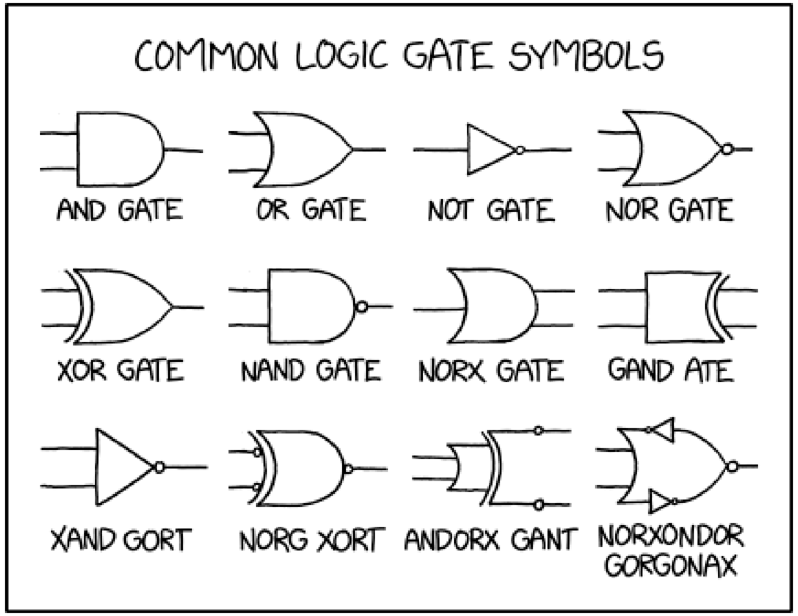
Logic and Computability

Lecture 1

Propositional Logic

Bettina Könighofer
bettina.koenighofer@lamarr.at

Stefan Pranger
stefan.pranger@iaik.tugraz.at



https://xkcd.com/2497/

Why are we interested in Logic?

Why are we interested in Logic?

- Logic in CS aims to formulate specifications such that we can *reason* about systems *formally*.

Why are we interested in Logic?

- Logic in CS aims to formulate specifications such that we can *reason* about systems *formally*.

- Automatically prove that the system is correct (does what it is supposed to do)

Why are we interested in Logic?

- Logic in CS aims to formulate specifications such that we can *reason* about systems *formally*.

- Automatically prove that the system is correct (does what it is supposed to do)
- First step: Formal specification that accurately captures desired system behavior

Why are we interested in Logic?

- Logic in CS aims to formulate specifications such that we can *reason* about systems *formally*.

- Automatically prove that the system is correct (does what it is supposed to do)
- First step: Formal specification that accurately captures desired system behavior
- Automatically prove that system satisfies specification
  - Use techniques like model checking, theorem proving, SMT solving

Why are we interested in Logic?

- **Example**: Prove that the argumentation is valid

1. If the plane arrives late and there are no taxis at the airport, then Alice is late for her appointment.
2. Alice is not late for her appointment.
3. The plane did arrive late.
4. *Therefore*, there were taxis at the airport.

Why are we interested in Logic?

Knowledge that we have
Facts that we know are true

- **Example**: Prove that the argumentation is valid

1. If the plane arrives late and there are no taxis at the airport, then Alice is late for her appointment.
2. Alice is not late for her appointment.
3. The plane did arrive late.
4. *Therefore*, there were taxis at the airport.

Why are we interested in Logic?

Knowledge that we have
Facts that we know are true

- **Example**: Prove that the argumentation is valid

1. If the plane arrives late and there are no taxis at the airport, then Alice is late for her appointment.
2. Alice is not late for her appointment.
3. The plane did arrive late.
4. *Therefore*, there were taxis at the airport.

Deduce new knowledge:
Prove that sentence 4 follows from the sentences 1,2, and 3.

Why are we interested in Logic?

- **Example**: Prove that the argumentation is valid

1. If the plane arrives late and there are no taxis at the airport, then Alice is late for her appointment.
2. Alice is not late for her appointment.
3. The plane did arrive late.
4. *Therefore*, there were taxis at the airport.

Why are we interested in Logic?

- **Example**: Prove that the argumentation is valid

1. If the plane arrives late and there are no taxis at the airport, then Alice is late for her appointment.
2. Alice is not late for her appointment.
3. The plane did arrive late.
4. *Therefore*, there were taxis at the airport.

p... the plane arrives late
t ... there are taxis at the airport
l... Alice is late for the appointment

Why are we interested in Logic?

- **Example**: Prove that the argumentation is valid

1. If the plane arrives late and there are no taxis at the airport, then Alice is late for her appointment.	1. $(p \land \neg t) \to l$
2. Alice is not late for her appointment.	2. $\neg l$
3. The plane did arrive late.	3. p
4. *Therefore*, there were taxis at the airport.	4. t

p… the plane arrives late
t … there are taxis at the airport
l… Alice is late for the appointment

Why are we interested in Logic?

- **Example**: Prove that the argumentation is valid

1. If the plane arrives late and there are no taxis at the airport, then Alice is late for her appointment.
2. Alice is not late for her appointment.
3. The plane did arrive late.
4. *Therefore*, there were taxis at the airport.

1. $(p \land \neg t) \to l$
2. $\neg l$
3. p
4. t

p… the plane arrives late
t … there are taxis at the airport
l… Alice is late for the appointment

How can we prove that?

- Natural Deduction (in 2 weeks)
- Defines fixed set of rewriting rules
- Creates watertight proves.

Proofs can be checked (and generated) automatically.

Why are we interested in Logic?

- Prove that the argumentation is valid:

1. If the sun is shining and John has no sunscreen,
   then John gets a sunburn.
2. John has no sunburn.
3. The sun is shining.
4. *Therefore*, John has a sunscreen.

Why are we interested in Logic?

- Prove that the argumentation is valid:

1. If the sun is shining and John has no sunscreen,
   then John gets a sunburn.
2. John has no sunburn.
3. The sun is shining.
4. *Therefore*, John has a sunscreen.

s… the sun is shining
c … John has a sunscreen
b… John has a sunburn

Why are we interested in Logic?

- Prove that the argumentation is valid:

1. If the sun is shining and John has no sunscreen, then John gets a sunburn.
2. John has no sunburn.
3. The sun is shining.
4. *Therefore*, John has a sunscreen.

1. $(s \wedge \neg c) \rightarrow b$
2. $\neg b$
3. s
4. c

s… the sun is shining
c … John has a sunscreen
b… John has a sunburn

Why are we interested in Logic?

- Prove that the argumentation is valid:

1. If the sun is shining and John has no sunscreen, then John gets a sunburn.
2. John has no sunburn.
3. The sun is shining.
4. *Therefore*, John has a sunscreen.

1. $(s \wedge \neg c) \rightarrow b$
2. $\neg b$
3. s
4. c

s… the sun is shining
c … John has a sunscreen
b… John has a sunburn

Same as before ☺
Reuse proof from before

Outline

- Declarative Sentences
- Syntax
  - Symbols & Rules
  - Parse Tree
- Semantics
  - Meaning
  - Models
  - Truth Tables
  - Validity, Satisfiability
- Examples

Declarative Sentence

Declarative Sentence

- Prop. logic is based on declarative sentences (also called propositions)

Declarative Sentence

- Prop. logic is based on declarative sentences (also called propositions)

- A declarative sentence
  - states a fact
  - can be *true* or *false*

Declarative Sentence

- Prop. logic is based on declarative sentences (also called propositions)
- A declarative sentence
  - states a fact
  - can be *true* or *false*
- Examples:
  - Simple
    - “Florian is back in Austria.”
    - “Tomorrow is Wednesday.”
    - “10 divided by 5 is 3.”
  - With Structure
    - “Tomorrow is Saturday and not Sunday.”

