Program Reasoning

4. Propositional Logic

Kihong Heo

KAIST

Logic

- What is logic? A tool for reasoning about truths
- Why logic for computer scientists? Reasoning about computation
- For example,
  - “Does this program accept an array of integers and produce a sorted array?”
  - “Does this program access an unallocated memory?”
  - “Does this function always halt?”
- This course: propositional logic (PL, 명제논리) and first-order logic (FOL, 일차 논리)

Syntax문법구조

- Atom: basic elements
  - Truth symbols: ⊤ (“true”) and ⊥ (“false”)
  - Propositional variables: $P, Q, R, \ldots$
- Literal: an atom α or its negation $\neg\alpha$
- Formula: a literal or the application of a logical connective to formulae

$$
\begin{array}{rcl}
F & \to & \bot \\
 & | & \top \\
 & | & P, Q, R, \ldots \\
 & | & \neg F \\
 & | & F_1 \wedge F_2 \\
 & | & F_1 \vee F_2 \\
 & | & F_1 \to F_2 \\
 & | & F_1 \leftrightarrow F_2
\end{array}
$$

Semantics 의미구조

- Give meaning to formulae
  - In propositional logic, the truth values
- The semantics of a formula is defined with an interpretation I
  - An interpretation assigns to every propositional variable exactly one truth value
- For example, $F : P \land Q \to P \lor \neg Q$ and $I : \{P \mapsto \top, Q \mapsto \bot\}$

Inductive Definition of PL

- Notation:
  - $I \vDash F$ if F evaluates to true under I
  - $I \nvDash F$ if F evaluates to false under I

$$
\begin{array}{ll}
I \models \top & \\
I \not\models \bot & \\
I \models P & \text{iff } I[P] = \mathsf{true} \\
I \not\models P & \text{iff } I[P] = \mathsf{false} \\
I \models \neg F & \text{iff } I \not\models F \\
I \models F_1 \wedge F_2 & \text{iff } I \models F_1 \text{ and } I \models F_2 \\
I \models F_1 \vee F_2 & \text{iff } I \models F_1 \text{ or } I \models F_2 \\
I \models F_1 \rightarrow F_2 & \text{iff, if } I \models F_1 \text{ then } I \models F_2 \\
I \models F_1 \leftrightarrow F_2 & \text{iff, if } I \models F_1 \text{ and } I \models F_2, \text{ or if } I \not\models F_1 \text{ and } I \not\models F_2
\end{array}
$$

