Propositional and Predicate Logic - V

Petr Gregor

KTIML MFF UK

WS 2023/2024

Compactness theorem

Theorem *A theory T has a model iff every finite subset of T has a model.*

Proof 1 The implication from left to right is obvious. If T has no model, then it is inconsistent, i.e. $\perp$ is provable by a systematic tableau τ from T. Since τ is finite, $\perp$ is provable from some finite $T' \subseteq T$, i.e. T' has no model. □

Remark *This proof is based on finiteness of proofs, soundness and completeness. We present an alternative proof (applying König's lemma).*

Proof 2 Let $T = \{\varphi_i \mid i \in \mathbb{N}\}$. Consider a tree S on (certain) finite binary strings σ ordered by being a prefix. We put $\sigma \in S$ if and only if there exists an assignment v with prefix σ such that $v \models \varphi_i$ for every $i \leq \mathrm{lth}(\sigma)$.

Observation *S has an infinite branch if and only if T has a model.*

Since $\{\varphi_i \mid i \in n\} \subseteq T$ has a model for every $n \in \mathbb{N}$, every level in S is nonempty. Thus S is infinite and moreover binary, hence by König's lemma, S contains an infinite branch. □

Application of compactness

A graph (V, E) is *k-colorable* if there exists $c\colon V \to \{1, \dots, k\}$ such that $c(u) \neq c(v)$ for every edge $\{u, v\} \in E$.

Theorem *A countably infinite graph $G = (V, E)$ is k-colorable if and only if every finite subgraph of G is k-colorable.*

Proof The implication $\Rightarrow$ is obvious. Assume that every finite subgraph of G is k-colorable. Consider $\mathbb{P} = \{p_{u,i} \mid u \in V, 1 \leq i \leq k\}$ and a theory T with axioms

$$\begin{aligned} & p_{u,1} \vee \dots \vee p_{u,k} && \text{for every } u \in V, \\ & \neg(p_{u,i} \wedge p_{u,j}) && \text{for every } u \in V, i < j \leq k, \\ & \neg(p_{u,i} \wedge p_{v,i}) && \text{for every } \{u, v\} \in E, i \leq k. \end{aligned}$$

Then G is k-colorable if and only if T has a model. By compactness, it suffices to show that every finite $T' \subseteq T$ has a model. Let G' be the subgraph of G induced by vertices u such that $p_{u,i}$ appears in T' for some i. Since G' is k-colorable by the assumption, the theory T' has a model. $\square$

Resolution method - introduction

Main features of the resolution method (*informally*)

- is the underlying method of many systems, e.g. Prolog interpreters, SAT solvers, automated deduction / verification systems, . . .
- assumes input formulas in CNF (in general, *"expensive"* transformation),
- works under set representation (clausal form) of formulas,
- has a single rule, so called a resolution rule,
- has no explicit axioms (or atomic tableaux), but certain axioms are incorporated *"inside"* via various formatting rules,
- is a *refutation* procedure, similarly as the tableau method; that is, it tries to show that a given formula (or theory) is unsatisfiable,
- has several refinements e.g. with specific conditions on when the resolution rule may be applied.

Set representation (clausal form) of CNF formulas

- A *literal* l is a prop. letter or its negation. $\bar{l}$ is its *complementary* literal.
- A *clause* C is a finite set of literals (*"forming disjunction"*). The empty clause, denoted by $\square$, is never satisfied (has no satisfied literal).
- A *formula* S is a (possibly infinite) set of clauses (*"forming conjunction"*). An empty formula $\emptyset$ is always satisfied (is has no unsatisfied clause). Infinite formulas represent infinite theories (as conjunction of axioms).
- A (*partial*) *assignment* $\mathcal{V}$ is a consistent set of literals, i.e. not containing any pair of complementary literals. An assignment $\mathcal{V}$ is *total* if it contains a positive or negative literal for each propositional letter.
- $\mathcal{V}$ *satisfies* S, denoted by $\mathcal{V} \models S$, if $C \cap \mathcal{V} \neq \emptyset$ for every $C \in S$.