Non-Declarative Sentence

Examples:

- Questions
  - “What time is it?”
- Commands
  - “Do your homework!”
- Exclamations
  - “Oh my god!”
- Various others
  - “Good morning.”
  - “Ready, steady, go.”
  - “May the force be with you.”
  - …

Outline

- Declarative Sentences
- **Syntax**
  - Symbols & Rules
  - Parse Tree
- Semantics
  - Meaning
  - Models
  - Truth Tables
  - Validity, Satisfiability, Entailment & Equivalence
- Examples

Syntax vs Semantics

Syntax vs Semantics

The **syntax** of a logical language consists

- of a **set of symbols** and
- **rules** for combining them

to **form formulas of the language**.

Syntax vs Semantics

The **syntax** of a logical language consists

- of a **set of symbols** and
- **rules** for combining them

to **form formulas of the language**.

The **semantics** of a logic provides its **meaning**. In prop logic, the meaning is given by the truth values $\top$ (true) and $\bot$(false).
→ The semantics of prop logic assigns a meaning ($\bot$, $\top$) to prop logic formulas.

Syntax of Propositional Logic

- Defines symbols and rules to form formulas in propositional logic

Syntax of Propositional Logic

- Defines symbols and rules to form formulas in propositional logic
    - Example: The string "$\boldsymbol{pq}()(\wedge)$" is no propositional logic formula
        - Uses allowed symbols, but does not adhere to the rules for combining the symbols

Syntax of Propositional Logic

- Defines symbols and rules to form formulas in propositional logic
- Elements/Symbols

Syntax of Propositional Logic

- Defines symbols and rules to form formulas in propositional logic

- Elements/Symbols
  - Turth symbols $\top$ (“true”) and $\bot$(“false”)

Syntax of Propositional Logic

- Defines symbols and rules to form formulas in propositional logic

- Elements/Symbols
  - Turth symbols $\top$ (“true”) and $\bot$(“false”)
  - Set of propositional variable symbols $\mathbf{p}$, $\mathbf{q}$, $\mathbf{r}$ …

Syntax of Propositional Logic

- Defines symbols and rules to form formulas in propositional logic

- Elements/Symbols
  - Turth symbols $\top$ ("true") and $\bot$("false")
  - Set of propositional variable symbols $\mathbf{p}$, $\mathbf{q}$, $\mathbf{r}$ ...
  - Logical connectors (logical operators, Boolean connectors)
    - Conjunction $\wedge$, Disjunction $\vee$, Negation $\neg$,
      Implication $\rightarrow$, Equivalence $\equiv$ or Biimplication $\leftrightarrow$, Exclusive Disjunction (XOR) $\oplus$

Syntax of Propositional Logic

- Defines symbols and rules to form formulas in propositional logic

- Elements/Symbols
  - Turth symbols ⊤ (“true”) and ⊥(“false”)
  - Set of propositional variable symbols **p**, **q**, **r** …
  - Logical connectors (logical operators, Boolean connectors)
    - Conjunction ∧, Disjunction ∨, Negation ¬, Implication →, Equivalence ≡ or Biimplication ↔ , Exclusive Disjunction (XOR) ⊕
  - Parentheses **()**

Syntax of Propositional Logic

- Defines symbols and rules to form formulas in propositional logic
- Rules

Syntax of Propositional Logic

- Defines symbols and rules to form formulas in propositional logic
- Rules
  - A propositional variable p, or a truth symbol $\top$ and $\bot$ are formulas

Syntax of Propositional Logic

- Defines symbols and rules to form formulas in propositional logic

- Rules
  - A propositional variable p, or a truth symbol $\top$ and $\perp$ are formulas
  - For any formula φ, $\neg\varphi$ is also a formula.

Syntax of Propositional Logic

- Defines symbols and rules to form formulas in propositional logic

- Rules
  - A propositional variable p, or a truth symbol $\top$ and $\bot$ are formulas
  - For any formula φ, $\neg\varphi$ is also a formula.
  - For any two formulas φ and ψ, the following are also formulas: $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \rightarrow \psi$, and $\varphi \leftrightarrow \psi$, $\varphi \oplus \psi$

Syntax of Propositional Logic

- Defines symbols and rules to form formulas in propositional logic

- Rules
  - A propositional variable p, or a truth symbol $\top$ and $\bot$ are formulas
  - For any formula φ, $\neg\varphi$ is also a formula.
  - For any two formulas φ and ψ, the following are also formulas: $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \rightarrow \psi$, and $\varphi \leftrightarrow \psi$, $\varphi \oplus \psi$
  - For any formula φ, (φ) is also a formula.

Syntax of Propositional Logic

- Defines symbols and rules to form formulas in propositional logic

- Rules
  - Backus-Naur form (BNF)
  - $\varphi := \top \mid \bot \mid < prop.variable > \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \rightarrow \varphi \mid \varphi \leftrightarrow \varphi \mid (\varphi)$

Notation: Atom and Literal

Notation: Atom and Literal

- An **atom** (atomic proposition) is a
  - propositional variable ($p, q, \ldots$),
  - or a truth symbol $\top$ or $\perp$.

Notation: Atom and Literal

- An **atom** (atomic proposition) is a
  - propositional variable ($p, q, \ldots$),
  - or a truth symbol $\top$ or $\bot$.
- A **literal** is an atom α or its negation $\neg\alpha$.

Operator Precedence

- Reduces number of parentheses needed

Operator Precedence

- Reduces number of parentheses needed

- Relative operator precedence
  - **Highest** $\neg \ \wedge \vee \rightarrow \leftrightarrow$ **Lowest**
  - Example

    $\neg a \vee b \wedge c \equiv (\neg a) \vee (b \wedge c)$

Operator Precedence

- Reduces number of parentheses needed

- Relative operator precedence
  - **Highest** $\neg \ \wedge \vee \rightarrow \leftrightarrow$ **Lowest**
  - Example
    $\neg a \vee b \wedge c \equiv (\neg a) \vee (b \wedge c)$

- Implication and bi-implication are right associative
  - Example:
    $\mathrm{a} \rightarrow b \rightarrow c \ \equiv a \rightarrow (b \rightarrow c)$

Operator Precedence

- Reduces number of parentheses needed

- Relative operator precedence
  - **Highest** $\neg \;\; \wedge \vee \rightarrow \leftrightarrow$ **Lowest**
  - Example
    $\neg a \vee b \wedge c \equiv (\neg a) \vee (b \wedge c)$

- Implication and bi-implication are right associative
  - Example:
    $\mathrm{a} \rightarrow b \rightarrow c \equiv a \rightarrow (b \rightarrow c)$

- Disjunction and conjunction are left associative
  - Example:
    $\mathrm{a} \wedge b \wedge c \equiv (a \wedge b) \wedge c$

Parse Tree

Parse Tree

- A string is a formula, if
  - all leaves are labelled with atomic propositions and
  - all other nodes are labelled with logical operators

