PROPOSITIONAL LOGIC

$\neg, \wedge, \vee, \rightarrow, \leftrightarrow$

$\wedge \equiv T$ – all operands are T

$\vee \equiv T$ – one operand is T

$\rightarrow \equiv F$ – hypothesis T, conclusion F

model: $i(U) = T$

anti-model: $i(U) = F$

U is **consistent**: at least one model

$\models U$ **tautology**: all its interpretations are T (valid)

U **inconsistent**: 0 models

U **contingent**: consistent but not tautology

$U \models V$: V **logical consequence** of U

$i(U) = T \Rightarrow i(V) = T$

$U \equiv V$: U, V **logically equivalent** : identical truth tables

$\{U_1 \dots U_m\}$ consistent: $i(U_1 \wedge \dots \wedge U_m) = T$

$\{U_1 \dots U_m\}$ inconsistent : $i(U_1 \wedge \dots \wedge U_m) = F$

$U_1 \dots U_m \models V$: $i(U_1 \wedge \dots \wedge U_m) = T \Rightarrow i(V) = T$

$\boxed{T}$

$\boxed{\models U \Leftrightarrow \neg U \text{ inconsistent}}$

$\boxed{U \models V \Leftrightarrow \models U \rightarrow V \Leftrightarrow \{U, \neg V\} \text{ inconsistent}}$

$\boxed{U \equiv V \Leftrightarrow \models U \leftrightarrow V}$

$\boxed{U_1 \dots U_m \models V \Leftrightarrow \models U_1 \dots U_m \rightarrow V \Leftrightarrow \{U_1 \dots U_m, \neg V\} \text{ inconsistent}}$

clause: disjunction of literals

cube: conjunction of literals

disjunctive normal form: disjunction of cubes $\vee(\wedge)$

conjunctive normal form: conjunction of clauses $\wedge(\vee)$

DNF → finds **models**: interpretations evaluated as Ⓣ

CNF → finds **anti-models**: interpretations evaluated as Ⓕ

Syntactic approach

A_1: $U \rightarrow (V \rightarrow U)$

A_2: $(U \rightarrow (V \rightarrow Z)) \rightarrow ((U \rightarrow V) \rightarrow (U \rightarrow Z))$

A_3: $(U \rightarrow V) \rightarrow (\neg V \rightarrow \neg U)$: **modus tollens**

$U, U \rightarrow V \vdash_{mp} V$: **modus ponens**

$\vdash U$ theorem : formula derivable only from axioms and modus ponens

Soundness and completeness : $\boxed{\vdash U \text{ if and only if } \models U}$

Theorem of deduction : if $\boxed{U_1 \dots U_{m-1}, U_m \vdash V} \Rightarrow \boxed{U_1 \dots U_{m-1} \vdash U_m \rightarrow V}$

Reverse of the theorem of deduction: if $\boxed{U_1 \dots U_{m-1} \vdash U_m \rightarrow V}$ then $\boxed{U_1 \dots U_{m-1}, U_m \vdash V}$

PREDICATE LOGIC

$A_1: U \rightarrow (V \rightarrow U)$

$A_2: (U \rightarrow (V \rightarrow Z)) \rightarrow ((U \rightarrow V) \rightarrow (U \rightarrow Z))$

$A_3: (U \rightarrow V) \rightarrow (\neg V \rightarrow \neg U)$: modus tollens

$A_4: (\forall x)\, U(x) \rightarrow U(t)$: universal instantiation

$A_5: (U \rightarrow V(y)) \rightarrow (U \rightarrow (\forall x)\, V(x))$

bound variables: the variables which are within the scope of a quantifier

closed formula: all its variables are bound

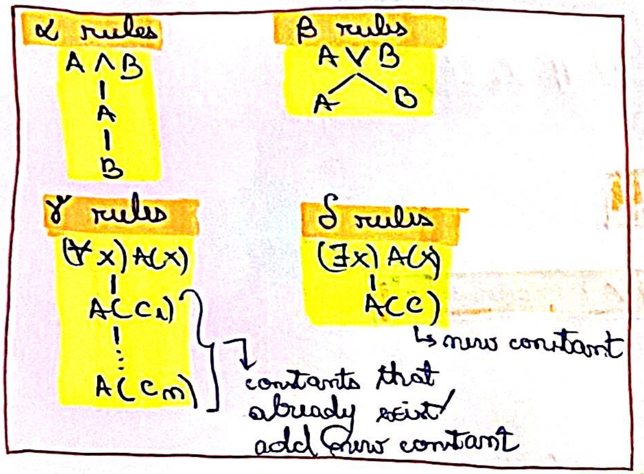
open formula: at least one free variable