$((\neg p \vee q) \wedge (\neg p \vee \neg q \vee r) \wedge (\neg r \vee \neg s) \wedge (\neg t \vee s) \wedge s)$ is represented by

$$S = \{\{\neg p, q\}, \{\neg p, \neg q, r\}, \{\neg r, \neg s\}, \{\neg t, s\}, \{s\}\} \quad \text{and}$$
$$\mathcal{V} \models S \quad \text{for} \quad \mathcal{V} = \{s, \neg r, \neg p\}$$

Resolution rule

Let C_1, C_2 be clauses with $l \in C_1$, $\bar{l} \in C_2$ for some literal l. Then from C_1 and C_2 infer through the literal l the clause C, called a *resolvent*, where

$$C = (C_1 \setminus \{l\}) \cup (C_2 \setminus \{\bar{l}\}).$$

Equivalently, if $\sqcup$ means union of disjoint sets,

$$\frac{C_1' \sqcup \{l\}, C_2' \sqcup \{\bar{l}\}}{C_1' \cup C_2'}$$

For example, from $\{p, q, r\}$ and $\{\neg p, \neg q\}$ we can infer $\{q, \neg q, r\}$ or $\{p, \neg p, r\}$.

Observation *The resolution rule is sound; that is, for every assignment $\mathcal{V}$*

$$\mathcal{V} \models C_1 \text{ and } \mathcal{V} \models C_2 \quad \Rightarrow \quad \mathcal{V} \models C.$$

Remark *The resolution rule is a special case of the (so called) cut rule*

$$\frac{\varphi \vee \psi, \ \neg\varphi \vee \chi}{\psi \vee \chi}$$

where φ, ψ, χ are arbitrary formulas.

Resolution proof

- A *resolution proof* of a clause C from a formula S is a finite sequence $C_0, \ldots, C_n = C$ such that for every $i \leq n$, $C_i \in S$ or C_i is a resolvent of some previous clauses,
- a clause C is (resolution) *provable* from S, denoted by $S \vdash_R C$, if it has a resolution proof from S,
- a (resolution) *refutation* of formula S is a resolution proof of $\square$ from S,
- S is (resolution) *refutable* if $S \vdash_R \square$.

Theorem (soundness) *If S is resolution refutable, then S is unsatisfiable.*

Proof Let $S \vdash_R \square$. If it was $\mathcal{V} \models S$ for some assignment $\mathcal{V}$, from the soundness of the resolution rule we would have $\mathcal{V} \models \square$, which is impossible. ■

Resolution trees and closures

A *resolution tree* of a clause C from formula S is finite binary tree with nodes labeled by clauses so that

- (i) the root is labeled C,
- (ii) the leaves are labeled with clauses from S,
- (iii) every inner node is labeled with a resolvent of the clauses in his sons.

Observation *C has a resolution tree from S if and only if $S \vdash_R C$.*

A *resolution closure* $\mathcal{R}(S)$ of a formula S is the smallest set satisfying

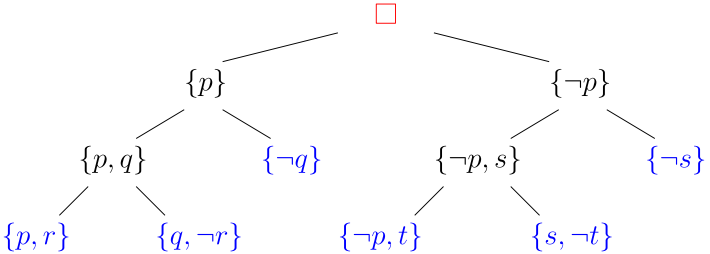
- (i) $C \in \mathcal{R}(S)$ for every $C \in S$,
- (ii) if $C_1, C_2 \in \mathcal{R}(S)$ and C is a resolvent of C_1, C_2, then $C \in \mathcal{R}(S)$.

Observation *$C \in \mathcal{R}(S)$ if and only if $S \vdash_R C$.*

Remark *All notions on resolution proofs can therefore be equivalently introduced in terms of resolution trees or resolution closures.*

Example

Formula $((p \vee r) \wedge (q \vee \neg r) \wedge (\neg q) \wedge (\neg p \vee t) \wedge (\neg s) \wedge (s \vee \neg t))$ is unsatisfiable since for $S = \{\{p, r\}, \{q, \neg r\}, \{\neg q\}, \{\neg p, t\}, \{\neg s\}, \{s, \neg t\}\}$ we have $S \vdash_R \square$.