Parse Tree

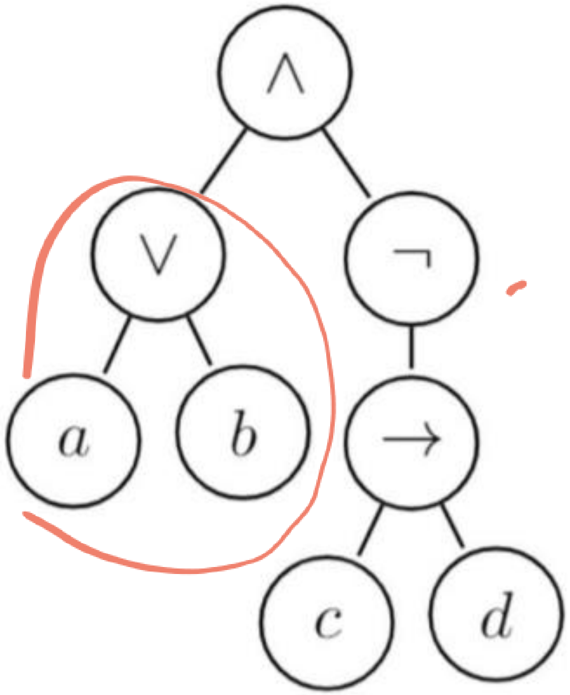
- A string is a formula, if
  - all leaves are labelled with atomic propositions and
  - all other nodes are labelled with logical operators

- **Example**:
  Is “$(a \vee b) \wedge (\neg\, (c \rightarrow d\,))$” a formula in prop logic?

Parse Tree

- A string is a formula, if
  - all leaves are labelled with atomic propositions and
  - all other nodes are labelled with logical operators

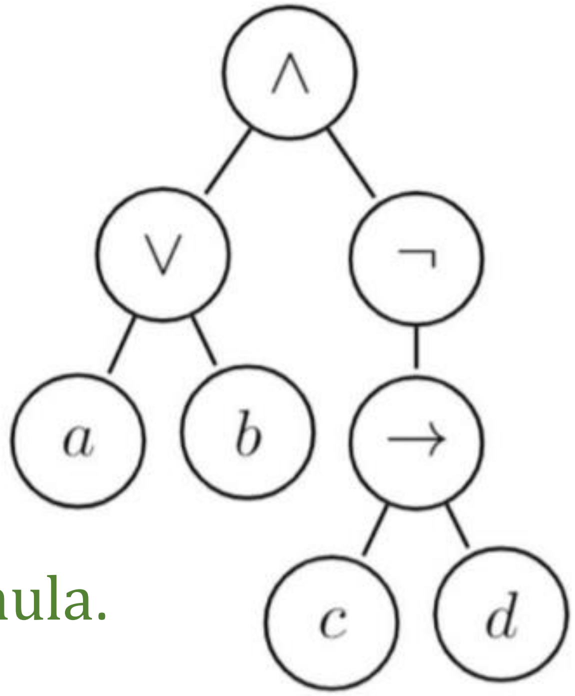
- **Example**:
  Is "$(a \lor b) \land (\neg (c \rightarrow d))$" a formula in prop logic?



Parse Tree

- A string is a formula, if
  - all leaves are labelled with atomic propositions and
  - all other nodes are labelled with logical operators

- **Example**:
  Is “$(a \lor b) \land \neg((c \to d))$” a formula in prop logic?



YES “$(a \lor b) \land \neg (c \to d)$" represents a formula.

Outline

- Declarative Sentences
- Syntax
  - Symbols & Rules
  - Parse Tree
- **Semantics**
  - Meaning
  - Models
  - Truth Tables
  - Validity, Satisfiability
- Examples

Syntax vs Semantics

The **syntax** of a logical language consists

- of a **set of symbols** and
- **rules** for combining them

to **form formulas of the language**.

The **semantics** of a logic provides its **meaning**. In prop logic, the meaning is given by the truth values $\top$ (true) and $\bot$(false).
→ The semantics of prop logic assigns meaning ($\bot$, $\top$) to prop logic formulas.

Semantics of Propositional Logic

- The semantics of prop logic assigns a **meaning (truth value $\perp$, $\top$)** to prop logic formulas.

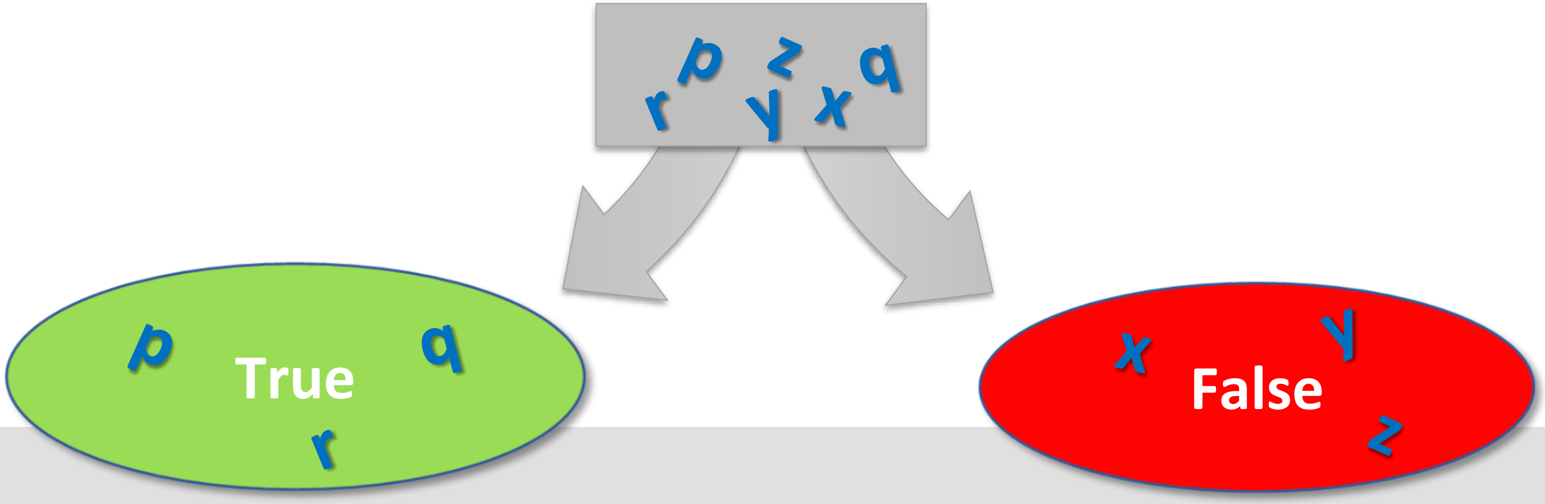
Semantics of Propositional Logic

- The semantics of prop logic assigns a **meaning (truth value $\bot$, $\top$)** to prop logic formulas.

