Number representation lecture summary

M. George

November 5, 2014

1 Division

We proved that you can divide integers:

Claim: For any integers $a, b > 0$, there exists q, r such that $0 \leq r < b$ and $a = qb + r$. You are more used to seeing this as $a/b = q + r/b$. q is called the quotient and is written $q \div r$, and r is called the remainder and is written $q \bmod r$.

Note: It's a good exercise to show that q and r are unique.

Proof: by induction on a. Base case: when $a = 0$, choose $q = 0$ and $r = 0$. Then clearly $a = qb + r$ and moreover $0 \leq r < b$ (since $b > 0$ by assumption).

For the inductive step, we wish to prove that there exists q and r such that $a + 1 = qb + r$. By the inductive hypothesis, we know that we can write a as $a = q'b + r'$. There are two cases: if $r' < b - 1$, then we can choose $q = q'$ and $r = r' + 1$. Then $0 < r < b$, and $qb + r = q'b + r' + 1 = a + 1$. If, on the other hand, $r' = b - 1$, then we can choose $q = q' + 1$ and $r = 0$. We have

$$\begin{aligned} qb + r &= (q' + 1)b + 0 \\ &= q'b + b + (b - 1) - (b - 1) \\ &= q'b + (b - 1) + 1 = q'b + r' + 1 = a + 1 \end{aligned}$$

as required.

2 Number bases

We discussed writing numbers in different bases. A number base is a rule for interpreting a string of digits as a number (an element of $\mathbb{N}$).

We are most used to decimal, or base 10. When I write the string "12476", you interpret this as "1 ten-thousand plus 2 thousands plus 4 hundreds plus 7 tens and 6 ones". More generally, we interpret the string $a_k a_{k-1} \cdots a_2 a_1 a_0$ as $a_k 10^k + a_{k-1} 10^{k-1} + \cdots + a_2 10^2 + a_1 10^1 + a_0 10^0$.

In computer science it is often helpful to write down numbers in different bases. For example, if we wish to work in octal (base 8), then we interpret strings of digits using the same formula as above, but with powers of 8 instead of 10. For example, we would interpret the string "12476_{oct}" as the number $1 \cdot 8^4 + 2 \cdot 8^3 + 4 \cdot 8^2 + 7 \cdot 8 + 6 \cdot 1 = 5431_{dec}$.

The reason this is useful is because it makes division (and remainder) easy. For example, in base 10, it is easy to divide by powers of 10. For example, you can tell by inspection that "$12476 \div 100 = 124$" while "$12476 \mod 100 = 76$". However doing division by non-powers of ten requires long division, which takes time and effort. For example, you probably can't work out "$12476 \div 64$" in your head.

In base b however, division by a power of b is as easy as dividing by powers of 10 in decimal. For example, $12476_{oct} \div 100_{oct} = 124_{oct}$[1]. Since division by powers of 2 is common in many data structures, writing numbers in base 2 (binary), base 8 (octal) and base 16 (hexadecimal) allows for easy division in your head (or in hardware).

3 Uniqueness

Division is unique: Given n, d, if there are two quotient and remainder pairs then the quotients and remainders must be the same. We are thus justified in talking about "the" quotient and "the" remainder.

Proof: assume $n = qd + r = q'd + r'$, and $0 \leq r < d$, and $0 \leq r' < d$. Then $(q - q')d + (r - r') = 0$. Since both r and r' are between 0 and d, we know that $-d < (r - r') < d$. This implies that $-d < (q - q')d < d$, so $-1 < q - q' < 1$. Since $q - q'$ is an integer, it must be 0. Thus $q = q'$. From this it follows immediately that $r = r'$.

Conversion between bases is also unique by applying a completely analogous argument inductively.

[1] note that 12467_{oct} is a completely different number than 12467_{dec}