INFORMATION & COMPUTER SCIENCES
UNIVERSITY of HAWAIʻI at MĀNOA

Ch 4.2: Integer Representations and Algorithms

ICS 141: Discrete Mathematics for Computer Science I

KYLE BERNEY
DEPARTMENT OF ICS, UNIVERSITY OF HAWAII AT MANOA

Representations of Integers

- In everyday life, we use decimal numbers
  - Base 10
  - 10 total digits: $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

- Ex: Two hundred and nine, written 209, stands for

$$2 \cdot 10^2 + 0 \cdot 10^1 + 9 \cdot 10^0$$

- Ex: Four thousand one hundred and twenty nine, written 4129, stands for

$$4 \cdot 10^3 + 1 \cdot 10^2 + 2 \cdot 10^1 + 9 \cdot 10^0$$

Representations of Integers

- Computers use binary numbers
  - Base 2
  - 2 total digits: $\{0, 1\}$
- <u>Ex:</u> Twenty three, written 10111, stands for

$$1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$

- <u>Ex:</u> Thirty six, written 100100, stands for

$$1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0$$

Representations of Integers

- Theorem 1: (Basis Representation Theorem) Let b be an integer such that $b > 1$. For every positive integer n, there exists a unique representation

$$n = a_k b^k + a_{k-1} b^{k-1} + \cdots + a_1 b^1 + a_0 b^0$$

where $a_k \neq 0$ and each a_i is non-negative and strictly less than b.

- Known as the representation of n to the base (or radix) b
- Written as $n = (a_k a_{k-1} \ldots a_1 a_0)_b$
  - The subscript b can be omitted if it is clear what the base of the number is
- *Remark:* We can also represent 0 by letting all the $a_i = 0$

Historical Number Systems

- Babylonains
  - Sexagesimal numbers
  - Base 60
  - We derive modern-day usage of
    - 60 seconds in a minute
    - 60 minutes in an hour
    - 360 degrees in a circle
- Mayans
  - Vigesimal numbers
  - Base 20

Octal and Hexadecimal Representation

- Representing large numbers in binary can result in a large number of digits
  - Octal (base 8)
  - Hexadecimal (base 16)
- Allow for easy conversion from/into binary representation
  - Every 3 binary digits correspond to a single octal digit
  - Every 4 binary digits correspond to a single hexadecimal digit

Octal Representation

- Octal Numbers
  - Base 8
  - 8 total digits: $\{0, 1, 2, 3, 4, 5, 6, 7\}$

- Ex: Three hundred and seventy, written $(101110010)_2$,
  - $(101)_2 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 5$
  - $(110)_2 = 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 6$
  - $(010)_2 = 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 2$

$$(101110010)_2 = (562)_8$$

Octal Representation

- Octal Numbers
  - Base 8
  - 8 total digits: $\{0, 1, 2, 3, 4, 5, 6, 7\}$
- <u>Ex:</u> Two hundred and eighty two, written $(432)_8$,
  - $4 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = (100)_2$
  - $3 = 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = (011)_2$
  - $2 = 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = (010)_2$

$$(432)_8 = (100011010)_2$$

Hexadecimal Representation

- Hexadecimal Numbers
  - Base 16
  - 16 total digits: $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$
- Additional digits:
  - $A = 10$
  - $B = 11$
  - $C = 12$
  - $D = 13$
  - $E = 14$
  - $F = 15$

Hexadecimal Representation

- Hexadecimal Numbers
  - Base 16
  - 16 total digits: $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$

- <u>Ex:</u> Three thousand and fourteen, written $(101111000110)_2$,
  - $(1011)_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 11 = B$
  - $(1100)_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 12 = C$
  - $(0110)_2 = 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 6$

$$(101111000110)_2 = (BC6)_{16}$$

Hexadecimal Representation

- Hexadecimal Numbers
  - Base 16
  - 16 total digits: $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$
- Ex: Three thousand eight hundred and eighty two, written $(F2A)_{16}$,
  - $15 = 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = (1111)_2$
  - $2 = 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = (0010)_2$
  - $10 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = (1010)_2$

$$(F2A)_{16} = (111100101010)_2$$

Base Conversion

- To convert an integer n into an arbitrary base b:
  - Successively divide quotients by b
  - Each remainder is a (right-most) digit of the base b representation

Base Conversion

- To convert an integer n into an arbitrary base b:
  - Successively divide quotients by b
  - Each remainder is a (right-most) digit of the base b representation

- Ex: Convert 12345 into octal (base 8)

1. $12345 = 8 \cdot 1543 + 1$
2. $1543 = 8 \cdot 192 + 7$
3. $192 = 8 \cdot 24 + 0$
4. $24 = 8 \cdot 3 + 0$
5. $3 = 8 \cdot 0 + 3$

$$12345 = (30071)_8$$

Base Conversion

- To convert an integer n into an arbitrary base b:
  - Successively divide quotients by b
  - Each remainder is a (right-most) digit of the base b representation

- Ex: Convert 177130 into hexadecimal (base 16)

1. $177130 = 16 \cdot 11070 + 10$
2. $11070 = 16 \cdot 691 + 14$
3. $691 = 16 \cdot 43 + 3$
4. $43 = 16 \cdot 2 + 11$
5. $2 = 16 \cdot 0 + 2$

$$177130 = (2B3EA)_{16}$$

Base Conversion

- To convert an integer n into an arbitrary base b:
  - Successively divide quotients by b
  - Each remainder is a (right-most) digit of the base b representation

```
BaseB(n, b)
    q = n
    k = 0
    while q ≠ 0
        a_k = q (mod b)
        q = q/b
        k = k + 1
    return (a_{k-1}, a_{k-2}, ..., a_1, a_0)        ▷ (a_{k-1} a_{k-2} ... a_1 a_0)_b
```

Binary Arithmetic

- Consider the problem of adding and multiplying two integers in binary
  - Works similarly to decimal additon and multiplication

Binary Arithmetic

- Consider the problem of adding and multiplying two integers in binary
  - Works similarly to decimal additon and multiplication

- Ex: Add 1110 and 1011

	c	c		
	1	1	1	0
+	1	0	1	1
1	1	0	0	1

Binary Arithmetic

- Consider the problem of adding and multiplying two integers in binary
  - Works similarly to decimal additon and multiplication
- Ex: Multiply 110 and 101

			1	1	0
		$\times$	1	0	1
			1	1	0
		0	0	0	
+	1	1	0		
	1	1	1	1	0