- To define the semantics / assign truth values to formulas…
  - We need possibility to assign truth values to propositional variables
  - Use assignment to interpret formulas

Models $\mathcal{M}$

- Model ≅ Valuation ≅ Interpretation ≅ Assignment
- Assignment: $\{Atomic\ propositions\} \mapsto \{\top, \bot\}$

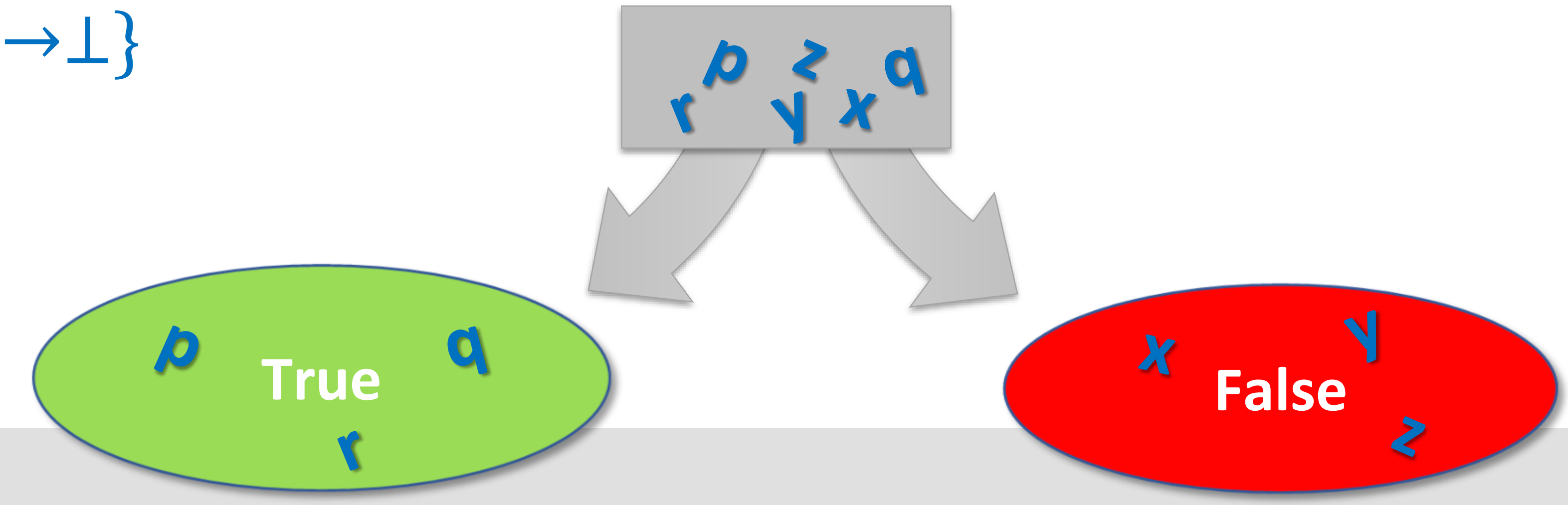


Models $\mathcal{M}$

- Model ≅ Valuation ≅ Interpretation ≅ Assignment
- Assignment: $\{\textit{Atomic propositions}\} \mapsto \{\top, \bot\}$

- Example
  - $\varphi = (p \vee y \vee \neg r) \wedge (\neg x \vee \neg q \vee z)$

  - $\mathcal{M}: \{p \rightarrow \top, q \rightarrow \top, r \rightarrow \top,$
    $x \rightarrow \bot, y \rightarrow \bot, z \rightarrow \bot\}$



Models $\mathcal{M}$

- $\varphi^{\mathcal{M}}$ ...φ is evaluated under $\mathcal{M}$

Models $\mathcal{M}$

- $\varphi^{\mathcal{M}}$ ...φ is evaluated under $\mathcal{M}$

- Satisfying Model: $\mathcal{M} \vDash \varphi$
  - $\mathcal{M}$ satisfies φ, or
  - φ evaluates to true under $\mathcal{M}$

  - Example
    - $\varphi = \overset{\top}{a} \vee \overset{\bot}{b}$
    - $\mathcal{M}: \{a \rightarrow \top, b \rightarrow \bot\}$
    - $\mathcal{M} \vDash \varphi$ or $\varphi^{\mathcal{M}} = \top$

Models $\mathcal{M}$

- $\varphi^{\mathcal{M}}$ ...φ is evaluated under $\mathcal{M}$

- Satisfying Model: $\mathcal{M} \vDash \varphi$
  - $\mathcal{M}$ satisfies φ, or
  - φ evaluates to true under $\mathcal{M}$
  - Example
    - $\varphi = a \vee b$
    - $\mathcal{M}: \{a \to \top, b \to \bot\}$
    - $\mathcal{M} \vDash \varphi$ or $\varphi^{\mathcal{M}} = \top$

- Falsifying Model: $\mathcal{M} \nvDash \varphi$
  - $\mathcal{M}$ does not satisfies φ, or
  - φ evaluates to false under $\mathcal{M}$
  - Example
    - $\varphi = a \vee b$
    - $\mathcal{M}: \{a \to \bot, b \to \bot\}$
    - $\mathcal{M} \nvDash \varphi$ or $\varphi^{\mathcal{M}} = \bot$

Semantics – Inductive Definition

$\mathcal{M} \vDash \varphi$

φ evaluates to true under $\mathcal{M}$

$\mathcal{M} \nvDash \varphi$

φ evaluates to false under $\mathcal{M}$

Semantics – Inductive Definition

$\mathcal{M} \vDash \varphi$
φ evaluates to true under $\mathcal{M}$
$\mathcal{M} \nvDash \varphi$
φ evaluates to false under $\mathcal{M}$

- Base cases for assignment of truth values
  - $\mathcal{M} \vDash \top$
  - $\mathcal{M} \nvDash \bot$
  - $\mathcal{M} \vDash p$ iff $\mathcal{M}[p] = \top$ — p has the value $\top$ if $\mathcal{M}$ assigns the value $\top$ to p
  - $\mathcal{M} \nvDash p$ iff $\mathcal{M}[p] = \bot$ — p has the value $\bot$ if $\mathcal{M}$ assigns the value $\bot$ to p

Semantics – Inductive Definition

$\mathcal{M} \vDash \varphi$

φ evaluates to true under $\mathcal{M}$

$\mathcal{M} \nvDash \varphi$

φ evaluates to false under $\mathcal{M}$

- Inductive step
- Assume formulas φ and ψ have truth values
  - $\mathcal{M} \vDash \neg\varphi$ iff $\mathcal{M} \nvDash \varphi$

Semantics – Inductive Definition

$\mathcal{M} \vDash \varphi$
φ evaluates to true under $\mathcal{M}$
$\mathcal{M} \nvDash \varphi$
φ evaluates to false under $\mathcal{M}$

- Inductive step
- Assume formulas φ and ψ have truth values
  - $\mathcal{M} \vDash \neg\varphi$ iff $\mathcal{M} \nvDash \varphi$
  - $\mathcal{M} \vDash \varphi \wedge \psi$ iff $\mathcal{M} \vDash \varphi$ and $\mathcal{M} \vDash \psi$

Semantics – Inductive Definition

$\mathcal{M} \vDash \varphi$

φ evaluates to true under $\mathcal{M}$

$\mathcal{M} \nvDash \varphi$

φ evaluates to false under $\mathcal{M}$

- Inductive step
- Assume formulas φ and ψ have truth values
  - $\mathcal{M} \vDash \neg\varphi$ iff $\mathcal{M} \nvDash \varphi$
  - $\mathcal{M} \vDash \varphi \land \psi$ iff $\mathcal{M} \vDash \varphi$ and $\mathcal{M} \vDash \psi$
  - $\mathcal{M} \vDash \varphi \lor \psi$ iff $\mathcal{M} \vDash \varphi$ or $\mathcal{M} \vDash \psi$
  -

