Learning Objectives

By the end of this worksheet, you will:

- Understand how to represent natural numbers in different bases and convert between representations in different bases.
- Represent fractional numbers in binary.

---

1. **Representing the natural numbers.** Recall that we can represent the natural numbers using decimal (base 10) and binary (base 2); we can generalize this to arbitrary bases. Let $b \in \mathbb{Z}^+$, and suppose $b \geq 2$. For $k \in \mathbb{Z}^+$ and $d_0, d_1, \ldots, d_{k-1} \in \{0, 1, \ldots, b-1\}$, the expression $(d_{k-1}d_{k-2}\ldots d_1 d_0)_b$ represents the number

$$x = \sum_{i=0}^{k-1} d_i \times b^i$$

Now we are going to look at two specific bases that are useful in computer science: octal (base 8) and hexadecimal (base 16).

(a) Octal numbers are base 8. For example, $(11)_8$ represents the number 9. What number does $(165)_8$ represent?[1]

Solution

117

(b) Hexadecimal numbers are base 16. Since we only have the digits 0-9, hexadecimal uses letters to stand in for 10-15: A = 10, B = 11, and so on. For example, $(1F)_{16}$ is the hexadecimal representation of the number 31. What number does $(B4)_{16}$ represent?[2]

Solution

180

2. **Converting between bases.** Since binary, octal, decimal, and hexadecimal number representations frequently come up in computer science, it is useful to know how to convert between them.

We will first look at converting from decimal to binary and then consider converting to octal or hex from there. One way to convert a decimal number into binary is to repeatedly divide it by two. The remainder from each division is then a bit in the binary representation and we continue with the quotient.

For example, if we want to convert 29 into binary, we proceed as follows:

$$\begin{aligned} 29 \div 2 &= 14, \quad \text{remainder } \mathbf{1} \\ 14 \div 2 &= 7, \quad \text{remainder } \mathbf{0} \\ 7 \div 2 &= 3, \quad \text{remainder } \mathbf{1} \\ 3 \div 2 &= 1, \quad \text{remainder } \mathbf{1} \\ 1 \div 2 &= 0, \quad \text{remainder } \mathbf{1} \end{aligned}$$

The binary representation of 29 is the sequence of remainders read *bottom to top*: $29 = (11101)_2$.

(a) Find the binary representation of 357.

[1] In Python, you can input numbers in octal format by putting `0o` in front of the octal representation. For example, `0o11` is Python's octal representation of the number 9.

[2] In Python, you can input numbers in hexadecmial format by putting `0x` in front of the representation. For example, `0x1F` is Python's hexadecimal representation of the number 31.

Solution

$357 = (101100101)_2$

(b) Now let's look at converting from binary to octal. Since $8 = 2^3$, every three binary bits correspond to one octal digit (starting from the right). For example, $(111\ 110)_2 = (76)_8$ (we used a space to divide the binary representation into groups of three bits).

Using your answer to the previous part, find the octal representation of 357.

Solution

$357 = (101\ 100\ 101)_2 = (545)_8$

(c) Using the same idea, find the hexadecimal representation of 357. (Hint: $16 = 2^4$.)

Solution

$357 = (0001\ 0110\ 0101)_2 = (165)_{16}$

3. **Representing fractions.** We have previously considered how we can represent the natural numbers using different bases. What if we want to represent real numbers instead? It turns out we can approach it in the same way. In this exercise, we'll consider how to represent a number between zero and one.

Let $b \in \mathbb{Z}^+$, and suppose $b \geq 2$. For $k \in \mathbb{Z}^+$ and $d_1, \ldots, d_{k-1} \in \{0, 1, \ldots, b-1\}$, the expression $(0.d_1 d_2 \ldots d_k)_b$ represents the number

$$x = \sum_{i=1}^{k} d_i \times b^{-i} = \sum_{i=1}^{k} \frac{d_i}{b^i}$$

How do we convert fractional numbers from decimal to binary? The conversion algorithm is essentially the same as the one for natural numbers, but we instead *multiply by 2*: if the result goes over 1, we record a 1 and keep going with the fractional part.

For example, here is how we can convert 0.8125 to binary:

$$\begin{aligned} 0.8125 \times 2 &= 0.625 \ + \mathbf{1} \\ 0.625 \times 2 &= 0.25 \ + \mathbf{1} \\ 0.25 \times 2 &= 0.5 \ + \mathbf{0} \\ 0.5 \times 2 &= 0 \ + \mathbf{1} \\ 0 & \end{aligned}$$

We are done when we reach 0. The binary representation is now the sequence of whole numbers on the right-hand side, read *top to bottom*, placed after the decimal point. So $0.8125 = (0.1101)_2$.

(a) Find the binary representation of 0.375. Explicitly write out the sum to verify your answer.

