$\varphi_1\ \varphi_2 \cdots \varphi_i\ \varphi_j \cdots \varphi_n = \varphi$

1.3.7 The Axiom System: Soundness

可靠

Theorem 1.3.21 (Soundness). *With Hilbert's axiom system, we have that $\Gamma \vdash \varphi$ implies $\Gamma \models \varphi$.*

Proof. By induction of the length of deductive sequence of $\Gamma \vdash \varphi$. □

$n=0)$ $\varphi = \varphi_0$

① $\varphi \in \Gamma$

$\sigma \Vdash \Gamma \Rightarrow \sigma \Vdash \varphi$ ✓

② $\varphi \in A_{123}$

t.p.: $\sigma \Vdash (\neg\varphi \to \neg\psi) \to (\psi \to \varphi)$

assume: $\sigma \Vdash \neg\varphi \to \neg\psi$

$- \sigma \Vdash \varphi$

$- \sigma \Vdash \neg\varphi,\ \sigma \Vdash \neg\psi$

Corollary 1.3.22. *If $\vdash \varphi$, then $\models \varphi$.*

1.3.8 The Axiom System: Completeness

Definition 1.3.23 (Consistency). *We say a formula set Γ is consistent, iff there is some φ such that $\Gamma \nvdash \varphi$.*

$(n>0)$ $\varphi_n \in \Gamma, A_{123}$ 同上

$$\frac{\varphi_j \quad \varphi_j \to \varphi_n}{\varphi_n}$$

$\Gamma \vdash \varphi_j \overset{IH}{\Longrightarrow} \Gamma \models \varphi_j$ (*)

$\Gamma \vdash \varphi_j \to \varphi_n \overset{IH}{\Longrightarrow} \Gamma \models \varphi_j \to \varphi_n$ (**)

Assume $\sigma \Vdash \Gamma$, $\overset{(*)(**)}{\Longrightarrow} \sigma \Vdash \varphi_j$, $\sigma \Vdash \varphi_j \to \varphi_n$ $\Rightarrow \sigma \Vdash \varphi_n = \varphi$

$\Rightarrow$ exists η: $\Gamma \nvdash \eta$

Assume: exists φ: $\Gamma \vdash \varphi$, $\Gamma \vdash \neg\varphi$, $\vdash \varphi \to (\neg\varphi \to \eta)$ $\overset{MP}{\Longrightarrow} \Gamma \vdash \eta$

Lemma 1.3.24. *Γ is consistent iff for each φ, either $\Gamma \nvdash \varphi$ or $\Gamma \nvdash \neg\varphi$.*

Proof. Exploit $\neg\varphi, \varphi \vdash \psi$. □

Lemma 1.3.25. *$\Gamma \cup \{\varphi\}$ is consistent iff $\Gamma \nvdash \neg\varphi$.*

Proof. $\Leftarrow$ Suppose that $\Gamma \nvdash \neg\varphi$ and $\Gamma \cup \{\varphi\}$ is inconsistent, then we have $\Gamma, \varphi \vdash \neg\varphi$

$\vdash (\varphi \to \neg\varphi) \to \neg\varphi$

$\to_I \overset{MP}{\Longrightarrow} \Gamma \vdash \neg\varphi$

$\neg\varphi$ hence $\Gamma \vdash \varphi \to \neg\varphi$. Recall that we have $\varphi \to \neg\varphi \vdash \neg\varphi$, and this implies $\Gamma \vdash \neg\varphi$, contradiction! The other direction is easy. $\square$

Lemma 1.3.26. *If the formula set Γ is inconsistent, then it has some finite inconsistent subset Δ.*

$\Gamma \vdash \varphi$. $\Gamma \vdash \neg\varphi$

$\varphi_0\ \varphi_1 \cdots \varphi_n$ $\quad \psi_0\ \psi_1 \cdots \psi_m$

$\| \quad\quad \|$

$\varphi \quad\quad \neg\varphi$

Δ

Theorem 1.3.27. *Γ is consistent iff Γ is satisfiable.*

$\Leftarrow$ Γ is sat. $\sigma \Vdash \Gamma$

Γ inconsist

Proof. The "if" direction is easy: suppose that $\sigma \Vdash \Gamma$ but $\Gamma \vdash \varphi$ and $\Gamma \vdash \neg\varphi$, then $\sigma \Vdash \varphi$ and $\sigma \Vdash \neg\varphi$, contradiction.

