10/8/17

Syllogism :

1) Premise : All cats have fur.
Premise : I have a cat
Conclusion : My cat has fur.

2) Premise : Some cars rattle.
Premise : I have a car.
Conclusion : My car rattles.
This conclusion may or may not be true.

* Predicate logic - Axioms should either be true or false.
* First order logic
* Temporal / modal logic

First order logic
Eg : $x+1 > y+2$
It may be true or false depending on the value of x and y.

Temporal logic
: It depends on time.
Eg : It is raining.

* Logic formalizes valid methods of reasoning along with grammar.
Rules of logic were classified and named.

The most widely used rules are the "syllogism".

Syllogism:

If both premise are true, the rule ensures that the conclusion is true.

Logic must be formalized because reasoning expressed in informal natural language can be flawed (false).

The formalization of logic began in 19th century.

It has 4 components:

1) Logical systems:
   - ~~Actions~~ Axioms and rules of inference were developed with the understanding that different set of axioms would lead to different theorem.

2) Consistancy:
   - An axioms of logical system is consistent if it is possible to prove both a formula & its negation.

3) Independence
   - The axioms of logical s/m are independent if no axioms can be proved from other.

Camlin Page
Date / /

4) Soundness
- All theorem that can be proved in the logical s/m are true.

If all the statements are true, then the entire s/m is true.

14/8/17

14/8/17

MODULE - I

PROPOSITIONAL LOGIC

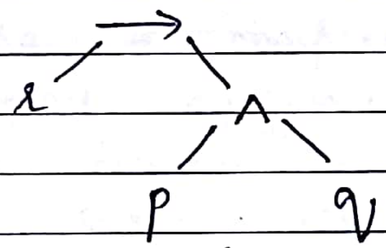
Consider

p : The car has four wheels

q : The car has a steering wheel

r : It is a car.

$$r \rightarrow p \wedge q$$



(Atoms are always represented using lower case alphabets).

* Negation $\neg$
Disjunction $\vee$
Conjunction $\wedge$
XOR $\oplus$
NOR $\Uparrow$ $\Uparrow$ $\downarrow$
Nand $\Downarrow$ $\Downarrow$ $\uparrow$
Implication $\rightarrow$
Equivalance $\longleftrightarrow$

These are 8 boolean expressions.

Camlin Page
Date / /

Note: It is a simple logical system that is a basis for all other systems.

Propositions are claims that cannot be further decomposed & that can be assigned a truth value of either true or false.

Logical s/ms formalize reasoning that are similar to programming languages that formalizes computation. In both cases, we need to define syntax & semantics.

Syntax defines what strings of symbols constitute a legal formula.

Semantics defines what the legal formula means.

Theorems:

* Symbols used to construct formulas in propositional logic are an unbounded set of symbols (τ) called atoms.

Atoms are denoted by lower case letters possibly with a subscript.

* Boolean Operators.

There are 8 boolean operations

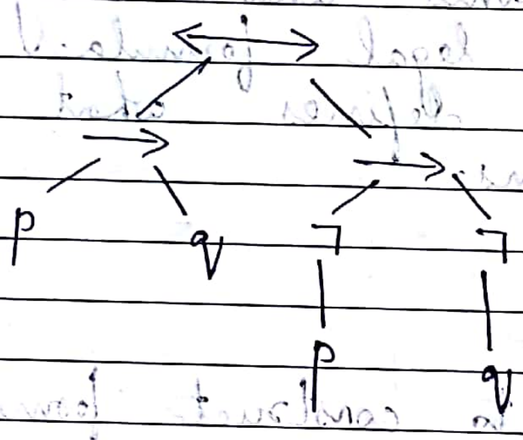
$\neg, \vee, \wedge, \oplus, \uparrow, \downarrow, \rightarrow, \leftrightarrow$

* A formula in propositional logic is a tree defined recursively.

It has 3 things:

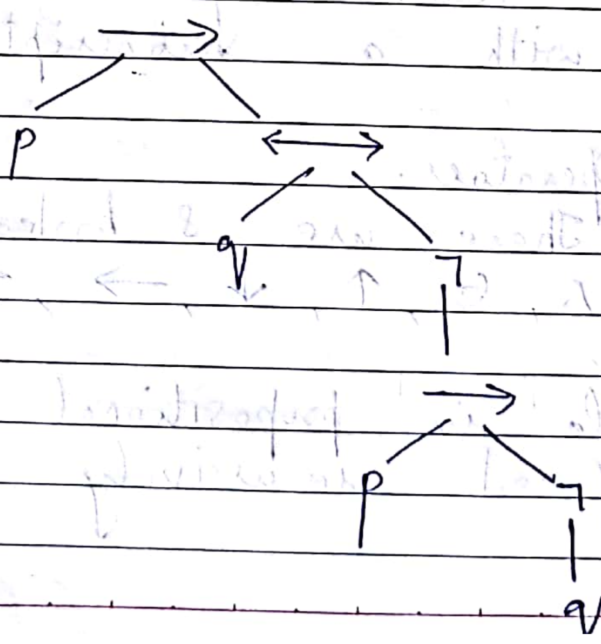
1) A formula is a leaf labelled by an atom.
2) A formula is a node labelled by one of the binary operation with two children, both of which are formulas.
3) A formula is a node labelled by negation ($\neg$) with a single child (right) that is a formula.

Q. (1)



$p \rightarrow q \leftrightarrow \neg p \rightarrow \neg q$

Q. (2)



$p \longrightarrow q \longleftrightarrow \neg\, p \longrightarrow \neg q$

* By default, we use inorder traversal.
* In both the examples, the o/p is same. Hence there is ambiguity.

22/8/17

Algorithm: Tree strings

Description : To generate a tree structure

Input : A formula in propositional logic.

Output : A string representation of it

Call recursively Inorder(A)

Inorder(A)
  If A is a leaf node
    write its label
    return
  Let A_1 and A_2 be the left node and right node respectively
  Inorder(A_1)
  Write label of root A
  Inorder (A_2)

* This algorithm returns the same string for both the tree structures. So we use parantheses.

Using Paranthesis :

Call recursively Inorder(A)

Inorder(A)

- If A is a leaf node
  - Insert '('
  - write its label
  - return
- Let A_1 & A_2 be the left and right node respectively
- Inorder(A_1)
- write label of root A
- Inorder(A_2)
- Insert ')'

① $(p \to q) \leftrightarrow (\neg p \to \neg q)$

② $(p \to (q \leftrightarrow \neg) (p \to \neg) q)$
(This has no meaning)

Using Preorder : (Polish Notation)

① $\leftrightarrow \; \to \; p \; q \; \to \; \neg p \; \neg q$

② $\to \; p \; \leftrightarrow \; q \; \neg \; \to \; p \; \neg q$

①

$\leftrightarrow$

$\to$

p

q

$\to$

$\neg$

p $\quad \not\to$

$\neg$ $\quad \not\to$

q $\quad \not\leftrightarrow$

$p \to q \leftrightarrow \neg p \to \neg q$

Scanned by CamScanner

$\rightarrow$

② p

$\leftrightarrow$

q $\quad\quad$ $\vee$

$\neg$ $\quad\quad$ $\rightarrow$

$\rightarrow$ $\quad\quad$ $\neg$

p $\quad\quad$ $\cancel{\leftrightarrow}$ $\quad$ $p \rightarrow q \rightarrow \neg p \rightarrow \neg q$

$\neg$ $\quad\quad$ $\rightarrow$

q

29/8/17

FORMAL GRAMMAR

Instead of defining formulas as tree, they can be defined as strings generated by context free grammar.

Formal Grammar:

Formula in propositional logic are derived from context free grammar whose terminals are an unbounded set of symbols ($\mathcal{P}$) called atomic proposition. The production of the grammar are:

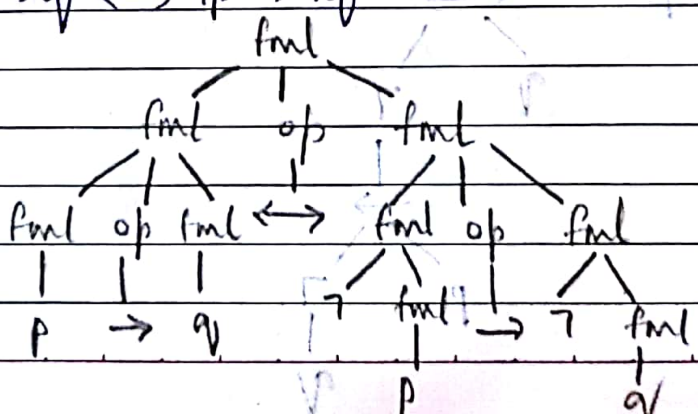
fml ::= p , where $p \in \mathcal{P}$

fml ::= $\neg$ fml

fml ::= fml op fml

op ::= $\vee \mid \wedge \mid \neg \mid \leftrightarrow \mid \uparrow \mid \downarrow \mid \oplus \mid \rightarrow$

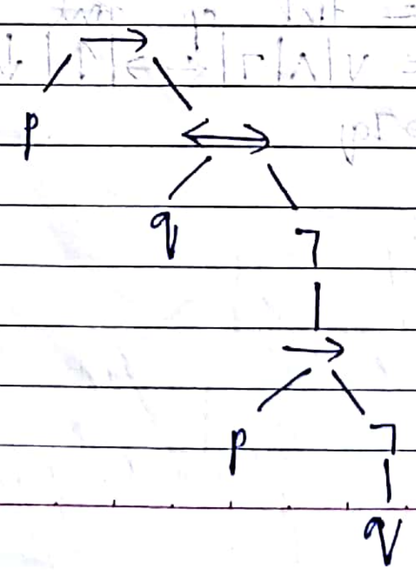
Eg: $p \rightarrow q \leftrightarrow \neg p \rightarrow \neg q$

