Lecture 1 — Propositional Logic

Ondřej Lengál

Faculty of Information Technology
Brno University of Technology

IAM'24

Logic

What is (formal) **logic**?

- Logic can be considered as a branch of mathematics that studies universal principles of correct reasoning in various formal systems.

"[Logic is] . . . the name of a discipline that analyzes the meaning of the concepts common to all the sciences, and establishes the general laws governing the concepts."

—Alfred Tarski

"To discover truths is the task of all sciences; it falls to logic to discern the laws of truth. . . . I assign to logic the task of discovering the laws of truth, not of assertion or thought."

—Gottlob Frege

A Brief History of Logic

pre-Aristotle

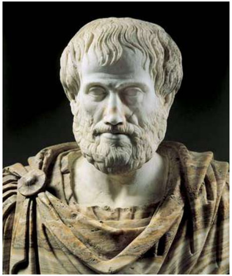
- Zeno of Elea (*490–430* BCE)
  - paradoxes: Achilles and the tortoise, arrow paradox, . . .
  - *reductio ad absurdum* = proof by contradiction
- Euclid (*325–270* BCE)
  - (**Elements**—geometry) **proofs** follow **axioms** in a **formal** way
  - (as opposed to empirical methods)
- Socrates, Plato, Parmenides, . . .
- Liar Paradox (Eubulides)

  This statement is false.

A Brief History of Logic

Aristotle (384–322 BCE)

- *The Father of Logic*
- Organon: works on logic
- **formal** term logic (predecessor of predicate logic)
- syllogisms to infer conclusions:



Example (Modus Barbara)

All **men** are **mortal**.
Socrates is a **man**.
$\therefore$ **Socrates** is **mortal**.

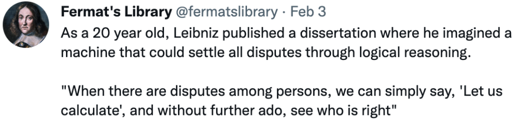
Example (Modus Calemes)

All men are mortal.
Elvis is immortal.
$\therefore$ Elvis is not a man.

A Brief History of Logic

Gottfried Wilhelm Leibniz (1646–1716)

- attempts to mechanise reasoning using universal calculus
  - *“Calculemus!” (Let us calculate!)*



Fermat's Library @fermatslibrary · Feb 3
As a 20 year old, Leibniz published a dissertation where he imagined a machine that could settle all disputes through logical reasoning.

"When there are disputes among persons, we can simply say, 'Let us calculate', and without further ado, see who is right"

A Brief History of Logic

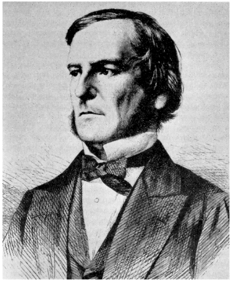
Gottfried Wilhelm Leibniz (1646–1716)

- attempts to mechanise reasoning using universal calculus
  - *“Calculemus!” (Let us calculate!)*

Fermat's Library @fermatslibrary · Feb 3

As a 20 year old, Leibniz published a dissertation where he imagined a machine that could settle all disputes through logical reasoning.

"When there are disputes among persons, we can simply say, 'Let us calculate', and without further ado, see who is right"



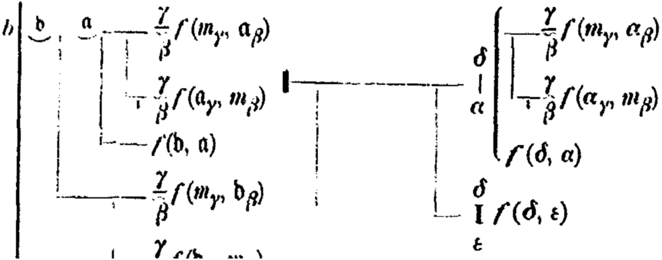
George Boole (1815–1864)

- indirect successor of Leibniz
- invents Boolean algebra (propositional logic)
- logical propositions as algebraic equations
  $\rightsquigarrow$ logic can be reduced to algebra

A Brief History of Logic

Gottlob Frege (1848–1925)

- *Begriffsschrift* (1879)
- rigorous formal language, quantifiers, variables
- shift from "*logic for calculation*" (Boole) to "*logic to model reality*"



- *The Foundations of Arithmetic (Die Grundlagen der Arithmetik)*
- found inconsistent (Russell's Paradox)
  - just as the 2^{nd} volume of Foundations was about to go to press

A Brief History of Logic

Foundational crisis in mathematics (early 20$^{\text{th}}$ century)

- many paradoxes occurring in foundations of mathematics:
  - $|\mathbb{N}| \neq |\mathbb{R}|$ (Cantor, 1873)
  - Russell's paradox (1901) — "Let $S = \{x \mid x \notin x\}$. Does $S \in S$?"
  - Berry Paradox — "The smallest positive integer not definable in under eleven words."
  - Richard's Paradox
  - Banach-Tarski paradox (1924)
  - ...

