consider the case where the domain of  $\mathcal{M}$  is the set of decimal numerals, it is more convenient to enlarge the domain of  $\mathcal{M}$  to include all the terms of *DECL*. Since a term is one of the ten decimal digits or a variable, or it has the form  $(\alpha \triangleright \beta)$  or  $(\alpha \oplus \beta)$ , where  $\alpha$  and  $\beta$  are terms, the mapping  $\mathcal{M}$  can be defined inductively by the following eleven equations:

$$\begin{aligned} \mathcal{M}[\overline{0}] &\equiv \emptyset \\ \mathcal{M}[\overline{1}] &\equiv s(\emptyset) \\ \mathcal{M}[\overline{2}] &\equiv s(s(\emptyset)) \\ \mathcal{M}[\overline{3}] &\equiv s(s(s(\emptyset))) \\ \mathcal{M}[\overline{4}] &\equiv s(s(s(s(\emptyset)))) \\ \mathcal{M}[\overline{5}] &\equiv s(s(s(s(s(\emptyset)))))) \\ \mathcal{M}[\overline{6}] &\equiv s(s(s(s(s(s(\emptyset))))))) \\ \mathcal{M}[\overline{7}] &\equiv s(s(s(s(s(s(s(\emptyset)))))))) \end{aligned}$$



$\mathcal{M}[\alpha = \beta] \equiv \mathcal{M}[\alpha] = \mathcal{M}[\beta]$ , where  $\alpha$  and  $\beta$  are terms of *DECL*.

$\mathcal{M}[\neg A] \equiv \neg \mathcal{M}[A]$ .

$\mathcal{M}[A \rightarrow B] \equiv \mathcal{M}[A] \rightarrow \mathcal{M}[B]$ .

$\mathcal{M}[\forall(\alpha)A] \equiv \forall(\alpha)\mathcal{M}[A]$ .

Similarly, we say that  $\models_{D,\mathcal{M}} F$  if and only if  $\vdash_A \mathcal{M}[F]$ , and that  $\vdash_D F$  if and only if  $F$  is provable from Axioms D1 through D5 and E1 through E7.

**Lemma 5.2.** Let  $G$  be a formula of *APPLE*. There exists another formula  $H$  of *APPLE* such that  $\vdash_A G \rightarrow H$ ,  $\vdash_A H \rightarrow G$ , and  $H$  is in the range of the mapping  $\mathcal{M}$ .

**Proof.**  $H$  is obtained by replacing every term of  $G$  with an equivalent term that is in the range of  $\mathcal{M}$ . By Axiom E7, we know that  $\vdash_A G \rightarrow H$  and  $\vdash_A H \rightarrow G$ .

**qed**

Based on the above definition of  $\mathcal{M}$ , we will proceed to prove the soundness property. The completeness property will be proved later.

5.1. **Soundness.** We will prove the soundness property in this subsection.

**Lemma 5.3.** Let  $\alpha \oplus \beta = \gamma$  be an equation in Axiom D4. Then  $\vdash_A \mathcal{M}[\alpha] + \mathcal{M}[\beta] = \mathcal{M}[\gamma]$ . (Hence,  $\models_{D,\mathcal{M}} \alpha \oplus \beta = \gamma$ .)

**Proof.** We can verify the formulae in Axiom D4 one by one. **qed**

**Lemma 5.4.** Let  $F$  be a formula in *DECL*. Then  $\vdash_D F$  implies  $\vdash_A \mathcal{M}[F]$ .

**Proof.** Assume that  $\vdash_D F$ . To derive  $\vdash_D F$ , we can use Axioms E1 through E7, D1 through D5 and the inference rules. We will prove this lemma by induction on the length of the derivation of  $\vdash_D F$ .

*Base case.* The length of the derivation is 1. This means that  $\vdash_D F$  must be one of Axioms E1 through E7 or D1 through D5. If  $\vdash_D F$  is one of Axioms E1 through E7, obviously,  $\vdash_A \mathcal{M}[F]$  must also be one of Axioms E1 through E7. We need to verify Axioms D1 through D5.

