Lecture 3 - propositional logic - Syntax

$\{p, q, r, \dots\}$ - finite set of propositional variables

connectives $= \{ \neg$ (negation), $\wedge$ (conjunction), $\vee$ (disj.), $\rightarrow$ (implication)
$\leftrightarrow$ (equivalence) $\}$ (in decreasing order of precedence

new connectives: $\uparrow$ ("nand"), $\downarrow$ ("nor"), $\oplus$ ("xor")

$p\uparrow q = \neg(p\wedge q) \quad p\downarrow q = (\neg(p \vee q) \quad p \oplus q = \neg(p \leftrightarrow q)$

p	q	$\neg p$	$p\wedge q$	$p\vee q$	$p\rightarrow q$	$p\leftrightarrow q$	$p\uparrow q$	$p\downarrow q$	$p\oplus q$
T	T	F	T	T	T	T	F	F	F
T	F	F	F	T	F	F	T	F	T
F	T	T	F	T	T	F	T	F	T
F	F	T	F	F	T	T	T	T	F

truth table for all the connectives

'$\wedge$' true $\Leftrightarrow$ all its operands are true
'$\vee$' false $\Leftrightarrow$ all its operands are false
$p \rightarrow q$ false $\Leftrightarrow$ p true, q false
$p \leftrightarrow$ true $\Leftrightarrow$ p, q same truth value

"$\oplus$" true only when one true & one false

Semantic concepts

Let $U(p_1, p_2, \dots, p_n) \in F_p$ formula $\Rightarrow$ $i: \{p_1, p_2, \dots, p_n\} \rightarrow \{T, F\}$ - interpretation

U has 2^n interpretations
i - model of $U \Leftrightarrow i(U) = T$ (evaluates U as true)
i - anti-model of $U \Leftrightarrow i(U) = F$ (evaluates as false)
U - consistent if it has a model ($\exists i: \{p_1, \dots, p_n\} \rightarrow \{T, F\}$ s.t. $i(U) = T$)
U - valid (tautology) if all interpretations are true (models) ($\forall i, i(U) = T$) not. $\models U$
U - inconsistent if it has no models
U - contingent if consistent but not valid

The formula V is a logical consequence of U ($U \models V$) if $\forall i \in F_p \rightarrow \{T, F\}$, $i(U) = T \Rightarrow i(V) = T$
U and V are logically equivalent if they have identical truth tables ($U \equiv V$)

$\neg\neg U \equiv U$	$U \rightarrow U \equiv T$
$U \wedge \neg U \equiv F$	$U \vee \neg U \equiv T$
$T \wedge U \equiv U$	$F \vee U \equiv U$
$U \rightarrow T \equiv T$	$U \rightarrow F \equiv \neg U$
$T \rightarrow U \equiv U$	$F \rightarrow U \equiv T$

simplif. laws

$U \wedge V \equiv V \wedge U$
$U \vee V \equiv V \vee U$

commutative laws

$U \wedge (V \vee Z) \equiv (U \wedge V) \vee (U \wedge Z)$
$U \vee (V \wedge Z) \equiv (U \vee V) \wedge (U \vee Z)$

$U \wedge U \equiv U$ / $U \vee U \equiv U$ — idempotency laws

$U \wedge (U \vee V) \equiv U$ / $U \vee (U \wedge V) \equiv U$ — absorption laws

$\neg(U \wedge V) \equiv \neg U \vee \neg V$ / $\neg(U \vee V) \equiv \neg U \wedge \neg V$ — De Morgan laws

$(U \wedge V) \wedge Z \equiv U \wedge (V \wedge Z)$ / $U \vee (V \vee Z) \equiv (U \vee V) \vee Z$ — associative laws