Semantics – Inductive Definition

$\mathcal{M} \vDash \varphi$

φ evaluates to true under $\mathcal{M}$

$\mathcal{M} \nvDash \varphi$

φ evaluates to false under $\mathcal{M}$

- Inductive step
- Assume formulas φ and ψ have truth values
  - $\mathcal{M} \vDash \neg\varphi$ iff $\mathcal{M} \nvDash \varphi$
  - $\mathcal{M} \vDash \varphi \land \psi$ iff $\mathcal{M} \vDash \varphi$ and $\mathcal{M} \vDash \psi$
  - $\mathcal{M} \vDash \varphi \lor \psi$ iff $\mathcal{M} \vDash \varphi$ or $\mathcal{M} \vDash \psi$
  - $\mathcal{M} \vDash \varphi \rightarrow \psi$ iff if $\mathcal{M} \vDash \varphi$ then $\mathcal{M} \vDash \psi$
  - $\mathcal{M} \vDash \varphi \leftrightarrow \psi$ iff $\mathcal{M} \vDash \varphi$ and $\mathcal{M} \vDash \psi$, or $\mathcal{M} \nvDash \varphi$ and $\mathcal{M} \nvDash \psi$

Semantics – Definition via Truth Tables

$\mathcal{M} \vDash \varphi$
φ evaluates to true under $\mathcal{M}$
$\mathcal{M} \nvDash \varphi$
φ evaluates to false under $\mathcal{M}$

- Base cases as before
  - Prop variables get truth values from $\mathcal{M}$

- **Truth tables** summarize truth assignments for compounded formulas

φ	ψ	$\varphi \wedge \psi$
F	F	F
F	T	F
T	F	F
T	T	T

φ	ψ	$\varphi \vee \psi$
F	F	F
F	T	T
T	F	T
T	T	T

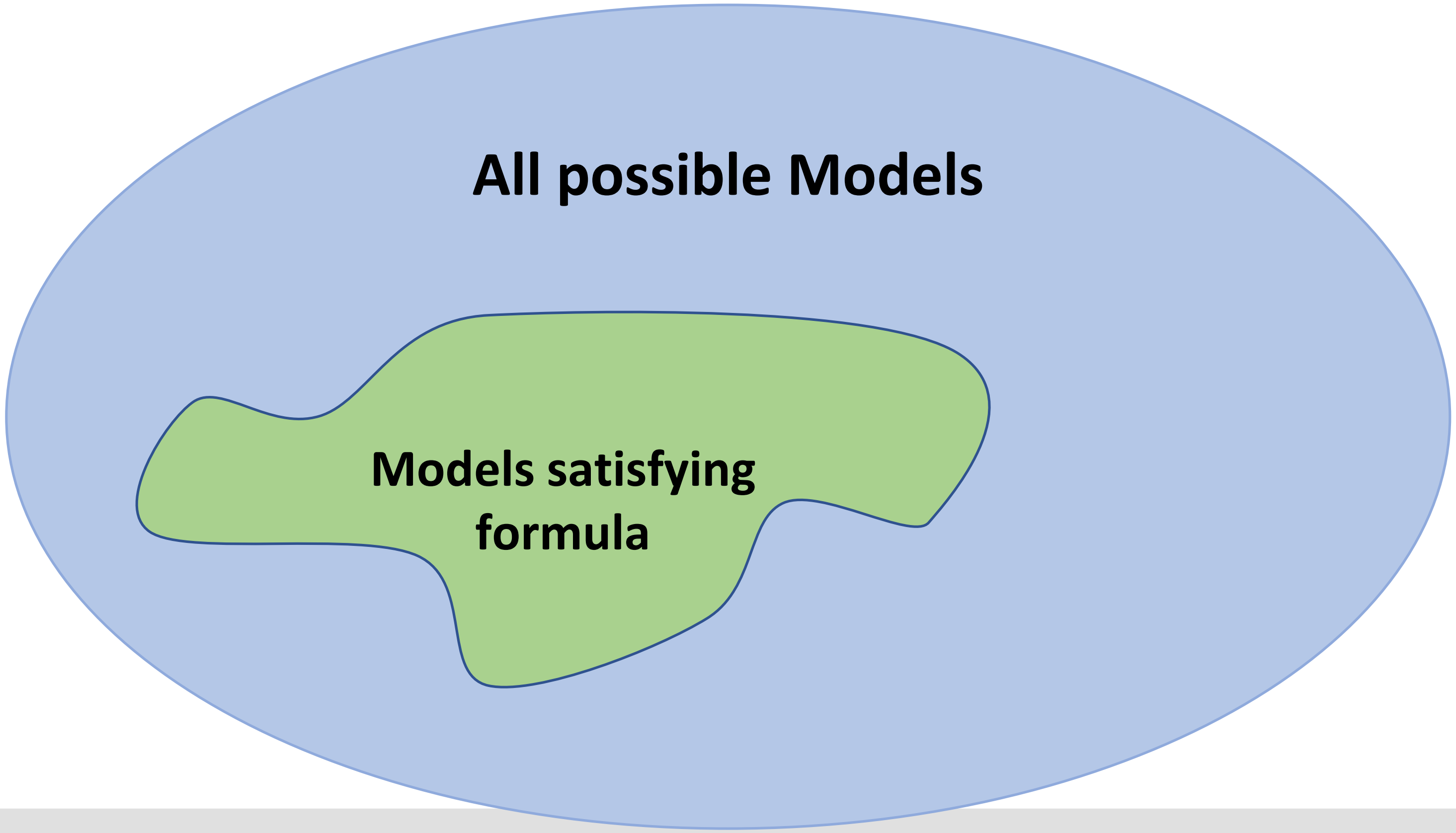
φ	$\neg\varphi$
F	T
T	F

φ	ψ	$\varphi \rightarrow \psi$
F	F	T
F	T	T
T	F	F
T	T	T

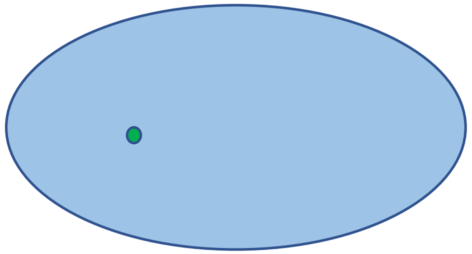
φ	ψ	$\varphi \leftrightarrow \psi$
F	F	T
F	T	F
T	F	F
T	T	T

Satisfiability (SAT)

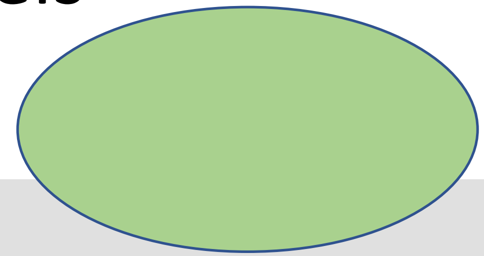
- At least one model satisfies the formula.
  - φ is SAT iff there exists a model $\mathcal{M}$ such that $\mathcal{M} \vDash \varphi$



