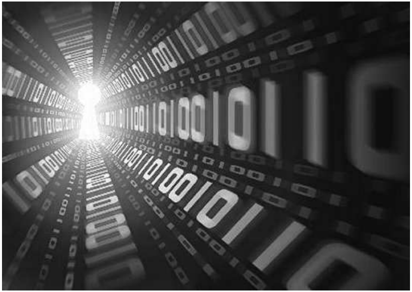
CSE100 Lecture02
Numbers, Representations, and Conversions
Introduction to Computer Systems

M.A.Hakim Newton

Computer Science and Engineering
Bangladesh University of Engineering and Technology
Dhaka – 1000, Bangladesh

CSE, BUET, 2009

Outline



Numbers and Symbols

Numbers as Symbols

- Numbers are used as data symbols.
- Numbers are used as program symbols.
- Numbers are used for any symbols.

Symbols as Numbers

- Symbols are used to represent Numbers.
- Symbol → Number → Symbol → Number
- Why are they so entangled with each other?

What is a Number?

Representation?

- A number is not just the symbol it is represented by.
- A number might have many symbols, many representations.
- 3, III, Three, and $\triangle$ refer to the same although look different.
- Same is hearing sound "three" or touching 3 things in dark.

Concept or Notion?

- Number is a notion; also, each number is a unique notion.
- We can feel the notions and use them appropriately.
- We just use various symbols to represent the notions.
- The notions remain the same, no matter what symbols we use.

Whole Numbers

Only Positive Whole Numbers?

- A countable (finite/infinite) set is mapped to whole numbers.
- An uncoutable set cannot be mapped to just whole numbers.
- Negative numbers can also be mapped to positive numbers.
- We are interested only in positive whole numbers and zero.

What about Fractions?

- Real numbers cannot be represented by whole numbers.
- There are infinite numbers between two real numbers.
- We have to truncate real numbers up to certain digits.
- Truncated numbers can be represented by whole numbers.

Whole Number Representation

Human Number Systems

- Tally: using just one symbol, for example cricket scoring.
- Roman: using a number of symbols, but not well organised.
- Arabic: using only ten symbols, very well organised and easy.

How Many Symbols?

- The set of whole numbers is infinite but countable.
- Do you need a distinct symbol for each whole number?
- What if the given set is finite in a particular case?
- In that case, you can use just a finite number of symbols?
- Can you generate infinite symbols from finite ones?

Number System

Uniform Weighted String: Standard

- "478321": Each digit in the number has exactly 10 times more weight than its immediate right digit; number of symbol 10.
- "12:34:56": Clock time is a number; hour, min, and sec are like digits. The uniform weight is 60; number of symbol 60.
- $S_{n-1} \ldots S_1 S_0$; uniform weight is r and number of symbol is r. Any S_k can be just one of the r symbols.

Non-uniform Weighted String: Non-standard

- 2009:04:20:11:15:30; current date and time is such a number.
- 15ml 5fr 200yd 2ft 9in; distance between two locations.

Uniform Weighted Number

$(S_{n-1}...S_1 S_0)_r$: length n, weight r, symbols r

- r: the number of symbols = weight = base = radix.
- S_0, S_{n-1}: respectively the least and most significant symbols.
- $S_k (0 \leq k < n)$ can assume any of the available r symbols.
- represents $(S_{n-1} \times r^{n-1} + \cdots + S_1 \times r^1 + S_0 \times r^0)$th string.

Standard Radices

- Decimal: $r = 10$; Symbols $= \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- Binary: $r = 2$; Symbols $= \{0, 1\}$
- Octal: $r = 8$; Symbols $= \{0, 1, 2, 3, 4, 5, 6, 7\}$
- Hexadecimal $r = 16$; Symbols $= \{0, \cdots, 9, A, B, C, D, E, F\}$

Number String Generation

Given 3 symbols A, B, C; generate $3^4 = 81$ strings of length 4

AAAA	AABA	AACA	ABAA	ABBA	ABCA	ACAA	ACBA	ACCA
AAAB	AABB	AACB	ABAB	ABBB	ABCB	ACAB	ACBB	ACCB
AAAC	AABC	AACC	ABAC	ABBC	ABCC	ACAC	ACBC	ACCC
BAAA	BABA	BACA	BBAA	BBBA	BBCA	BCAA	BCBA	BCCA
BAAB	BABB	BACB	BBAB	BBBB	BBCB	BCAB	BCBB	BCCB
BAAC	BABC	BACC	BBAC	BBBC	BBCC	BCAC	BCBC	BCCC
CAAA	CABA	CACA	CBAA	CBBA	CBCA	CCAA	CCBA	CCCA
CAAB	CABB	CACB	CBAB	CBBB	CBCB	CCAB	CCBB	CCCB
CAAC	CABC	CACC	CBAC	CBBC	CBCC	CCAC	CCBC	CCCC