logical equivalences

$U \to V \equiv \neg U \vee V$	$U \to V \equiv \neg(U \wedge \neg V)$
$U \to V \equiv U \leftrightarrow (U \vee V)$	$U \to V \equiv V \leftrightarrow (U \vee V)$
$U \leftrightarrow V \equiv (U \to V) \wedge (V \to U)$	$U \oplus V \equiv \neg(U \to V) \vee \neg(V \to U)$
$U \leftrightarrow V \equiv (U \vee V) \to (U \wedge V)$	$U \wedge V \equiv \neg(\neg U \vee \neg V)$
$U \vee V \equiv \neg(\neg U \wedge \neg V)$	$U \wedge V \equiv \neg(U \to \neg V)$
$U \vee V \equiv \neg U \to V$	$\neg U \equiv U \downarrow U$
$\neg U \equiv U \uparrow U$	$U \wedge V \equiv (U \downarrow U) \downarrow (V \downarrow V)$
$U \vee V \equiv (U \uparrow U) \uparrow (V \uparrow V)$	$U \wedge V \equiv (U \uparrow V) \uparrow (U \uparrow V)$
$U \vee V \equiv (U \downarrow V) \downarrow (U \downarrow V)$	

Sets of propositional formulas; $\{U_1, U_2, \dots, U_m\}$ - set (formula)

- consistent if $U_1 \wedge U_2 \wedge \dots \wedge U_m$ consistent; $\exists i$ s.t. $i(U_1 \wedge \dots \wedge U_m) = T$
- inconsistent if $U_1 \wedge \dots \wedge U_m$ inconsistent; $\forall i$ s.t. $i(U_1 \wedge \dots \wedge U_m) = F$
- formula V - logical consequence of the set; $\forall i$ s.t. $i(U_1 \wedge \dots \wedge U_m) = T \Rightarrow i(V) = T$

$U_1, \dots, U_m$ - premises / hypotheses / facts

V - conclusion

$\models U \Leftrightarrow \neg U$ inconsistent

$U \models V \Leftrightarrow \models U \to V \Leftrightarrow \{U, \neg V\}$ inconsist.

$U \equiv V \Leftrightarrow \models U \leftrightarrow V$

$U_1, U_2, U_m \models V \Leftrightarrow U_1 \wedge \dots \wedge U_m \to V \Leftrightarrow \{U_1 \wedge \dots \wedge U_m, \neg V\}$ inconsistent

CL → notes
↘ exam papers

Axioms → A1: $U \to (V \to U)$

A2: $(U \to (V \to Z)) \to ((U \to V) \to (V \to Z))$

A3: $(U \to V) \to (\neg V \to \neg U)$ (modus tollens)

Semantics of propositional logic

- logical prop. are models of propositional assertions from natural language, which can be true or false
- the aim of the semantics is to give a meaning to the propositional formulas (to assign a truth value)
- the semantic domain is the set of truth values (T, F) s.t. $\neg T = F$, $\neg F = T$
- new connectives: $\uparrow$, $\downarrow$, $\oplus$ - used in the design of logic circuits
- the semantics of the connectives are provided by the truth table
- the semantics is compositional, meaning that the truth value of a formula is obtained from the truth values of its subformulas.

Axiomatic system of prop. logic $P = (\Sigma_P, F_P, A_P, R_P)$

$\Sigma_P = Var\_props \cup Connectives \cup \{(,)\}$ - vocabulary

set of prop. variables

F_P - set of well formed formulas; A_P - set of axioms

R_P - set of inference rules containing modus ponens rule (not. $U, U \to V \vdash_{mp} V$)

A formula $U \in F_P$ s.t. $\emptyset \vdash U$ (or $\vdash U$) is called theorem (derivable / inferable only from the axioms & using mp as inference rule) Ex2 L4

properties of prop. logic: soundness th. (if $\vdash U$, then $\models U$) a th. is a tautology

completeness th (if $\models U$, then $\vdash U$) a tautology is a th.

$\Rightarrow$ th. of soundness & completeness ($\vdash U \Leftrightarrow \models U$)

consequences → prop. logic is non contradictory (we can't have $\vdash U$ & $\vdash \neg U$ simult.)

is coherent (not every prop. formula is a theorem)

decidable (we can always decide whether a prop. form. is a th. or not)

Th. of deduction: if $U_1, ..., U_{m-1}, U_m \vdash V \Rightarrow U_1, ..., U_{m-1} \vdash U_m \to V$

Reverse: if $U_1, ..., U_{m-1} \vdash U_m \to V \Rightarrow U_1, ..., U_{m-1}, U_m \vdash V$