$\Rightarrow$ $\Gamma \vDash \varphi$, $\Gamma \vDash \neg\varphi$

For the "only if" direction, let us enumerate all propositional formulas as following (note the cardinality of all such formulas is $\aleph_0$):

$$\varphi_0, \varphi_1, \ldots, \varphi_n, \ldots$$

Let $\Gamma_0 = \Gamma$ and consistent

$$\Gamma_{i+1} = \begin{cases} \Gamma_i \cup \{\varphi_i\} & \text{if } \Gamma_i \nvdash \neg\varphi_i \\ \Gamma_i \cup \{\neg\varphi_i\} & \text{otherwise} \end{cases}$$

Lemma

$\Gamma_i \vdash \neg\varphi_i$

and finally let $\boxed{\Gamma^* = \lim_{i\to\infty} \Gamma_i}$.

The formula set Γ^* has the following properties:

(a) *Each Γ_{i+1} is consistent, and Γ^* is also consistent.*

Δ finite

$\Delta \subseteq \Gamma^*$

Δ incons

- Assume Γ_i is consistent. If $\Gamma_i \nvdash \neg\varphi_i$, by Lemma 1.3.25, Γ_{i+1} is consistent. Otherwise, $\Gamma_i \vdash \neg\varphi_i$. Note there exits η with $\Gamma_i \nvdash \eta$, and obviously $\Gamma_i \cup \{\neg\varphi_i\} \nvdash \eta$.
- Assume Γ^* is not consistent, there exists a finite set $\Delta \subseteq \Gamma^*$ which is inconsistent. We can always find an index i such that $\Delta \subseteq \Gamma_i$, implying that Γ_i is inconsistent, contradiction.

(b) Γ^* is a maximal set, i.e., for each formula φ, either $\varphi \in \Gamma^*$ or $\neg\varphi \in \Gamma^*$.

(c) For each formula φ, we have $\Gamma^* \models \varphi$ iff $\varphi \in \Gamma^*$.

by def

- $\Leftarrow$ If $\varphi \in \Gamma^*$, then $\Gamma^* \vdash \varphi$, then $\Gamma^* \models \varphi$ by the soundness result.

反证

- $\Rightarrow$ If $\Gamma^* \models \varphi$, assume $\varphi \notin \Gamma^*$. By maximality, $\neg\varphi \in \Gamma^*$, then $\Gamma^* \vdash \neg\varphi$, then $\Gamma^* \models \neg\varphi$ by the soundness result, contradiciton.

Let $\sigma = \Gamma^* \cap AP$, then, we prove $\sigma \Vdash \Gamma^*$. We prove $\sigma \Vdash \varphi$ iff $\varphi \in \Gamma^*$.

It follows by structural induction over φ:

- The base case $\varphi = p \in AP$ is easy.
- Let $\varphi = \boxed{\neg\psi}$. Then, by max $\psi \notin \Gamma^*$. By induction hypothesis, $\psi \in \Gamma^*$ iff $\sigma \Vdash \psi$.

  $\varphi \in \Gamma^*$?
- Let $\varphi = \psi \to \eta$. Exercise!

$\square$

Theorem 1.3.28 (Completeness)**.** *If* $\Gamma \models \varphi$*, then* $\Gamma \vdash \varphi$. 完备

Proof. Assume by contradiction that $\Gamma \not\vdash \varphi$, then $\Gamma \cup \{\neg\varphi\}$ is consistent. Thus it is satisfiable, and there is an assignment σ such that $\sigma \Vdash \Gamma \cup \{\neg\varphi\}$. However, this implies that $\sigma \Vdash \Gamma$ and $\sigma \nVdash \varphi$, which violates the assumption $\Gamma \models \varphi$. $\square$

Corollary 1.3.29. $\models \varphi$ *implies that* $\vdash \varphi$.

Theorem 1.3.30 (Compactness)**.** *Given a formula set* Γ*, we have* exercise!

1. Γ *is consistent iff each of its finite subsets is consistent;*
2. Γ *is satisfiable iff each of its finite subsets is satisfiable.*

Rules of Inference for Propositional Logic (cf. page 72):

Rule of Inference	*Tautology*	*Name*
p $p \to q$ $\therefore q$	$(p \land (p \to q)) \to q$	Modus ponens
$\neg q$ $p \to q$ $\therefore \neg p$	$(\neg q \land (p \to q)) \to \neg p$	Modus tollens
$p \to q$ $q \to r$ $\therefore p \to r$	$((p \to q) \land (q \to r)) \to (p \to r)$	Hypothetical syllogism
$p \lor q$ $\neg p$ $\therefore q$	$((p \lor q) \land \neg p) \to q$	Disjunctive syllogism
p $\therefore p \lor q$	$p \to (p \lor q)$	Addition
$p \land q$ $\therefore p$	$(p \land q) \to p$	Simplification
p q $\therefore p \land q$	$((p) \land (q)) \to (p \land q)$	Conjunction
$p \lor q$ $\neg p \lor r$ $\therefore q \lor r$	$((p \lor q) \land (\neg p \lor r)) \to (q \lor r)$	Resolution