A Brief History of Logic

Foundational crisis in mathematics (early 20th century)

- many paradoxes occurring in foundations of mathematics:
  - $|\mathbb{N}| \neq |\mathbb{R}|$ (Cantor, 1873)
  - Russell's paradox (1901) — "Let $S = \{x \mid x \notin x\}$. Does $S \in S$?"
  - Berry Paradox — "The smallest positive integer not definable in under eleven words."
  - Richard's Paradox
  - Banach-Tarski paradox (1924)
  - ...
- Henri Poincaré (Rome, 1908):
  - *"Later generations will regard Mengenlehre [set theory] as a disease from which one has recovered."*

A Brief History of Logic

Foundational crisis in mathematics (early 20$^{\text{th}}$ century)

- many paradoxes occurring in foundations of mathematics:
  - $|\mathbb{N}| \neq |\mathbb{R}|$ (Cantor, 1873)
  - Russell's paradox (1901) — "Let $S = \{x \mid x \notin x\}$. Does $S \in S$?"
  - Berry Paradox — "The smallest positive integer not definable in under eleven words."
  - Richard's Paradox
  - Banach-Tarski paradox (1924)
  - ...
- Henri Poincaré (Rome, 1908):
  - *"Later generations will regard Mengenlehre [set theory] as a disease from which one has recovered."*
- David Hilbert (1926):
  - *"No one shall expel us from the Paradise that Cantor has created."*

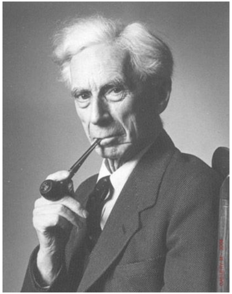
A Brief History of Logic

Foundational crisis in mathematics (early 20th century)

- many paradoxes occurring in foundations of mathematics:
  - $|\mathbb{N}| \neq |\mathbb{R}|$ (Cantor, 1873)
  - Russell's paradox (1901) — "Let $S = \{x \mid x \notin x\}$. Does $S \in S$?"
  - Berry Paradox — "The smallest positive integer not definable in under eleven words."
  - Richard's Paradox
  - Banach-Tarski paradox (1924)
  - ...
- Henri Poincaré (Rome, 1908):
  - *"Later generations will regard Mengenlehre [set theory] as a disease from which one has recovered."*
- David Hilbert (1926):
  - *"No one shall expel us from the Paradise that Cantor has created."*
- Main approaches:
  - Formalism (Hilbert's programme)
    - ground all theories on a finite, complete set of axioms, and prove they are consistent
  - Intuitionism (constructive nature)
    - Brouwer, Kleene, ...
    - Poincaré: *"Logic remains barren unless fertilized by intuition."*

A Brief History of Logic

Bertrand Russell (1872–1970)



- Russell's Paradox (1901) — Let $S = \{x \mid x \notin x\}$. Does $S \in S$?
- Principia Mathematica (1910–13)
  - with A. Whitehead
  - foundations of mathematics
    - sets (using type theory) and logic
  - goal: forbid self-reference
  - takes 362 pages to prove $1 + 1 = 2$:

$*54{\cdot}43.\quad \vdash :. \alpha, \beta \in 1 \,.\, \supset : \alpha \cap \beta = \Lambda \,.\, \equiv \,.\, \alpha \cup \beta \in 2$

Dem.

$\vdash . *54{\cdot}26 . \supset \vdash :. \alpha = \iota`x . \beta = \iota`y . \supset : \alpha \cup \beta \in 2 . \equiv . x \neq y .$

$[*51{\cdot}231] \qquad \equiv . \iota`x \cap \iota`y = \Lambda .$

$[*13{\cdot}12] \qquad \equiv . \alpha \cap \beta = \Lambda \qquad (1)$

$\vdash . (1) . *11{\cdot}11{\cdot}35 . \supset$

$\vdash :. (\exists x, y) . \alpha = \iota`x . \beta = \iota`y . \supset : \alpha \cup \beta \in 2 . \equiv . \alpha \cap \beta = \Lambda \qquad (2)$

$\vdash . (2) . *11{\cdot}54 . *52{\cdot}1 . \supset \vdash . \text{Prop}$

From this proposition it will follow, when arithmetical addition has been defined, that $1 + 1 = 2$.

