## Today's lecture: Mathematical Background

**I.** Induction
**II.** Strings
**III.** An Example

## Schedule:

**Today:** Mathematical Background – Read: *Sipser* chapter 0
Homework 0 goes out (due Sept. 18).

**September 11:** Finite Automata – Read: *Sipser* 1.1.
Lecture will cover through Example 1.15 (i.e. pages 31–40).

**September 13:** Regular Languages.
The rest of *Sipser* 1.1 (i.e. pages 40–47).

**September 15:** Non-Determinism – Read: *Sipser* 1.2.
Lecture will cover through Example 1.35 (i.e. pages 47–52).
Homework 1 goes out (due Sept. 25).

**September 18:** NFAs
Lecture will cover through Example 1.38 (i.e. pages 53-54).
Homework 0 due.

**September 20** and beyond: see Sept. 6 notes.

---

**I.** Induction

**A.** The Natural Numbers:

**1.** Defining the natural numbers:

**Axiom 0:** $0$ is a natural number.

**Axiom 1:** if $x$ is a natural number, so is $succ(x)$

**Axiom 2:** if $x$ is a natural number, $succ(x) > x$.

**Axiom 3:** if $x$ and $y$ are natural numbers and $x > y$, then $succ(x) > y$.

**Axiom 4:** if $x$ and $y$ are natural numbers and $x > y$, then $(x \neq y)$.

In English, this says that you can obtain any natural numbers by starting with $0$ and then successively adding one until you get to the one that you want – $succ$ is the *successor* function. Furthermore, this produces the numbers in ascending order, and each one is unique (i.e. you can't have $succ(succ(succ(\ldots succ(x)\ldots))) = x$, for any $x$ and any number of successor operations greater than zero). Having introduced $>$ and $\neq$, we can use the other comparison operators as well. More formally,

$$(x = y) \equiv \neg(x \neq y), \quad (x \leq y) \equiv ((x < y) \vee (x = y)),$$
$$(x \geq y) \equiv (y \leq x), \qquad (x > y) \equiv (y < x).$$

**2.** Operations on the natural numbers:

**a.** Addition:

$$\begin{aligned} x + 0 &= x, \\ x + succ(y) &= succ(x + y). \end{aligned}$$

**b.** Multiplication:

$$\begin{aligned} x * 0 &= 0, \\ x * succ(y) &= (x * y) + x. \end{aligned}$$

**c.** Division:

$$(x/y) = q \quad \Leftrightarrow \quad y * q = x.$$

Note that there division is not defined for all possible choices of $x$ and $y$. For example, if $x = 5$ and $y = 3$, there is no natural number, $q$, such that $x = q * y$. If $x = 0$ and $y = 0$, then $q$ can be any natural number. If $x \neq 0$ and $y = 0$, then there is no natural number $q$ that such that $y * q = x$.

**d.** Exponentiation:

$$\begin{aligned} x^0 &= succ(0), \\ x^{succ(y)} &= (x^y) * x. \end{aligned}$$

I will assume the usual precedence rules for expressions consisting of multiple operators.

**3.** Some handy abbreviations:

**a.** Decimal digits:

$$\begin{array}{llllll} 1 = succ(0), & 2 = succ(1), & 3 = succ(2), & 4 = succ(3), & 5 = succ(4), \\ 6 = succ(5), & 7 = succ(6), & 8 = succ(7), & 9 = succ(8), & 10 = succ(9). \end{array}$$

**b.** Multidigit numbers:

$$1437 \quad = \quad 1 * 10^3 + 4 * 10^2 + 3 * 10^1 + 7 * 10^0.$$

**B.** Let's prove that for any natural number, $n$,

$$\sum_{k=0}^{n} k \quad = \quad \frac{k^2 + k}{2}$$

Mark's Lazy Proof: I won't bother to do anything. If you question my claim, I'll ask you to propose a counterexample. Let's say you tell me that $m$ is a counterexample. I'll stall for time and ask you to prove to me that $m$ is a natural number. This means that you have to show me that

$$m \quad = \quad succ(succ(succ(\ldots succ(0) \ldots))).$$

Now, I'll prove that the formula holds when $m = 0$:

$$\textstyle\sum_{k=0}^{0} k \quad = \quad 0$$

by the definition of summation (which I haven't bothered to include in these notes). Now, we look at

$$\frac{0^2 + 0}{2}$$

We get

$$\begin{array}{lcll} \frac{0^2+0}{2} & = & \frac{0^2}{2}, & \text{def. } + \\ & = & \frac{0^{succ(succ(0))}}{2}, & \text{def. 2} \\ & = & \frac{(0*0)*0}{2}, & \text{def. exponentiation} \\ & = & \frac{0}{2}, & \text{def. multiplication} \\ & = & 0, & 2 * 0 = 0, \text{ def. division} \end{array}$$

