enisova

Test I Study Sheet

Sept 8, 2022

Binary - base 2
- computers think & communicate in binary
- 0 = false/off, 1 = true/on

Decimal - base 10, digits 0-9

Hexadecimal - base 16
- 2 digits can represent values from 0-255
  - ↳ same as 8 bits / 1 byte of binary

Bit - a single binary digit → 0110 0111

Byte - 8 bits of data → 0110 0111
- 256 different combinations ☆
- max val - 255, min val - 0

Binary → Decimal ~ each power of 2 on/off

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

e.g. 0001 1110 → 16+8+4+2 = 30 //
1111 1111 → 255 (all are on & max is 255)
0010 0000 → 32 (one is on)
0011 1111 → 64 (everything on BEFORE 64)

Decimal → Binary ~ find largest power of 2 < decimal value
→ turn on corresponding pos, subtract & repeat

(128 64 32 16 8 4 2 1)
e.g. 24 → 0001 1000 $\quad 24 - 16 = 8 \quad 8 - 8 = 0$
10 → 0000 1010
31 → 0001 1111

Adding Binary Nums

0+1=1
1+1=0 carry 1
1+1+1=1 carry 1

e.g.
```
   1 1 1 1
  0001 0111        0111 1111
+ 0011 1110      + 0101 0101
  0101 0101        1101 0100
```

HEX table

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

powers of 16: 4096 (16^3), 256 (16^2), 16 (16^1), 1 (16^0)

Binary to Hex → binary → dec → hex or TRICK - split in half:

e.g. 0001 | 0111 → 17_{16} $\qquad$ 0011 | 1010 → 3A
(8 4 2 1 | 8 4 2 1)

~→

or TRICK

AF → 255
1111 | 1111

B2
1011 | 0010

HEX to Bin

or use hex → dec → binary

Hex to Dec

→ convert hex to numbers & mult by power of 16 & add tog

e.g.
- B2 → 11(16) + 2(1) = 178
- 178_{16} → 1(256) + 7(16) + 8 = 376
- FF → 255

Dec to Hex

→ determine how many digits of hex are, by finding largest num < decima

4096 256 16 1

→ see how many of each power will

e.g. 2000 = 7 D 0 (256 × 7 = 1792) (16 × 13 = 208) 44 = 2 C 4097_{10} → 100

ASCII

- an 8-bit code
- used to represent a letter/number/symb

char → ASCII code → Binary

' _ ' → 32 → 0010 0000

Propositional Logic

- tied to philosophy
- true (T) - 1, false (F) - 0

3 logical connectives:

AND	$\wedge$	&&
OR	$\vee$	||
NOT	$\neg$ ‾	!

Truth tables

- 2^n rows where n is # of variable
- cols
  - 1 for each var
  - 1 for single not vars
  - 1 for each operation (inside

e.g. $\overline{(A \vee B) \wedge \bar{C}}$ $2^3 = 8$ rows

A	B	C	$\bar{C}$	$(A \vee B)$	$(A \vee B) \wedge \bar{C}$	$\overline{(A \vee B) \wedge \bar{C}}$
1	1	1	0	1	0	1
1	1	0	1	1	1	0
1	0	1	0	1	0	1
1	0	0	1	1	1	0
0	1	1	0	1	0	1
0	1	0	1	1	1	0
0	0	1	0	0	0	1
0	0	0	1	0	0	1

Tautology - proposition that is always true
- $p \vee \neg p$

p	$\neg p$	$p \vee \neg p$
1	0	1
0	1	1

Contradiction - proposition that is always false
- $p \wedge \neg p$

p	$\neg p$	$p \wedge \neg p$
0	1	0
1	0	0

Eng → Propositional Logic ~ English sentences can be converted to logical operators to determine truth value
* let statements
* determine propositions
* determine logical operators

e.g. the cat is not brown or the truck flies
let p represent "the cat is brown" $\neg p \vee q$ ∵ trucks can't fly
let q represent "the truck flies" ∴ true when cat is not brown.

Types of Operators -

LOGICAL	$\vee$	(||)	OR
	$\wedge$	(&&)	AND
	$\neg$, $-$	(!)	NOT
ARITHMETIC	$+$	(+)	
	$-$	(-)	
	$\times$	(*)	
	$\div$	(/)	
	R	(%)	
ASSIGNMENT	$=$	(=)	
RELATIONAL	$\leq$	(<=)	$>$ (>)
	$<$	(<)	$=$ (==)
	$\geq$	(>=)	$\neq$ (!=)

Logical Properties - DeMorgan's Law, Distributive, Commutative, Tautology, Contradiction, Associative, +

How to do a proof:
① Split into LS and RS ★ don't skip steps & write which laws used
② Goals:
- work on one (or both) sides until they are equal
- start with more complex side

③ Make conclusion (∵ LS ≂ RS ∴ ...... ≂ ......)