- Ludwig Wittgenstein (student): *Tractatus Logico-Philosophicus*
  - **5.43**: …"*But the propositions of logic say the same thing. That is, nothing.*"

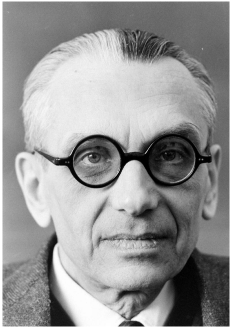
A Brief History of Logic

David Hilbert (1862–1943)

- Hilbert's problems (1900):
  - 23 open problems presented at a conference in Paris as challenges for the 20th century
  - #1: Continuum hypothesis ($2^{\aleph_0} \stackrel{?}{=} \aleph_1$)
  - #2: *Are axioms of arithmetic consistent?*
  - *"In mathematics there is no* ignorabimus*."*
    - *"Ignoramus et ignorabimus."* (Bois-Reymond, on the limits of scientific knowledge)
- Hilbert's programme (started in 1920's):
  - formalisation of all mathematics using finite number of axioms, which can be manipulated using well-defined rules (formula game)
  - completeness: all truths are provable
  - consistency: no falsehood is provable
  - conservation: if a result can be obtained using "ideal objects" (e.g. uncountable sets), it can also be obtained using "real objects"
  - decidability: algorithm for deciding validity *(Entscheidungsproblem)*
- *"We must know — we shall know!"*

A Brief History of Logic

Kurt Gödel (1906–1978)

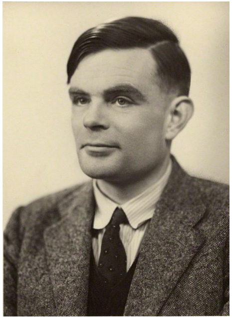


- born in Brno (Pekařská 3)
- member of Vienna circle (moved there in 1924)
  - making philosophy scientific with the help of modern logic
  - unified science
- 1929: Gödel's completeness theorem
  - Every theorem of FOL is provable.
- 1931: Gödel's incompleteness theorem (lecture 3)
  - Every sufficiently strong consistent formal system is incomplete.
    - incomplete = there exist unprovable statements
  - Gödel effectively encoded the formal system of Principia Mathematica into number theory, thus creating self-reference.
  - Destroys Hilbert's programme
    - John von Neumann's reaction: *"It's all over."*

A Brief History of Logic

Alan Turing (1912–1954)

- *Father of Computer Science*
- 1936: solves *Entscheidungsproblem*
  - defines the Turing machine as a formal notion of an *algorithm*
  - there is no algorithm that decides validity of FOL statements (diagonalization argument)
  - final blow to Hilbert's programme
  - also solved by Alonzo Church (λ-calculus)
- WW2: helps to break *Enigma*
- 1950: Turing test
  - can be used to evaluate maturity of artificial intelligence



A Brief History of Logic

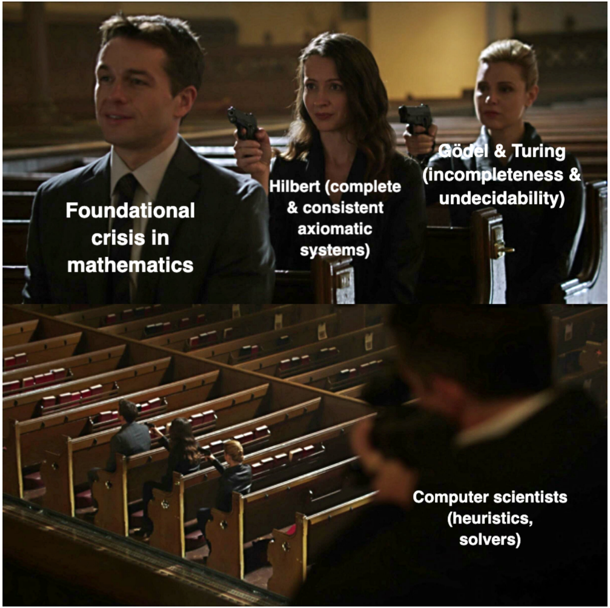
Current status

- Mathematicians don't really care about formal logic
- **Logic is the basis of computer science**

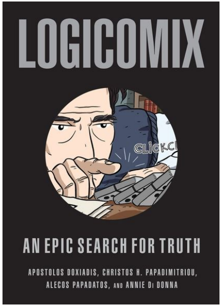
> *"Computer science is the continuation of logic by other means."*
> *—Georg Gottlob, 2007*

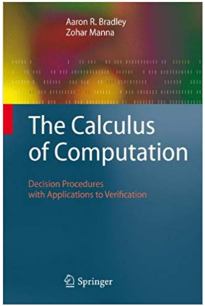
- incompleteness & undecidability
  - real problems in analysis of programs
  - theoretical computer science: finding heuristics that work in many practical cases

A Brief History of (Modern) Logic (in 1 Meme)



Literature





- https://wiki.lesswrong.com/wiki/Highly_Advanced_Epistemology_101_for_Beginners

Formal systems

A formal system consists of

- alphabet: contains **symbols**
- grammar: rules for constructing **well-formed formulae** (wff)
- axioms/axiom schemata: contain wff
- inference rules

Logic

Division of **logics**:

- classical
- non-classical — examples:
  - intuitionistic logic: no LEM, DNE, DeML
  - many-valued logic: not only true/false (e.g. fuzzy logic)

According to allowed values of variables

- propositional logic (PL): true/false
- first-order predicate logic (FOL): an **element** of the **universe**
- second-order logic: a **relation** on the universe
  - monadic second-order logic (MSO): unary relation (i.e. **sets**) only
- ...