Consequences: $\vdash U \to ((U \to V) \to V)$

$\vdash (U \to V) \to ((V \to Z) \to (U \to Z))$

$\vdash (U \to (V \to Z)) \to (V \to (U \to Z))$ - permutation of the premises

Normal forms

$p, \neg q, r$ - literals (a prop. variable or its negation)
clause - disjunction of literals ($p, \neg p \vee q, q \vee r \vee s$)
cube - conjunction of literals ($q, p \wedge \neg q, r \wedge s \wedge p$)
DNF (disjunctive normal form - disj. of cubes ($p \vee \neg q \vee r$; $p \wedge q$; $p \vee (q \wedge r)$)
CNF - conj. of clauses ($p \vee \neg q \vee r$; $p \wedge q$; $p \wedge (q \wedge r)$)

the clause $U = p \vee q \vee r \vee \neg p$ - tautology ($U \equiv T$) bcs of. $p, \neg p$
the cube $V = p \wedge q \wedge r \wedge \neg p$ - inconsistent ($V \equiv T$) — // —

Normalization algorithm

1) $X \to Y$ becomes $\neg X \vee Y$; $X \leftrightarrow Y$ becomes $(\neg X \vee Y) \wedge (\neg Y \vee X)$
2) De Morgan's laws ; $\neg\neg X \equiv X$
3) distribution laws

CNF tautology $\Leftrightarrow$ all clauses are tautologies - direct form to prove tautology
DNF inconsistent $\Leftrightarrow$ all cubes are inconsistent
DNF finds all the models of a formula ; CNF finds all the anti-models

$\models U$ - tautology (all interpretations are models - evaluate to True)

$U \models V$ - the formula V is a logical consequence of U

$\cancel{\forall i \in}$ $\forall i: \mathcal{F}_p \to \{T, F\}$, s.t. $i(U) = T \Rightarrow i(V) = T$

$U \equiv V$ - the formulas are logically equivalent if they have identical truth tables

the set $\{U_1, U_2, ..., U_m\}$ - consistent - if $U_1 \wedge U_2 \wedge ... \wedge U_m$ is consistent

$\exists i: \mathcal{F}_p \to \{T, F\}$ s.t. $i(U_1 \wedge U_2 \wedge ... \wedge U_m) = T$

$\{U_1, U_2, ..., U_m\}$ - inconsistent - if $U_1 \wedge U_2 \wedge ... \wedge U_m$ inconsistent ; $\nexists i: \mathcal{F}_p \to \{T,F\}$ s.t. $i(U_1 \wedge U_2 \wedge ... U_n) = F$

V - logical consequence of the set - if $\forall i: \mathcal{F}_p \to \{T, F\}$, $i(U_1 \wedge U_2 \wedge ... \wedge U_m) = T \Rightarrow i(V) = T$

$U_1, U_2, ..., U_m \models V$

- $\models U \Leftrightarrow \neg U$ inconsistent
- $U \models V \Leftrightarrow \models U \to V \Leftrightarrow \{U, \neg V\}$ inconsistent
- $U \equiv V \Leftrightarrow \models U \leftrightarrow V$
- $U_1, U_2, ..., U_m \models V \Leftrightarrow \models U_1 \wedge U_2 \wedge ... \wedge U_m \to V$ inconsistent $\Leftrightarrow \{U_1, U_2, ..., U_m, \neg V\}$ incons.

$p, \neg q, r$ - literals ; $p, \neg p \vee q, r \vee q \vee s$ - clauses (disjunction)

$q, p \wedge \neg q, r \wedge s \wedge p$ - cubes (conjunction)

clause

$U = p \vee q \vee r \vee \neg p$ - tautology, bcs. $p, \neg p$ - opposite

$V = p \wedge q \wedge r \wedge \neg p$ - inconsistent, bcs. $p, \neg p$ - opposite

