Logical Entailment

A set of premises $\Delta$ *logically entails* a conclusion $\varphi$ if and only if every truth assignment that satisfies the premises also satisfies the conclusion.

Truth Table Method

Truth Table Method: Form a truth table for our language and check that every truth assignment that satisfies the premises also satisfies the conclusion.

Problem: For a language with $n$ constants, there are $2^n$ possible truth assignments.
Proofs

Proofs:
Symbolic Manipulation of sentences rather than enumeration of truth assignments

Benefits:
Proofs usually smaller than truth tables
Proofs can often be found with less work

Programme

Linear Proofs
Structured Proofs
Fitch System
Soundness and Completeness
Linear Proofs

Schemas

A *schema* is an expression satisfying the grammatical rules of our language except for the occurrence of metavariables (written here as Greek letters) in place of various subparts of the expression.

\[ \varphi \Rightarrow \psi \]
Rules of Inference

A rule of inference is a pattern of reasoning consisting of some schemas, called premises, and one or more additional schemas, called conclusions.

\[
\phi \Rightarrow \psi \\
\phi \\
\psi
\]

This rule is called Implication Elimination or Modus Ponens.

Rule Instances

An instance of a rule of inference is a rule in which all metavariables have been consistently replaced by legal sentences.

Rule:

\[
\phi \Rightarrow \psi \\
\phi \\
\psi
\]

Instances:

\[
\begin{align*}
p \Rightarrow q & \quad (p \Rightarrow q) \Rightarrow (q \Rightarrow r) \\
p & \quad (p \Rightarrow q) \\
q & \quad (q \Rightarrow r)
\end{align*}
\]
Rule Instances

An *instance* of a rule of inference is a rule in which all metavariables have been consistently replaced by legal sentences.

Rule:

\[ \phi \Rightarrow \psi \]

\[ \phi \]

\[ \psi \]

Instances:

\[ p \Rightarrow q \]

\[ p \]

\[ q \]

\[ (p \Rightarrow q) \Rightarrow (q \Rightarrow r) \]

\[ (q \Rightarrow r) \]

Rule Application

A rule *applies* to a set of sentences if and only if there is an instance of the rule in which all of the premises are in the set.

In this case, the conclusions of the instance are the *results* of the rule application.
Rule Application Example

Set of Sentences:
\{p, \quad p \Rightarrow q, \quad (p \Rightarrow q) \Rightarrow (q \Rightarrow r)\}

Rule:
\[
\frac{\phi \Rightarrow \psi}{\psi} \\
\frac{\phi}{\psi}
\]

Instance:
\[
\frac{p \Rightarrow q}{p} \\
\frac{p}{q}
\]

Result:
\[q\]
Rule Application Example

Set of Sentences:
\{p, \quad p \Rightarrow q, \quad (p \Rightarrow q) \Rightarrow (q \Rightarrow r)\}

Rule:
\[
\begin{array}{c}
\phi \Rightarrow \psi \\
\phi \\
\end{array}
\]

Instance:
\[
\begin{array}{c}
(p \Rightarrow q) \Rightarrow (q \Rightarrow r) \\
(p \Rightarrow q) \\
(q \Rightarrow r) \\
\end{array}
\]

Result:
\[(q \Rightarrow r)\]

Axiom Schemata

An *axiom schema* is a rule of inference without premises.

Axiom Schema written as rule:
\[
\phi \Rightarrow (\psi \Rightarrow \varphi)
\]

Axiom Schema written as schema:
\[
\phi \Rightarrow (\psi \Rightarrow \varphi)
\]
Axiom Schema Instances

Axiom Schema:

\[ \varphi \Rightarrow (\psi \Rightarrow \varphi) \]

Instances:

\[ p \Rightarrow (q \Rightarrow p) \]
\[ q \Rightarrow (p \Rightarrow q) \]
\[ (p \Rightarrow q) \Rightarrow ((q \Rightarrow r) \Rightarrow (p \Rightarrow q)) \]

Common Axiom Schemata

Implication Creation

\[ \varphi \Rightarrow (\psi \Rightarrow \varphi) \]

Implication Distribution

\[ (\varphi \Rightarrow (\psi \Rightarrow \chi)) \Rightarrow ((\varphi \Rightarrow \psi) \Rightarrow (\varphi \Rightarrow \chi)) \]

Contradiction Realization

\[ (\varphi \Rightarrow \psi) \Rightarrow ((\varphi \Rightarrow \neg \psi) \Rightarrow \neg \varphi) \]
Linear Proof

A linear proof of a conclusion from a set of premises is a sequence of sentences terminating in the conclusion in which each item is either (1) a premise or (2) an instance of an axiom schema or (3) the result of applying a rule of inference to earlier items in sequence.