For Axiom D1, we have  $\mathcal{M}[\forall(\alpha)((\bar{0} \triangleright \alpha) = \alpha)] \equiv \forall(\alpha)(\emptyset + \emptyset + \emptyset + \emptyset + \emptyset + \emptyset + \emptyset + \emptyset + \emptyset + \emptyset + \alpha = \alpha)$ , which can be derived in *APPLE*.



$\alpha + \alpha + \alpha + s(s(s(s(s(s(s(\emptyset)))))))) \rightarrow (\forall(\alpha)(\mathcal{M}[A](\alpha) \rightarrow \mathcal{M}[A](\alpha + \alpha + \alpha + \alpha + \alpha + \alpha + \alpha + s(s(s(s(s(s(s(\emptyset)))))))))) \rightarrow \forall(\alpha)A(\alpha))$ .

The last formula can be proved in *APPLE*. For the sake of brevity, we will not include the details here.

*Induction hypothesis.* We may assume that  $\vdash_D F$  implies  $\vdash_A \mathcal{M}[F]$  if the length of the derivation of  $\vdash_D F$  is less than  $k$ , where  $k$  is a positive integer.

*Induction step.* Suppose that the length of the derivation of  $\vdash_D F$  is  $k$ . We need to show that  $\vdash_A \mathcal{M}[F]$ . There are three subcases to consider:

subcase 1.  $F$  is an axiom. This subcase is similar to the above base case.

subcase 2.  $F$  is obtained from two previous formulae  $G$  and  $G \rightarrow F$  by the *modus ponens* inference rule. Because both formulae  $\vdash_D G$  and  $\vdash_D G \rightarrow F$  may be derived in less than  $k$  steps, by the induction hypothesis, we have  $\vdash_A \mathcal{M}[G]$  and  $\vdash_A \mathcal{M}[G \rightarrow F]$ . Note that  $\vdash_A \mathcal{M}[G \rightarrow F]$  is the same as  $\vdash_A \mathcal{M}[G] \rightarrow \mathcal{M}[F]$ . Again, by the *modus ponens* rule, we have  $\vdash_A \mathcal{M}[F]$ .

subcase 3.  $F$  is obtained from a previous formula by the *generalization* inference rule. Thus,  $F$  must have the form  $\forall(\alpha)G$ , where  $\alpha$  is a variable. By the above assumption,  $\vdash_D G$  can be derived in less than  $k$  steps. Therefore, by the induction hypothesis, we have  $\vdash_A \mathcal{M}[G]$ . Again, by the *generalization* rule, we have  $\vdash_A \forall(\alpha)\mathcal{M}[G]$ , which is equivalent to  $\vdash_A \mathcal{M}[\forall(\alpha)G]$  and  $\vdash_A \mathcal{M}[F]$ . **qed**

**Corollary.** Let  $\alpha$  and  $\beta$  be any two terms of *DECL*.  $\vdash_D \alpha = \beta$  implies  $\vdash_A \mathcal{M}[\alpha] = \mathcal{M}[\beta]$ .

Note that the soundness property is a corollary to the above lemma. Therefore, we have proved the soundness property. In what follows, we will prove the completeness property.

**5.2. Completeness.** Note that every derivable formula  $K$  in *APPLE* (i.e.  $\vdash_A K$ ) may have many different derivations, the lengths of which may also be different. First we choose a derivation  $K_1, K_2, \dots, K$  of  $K$ . We call this derivation  $\Delta$ . Then we augment  $\Delta$  as follows: For every formula  $K_i$  in the derivation, if  $K_i$  is not in the range of  $\mathcal{M}$ , by Lemma 5.2, there is another formula  $L_i$  such that (1)  $\vdash_A K_i \rightarrow L_i$ ,

(2)  $\vdash_A L_i \rightarrow K_i$ , and (3)  $L_i$  is in the range of  $\mathcal{M}$ . Furthermore,  $L_i$  can be derived from  $K_i$  directly with Axiom E7 (by replacing the terms in  $K_i$  one after another).

