# Propositional Calculus - Hilbert system H

Moonzoo Kim CS Dept. KAIST

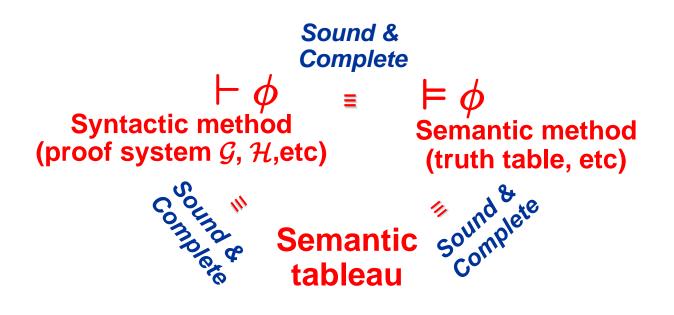
moonzoo@cs.kaist.ac.kr



#### Review

#### Goal of logic

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





## **Review (cont.)**

#### Remember the following facts

- Although we have many binary operators ({∨,∧,→,←,↔, ↓, ↑,⊕}), ↑ can replace all other binary operators through semantic equivalence. Similarly, {→, ¬} is an adequate set of binary operators.
- $\nvDash \phi$  does not necessarily mean  $\vDash \neg \phi$
- Deductive proof cannot disprove \u03c6 (i.e. claiming that there does not exist a proof for \u03c6) while semantic method can show both validity and satisfiability of \u03c6
- 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 G grows up while a proof tree in 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

Suppose that

# Sound verification tools

- there is a target software S
  there is a formal requirement R
- We can make a state machine (automata) of S, say A<sub>S</sub>
  - A state of A<sub>S</sub> consists of all variables including a program counter.
- Any state machine can be encoded into a predicate logic formual  $\phi_{a_{\rm s}}$ 
  - We will see this encoding in the first order logic classes
- Program verification is simply to prove  $\phi_{A_S} \vDash \mathsf{R}$
- For this purpose, we use a formal verification tool V so that  $\phi_{A_S} \vdash_{\sf V} \sf R$ 
  - We call V is sound whenever S has a bug, V always detects the bug ■  $\phi_{A_S} \nvDash R \Rightarrow \phi_{A_S} \nvDash_V R$  (iff  $\phi_{A_S} \vdash_V R \Rightarrow \phi_{A_S} \vDash R$ )
  - We call V is complete whenever V detects a bug, that bug is a real bug. ■  $\phi_{A_S} \nvDash_V \mathsf{R} \to \phi_{A_S} \nvDash \mathsf{R}$  (iff  $\phi_{A_S} \vDash \mathsf{R} \Rightarrow \phi_{A_S} \vdash_V \mathsf{R}$ )

 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.

CS402 Fall 2007

## The Hilbert system ${\cal H}$

- 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))
    - $\quad \ \ \, \text{Axiom2:} \ \ \vdash (\mathsf{A} \rightarrow (\mathsf{B} \rightarrow \mathsf{C})) \rightarrow ((\mathsf{A} \rightarrow \mathsf{B}) \rightarrow (\mathsf{A} \rightarrow \mathsf{C})) \\$
    - 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

$$\begin{array}{c|c} \vdash A & \vdash A \rightarrow B \\ \hline & \vdash B \end{array}$$

- Note that axioms used in a proof in *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. $\mathcal{H}$

- G is a deductive system for a set of formulas while
   H is a deductive system for a single formula
- 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 ⊢ A means that the formulas in U are assumptions in the proof of A. If A<sub>i</sub> ∈ U, a proof of U ⊢ A may include an element of the form U ⊢ A<sub>i</sub>
- Collorary.  $U \cup \{A\} \vdash A$
- Rule 3.13 Deduction rule

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

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



#### Soundness of deduction rule

 $U \vdash A \rightarrow B$  conclusion

 $U \cup \{A\} \vdash B$  premise

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  $\vdash A \rightarrow B$  by Thm 3.10 ( $\vdash A \rightarrow A$ ), so certainly U  $\vdash A \rightarrow B$ .
    - Otherwise (i.e., B∈U or B is an axion), the following is a proof of U ⊢ A → B

axiom 1  

$$U \vdash B$$
  $U \vdash B \rightarrow (A \rightarrow B)$   
 $U \vdash A \rightarrow B$  MP

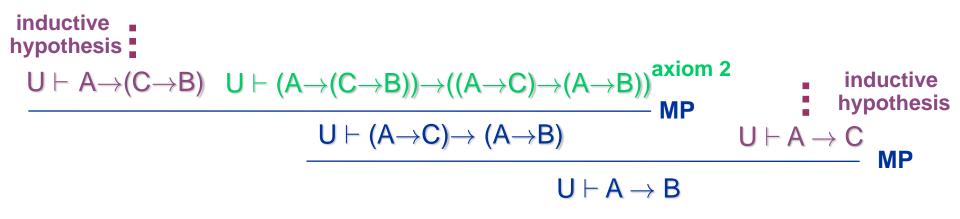


### Soundness of deduction rule

 $UDash A{ o}B$  conclusion

 $U \cup \{A\} \vdash B$  premise

- 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 ∪ {A} ⊢ C and formula j is U ∪ {A} ⊢ C → B, for i, j < n. By the inductive hypothesis U ⊢ A → C and U ⊢ A → (C → B). A proof of U ⊢ A → B is given by





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

- Note that any theorem of the form  $U \vdash A \rightarrow B$  justifies a derived rule of the form  $\underbrace{U \vdash A}_{U \vdash B}$  simply by using MP on
- Rule 3.15 Contrapositive rule
   by Axiom 3 ⊢ (¬B→¬A) → (A→B))

$$\frac{U \vdash \neg B \to \neg A}{U \vdash A \to B}$$

- Rule 3.17 Transitivity rule
  by Thm 3.16  $\vdash$  (A $\rightarrow$ B) $\rightarrow$ [(B $\rightarrow$ C) $\rightarrow$ (A $\rightarrow$ C)]  $\frac{U \vdash A \rightarrow B \qquad U \vdash B \rightarrow C}{U \vdash A \rightarrow C}$
- Rule 3.19 Exchange of antecedent rule  $U \vdash A \rightarrow (B \rightarrow C)$ by Thm 3.18  $\vdash$  [(A  $\rightarrow$  (B  $\rightarrow$ C)]  $\rightarrow$  [(B  $\rightarrow$ (A  $\rightarrow$ C)]  $U \vdash B \rightarrow (A \rightarrow C)$ ]

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

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



 $II \vdash A$