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

Moonzoo Kim
CS Dept. KAIST

moonzoo@cs.kaist.ac.kr
Goal of logic

- To check whether given a formula \( \phi \) is valid
- To prove a given formula \( \phi \)
Remember the following facts

- Although we have many binary operators ($\{\lor, \land, \rightarrow, \leftrightarrow, \downarrow, \uparrow, \oplus\}$), $\uparrow$ can replace all other binary operators through semantic equivalence. Similarly, $\{\rightarrow, \neg\}$ is an adequate set of binary operators.

- $\not\phi$ 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
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

- We call V is sound whenever S has a bug, V always detects the bug
  \( \phi_{A_S} \not\models R \Rightarrow \phi_{A_S} \not\models_V R \) (iff \( \phi_{A_S} \models V R \Rightarrow \phi_{A_S} \models R \) )

- We call V is complete whenever V detects a bug, that bug is a real bug.
  \( \phi_{A_S} \not\models V R \Rightarrow \phi_{A_S} \not\models R \) (iff \( \phi_{A_S} \models R \Rightarrow \phi_{A_S} \models V R \) )

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: $\vdash (A \to (B \to A))$
  - Axiom2: $\vdash (A \to (B \to C)) \to ((A \to B) \to (A \to C))$
  - Axiom3: $\vdash (\neg B \to \neg A) \to (A \to B)$
- The rule of inference is called modus ponens (MP). For any formulas $A, B$

\[
\begin{align*}
\vdash A & \quad \vdash A \to B \\
\hline 
\vdash B 
\end{align*}
\]

- 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 ($\vdash A \to B$) is longer than conclusion ($B$)
\( G \) v.s. \( H \)

- \( G \) 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 \( H \) 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 $U \vdash A_i$

- **Corollary.** $U \cup \{A\} \vdash A$

- **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
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 \( U \vdash A \rightarrow B \)

\[
\begin{align*}
U \vdash B & \quad U \vdash B \rightarrow (A \rightarrow B) \\
& \quad \text{axiom 1} \\
\hline
U \vdash A \rightarrow B & \quad \text{MP}
\end{align*}
\]
For \( n > 1 \), the last step in the proof of \( U \cup \{A\} \vdash B \) is an inference of \( B \) using \( \text{MP} \).

- there is a formula \( C \) such that formula \( i \) in the proof is \( U \cup \{A\} \vdash C \) and formula \( j \) is \( U \cup \{A\} \vdash C \rightarrow B \), for \( i, j < n \). By the inductive hypothesis \( U \vdash A \rightarrow C \) and \( U \vdash A \rightarrow (C \rightarrow B) \). A proof of \( U \vdash A \rightarrow B \) is given by:

\[
\begin{align*}
U \vdash A \rightarrow (C \rightarrow B) &\quad U \vdash (A \rightarrow (C \rightarrow B)) \rightarrow ((A \rightarrow C) \rightarrow (A \rightarrow B)) \\
\text{axiom 2} &\quad \text{MP} \\
U \vdash (A \rightarrow C) \rightarrow (A \rightarrow B) &\quad \text{MP} \\
U \vdash A \rightarrow B &\quad \text{MP}
\end{align*}
\]
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 \( \frac{U \vdash A}{U \vdash B} \) simply by using MP on \( A \) and \( A \rightarrow B \)

- Rule 3.15 Contrapositive rule
  - by Axiom 3 \( \vdash (\neg B \rightarrow \neg A) \rightarrow (A \rightarrow B) \)

- Rule 3.17 Transitivity rule
  - by Thm 3.16 \( \vdash (A \rightarrow B) \rightarrow [(B \rightarrow C) \rightarrow (A \rightarrow C)] \)

- Rule 3.19 Exchange of antecedent rule
  - by Thm 3.18 \( \vdash [(A \rightarrow (B \rightarrow C)] \rightarrow [(B \rightarrow (A \rightarrow C)] \)
Theorems and derived rules in $\mathcal{H}$

- **Rule 3.23 Double negation rule**
  - 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**
  - $U \vdash \neg A \rightarrow \neg false$

- **Thm 3.28**: $U \vdash (A \rightarrow \neg A) \rightarrow \neg A$

- **Thm 3.31** **Weakening**
  - $U \vdash A \rightarrow A \lor B$
  - $U \vdash B \rightarrow A \lor B$
  - $U \vdash (A \rightarrow B) \rightarrow ((C \lor A) \rightarrow (C \lor B))$