The resolution closure of S (*the closure of S under resolution*) is

$$\mathcal{R}(S) = \{\{p, r\}, \{q, \neg r\}, \{\neg q\}, \{\neg p, t\}, \{\neg s\}, \{s, \neg t\}, \{p, q\}, \{\neg r\}, \{r, t\}, \{q, t\}, \{\neg t\}, \{\neg p, s\}, \{r, s\}, \{t\}, \{q\}, \{q, s\}, \square, \{\neg p\}, \{p\}, \{r\}, \{s\}\}.$$

Reduction by substitution

Let S be a formula and l be a literal. Let us define

$$S^l = \{C \setminus \{\bar{l}\} \mid l \notin C \in S\}.$$

Observation

- S^l is equivalent to a formula obtained from S by substituting the constant $\top$ (true, 1) for all literals l and the constant $\bot$ (false, 0) for all literals $\bar{l}$ in S,
- Neither l nor $\bar{l}$ occurs in (the clauses of) S^l.
- if $\{\bar{l}\} \in S$, then $\square \in S^l$.

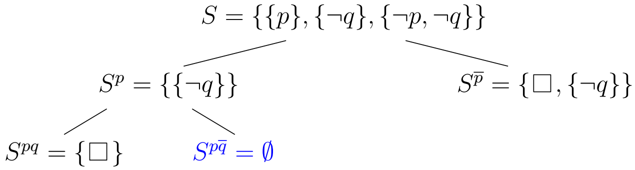
Lemma *S is satisfiable if and only if S^l or $S^{\bar{l}}$ is satisfiable.*

Proof ($\Rightarrow$) Let $\mathcal{V} \models S$ for some $\mathcal{V}$ and assume (w.l.o.g.) that $\bar{l} \notin \mathcal{V}$.

- Then $\mathcal{V} \models S^l$ as for $l \notin C \in S$ we have $\mathcal{V} \setminus \{l, \bar{l}\} \models C$ and thus $\mathcal{V} \models C \setminus \{\bar{l}\}$.
- On the other hand ($\Leftarrow$), assume (w.l.o.g.) that $\mathcal{V} \models S^l$ for some $\mathcal{V}$.
- Since neither l nor $\bar{l}$ occurs in S^l, we have $\mathcal{V}' \models S^l$ for $\mathcal{V}' = (\mathcal{V} \setminus \{\bar{l}\}) \cup \{l\}$.
- Then $\mathcal{V}' \models S$ since for $C \in S$ containing l we have $l \in \mathcal{V}'$ and for $C \in S$ not containing l we have $\mathcal{V}' \models (C \setminus \{\bar{l}\}) \in S^l$. ■

Tree of reductions

Step by step reductions of literals can be represented in a binary tree.



Corollary *S is unsatisfiable if and only if every branch contains $\square$.*

Remarks *Since S can be infinite over a countable language, this tree can be infinite. However, if S is unsatisfiable, by the compactness theorem there is a finite $S' \subseteq S$ that is unsatisfiable. Thus after reduction of all literals occurring in S', there will be $\square$ in every branch after finitely many steps.*

(Refutation) completeness of resolution

Theorem *If a finite S is unsatisfiable, it is resolution refutable, i.e. $S \vdash_R \square$.*

Proof By induction on the number of variables in S we show that $S \vdash_R \square$.

- If unsatisfiable S has no variable, it is $S = \{\square\}$ and thus $S \vdash_R \square$,
- Let l be a literal occurring in S. By Lemma, S^l and $S^{\bar{l}}$ are unsatisfiable.
- Since S^l and $S^{\bar{l}}$ have less variables than S, by induction there exist resolution trees T^l and $T^{\bar{l}}$ for derivation of $\square$ from S^l resp. $S^{\bar{l}}$.
- If every leaf of T^l is in S, then T^l is a resolution tree of $\square$ from S, $S \vdash_R \square$.
- Otherwise, by appending the literal $\bar{l}$ to every leaf of T^l that is not in S, (and to all predecessors) we obtain a resolution tree of $\{\bar{l}\}$ from S.
- Similarly, we get a resolution tree $\{l\}$ from S by appending l in the tree $T^{\bar{l}}$.
- By resolution of roots $\{\bar{l}\}$ and $\{l\}$ we get a resolution tree of $\square$ from S. ■

Corollary *If S is unsatisfiable, it is resolution refutable, i.e. $S \vdash_R \square$.*

Proof Follows from the previous theorem by applying compactness.

Linear resolution - introduction

The resolution method can be significantly refined.

