Mathematics in Computing
4COSC007C
Proofs

Logic Engineering in Classical Propositional Logic (CPL)

To build a logic we must define:

- its syntax,
- its semantics,
- provide its proof theory,
- establish correctness of this construction

Logic Engineering in Classical Propositional Logic (CPL)

- syntax
  - This is where we define which expressions are acceptable and which are not
- semantics
  - This is where we define the meaning of acceptable expressions – logical formulae. In our case – meaning of logical formulae are their truth values – true or false.
- proof theory
  - This is where we define techniques to formally justify the expressions that are always true
- correctness
  - This where we justify that what we can only prove formulae that are always true

Syntax

Alphabet of logic

- a set, Prop, of atomic propositions:p, q, r . . . p1, q1, r1 . . .
- a set, Log, of logical operators: $\neg, \wedge, \implies, \vee$
- technical symbols( ).

Language of Propositional logic

- **Definition 1 [Well Formed Formulae (wff)]**
  - any atomic proposition is wff;
  - if A and B are wff then $\neg A$, $A \wedge B$, $A \vee B$, $A \implies B$ are wff;
  - nothing else is a wff.
- Note:
  Definition 1 is a recursive (or inductive) definition and iseffective for any object we have an effective procedure to determine if it is a wff.
- In Definition 1 letters A, B are expressions of the metalanguage serving to abbreviate expressions of the object language (CPL).

Satisfiability and Validity

- **Definition 2**
  - Formula A is satisfiable if there is an interpretation of its atomic propositions which makes A true.
  - Alternatively, if no interpretation of atomic propositions makes A true then A is unsatisfiable

- **Definition 3**
  - Formula A is valid if every interpretation of its atomic propositions makes it true.
  - $\models$ notation is used to denote the fact that A is valid.

Logical Consequence

Definition 4

- Formula B is a logical consequence of a knowledge base $A_1, A_2, A_3, \ldots A_n$ if the following formula is valid:
  - $(A1 \wedge (A2 \wedge A3 \wedge An))) \implies B$
  - $A_1, A_2, A_3, \ldots A_n \models B$ is used to denote logical consequence of B.

Proof Strategies-Axiomatic Approach

- An axiom is a proposition formally accepted without demonstration, proof, or evidence as one of the starting points for the systematic derivation of an organized body of knowledge.
- From all valid formulae we choose a set of formulae and assume they do not need proofs.
- Next, we formulate the proof technique such that all other valid formulae can be proven from axioms using corresponding rules.

Axioms in CPL

Definition 5- [Logic CPLAx axiomatic formulation of CPL]

- $(p \implies (q \implies p))$
- $(p \implies (q \implies r)) \implies ((p \implies q) \implies (q \implies p))$
- $(\neg p \implies \neg q) \implies (q \implies p)$

Rules of Inference

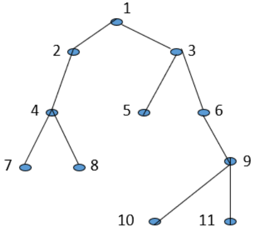
- **Substitution** : Let A be a formula of CPL and p a propositional variable in A . Let A(p/B) be a result of substituting all occurrences of p in A by a formula B.
- Then the following rule can be carried out: If A then A(p/B)
- $(p \implies (q \implies r)) \implies ((p \implies q) \implies (p \implies r))$ Ax. 2
- $(s \implies (q \implies r)) \implies ((s \implies q) \implies (s \implies r))$ from 1, substitution p/s

Rules of Inference

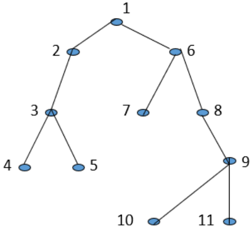
- **Modus Ponens (Implication Elimination)**
  From A and A $\implies$ B follows B
- It can be summarized as "P implies Q and P is asserted to be true, therefore Q must be true."