Extensions:

- modal logic, temporal logic, dynamic logic, ...

Formal systems in CS

In CS, logics are studied from the point of view of:

- **decidability**: can we, for any formula φ in $\mathcal{F}$, decide if $\models_{\mathcal{F}} \varphi$?
  - decidable logics, e.g.
    - propositional logic (PL),
    - fragments of first-order logic (FOL): e.g. Presburger arithmetic $(\mathbb{N}, +)$
    - fragments of second-order logic: e.g. MSO(Str), WSkS
  - undecidable logics (cf. Gödel's incompleteness theorems), e.g.
    - general first-order logic: enough if contains Peano arithmetic $(\mathbb{N}, +, \cdot)$
    - general second-order logic: enough if contains Presburger arithmetic

For decidable logics, we study their

- **decision procedures**: algorithms that decide whether $\models_{\mathcal{F}} \varphi$
- **complexity**: how difficult is it to decide validity?
- **expressivity**: what is expressible using the logic?
  - higher expressivity $\approx$ higher complexity

Proof Theory vs. Model Theory

- Logic itself can be divided into several branches depending on how we look at formulae:
  - syntactically: proof theory
    - studies proofs as first-class citizens
    - e.g. the **semantic argument** proof technique introduced later
  - semantically: model theory
    - focuses on the entities denoted by formulae
    - e.g. the **decision procedures** for Presburger arithmetic
    - **model**: an entity satisfying a formula

Logic in Computer Science

- An essential knowledge of every SW architect/designer/engineer.
- Ubiquitous in CS.

  Examples:
  - Hardware design: 2-bit multiplexor

    $out \equiv (in_0 \wedge \neg addr) \vee (in_1 \wedge addr)$ (propositional logic—PL)
  - Function contracts: Sorting sorts.

    $\{array(\mathtt{x})\}$

    $\mathtt{y} = \mathtt{sort}(\mathtt{x});$ (first-order predicate logic—FOL)

    $\{array(\mathtt{y}) \quad \wedge \quad \forall 0 \leq i < j < len(\mathtt{y}) : \mathtt{y}[i] \leq \mathtt{y}[j] \quad \wedge \quad \mathtt{y} = perm(\mathtt{x})\}$
  - System specifications: Every request is eventually granted.

    $\Box(Req \rightarrow \Diamond Grt)$ (linear temporal logic—LTL)

Logic in Computer Science

- Examples (cont.):
  - Reasoning in abstract structures: Node y is reachable from node x in (finite) graph $\mathcal{G}$.

$$reach(x,y) \equiv \exists V \subseteq nodes(\mathcal{G}) \colon x, y \in V \land \forall a \in V : \big(a \neq y \to \exists! b \in V \colon edge_{\mathcal{G}}(a,b) \land a \neq b\big) \land \big(a \neq x \to \exists! b \in V \colon edge_{\mathcal{G}}(b,a) \land a \neq b\big)$$

(monadic second-order logic on graphs—MSO(Graphs))

  - Artificial intelligence, type theory, databases, ...
- Working with logic in CS:
  - Propositional logic: SAT (**SAT**isfiability) solvers
    - MiniSat, Glucose, Lingeling, ... (SAT competition)
  - First-order logic:
    - SMT (**S**atisfiability **M**odulo **T**heories) solvers: Z3, CVC5, Yices, ... (SMT-COMP)
    - Automated theorem provers: Vampire, E, iProver, ... (CASC)
    - Interactive theorem provers: Coq, Isabelle, Lean, ...
  - Second-order logic: specialized solvers: Mona [WSkS], ...

Propositional Logic

Propositional Logic

- Reasons about propositions
  - substituted by propositional variables from the set $\mathbb{X} = \{X, Y, \ldots\}$
  - statements that can be either true or false
- Propositions are atomic:
  - we do not look inside them (we will later, in FOL)
  - they have no implicit relation
    - relations needs to be given explicitly by PL formulae

Propositional Logic — Examples

Example (1)

If the train arrives late and there are no taxis at the station, then John is late for his date. John is not late for his date. The train did arrive late. **Therefore**, there were taxis at the station.

Example (2)

If it is raining and John does not have his umbrella with him, then he will get wet. John is not wet. It is raining. **Therefore**, John has his umbrella with him.

