Math 202 notes

Jason Riedy

22 September, 2008

Contents

Notes also available as PDF.

What we will cover from Chapter 4:

- Numbers and digits in different bases, with historical context
- Arithmetic, digit by digit

And additionally, I'll give a brief summary of computer arithmetic.

1 Positional Numbers

A number is a concept and not just a sequence of symbols. We will be discussing ways to express numbers.

Before our current form of expressing *cardinal* numbers:

- Piles of rocks don't work well for merchants.
- **Marks** on sticks, then marks on papyrus.

Marking numbers is costly. A large number becomes a large number of marks. Many marks lead to many errors. Merchants don't like errors. So people started using symbols rather than plain marks.

An intermediate form, **grouping**:

- Egyptian: Different symbols for different levels of numbers: units, tens, hundreds. Grouping within the levels.
- Roman: Symbols for groups, with addition and subtraction of symbols for smaller groups.
- Greek (and Hebrew and Arabic): Similar, but using all their letters for many groups.
- Early Chinese: Denote the number of marks in the group with a number itself...

Getting better, but each system still has complex rules. The main problems are with skipping groups. We now use zero to denote an empty position, but these systems used varying amounts of space. Obviously, this could lead to trade disagreements. Once zeros were adopted, many of these systems persisted in trade for centuries.

Now into forms of positional notation, shorter and more direct:

- Babylonian:
  - Two marks, tens and units.
  - Now the marks are placed by the number of 60s.
  - Suffers from complicated rules about zeros.
  - (Using 60s persists for keeping time...)
- Mayan:
  - Again, two kinds of marks for fives and units.
  - Two positional types: by powers of 20, and by powers of 20 except for one power of 18.
  - (Note that $18 \cdot 20 = 360$, which is much closer to a year.)
  - Essentially equivalent to what we use, but subtraction in Mayan is much easier to see.
- (many other cultures adopted similar systems (*e.g.* Chinese rods)

Current: **Hindu-Arabic numeral system**

The characters differ between cultures, but the idea is the same. The characters often are similar as well. Originated in the region of India and was carried west through trade. No one knows when zero was added to the digits. The earliest firm evidence is in Arab court records regarding a visitor from India and a description of zero from around 776 AD. The first inscription found with a zero is from 876 AD in India. However, the Hindu-Arabic system was not adopted outside mathematics even in these cultures. Merchants kept to a system similar to the Greek and Hebrew systems using letters for numbers.

Leonardo Fibonacci brought the numerals to Europe in the 13th century (after 1200 AD) by translating an Arabic text to Latin. By 15th century, the numeral system was in wide use in Europe. During the 19th century, this system supplanted the rod systems in Asia.

The final value of the number is based on the positions of the digits:

$$1234 = 1 \cdot 10^3 + 2 \cdot 10^2 + 3 \cdot 10^1 + 4 \cdot 10^0.$$

We call ten the **base**. Then numbers becomes polynomials in terms of the base b,

$$1234 = b^3 + 2 \cdot b^2 + 3 \cdot b^1 + 4.$$

Here $b = 10$.

So we moved from marks, where 1000 would require 1000 marks, to groups, where 1000 may be a single mark but 999 may require dozens of marks. Then we moved to positional schemes where the number of symbols depends on the *logarithm* of the value; $1000 = 10^3$ requires $4 = 3 + 1$ symbols.

After looking at other bases, we will look into operations (multiplication, addition, *etc.*) using the base representations.

2 Converting Between Bases

Only three bases currently are in wide use: base 10 (decimal), base 2 (binary), and base 16 (hexadecimal). Occasionally base 8 (octal) is used, but that is increasingly rare. Other conversions are useful for practice and for seeing some structure in numbers. The structure will be useful for computing.

Before conversions, we need the digits to use. In base b, numbers are expressed using digits from 0 to $b - 1$. When b is past 10, we need to go beyond decimal numerals:

Value:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Digit:	0	1	2	3	4	5	6	7	8	9	**A**	**B**	**C**	**D**	**E**	**F**

Upper- and lower-case are common.