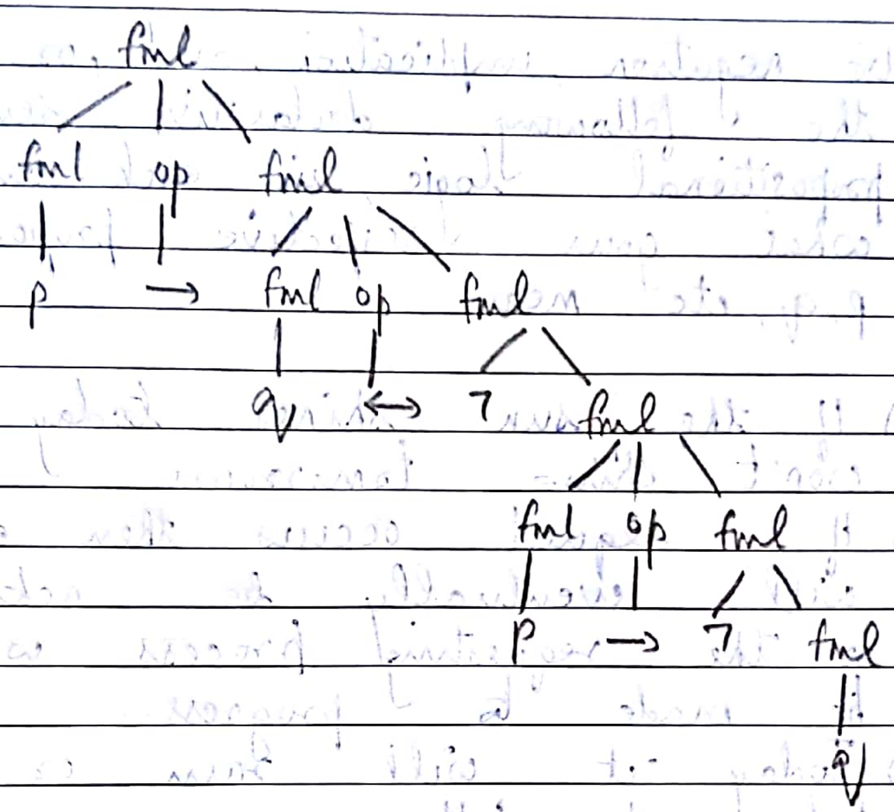


1. fml
2. fml op fml
3. fml $\leftrightarrow$ fml
4. fml op fml $\leftrightarrow$ fml
5. fml $\rightarrow$ fml $\leftrightarrow$ fml
6. $p \rightarrow$ fml $\leftrightarrow$ fml
7. $p \rightarrow q \leftrightarrow$ fml
8. $p \rightarrow q \leftrightarrow$ fml op fml
9. $p \rightarrow q \leftrightarrow$ fml $\rightarrow$ fml
10. $p \rightarrow q \leftrightarrow \neg$ fml $\rightarrow$ fml
11. $p \rightarrow q \leftrightarrow \neg p \rightarrow$ fml
12. $p \rightarrow q \leftrightarrow \neg p \rightarrow \neg$ fml
13. $p \rightarrow q \leftrightarrow \neg p \rightarrow \neg q$

* A formula is a word that can be derived from the non-terminal fml.
* The set of all formulas that can be derived from the grammar is denoted by f.
* The derivation of strings in formal language can be represented as trees.
* The word generated by the derivation can be read of the leaves from left to right.

g.





* Both generates the same string. Hence, there is ambiguity.

So we introduce paranthesis,
for fml ::= (¬ fml)
fml ::= (fml op fml)

INTERPRETATION

In interpretation, we define the semantics of formulas. We assign values to variables and evaluate the expression. Truth table are assigned to the atoms of the formula, inorder to evaluate the truth value of the formula.

11/09/17

Q1. Use negation, implication, and, or to express the following declarative sentences in propositional logic, in each case take what your respective propositional atom p, q, etc mean

a) If the sun shines today then, it won't shine tomorrow.

b) If a request occurs then either it will eventually be acknowledged or the requesting process won't ever be made to progress.

c) Today it will rain or shine but not both.

d) If Jack met Jake yesterday, they had a cup of coffee together or they took a walk in the park.

Q2. Draw corresponding parse tree

(1) $p \land (\neg q \rightarrow \neg p)$

(2) $\neg((\neg q \land (p \rightarrow r)) \land (r \rightarrow q))$

(3) $(p \land q) \rightarrow (\neg r \lor (q \rightarrow r))$

(4) $(p \rightarrow q) \land (\neg r \rightarrow (q \lor (\neg p \land r)))$

Scanned by CamScanner

Answers

Q1.

a) p : Sun shines today
q : Sun shines tomorrow

$p \rightarrow \neg q$

b) $p \rightarrow (q \vee \neg r)$

p : Request occurs
q : It is acknowledged
r : Requesting process made to process

c) $(p \wedge \neg q) \vee (\neg p \wedge q)$

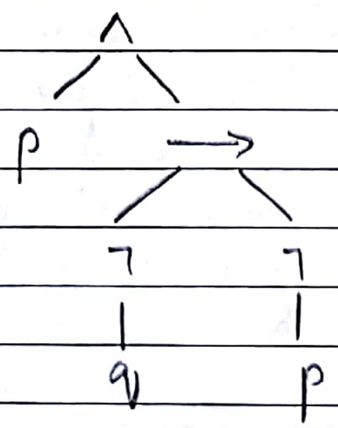
p : It will rain today
q : It will shine today

d) $p \rightarrow (q \vee r)$

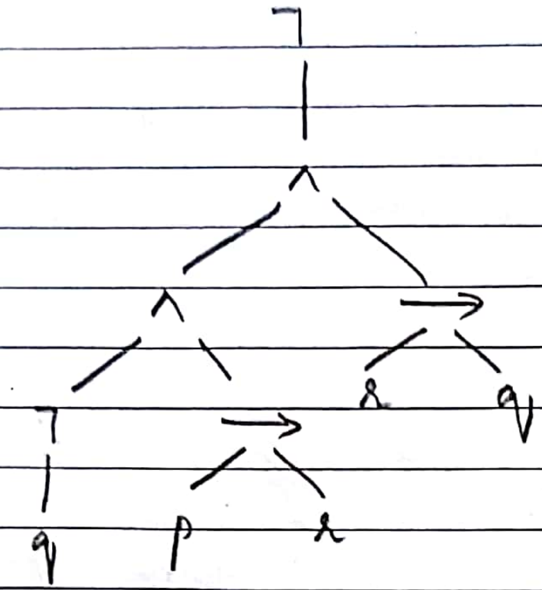
p : Jack met Jake yesterday
q : They have a cup of coffee together
r : They took a walk in the park.

Q2.

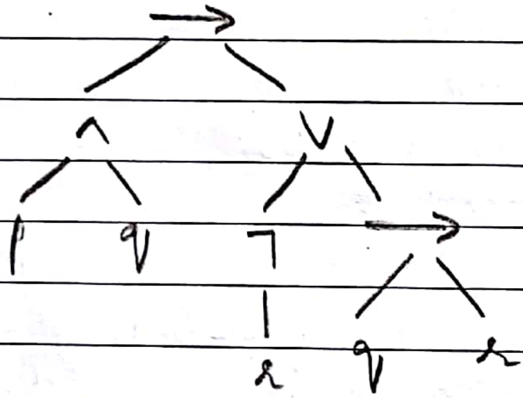
1.



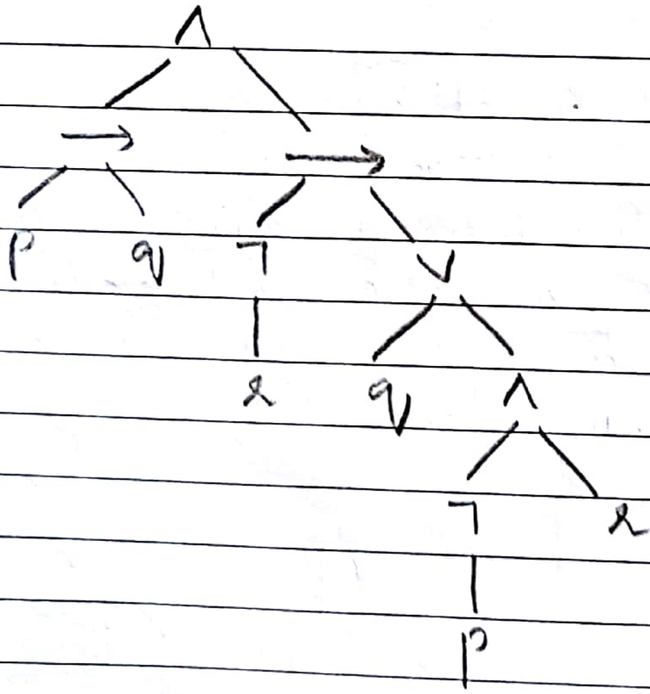
2.



3.



4.



14/09/'17.

PARTIAL INTERPRETATION

$p \vee q$

If it is known that p is TRUE then, irrespective of q value, the statement is true.

* Note: Antedecant → Consedent

Q. $p \to (q \vee r)$

p	q	r	$q \vee r$	$p \to q \vee r$
F	F	F	F	T
F	F	T	T	T
F	T	F	T	T
F	T	T	T	T
T	F	F	F	F
T	F	T	T	T
T	T	F	T	T
T	T	T	T	T

* A truth table for 'n' atoms is a table with $n+1$ columns and 2^n rows.
There is a column for each atom + a column for formula.
The first n columns specify the interpretation that maps an atom to either true or false.
The last column shown is a truth value for the interpretation.

* Inclusive OR ($\vee$) and Exclusive OR ($\oplus$)

$p \to$ I will go to theater } $p \oplus q$
$q \to$ I will go to park

$p_1 \to$ I will have popcorn } $p_1 \vee q_1$
$q_1 \to$ I will have juice }

* $\longleftrightarrow$ equivalence (boolean operator)

$\cong$ logical equivalence

LOGICAL EQUIVALENCE:

Let p and q belong to a formula f. If p is equal to q for all interpretation of the formula, then p is logically equivalent to q. ($p \cong q$)

SUBSTITUTION

$B : (p \to q) \longleftrightarrow (\neg q \to \neg p)$
$A : (p \to q)$
$A' : \neg p \vee q$
$B : (\neg p \vee q) \longleftrightarrow (\neg q \to \neg p)$
$B[A \leftarrow A']$

Let A be a sub formula of B and let A' be any formula. The substitution of A' for A in B $B[A \leftarrow A']$ is the formula obtained replacing all occurrances of the sub tree of A in B by A'.

16/09/17

Logical Equivalent Formula

1. Absorption of Constant

$p \wedge TRUE \cong p$

$p \vee TRUE \cong TRUE$

$p \wedge FALSE \cong FALSE$

$p \vee FALSE \cong p$

$p \rightarrow TRUE \cong TRUE$

$p \rightarrow FALSE \cong \neg p$

2. Identical Operands

$\neg\neg A \cong A$

$A \wedge A \cong A$

$A \vee \neg A \cong TRUE$

$A \wedge \neg A \cong FALSE$

$A \uparrow \neg A \cong TRUE$

A	¬A	A ∧ ¬A	¬(A ∧ ¬A)
T	F	F	T
F	T	F	T

3. Commutative, Associative and Distributive

$A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$

Satisfiability, Validity, Consequence

If for a set of formulas, the given formula is true for ~~all~~ some interpretation then it is satisfiable.

$U \models A$ valid statement

$U \not\models A$ unsatisfiable

Let F be a formula, A an atom.
Let $A \in F$.
A is satisfiable iff $\nu_f(A) = \text{TRUE}$ (for all formulas of A) for some interpretation in F.
A valid iff $\nu_f(A)$ is TRUE for all interpretation of F.
A is unsatisfiable iff for all ~~interpretation~~ formulas of A = FALSE for all interpretation of F.
A is falsifiable iff it is not valid.