Both examples have the same structure:

	Example (1)	**Example (2)**
F	the train is late	it is raining
G	there are taxis at the station	John has his umbrella with him
H	John is late for his date	John is wet

If F and not G, then H. Not H. F. **Therefore**, G.

$$((F \land \neg G) \to H) \land (\neg H) \land (F) \to (G)$$

Syntax

$\varphi ::= X$	occurrence of a propositional variable $X \in \mathbb{X}$
$\mid \bot \quad (0)$	false
$\mid \top \quad (1)$	true
$\mid (\neg\varphi)$	negation, pronounced "not"
$\mid (\varphi_1 \wedge \varphi_2)$	conjunction, pronounced "and"
$\mid (\varphi_1 \vee \varphi_2)$	disjunction, pronounced "or"
$\mid (\varphi_1 \rightarrow \varphi_2)$	(material) implication, pronounced "implies"
$\mid (\varphi_1 \leftrightarrow \varphi_2)$	iff (material biconditional, equivalence), pronounced "if and only if"

Example

$$((A \rightarrow B) \leftrightarrow ((\neg A) \vee B))$$

- Parenthesis: linearize the syntax tree
  - sometimes can be avoided (top-level formula, negation)
- Syntax tree of φ: a derivation tree of φ

Semantics

- Interpretation I (variable assignment):
  - Assigns every variable from $\mathbb{X}$ a truth value:

$$I : \mathbb{X} \to \{true, false\}$$

e.g. $I = \{X \mapsto true, Y \mapsto false, \ldots\}$

- Truth value:
  - The truth value of φ under I is defined inductively using the table

ψ_1	ψ_2	$\neg\psi_1$	$\psi_1 \vee \psi_2$	$\psi_1 \wedge \psi_2$	$\psi_1 \to \psi_2$	$\psi_1 \leftrightarrow \psi_2$
0	0	1	0	0	1	1
0	1	1	1	0	1	0
1	0	0	1	0	0	0
1	1	0	1	1	1	1

- Example: $(X \wedge Y) \to (X \vee \neg Y)$

Semantics

- Truth value (cont.):
  - a better notation:
    - $I \models \varphi$: iff φ evaluates to true under I
    - $I \not\models \varphi$: iff φ evaluates to false under I
  - inductive definition:
    - base cases: $I \models \top$
      $I \not\models \bot$
      $I \models X$ iff $I[X] = \top$ (for $X \in \mathbb{X}$)
    - inductive steps: $I \models \neg\psi$ iff $I \not\models \psi$
      $I \models \psi_1 \wedge \psi_2$ iff $I \models \psi_1$ and $I \models \psi_2$
      $I \models \psi_1 \vee \psi_2$ iff $I \models \psi_1$ or $I \models \psi_2$
      $I \models \psi_1 \rightarrow \psi_2$ iff, if $I \models \psi_1$ then $I \models \psi_2$
      $I \models \psi_1 \leftrightarrow \psi_2$ iff $I \models \psi_1$ and $I \models \psi_2$, or $I \not\models \psi_1$ and $I \not\models \psi_2$
    - $\not\models$: well defined from the previous
  - if $I \models \psi$, we say that I is a model of ψ

Semantics

Example

Consider the formula

$$\varphi = (X \wedge Y) \rightarrow (X \vee \neg Y)$$

and the interpretation

$$I = \{X \mapsto \top, Y \mapsto \bot\}.$$

Compute the truth value of φ under I as follows:

Semantics

Example

Consider the formula

$$\varphi = (X \wedge Y) \to (X \vee \neg Y)$$

and the interpretation

$$I = \{X \mapsto \top, Y \mapsto \bot\}.$$

Compute the truth value of φ under I as follows:

1.	$I \models X$	since $I[X] = \top$
2.	$I \not\models Y$	since $I[Y] = \bot$
3.	$I \models \neg Y$	by 2 and semantics of $\neg$
4.	$I \not\models X \wedge Y$	by 2 and semantics of $\wedge$
5.	$I \models X \vee \neg Y$	by 1 and semantics of $\vee$
6.	$I \models \varphi$	by 4 and semantics of $\to$

Note that we follow the order from *simpler* to *more complex*.

Satisfiability and Validity

- Satisfiability and validity are the key concepts in logic.
- Satisfiability: *can* a formula φ be true?
  - Is there an interpretation I such that $I \models \varphi$?
- Validity: is a formula φ *always* true?
  - Does it for all interpretations I hold that $I \models \varphi$?
  - Denoted as $\models \varphi$.
  - Tautology: a *valid* formula
  - Contradiction: an *unsatisfiable* formula

Satisfiability and Validity

- Satisfiability and validity are the key concepts in logic.

- Satisfiability: *can* a formula φ be true?
  - Is there an interpretation I such that $I \models \varphi$?