Searching for Proofs

Breadth-First Search
Here we explore the tree in order of levels
Level 1 – nodes 2 and 3
Level 2 – nodes 4, 5, 6
Level 3 – nodes 7, 8 and 9
Level 4 – nodes 10 and 11



Depth-First Search
Here we explore the tree in order of paths
Path 1 – nodes 1, 2, 3, 4
Path 2 – nodes 1, 2, 3, 5
Path 3 – nodes 1, 6, 7
Path 4 – nodes 1, 6, 8, 9, 10
Path 5 – nodes 1, 6, 8, 9, 11



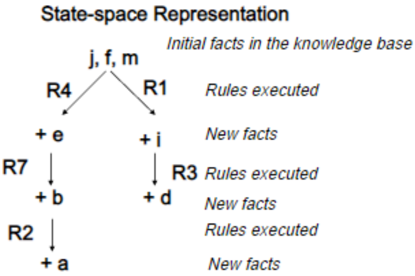
Forward Chaining

- Match the FACTS contained in the knowledge base to left hand sides of rules
- Apply the rules and instantiate new (temporary) facts
- Repeat until: the goal is achieved or until no more rules fire
- Operates as a breadth first search

Forward Chaining

KNOWLEDGE BASE

Rules	Initial facts
R1: m⇒i R2: b⇒a R3: i⇒d R4: j⇒e R5: k⇒e R6: g&h⇒c R7: e&f⇒b R8: c&d⇒a R9: l⇒i	j, f, m



Backward Chaining

- Match the GOAL to right hand side of rules
- Set up left side of rule as sub goals
- Repeat until: all sub goals match directly with known facts (goal is true) or some sub goals fail to match (goal is false)
- Backward chaining is usually implemented as depth first search

Backward Chaining

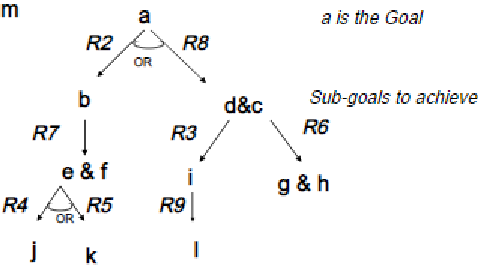
Rules

R1: m⇒i
R2: b⇒a
R3: i⇒d
R4: j⇒e
R5: k⇒e
R6: g&h⇒c
R7: e&f⇒b
R8: c&d⇒a
R9: l⇒i

Initial facts

j, f, m

Problem-space Representation



Proving validity by Contradiction

Consider Axiom 1. $p \implies (q \implies p)$
Goal: prove that Axiom 1 is valid but not by constructing the full truth table

1. Assume the contrary $p \implies (q \implies p)$ is not valid.
2. It should be the case then that p is true but $q \implies p$ is false
3. Since $q \implies p$ is false q should be true but p false
4. Contradiction:
5. Therefore our assumption that $p \implies (q \implies p)$ s not valid is wrong!
6. Therefore, $p \implies (q \implies p)$ is valid

Mathematics in Computing
4COSC007C
Tableaux Technique

*

Signed Formulae

- Given a CPL formula A, let us abbreviate by T[A] the situation when A is true and by F[A] the situation when A is false.
- We will call the expressions T[A] and F[A] signed formulae.

α Formulae

- For α formulae the conditions for being true or false are unique!
- T[A $\wedge$ B] this is only when T[A] AND T[B]
- F[A V B] this is only when F[A] AND F[B]
- F[A $\implies$ B] this is only when T[A] AND F[B]
- T[$\neg$ A] this is only when F[A]
- F[$\neg$ A] this is only when T[A]

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":
- T[A V B] this is when T[A] OR T[B]
- F[A $\wedge$ B] this is when F[A] OR F[B]
- T[A $\Longrightarrow$ B] this is when F[A] OR T[B]

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 Tableau

- based on the $\alpha - \beta$ rules.
- 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 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

Construction of Tableau

- We are ready to define the construction of the tableau for a formula A. The tableau is built as a labelled finite graph – in other words, each node in a graph is labelled by a configuration.
    1). The initial node is labelled by F[A] itself (eg assume that A is false).
    2). The α and β expansion rules are applied to the formulae within labels of nodes of the graph.
    2.1). 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}.
    2.2). 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, $\beta 1$ and $\beta 2$ (of the rule being applied) into $n_{i.1}$ and $n_{i.2}$, respectively.