- One Model

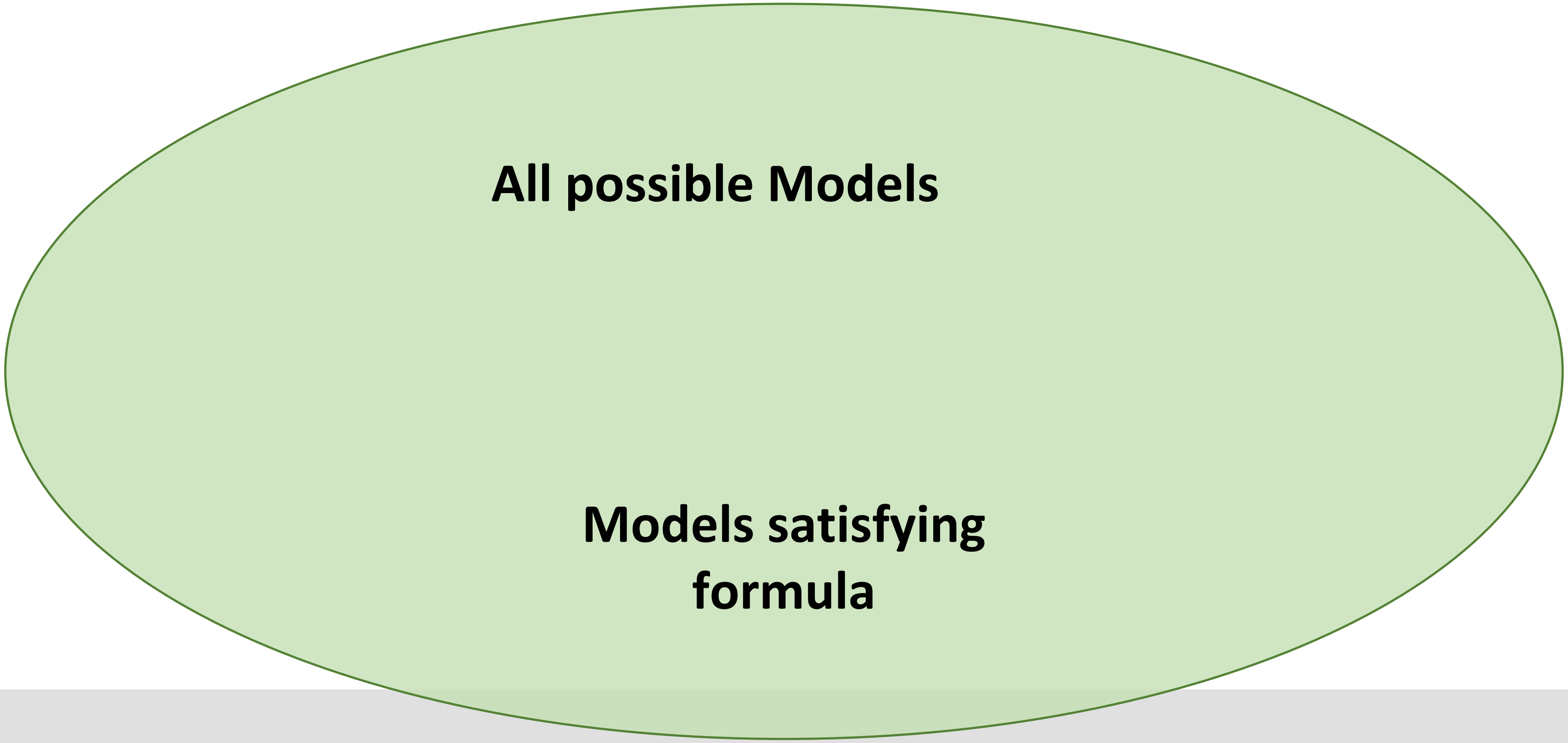


- All Models



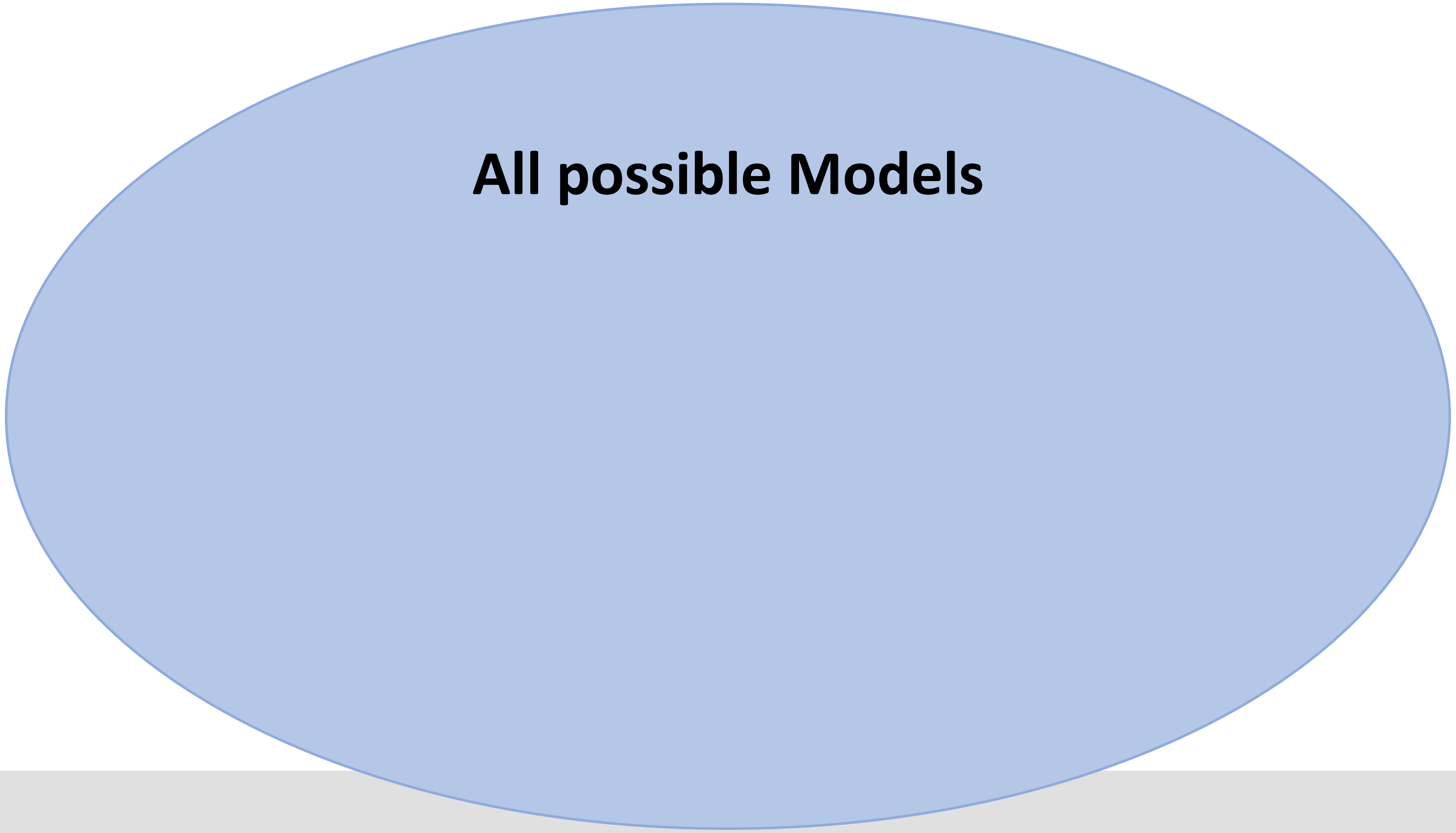
Validity - Tautology

- **All** models satisfy the formula
  - φ is valid iff for all models $\mathcal{M}$, $\mathcal{M} \vDash \varphi$