Example

Premises: \{p, p \implies q, (p \implies q) \implies (q \implies r)\}

Conclusion: r

1. p \hspace{1cm} \text{Premise}
2. p \implies q \hspace{1cm} \text{Premise}
3. (p \implies q) \implies (q \implies r) \hspace{1cm} \text{Premise}
4. q \hspace{1cm} \text{IE: 1,2}
5. q \implies r \hspace{1cm} \text{IE: 2,3}
6. r \hspace{1cm} \text{IE: 4,5}
Example

Premises: \( \{p \Rightarrow q, q \Rightarrow r\} \)
Conclusion: \( p \Rightarrow r \)

1. \( p \Rightarrow q \)  \hspace{1cm} \text{Premise}
2. \( q \Rightarrow r \)  \hspace{1cm} \text{Premise}
3. \( (q \Rightarrow r) \Rightarrow (p \Rightarrow (q \Rightarrow r)) \)  \hspace{1cm} \text{IC}
4. \( p \Rightarrow (q \Rightarrow r) \)  \hspace{1cm} \text{IE: 3,2}
5. \( (p \Rightarrow (q \Rightarrow r)) \Rightarrow ((p \Rightarrow q) \Rightarrow (p \Rightarrow r)) \)  \hspace{1cm} \text{ID}
6. \( (p \Rightarrow q) \Rightarrow (p \Rightarrow r) \)  \hspace{1cm} \text{IE: 5,4}
7. \( p \Rightarrow r \)  \hspace{1cm} \text{IE: 6,1}

Provability

A proof system is a collection of rules of inference.

A conclusion is said to be provable from a set of premises using a proof system \( R \) if and only if there is a finite proof of the conclusion from the premises using only the rules in \( R \).

In what follows, we frequently use the notation \( \Delta \vdash_\mathcal{R} \varphi \) to express the fact that \( \varphi \) is provable from \( \Delta \) using \( R \).

If the set \( R \) of rules is clear from context, we drop the \( R \) and write \( \Delta \vdash \varphi \).
Mendelson System

Rule of Inference and Axiom Schemata
  Implication Elimination
  Implication Creation
  Implication Distribution
  Conflict Realization

Completeness
  Sufficient to prove logically entailed conclusions
  where premises and conclusion are written
  using only \( \neg \) and \( \Rightarrow \).

Structured Proofs
Structured Proofs

Problem: \{p \Rightarrow q, q \Rightarrow r\} \vdash (p \Rightarrow r)\

1. \(p \Rightarrow q\) Premise
2. \(q \Rightarrow r\) Premise
3. \(p\) Assumption
4. \(q\) IE: 1, 3
5. \(r\) IE: 2, 4
6. \(p \Rightarrow r\) II: 3, 5

Operations on Structured Proofs

Making Assumptions

Applying Ordinary Rules of Inference

Applying Structured Rules of Inference
Making Assumptions

In a structured proof, it is permissible to make an arbitrary assumption in any nested proof. The assumption need not be a member of the initial premise set.

Such assumptions cannot be used directly outside of the subproof, only as conditions in a derived implication, so they do not contaminate other subproofs or the outer proof.

Example

1. $p \Rightarrow q$  
   Premise
2. $q \Rightarrow r$  
   Premise
3. $p$  
   Assumption
4. $q$  
   IE: 1, 3
5. $r$  
   IE: 2, 4
6. $p \Rightarrow r$  
   II: 3, 5
Ordinary Rules of Inference

An ordinary rule applies to a particular subproof of a structured proof if and only if there is an instance of the rule in which all of the premises occur earlier in the subproof or some superproof of the subproof.

Importantly, it is not permissible to use sentence in subproofs of that subproof or in other subproofs of its superproofs.