3). . Apply 2.1 and 2.2 until:
    3.1). no expansion rule to a configuration – label of a node - is applicable; such a configuration is called completed.
    3.2). 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.

Tableau as a graph

- Now a tableau is a graph G = (N,E,L), which satisfies the following conditions
- N is a set of those nodes in the construction above,
- E is a set of edges such that forn_i, $n_j \in N$, we have $E(n_i, n_i)$ if, and only if, n_j is an immediate successor of n_i, and
- L is a set of labels, so for $n_i \in N$, the label of n_i is $L(n_i)$. The labels of leafs of G are completed configurations or closed configurations

Reduced graph for a tableau

- Given a tableau G, apply repeatedly 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.
- The resulting graph is called the reduced graph.

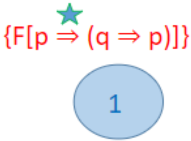
Closed Tableau

- Let G be the initial graph representing a tableau and G' be a reduced graph. G' is empty if the initial node of G has been deleted.
- A tableau is called closed if its reduced graph is empty.
- Statement 1. For any CPL formula A, a tableau is closed, if and only if, A 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.

Example 1

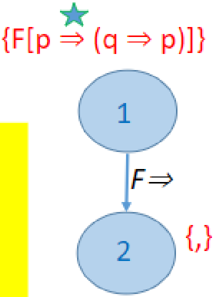
- Consider Axiom 1. $p \implies (q \implies p)$
- Assuming $F[p \implies (q \implies p)]$, we construct a tableau as follows: ...

Example 1



root node, node 1: negation of the formula.

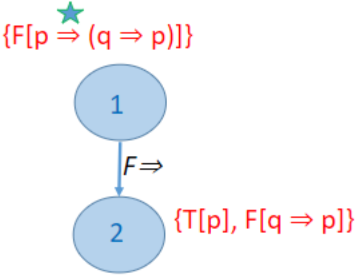
Example 1



By writing ★ above a signed formula in the configuration we mark the formula to which an expansion rule applies

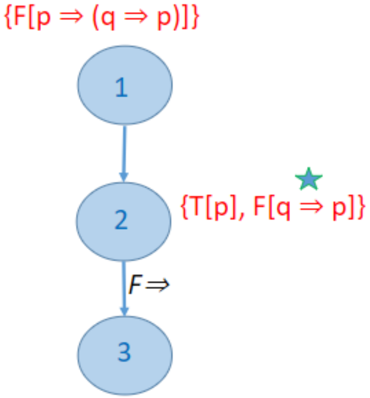
Transitions are labelled by the name of the expansion rule applied with ★

Example 1



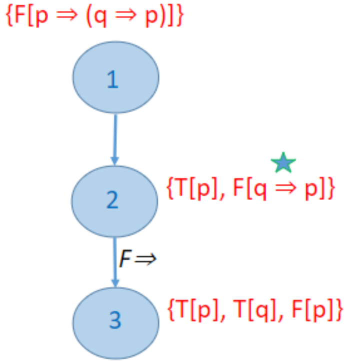
we apply the α rule $F\Rightarrow$ to F[p ⇒ (q ⇒ p)]
constructing one children of configuration 1

Example 1



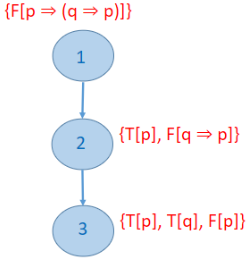
we mark the formula to decompose and identify the rule, here the α rule $F{\Rightarrow}$