- interpretation : $I = \langle \Delta, m \rangle$
  - Δ: domain
  - m: function that assigns: fixed value $m(c) \in \Delta$ to the constant c
    - function $m(f): \Delta^m \rightarrow \Delta$ to each f
    - predicate $m(P): \Delta^m \rightarrow \{T, F\}$ to each P

$\forall \rightarrow \wedge$

$\exists \rightarrow \vee$

Semantic Tableaux



- formula + its negation ⇒ ⊗ closed $\xrightarrow{\text{complete}}$ ⊙ not closed
- all formulas are decomposed / no new formulas obtained

- semantic tableaux closed → all branches are closed
- semantic tableaux open → at least one open branch
- open branch ⇒ models;

U tautology ⇔ ¬U closed semantic tableaux

$U_1, U_2 \dots U_m \models V \Leftrightarrow U_1 \wedge U_2 \wedge \dots \wedge U_m \wedge \neg V$ closed semantic tableau

models U → open branches U

anti-models U → open branches ¬U

semantic tableau U ⇒ DNF

Resolution method

$S \vdash_{Res} \square \Leftrightarrow$ inconsistent

U theorem $\Leftrightarrow \boxed{CNF(\neg U) \vdash_{Res} \square}$

$U_1, U_2, \dots, U_m \vdash V \Leftrightarrow \boxed{CNF(U_1 \wedge U_2 \wedge \dots \wedge U_m \wedge \neg V) \vdash_{Res} \square}$

Davis Putman

- delete tautologies
- delete clauses subsumed by other clauses : $C_1 = C_2 \vee C_3 \Rightarrow \cancel{C_1}$
- delete pure literals (we can't find its negation)

Level saturation strategy

$$\boxed{S^k = \{Res(C_i, C_j) \mid C_i \in S^{k-1}, C_j \in S^0 \cup S^1 \cup \dots \cup S^{k-1}\}}$$

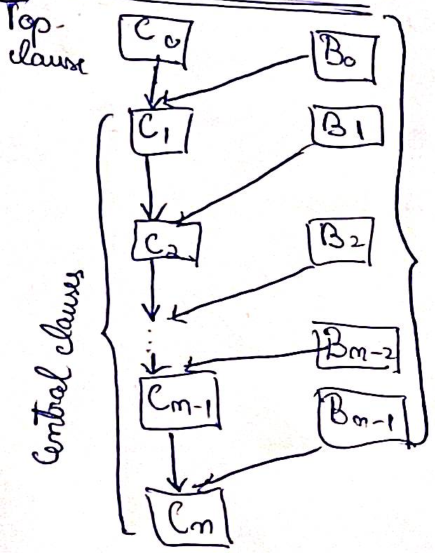
Deletion strategy

- tautologies or subsumed resolvents are eliminated

Lock resolution (with level saturation strategy)

- each literal arbitrarily indexed
- the indices resolved upon must have the lowest indices in their clauses
- the literals from the resolvents inherit the indices from their parent clauses; if the parent clauses have a common literal, in the resolvent literal we put the lowest index

Linear resolution (with deletion strategy)



Resolution in predicate logic

$\boxed{\text{Prenex}}$ normal form : $\underbrace{(Q_1x_1)\dots(Q_mx_m)}_{\text{prefix}}\underbrace{M}_{\text{matrix}}$ ← CNF ⇒ conjunctive prenex normal form

$\boxed{\text{Skolem}}$ normal form :
- for each $\exists$
  - if there are no $\forall \Rightarrow$ we introduce a new constant a, replace in M all occurances of x by a. $\exists$ is deleted
  - if there are $\forall \Rightarrow$ we replace x by $f(\dots)$
- we only have $\forall$ after ↑

$\boxed{\text{Clausal}}$ normal form : we delete the prefix of U^s

$c_1 = f \vee l_1,\ c_2 = g \vee \neg l_2$ = clashing clauses $\Longleftrightarrow \boxed{\exists \lambda = mgu(l_1, l_2)}$

$\boxed{\models V \Longleftrightarrow (\neg V)^c \vdash^{Pr}_{Res} \square}$

$\boxed{U_1, U_2 \ldots U_m \vdash V \Longleftrightarrow \{U_1^c, U_2^c \ldots U_m^c, (\neg V)^c\} \vdash^{Pr}_{Res} \square}$

monom = conjunction of variables

minterm = canonical monom / monom with n variables $x_1^{\alpha} \wedge \ldots \wedge x_m^{\alpha}$: m

maxterm = disjunction ~~of~~ with n variables : M

$\Delta CF = \bigvee$ minterms

$\epsilon CF = \bigwedge$ maxterms

adjacent (neighboring) monoms → differ only by the power of x_i

factorisation of m and m' → $m \boxed{\vee} m'$

$M(f)$: set of maximal monoms - obtained by factorisation

$C(f)$: set of central monoms - at least one minterm circled once

Simplification process:
- transform to ΔCF
- factorisation ⇒ maximal monoms
- central monoms
- simplified forms are obtained