- Validity: is a formula φ *always* true?
  - Does it for all interpretations I hold that $I \models \varphi$?
  - Denoted as $\models \varphi$.
  - Tautology: a *valid* formula
  - Contradiction: an *unsatisfiable* formula

- Dual concepts:
  - φ is valid iff $\neg\varphi$ is unsatisfiable
  - φ is satisfiable iff $\neg\varphi$ is invalid

- Checking satisfiability and validity:
  - Option 1: construct truth tables
  - Option 2: semantic arguments (next slide)
  - Options 3–n: natural deduction, resolution, Hilbert-style system, ...

Semantic Argument

Semantic argument:

- a method for establishing validity of a formula
- also called *analytic tableau*, *semantic tableau*, *truth tree*, ...
- more complicated than a *truth table*, but we will also use it for predicate logic (where truth tables are not applicable)
- start by assuming that a formula φ is invalid, and show that:
  - either all branches lead to a contradiction (then $\models \varphi$), or
  - some branch does not (then $\not\models \varphi$ and there exists a falsifying interpretation I s.t. $I \not\models \varphi$)
- the proof proceeds by applying proof rules:

$$\frac{premises}{deductions}$$

  - if all $premises$ hold, we can deduce all $deductions$.

Semantic Argument (proof rules)

- negation:

$$\frac{I \models \neg\varphi}{I \not\models \varphi} \qquad\qquad \frac{I \not\models \neg\varphi}{I \models \varphi}$$

- conjunction:

$$\frac{I \models \varphi \wedge \psi}{\begin{array}{c} I \models \varphi \\ I \models \psi \end{array}} \qquad\qquad \frac{I \not\models \varphi \wedge \psi}{I \not\models \varphi \quad \mid \quad I \not\models \psi}$$

('|' forks computation in two branches that both need to be proved)

- disjunction:

$$\frac{I \models \varphi \vee \psi}{I \models \varphi \quad \mid \quad I \models \psi} \qquad\qquad \frac{I \not\models \varphi \vee \psi}{\begin{array}{c} I \not\models \varphi \\ I \not\models \psi \end{array}}$$

Semantic Argument (proof rules)

- implication:

$$\frac{I \models \varphi \rightarrow \psi}{I \not\models \varphi \quad | \quad I \models \psi}$$

$$\frac{I \not\models \varphi \rightarrow \psi}{\begin{array}{c} I \models \varphi \\ I \not\models \psi \end{array}}$$

- iff:

$$\frac{I \models \varphi \leftrightarrow \psi}{I \models \varphi \wedge \psi \quad | \quad I \not\models \varphi \vee \psi}$$

$$\frac{I \not\models \varphi \leftrightarrow \psi}{I \models \varphi \wedge \neg\psi \quad | \quad I \models \neg\varphi \wedge \psi}$$

- contradiction:

$$\frac{I \models \varphi \qquad I \not\models \varphi}{I \models \bot}$$

Semantic Argument (proofs)

- In a semantic argument, a proof of φ is a tree.

Example

1. $I \not\models \varphi$
2. $I \not\models \neg P \rightarrow \neg Q$

$\vdots$

- The root is an assumption that φ is invalid:

Example

1. $I \not\models \varphi$ (assumption)

Semantic Argument (proofs)

- Every other line l is obtained as a deduction of a proof rule where the premise(s) are **before** l.

Example

$\vdots$

7. $I \models P \land R$ $\quad\ldots$
8. $I \models P$ $\quad$ by 7 and semantics of $\land$

$\vdots$

Semantic Argument (example)

Example (1)

Prove that the formula $\psi: (P \wedge Q) \rightarrow (P \vee \neg Q)$ is valid.

Semantic Argument (example)

Example (1)

Prove that the formula $\psi : (P \land Q) \rightarrow (P \lor \neg Q)$ is valid.

Solution.

Assume ψ is invalid, i.e., there exists I s.t. $I \not\models \psi$. Then,

1.	$I \not\models (P \land Q) \rightarrow (P \lor \neg Q)$	assumption
2.	$I \models P \land Q$	by 1 and semantics of $\rightarrow$
3.	$I \not\models P \lor \neg Q$	by 1 and semantics of $\rightarrow$
4.	$I \models P$	by 2 and semantics of $\land$
5.	$I \not\models P$	by 2 and semantics of $\lor$
6.	$I \models \bot$	from 4 and 5

$\square$

Semantic Argument (example)

Example (2)

Prove that the formula $\psi \colon \big((P \rightarrow Q) \land (Q \rightarrow R)\big) \rightarrow (P \rightarrow R)$ is valid.

Semantic Argument (example)

Example (2)

Prove that the formula $\psi\colon \big((P \to Q) \land (Q \to R)\big) \to (P \to R)$ is valid.

