# Propositional Calculus
# *- Hilbert system $\mathcal{H}$*

Moonzoo Kim
CS Dept. KAIST

moonzoo@cs.kaist.ac.kr

# Review

- Goal of logic
  - To check whether given a formula $\phi$ is valid
  - To prove a given formula $\phi$

***Sound & Complete***

$\vdash \phi$ $\equiv$ $\models \phi$

**Syntactic method (proof system $\mathcal{G}$, $\mathcal{H}$,etc)**

**Semantic method (truth table, etc)**

***Sound & Complete*** $\equiv$

**Semantic tableau**

$\equiv$ ***Sound & Complete***

# Review (cont.)

- Remember the following facts
  - Although we have many binary operators ($\{\vee, \wedge, \rightarrow, \leftarrow, \leftrightarrow, \downarrow, \uparrow, \oplus\}$), $\uparrow$ can replace all other binary operators through semantic equivalence. Similarly, $\{\rightarrow, \neg\}$ is an adequate set of binary operators.
  - $\not\models \phi$ does not necessarily mean $\models \neg\phi$
  - Deductive proof cannot disprove $\phi$ (i.e. claiming that there does not exist a proof for $\phi$) while semantic method can show both validity and satisfiability of $\phi$
  - Very few logics have decision procedure for validity check (i.e., truth table). Thus, we use deductive proof in spite of the above weakness.
  - A proof tree in $\mathcal{G}$ grows up while a proof tree in $\mathcal{H}$ shrinks down according to characteristics of its inference rules
    - Thus, a proof in $\mathcal{G}$ is easier than a proof in $\mathcal{H}$ in general

# Sound verification tools

- Suppose that
  - there is a target software S
  - there is a formal requirement R
- We can make a state machine (automata) of S, say $A_S$
  - A state of $A_S$ consists of all variables including a program counter.
- Any state machine can be encoded into a predicate logic formual $\phi_{a_s}$
  - We will see this encoding in the first order logic classes
- Program verification is simply to prove $\phi_{A_S} \models$ R
- For this purpose, we use a formal verification tool V so that $\phi_{A_S} \vdash_V$ R
  - We call V is sound whenever S has a bug, V always detects the bug
    - $\phi_{A_S} \not\models R \Rightarrow \phi_{A_S} \nvdash_V R$ (iff $\phi_{A_S} \vdash_V R \Rightarrow \phi_{A_S} \models R$ )
  - We call V is complete whenever V detects a bug, that bug is a real bug.
    - $\phi_{A_S} \nvdash_V R \rightarrow \phi_{A_S} \not\models R$ (iff $\phi_{A_S} \models R \Rightarrow \phi_{A_S} \vdash_V R$)
  - In reality, most formal verification tools are just sound, not complete (I.e., formal verification tools may raise false alarms). However, for debugging purpose, soundness is great.

# The Hilbert system $\mathcal{H}$

- Def 3.9 $\mathcal{H}$ is a deductive system with three axiom schemes and one rule of inference.
  - For any formulas A,B,C, the following formulas are axioms (in fact axiom schemata):
    - Axiom1: $\vdash (A \rightarrow (B \rightarrow A))$
    - Axiom2: $\vdash (A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$
    - Axiom3: $\vdash (\neg B \rightarrow \neg A) \rightarrow (A \rightarrow B))$
  - The rule of inference is called modus ponens (MP). For any formulas A,B

$$\frac{\vdash A \qquad \vdash A \rightarrow B}{\vdash B}$$

- Note that axioms used in a proof in $\mathcal{H}$ are usually very long because the MP rule reduces a length of formula (see Thm 3.10)
  - at least one premise ($\vdash A \rightarrow B$) is longer than conclusion (B)

# $\mathcal{G}$ v.s. $\mathcal{H}$

- $\mathcal{G}$ is a deductive system for a set of formulas while $\mathcal{H}$ is a deductive system for a single formula
- $\mathcal{G}$ has one form of axiom and many rules (for 8 $\alpha$-rules and 7 $\beta$-rules) while $\mathcal{H}$ has several axioms (in fact axiom schemes) but only one rule