Veitch

	x_1	$\overline{x}_1$
x_2	m_3	m_1
$\overline{x}_2$	m_2	m_0

	x_1	x_1	$\overline{x}_1$	$\overline{x}_1$
x_2	m_7	m_6	m_2	m_3
$\overline{x}_2$	m_5	m_4	m_0	m_1
	x_3	$\overline{x}_3$	$\overline{x}_3$	x_3

	x_1	x_1	$\overline{x}_1$	$\overline{x}_1$	
x_2	m_{16}	m_{15}	m_5	m_7	x_4
x_2	m_{14}	m_{12}	m_4	m_6	$\overline{x}_4$
$\overline{x}_2$	m_0	m_8	m_0	m_2	$\overline{x}_4$
$\overline{x}_2$	m_{11}	m_9	m_1	m_3	x_4
	x_3	$\overline{x}_3$	$\overline{x}_3$	x_3	

Karnaugh

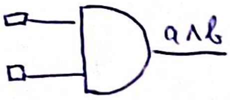
$x_1 \backslash x_2$	0	1
0	m_0	m_1
1	m_2	m_3

$x_1 \backslash x_2x_3$	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

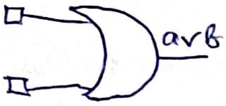
$x_1x_2 \backslash x_3x_4$	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{16}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

BASIC GATES

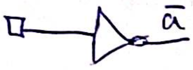
AND



OR



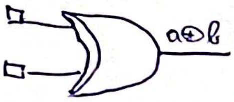
NOT



DERIVED GATES

XOR

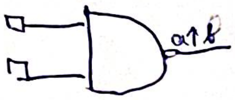
$a \oplus b = \bar{a} b \vee a \bar{b}$



NAND

$a \uparrow b = \overline{ab}$

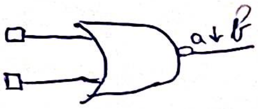
$= \bar{a} \vee \bar{b}$



NOR

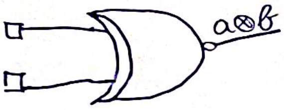
$a \downarrow b = \overline{a \vee b}$

$= \bar{a} \bar{b}$



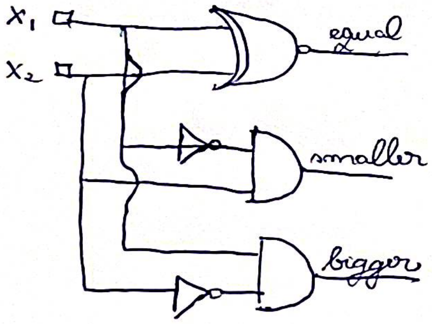
NXOR

$a \otimes b = \bar{a} \bar{b} \vee a b$



Useful combinational circuits
- comparator
- adder
- subtractor
- encoder
- decoder

Comparator



Full adder

Input			Outputs	
a	b	c_{in}	S	c_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$\Rightarrow$ $S(a, b, c_{in}) = \bar{a} \bar{b} c_{in} \vee \bar{a} b \bar{c}_{in} \vee a \bar{b} \bar{c}_{in} \vee a b c_{in}$

$C_{out}(a, b, c_{in}) = \bar{a} b c_{in} \vee a \bar{b} c_{in} \vee a b \bar{c}_{in} \vee a b c_{in}$

→ Veitch/Karnough/Quimie/Moisil simplified form

$S(a, b, c_{in}) = a \oplus b \oplus c_{in}$; $C_{out}(a, b, c_{in}) = a b \vee a c_{in} \vee b c_{in}$

Full subtractor

$a-b-c_{in} \geq 0$

$d = a-b-c_{in}, \ c_{out}=0$

else

$d = 2+a-b-c_{in}, \ c_{out}=1$

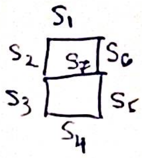
Inputs			Outputs	
a	b	c_{in}	d	c_{out}
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$d(a,b,c_{in}) = \bar{a}\bar{b}c_{in} \vee \bar{a}b\bar{c}_{in} \vee a\bar{b}\bar{c}_{in} \vee abc_{in} \Rightarrow d = a \oplus b \oplus c_{in}$

$c_{out}(a,b,c_{in}) = \bar{a}\bar{b}c_{in} \vee \bar{a}b\bar{c}_{in} \vee \bar{a}bc_{in} \vee abc_{in} \Rightarrow c_{out} = \bar{a}b \vee \bar{a}c_{in} \vee bc_{in}$

~~Encoder~~

Segment circuit



Binary codes