Example 1



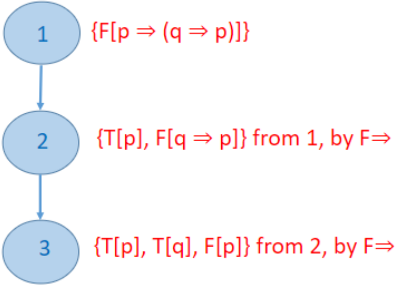
we apply the α rule $F\Rightarrow$ to F[q ⇒ p]
constructing one children of configuration 2

Example 1



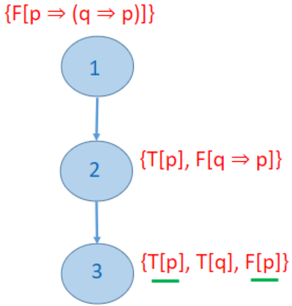
Configuration on step 3 is completed, hence, we stop the tableau construction as there are no more expansion rules applicable.

Example 1



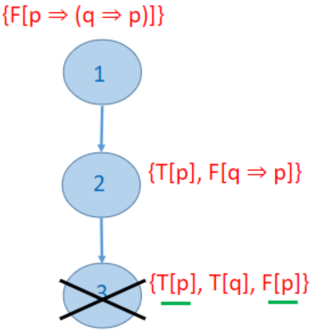
recap of building the tableau

Example 1



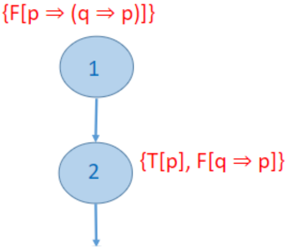
In the configuration on step 3, we have T[p] and F[p], hence, this configuration is closed

Example 1



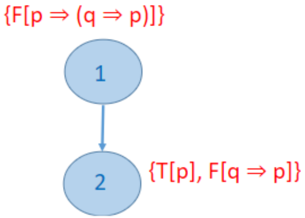
we have T[p] and F[p], hence configuration in step 3 is closed so we delete the node 3 labelled by this configuration applying rule Delnode.1.

Example 1



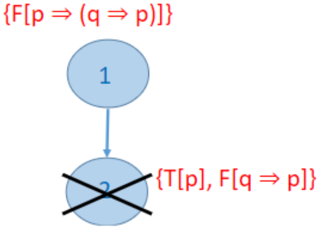
After we have deleted node 3 we have an edge from 2 leading to nowhere so we delete this edge

Example 1



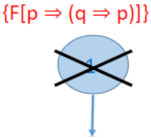
After we have deleted an edge from node 2 we have node 2 as the node with all its successors being deleted

Example 1



Now, 2 is a node all of whose successors have been deleted so we delete 2 applying rule Delnode.2.

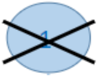
Example 1



An edge leads from node 1 to nowhere so we delete this edge and then node 1 becomes the one whose successors have been deleted so we delete 3 applying rule Delnode.2., obtaining the empty reduced graph

Example 1

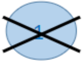
{F[p ⇒ (q ⇒ p)]}



Now, we have obtained the empty reduced graph

Example 1

$\{F[p \Rightarrow (q \Rightarrow p)]\}$



- We have obtained the reduced graph.
- This graph is empty, so the tableau for given formula $p \Rightarrow (q \Rightarrow p)$ is closed hence we conclude $p \Rightarrow (q \Rightarrow p)$ is valid

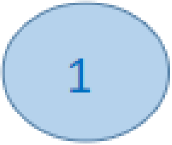
- NOTE THAT IN THE CONSTRUCTION ABOVE WE ONLY USED α rules
- ALSO NOTE THAT WE CONSTRCUTED THE TABLEAU IN BREADTH-FIRST SEARCH

Example 2

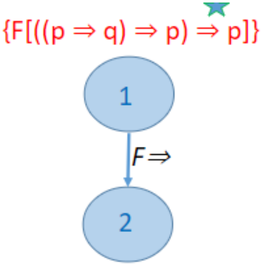
- Consider Pierce law.
  $((p \implies q) \implies p) \implies p$
- Assuming $F[((p \implies q) \implies p) \implies p]$, we construct a tableau as follows:

Example 2

$\{F[((p \Rightarrow q) \Rightarrow p) \Rightarrow p]\}$

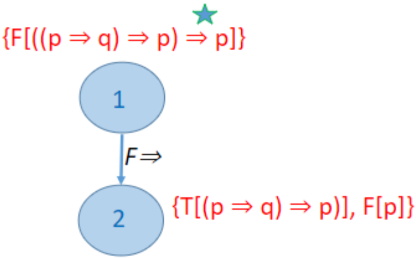


Example 2

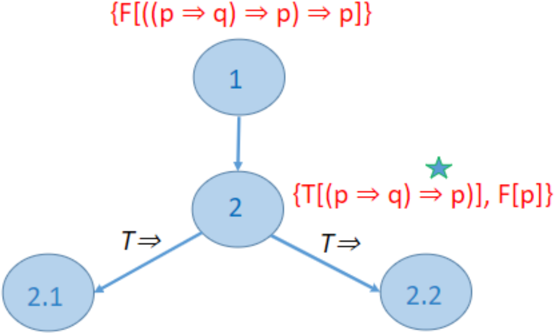


we apply the α rule $F\Rightarrow$ to F[((p ⇒q) ⇒p) ⇒p]
constructing one children of configuration 1

Example 2

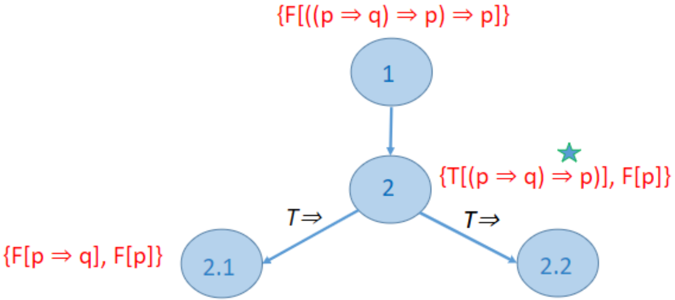


Example 2

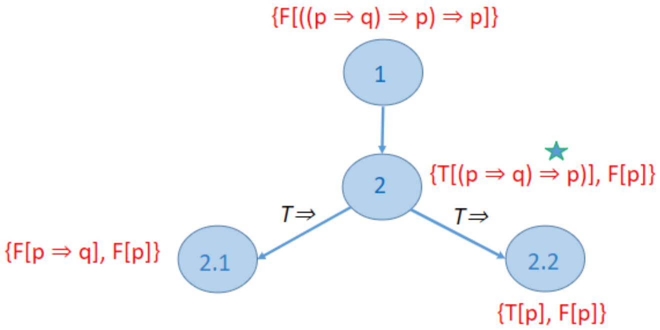


we apply the β rule T⇒ to T[(p ⇒ q) ⇒ p]
constructing two children of configuration 2:

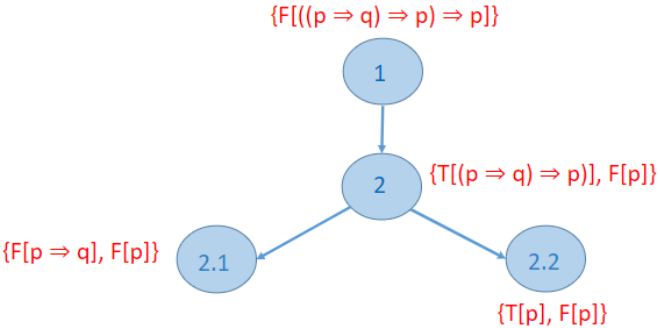
Example 2



Example 2

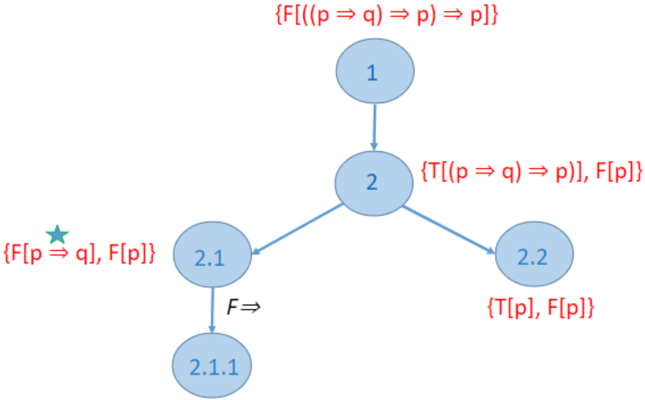


Example 2

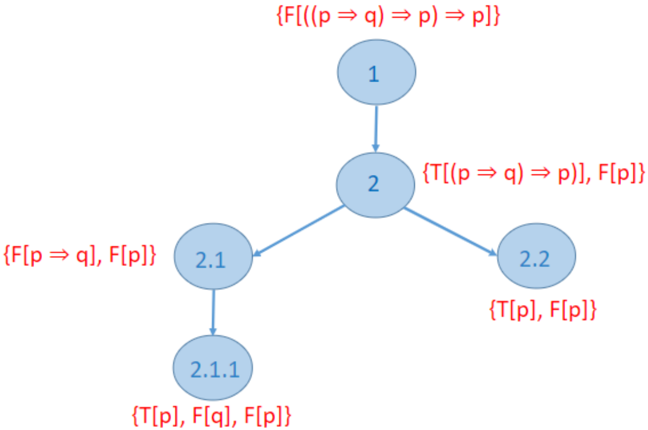


Configuration 2.2 is closed, we proceed with 2.1.

Example 2



Example 2



Example 2

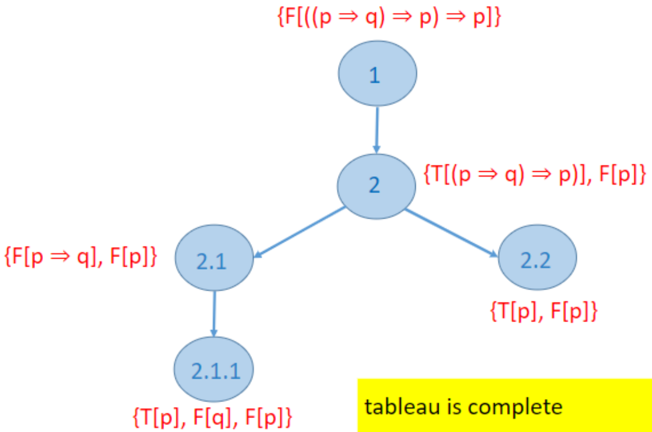
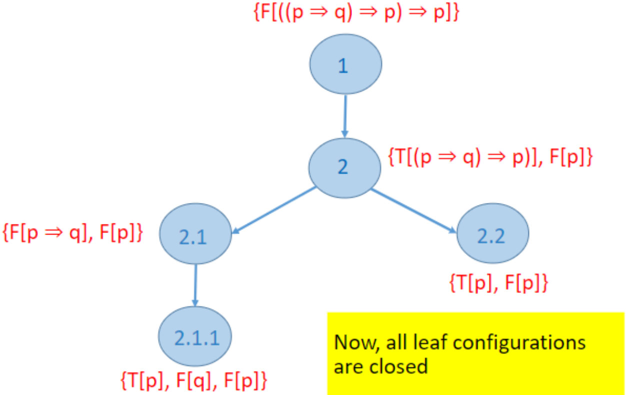


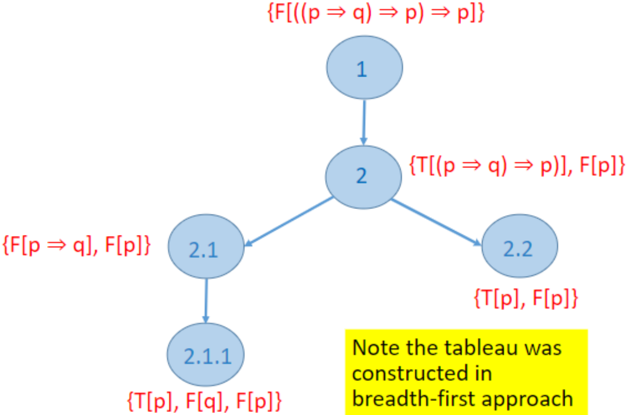
tableau is complete

Example 2



{F[((p ⇒ q) ⇒ p) ⇒ p]}
1
2
{T[(p ⇒ q) ⇒ p)], F[p]}
{F[p ⇒ q], F[p]}
2.1
2.2
{T[p], F[p]}
2.1.1
{T[p], F[q], F[p]}
Now, all leaf configurations are closed