Eg: $[p, \neg p \vee q, q \wedge r] = T \rightarrow$ Valid, Satisfiable
p, q, r is true

$[p, \neg p \vee q, q \wedge r]$

Q. Check the validity

$$[p, \neg q] \models (p \vee r) \wedge (\neg q \vee \neg r)$$

25/09/17

SEMANTIC TABLEAUX

1. Decomposing formula into a set of literals.
2. Constructing the semantic tableaux.
3. Termination of Tableaux Construction.
   X (F) ⊙ (T)

Eg : p : Jack will go to party
q : Jill will go to party
r : James will go to party
s : John will go to party

1st assumption : Jack and Jill or both will go to the party.
$p \vee q$

2nd : If ~~Jack~~ Jill goes to the party then James will go unless John goes.
$q \rightarrow (\neg r \rightarrow s)$

3rd : John will go if Jack doesn't go
$\neg p \rightarrow s$

Conclusion : James will go to the party.
r

26/09/17 Assumptions, $\Phi = \{(p \vee q),\ q \rightarrow (\neg r \rightarrow s),\ (\neg p \rightarrow s)\}$

$\Phi_\wedge = \{(p \vee q) \wedge q \rightarrow (\neg r \rightarrow s) \wedge (\neg p \rightarrow s)\}$

$\Phi_\wedge \models r$ (valid) $\implies \Phi_\wedge \wedge \neg r$ (unsatisfiable)

① To create the semantic tableaux, a statement should only have $\wedge$ and $\vee$.

<u>Conjunction</u>

α	α_1	α_2
$\neg\neg A_1$	A_1	
$A_1 \wedge A_2$	A_1	A_2
$\neg(A_1 \vee A_2)$	$\neg A_1$	$\neg A_2$
$\neg(A_1 \rightarrow A_2)$	A_1	$\neg A_2$
$\neg(A_1 \uparrow A_2)$	A_1	A_2
$(A_1 \downarrow A_2)$	$\neg A_1$	$\neg A_2$

α	α_1	α_2
$(A_1 \leftrightarrow A_2)$	$A_1 \to A_2$	$A_2 \to A_1$
$\neg(A_1 \oplus A_2)$	$A_1 \to A_2$	$A_2 \to A_1$

Disjunction

β	β_1	β_2
$\neg(B_1 \wedge B_2)$	$\neg B_1$	$\neg B_2$
$B_1 \vee B_2$	B_1	B_2
$B_1 \to B_2$	$\neg B_1$	B_2
$B_1 \uparrow B_2$	$\neg B_1$	$\neg B_2$
$\neg(B_1 \downarrow B_2)$	B_1	B_2
$\neg(B_1 \leftrightarrow B_2)$	$\neg(B_1 \to B_2)$	$\neg(B_2 \to B_1)$
$B_1 \oplus B_2$	$\neg(B_1 \to B_2)$	$\neg(B_2 \to B_1)$

Eg: 1) $p \wedge (\neg q \vee \neg p)$

$p, \neg q$ $\qquad$ $p, \neg p$

$\odot$ $\qquad$ $\times$

It is not valid

2) $(p \vee q) \wedge (\neg p \wedge \neg q)$

$(p \vee q), \neg p, \neg q$

$p, \neg p, \neg q$ $\qquad$ $q, \neg p, \neg q$

$\times$ $\qquad$ $\times$

It is valid.

Prev. qs

$(p \vee q) \wedge (q \to (\neg s \to r)) \wedge (\neg p \to s) \wedge \neg r$

$\Rightarrow (p \vee q) \wedge (q \to (s \vee r)) \wedge (p \vee s) \wedge \neg r$

$\Rightarrow (p \vee q) \wedge (\neg q \vee (s \vee r)) \wedge (p \vee s) \wedge \neg r$

$\downarrow$

$(p \vee q), (\neg q \vee (s \vee r)), (p \vee s), \neg r$

$\downarrow$

$p, \neg q, p, \neg r$

It is satisfiable. Hence it is not a valid conclusion.

28/09/17

Soundness & Completeness

* If a conclusion is sound, it is valid. But a valid statement may or may not be sound.

Eg: All men are mortals (T)
Socrates is a man (T)

Socrates is mortal (T)
↳ valid & sound

All organisms who have wings can fly (F)
Penguins have wings (T)

Penguins can fly (T)
↳ valid, not sound

Soundness means that a proposition must be valid if it has a semantic tableaux proof. (Complete tree structure is possible)

Completeness says that if a proposition is valid, we can find it in a tableaux proof.

03/10/17

Module - II

DEDUCTIVE SYSTEMS

* It is a set of formula called axioms and set of rules of inference.

A proof of deductive system is the sequence of formula

$$S = \{A_1, A_2, \ldots A_n\}$$

such that each formula A_i is either an axiom or it can be inferred from previous formula of the sequence using rules of inference.

A_n, the last formula of the sequence is a theorem.

The sequence S is a proof of A_n and A_n is provable denoted by $\vdash A_n$.

Note: $\vDash$ valid

$\vdash$ provable

If $\vdash A$ then A can be used as axiom in subsequent proof.

* A proof can be written as a sequence of set of formulas which are numbered for convenient reference. On the right of each line is its justification.

Justification is either the set of formula is an axiom or it is a conclusion of a rule of inference applied to a set or sets of formulas earlier in the sequence.

There are two types of deductive systems

① Gentzen System $\mathcal{G}$ (1 axiom, Many RF)
② Hilbert System $\mathcal{H}$ (1 RF, Many AXIOMS)

Q. Proof $\vdash (p \vee q) \rightarrow (q \vee p)$ $\mathcal{G}$

1. $\vdash \neg p, p, q$ Axiom
2. $\vdash \neg q, p, q$ Axiom
3. $\vdash \neg p, (q \vee p)$ $\beta\vee$, 1
4. $\vdash \neg q, (q \vee p)$ $\beta\vee$, 2
5. $\vdash \neg(p \vee q), (q \vee p)$ $\alpha\vee$, 3, 4
6. $\vdash (p \vee q) \rightarrow (q \vee p)$ $\beta\rightarrow$, 5

$\neg p, \neg q, \neg r$ $(p \vee q), (p \vee r)$

Q. $\vdash p \vee (q \wedge r) \rightarrow (p \vee q) \wedge (p \vee r)$ in $\mathcal{G}$

$(p \vee q)$

1. $\vdash \neg p, p, q$ Axiom
2. $\vdash \neg p, (p \vee q)$ $\beta\vee$, 1
3. $\vdash \neg p, p, r$ Axiom
4. $\vdash \neg p, (p \vee r)$ $\beta\vee$, 3
5. $\vdash \neg p, (p \vee q) \wedge (p \vee r)$ $\alpha\wedge$, 2, 4
6. $\vdash \neg q, \neg r, p, q$ Axiom
7. $\vdash \neg q, \neg r, (p \vee q)$ $\beta\vee$, 6
8. $\vdash \neg q, \neg r, p, r$ Axiom
9. $\vdash \neg q, \neg r, (p \vee r)$ $\beta\vee$, 8
10. $\vdash \neg q, \neg r, (p \vee r) \wedge (p \vee q)$ $\alpha\wedge$, 7, 9
11. $\vdash \neg(q \wedge r), (p \vee r) \wedge (p \vee q)$ $\beta\wedge$, 10
12. $\vdash \neg(p \vee (q \wedge r)), (p \vee r) \wedge (p \vee q)$ $\alpha\vee$, 5, 11
13. $\vdash p \vee (q \wedge r) \rightarrow (p \vee r) \wedge (p \vee q)$ $\beta\rightarrow$, 12

05/10/'17

MODUS PONENS

$$\frac{\vdash A \qquad \vdash A \to B}{\vdash B}$$

This is the only rule of inference used in Hilbert System.

Axioms (Assumptions)

1. $\vdash A \to (B \to A)$ — law of simplification
2. $\vdash A \to (B \to C) \to (A \to B) \to (A \to C)$ Frege's law
3. $\vdash (\neg B \to \neg A) \to A \to B$ law of Inverse

Q. Prove $\vdash A \to A$

Let $B = A \to A$ $\qquad$ $B \to A$

$C = A$ $\qquad$ $B = \neg A$

$\neg A$

1. $A \to ((A \to A) \to A)$ — Axiom 1
2. $A \to ((A \to A) \to A) \to (A \to (A \to A)) \to (A \to A)$ Axiom 2
3. $(A \to (A \to A)) \to (A \to A)$ MP 1, 2

Let $B = A$

4. $A \to (A \to A)$ Ax(1)
5. $(A \to A)$ MP 3, 4

Hilbert System

In Gentzen System, there is one axiom and many rules of inference while in a Hilbert System, there are several axioms but only one rule of inference.

The rule of inference is known as Modus ponens (MP). It is represented as

$$\frac{\vdash A \quad \text{and} \quad \vdash A \to B}{\vdash B}$$

Formula B can be inferred from A and $A \to B$.

A, B, C represents arbitrary formula in propositional logic.

Q. Prove $\vdash ((A \to B) \to A) \to A$

$(A \to B) \to A \vdash A$

$H_1 : (A \to B) \to A$

Axiom 2 : $A \to (B \to C) \to (A \to B) \to (A \to C)$

Put $C = A$

MP on Axioms 1, 2

$(A \to B) \to (A \to A)$

09/10/17 Desired Rules (Assignment)

1) Contrapositive Rule

$$\frac{U \vdash \neg B \to \neg A}{U \vdash A \to B}$$

2) Transitivity Rule

$$\frac{U \vdash A \to B \quad U \vdash B \to C}{U \vdash A \to C}$$

3) Exchange of Antecedent Rule

$$\frac{U \vdash A \to (B \to C)}{U \vdash B \to (A \to C)}$$

Camlin | Page
Date / /

4) Double Negation

$$\frac{U \vdash \neg\neg A}{U \vdash A}$$

Resolution in Propositional Logic

Imp * Inference Rule

$$\frac{\begin{array}{c} P \vee Q \\ \neg P \vee R \end{array}}{\therefore\ Q \vee R}$$

* $p \rightarrow q \iff \neg p \vee q$

* $p \iff q$
$(\neg p \vee q) \wedge (\neg q \vee p)$ OR $(p \rightarrow q) \wedge (q \rightarrow p)$

Conversion to CNF

$P \leftrightarrow (Q \vee R)$

1. Eliminate $\leftrightarrow$, replace $\alpha \leftrightarrow \beta$ with $(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$
$(P \rightarrow (Q \vee R)) \wedge ((Q \vee R) \rightarrow P)$
2. Eliminate $\rightarrow$ replacing $\alpha \rightarrow \beta$ as $\neg\alpha \vee \beta$
$(\neg P \vee Q \vee R) \wedge (\neg(Q \vee R) \vee P)$
3. Move $\neg$ inwards using de Morgan's rules and double negation
$(\neg P \vee Q \vee R) \wedge ((\neg Q \wedge \neg R) \vee P)$
4. Apply distributive law ($\wedge$ over $\vee$) and flatten
$(\neg P \vee Q \vee R) \wedge ((\neg Q \vee P) \wedge (\neg R \vee P))$

Camlin Page
Date / /

10/10/17 Transform the set of formulas $[p, p \to ((q \vee r) \wedge \neg(q \wedge r)), p \to ((s \vee t) \wedge \neg(s \wedge t)), s \to q, \neg r \to t, t \to s]$ into clausal form and refute using resolution:

1. p
2. $p \to ((q \vee r) \wedge \neg(q \wedge r))$
   $\neg p \vee ((q \vee r) \wedge \neg(q \wedge r))$
   $\neg p \vee ((q \vee r) \wedge (\neg q \vee \neg r))$
   $(\neg p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$
3. $p \to ((s \vee t) \wedge \neg(s \wedge t))$
   $\neg p \vee ((s \vee t) \wedge \neg(s \wedge t))$
   $\neg p \vee ((s \vee t) \wedge (\neg s \vee \neg t))$
   $(\neg p \vee s \vee t) \wedge (\neg p \vee \neg s \vee \neg t)$

4. $s \to q$
   $\neg s \vee q$

5. $\neg r \to t$
   $r \vee t$

6. $t \to s$
   $\neg t \vee s$

These are clausal forms.
If it becomes empty, it is valid & otherwise it is satisfiable.