How to write the strings – permutation

- Write length 1 strings – A B C
- Prepend one symbol before each of the length 1 string
  AA AB AC BA BB BC CA CB CC
- Prepend one symbol before each of the length 2 string
- Prepend one symbol before each of the length 3 string

Strings to Number

Assign numbers to each string

AAAA	AABA	AACA	ABAA	ABBA	ABCA	ACAA	ACBA	ACCA
AAAB	AABB	AACB	ABAB	ABBB	ABCB	ACAB	ACBB	ACCB
AAAC	AABC	AACC	ABAC	ABBC	ABCC	ACAC	ACBC	ACCC
BAAA	BABA	BACA	BBAA	BBBA	BBCA	BCAA	BCBA	BCCA
BAAB	BABB	BACB	BBAB	BBBB	BBCB	BCAB	BCBB	BCCB
BAAC	BABC	BACC	BBAC	BBBC	BBCC	BCAC	BCBC	BCCC
CAAA	CABA	CACA	CBAA	CBBA	CBCA	CCAA	CCBA	CCCA
CAAB	CABB	CACB	CBAB	CBBB	CBCB	CCAB	CCBB	CCCB
CAAC	CABC	CACC	CBAC	CBBC	CBCC	CCAC	CCBC	CCCC

Given a string BCAB, what is the number?

- Assume $A = 0$, $B = 1$, $C = 2$, BCAB means 1201
- By just counting BCAB stands for decimal 46
- Can you find 46 from 1201? Perhaps weight 3 would help.
- $1 \times 3^3 + 2 \times 3^2 + 0 \times 3^1 + 1 \times 3^0 = 46$

Different Base Strings

Decimal

00	01	02	03	04	05	06	07	08	09
	...			...			...		
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
	...			...			...		
90	91	92	93	94	95	96	97	98	99

Octal

00	01	02	03	04	05	06	07
10	11	12	13	14	15	16	17
	...			...		...	
40	41	42	43	44	45	46	47
50	51	52	53	54	55	56	57
60	61	62	63	64	65	66	67
70	71	72	73	74	75	76	77

Binary

000000	000001	000010	000011	000100	000101	000110	000111
	...			...		...	
100000	100001	100010	100011	100100	100101	100110	100111
101000	101001	101010	101011	101100	101101	101110	101111
110000	110001	110010	110011	110100	110101	110110	110111
111000	111001	111010	111011	111100	111101	111110	111111

Hexadecimal

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
		...					...					...			
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF

Equivalence of Representations

$(56)_o = (101110)_b = (2E)_h = (46)_d$

- $(46)_d = (4 \times 10^1 + 6 \times 10^0)_d$.
- $(56)_0 = (5 \times 8^1 + 6 \times 8^0)_d$.
- $(101110)_b = (1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 01 \times 2^0)_d$
- $(2E)_h = (2 \times 16^1 + 14 \times 16^0)_d$.

Excercises

- $(3675)_o = (\cdots)_d$
- $(3675)_h = (\cdots)_d$
- $(101011)_b = (\cdots)_d$
- $(10 : 45 : 53)_{60} = (\cdots)_d$

Minimum String Length

Number $(N)_d$, min Length $n =$? when Base r

- Base r and Length n mean r^n different strings.
- The range of numbers is $0, 1, \cdots, (r^n - 1)$.
- $(N)_d < (r^n)_d$, therefore, $n > \log_r N$

Exercises

- What is the range of numbers for $n = 6, r = 2$?
- What is the min string length n for $N = 1000, r = 2$?
- What is the range of numbers for $n = 3, r = 6$?
- What is the min string length n for $N = (657)_8, r = 2$?
- What is the min string length n for $N = (657)_9, r = 5$?

Number System Conversion

$(S_{n-1} \cdots S_1 S_0)_r = (N)_d$, any base to decimal

- $N = S_{n-1} \times r^{n-1} + \cdots + S_1 \times r^1 + S_0 \times r^0$.
- $(1011)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = (11)_d$

$(N)_d = (S_{n-1} \cdots S_1 S_0)_r$, decimal to any base

- Assume $R_0 = (N)_d$
- $S_0 = R_0 \% r, R_1 = R_0 / r$
- $S_1 = R_1 \% r, R_2 = R_1 / r$
- $\cdots\cdots\cdots$
- $S_k = R_k \% r, R_{k+1} = R_k / r$

r	R	S
	11	
2	5	1
2	2	1
2	1	0
2	0	1

Number System Conversion ...

