Understanding Decimal Addition

2.1 Experience Versus Understanding

This book is about understanding system architecture in a quick and clean way: no black art, nothing you can only get a feeling for after years of programming experience. While experience is good, it does not replace understanding. For illustration, consider the basic method for decimal addition of one-digit numbers as taught in the first weeks of elementary school: everybody has experience with it, almost nobody understands it; very few of the people who do not understand it realize that they don’t.

Recall that, in mathematics, there are definitions and statements. Statements that we can prove are called theorems. Some true statements, however, are so basic that there are no even more basic statements that we can derive them from; these are called axioms. A person who understands decimal addition will clearly be able to answer the following simple

Questions: Which of the following equations are definitions? Which ones are theorems? If an equation is a theorem, what is the proof?

\[
\begin{align*}
2 + 1 & = 3, \\
1 + 2 & = 3, \\
9 + 1 & = 10.
\end{align*}
\]

We just stated that these questions are simple; we did not say that answering them is easy. Should you care? Indeed you should, for at least three reasons: i) In case you don’t even understand the school method for decimal addition, how can you hope to understand computer systems? ii) The reason why the school method works has very much to do with the reason why binary addition in the fixed-point adders of processors works. iii) You should learn to distinguish between having experience with something that has not gone wrong (yet) and having an explanation of why it always works. The authors of this text consider iii) the most important.
2.2 The Natural Numbers

In order to answer the above questions, we first consider counting. Since we count by repeatedly adding 1, this should be a step in the right direction.

The set of natural numbers \( \mathbb{N} \) and the properties of counting are not based on ordinary mathematical definitions. In fact, they are so basic that we use five axioms due to Peano to simultaneously lay down all properties about the natural numbers and of counting we will ever use without proof. The axioms talk about

- a special number 0,
- the set \( \mathbb{N} \) of all natural numbers (with zero),
- counting formalized by a successor function \( S : \mathbb{N} \to \mathbb{N} \), and
- subsets \( A \subseteq \mathbb{N} \) of the natural numbers.

Peano’s axioms are

1. \( 0 \in \mathbb{N} \). Zero is a natural number. Note that this is a modern view of counting, because zero counts something that could be there but isn’t.
2. \( x \in \mathbb{N} \rightarrow S(x) \in \mathbb{N} \). You can always count to the next number.
3. \( x \neq y \rightarrow S(x) \neq S(y) \). Different numbers have different successors.
4. \( \nexists y \in \mathbb{N} \). By counting you cannot arrive at 0. Note that this isn’t true for computer arithmetic, where you can arrive at zero by an overflow of modulo arithmetic (see Sect. 3.2).
5. \( A \subseteq \mathbb{N} \land 0 \in A \land (n \in A \rightarrow S(n) \in A) \rightarrow A = \mathbb{N} \). This is the famous induction scheme for proofs by induction. We give plenty of examples later.

In a proof by induction, one usually considers a set \( A \) consisting of all numbers \( n \) satisfying a certain property \( P(n) \):

\[
A = \{ n \in \mathbb{N} \mid P(n) \}.
\]

Then,

\[
n \in A \iff P(n),
A = \mathbb{N} \iff \forall n \in \mathbb{N} : P(n),
\]

and the induction axiom translates into a proof scheme you might or might not know from high school:

- Start of the induction: show \( P(0) \).
- Induction step: show that \( P(n) \) implies \( P(S(n)) \).
- Conclude \( \forall n \in \mathbb{N} : P(n) \). Property \( P \) holds for all natural numbers.

With the rules of counting laid down by the Peano axioms, we are able to make two ‘ordinary’ definitions. We define 1 to be the next number after 0 if you count. We also define that addition of 1 is counting.

**Definition 1 (Adding 1 by Counting).**

\[
1 = S(0),
\]
\[
x + 1 = S(x).
\]