1. p
2. $\neg p \vee q \vee r$
3. $\neg p \vee \neg q \vee \neg r$
4. $\neg p \vee s \vee t$
5. $\neg p \vee \neg s \vee \neg t$
6. $\neg s \vee q$
7. $r \vee t$

8. $\neg t \vee s$

(At a point of time, only 1 literal & its negation must be present. Only 1 literal can be removed at a time!)

From 5 and 7

9. $\neg p \vee \neg s \vee r$

4, 8

10. $\neg p \vee s$

9, 10

11. $\neg p \vee r$

3, 11.

12. $\neg p \vee \neg q$

6, 12

13. $\neg p \vee \neg s$

10, 13.

14. $\neg p$

14, 1

15. NULL

Binary Decision Diagram

Q. $p \to (q \wedge r)$

p	q	r	$p \to (q \wedge r)$
T	T	T	T
T	T	F	F
T	F	T	F
T	F	F	F
F	T	T	T
F	T	F	T
F	F	T	T
F	F	F	T

By combining certain rows, we get

p	q	r	$p \to (q \wedge r)$
T	T	T	T
T	T	F	F
T	F	*	F
F	T	*	T
F	F	*	T

Q. $p \vee (q \wedge r)$

p	q	r	$p \vee (q \wedge r)$
T	T	T	T
T	T	F	T
T	F	T	T
T	F	F	T
F	T	T	T
F	T	F	F
F	F	T	F
F	F	F	F

p	q	r	$p \vee (q \wedge r)$
T	*	*	T
F	T	T	T
F	T	F	F
F	F	*	F

Camlin Page
Date / /

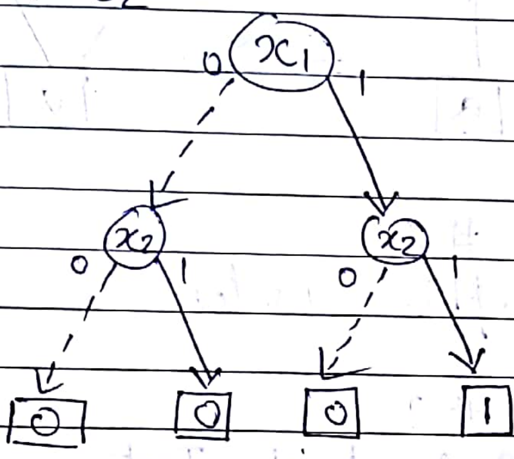
14/10/17

Binary Decision Tree:

0 : - - - - >

1 : ——>

1) $x_1 \wedge x_2$



Here, decision nodes = 3, leaf = 4.

In general,

decision nodes = $2^n - 1$

leaf = 2^n

BDD : It is a data structure for representing semantics of a formula in propositional logic. A formula is represented by a direct graph & an algorithm is used to reduce the graph. The reduced graph have the property that is same as the graph of logical equivalence formula.

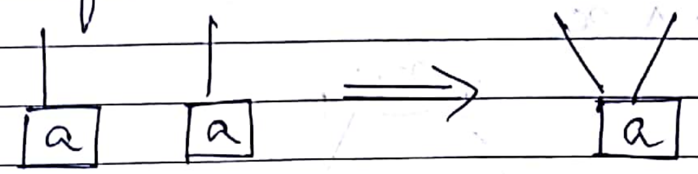
For an 'n' variable, we create $2^n - 1$ decisions nodes & 2^n links for the lowest leaf nodes.

0 is represented as - - ->, 1 as ⟶

* Ordered format $x_1, x_2, \ldots x_n$.
* Ordered BDD (1st condition to reduce a BDD)

Reduction Rule #1

Merge equivalent leaves

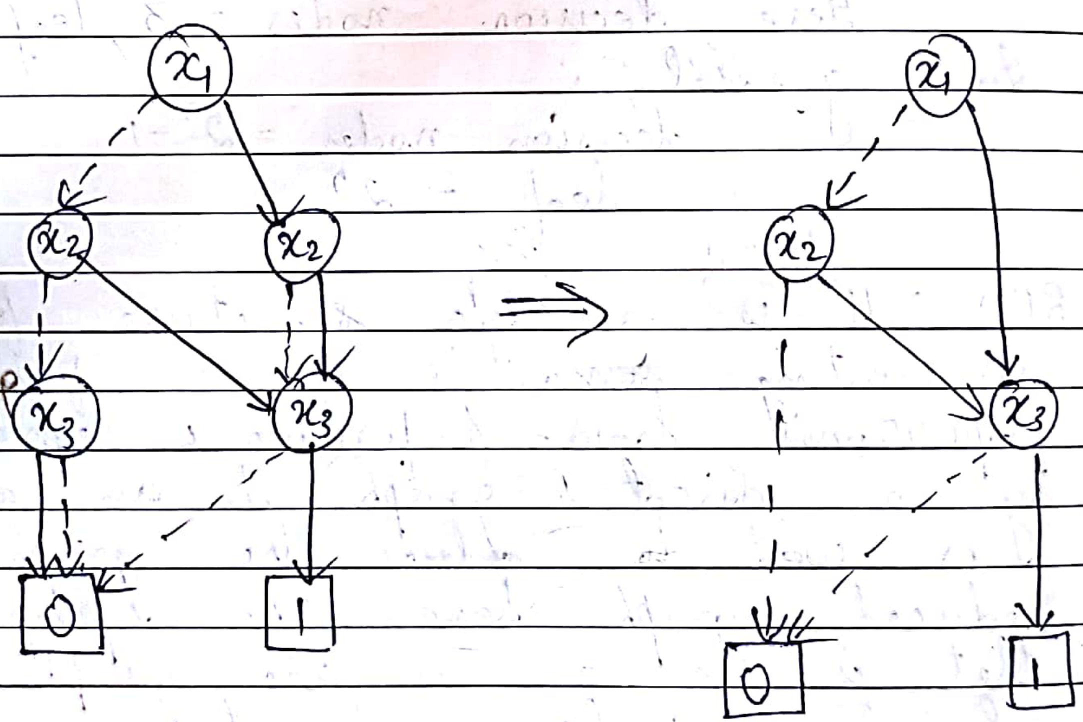


Reduction Rule #2

Merge isomorphic nodes.

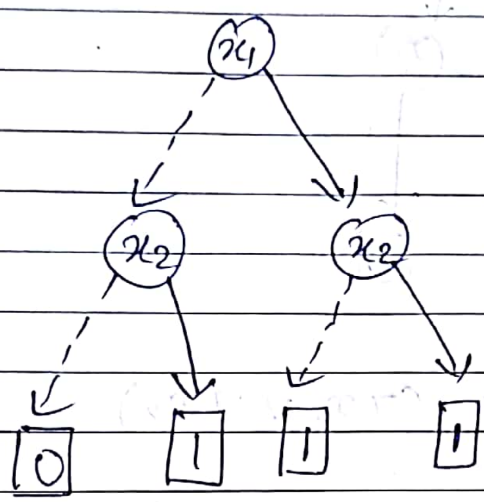
Reduction Rule #3:

Eliminate Redundant Tests

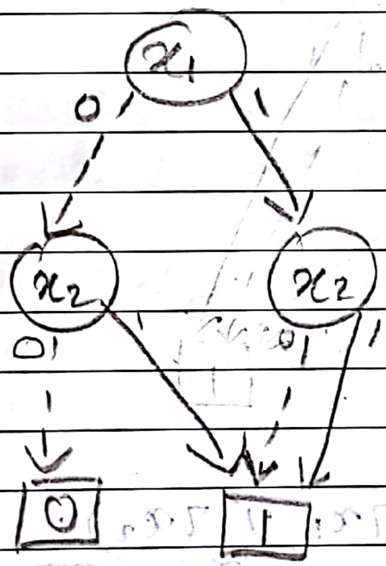


Q. $(x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2)$

$x_1 \vee x_2$



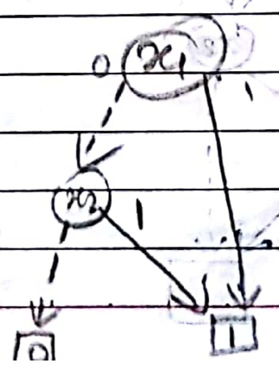
Rule #1



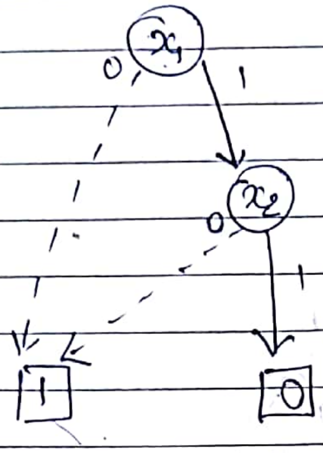
Rule #2.

No isomorphic nodes.

Rule #3.