So in hexadecimal, DECAFBAD is a perfectly good number, as is DEADBEEF. If there is a question of what base is being used, the base is denoted by a subscript. So 10_{10} is a decimal ten and 10_2 is in binary.

To find values we recognize more easily, we convert to decimal. Then we will convert *from* decimal.

2.1 Converting to Decimal

Converting to decimal using decimal arithmetic is straight-forward. Remember the expansion of 1234 with base $b = 10$,

$$\begin{aligned}1234 &= 1 \cdot 10^3 + 2 \cdot 10^2 + 3 \cdot 10^1 + 4 \cdot 10^0 \\ &= b^3 + 2 \cdot b^2 + 3 \cdot b^1 + 4.\end{aligned}$$

Each digit of DEAD has a value, and these values become the coefficients. Then we expand the polynomial with $b = 16$. In a straight-forwart way,

$$\begin{aligned}\mathsf{DEAD} &= \mathsf{D} \cdot 16^3 + \mathsf{E} \cdot 16^2 + \mathsf{A} \cdot 16^1 + \mathsf{D} \\ &= 13 \cdot 16^3 + 14 \cdot 16^2 + 10 \cdot 16 + 13 \\ &= 13 \cdot 4096 + 14 \cdot 256 + 10 \cdot 16 + 13 \\ &= 57005.\end{aligned}$$

We an use **Horner's rule** to expand the polynomial in a method that often is faster,

$$\begin{aligned}\mathsf{DEAD} &= ((13 \cdot 16 + 14) \cdot 16 + 10) \cdot 16 + 13 \\ &= (222 \cdot 16 + 10) \cdot 16 + 13 \\ &= 3562 \cdot 16 + 13 \\ &= 57005.\end{aligned}$$

Let's try a binary example. Convert 1101_2 to decimal:

$$\begin{aligned}1101_2 &= (((1 \cdot 2 + 1) \cdot 2 + 0) \cdot 2 + 1 \\ &= (3 \cdot 2 + 0) \cdot 2 + 1 \\ &= 6 \cdot 2 + 1 \\ &= 13.\end{aligned}$$

Remember the rows of a truth table for two variables? Here,

$$\begin{aligned}11_2 &= 2 + 1 = 3, \\ 10_2 &= 2 + 0 = 2, \\ 01_2 &= 0 + 1 = 1, \text{ and} \\ 00_2 &= 0 + 0 = 0.\end{aligned}$$

2.2 Converting from Decimal

To convert to binary from decimal, consider the previous example:

$$\begin{aligned}13 &= 8 + \mathbf{5}\\ &= 8 + 4 + \mathbf{1}\\ &= 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0\\ &= 1101_2.\end{aligned}$$

At each step, we find the largest power of two less than the remaining number. Another example for binary:

$$\begin{aligned}293 &= 256 + \mathbf{37}\\ &= 256 + 32 + \mathbf{5}\\ &= 256 + 32 + 4 + 1\\ &= 1 \cdot 2^8 + 1 \cdot 2^5 + 1 \cdot 2^2 + 1\\ &= 100100101_2.\end{aligned}$$

And in hexadecimal,

$$\begin{aligned}293 &= 256 + \mathbf{37}\\ &= 1 \cdot 256 + 2 \cdot 16 + 5\\ &= 125_{16}.\end{aligned}$$

You can see why some people start remembering powers of two.

If you have no idea where to start converting, remember the relations $b^{\log_b x} = x$ and $\log_b x = \log x / \log b$. Rounding $\log_b x$ up to the larger whole number gives you the number of base b digits in x.

The text has another version using remainders. We will return to that in the next chapter. And conversions to and from binary will be useful when we discuss how computers manipulate numbers.

3 Operating on Numbers

Once we split a number into digits (decimal or binary), operations can be a bit easier.

We will cover multiplication, addition, and subtraction both

- to gain familiarity with positional notation, and
- to compute results more quickly and mentally.

Properties of positional notation will help when we explore number theory.