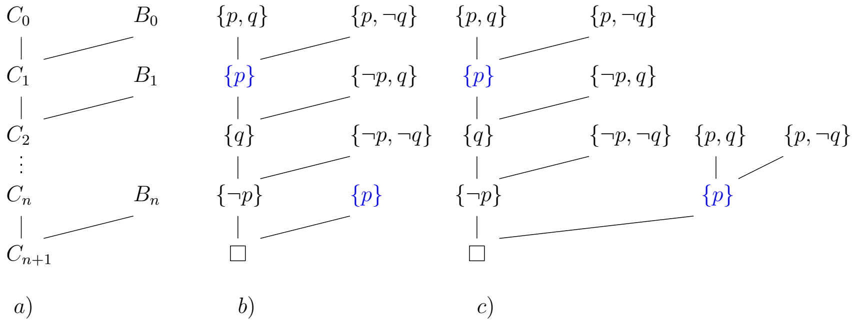
- A *linear proof* of a clause C from a formula S is a finite sequence of pairs $(C_0, B_0), \ldots, (C_n, B_n)$ such that $C_0 \in S$ and for every $i \leq n$
  - *i)* $B_i \in S$ or $B_i = C_j$ for some $j < i$, and
  - *ii)* C_{i+1} is a resolvent of C_i and B_i where $C_{n+1} = C$.
- C_0 is called a *starting* clause, C_i a *central* clause, B_i a *side* clause.
- C is *linearly provable* from S, $S \vdash_L C$, if it has a linear proof from S.
- A *linear refutation* of S is a linear proof of $\square$ from S.
- S is *linearly refutable* if $S \vdash_L \square$.

Observation (soundness) *If S is linearly refutable, it is unsatisfiable.*

Proof Every linear proof can be transformed to a (general) resolution proof.

Remark *The completeness is preserved as well (proof omitted here).*

Example of linear resolution



- $a)$ a general form of linear resolution,
- $b)$ for $S = \{\{p, q\}, \{p, \neg q\}, \{\neg p, q\}, \{\neg p, \neg q\}\}$ we have $S \vdash_L \square$,
- $c)$ a transformation of a linear proof to a (general) resolution proof.

LI-resolution

Linear resolution can be further refined for Horn formulas as follows.

- a *Horn clause* is a clause containing at most one positive literal,
- a *Horn formula* is a (possibly infinite) set of Horn clauses,
- a *fact* is a (Horn) clause $\{p\}$ where p is a positive literal,
- a *rule* is a (Horn) clause with exactly one positive literal and at least one negative literal. Rules and facts are *program clauses*,
- a *goal* is a nonempty (Horn) clause with only negative literals.

Observation *If a Horn formula S is unsatisfiable and $\square \notin S$, it contains some fact and some goal.*

Proof If S does not contain any fact (goal), it is satisfied by the assignment of all propositional variables to 0 (resp. to 1). ■

A *linear input resolution* (*LI-resolution*) from a formula S is a linear resolution from S in which every side clause B_i is from the (input) formula S. We write $S \vdash_{LI} C$ to denote that C is provable by LI-resolution from S.

Completeness of LI-resolution for Horn formulas

Theorem *If T is satisfiable Horn formula but $T \cup \{G\}$ is unsatisfiable for some goal G, then $\square$ has a LI-resolution from $T \cup \{G\}$ with starting clause G.*

Proof By the compactness theorem we may assume that T is finite.

- We proceed by induction on the number of variables in T.
- By Observation, T contains a fact $\{p\}$ for some variable p.
- By Lemma, $T' = (T \cup \{G\})^p = T^p \cup \{G^p\}$ is unsatisfiable where $G^p = G \setminus \{\overline{p}\}$.
- If $G^p = \square$, we have $G = \{\overline{p}\}$ and thus $\square$ is a resolvent of G and $\{p\} \in T$.
- Otherwise, since T^p is satisfiable (by the assignment satisfying T) and has less variables than T, by induction assumption, there is an LI-resolution of $\square$ from T' starting with G^p.
- By appending the literal $\overline{p}$ to all leaves that are not in $T \cup \{G\}$ (and nodes below) we obtain an LI-resolution of $\{\overline{p}\}$ from $T \cup \{G\}$ that starts with G.
- By an additional resolution step with the fact $\{p\} \in T$ we resolve $\square$. ■