$\neg x_1 \vee \neg x_2$

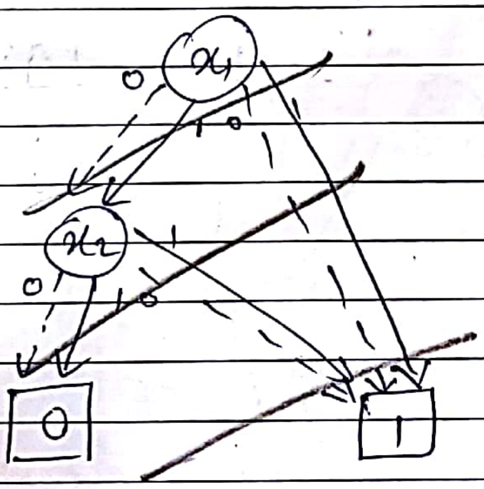


$(x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2)$

17/10/17

Merge

If roots are same check left nodes



$x_1 = 0 \quad x_2 = 0 \quad F(0) \wedge T = F$

$x_1 = 0 \quad x_2 = 1 \quad T \wedge T = T$

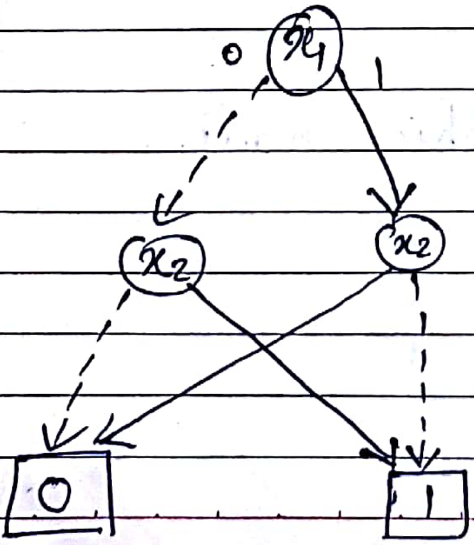
Right

$x_1 = 1 \quad T \wedge F \quad x_1 = 1 \; x_2 = 1$

$x_1 = 1 \quad T \wedge T \quad x_1 = 1 \; x_2 = 0$

$(\overset{0}{x_1} \vee \overset{0}{x_2}) \wedge (\neg \overset{1}{x_1} \vee \neg \overset{1}{x_2})$ [XOR]

$\cong x_1 \oplus x_2$

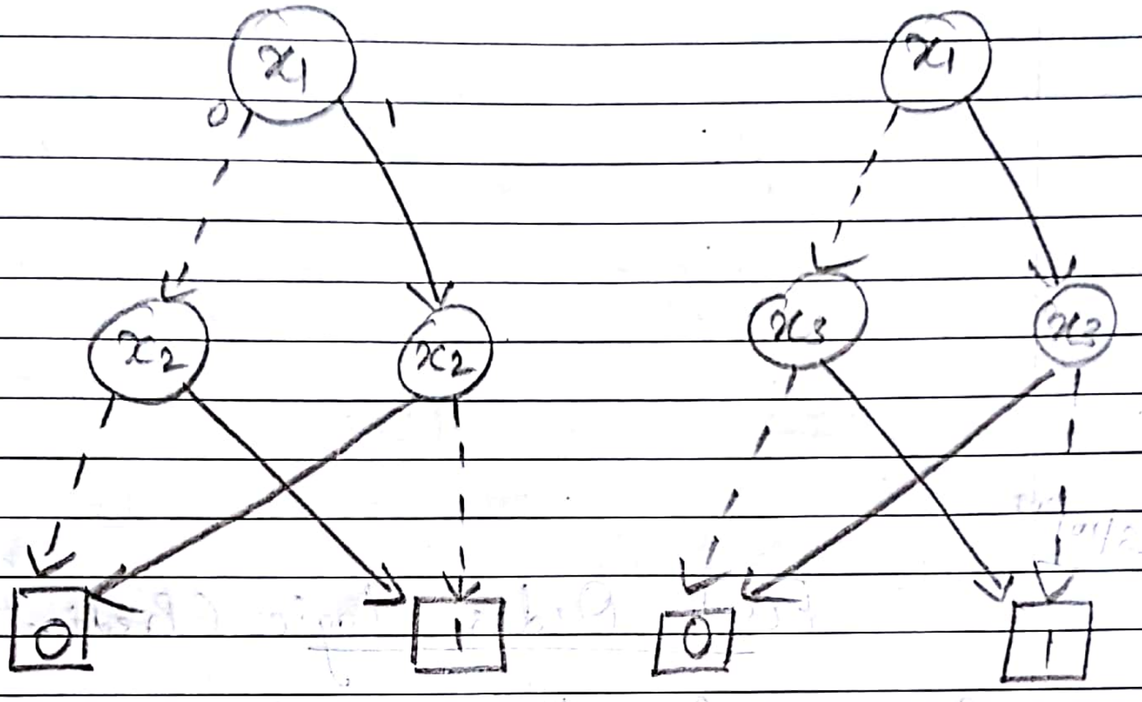


(ROBDD)

Q. $(x_1 \oplus x_2) \oplus (x_1 \oplus x_3)$

$(x_1 \oplus x_2) \oplus (x_3 \oplus x_1)$ → This is a complex process so we make $x_3 \oplus x_1 = x_1 \oplus x_3$

$x_1 \oplus x_2$ $\qquad\qquad$ $x_1 \oplus x_3$



Already reduced.

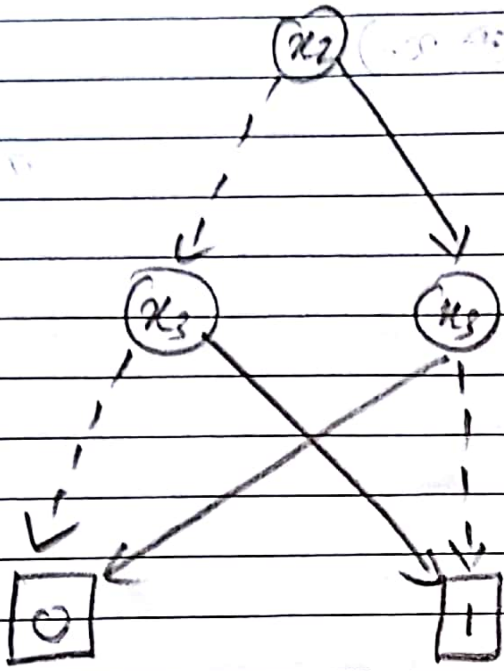
Here, the left nodes are different

$x_1 = 0$	$x_2 = 0$	F	$\oplus$	$x_3 = 0$	F	F
$x_1 = 0$	$x_2 = 0$	F	$\oplus$	$x_3 = 1$	T	T
	$x_2 = 1$	T	$\oplus$	$x_3 = 0$	F	T
	$x_2 = 1$	T	$\oplus$	$x_3 = 1$	T	F

Right

$x_2 = 0$	T	$\oplus$	$x_3 = 0$	T	F
$x_2 = 0$	T	$\oplus$	$x_3 = 1$	F	T
$x_2 = 1$	F	$\oplus$	$x_3 = 0$	T	T
$x_2 = 1$	F	$\oplus$	$x_3 = 1$	F	F

$\therefore$ We get $(x_1 \oplus x_2) \oplus (x_1 \oplus x_3) = (x_2 \oplus x_3)$



23/10/'17

First-Order Logic (Predicate Logic)

* Propositional logic : Statements can take only true or false values.
* Propositional logic + quantifier function ≡ first-order logic.
* It is an extension of propositional logic that includes predicates, relation, domain & quantifiers.

$\forall$: for all (universal)
$\exists$: there exists (existential)

Eg: $I = \{1, 2, 3, 4 \ldots\}$

$P = \{2, 3, 5 \ldots\}$

$P \subset I$.

$\forall x \; P(x)$ = Prime
↓
variable.

Camlin Page
Date / /

Square of prime numbers
$\forall P(x,y)$
$y = x^2$ $\quad x \geq 2$
$P(2), P(3)$ exists $\quad P(4) = \text{False}$

* Predicate can have variables or constant values.

Eg: All mammals having wings can fly.
$\forall x, \forall y \ \text{mammal}(x) \wedge \text{wings}(y) \Rightarrow \text{fly}(x,y)$
[Note: Not all mammals have wings].

FOL has
1) Objects : things with individual identities.
Eg: Students, lectures, cars etc.

2) Relations: Holds among set of objects
Eg: Brother of, taller than
Jim is taller than Jack taller(Jim, Jack)

3) Properties
4) Functions
Eg: color-of(Sky) = Blue

* Predicate symbols
Eg: greater(5, 3)
color(grass, green)

Scanned by CamScanner

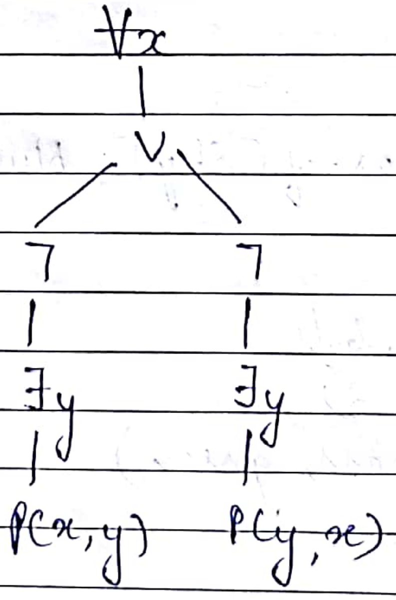
Formal Grammar for Formula

Formula ::= Atomic formula
Formula ::= $\neg$ Formula
Formula ::= Formula op Formula
Formula ::= $\forall x$ formula
Formula ::= $\exists x$ formula

argument ::= x $\quad$ $x \in$ variable
argument ::= a $\quad$ $a \in$ constant
argument ::= argument
argument_list ::= argument , argument_list
atomic formula ::= P (argument_list)

Note : $\exists, \forall$ has same precedence as $\neg$.

Q. $\forall x (\neg \exists y \, P(x,y) \vee \neg \exists y \, P(y,x))$



Camlin Page
Date / /

Q. All kings are persons

$\forall x \; king(x) \Rightarrow Person(x)$

Note: $\forall x \; King(x) \wedge person(x)$ means all people are kings.

24/10/17

* A term is a constant symbol, variable symbol, or an n-place function of n terms.
* A complex sentence is formed from atomic sentences.

* $(\forall x) P(x)$: P holds for all values of x in the domain.
* $(\exists x) P(x)$: P holds for some values of x in the domain
* If there is atleast 1 false condition, it becomes existential.
* Only if there is atleast 1 true condition, it can be existential.

* $(\forall x)$ student $(x) \rightarrow$ smart (x)
  "All students are smart".
* $(\forall x)$ student
  Everyone in the world is a student and is smart.

* We can switch the order of
  1) universals only
  2) existential only
* Switching $\forall$ & $\exists$ does change meaning.

Camlin Page
Date / /

Note:
Existential $\wedge$
Universal $\rightarrow$

* De-Morgan's laws:

$$(\forall x)\neg P(x) \leftrightarrow \neg(\exists x)P(x)$$
$$\neg(\forall x)P \leftrightarrow (\exists x)\neg P(x)$$
$$(\forall x)P(x) \leftrightarrow \neg(\exists x)\neg P(x)$$
$$(\exists x)P(x) \leftrightarrow \neg(\forall x)\neg P(x)$$

* Quantified inference rules

1 * Universal instantiation
$$(\forall x)P(x) \therefore P(A)$$

Q1. Every gardener like the sun.
$$\forall x\, g(x) \rightarrow like(x, Sun)$$

2. You can fool some of the people all of the time.
$$(\exists x)(\forall y)\ person(x) \wedge time(y) \rightarrow can\_fool(x,y)$$

3. You can fool all of the people some of the time.
$$(\forall x)(\exists y)(person(x) \rightarrow time(y) \wedge can\_fool(x,y))$$
$$\forall x(person(x) \rightarrow \exists t(time(t) \wedge can\_fool(x,t)))$$

4. All purple mushrooms are poisonous.
$$\forall x(mushroom(x) \wedge purple(x)) \rightarrow poisonous(x)$$

5. No purple mushroom is poisonous.

$$\forall x(mushroom(x) \wedge purple(x)) \rightarrow \neg poisonous(x)$$

Scanned by CamScanner

6. There are exactly two purple mushrooms.

$(\forall x)(\text{mushroom}(x) \wedge \text{purple}(x)) \rightarrow \text{two}(x)$

7. Clinton is not tall

$\neg \text{tall}(\text{Clinton})$

* Model : Every sentence is true.