Unsatisfiability - UNSAT

- A formula that is not satisfiable
  - φ is UNSAT iff for all models $\mathcal{M}$, $\mathcal{M} \nvDash \varphi$



SAT and Validity are **dual** concepts

φ is valid iff $\neg\varphi$ is UNSAT

Truth Tables

- Used to check for *validity* or *satisfiability*

- Row for each Model $\mathcal{M}_i$
  - $\#Rows \ = \ 2^{\#Vars}$

Huge disadvantage of truth tables

- Entry E_{ij}
  - True, if $\mathcal{M}_i \ \vDash \ \varphi_j$
  - False, if $\mathcal{M}_i \ \nvDash \ \varphi_j$

Truth Tables

- Used to check for *validity* or *satisfiability*

- Row for each Model $\mathcal{M}_i$
  - $\#Rows = 2^{\#Vars}$ ⇦ Huge disadvantage of truth tables

- Entry E_{ij}
  - True, if $\mathcal{M}_i \vDash \varphi_j$
  - False, if $\mathcal{M}_i \nvDash \varphi_j$

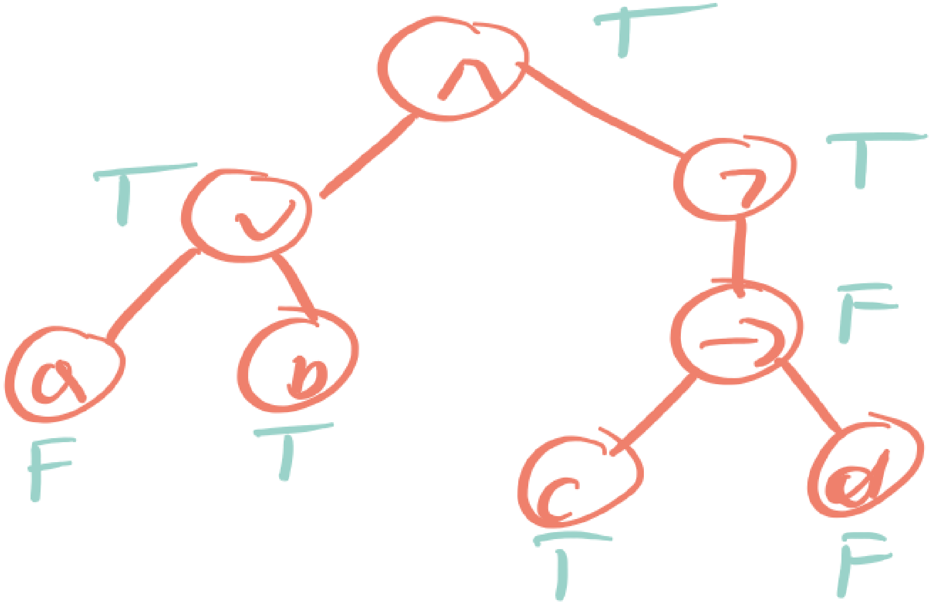
- Satisfiability: Check if at least one row with **True**?
- Validity check if all rows **True**?

Outline

- Declarative Sentences
- Syntax
  - Symbols & Grammar
  - Parse Tree
- Semantics
  - Meaning
  - Models
  - Truth Tables
  - Validity, Satisfiability
- **Examples**

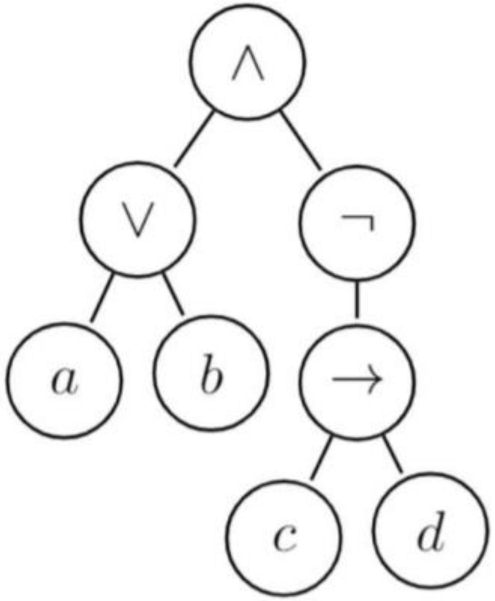
Draw parse tree for φ and evaluate φ under $\mathcal{M}_1$

- $\varphi = (a \lor b) \land (\neg(c \to d))$
- $\mathcal{M}_1 : a = F,\ b = T,\ c = T,\ d = F$



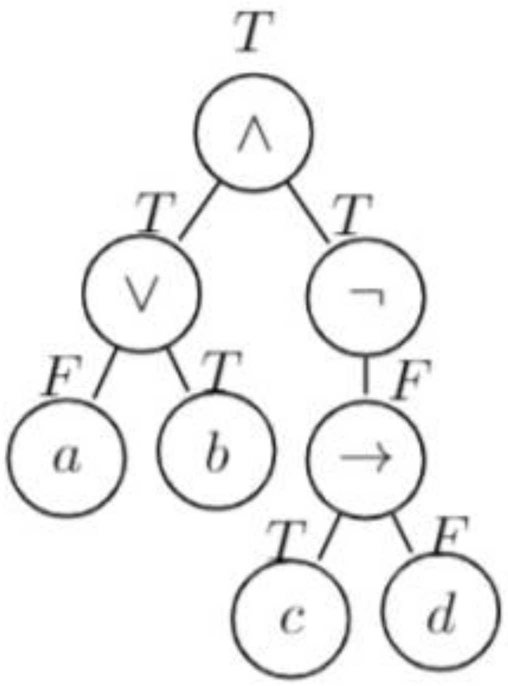
Draw parse tree for φ and evaluate φ under $\mathcal{M}_1$

- $\varphi = (a \vee b) \wedge (\neg(c \rightarrow d))$
- $\mathcal{M}_1 : a = F,\ b = T,\ c = T,\ d = F$



Draw parse tree for φ and evaluate φ under $\mathcal{M}_1$

- $\varphi = (a \vee b) \wedge (\neg(c \to d))$
- $\mathcal{M}_1 : a = F, \; b = T, \; c = T, \; d = F$