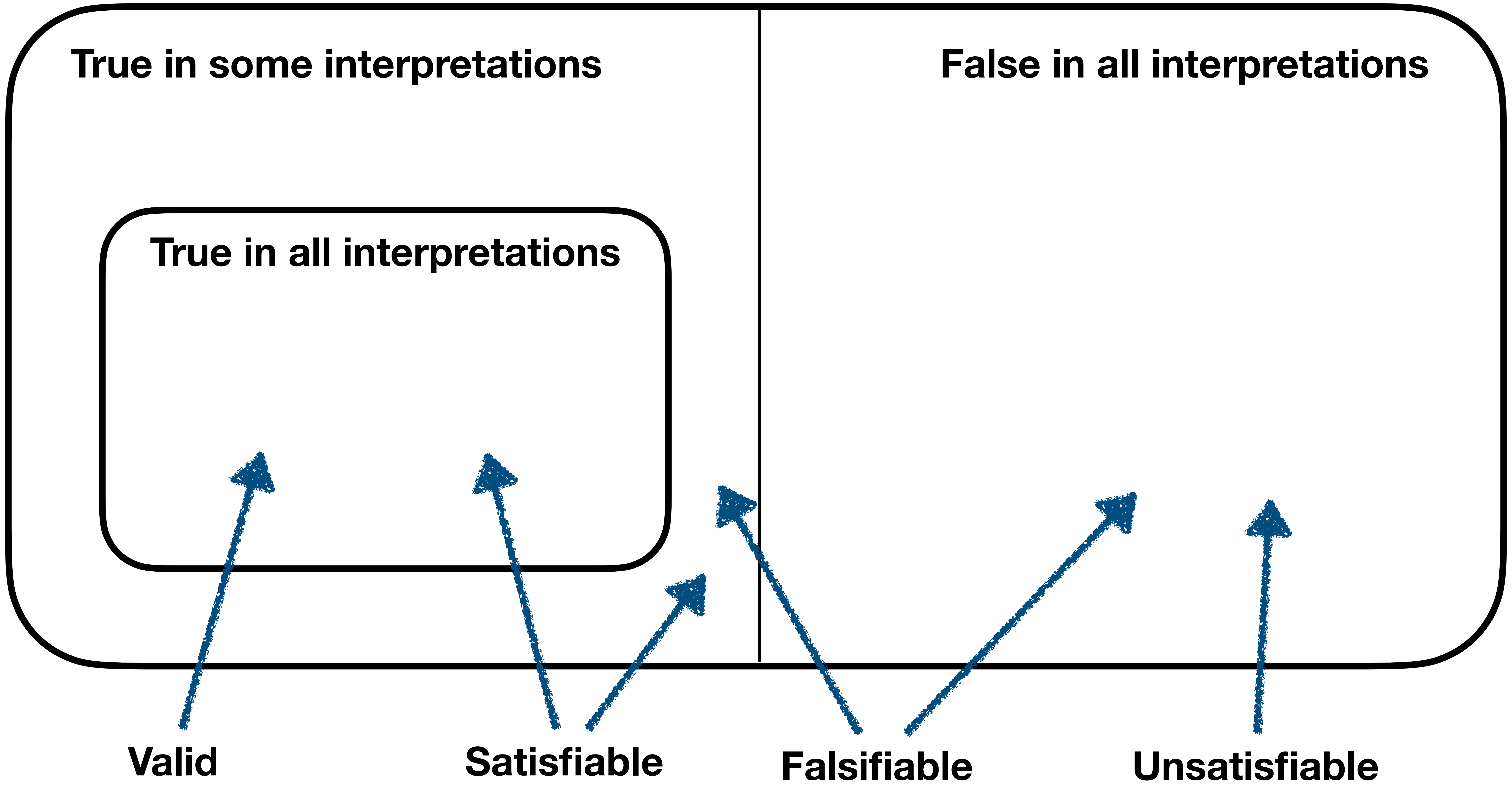
Example

- $F : P \land Q \rightarrow P \lor \neg Q$ and $I : \{P \mapsto \top , Q \mapsto \bot \}$

Satisfiability만족가능성 and Validity타당성

- Two important tasks in logic (why? when?)
- A formula F is satisfiable iff there exists an interpretation I such that $I \models F$
- A formula F is valid iff for all interpretations I, $I \models F$
- Satisfiability and validity are dual: F is valid iff $\neg F$ is unsatisfiable
- We are free to focus on either one; the other will follow

Valid, Satisfiable, Falsifiable and Unsatisfiable



Determining Validity and Satisfiability (1)

- Truth table method
  - For example, $F : P \land Q \to P \lor \neg Q$
- Impractical: 2^n interpretations
- Impossible: for any other logic where the domain is not finite (e.g., first-order logic)

P	Q	P ⋀ Q	¬Q	P ⋁ ¬Q	F
0	0	0	1	1	1
0	1	0	0	0	1
1	0	0	1	1	1
1	1	1	0	1	1

Determining Validity and Satisfiability (2)

- Semantic argument method (proof by contradiction)
  - Assume F is invalid: $I \not\models F$
  - Apply proof rules to derive new facts
  - Derive a contradiction in every branch of the proof
  - Then, F is valid

Proof Rules (1)

- According to the semantics of negation,

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

- According to the semantics of conjunction,

$$\frac{I \models F \land G}{I \models F, I \models G} \qquad \frac{I \not\models F \land G}{I \not\models F \mid I \not\models G}$$

Proof Rules (2)

- According to the semantics of disjunction,

$$\frac{I \models F \vee G}{I \models F \quad | \quad I \models G} \qquad\qquad \frac{I \not\models F \vee G}{I \not\models F,\ I \not\models G}$$

