MATHEMATICS FOR COMPUTING

WEEK 5:2

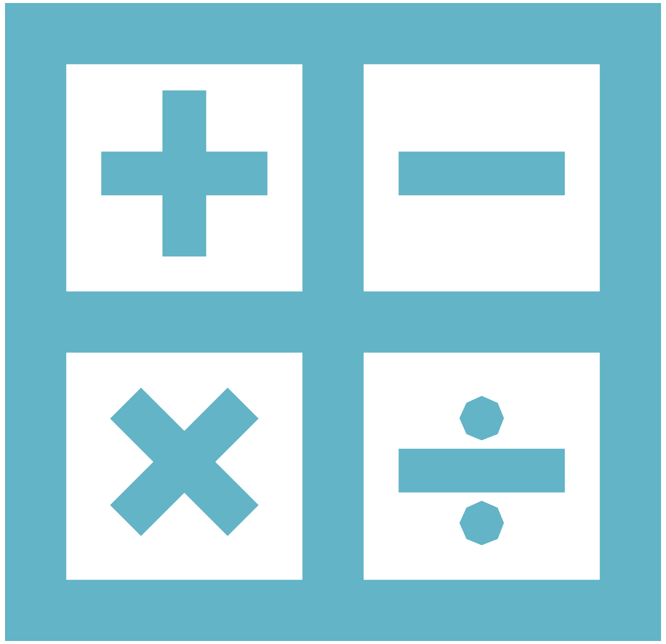


TABLEAU TECHNIQUE

SIGNED FORMULAE

- Given a CPL formula A, let us abbreviate by
  - T[A] - the situation when A is true
  - F[A] - the situation when A is false.

α FORMULAE

For α formulae the conditions for being true or false are unique!

α	α_1	α_2
$T[A \wedge B]$	$T[A]$,	$T[B]$
$F[A \vee B]$	$F[A]$,	$F[B]$
$F[A \Rightarrow B]$	$T[A]$,	$F[B]$
$T[\neg A]$	$F[A]$	
$F[\neg A]$	$T[A]$	

β FORMULAE

For β formulae the conditions for being true or false are not unique!
Here we have options, or so called “branching conditions”:

β	β_1	('or') β_2
$T[A \vee B]$	$T[A]$	$T[B]$
$F[A \wedge B]$	$F[A]$	$F[B]$
$T[A \Rightarrow B]$	$F[A]$	$T[B]$

CONSTRUCTION OF A TABLEAU

- **Definition** (Configuration in a tableau) Sets of signed formulae are called *configurations*.
- Below we define tableau construction rules, so that a rule is applied to a signed formula in the configuration above the horizontal line and the rule's conclusion is a configuration(s) below the horizontal line.
- Remember: in α rules, we have unique conditions for true and false, so α rules simply transform some given configuration to a new one
- Remember: in β rules, we have branching conditions for true and false, so these rules transform some given configuration new configurations reflecting branches

TABLEAU ALGORITHM

- Step 1: The initial node is labelled by F[A] itself (e.g.: assume that A is false).
- Step 2: The α and β expansion rules are applied to the formulae within labels of nodes of the graph.
- Step 3.1: If an expansion rule applies to α-formula in a label of a node n_i then create a new node, n_{i+1}, the successor of n_i, and put both conclusions of the rule into the label of n_{i+1}.

TABLEAU ALGORITHM

- Step 3.2: If an expansion rule applies to β-formula in a label of a node n_i then create two nodes $n_{i.1}$ and $n_{i.2}$, the children of n_i, and put the conclusions, β1 and β2 (of the rule being applied) into $n_{i.1}$ and $n_{i.2}$, respectively.
- Step 4: Apply 3.1 and 3.2 until no expansion rule to a configuration – label of a node - is applicable; such a configuration is called completed.
- Step 5: a derived configuration - label of a node - contains both T[B] and F[B], for some CPL formula B. Such a configuration is called closed.

OBTAIN REDUCED GRAPH

- Step 6: Obtain the reduced graph by applying the following deletion rules.
  - Delnode.1 Delete every node if is labelled by a closed configuration, e.g. the configuration contains both T[B] and F[B] for some formula B.
  - Delnode.2 If all the successors of a node have been deleted then delete this node.
  - Reduced graph G’ is empty if the initial node of the original graph is deleted.
- Step 7: A tableau is called closed if its reduced graph is empty.