1.3.9 Satisfiability

For propositional logic, we consider two important (decision) problems:

1. The SATISFIABILITY problem. i.e, given a formula φ, to decide that if φ is satisfiable.

2. The VADILITY problem. i.e., give a formula φ, to decide that if φ is a tautology. We say it is *valid* in this case.

Notably, these two problems are closely related:

- φ is satisfiable iff $\neg\varphi$ is not valid.
- φ is valid iff $\neg\varphi$ is not satisfiable.

Instead of the canonical VALIDITY problem, we are sometimes concerns about whether $\Gamma \models \varphi$, equivalently, whether $\Gamma \cup \{\neg\varphi\}$ is unsatisfiable. The validity problem can then be considered as the special case of $\Gamma = \emptyset$.

Remark 1.3.31. *At least, we have a naïve approach to test all assignments restricted to propositions occurring in φ.*

$\sigma \in 2^{AP}$

存在 $\sigma \vDash \varphi$?

However, such approach is usually inefficient, as the set of assignments is exponential in $|AP|$.

We now introduce two other classical approaches respectively for satisfiability and validity checking — i.e., the tableau approach and the resolution approach.

Tableau Approach

The main idea of tableau approach is to eventually "decompose" the formula into a set of literals, and finally perform a local satisfiability checking.

Central part of this approach is a set of rewriting rules.

Example 1.3.32. *Tableau Rules*

$$\frac{a \wedge \neg\neg a}{a \wedge a}$$

$$\frac{\Gamma, \neg\neg\varphi}{\Gamma, \varphi}\ (\neg\neg) \qquad \frac{\Gamma, \overbrace{\varphi \to \psi}^{\equiv \neg\varphi \vee \psi}}{\Gamma, \neg\varphi}\ (\to_l) \qquad (\wedge)\ \frac{\Gamma, \varphi \wedge \psi}{\Gamma, \varphi, \psi}$$

left

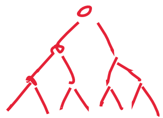
$$\frac{\Gamma, \varphi \to \psi}{\Gamma, \psi}\ (\to_r) \qquad \frac{\Gamma, \overbrace{\neg(\varphi \to \psi)}^{\equiv \varphi \wedge \neg\psi}}{\Gamma, \varphi, \neg\psi}\ (\neg \to) \qquad (\vee_l)\ \frac{\Gamma, \varphi \vee \psi}{\Gamma, \varphi} \qquad (\vee_r)\ \frac{\Gamma, \varphi \vee \psi}{\Gamma, \psi}$$

right

Definition 1.3.33 (Tableau)**.** *Given a formula φ, a tableau of φ is a series of formula set $\Gamma_0, \Gamma_1, \ldots, \Gamma_n$, where:*

- $\Gamma_0 = \{\varphi\}$.
- *Each Γ_{i+1} is obtained from Γ_i by applying some tableau rule.*

- Γ_n *consists of only literals.*



$$\begin{tikzcd} & \{\psi \vee \varphi\} \arrow[dl, "v_l"'] \arrow[dr, "v_r"] & \\ \{\psi\} & & \{\eta \wedge \eta'\} \arrow[d] \\ & & \{\eta, \eta'\} \end{tikzcd}$$

A tableau is consistent if its last formula set contains no conflicting literals. $\{a, \neg a\}$ conflicting

Theorem 1.3.34. *A formula φ is satisfiable iff it has a consistent tableau.*

Example 1.3.35. *Suppose that $\varphi = (p \to \neg p) \to p$, then we have the following tableau for φ: $\{(p \to \neg p) \to p\}$, $\{\neg(p \to \neg p)\}$, $\{p, \neg\neg p\}$, $\{p\}$. Hence φ is satisfiable.*

Example 1.3.36. *Suppose that $\psi = \neg((\neg p \to p) \to p)$, then we have two possible tableaux: $\{\neg((\neg p \to p) \to p)\}$, $\{\neg p \to p, \neg p\}$, $\{\neg\neg p, \neg p\}$, $\{p, \neg p\}$, $\{\neg((\neg p \to p) \to p)\}$, $\{\neg p \to p, \neg p\}$, $\{p, \neg p\}$ — none all of is consistent, and hence ψ is not satisfiable.*

Resolution Approach

We first transform the formula φ into CNF. A CNF formula φ can also be seen as a set of clauses, and a clause is a set of literals.