Example

1. $p \Rightarrow q$  Premise
2. $q \Rightarrow r$  Premise
3. $p$  Assumption
4. $q$  IE: 1, 3
5. $r$  IE: 2, 4
6. $p \Rightarrow r$  II: 3, 5
Bad Example

1. \( p \implies q \)  Premise
2. \( q \implies r \)  Premise
3. \( p \)  Assumption
4. \( q \)  IE: 1, 3
5. \( r \)  IE: 2, 4

Structured Rules of Inference

A *structured rule of inference* is a pattern of reasoning consisting of one or more schemas, called *premises*, and one or more additional schemas, called *conclusions*, in which at least one of the premises is a condition of the form \( \phi \vdash \psi \).

\[
\frac{\phi \vdash \psi}{\phi \implies \psi}
\]

This rule is called *Implication Introduction*. 
Structured Rule Application

A structured rule applies to a particular subproof of a structured proof if and only if there is an instance of the rule in which all of the premises are satisfied.

A premise that is a simple schema is satisfied if and only if premise occurs earlier in the subproof or in some superproof of the subproof.

A premise of the form \( \varphi \rightarrow \psi \) is satisfied if and only if there is an earlier subproof within the current subproof where \( \varphi \) is the only premise and \( \psi \) is any conclusion.

Example

1. \( p \Rightarrow q \) Premise
2. \( q \Rightarrow r \) Premise
3. \( p \) Assumption
4. \( q \) IE: 1, 3
5. \( r \) IE: 2, 4
6. \( p \Rightarrow r \) II: 3, 5
Fitch System

Fitch System is a popular Natural Deduction proof system.

- Negation Introduction
- Negation Elimination
- Implication Introduction
- Implication Elimination
- And Introduction
- And Elimination
- Biconditional Introduction
- Biconditional Elimination
- Or Introduction
- Or Elimination
Negations

Negation Introduction (NI)

\[
\begin{align*}
\phi &\implies \psi \\
\phi &\implies \neg \psi \\
\neg \phi
\end{align*}
\]

Negation Elimination (NE)

\[
\begin{align*}
\neg \neg \phi \\
\phi
\end{align*}
\]

Conjunctions

And Introduction (AI)

\[
\begin{align*}
\phi_1 \\
\vdots \\
\phi_n \\
\phi_1 \land \ldots \land \phi_n
\end{align*}
\]

And Elimination (AE)

\[
\begin{align*}
\phi_1 \land \ldots \land \phi_n \\
\phi_1 \\
\vdots \\
\phi_n
\end{align*}
\]
Disjunctions

Or Introduction (OI)

\[ \frac{q_i}{q_i \lor \ldots \lor q_i} \]

Or Elimination (OE)

\[ q_i \lor \ldots \lor q_n \]
\[ q_i \Rightarrow \psi \]
\[ \ldots \]
\[ q_n \Rightarrow \psi \]
\[ \psi \]

Implications

Implication Introduction (II)

\[ \frac{q \vdash \psi}{q \Rightarrow \psi} \]

Implication Elimination (IE)

\[ q \Rightarrow \psi \]
\[ q \]
\[ \psi \]
Equivalences / Biconditionals

Biconditional Introduction (BI)
\[ \varphi \implies \psi \]
\[ \psi \implies \varphi \]
\[ \varphi \iff \psi \]

Biconditional Elimination (BE)
\[ \varphi \iff \psi \]
\[ \varphi \implies \psi \]
\[ \psi \implies \varphi \]

Logical Entailment and Provability

A set \( \Delta \) of premises logically entails a conclusion \( \varphi \) (written \( \Delta \models \varphi \)) if and only if every truth assignment that satisfies \( \Delta \) also satisfies \( \varphi \).

A conclusion \( \varphi \) is said to be provable from a set \( \Delta \) of premises (written \( \Delta \vdash \varphi \)) if and only if there is a finite proof of \( \varphi \) from \( \Delta \).
Soundness and Completeness

Soundness: A proof system is *sound* if and only if every provable conclusion is logically entailed.

\[(\Delta \vdash q) \implies (\Delta \models q)\]

Completeness: Our proof system is *complete* if and only if every logically entailed conclusion is provable.

\[(\Delta \models q) \implies (\Delta \vdash q)\]

Results

The Mendelson System is sound and complete for all premises and conclusions that can be written in terms of \(\neg\) and \(\Rightarrow\).

The Fitch System is sound and complete for the entire language of Propositional Logic.
Upshot

The truth table method and the proof method succeed in exactly the same cases.

On large problems, the proof method usually takes fewer steps than the truth table method. (In the worst case, the proof method can take as many steps to find an answer as the truth table method.)

Proofs are usually much smaller than the corresponding truth tables. So writing an argument to convince others does not take as much space.