OBTAINING VALIDITY

- Statement 1. For any CPL formula G, a tableau is closed, if and only if, G is unsatisfiable.
- Statement 2 [correctness of tableau for CPL] A tableau constructed for the assumption F[A] is closed if, and only if A is valid.

TEST YOUR KNOWLEDGE

- Consider $((p \Rightarrow q) \Rightarrow p) \Rightarrow p$. Assuming $F[((p \Rightarrow q) \Rightarrow p) \Rightarrow p]$, construct the tableau
- Consider Axiom 2: $(p \Rightarrow (q \Rightarrow r)) \Rightarrow ((p \Rightarrow q) \Rightarrow (p \Rightarrow r))$ Assuming $F[(p \Rightarrow (q \Rightarrow r)) \Rightarrow ((p \Rightarrow q) \Rightarrow (p \Rightarrow r))]$, construct the tableau.

INDUCTION

MATHEMATICAL INDUCTION

- Mathematical Induction is a special way of proving things. It has only 2 steps:
  - Step 1: Show it is true for the **first one**
  - Step 2: Show that if **any one** is true then the **next one** is true
  - Then **all** are true

WHAT IS MATHEMATICAL INDUCTION FOR?

- When we want to prove that all objects of some domain D have a property P, our reasoning depends on the type of D.
  - If D is a finite set of objects then we can simply investigate each of them.
  - If D is an infinite or a very large set then simple observation at its best will only give us some approximate knowledge.

EXAMPLE PROOF OF INDUCTION

Prove that for any integer n, the following is true

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$

We will prove this statement by mathematical induction

Base case: n = 1. We need to prove that (substituting n with 1)

$$\sum_{i=1}^{1} 1 = \frac{1(1+1)}{2} = 1$$

EXAMPLE PROOF OF INDUCTION

- **Inductive Hypothesis:** Suppose that the formula is valid for some integer n.
- **Inductive Step:** Now, based on this hypothesis, we need to show the summation formula is valid for n + 1
- Add n + 1 to both sides of the equation to demonstrate that the formula is still valid for n + 1.

INDUCTIVE STEP

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$

$$+(n+1)$$

$$+(n+1)$$

$$\sum_{i=1}^{n} i + (n+1) = \sum_{i=1}^{n+1} i$$

$$\frac{n(n+1)}{2} + (n+1) = \frac{n(n+1)+2(n+1)}{2} = \frac{(n+1)(n+2)}{2}$$

$$\sum_{i=1}^{i+1} i = \frac{(n+1)(n+2)}{2}$$

EXAMPLE PROOF OF INDUCTION

P(n) statement **1 + 2 + 3 + 4 + 5 + ... n = n(n+1)/2**				
	n	**recursive form (LHS)**	**function form (RHS)**	
Basic step	1	**1**	**1(1+1)/2 =1**	
Check step	2	**1 + 2 = 3**	**2(2+1)/2 =6/2=3**	
Check step	3	**etc.**	**etc.**	
Induction step *(Assume **P(k)** to be true for some n=k)*	$k \geq 1$	**1 + 2 + 3 + ... + k**	**k(k+1)/2**	
P(k+1) step *- State - Make appear P(k) form - Use P(k) assumption - Work out a simplification of LHS that matches RHS or vice-versa*	k+1	**1 + 2 + ... + k + (k+1)**	**(k+1)((k+1)+1)/2 = (k+1)(k+2)/2 = k(k+1)/2 + 2 (k+1)/2 = k(k+1)/2 + (k+1)**	
Conclude: Thus, the statement **P(k+1)** is true if the statement **P(k)** is true. Since **P(1)** is true, then by induction, the statement **P(n)** is true for $n \geq 1$				

RECURSION, RECURSIVE FUNCTIONS, RECURSIVE CONSTRUCTIONS

- Based on mathematical induction, we can now introduce recursion.
- Sometimes it is possible to define an object (function, sequence, algorithm, structure) "in terms of itself".

This process is called recursion.