The derivation goes as follows:

1.  $K_i(s)$ , already proved (where  $s$  is a term in  $K_i$ ).
2.  $s = t$ , where  $t$  is another term that is equivalent to  $s$  and that is in the range of  $\mathcal{M}$ . Note that  $t$  exists by Lemma 5.1.
3.  $s = t \rightarrow [K_i(s) \rightarrow K_i(t)]$ , Axiom E7.
4.  $K_i(s) \rightarrow K_i(t)$ , 2 and 3 by *modus ponens*.
5.  $K_i(t)$ , 1 and 4 by *modus ponens*.

We can repeat the above five steps for every term in  $K_i$ . The final result is the desired  $L_i$ .

For every formula  $K_i$  in the original derivation  $\Delta$  that is not in the range of  $\mathcal{M}$ , we add the derivation of  $L_i$  (from  $K_i$ ) immediately after  $K_i$  in the original derivation  $\Delta$ . We will call the result the *augmented derivation* of  $K$ .

**Lemma 5.5.** Let  $F$  be a formula in *DECL*. Then  $\vdash_A \mathcal{M}[F]$  implies  $\vdash_D F$ .

**Proof.** We will prove this lemma by induction on the length of an augmented derivation of  $\mathcal{M}[F]$  in *APPLE*.

*Base case.* The length of the derivation of  $\mathcal{M}[F]$  is 1. Then  $\mathcal{M}[F]$  must be one of Axioms E1 through E7 or A1 through A5. If  $\mathcal{M}[F]$  is an instance of Axioms E1 through E7, then we have  $\vdash_D F$  since Axioms E1 through E7 are also axioms of *DECL*. We need to verify Axioms A1 through A5.

First notice that  $\mathcal{M}[F]$  cannot be an instance of Axioms A1, A3, A4 or A5 because  $\mathcal{M}[F]$  cannot contain a term of the form  $s(x)$ .

If  $\mathcal{M}[F]$  has the form  $\forall(x)(x+\emptyset = x)$ , then  $F$  must have the form  $\forall(\alpha)(\alpha\oplus\bar{0} = \alpha)$ , which has been proved as Lemma 4.1.

*Induction hypothesis.* We may assume that the lemma holds for any formula  $F$  for which the length of the derivation of  $\mathcal{M}[F]$  in *APPLE* is less than  $k$ , where  $k$  is a positive integer.

*Induction step.* We need to prove that the lemma holds for any formula  $F$  such that the length of the derivation of  $\mathcal{M}[F]$  in *APPLE* is  $k$ . Suppose that the length of the derivation of  $\mathcal{M}[F]$  in *APPLE* is  $k$ .

There are three subcases to consider:

subcase 1.  $\mathcal{M}[F]$  is an axiom. This subcase is similar to the above base case.

subcase 2.  $\mathcal{M}[F]$  is obtained from a previous formula by the *generalization* inference rule. Thus,  $\mathcal{M}[F]$  must have the form  $\forall(\alpha)G$ , where  $\alpha$  is a variable. Because  $\forall(\alpha)G$  is in the range of  $\mathcal{M}$ , so is  $G$ . In this subcase, there must be a formula  $G'$  in *DECL* such that  $\mathcal{M}[G']$  is  $G$ . By the above assumption,  $\vdash_A \mathcal{M}[G']$  can be derived in less than  $k$  steps. Therefore, by the induction hypothesis, we have  $\vdash_D G'$ . Again by the *generalization* rule, we have  $\vdash_D \forall(\alpha)G'$ , which is equivalent to  $\vdash_D F$ .

