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

Moonzoo Kim
CS Division of EECS Dept.
KAIST

moonzoo@cs.kaist.ac.kr
http://pswlab.kaist.ac.kr/courses/cs402-07

Review

- Goal of logic
  - To check whether given a formula ϕ is valid
  - To prove a given formula ϕ

Sound & Complete

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

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

Semantic method
(truth table, etc)

Sound & Complete $\equiv$

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

Semantic tableau

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 ϕ (i.e. claiming that there does not exist a proof for ϕ) while semantic method can show both validity and satisfiability of ϕ
  - 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 ϕ_{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 α-rules and 7 β-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 $U \vdash A_i$
- 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

$$\frac{U \cup \{A\} \vdash B}{U \vdash A \rightarrow B} \quad \begin{matrix}\textbf{premise} \\ \textbf{conclusion}\end{matrix}$$

Soundness of deduction rule

- 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., $B \in U$ or B is an axion), the following is a proof of $U \vdash A \rightarrow B$

$$\frac{U \vdash B \qquad U \vdash B \rightarrow (A \rightarrow B)}{U \vdash A \rightarrow B}\ \textbf{MP}$$

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

Soundness of deduction rule

- For n>1, the last step in the proof of $U \cup \{A\} \vdash B$ is either
  - a one-step inference of B
    - the result holds by the proof for n =1
  - 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

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

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

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

- Rule 3.15 Contrapositive rule $\dfrac{U \vdash \neg B \rightarrow \neg A}{U \vdash A \rightarrow B}$
  - by Axiom 3 $\vdash$ ($\neg$B$\rightarrow$$\neg$A) $\rightarrow$ (A$\rightarrow$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 $\vdash$ (A$\rightarrow$B)$\rightarrow$[(B$\rightarrow$C)$\rightarrow$(A$\rightarrow$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 $\vdash$ [(A$\rightarrow$ (B$\rightarrow$C)] $\rightarrow$ [(B$\rightarrow$(A$\rightarrow$C)]