26/10/17

Semantic Tableaux

1. Instantiate Universal formula with constants.

$$\forall x\,(P(x) \rightarrow q(x)) \rightarrow (\forall x\, P(x) \rightarrow \forall x\, q(x)) \quad \text{— (1)}$$

Suppose if $P(x) = \{a_1, a_2, a_3, \dots a_n\}$, then it is difficult to verify for all cases in (1).

So we convert it to $\exists$.

We know $A_1 \rightarrow A_2 = \neg A_1 \vee A_2$

$\neg(A_1 \rightarrow A_2) = A_1 \wedge \neg A_2$

$\forall x\,(P(x) \rightarrow q(x)),\ \neg(\forall x\, P(x) \rightarrow \forall x\, q(x))$

$\downarrow$

$\forall x\,(P(x) \rightarrow q(x)),\ \forall x\, P(x),\ \neg\forall x\, q(x)$

$\downarrow$

$\forall x\,(P(x) \rightarrow q(x)),\ \forall x\, P(x),\ \exists x\, \neg q(x)$

$\downarrow$

$P(a_1) \rightarrow q(a_1),\ P(a_1),\ \neg q(a_1)$

$\downarrow$

$\neg P(a_1) \vee q(a_1),\ P(a_1),\ \neg q(a_1)$

$\swarrow \qquad \searrow$

$\neg P(a_1), P(a_1), \neg q(a_1) \qquad q(a_1), P(a_1), \neg q(a_1)$

$\downarrow \qquad \downarrow$

$\otimes \qquad \otimes$

Camlin Page
Date / /

Q. $\neg(\forall x(p(x) \vee q(x)) \longrightarrow \forall x\, p(x) \vee \forall x\, q(x))$

$\downarrow$

$\forall x(p(x) \vee q(x)),\ \neg(\forall x\, p(x) \vee \forall x\, q(x))$

$\downarrow$

$\forall x(p(x) \vee q(x)),\ \neg\forall x\, p(x),\ \neg\forall x\, q(x)$

$\downarrow$

$\forall x(p(x) \vee q(x)),\ \exists x\, \neg p(x),\ \exists x\, \neg q(x)$

(we can't give the same constant as there are different $\exists$ statements)

$\downarrow$

$p(a_2) \vee (q(a_1)),\ p(a_2),\ q(a_1)$

2. Don't use the same constant twice to instantiate the existential formula.

3. A branch may not terminate.

$\forall x\, \exists y\, p(x,y)$

$\downarrow$

$\forall x\, \exists y\, p(x,y),\ \exists y\, p(a_1, y)$

$\downarrow$

$\forall x\, \exists y\, p(x,y),\ \exists y\, p(a_1, y),\ p(a_1, a_2)$

* Here x can take lots of values. Hence we write general case and one example.

Camlin Page

Date / /

$$\exists y \, \forall x \, p(x,y)$$
$$\downarrow$$
$$\forall x \, p(x, a_1)$$
$$\neg \forall x \, p(x, a_1)$$
$$\exists x \, \neg (p(x, a_1))$$
$$\neg p(a_2, a_1)$$

28/10/17

Q. $\forall x \, (p(x) \to q(x)) \to (\forall x \, p(x) \to \forall x \, q(x))$ and its converse. Check if both are valid.

$$\neg\Big(\forall x \, (p(x) \to q(x)) \to (\forall x \, p(x) \to \forall x \, q(x))\Big)$$
$$\downarrow$$
$$\forall x \, (p(x) \to q(x)) \wedge \neg (\forall x \, p(x) \to \forall x \, q(x))$$
$$\downarrow$$
$$\forall x \, (p(x) \to q(x)), \; \forall x \, p(x), \; \neg \forall x \, q(x)$$
$$\downarrow$$
$$\forall x \, (p(x) \to q(x)), \; \forall x \, p(x), \; \exists x \, \neg q(x)$$
$$\downarrow$$
$$p(a_1) \to q(a_1), \; p(a_1), \; \neg q(a_1)$$
$$\downarrow$$
$$\neg p(a_1) \vee q(a_1), \; p(a_1), \; \neg q(a_1)$$

$$\neg p(a_1), \; p(a_1), \; \neg q(a_1) \qquad q(a_1), \; p(a_1), \; \neg q(a_1)$$
$$\downarrow \qquad\qquad \downarrow$$
$$\times \qquad\qquad \times$$

Since both are $\times$, this is a valid proof.

Converse :

$$\neg((\forall x\, p(x) \to \forall x\, q(x)) \to \forall x\,(p(x) \to q(x)))$$
$$\downarrow$$
$$\forall x\, p(x) \to \forall x\, q(x),\ \neg \forall x\,(p(x) \to q(x))$$
$$\downarrow$$
$$\neg \forall x\, p(x) \vee \forall x\, q(x),\ \neg \forall x\,(p(x) \to q(x))$$
$$\downarrow$$
$$\exists x\, \neg p(x) \vee \neg \exists x \neg q(x),\ \neg \forall x\,(\neg p(x) \vee q(x))$$
$$\downarrow$$
$$\exists x\, \neg p(x) \vee \neg \exists x\, \neg q(x)),\ \forall x\, p(x) \wedge \forall x\, \neg q(x)$$
$$\downarrow$$
$$\exists x\, \neg p(x)\ \ \neg p(a_1) \vee \neg q(a_2),\ p(a_1),\ \neg q(a_2)$$

$$\neg p(a_1),\ p(a_1),\ \neg q(a_2) \qquad \neg q(a_2),\ p(a_1),\ q(a_2)$$
$$\downarrow \qquad\qquad\qquad \downarrow$$
$$\times \qquad\qquad\qquad \odot$$

Converse is false.

γ Rules | **δ Rules**

γ	$\gamma(a)$	δ	$\delta(a)$
$\exists_x A(x)$	$A(a)$	$\forall_x A(x)$	$A(a)$
$\neg\forall_x A(x)$	$\neg A(a)$	$\neg\exists_x A(x)$	$\neg A(a)$

Camlin Page

Date / /

Q. $\forall x (p(x) \to q(x)) \to (\forall x\, p(x) \to \forall x\, q(x))$

30/10/17

Hilbert Model For FOL

1. $\vdash A \to (B \to A)$
2. $\vdash A \to (B \to C) \to ((A \to B) \to (A \to C))$
3. $\vdash (\neg B \to \neg A) \to (A \to B)$
4. $\vdash \forall x\, A(x) \to A(a)$
5. $\vdash \forall x\, (A \to B(x)) \to (A \to \forall x\, B(x))$

(1–3) PC

(4–5) Axiom

$$MP: \frac{\vdash A \to B \quad \vdash A}{\vdash B}$$

* If we have $\exists$ in our formula, we must convert it to universal formula.

Generalization

$$\frac{U \vdash A(a)}{U \vdash \forall x\, A(x)}$$

PC

↓

Propositional Deduction (1, 2, 3)

4, 5 - Axiom

$$\vdash \underbrace{\forall x\, p(x)}_{A} \to (\underbrace{\exists y\, \forall x q(x, y)}_{B} \to \underbrace{\forall x\, p(x)}_{A})$$

PC1

Camlin Page
Date / /

Q. $\vdash \neg\forall x\, A(x)$
$\vdash U \vee \forall x\, A(x) \vee A(a)$
$\vdash U \vee \neg\forall x\, A(x) \vee A(a)$
$\vdash U \vee \neg\forall x\, A(x) \vee \neg A(a)$
$\vdash U \vee \neg\forall x\, A(x)$
$\vdash \neg\forall x\, A(x) \rightarrow$ Sound.

If not specified x can take any
$U \vdash A(a)$
~~$U \vdash \forall x\, A(x)$~~

If $A(x)$ is Even(x), x can take only even numbers.

31/10/17

Q. $\vdash \forall x\, A(x) \rightarrow \exists x\, A(x)$

$\forall x\, A(x) \vdash \forall x\, A(x)$
$\forall x\, A(x) \vdash A(a)$
$\forall x\, A(x) \vdash A(a) \rightarrow \exists x A(x)$
$\forall x\, A(x) \vdash \exists x\, A(x)$ MP
$\vdash \forall x\, A(x) \rightarrow \exists x\, A(x)$

* If universal set is not defined $\neg A(x)$ is empty

$U \vdash A(a)$
$U \vdash \forall x\, A(x)$
For $\exists$

Q. $\vdash \exists x\, \forall y\, A(x,y) \rightarrow \forall y\, \exists x\, A(x,y)$

$\exists x\, \forall y\, A(x,y) \vdash \exists x\, \forall y\, A(x,y)$

$\vdash A(a,b) \rightarrow \exists x\, A(x,b)$
$\vdash \forall y\, A(a,y) \rightarrow \forall y\, \exists x\, A(x,y)$
$\vdash \neg(\forall y\, \exists x\, A(x,y)) \rightarrow \neg(\forall y\, A(a,y))$
$\vdash \forall x(\neg(\forall y\, \exists x\, A(x,y)) \rightarrow \neg(\forall y\, A(x,y)))$
$\vdash \neg\forall y\, \exists x\, A(x,y) \rightarrow \forall x\, \neg\forall y\, A(x,y)$
$\vdash \neg\forall x\, \neg\forall y\, A(x,y) \rightarrow \forall y\, \exists x\, A(x,y)$
$\vdash \exists x\, \forall y\, A(x,y) \rightarrow \forall y\, \exists x\, A(x,y)$

Scanned by CamScanner

02/11/17

FOL → Functions and Terms

term := x x is a variable
term := a a is a constant
term := f^0 0-arity function
term := f^n(term list) n-arity function
term list := terms
term list := term, term_list
atomic formula := P(term_list)

* arity means number of terms

$$f^n(t_1, t_2, t_3, \dots t_n)$$

$$a, x, f^1(x), f^2(a, b)$$

Functions and terms

Let F be a countable set of function symbols where each fn. symbol has an arity denoted by a super script. Terms are defined recursively as follows (as shown above)

Term can be a variable, constant, 0-arity function.

If f^n is an n array function symbol then $t_1, t_2, t_3, \dots t_n$ are its corresponding terms.

Notations:

Super script denoting the arity of

function are not used. (Eg $f(a,b)$)
We drop the word symbol and call it as a function.
By convention, functions is denoted by $\{f, g, h\}$
(Some fns. may have sub script Eg: f_I)
Constant symbols are not needed.