cube

CNF - tautology $\Leftrightarrow$ all clauses are tautologies

DNF - inconsistent $\Leftrightarrow$ all cubes are inconsistent

DNF (disjunctive normal form) $\to$

$\to$ disjunction of cubes

$p \vee \neg q \wedge r$ - 3 unit cubes

$p \wedge q$ - DNF, 1 cube

$p \vee (q \wedge r) \vee (\neg p \wedge \neg r \wedge s)$ - 3 cubes

CNF (conjunctive normal form) $\to$

$\to$ conjunction of clauses

$p \vee \neg q \vee r$ - CNF, 1 clause

$p \wedge q$ - 2 unit clauses

$p \wedge (q \vee r) \wedge (\neg p \vee \neg r \vee s)$ - 3 clauses

Resolution method

Formal system

- $\Sigma_{res} = \Sigma_p - \{\to, \leftrightarrow, \wedge\}$ - the alphabet
- $F_{res} \cup \{\square\} \to$ the empty clause, symbolises inconsistency
  ↳ set of all clauses built using the alphabet
- $A_{res} = \emptyset$ - set of axioms
- R_{res} - set of inference rules containing the resolution rule $(f \vee l, g \vee \neg l \vdash_{res} f \vee g)$ ↗ l literal, $f, g \in F_{res}$

$C_1 = f \vee l$, $C_2 = g \vee \neg l$ - clashing clauses; they resolve upon the literal l

not. $C = Res_l(C_1, C_2) = f \vee g$; C - the resolvent of the parent clauses C_1 and C_2

example

$C_1 = l, C_2 = \neg l \Rightarrow Res_l(C_1, C_2) = \square$ - inconsistent

Soundness & completeness th: a set S of prop. clauses is inconsistent $\Leftrightarrow S \vdash_{Res} \square$ (the empty clause is derived from the set S)

U is a th. $\Leftrightarrow CNF(\neg U) \vdash_{Res} \square$

$U_1, ..., U_m \vdash V \Leftrightarrow U_1, ..., U_m \models V \Leftrightarrow CNF(U_1 \wedge ... \wedge U_m \wedge \neg V) \vdash_{Res} \square$

Lock resolution

- each occurrence of a literal from a set of clauses is arbitrarily indexed w/ an integer
- the literals resolved upon must have the lowest indices in their clauses
- the literals from resolvents inherit the indices from parents; lowest index

consistency - lock res. must be combined w/ level saturation strat. ($S^k = \emptyset$)

inconsistency - we don't need a strat. ($\square \in S^k$)

Predicate logic

Axiomatic system $P_r = (\Sigma_{Pr}, F_{Pr}, A_{Pr}, R_{Pr})$

$\Sigma_{Pr} = Var \cup Const \cup (\cup_{j=1}^{m} F_j) \cup (\cup_{j=1}^{m} P_j) \cup$ Connectives $\cup$ Quantifiers = VOCABULARY

var - set of variable symbols $\{x, y, z, ...\}$
const - set of constants $\{a, b, c, ...\}$
$F_i = \{f \mid f : D^i \to D\}$ - set of function symbols of arity 'i'
$P_i = \{p \mid p : D^i \to \{T, F\}\}$ - set of predicate —//—

Connectives $= \{\neg, \wedge, \vee, \to, \leftrightarrow\}$; Quantifiers $= \{\forall$(universal q.), $\exists$(exist. q.)$\}$

F_{Pr} - set of well-formed formulas

A_{Pr} - set of axioms $\to$ $A_1 : U \to (V \to U)$
$A_2 : (U \to (V \to Z)) \to ((U \to V) \to (U \to Z))$
$A_3 : (U \to V) \to (\neg V \to \neg U)$ modus tollens
$A_4 : (\forall x) U(x) \to U(t)$, t a term universal instantiation
$A_5 : (U \to V(y)) \to (U \to (\forall x) V(x))$ x not free in U or V
y free, V does not appear in U

$R_{Pr} = \{mp, gen\}$ - set of inference rules
- modus ponens : $U, U \to V \vdash_{mp} V$
- univ. generalization rule: $U(x) \vdash_{gen} (\forall x) U(x)$, x a free var. in U

variables which are within the scope of a quantifier - bound var. (free otherwise)
closed formula - all variables are bound
open formula - at least one free variable

Inference rules: univ. instantiation $(\forall x) U(x) \vdash_{univ\_inst} U(t)$, t is a term (variable/constant)
univ. generalization $U(x) \vdash_{univ\_gen} (\forall x) U(x)$ x is a free var. in U
exist. inst. $(\exists x) U(x) \vdash_{exist\_inst} U(c)$, c is a new constant
exist. gen $U(t) \vdash_{exist\_gen} (\exists x) U(x)$, x must not appear free in U