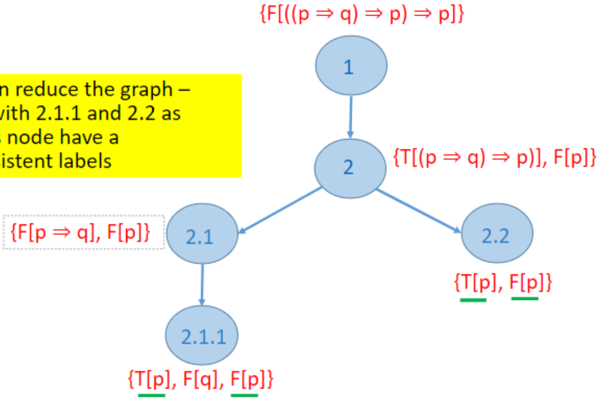
Example 2



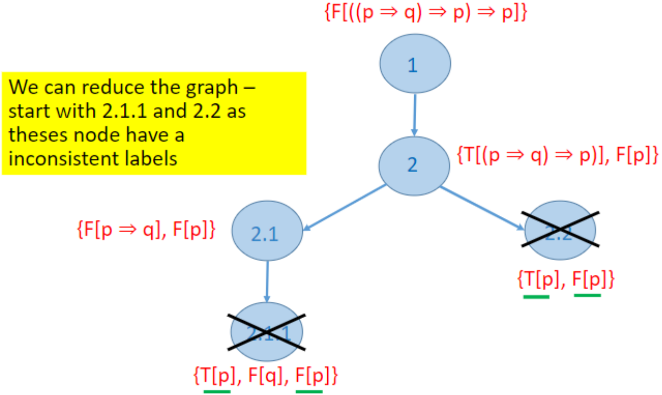
Note the tableau was constructed in breadth-first approach

Example 2

We can reduce the graph – start with 2.1.1 and 2.2 as theses node have a inconsistent labels