We will use two properties frequently:

- Both multiplication and addition **commute** ($a + b = b + a$) and re-**associate** $(a + b) + c = a + (b + c)$.
- Multiplication **distributes** over addition, so $a(b + c) = ab + ac$.
- Multiplying powers of a common base adds exponents, so $b^a \cdot b^c = b^{a+c}$.

3.1 Multiplication

Consider multiplication. I once had to learn multiplication tables for 10, 11, and 12, but these are completely pointless.

Any decimal number multiplied by 10 is simply shifted over by one digit,

$$\begin{aligned}
123 \cdot 10 &= (1 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0) \cdot 10 \\
&= 1 \cdot 10^3 + 2 \cdot 10^2 + 3 \cdot 10^1 \\
&= 1230.
\end{aligned}$$

Multiplying by $11 = 1 \cdot 10 + 1$ is best accomplished by adding the other number to itself shifted,

$$123 \cdot 11 = 123 \cdot (10 + 1) = 1230 + 123 = 1353.$$

And for $12 = 1 \cdot 10 + 2$, you double the number,

$$123 \cdot 12 = 123 \cdot (10 + 2) = 1230 + 246 = 1476.$$

Multiplying longer numbers quickly follows the same pattern of shifting and adding. We can expand $123 \cdot 123 = 123 \cdot (1 \cdot 10^2 + 2 \cdot 10 + 3)$ to

$$\begin{array}{r}
123 \\
\times\ 123 \\
\hline
369 \\
2460 \\
12300 \\
\hline
15129
\end{array}$$

Another method expands the product of numbers as a product of polynomials, working one term at a time. This is essentially the same but not in tabular form:

$$\begin{aligned}
123 \cdot 123 &= (1 \cdot 10^2 + 2 \cdot 10 + 3) \cdot (1 \cdot 10^2 + 2 \cdot 10 + 3) \\
&= (1 \cdot 10^2 + 2 \cdot 10 + 3) \cdot (1 \cdot 10^2 + 2 \cdot 10) + (1 \cdot 10^2 + 2 \cdot 10 + 3) \cdot 3 \\
&= (1 \cdot 10^2 + 2 \cdot 10 + 3) \cdot (1 \cdot 10^2 + 2 \cdot 10) + (3 \cdot 10^2 + 6 \cdot 10 + 9) \\
&= \ldots
\end{aligned}$$

This form splits the sums apart as well; we will cover that next.

Bear in mind that short-term memory is limited to seven to eight pieces of information. Structure mental arithmetic to keep as few pieces in flight as possible. One method is to break multiplication into stages. In long form, you can group the additions. For example, expanding $123 \cdot 123 = 123 \cdot (1 \cdot 10^2) + (123 \cdot 23) = 123 \cdot (1 \cdot 10^2) + (123 \cdot 2 \cdot 10 + 123 \cdot 3)$,

$$\begin{array}{r} 123 \\ \times\ 123 \\ \hline 369 \\ 2460 \\ \hline 2829 \\ 12300 \\ \hline 15129 \end{array}$$

Assuming a small number uses only one slot in your short-term memory, need track only where you are in the multiplier, the current sum, the current product, and the next sum. That leaves three to four pieces of information to use while adding.

One handy trick for 15% tips: divide by ten, divide that amount by two, and add the pieces. We can use positional notation to demonstrate how that works,

$$\begin{aligned} x \cdot 15\% &= (x \cdot 15)/100 \\ &= ((x \cdot (10 + 5))/100 \\ &= ((x \cdot 10) + (x \cdot (10/2)))/100 \\ &= x/10 + (x/10)/2 \end{aligned}$$

3.2 Addition

Digit-by-digit addition uses the commutative and associative properties:

$$\begin{aligned} 123 + 456 &= (1 \cdot 10^2 + 2 \cdot 10 + 3) + (4 \cdot 10^2 + 5 \cdot 10 + 6) \\ &= (1 + 4) \cdot 10^2 + (2 + 5) \cdot 10 + (3 + 6) \\ &= 579. \end{aligned}$$