Interpretation - a pair $I = \langle D, m \rangle$, where
D - nonempty set ("domain of interpretation")
m - function that assigns
- a fixed val. $m(c) \in D$ to the constant c
- a function $m(f) : D^n \to D$ to each n-ary function symbol f
- a predicate $m(P) : D^n \to \{T, F\}$ to each n-ary predicate symbol P

modus ponens : $U, U \to V \vdash_{mp} V$ $\qquad \vdash U$ – theorem

axioms : A1: $U \to (V \to U)$

A2: $(U \to (V \to Z)) \to ((U \to V) \to (U \to Z))$

A3: $(U \to V) \to (\neg V \to \neg U)$ – modus tollens

Properties of propositional logic:

Soundness th.: if $\vdash U \Rightarrow \models U$ (a theorem is a tautology)

Completeness th: if $\models U \Rightarrow \vdash U$ (a tautology is a theorem)

Th. of soundness and completeness $\vdash U \Leftrightarrow \models U$

Consequences: • we can't have $\vdash U$ and $\vdash \neg U$ simultaneously

• prop. logic is coherent ; –"– is decideable

Theorem of deduction : if $U_1, U_2, ..., U_n \vdash V \Rightarrow U_1, U_2, ..., U_{n-1} \vdash U_n \to V$

reverse th: if $U_1, U_2, ..., U_{n-1} \vdash U_n \to V \Rightarrow U_1, U_2, ..., U_n \vdash V$

PREDICATE LOGIC

the set of axioms: A1: $U \to (V \to U)$

A2: $(U \to (V \to Z)) \to ((U \to V) \to (U \to Z))$

A3: $(U \to V) \to (\neg U \to \neg V)$ – modus tollens

A4: $(\forall x) U(x) \to U(t)$ – universal instantiation

A5: $(U \to V(y)) \to (U \to (\forall x) V(x))$

set of inference rules: $U, U \to V \vdash_{mp} V$ – modus ponens

$U(x) \vdash_{gen} (\forall x) U(x)$ – univ. generalization rule (x - free var. in U)

universal instantiation $(\forall x) U(x) \vdash_{univ\_inst} U(t)$ (t - variable or constant)

universal generalization $U(x) \vdash_{univ\_gen} (\forall x) U(x)$

existential instantiation $(\exists x) U(x) \vdash_{exist\_inst} U(c)$ c – new constant

–"– general. $U(t) \vdash_{exist\_gen} (\exists x) U(x)$

Expansion laws : $(\forall x) A(x) \equiv (\forall x) A(x) \wedge A(t)$

$(\exists x) A(x) \equiv (\exists x) A(x) \vee A(t)$

De Morgon infinitary laws: $\neg(\exists x) A(x) \equiv (\forall x) \neg A(x)$ and $\neg(\forall x) A(x) \equiv (\exists x) \neg A(x)$

Quantifiers interchanging laws: $(\exists x)(\exists y) A(x,y) \equiv (\exists y)(\exists x) A(x,y)$ and the same w/ '$\forall$'

quantifiers of different types do not commute

Semantic tableaux method

α-rules

$$\begin{array}{c} A \wedge B \\ | \\ A \\ | \\ B \end{array} \qquad \begin{array}{c} \neg(A \vee B) \equiv \neg A \wedge \neg B \\ | \\ \neg A \\ | \\ \neg B \end{array} \qquad \begin{array}{c} \neg(A \to B) \equiv A \wedge \neg B \\ | \\ A \\ | \\ \neg B \end{array}$$

apply α-rules which keep one branch before β-rules

β-rules

$$\begin{array}{c} A \vee B \\ / \quad \backslash \\ A \qquad B \end{array} \qquad \begin{array}{c} \neg(A \wedge B) \equiv \neg A \vee \neg B \\ / \quad \backslash \\ \neg A \qquad \neg B \end{array} \qquad \begin{array}{c} A \to B \equiv \neg A \vee B \\ / \quad \backslash \\ \neg A \qquad B \end{array}$$

γ-rules

$$\begin{array}{c} (\forall x)A(x) \\ | \\ A(c_1) \\ | \\ --- \\ A(c_n) \\ | \\ (\forall x)A(x) \end{array} \qquad \begin{array}{c} \neg(\exists x)A(x) \\ | \\ \neg A(c_1) \\ --- \\ | \\ \neg A(c_n) \\ | \\ \neg(\exists x)A(x) \end{array}$$