Any source base to any destination base

- Convert from the source base to decimal first.
- Convert from decimal to the destination base.

Exercises

- Convert $(46)_d$ to base 2.
- Convert $(46)_d$ to base 3.
- Convert $(56)_o$ to hexadecimal.
- Convert $(2E)_h$ to octal.
- Convert $(46)_7$ to base 5.

Some Shortcut Methods

Binary to Octal

- Group 3 bits at a time from right to left.
- Pad with 0s at the left whenever needed.
- Replace the octal digit for each 3-bit group.
- $(10101110)_b = (010\ 101\ 110)_b = (256)_o$

Binary to Hexadecimal

- Group 4 bits at a time from right to left.
- Pad with 0s at the left whenever needed.
- Replace the hexadecimal digit for each 4-bit group.
- $(10101110)_b = (1010\ 1110)_b = (AE)_h$

Some Shortcut Methods ...

Octal to Binary

- Replace each digit by the binary in 3 bits.
- $(256)_o = (010\ 101\ 110)_b = (10101110)_b$

Hexadecimal to Binary

- Replace each digit by the binary in 4 bits.
- $(AE)_h = (1010\ 1110)_b = (10101110)_b$

Octal to Hexadecimal and vice versa

- From the given base to the binary then to the desired base.

Fractional Numbers

$(0.S_{-1}S_{-2}\cdots S_{-n})_r = (N)_d$, any base to decimal

- $N = S_{-1} \times r^{-1} + S_{-2} \times r^{-2} + \cdots + S_{-n} \times r^{-n}$
- $(0.011)_2 = 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} = (0.375)_d$
- $(101.011)_2 =?$ and $(201.021)_3 =?$

$(N)_d = (0.S_{-1}S_{-2}\cdots S_{-n})_r$, decimal to any base

- Assume $R_{-1} = (N)_d$
- $S_{-1} = \mathrm{int}(R_{-1} \times r), R_{-2} = \mathrm{frac}(R_{-1} \times r)$
- $S_{-2} = \mathrm{int}(R_{-2} \times r), R_{-3} = \mathrm{frac}(R_{-2} \times r)$
- $S_{-k} = \mathrm{int}(R_{-k} \times r), R_{-k-1} = \mathrm{frac}(R_{-k})/r$

b	R	S
	0.375	
2	0.750	0
2	0.50	1
2	0.0	1

Negative Numbers

How do we represent?

- We normally use different signs such as $+$ and $-$.
- This means an extra symbol – should not be allowed.
- We should somehow use only the given r symbols.
- In binary, 0 and 1 could represent $+$ and $-$ respectively.

Binary Negative Numbers

- Computers support only binary numbers.
- $+$ and $-$ Signs can be represented by 0 and 1.
- There are still different ways to represent.
- Signed-Magnitude, 1's complement, 2's complement, etc.

Binary Negative Numbers

Signed-Magnitude: Use MSB for sign, rests for magnitude

- 0000 0001 0010 0011 0100 0101 0110 0111
- 1000 1001 1010 1011 1100 1101 1110 1111
- MSB indicates sign, the rest 3 bits for magnitude
- Determine the range, how many zeros, add and subtract

1's Complement: Flip bits, MSB - sign, flip for magnitude

- 0000 0001 0010 0011 0100 0101 0110 0111
- 1111 1110 1101 1100 1011 1010 1001 1000
- How many zeros? Add and subtract and note problems.
- Perform $6 - 4, 6 + (-4), 4 - 6, 4 + (-6), 4 - 4, 4 + (-4)$.

Binary Negative Numbers . . .

2's Complement: 1's Complement + 1

- 0000 0001 0010 0011 0100 0101 0110 0111
- 1111 1110 1101 1100 1011 1010 1001 1000
- Keep unchanged until first 1 from LSB, flip all others
- How many zeros? What is the range now?
- Perform $6 - 4, 6 + (-4), 4 - 6, 4 + (-6), 4 - 4, 4 + (-4)$.
- Observe $8, -8, -(-8)$ and comment on these.

2's Complement: Computers' Number System

- This is the mostly used number system in computers.
- Unique string for each number, convenient arithmetic.

Conclusion

Remarks

- Computers use only binary number representations
- Computers use the 2's complement number system.

References

- From Digital Logic Design: by Morris Mano.
- Number Systems: resources are abdundantly available
- www.buet.ac.bd/cse/users/faculty/newton/teaching.html

Questions

Please ask any questions that you might have.