Example: a recursive function on positive integers:

- ❑ $f(0) = 3$
- ❑ $f(n+1) = 2f(n) + 3$

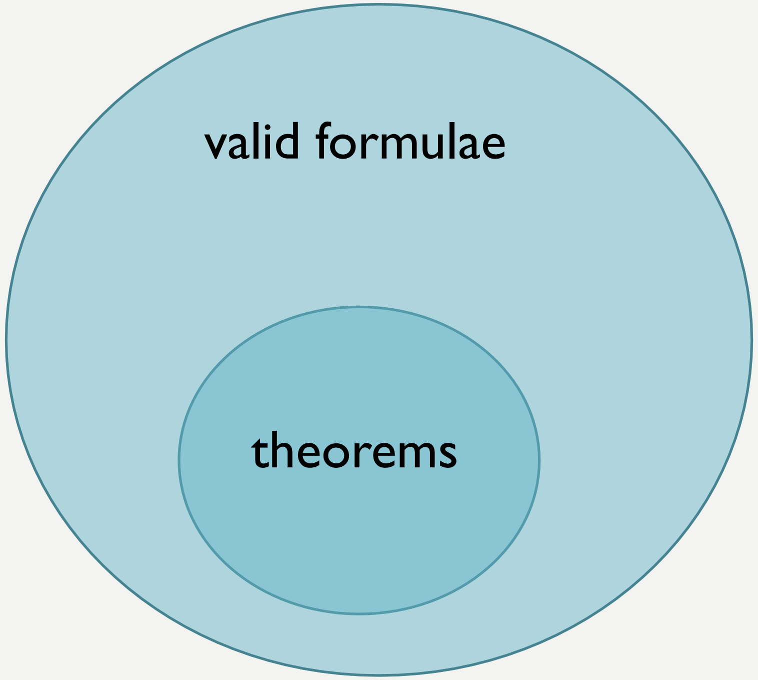
CHECK YOUR KNOWLEDGE

Prove by induction that the following statement is true for whichever value of natural number $\boldsymbol{n}$:

P($\boldsymbol{n}$) statement: $\sum_{i=0}^{n} 2^{(i-1)} = 2^n - 1$

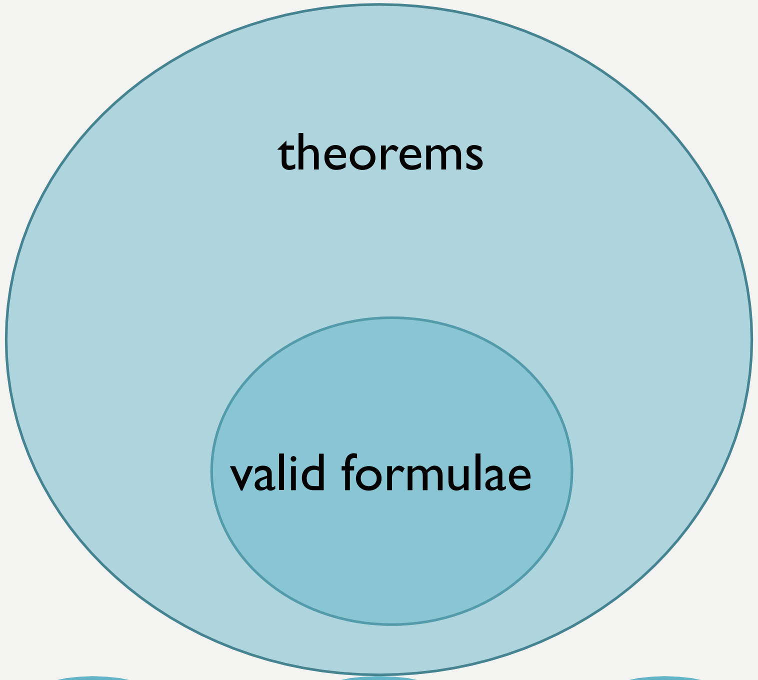
METAPROPERTIES OF LOGIC - SOUNDNESS

Definition [Soundness] Logical System L is sound if for every formula A of L, it is correct that if A is a theorem then A is a valid formula.



METAPROPERTIES OF LOGIC - COMPLETENESS

Definition [Completeness] Logical System L is complete if for every formula A of L, it is correct that if A is valid then A is a theorem.



QUESTIONS?

SAPNA.K@IIT.AC.LK