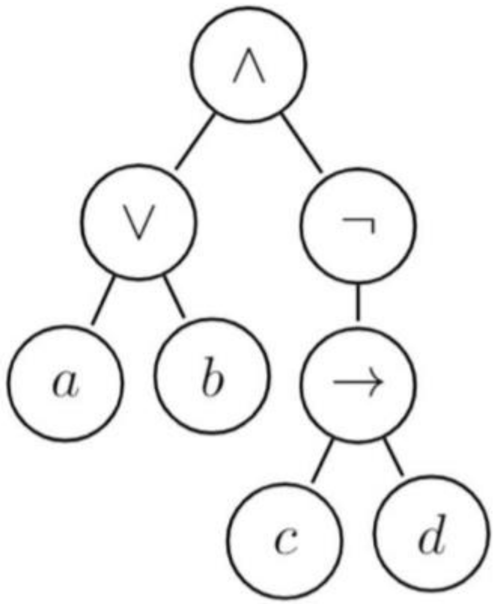


$$\varphi^{\mathcal{M}_1} = T$$

$$M_1 \vDash \varphi$$

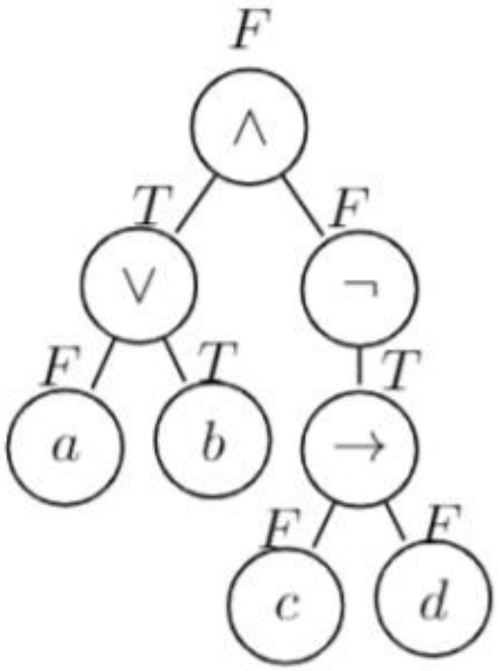
Draw parse tree for φ and evaluate φ under $\mathcal{M}_2$

- $\varphi = (a \vee b) \wedge (\neg (c \rightarrow d))$
- $\mathcal{M}_2 : a = F, \; b = T, \; c = F, \; d = F$



Draw parse tree for φ and evaluate φ under $\mathcal{M}_2$

- $\varphi = (a \vee b) \wedge (\neg(c \to d))$
- $\mathcal{M}_2 : a = F, \ b = T, \ c = F, \ d = F$



$\varphi^{\mathcal{M}_2} = F$

$M_2 \nvDash \varphi$

Usage of Truth Table: $\varphi = a \wedge \neg(b \rightarrow c)$

- Draw a truth table
- Answer the following questions:
  a. Is φ Satisfiability?
  b. Is φ valid?

Usage of Truth Table: $\varphi = a \wedge \neg(b \rightarrow c)$

- Draw a truth table
- Answer the following questions:
  a. Is φ Satisfiability?
  b. Is φ valid?

a	b	c	$b \rightarrow c$	$\neg(b \rightarrow c)$	φ
F	F	F	T	F	F
F	F	T	T	F	F
F	T	F	F	T	F
F	T	T	T	F	F
T	F	F	T	F	F
T	F	T	T	F	F
T	T	F	F	T	T
T	T	T	T	F	F

Usage of Truth Table: $\varphi = a \wedge \neg(b \rightarrow c)$

- Draw a truth table
- Answer the following questions:
  a. Is φ Satisfiability?
  b. Is φ valid?

a	b	c	$b \rightarrow c$	$\neg(b \rightarrow c)$	φ
F	F	F	T	F	F
F	F	T	T	F	F
F	T	F	F	T	F
F	T	T	T	F	F
T	F	F	T	F	F
T	F	T	T	F	F
T	T	F	F	T	T
T	T	T	T	F	F

Solution

a. Yes

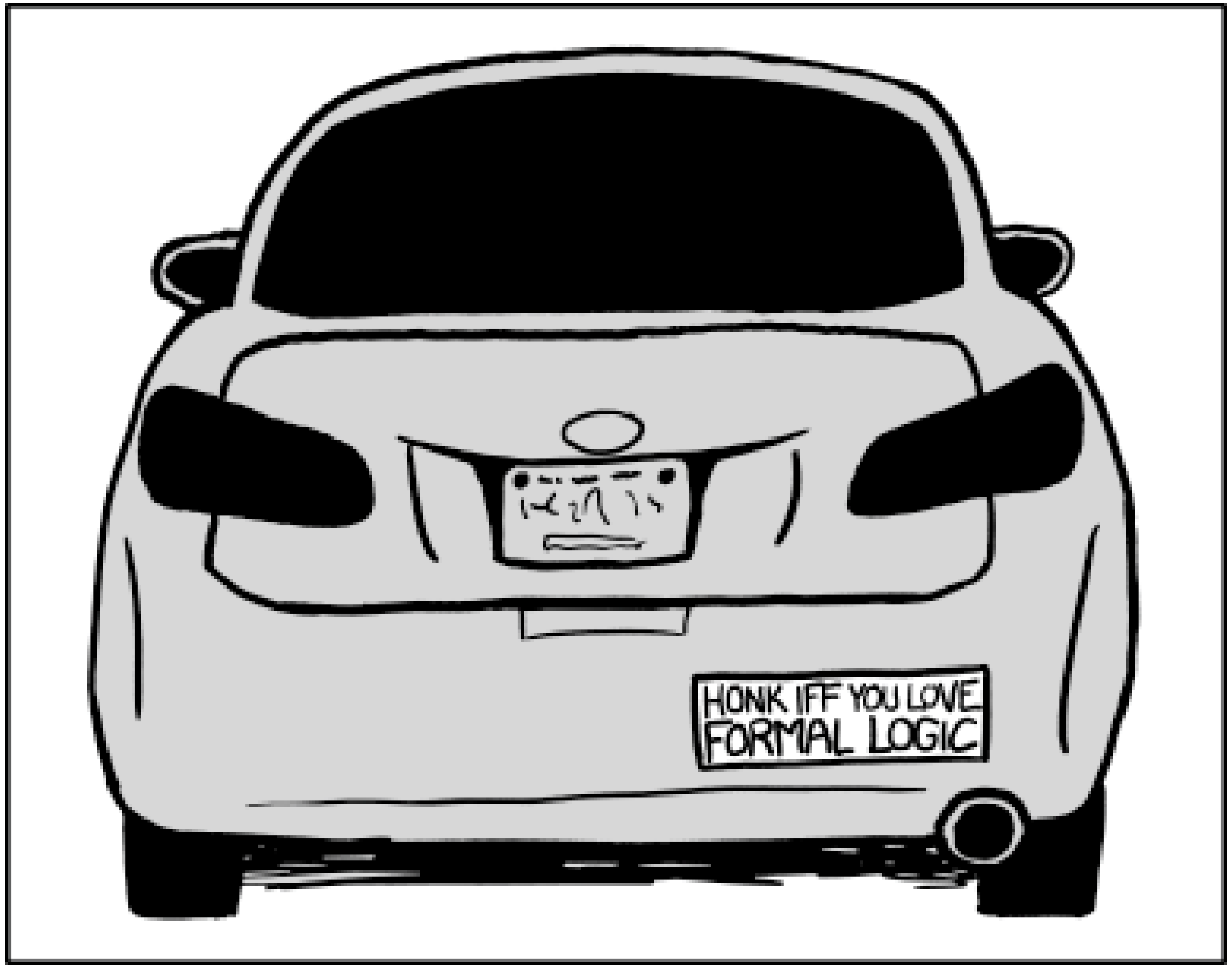
b. No

Learning Targets

- Syntax
  - Explain syntax of prop. formulas
  - Draw parse tree of prop. formulas

- Semantics
  - Model sentences as prop. formula
  - Explain semantics of prop. Formulas
  - Explain what models are
  - Construct and use truth tables
  - Explain and decide validity and satisfiability
    - Using truth tables

Thank You



https://xkcd.com/1033/