$c_1, ..., c_n$ are all the parameters on that branch and are used for the instantiation of A

if no constant → a new one is introduced & used for instantiation

δ-rules

$$\begin{array}{c} (\exists x)A(x) \\ | \\ A(c) \end{array} \qquad \begin{array}{c} \neg(\forall x)A(x) \\ | \\ A(c) \end{array}$$

c is a new constant on that branch

branch is called closed (⊗) if it contains a formula & its negation, otherwise it is called open (⊙)

a branch is called complete if it is closed or all the formulas on the branch are alr. decomposed

a semantic tableau is called closed if all its branches are closed; if it has at least one open branch, then it's called open

sem. tableau → complete if all the branches are complete.

a closed branch symbolizes inconsistency among the formulas on that branch

an inconsistent formula has associated a closed semantic tableau

a consistent formula has associated a complete & open semantic tableau

soundness & completeness: a formula U is a tautology $\Leftrightarrow$ $\neg U$ has a closed sem. tableau

$U_1, ..., U_n \models V \Leftrightarrow U_1 \wedge U_2 \wedge ... \wedge U_n \wedge \neg V$ has a closed semantic tableau

Definitions / Semantic concepts A - formula, J - interpretation

A - consistent if there is an J and an assignment formula s.t. $v_a^J(A) = T$ $(a \in As(J))$

(inconsistent / unsatisfiable otherwise)

A - true under the interpr. of J if for any assignment form. a, $v_a^J(A) = T$ ($\models_J A$), J - model of A

A - false - - -, J anti-model

A - valid / tautology if true under all possible interp. ($\models A$)

A, B - logical equivalent if $v_a^J(A) = v_a^J(B)$, $\forall a, J$ ($A \equiv B$)

Expansion laws $(\forall x) A(x) \equiv (\forall x) A(x) \wedge A(t)$ - infinitary conj.

$(\exists x) A(x) \equiv (\exists x) A(x) \vee A(t)$ - - // - disj.

De Morgan infinitary laws $\neg(\exists x) A(x) \equiv (\forall x) \neg A(x)$

$\neg(\forall x) A(x) \equiv (\exists x) \neg A(x)$

Quantifiers interchanging laws $(\exists x)(\exists y) A(x,y) \equiv (\exists y)(\exists x) A(x,y)$

$(\forall x)(\forall y) A(x,y) \equiv (\forall y)(\forall x) A(x,y)$

Resolution in predicate logic

Prenex normal form (th → a predicate formula admits a logical equivalent conjunctive prenex normal form)

↳ steps: 1) connectives '→', '↔' are replaced w/ '¬', '∧', '∨'

2) bound variables are renamed s.t. they will be distinct

3) infinitary De Morgan's rules

4) extraction of quantifiers in front of the formula

5) matrix is transformed into CNF using De Morgan's laws & distributive laws

U^P - conjunctive prenex form

a formula in Skolem normal form (U^S) corresponds to U

for each ~~ex~~ exist. quantifier (Q_r)

- if on the left side of Q_r are no ~~exist.~~ univ. quantifiers, a new const. (ex: a) is introduced; all occurencies of x_r are replaced and ($Q_r x_r$) is deleted

- if $Q_{s_1}, \dots, Q_{s_m}$, $1 \le s_1 < \dots < s_m < r$ are all univ. quantifiers on the left side of Q_r, then we introduce a function symbol f and replace all occurencies of x_r by $f(x_{s_1}, \dots, x_{s_m})$, ($Q_r x_r$) being deleted

- the constants & functions used to replace are called Skolem constants and Skolem functions; U^S contains only universal quantifiers & the matrix in CNF

$U_1, ..., U_m, V$ - first order formulas

V is inconsistent $\Leftrightarrow$ V^P is inconsistent $\Leftrightarrow$ V^S is inconsistent $\Leftrightarrow$ V^C is inconsistent

$\{U_1, ..., U_m\}$ inconsistent $\Leftrightarrow$ $\{U_1^C, ..., U_m^C\}$ inconsistent

Substitution: mapping from the set of variables into the set of terms

not: $\theta = [x_1 \leftarrow t_1, ..., x_i \leftarrow t_i]$; $t_i \neq x_i$ and x_i not a subterm of t_i

a subst. θ is a unifier of the terms t_1 & t_2 if $\theta(t_1) = \theta(t_2)$

$\theta(t_1)$ - common instance of the unified terms

most gen. unifier (mgu) is a unifier μ s.t. any other unifier can be obtained from μ by a further substit.