- According to the semantics of implication,

$$\frac{I \models F \rightarrow G}{I \not\models F \quad | \quad I \models G} \qquad\qquad \frac{I \not\models F \rightarrow G}{I \models F,\ I \not\models G}$$

Proof Rules (3)

- According to the semantics of iff,

$$\frac{I \models F \leftrightarrow G}{I \models F \land G \;\mid\; I \models \neg F \land \neg G} \qquad \frac{I \not\models F \leftrightarrow G}{I \models F \land \neg G \;\mid\; I \models \neg F \land G}$$

- Contradiction

$$\frac{I \models F,\; I \not\models F}{I \models \bot}$$

Example

- Prove $F : P \land Q \rightarrow P \lor \neg Q$ is valid

Example

- Prove $F : (P \rightarrow Q) \land (Q \rightarrow R) \rightarrow (P \rightarrow R)$ is valid

Proof Tree

- Proof evolves as a tree rather than linearly
  - A branch of the tree is a sequence of lines descending from the root
  - A branch is closed if it contains a contradiction, otherwise open
  - A semantic argument is finished when no more proof rules are applicable
- Proof of the validity of F : if every branch is closed
  - Otherwise, each open branch describes a falsifying interpretation of F

Derived Rules

- The proof rules are theoretically sufficient
- However, derived proof rules can make proofs more concise (c.f., procedure, subroutine)
- Example: modus ponens

$$\frac{I \models F, \quad I \models F \rightarrow G}{I \models G}$$

Example

- Prove $F : (P \to Q) \land (Q \to R) \to (P \to R)$ is valid

Proof of Satisfiability

- Dual of the validity proof: F is satisfiable iff $\neg F$ invalid
- Truth-table or semantic argument methods
- Example: $\neg(P \vee Q \rightarrow P \wedge Q)$

P	Q	P ⋁ Q	P ⋀ Q	F	¬F
0	0	0	0	1	0
0	1	1	0	0	1
1	0	1	0	0	1
1	1	1	1	1	0

Example

- Prove $G : \neg(P \vee Q \to P \wedge Q)$ is satisfiable

  We prove that $P \vee Q \to P \wedge Q$ is invalid

Equivalence and Implication

- Important properties of pairs of formulae
- Two formulae F_1 and F_2 are equivalent iff $F_1 \leftrightarrow F_2$ is valid: $F_1 \iff F_2$
  - $F_1 \iff F_2$ is not a formula but a statement
- Formula F_1 implies formula F_2 iff $F_1 \rightarrow F_2$ is valid: $F_1 \implies F_2$
  - $F_1 \implies F_2$ is not a formula but a statement

Examples

- Prove $P \iff \neg\neg P$ (using the truth table method)

We prove that $P \leftrightarrow \neg\neg P$ is valid:

P	¬P	¬¬P	P ↔ ¬¬P
0	1	0	1
1	0	1	1

- Prove $P \to Q \iff \neg P \vee Q$ (using the truth table method)

We prove that $F : P \to Q \leftrightarrow \neg P \vee Q$ is valid:

P	Q	P → Q	¬P	¬P ∨ Q	F
0	0	1	1	1	1
0	1	1	1	1	1
1	0	0	0	0	1
1	1	1	0	1	1

Example

- Prove that $R \wedge (\neg R \vee P) \implies P$

  We prove that $R \wedge (\neg R \vee P) \rightarrow P$ is valid

Application: Hardware Verification

- The Pentium FDIV bug (1994): incorrect binary floating point division
  - Result: full recall ($475M)
- Intel started using formal verification after the issue
- Turing Award (2007): Edmund Clarke

Summary

- Propositional logic: the simplest form of logic
- Interpretation: decide the meaning of a formula (either true or false)
- Satisfiability: is there any interpretation that makes the formula be true?
- Validity: does the formula evaluate to be true for all interpretations?
- Duality of satisfiability and validity
    - E.g., “no input can trigger this bug” = “work well with all inputs”
- Equivalence and implication: properties of two formulae