Remark 1.3.37. *The empty clause, denoted by $\square$, is unsatisfiable by definition. The empty formula describes an empty set of clauses and is satisfiable by definition.*

$\Gamma = \{C_1, C_2, \dots C_i\}$

We allow set operations on clauses as expected.

Definition 1.3.38 (Resolution)**.** *Let C_1 and C_2 be two clauses and L) be a literal with the following property: $L \in C_1$ and $\neg L \in C_2$. Then one can compute the clause*

$L \in AP$

$$R = (C_1 \setminus \{L\}) \cup (C_2 \setminus \{\neg L\})$$

that is denoted as the resolvent *of the clauses C_1 and C_2 over L.*

Lemma 1.3.39. *Let φ be a CNF formula and R be the resolvent of two clauses C_1 and C_2 from φ. Then φ and $\varphi \cup \{R\}$ are equivalent.*

C_1, C_2

$\varphi \equiv \varphi \cup R$

$\{\{a\}\{\neg a\}\} \cup R \equiv \emptyset$

Define $Res(\varphi)$ the formula $\varphi \cup \{R\}$ where R is a resolvant of two clauses in φ. Let $Res^0(\varphi) = \varphi$, and $Res^{i+1}(\varphi) =$

$Res(Res^i(\varphi))$. Let $Res^*(\varphi) = \lim_i Res^i(\varphi)$. Then, we have:

Theorem 1.3.40. *A CNF formula φ is unsatisfiable iff $\square \in Res^*(\varphi)$.*

The resolution based algorithm. Construct $Res^*(\varphi)$:

- if for some $i > 0$: $\square \in Res^i(\varphi)$, φ is unsatisfiable.
- if for some $\square \notin Res^i(\varphi) = Res^{i+1}(\varphi)$, φ is satisfiable.

Complexity of this naive method. Since in a clause a variable occurs either as a positive literal, or negative literal, or it does not occur at all, for a formula having n variables the runtime and memory consumption lie in the order of 3^n in the worst case.

$\{P_1\ P_2 \cdots P_n\}$

1) $P_1 \in C$
2) $\neg P_1 \in C$
3) other

DPLL

The SAT-algorithm, which was proposed in 1960 by Davis and Putnam, is based on the elimination method and uses a couple of optimizations:

- Subsumption checks

  $C_1 \subseteq C_2$

- *Pure literal* detection: literal occurring only positive or only negative. If there is a pure literal l

$(a\quad) \wedge (a\quad) \wedge (a\quad)$

in Γ, then remove all clauses containing l.

- Variable elimination (by adding all resolvent clauses)

The optimizations improve the runtime behavior in practice, but not the worst case complexity of the naive method. To further improve the efficiency, Davis, Putnam, Logemann and Loveland proposed in 1962 the following process to accelerate the resolution process of Γ, and it consist of four rules.

1. Tautology rule: remove all tautologies from Γ.

2. Single literal rule: if there is some $l \in \Gamma$ and l is a literal, then remove all clauses containing l, and delete all complementary literals occurring in the rest clauses.

   $a = tt$

   $C_1 \; C_2 \; \{a\} \; C_3 \; C_4$

3. Pure literal detection: if there is a pure literal l in Γ, then remove all clauses containing l.

4. Splitting-rule Suppose that

   $$\Gamma = \{C_1 \vee p, \ldots, C_n \vee p, C_1' \vee \neg p, \ldots C_m' \vee \neg p, D_1, \ldots, D_k\}$$

   then split Γ into

   $$\Gamma' = \{C_1, \ldots, C_n, D_1, \ldots D_k\}$$

   ① $p \in 6$

   ② $p \notin 6$

and

$$\Gamma'' = \{C'_1, \ldots, C'_m, D_1, \ldots D_k\}$$

Γ is satisfiable iff Γ' or Γ'' is satisfiable.

Example 1.3.41. *Suppose that* $\Gamma = \{p, p \to q, q \to r, \neg(p \to r)\}$. *Then we first normalize the set as* $\{p, \neg p \vee q, \neg q \vee r, \neg r\}$.

Further Reading.

- A. Biere, M. J. H. Heule, H. van Maaren, T. Walsh: Handbook of Satisfiability, IOS Press, 2009
- The International Conferences on Theory and Applications of Satisfiability Testing (SAT): `http://www.satisfiability.org/`
- Tons of workshop, conference, and journal papers