Interpretation

It is a 4-tuple.

$$\{\underset{\text{Domain}}{D}, \underset{\text{Relations}}{\{R_1 \ldots R_L\}}, \underset{\text{functions}}{\{f_1, f_2, \ldots f_n\}}, \underset{\text{constants}}{\{a_1 \ldots a_m\}}\}$$

Consider $\{I, \{\leq\}, \{+\}, 1\}$

$$D_I = f_1(t_1, t_2, t_3 \ldots t_n) = f_1(D_I(t_1), D_I(t_2), \ldots D_I(t_n))$$

$$f(x, a)$$
$$= +(D_I(x), D_I(a))$$
$$= +(D_I(x), D_I(1))$$
$$= +(m, 1)$$
$$= m+1.$$

$x \Rightarrow$ give any arbitrary value

If we give x as n, we get $n+1$

$\therefore \quad m+1 \leq n+1$

That is $m \leq n$

Q. What is the interpretation for $\forall x \exists y P(x,y)$
Here, we cannot assign a constant value to y.

For all x, there exists some $y \Rightarrow$
$y = f(x)$

$\therefore$ We get $\forall x\, P(x, f(x))$ Scholum

26/11/17

Ground Term: It is a term which does not contain any variables. Eg: $f(a)$

Ground atomic formula: It is an atomic formula all of whose terms are ground.

Ground literals: It is a ground atomic formula or its negation.

Ground formula: It is a quantifier free formula all of whose atomic formula is ground.

p: atom
$p(x, y)$: atomic formula
$p(x, y) \vee q(x, y)$: formula

Skolem's Algorithm

Clausal Normal Form

$$\forall x\, (p(x) \rightarrow q(x)) \rightarrow (\forall x\, p(x) \rightarrow \forall x\, q(a))$$

1) Rename bound variables so that no variables appear in two quantifiers.

$$\forall x\, (p(x) \rightarrow q(x)) \rightarrow (\forall y\, p(y) \rightarrow \forall z\, q(z))$$

2) Eliminate all binary boolean operators other than OR & AND

$$\neg \forall x\, (\neg p(x) \vee q(x)) \vee (\neg \forall y\, p(y) \vee \forall z\, q(z))$$

3) Push $\neg$ operators inward.

$$\exists x\, \neg(\neg p(x) \vee q(x)) \vee (\neg \forall y\, p(y) \vee \forall z\, q(z))$$
$$\exists x\, (p(x) \wedge \neg q(x)) \vee (\cancel{\neg \forall y\, p}\ \exists y\, \neg p(y) \vee \forall z\, q(z))$$

Scanned by CamScanner

Camlin Page
Date / /

4> Move the quantifiers as prefix of formula

$\exists x \exists y \forall z \left((p(x) \wedge \neg q(x)) \vee \neg p(y) \vee q(z) \right)$

5) Apply distributive law after assigning constants to existential quantifiers.

$\cancel{\forall z ((p(a) \wedge \neg q(b)) \vee \neg p(c) \vee q(z))}$

07/11/17

$\forall z (p(a) \vee \neg p(b) \vee q(z) \wedge \neg q(a) \vee \neg p(b) \vee q(z))$

Clausal form is

$[\{ p(a) \vee \neg p(b) \vee q(z) \}, \{ \neg q(a) \vee \neg p(b) \vee q(z) \}]$

To represent in form of functions, PCNF, till step 4 everything remains same.

$x = f(z)$

$y = \cancel{f}\ g(z)$

$\forall z (p(f(z)) \vee \neg p(g(z)) \vee q(z) \wedge \cancel{\forall p}\ \neg q(f(z)) \vee \neg p(g(z)) \vee q(z))$

PCNF is

$[\{ p(f(z)) \vee \neg p(g(z)) \vee q(z) \}, \{ \neg q(f(z)) \vee \neg p(g(z)) \vee q(z) \}]$

Camlin Page
Date / /

Note: Propositional logic

p — Either true or false.

q ...

Predicate logic

father (x)

son (x, y)

To prove parent (x,y) : (father(x) $\vee$ mother(x)) $\wedge$ child(y,x)

Using fns.

parent(x,y) : (father(x) $\vee$ mother(x)) $\wedge$ child$(f(x), x)$

If x is assigned a name, $f(x)$ contains only the no. of children of x.

parent(x,y) : modal

When we represent using a single variable & fns. of that variable it is called Herbrand model and its universe is Herbrand's Universe.

Q. Find PCNF (Prenex CNF)

$$\forall x\,(p(x) \rightarrow \exists y\, q(y))$$

$$\forall x\,(\neg p(x) \vee \exists y\, q(y))$$

$$\forall x\, \exists y\,(\neg p(x) \vee q(y))$$

$$y = f(x)$$

$$\forall x\,(\neg p(x) \vee q(f(x)))$$

Q. $\forall x \forall y\,(\exists z\, p(z) \wedge \exists u\,(q(x,u) \rightarrow \exists v\, q(y,v)))$

$$\forall x \forall y\,(\exists z\, p(z) \wedge \exists u\,(\neg q(x,u) \vee \exists v\, q(y,v)))$$

$\forall x \forall y \exists z \exists u \exists v (p(z) \wedge (\neg q(x,u) \vee q(y,v)))$

$z = f(x,y)$
$u = g(x,y)$
$v = h(x,y)$

$\forall x \forall y (p(f(x,y)) \wedge (\neg q(x, g(x,y)) \vee q(y, h(x,y))))$

09/11/17

$y = f(x)$
$\$ \rightarrow p(x, f(x))$
$H_s \Rightarrow a, f(a), f(g(a)), f(g(g(a)))\ldots$

Herbrand's Universe

Let S be a set of clauses. A - set of constant symbol in S. F - set of functions in S. H_s - Herbrand's universe in S.

Interpretation $a_l \in H_s$ iff $a_l \in A$

$f_l^0 \in H_s$ iff $f_l^0 \in f$.

$f_l^n(t_1, t_2, t_3, \ldots t_n) \in H_s$ for $n \geq 1$, $f_l^n \in f$.

Q. Find Herbrand's Universe of

i) $S = [\{p(a), \neg p(b), q(z)\}, \{\neg p(b), \neg q(z)\}]$

$H_s = \{a, b\}$

ii) $S = [\ \{\neg p(x, f(y))\}, \{p(w, g(w))\}\ ]$

$$H_S = \{a, f(a), g(a), f(a, g(a)), f(g(a)), g(g(a)), \ldots\}$$

Camlin Page
Date / /

13/11/17

Module - V

Temporal logic

- Completely based on time

* $\square \rightarrow$ Always
  $\diamond \rightarrow$ Eventually / Sometimes

Q. After the reset line of a flip flop is asserted, the zero line is asserted.

reset_line (t_1)
zero_line (t_2)

FOL: $\forall t_1 ((reset\_line(t_1)) \rightarrow \exists t_2 (t_2 \geq t_1 \wedge zero(t_2)))$

Temporal:

$$\square (reset \rightarrow \diamond zero)$$

Eg: 7 is a prime number - Always true
The head of state of a country is a monarch - Eventually true.

Note: Temporal Logic
It is a formal s/m for reasoning about time.
Syntax: It uses all the syntax of propositional logic & two additional operators
$\square$ Always
$\diamond$ Eventually

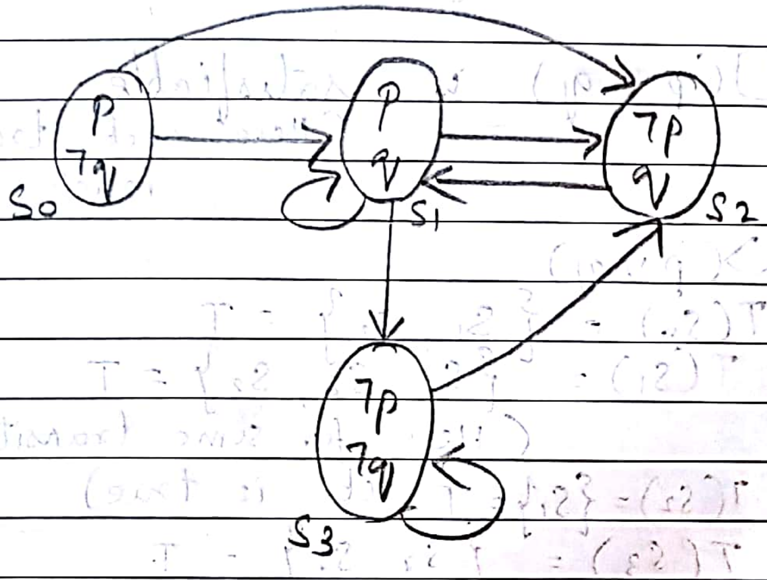
Camlin Page
Date / /

Semantics:

* Always means for any time t in the future.
* Eventually means for some time t in the future.

Interpretation

State transition diagram.



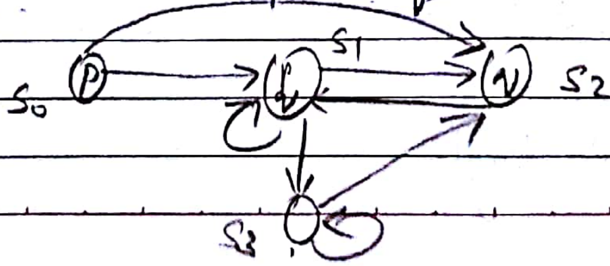
$S_2 \in P(S_1)$
Transition from S_1 to S_2

$S_0(P) = T$, $S_0(q) = F$
$S_1(p) = T$, $S_1(q) = T$
$S_2(p) = F$, $S_2(q) = T$
$S_3(p) = F$, $S_3(q) = F$

$T(S_0) = \{S_1, S_2\}$
$T(S_1) = \{S_1, S_2, S_3\}$
$T(S_2) = \{S_1\}$
$T(S_3) = \{S_3, S_2\}$

Satisfiable - atleast 1 state produce T
Valid - All states produce T.

To prove $\Box(p \vee q)$



$T(S_0) = \{S_1, S_2\} = T$

Here q is true in both S_1 & S_2.

$T(S_1) = \{S_1, S_2, S_3\} = F$

Here $S_3 = F$

$T(S_2) = \{S_1\} = T$

$T(S_3) = \{S_2, S_3\} = F$

$\therefore$ $\Box(p \vee q)$ is satisfiable.
(Here each transition must be true)

But $\Diamond(p \vee q)$

$T(S_0) = \{S_1, S_2\} = T$

$T(S_1) = \{S_1, S_2, S_3\} = T$
(Here for some transition (S_1, S_2) it is true)

$T(S_2) = \{S_1\} = T$

$T(S_3) = \{S_2, S_3\} = T$

$\therefore$ $\Diamond(p \vee q)$ is valid.

20/11/17

4 basic operators:

$\Box$: always (universal) $\quad A \rightarrow (B \rightarrow A)$

$\Diamond$: eventually (existential) $\quad \Box A \rightarrow (\Box B \rightarrow \Box A)$

$\bigcirc$: next time or next step $\quad \Diamond A \rightarrow (\Diamond B \rightarrow \Diamond A)$

U : until

Classical Logic:

* $S \models f$: sequence S satisfies formula f.