Solution.
Assume ψ is invalid, i.e., there exists I s.t. $I \not\models F$. Then,

1.	$I \not\models \big((P \to Q) \land (Q \to R)\big) \to (P \to R)$	assumption
2.	$I \models (P \to Q) \land (Q \to R)$	by 1 and semantics of $\to$
3.	$I \not\models (P \to R)$	by 1 and semantics of $\to$
4.	$I \models P$	by 3 and semantics of $\to$
5.	$I \not\models R$	by 3 and semantics of $\to$
6.	$I \models P \to Q$	by 2 and semantics of $\land$
7.	$I \models Q \to R$	by 2 and semantics of $\land$

To discharge 6 and 7, we need to fork the proof.

continue on next slide ...

Semantic Argument (example)

Example (2 *cont.*)

$\vdots$

4.	$I \models P$	by 3 and semantics of $\rightarrow$
5.	$I \not\models R$	by 3 and semantics of $\rightarrow$
6.	$I \models P \rightarrow Q$	by 2 and semantics of $\wedge$
7.	$I \models Q \rightarrow R$	by 2 and semantics of $\wedge$

First, we discharge 6 by forking into branches 'a' and 'b':

a8. $I \not\models P$ $\{6 \text{ and } \rightarrow\}$
a9. $I \models \bot$ $\{4 \text{ and a8}\}$
[branch closed]

b8. $I \models Q$ $\{6 \text{ and } \rightarrow\}$

Then, we discharge 7 by forking the 'b' branch into 'ba' and 'bb':

ba9. $I \not\models Q$ $\{7 \text{ and } \rightarrow\}$
ba10. $I \models \bot$ $\{\text{b8 and ba9}\}$
[branch closed]

bb9. $I \models R$ $\{7 \text{ and } \rightarrow\}$
bb10. $I \models \bot$ $\{\text{5 and bb9}\}$
[branch closed]

□

Semantic Argument (modus ponens)

Modus ponens (MP) is the following useful rule:

$$\frac{I \models F \qquad I \models F \to G}{I \models G}$$

MP is sometimes also called implication elimination.

Equivalence and Implication

Equivalence and implication are used to talk about pairs of formulae.

- (logical) equivalence ($\Leftrightarrow$):
  - $F \Leftrightarrow G$ if F and G evaluate to the same truth value under all I
  - $F \Leftrightarrow G$ can be proved by showing $\models F \leftrightarrow G$
- (logical) implication ($\Rightarrow$):
  - $F \Rightarrow G$ if for every I: if $I \models F$ then $I \models G$
  - $F \Rightarrow G$ can be proved by showing $\models F \rightarrow G$
  - also called: logical consequence, entailment
  - also denoted: $F \models G$

What is the difference between $F \leftrightarrow G$ and $F \Leftrightarrow G$?

- $F \leftrightarrow G$ is a formula of PL.
- $F \Leftrightarrow G$ is a statement about formulae F and G. It is **not** a formula.
  - (More precisely, $F \Leftrightarrow G$ is a statement in the metalanguage we use to talk about PL.)

Similarly for $F \rightarrow G$ and $F \Rightarrow G$.

Useful Equivalences 1

$$\begin{array}{rcll}
F & \Leftrightarrow & \neg\neg F & \text{(double negative elimination)} \\
\neg\top & \Leftrightarrow & \bot & \\
\neg\bot & \Leftrightarrow & \top & \\
\neg(F \wedge G) & \Leftrightarrow & \neg F \vee \neg G & \text{(De Morgan's law)} \\
\neg(F \vee G) & \Leftrightarrow & \neg F \wedge \neg G & \text{(De Morgan's law)} \\
F \rightarrow G & \Leftrightarrow & \neg F \vee G & \\
F \leftrightarrow G & \Leftrightarrow & (F \rightarrow G) \wedge (G \rightarrow F) & \\
F \wedge (G \wedge H) & \Leftrightarrow & (F \wedge G) \wedge H & \text{(associativity)} \\
F \vee (G \vee H) & \Leftrightarrow & (F \vee G) \vee H & \text{(associativity)} \\
F \wedge (G \vee H) & \Leftrightarrow & (F \wedge G) \vee (F \wedge H) & \text{(distributivity)} \\
F \vee (G \wedge H) & \Leftrightarrow & (F \vee G) \wedge (F \vee H) & \text{(distributivity)}
\end{array}$$

Useful Equivalences 2

$$
\begin{array}{rcll}
F \to G & \Leftrightarrow & \neg G \to \neg F & \text{(contrapositive, modus tollens)} \\
F \to (G \to H) & \Leftrightarrow & (F \wedge G) \to H & \text{(exportation)} \\
F \wedge \neg F & \Leftrightarrow & \bot & \text{(law of noncontradiction)} \\
F \vee \neg F & \Leftrightarrow & \top & \text{(law of excluded middle)} \\
F \vee F & \Leftrightarrow & F & \text{(idempotence)} \\
F \wedge F & \Leftrightarrow & F & \text{(idempotence)} \\
F \vee \bot & \Leftrightarrow & F & \\
F \wedge \top & \Leftrightarrow & F & \\
F \vee \top & \Leftrightarrow & \top & \\
F \wedge \bot & \Leftrightarrow & \bot & \\
(F \to G) \wedge (F \to \neg G) & \Leftrightarrow & \neg F &
\end{array}
$$