# Derived rules

- Def. 3.12 Let U be a set of formulas and A a formula. The notation $U \vdash A$ means that the formulas in U are assumptions in the proof of A. If $A_i \in U$, a proof of $U \vdash A$ may include an element of the form $U \vdash A_i$
- Collorary. $U \cup \{A\} \vdash A$
- Rule 3.13 Deduction rule

$$\frac{U \cup \{A\} \vdash B}{U \vdash A \rightarrow B}$$

  - Note that deduction rule increase the size of a formula, thus making a proof easier compared to MP rule

# Soundness of deduction rule

$$\frac{U \cup \{A\} \vdash B}{U \vdash A \to B}$$ premise / conclusion

- Thm 3.14 The deduction rule is a sound derived rule
  - By induction on the length n of the proof $U \cup \{A\} \vdash B$
    - For n=1, B is proved in one step, so B must be either an element of $U \cup \{A\}$ or an axiom of $\mathcal{H}$
      - If B is A, then $\vdash A \to B$ by Thm 3.10 ($\vdash A \to A$), so certainly $U \vdash A \to B$.
      - Otherwise (i.e., $B \in U$ or B is an axion), the following is a proof of $U \vdash A \to B$

axiom 1

$$\frac{U \vdash B \qquad U \vdash B \to (A \to B)}{U \vdash A \to B} \text{ MP}$$

# Soundness of deduction rule

$$\frac{U \cup \{A\} \vdash B}{U \vdash A \to B}$$ premise / conclusion

- For n>1, the last step in the proof of U∪{A}⊢B is an inference of B using MP.
  - there is a formula C such that formula i in the proof is $U \cup \{A\} \vdash C$ and formula j is $U \cup \{A\} \vdash C \to B$, for i, j < n. By the inductive hypothesis $U \vdash A \to C$ and $U \vdash A \to (C \to B)$. A proof of $U \vdash A \to B$ is given by

inductive hypothesis ⋮     axiom 2     ⋮ inductive hypothesis

$$\dfrac{\dfrac{U \vdash A\to(C\to B) \qquad U \vdash (A\to(C\to B))\to((A\to C)\to(A\to B))}{U \vdash (A\to C)\to (A\to B)}\ \text{MP} \qquad U \vdash A \to C}{U \vdash A \to B}\ \text{MP}$$

# Theorems and derived rules in $\mathcal{H}$

- Note that any theorem of the form U ⊢ A → B justifies a derived rule of the form $\dfrac{U \vdash A}{U \vdash B}$ simply by using MP on A and A → B

- Rule 3.15 Contrapositive rule $\dfrac{U \vdash \neg B \rightarrow \neg A}{U \vdash A \rightarrow B}$
  - by Axiom 3 ⊢ (¬B→¬A) → (A→B))

- Rule 3.17 Transitivity rule $\dfrac{U \vdash A \rightarrow B \qquad U \vdash B \rightarrow C}{U \vdash A \rightarrow C}$
  - by Thm 3.16 ⊢ (A→B)→[(B→C)→(A→C)]

- Rule 3.19 Exchange of antecedent rule $\dfrac{U \vdash A \rightarrow (B \rightarrow C)}{U \vdash B \rightarrow (A \rightarrow C)}$
  - by Thm 3.18 ⊢ [(A→ (B→C)] → [(B→(A→C)]

# Theorems and derived rules in $\mathcal{H}$

- Rule 3.23 Double negation rule $\dfrac{U \vdash \neg\neg A}{U \vdash A}$
  - by Thm 3.22 $U \vdash \neg\neg A \rightarrow A$
- Let true be an abbreviation for $p \rightarrow p$ and false be an abbreviation for $\neg(p \rightarrow p)$
- Rule 3.27 Reductio ad absurdum (RAA) rule $\dfrac{U \vdash \neg A \rightarrow false}{U \vdash A}$
- Thm 3.28 $\vdash (A \rightarrow \neg A) \rightarrow \neg A$
- Thm 3.31 Weakening
  - $\vdash A \rightarrow A \vee B$
  - $\vdash B \rightarrow A \vee B$
  - $\vdash (A \rightarrow B) \rightarrow ((C \vee A) \rightarrow (C \vee B))$