$S \models true$: for any S

$S \models (f \text{ and } g)$: $S \models f$ and $S \models g$

Temporal Logic:

$p \wedge \neg q$
$\neg p \vee q$

- $S \models \Box f$: if for any j, $S[j \ldots] \models f$ valid
- $S \models \Diamond f$: if for some j, $S[j \ldots] \models f$
- $S \models \bigcirc f$: if $S[1 \ldots] \models f$

Q. $\models \Box p \leftrightarrow \neg \Diamond \neg p$

Assume $\models \Box p$

$S \models p$

Let $S' \models \neg p$

S' can never be a part of S since it gives a contradiction. It's possible when $S' \models \neg\neg p$

$\therefore \models \Box p \leftrightarrow \neg \Diamond \neg p$

Q. $\models \Box(p \to q) \to (\Box p \to \Box q)$

$\neg(\Box(p \to q) \wedge \neg(\Box p \to \Box q))$

$\neg(\Box(p \to q) \wedge \Box p \wedge \neg \Box q)$

$\left.\begin{array}{l} S \models \Box(p \to q) \\ S \models \Box p \end{array}\right\} S \models q$

$S \models \neg \Box q$

$S' \models \Diamond \neg q$

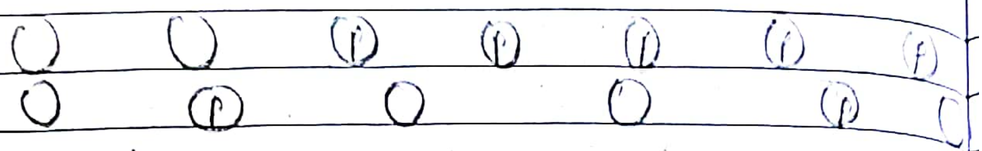
$\therefore$ We get a contradiction.

Properties:

1. $\models \Box p \to \bigcirc p$
2. $\models \Box p \to \Diamond p$
3. $\models \bigcirc p \leftrightarrow \Diamond p$
4. $\models \bigcirc p \leftrightarrow \neg \Diamond \neg p$
5. $\models \Box p \leftrightarrow p \wedge \bigcirc \Box p$
6. $\models \Diamond p \leftrightarrow p \vee \bigcirc \Diamond p$

Note ⊛ $\Diamond \Box p$

$\Box \Diamond p$



1. Reflexive

$$\Box A \rightarrow A \quad , \quad A \rightarrow \Diamond A$$

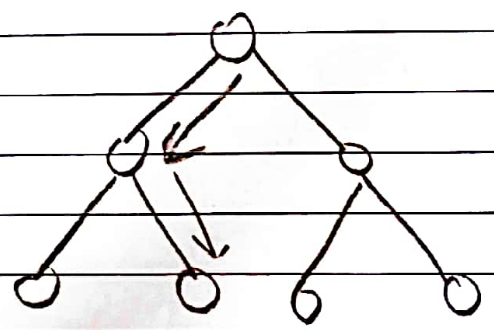
2. Transitive

$$S_1, S_2, S_3 : \text{states}$$
$$S_2 \in P(S_1) \wedge S_3 \in P(S_2)$$
$$\rightarrow S_3 \in P(S_1)$$

3. Linearity

<u>Linear Time logic</u>

From one state, we will have one & only one transition.



An interpretation of the form LTL is a part of state denoted by [illegible]

$$\sigma = S_0, S_1, S_2 \ldots$$

$$S_i = \wp[T, F]$$

Distributive

1. $\models \Box(p \wedge q) \leftrightarrow (\Box p \wedge \Box q)$
2. $\models \bigcirc(p \wedge q) \leftrightarrow \bigcirc p \wedge \bigcirc q$
3. $\models (\Box p \vee \Box q) \rightarrow \Box(p \vee q)$
4. $\models \bigcirc(p \vee q) \leftrightarrow \bigcirc p \vee \bigcirc q$
5. $\models \Diamond(p \vee q) \leftrightarrow \Diamond p \vee \Diamond q$
6. $\models \Diamond(p \wedge q) \rightarrow \Diamond p \wedge \Diamond q$
7. $\models \Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q)$
8. $\models (\Diamond p \rightarrow \Diamond q) \rightarrow \Diamond(p \rightarrow q)$
9. $\models \bigcirc(p \rightarrow q) \leftrightarrow (\bigcirc p \rightarrow \bigcirc q)$

Commutative

1. $\models \Box \bigcirc p \leftrightarrow \bigcirc \Box p$
2. $\models \Diamond \bigcirc p \leftrightarrow \bigcirc \Diamond p$
3. $\models \Diamond \Box p \rightarrow \Box \Diamond p$

Collapsing

1. $\models \Box\Box p \leftrightarrow \Box p$
2. $\models \Diamond\Diamond p \leftrightarrow \Diamond p$
3. $\models \Box\Diamond\Box p \leftrightarrow \Diamond\Box p$
4. $\models \Diamond\Box\Diamond p \leftrightarrow \Box\Diamond p$

21/11/17

Semantic Tableaux

α	α_1	α_2	β	β_1	β_2
$\Box A$	A	$\bigcirc\Box A$	$\Diamond A$	A	$\bigcirc\Diamond A$
$\neg\Box A$	$\neg A$	$\neg\bigcirc\Box A$	$\neg\Box A$	$\neg A$	$\neg\bigcirc\Diamond A$

X	X_1
OA	A
$\neg OA$	$\neg A$

* Semantic tableaux helps to check satisfiability and validity of a program. (T, F, infinite loop does not exist)
* Resolution : Helps to prove a formula from the given axioms.
* O – used when we don't know what will be executed next. There is a change from the normal flow to another flow.

Q $A = (p \vee q) \wedge O(\neg p \wedge \neg q)$

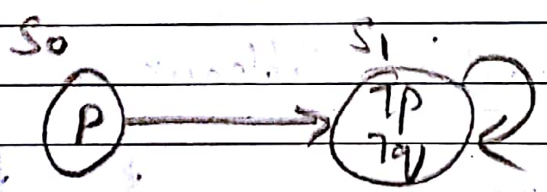
$$\downarrow$$

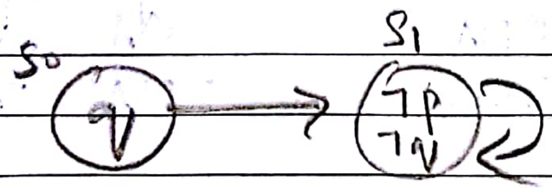
$$(p \vee q),\ O(\neg p \wedge \neg q)$$

$$p,\ O(\neg p \wedge \neg q) \qquad q,\ O(\neg p \wedge \neg q)$$

$$\neg p \wedge \neg q \qquad \neg p \wedge \neg q$$

$$\neg p,\ \neg q \qquad \neg p,\ \neg q$$





q. $A = \neg(\Box(p \wedge q) \rightarrow \Box p)$

$$\neg(\Box(p \wedge q) \rightarrow \Box p)$$
$$\downarrow$$
$$\Box(p \wedge q) \wedge \neg\Box p$$
$$\downarrow$$
$$\Box(p \wedge q), \neg\Box p$$
$$\downarrow$$
$$\Box(p \wedge q), \Diamond\neg p$$
$$\downarrow$$
$$(p \wedge q) \wedge O\Box(p \wedge q), \Diamond\neg p$$
$$\downarrow$$
$$(p \wedge q), O\Box(p \wedge q), \Diamond\neg p$$
$$\downarrow$$
$$(p \wedge q), O\Box(p \wedge q), \neg p \vee O\Diamond\neg p$$

$$\neg p, p, q, O\Box(p \wedge q) \qquad\qquad p, q, O\Box(p \wedge q), O\Diamond\neg p$$

$$\times \qquad\qquad \uparrow \times$$

$$p \wedge q, \Box(p \wedge q), \Diamond\neg p$$

It goes into a infinite loop. Hence, we cannot say if it's valid or not.

Program Verification

24/11/17

- Checks validity of a program.
- 4 components
  1) Assignment
  2) Compound
  3) Condition
     If else
  4) Looping
     for
     while
- Each line in the program is a state.

(S₀) Input (x, y)
↓
Program (50 lines)
↓
Output (x', y')

- $\vDash \{p\}\, S\, \{q\}$ (used to represent each line)
  S → segment
  p → precondition
  q → post condition

If $x = 3y + 1$,
Then we have
$$\vDash (x, y) \mid x = 3y+1 \mid (x', y')$$
Here, we don't know how many times this will get executed.
To ~~execute~~ get it executed 10 times we write
$$\vDash (x, y) \mid x = 3y + 1 \mid \wedge\ y \leq 10.$$
x can't be chosen here.

* A computer program consist of a sequence of symbols constructed according to formal syntactical rules.

The symbols has meaning which is assigned by an interpretation of the elements of the language. The symbols are called statements or commands.

* Correctness formula

A statement in the programming language can be considered to be a fn. that transforms a state of computation. A correctness formula is a triplet

$$\models \{p\} S \{q\}$$

where S is a program, p, q are formulas called precondition & postcondition. S is partially correct with respect to p & q iff S is started in a state where p is true, and if the computation of S terminates, then the terminating state is the state where q is true.

$$\models \{false\} S \{q\}$$

If after the transition, q is true, it is a valid statement.

$$\models \{p\} S \{false\}$$ invalid

* Post condition must be true.

Pre condition may be true or false.

$\Diamond(p \vee q) \leftrightarrow (\Diamond p \vee \Diamond q)$

$\exists \neg(\Diamond(p \vee q) \to (\Diamond p \vee \Diamond q))$

$\cancel{\neg\Diamond(p \vee q) \vee (\Diamond p \vee \Diamond q)}$

$\Box\neg(p \vee q) \vee (\Diamond p \vee \Diamond q)$

$\exists(\Diamond(p \vee q) \wedge \neg(\Diamond p \vee \Diamond q))$

$\downarrow$

$\Diamond(p \vee q), \neg(\Diamond p \vee \Diamond q)$

$\downarrow$

$\Diamond p \vee \Diamond q, \neg\Diamond p \wedge \neg\Diamond q$

$\downarrow$

$\Diamond p \vee \Diamond q, \neg\Diamond p, \neg\Diamond q$

$\swarrow \qquad \searrow$

$\Diamond p, \neg\Diamond p, \neg\Diamond q \qquad \Diamond q, \neg\Diamond p, \neg\Diamond q$

$\downarrow \qquad \downarrow$

$\times \qquad \times$

1) $\Diamond p, \neg\Diamond p, \neg\Diamond q$, 2) $\Diamond q, \neg\Diamond p, \neg\Diamond q$

3) $\Diamond p \vee \Diamond q$ from 1

4) $\neg\Diamond p, \neg\Diamond q$ from 2

5) $\neg\Diamond p \wedge \neg\Diamond q$ from 4.

6) $\Diamond(p \vee q)$ from 3

7) $\cancel{\Diamond(p \vee q)} \wedge \neg(\Diamond p \vee \Diamond q)$ from 5.

8). $\Diamond(p \vee q) \wedge \neg(\Diamond p \vee \Diamond q)$ from 6 & 7.

9). $\neg\neg(\Diamond(p \vee q) \rightarrow (\Diamond p \vee \Diamond q))$