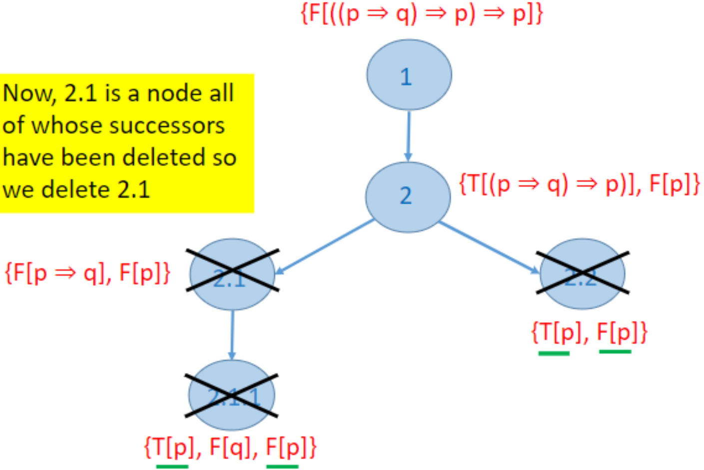
Example 2



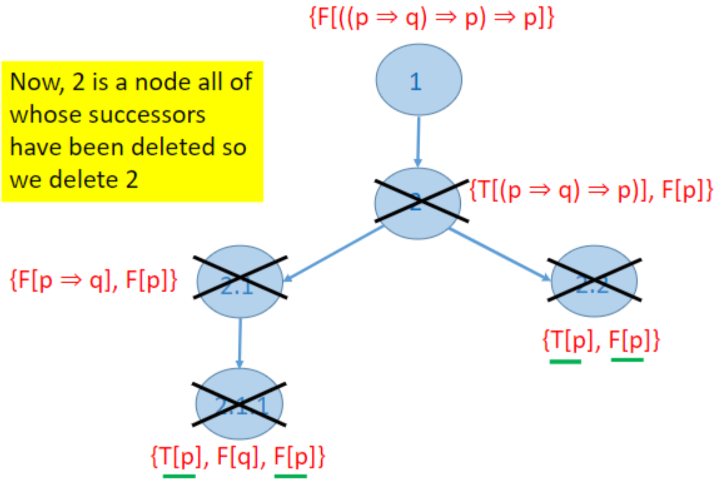
{F[((p ⇒ q) ⇒ p) ⇒ p]}
1
We can reduce the graph – start with 2.1.1 and 2.2 as theses node have a inconsistent labels
2
{T[(p ⇒ q) ⇒ p)], F[p]}
{F[p ⇒ q], F[p]}
2.1
{T[p], F[p]}
{T[p], F[q], F[p]}

Example 2

Now, 2.1 is a node all of whose successors have been deleted so we delete 2.1



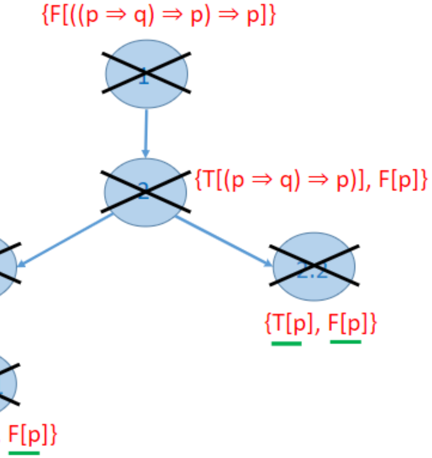
Example 2



{F[((p ⇒ q) ⇒ p) ⇒ p]}
1
Now, 2 is a node all of whose successors have been deleted so we delete 2
2
{T[(p ⇒ q) ⇒ p)], F[p]}
{F[p ⇒ q], F[p]}
2.1
2.2
{T[p], F[p]}
2.1.1
{T[p], F[q], F[p]}

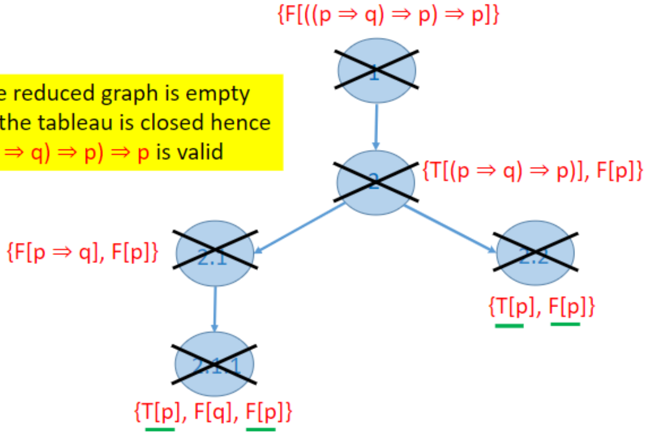
Example 2

Now, 1 is a node all of whose successors have been deleted so we delete 1



Example 2

The reduced graph is empty so the tableau is closed hence $((p \Rightarrow q) \Rightarrow p) \Rightarrow p$ is valid



Example 2

Below we represent the tableau as a tree:

$$
\begin{array}{c}
\{F[((p \Rightarrow q) \Rightarrow p) \Rightarrow p]\} \\
\downarrow \\
\{T[(p \Rightarrow q) \Rightarrow p)],\ F[p]\} \\
\swarrow \qquad\qquad \searrow \\
\begin{array}{cc}
\{F[p \Rightarrow q],\ F[p]\} & \{T[p],\ F[p]\} \\
\downarrow & \\
\{T[p],\ F[q],\ F[p]\} &
\end{array}
\end{array}
$$

THANK YOU

Any Questions?