Now consider the case when $m = succ(succ(succ(\ldots succ(0)\ldots)))$. Here's my strategy. Having proven the formula for $m = 0$, I'll construct a proof that goes along side the chain of $succ$ functions. Thus, I'll prove the claim first for 0, then for $succ(0)$, then for $succ(succ(0))$ and so on until I reach the number that you proposed. Let's say that I've proven the claim for $x$, then to prove it for $succ(x)$, I just have to show

$$
\begin{array}{lll}
\frac{succ(x)^2 + succ(x)}{2} & & \\
= & \frac{(x+1)^2+(x+1)}{2}, & x+1 = succ(x) \\
= & \frac{(x^2+2*x+1)+(x+1)}{2}, & \text{algebra} \\
= & \frac{(x^2+x)+2*(x+1)}{2}, & \text{more algebra} \\
= & \frac{x^2+x}{2} + \frac{2*(x+1)}{2}, & \text{more algebra} \\
= & \frac{x^2+x}{2} + (x+1), & \text{def. division} \\
= & (\sum_{k=0}^{x} k) + (x+1), & \sum_{k=0}^{x} k = \frac{k^2+k}{2} \text{ as already shown} \\
= & \sum_{k=0}^{succ(x)} k & \text{def. summation}
\end{array}
$$

When I write that something follows from "algebra", I'm using the distributive and associative properties for addition and multiplication. These can be proven using the definitions I gave above for natural numbers, multiplication, and addition.

**C.** Proof by induction:

Note that my "lazy proof" constructed a proof that followed the definition of a natural number. In other words, given that $m$ is a natural number, we must be able to derive $m$ by the cases of the definition. We can derive a proof that follows the same cases. By writing a separate proof for each case in the definition, we prove the result for anything that satisfies the definition.

This is called **proof by induction**. We also see that there is nothing magical about it; it's just "proof by the definition of the domain". Furthermore, induction is not limited to proofs about natural numbers. Induction can be applied to *anything* that has an inductive definition.

An inductive definition is one that is broken into multiple cases. Some of the cases define the primitive elements of the set. The other cases describe how to construct larger elements of the set from smaller ones. For the natural numbers, 0 was the primitive element, and $succ$ built large elements from smaller ones. We'll now see how we can do similar things with strings.

## II. Strings

### A. Definition

**1.** Informal: A string is a sequence of zero or more elements from a finite alphabet, $\Sigma$.

**2.** Inductive: given a finite alphabet, $\Sigma$, a string of elements of $\Sigma$ is either

$$
\begin{array}{rll}
 & \epsilon, & \text{the empty string} \\
\text{or} & w \cdot c, & \text{where } c \in \Sigma, \text{ and } w \text{ is a string.}
\end{array}
$$

**3.** $\Sigma^*$: we write $\Sigma^*$ to indicate the set of all strings composed of elements of $\Sigma$.

### B. Operations on Strings

**1.** Length. We can now define the length of a string:

$$
\begin{array}{rcl}
length(\epsilon) & = & 0 \\
length(w \cdot c) & = & length(w) + 1
\end{array}
$$

Note how the definition of $length$ parallels the structure of the definition of a string. During the term, we'll see that when we have a function that takes a string as an argument, the definition of the function usually has this form. More generally, when we have a function that takes an argument from a set that we've defined inductively, the definition of the function will typically have the same structure as the definition of the set.

**2.** Ordering. We'll say that two strings are equal if they consist of the same seqence of symbols. Written mathematically, we get:

$$\begin{array}{rcll} x = y & \leftrightarrow & & (x = \epsilon) \wedge (y = \epsilon) \\ & & \vee & (x = u \cdot c) \wedge (y = v \cdot d) \wedge (u = v) \wedge (c = d) \end{array}$$

If the elements of $\Sigma$ are ordered, then we can use that order to define an ordering of the elements of $\Sigma^*$. For $x$ and $y$ in $\Sigma^*$, we'll say that $x < y$ iff

$$\begin{array}{ll} & length(x) < length(y) \\ \vee & (length(x) = length(y)) \wedge (x = c \cdot u) \wedge (y = d \cdot v) \wedge (c < d) \\ \vee & (length(x) = length(y)) \wedge (x = c \cdot u) \wedge (y = c \cdot v) \wedge (u < v) \end{array}$$

Note how the definitions for length and ordering follow the inductive definition of strings.
The ordering defined above is called the *lexigraphical ordering* and is similar to the ordering used in the dictionary. Here's the difference. In our lexigraphical ordering, shorter strings occur before longer ones. Thus, "zebra" comes before "aardvark". This is because we allow strings of arbitrarily long length. If we use dictionary order, our "dictionary" would start: a, aa, aaa, aaaa, aaaaa, ..., and we'd never get to anything with a "b" (or anything else other than "a"s) in it.

**III.** An example.

Let's combine strings and induction. Let $\Sigma = \{0, 1\}$. Consider the set $S \subseteq \Sigma^*$, such that $w$ is in $S$ iff

$w = \epsilon$; or

There is a string $x$ in $S$ such that $w = 0x1$ or $w = 1x0$; or

There are strings $x$ and $y$ in $S$ such that $w = xy$.

Prove that $w$ is in $S$ iff the number of $O$'s in $w$ is equal to the number of 1's.

We'll work this out in class on the whiteboard, and a complete version of the proof should appear at the end of the Sept. 11 notes.