Predicate resolution axiomatic system

$Res^{Pr} = (\Sigma_{Res}^{Pr}, F_{Res}^{Pr}, A_{Res}^{Pr}, R_{Res}^{Pr})$

$\Sigma_{Res}^{Pr} = \Sigma_{Pr}$ - the alphabet $\{\rightarrow, \leftrightarrow, \wedge, \exists, \forall\}$

$F_{Res}^{Pr} \cup \{\square\}$ - set of well formed formulas

→ empty clause

→ set of all clauses built using the alphabet

$A_{Res}^{Pr} = \emptyset$ set of axioms

$R_{Res}^{Pr} = \{res^{Pr}, fact\}$ - set of inference rules containing the resolution rule & factoring rule

$\{f \vee l_1, g \vee \neg l_2\} \vdash_{res^{Pr}} \lambda(f) \vee \lambda(g)$, $\lambda = mgu(l_1, l_2)$

$l_1 \vee l_2 \vee ... \vee l_k \vee l_{k+1} \vee ... \vee l_m \vdash_{fact} \lambda(l_1 \vee ... \vee l_m)$, $\lambda = mgu(l_1, ..., l_k)$

$C_1 = f \vee l_1$, $C_2 = g \vee \neg l_2$ - clashing clauses if $\exists\ \lambda = mgu(l_1, l_2)$

$C = Res_\lambda^{Pr}(C_1, C_2) = \lambda(f) \vee \lambda(g)$ - binary resolvent of C_1 & C_2

Th. of soundness & completeness : S is inconsistent $\Leftrightarrow$ $S \vdash_{Res}^{Pr} \square$

Resolution: $\vdash V (\models V) \Leftrightarrow (\neg V)^C \vdash_{Res}^{Pr} \square$

$U_1, ..., U_m \vdash V \Leftrightarrow \{U_1^C, ..., U_m^C, (\neg V)^C\} \vdash_{Res}^{Pr} \square$

Boolean function

of n variables is a function $f:(B_2)^n \to B_2$

- the projection function $P_i: B_2^n \to B_n$, $P_i(x_1,\dots,x_n) = x_i$ is a bool. f.
- $f,g: B_2^n \to B_2 \Rightarrow f\wedge g, f\vee g, \bar{f}$ are functions

$$(f\wedge g)(x_1,\dots,x_n) = f(x_1,\dots,x_n)\wedge g(x_1,\dots,x_n)$$

— — —

$\exists\ 2^{2^n}$ functions of n variables

disj. canonical form (DCF) $f(x_1,\dots,x_n) = \vee\left(f(\alpha_1,\dots,\alpha_n)\wedge x_1^{\alpha_1}\wedge\dots x_n^{\alpha_n}\right) \Leftrightarrow \vee\left(x_1^{\alpha_1}\wedge\dots x_n^{\alpha_n}\right)$

(CCF) $f(x_1,\dots,x_n)\ \wedge\left(f(\alpha_1,\dots,\alpha_n)\vee x_1^{\bar{\alpha}_1}\vee\dots\vee x_n^{\bar{\alpha}_n}\right) \Leftrightarrow \wedge\left(x_1^{\bar{\alpha}_1}\vee\dots x_n^{\bar{\alpha}_n}\right)$

monom – conjunction of variables

minterm – monom which contains all variables $(x_1^{\alpha_1}\wedge\dots\wedge x_n^{\alpha_n})$ $m_0,\dots,m_{2^n-1}$ (-2^n)

maxterm – disjunction containing all variables $(x_1^{\alpha_1}\vee\dots\vee x_n^{\alpha_n})$ $M_0,\dots,M_{2^n-1}$ (-2^n)