Substitution

Substitution σ:

- mapping from formulae to formulae $\sigma\colon \{F_1 \mapsto G_1, \ldots, F_n \mapsto G_n\}$
- domain: $\mathrm{dom}(\sigma) = \{F_1, \ldots, F_n\}$, range: $\mathrm{rng}(\sigma) = \{G_1, \ldots, G_n\}$
- the formula $F\sigma$ is obtained from F by replacing every F_i with G_i
- all replacements occur at once
  - if there are F_j and F_k such that F_k is a strict subformula of F_j and F_j occurs in F, then we substitute F_j with G_j

Example

$$F\colon (P \wedge Q) \quad \rightarrow \quad (P \vee \neg Q)$$
$$\sigma\colon \{P \mapsto R,\ \ P \wedge Q \ \mapsto\ P \rightarrow Q\}$$

$$F\sigma\colon (P \rightarrow Q) \quad \rightarrow \quad (R \vee \neg Q)$$

Proposition (Substitution of Equivalent Formulae)

If, given σ, for each i it holds that $F_i \Leftrightarrow G_i$, then $F \Leftrightarrow F\sigma$.

Substitution

Variable substitution:

- substitution σ such that $\mathrm{dom}(\sigma) \subseteq \mathbb{X}$

Example

$$\sigma = \{F \mapsto J \wedge H, \quad G \mapsto H \to J\}$$

Proposition

If $\models F$ and σ is a variable substitution, then $\models F\sigma$.

Example

If

$$\models F \;\to\; (G \to F),$$

then also

$$\models (H \wedge I) \quad\to\quad ((\neg H) \;\to\; (H \wedge I)).$$

Normal Forms (NNF)

Negation Normal Form (NNF):

- contains only $\wedge$, $\vee$, and $\neg$ as connectives
- $\neg$ appears only in front of variables

Example

Let

$$F:\neg(P\rightarrow\neg(P\wedge Q)).$$

The formula

$$G:P\wedge Q$$

is equivalent to F and is in NNF.

Normal Forms (DNF)

Disjunctive Normal Form (DNF):

- is a disjunction of conjunction of literals:

$$\bigvee_i \bigwedge_j \ell_{i,j}$$

- a literal is a variable (X) or its negation ($\neg X$)

Example

Let

$$F : (P \vee \neg\neg Q) \wedge (R \rightarrow S).$$

The formula

$$G : (P \wedge \neg R) \vee (P \wedge S) \vee (Q \wedge \neg R) \vee (Q \wedge S)$$

is equivalent to F and is in DNF.

Normal Forms (CNF)

Conjunctive Normal Form (CNF):

- is a conjunction of disjunction of literals:

$$\bigwedge_i \bigvee_j \ell_{i,j}$$

- a disjunction of literals is called a clause

Example

Let

$$F\colon (P \wedge \neg\neg Q) \vee (R \rightarrow S).$$

The formula

$$G\colon (P \vee \neg R \vee S) \wedge (Q \vee \neg R \vee S)$$

is equivalent to F and is in CNF.

Normal Forms (CNF)

SAT:

- the problem of deciding whether a formula in CNF is satisfiable

Proposition (Complexity)

SAT is $\mathbf{NP}$-complete.

- $\mathbf{NP}$-complete problems (informally):
  - the best algorithm known is exponential
  - but if we guess a solution, we can quickly check if it is correct
    - e.g., for SAT, if we guess I, we can quickly check whether $I \models \varphi$
  - other examples:
    - travelling salesman problem, knapsack, graph colouring, . . .
  - often considered not efficiently solvable
  - BUT!
    - SAT solvers: programs that can quickly solve many real-life instances
    - other $\mathbf{NP}$-complete problems are often reduced to SAT

Notes

- $\varphi \to \psi$ is sometimes written as $\varphi \supset \psi$.
- In $\varphi \to \psi$, we call φ the antecedent and ψ the consequent.

- Horn clause: a clause that has at most one positive literal
  - often represented in an implication form (cf. PROLOG):

$$(F \vee \neg G \vee \neg H \vee \neg I) \qquad \Leftrightarrow \qquad F \leftarrow (G \wedge H \wedge I)$$

  - SAT for Horn clauses (HORNSAT) is linear-time (can you see why?)

- Resolution: an inference technique for CNF:

$$\frac{F_1 \vee \cdots \vee F_n \vee H \qquad G_1 \vee \cdots \vee G_m \vee \neg H}{F_1 \vee \cdots \vee F_n \vee G_1 \vee \cdots G_m}$$

References

[A.R. Bradley and Z. Manna. The Calculus of Computation.]