Solution

$0.375 = (0.011)_2$. We can check this by computing $0 \times 0.5 + 1 \times 0.25 + 1 \times 0.125 = 0.375$.

Even though the last example worked out cleanly, this conversion process doesn't always reach zero! Just like decimal representation, binary representations sometimes have repeating bits after the decimal point. For example, $\frac{1}{3}$ has representation $0.\overline{3}$ in decimal, where the overline indicates that the 3 repeats. This is true in binary as well: $\frac{1}{3} = (0.\overline{01})_2$. Let's see how we would obtain this using the algorithm we learned at the start of this section:

$$\begin{aligned}\frac{1}{3} \times 2 &= \frac{2}{3} + \mathbf{0}\\ \frac{2}{3} \times 2 &= \frac{1}{3} + \mathbf{1}\\ \frac{1}{3} \times 2 &= \ldots\end{aligned}$$

After the second step, we've returned to $\frac{1}{3}$, and the sequence 01 we obtained up to this point repeats.

(b) Find the binary representation of $\frac{1}{10}$.[3]
Pay attention to when the pattern starts to repeat—it's not the entire sequence of bits!

Solution

$$\begin{aligned}0.1 \times 2 &= 0.2 + \mathbf{0}\\ \mathbf{0.2} \times 2 &= 0.4 + \mathbf{0}\\ 0.4 \times 2 &= 0.8 + \mathbf{0}\\ 0.8 \times 2 &= 0.6 + \mathbf{1}\\ 0.6 \times 2 &= 0.2 + \mathbf{1}\\ \mathbf{0.2} \times 2 &= \ldots\end{aligned}$$

So the binary representation is $0.1 = (0.0\overline{0011})_2$.
Note that what repeats is the sequence of values $[0.2, 0.4, 0.8, 0.6]$, and so the four bits corresponding to those steps repeat as well. The first bit 0, corresponding to the first step for 0.1, doesn't repeat.

[3]You should find that this has an infinitely repeating binary representation, even though its decimal representation requires just one decimal place! Since computers have a finite amount of memory, this infinite sequence of bits needs to be truncated, and so the 0.1 stored in a Python program isn't exactly 0.1, but it is close.

4. **Infinite geometric series.**

Given an infinite repeating binary representation, how can we verify that it is correct? To do this, we need to be able to sum an infinite geometric series, using the following formula (valid for all $a, r \in \mathbb{R}$ when $|r| < 1$):[4]

$$\sum_{i=1}^{\infty} ar^i = \frac{ar}{1-r}$$

(a) As a warm-up, use the formula to verify that $(0.\overline{1})_2 = 1$. You'll need to determine the appropriate values for a and r; writing out the first few terms in the summation may be helpful.

Solution

$$\begin{aligned}
(0.\overline{11})_2 &= \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots \\
&= \sum_{i=1}^{\infty} (\frac{1}{2})^i \\
&= \frac{\frac{1}{2}}{1-\frac{1}{2}} \\
&= 1
\end{aligned}$$

(b) Now, show that $(0.0\overline{0011})_2 = \frac{1}{10}$. This is a bit more complex: to find the a and r, write out the first few terms in the summation, group the terms into pairs, and pull out common factors (powers of $\frac{1}{2^4}$).

Solution

$$\begin{aligned}
(0.0\overline{0011})_2 &= \left(\frac{1}{2^4} + \frac{1}{2^5}\right) + \left(\frac{1}{2^8} + \frac{1}{2^9}\right) + \left(\frac{1}{2^{12}} + \frac{1}{2^{13}}\right) \dots \\
&= \frac{1}{2^4}\left(1 + \frac{1}{2}\right) + \frac{1}{2^8}\left(1 + \frac{1}{2}\right) + \frac{1}{2^{12}}\left(1 + \frac{1}{2}\right) + \dots \\
&= \frac{1}{2^4} \cdot \frac{3}{2} + \left(\frac{1}{2^4}\right)^2 \cdot \frac{3}{2} + \left(\frac{1}{2^4}\right)^3 \cdot \frac{3}{2} + \dots \\
&= \sum_{i=1}^{\infty} \frac{3}{2}\left(\frac{1}{2^4}\right)^i \\
&= \frac{\frac{3}{2} \cdot \frac{1}{2^4}}{1-\frac{1}{2^4}} && \text{(where } a = \frac{3}{2},\ r = \frac{1}{2^4}\text{)} \\
&= \frac{\frac{3}{2}}{2^4 - 1} && \text{(multiplying by } \frac{2^4}{2^4}\text{)} \\
&= \frac{1.5}{15} \\
&= 0.1
\end{aligned}$$

[4] Note that this version starts with $i = 1$. If we started from $i = 0$, the right-hand side would be $\frac{a}{1-r}$ instead.