↳ the index is obtained by the conversion $b_2 \to b_{10}$

ex $m_3 = m_{011} = x^0\wedge y^1\wedge z^1 = \bar{x}\wedge y\wedge z$

$M_7 = M_{111} = x^{\bar{1}}\vee y^{\bar{1}}\vee z^{\bar{1}} = \bar{x}\vee\bar{y}\vee\bar{z}$

CCF – conj. of maxterms corresponding to the values 0 of the function

DCF – disj of the minterms corresponding to the values 1 of the function

simplif. of boolean functions

$S_f = \{(x_1,\dots,x_n) \mid f(x_1,\dots,x_n) = 1\}$ - support set of f

m, m' are adjacent/neighbor monoms if they differ by a single variable change

maximal monoms - minterms or monoms obtained by using factorization

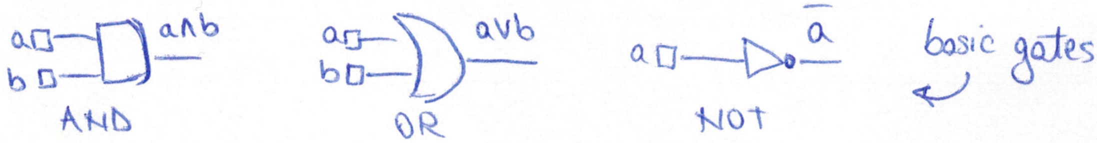
simplification process

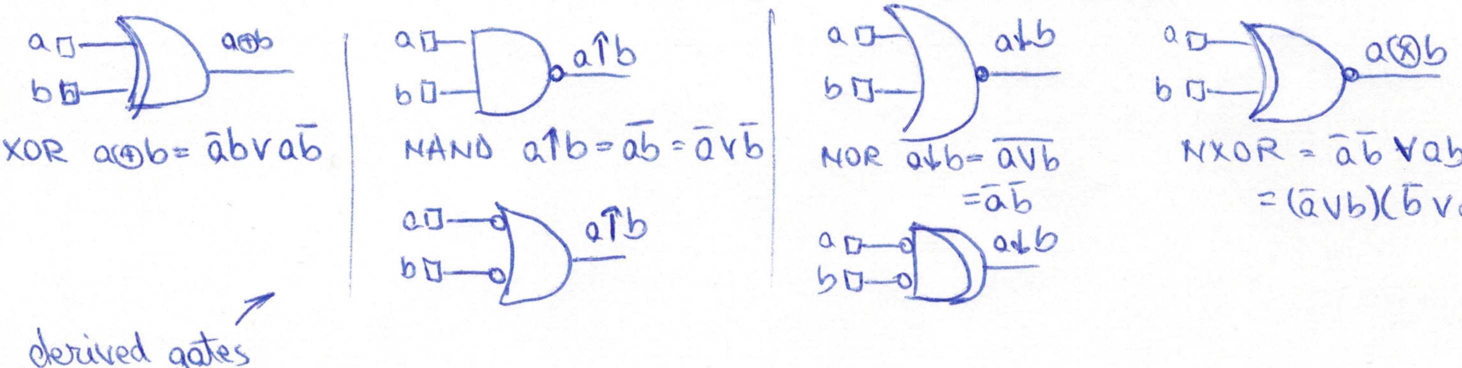
1) f transformed into $DCF(f)$
2) factorization process → set of maximal monoms $M(f)$
3) central monoms are selected → $C(f)$
4) obtain simplified forms

Logic circuits

basic gates implement one of the Boolean op : $\wedge, \vee, \bar{\ }$

gate - takes the values from its input wires & combines them with the appropriate Boolean op. to produce the label on it's output wire





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 $= \bar{a}\bar{b} \vee ab = (\bar{a} \vee b)(\bar{b} \vee a)$

Combinational circuits - logic circuit which does not use memory & the outputs respond immediately to the inputs
- can have m outputs, each corresponding to a boolean f.

input : problem specification describing the functionality of the desired circuit

output : the implementation of the logic circuit

ex : comparator, adder (half / full), subtractor (- // -), encoder, decoder

Binary codes - used to repr. the decimal digits on 4 bits
- weighted / unweighted

each binary digit has a weight ; the b_{10} value = weighted sum of the bits
ex : BCD (binary coded decimal - weights : 8, 4, 2, 1)

unweighted - Excess 3 (add '0011' to each BCD)
- Gray code all successive words differ by one digit