Naturally, when a digit threatens to roll over ten, it **carries** to the next digit. Expanding the positional notation,

$$\begin{aligned} 123 + 987 &= (1 \cdot 10^2 + 2 \cdot 10 + 3) + (9 \cdot 10^2 + 8 \cdot 10 + 7) \\ &= (1 + 9) \cdot 10^2 + (2 + 8) \cdot 10 + (3 + 7) \\ &= 10 \cdot 10^2 + 10 \cdot 10 + 10. \end{aligned}$$

Because the coefficients are greater than $b - 1 = 9$, we expand those coefficients. Commuting and reassociating,

$$\begin{aligned}
123 + 987 &= 10 \cdot 10^2 + 10 \cdot 10 + 10 \\
&= (1 \cdot 10 + 0) \cdot 10^2 + (1 \cdot 10 + 0) \cdot 10 + (1 \cdot 10 + 0) \\
&= 1 \cdot 10^3 + 1 \cdot 10^2 + 1 \cdot 10 + 0 \\
&= 1110.
\end{aligned}$$

However, when working quickly, or when the addition will be used in another operation, you do not need to expand the carries immediately. This is called a **redundant representation** because numbers now have multiple representations. You can represent 13 as $1 \cdot 10 + 3$ or simply as 13.

If you work that way mentally, you need to keep the intermediate results in memory. So during multiplying, you only need to work out the carries every three to four digits...

3.3 Subtraction

In systems with signed numbers, we know that subtracting a number is the same as adding its negation: $a - b = a + (-b)$. So we expect the digit-by-digit method to work with each digit subtracted, and it does. Because $-a = -1 \cdot a$, we can distribute the sign over the digits:

$$\begin{aligned}
456 - 123 &= (4 \cdot 10^2 + 5 \cdot 10 + 6) - (1 \cdot 10^2 + 2 \cdot 10 + 3) \\
&= (4 \cdot 10^2 + 5 \cdot 10 + 6) + (-(1 \cdot 10^2 + 2 \cdot 10 + 3)) \\
&= (4 \cdot 10^2 + 5 \cdot 10 + 6) + (-1 \cdot 10^2 + -2 \cdot 10 + -3) \\
&= (4 - 1) \cdot 10^2 + (5 - 2) \cdot 10 + (6 - 3) \\
&= 333.
\end{aligned}$$

As with carrying, **borrowing** occurs when a digit goes negative:

$$\begin{aligned}
30 - 11 &= (3 \cdot 10^1 + 0) - (1 \cdot 10^1 + 1) \\
&= (3 - 1) \cdot 10^1 + (0 - 1) \\
&= 2 \cdot 10^1 + -1 \\
&= 1 \cdot 10^1 + (10 - 1) \\
&= 1 \cdot 10^1 + 9 \\
&= 19.
\end{aligned}$$

Again, you can use a redundant intermediate representation of $2 \cdot 10^1 - 1$ if you're continuing to other operations. And if **all** the digits are negative, you can factor out -1,

$$\begin{aligned}123 - 456 &= (1 \cdot 10^2 + 2 \cdot 10 + 3) - (4 \cdot 10^2 + 5 \cdot 10 + 6)\\ &= (1-4) \cdot 10^2 + (2-5) \cdot 10 + (3-6)\\ &= (-3) \cdot 10^2 + (-3) \cdot 10 + (-3)\\ &= -(3 \cdot 10^2 + 3 \cdot 10 + 3)\\ &= -333.\end{aligned}$$

3.4 Division and Square Root: Later

We will cover these later with number theory.

4 Homework

- Problem Set 3.1 (p154)
  - 8, 16
  - 32, 34, 35
- Problem Set 3.2 (p161)
  - 3, 4, 5, 6, 8, 11, 20
- Problem Set 3.3 (p177)
  - 10, 12, 20 (the zeros are indicators; find the *least* base where the sum is correct)
- Problem Set 3.4 (p188)
  - 5
  - 17 (this is Napier's method; he made physical rods to accelerate the process)
  - 19, 33