subcase 3.  $\mathcal{M}[F]$  is obtained from two previous formulae  $G$  and  $G \rightarrow \mathcal{M}[F]$  by the *modus ponens* inference rule.

subcase 3.1. First assume that  $G$  is in the range of the mapping  $\mathcal{M}$ . Then there is a formula  $G'$  in *DECL* such that  $\mathcal{M}[G']$  is  $G$ . In this case, we can use an argument similar to subcase 2 above to prove that  $\vdash_D F$ .

subcase 3.2. However, if  $G$  is not in the range of the mapping  $\mathcal{M}$ , by Lemma 5.2, we may find another formula  $H$  in *APPLE* such that  $\vdash_A G \rightarrow H$  and  $\vdash_A H \rightarrow G$  and  $H$  is in the range of the mapping  $\mathcal{M}$ .

Due to our choice of the derivation of  $\mathcal{M}[F]$ ,  $H$  is also in the derivation of  $\mathcal{M}[F]$ .

From  $\vdash_A G$  and  $\vdash_A G \rightarrow H$ , we can obtain  $\vdash_A H$  by the *modus ponens* rule. Similarly, from  $\vdash_A H \rightarrow G$  and  $\vdash_A G \rightarrow \mathcal{M}[F]$ , we can obtain  $\vdash_A H \rightarrow \mathcal{M}[F]$  by the *modus ponens* rule. Therefore, we can say that  $\mathcal{M}[F]$  is obtained from  $H$  and  $H \rightarrow \mathcal{M}[F]$  by the *modus ponens* rule. Furthermore,  $H$  is in the range of the mapping  $\mathcal{M}$ . We can use the argument in subcase 3.1 to prove that  $\vdash_D F$ . **qed**

Note that the completeness property is a corollary to the above lemma. Therefore, we have proved the completeness property.

The crux of the above proof is that every formula that is in the range of the mapping  $\mathcal{M}$  can be proved with the help of other formulae of *APPLE* that are also in the range of  $\mathcal{M}$ .

## 6. EXTENSION TO OTHER NUMERICAL NOTATIONS

The *DECL* language deals with decimal numerals. It should be obvious that the above method can be applied to other similar systems, such as binary numerals, octal numerals, and hexadecimal numerals.

We can also extend the method presented in this paper to other dissimilar systems, such as Roman numerals. A key technique is to define a language (similar to *DECL*) that *covers* the language of Roman numerals. Note that *DECL* is also a superset of the language of decimal numerals. Then we can list several axioms for this language. The completeness and soundness properties of this system can be proved in a similar way. This technique is applicable to the Chinese abacus numerals as well.

## 7. CONCLUSION

In this paper, we have carefully provided an axiomatic system for decimal addition and proved the soundness and completeness of this system. Our method can be easily generalized to other numeral systems.

May you sleep well tonight, with increased confidence in decimal addition and, consequently, in all the artifacts built with decimal addition, which include, I guess, the bed you sleep on.

## REFERENCES

- [1] J.V. Guttag and J.J. Horning, "The algebraic specification of abstract data types," in D. Gries (ed.), *Programming Methodology*, Springer-Verlag, New York, 1978, pp. 282-308.
- [2] D. Gries, *The Science of Programming*, Springer-Verlag, New York, 1981.
- [3] C.A.R. Hoare, "An axiomatic basis for computer programming," *Comm. ACM*, Vol. 12, No. 10, 1969, pp. 576-583.
- [4] J. Kelly, *The Essence of Logic*, Prentice-Hall, New York, 1997.
- [5] R.E. Hodel, *An Introduction to Mathematical Logic*, PWS Publishing, Boston, 1995.

AUTHOR'S ADDRESS: WUU YANG, DEPARTMENT OF COMPUTER AND INFORMATION SCIENCE,  
NATIONAL CHIAO-TUNG UNIVERSITY, HSIN-CHU, TAIWAN, REPUBLIC OF CHINA  
*E-mail address:* `wuyang@cis.nctu.edu.tw`