Example of LI-resolution

$$T = \{\{p, \neg r, \neg s\}, \{r, \neg q\}, \{q, \neg s\}, \{s\}\}, \qquad G = \{\neg p, \neg q\}$$

$$T^{s} = \{\{p, \neg r\}, \{r, \neg q\}, \{q\}\}$$

$$T^{sq} = \{\{p, \neg r\}, \{r\}\}$$

$$T^{sqr} = \{\{p\}\}$$

$$G^{sqr} = \{\neg p\} \quad \{p\}$$

$$G^{sqrp} = \square$$

$$G^{sq} = \{\neg p\} \quad \{p, \neg r\}$$
$$\{\neg r\} \quad \{r\}$$
$$\square$$

$$G^{s} = \{\neg p, \neg q\} \quad \{p, \neg r\}$$
$$\{\neg q, \neg r\} \quad \{r, \neg q\}$$
$$\{\neg q\} \quad \{q\}$$
$$\square$$

$$G = \{\neg p, \neg q\} \quad \{p, \neg r, \neg s\}$$
$$\{\neg q, \neg r, \neg s\} \quad \{r, \neg q\}$$
$$\{\neg q, \neg s\} \quad \{q, \neg s\}$$
$$\{\neg s\} \quad \{s\}$$
$$\square$$

$$T^{sqr}, G^{sqr} \vdash_{LI} \square \qquad T^{sq}, G^{sq} \vdash_{LI} \square \qquad T^{s}, G^{s} \vdash_{LI} \square \qquad T, G \vdash_{LI} \square$$

Program in Prolog

A (propositional) *program* (in Prolog) is a Horn formula containing only program clauses, i.e. facts or rules.

a rule	p :– $q, r.$	$q \wedge r \rightarrow p$	$\{p, \neg q, \neg r\}$	
	p :– $s.$	$s \rightarrow p$	$\{p, \neg s\}$	
	q :– $s.$	$s \rightarrow q$	$\{q, \neg s\}$	
a fact	$r.$	r	$\{r\}$	
	$s.$	s	$\{s\}$	*a program*
a query	?– $p, q.$		$\{\neg p, \neg q\}$	*a goal*

We would like to know whether a given query follows from a given program.

Corollary *For every program P and query $(p_1 \wedge \ldots \wedge p_n)$ it is equivalent that*

(1) $P \models p_1 \wedge \ldots \wedge p_n$,

(2) $P \cup \{\neg p_1, \ldots, \neg p_n\}$ is unsatisfiable,

(3) $\square$ has LI-resolution from $P \cup \{G\}$ starting by goal $G = \{\neg p_1, \ldots, \neg p_n\}$.

Hilbert's calculus

- basic connectives: $\neg$, $\rightarrow$ (others can be defined from them)
- *logical axioms* (schemes of axioms):

$$
\begin{array}{rc}
(i) & \varphi \rightarrow (\psi \rightarrow \varphi) \\
(ii) & (\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi)) \\
(iii) & (\neg\varphi \rightarrow \neg\psi) \rightarrow (\psi \rightarrow \varphi)
\end{array}
$$

  where φ, ψ, χ are any propositions (of a given language).
- *a rule of inference*:

$$\frac{\varphi,\ \varphi \rightarrow \psi}{\psi} \qquad \text{(modus ponens)}$$

A *proof* (in *Hilbert-style*) of a formula φ from a theory T is a finite sequence $\varphi_0, \ldots, \varphi_n = \varphi$ of formulas such that for every $i \leq n$

- φ_i is a logical axiom or $\varphi_i \in T$ (an axiom of the theory), or
- φ_i can be inferred from the previous formulas applying a rule of inference.

Remark *Choice of axioms and inference rules differs in various Hilbert-style proof systems.*

Example and soundness

A formula φ is *provable* from T if it has a proof from T, denoted by $T \vdash_H \varphi$.

If $T = \emptyset$, we write $\vdash_H \varphi$. E.g. for $T = \{\neg\varphi\}$ we have $T \vdash_H \varphi \to \psi$ for every ψ.

1)	$\neg\varphi$	an axiom of T
2)	$\neg\varphi \to (\neg\psi \to \neg\varphi)$	a logical axiom (i)
3)	$\neg\psi \to \neg\varphi$	by modus ponens from 1), 2)
4)	$(\neg\psi \to \neg\varphi) \to (\varphi \to \psi)$	a logical axiom (iii)
5)	$\varphi \to \psi$	by modus ponens from 3), 4)

Theorem *For every theory T and formula φ, $T \vdash_H \varphi \;\Rightarrow\; T \models \varphi$.*

Proof

- If φ is an axiom (logical or from T), then $T \models \varphi$ (l. axioms are tautologies),
- if $T \models \varphi$ and $T \models \varphi \to \psi$, then $T \models \psi$, i.e. modus ponens is sound,
- thus every formula in a proof from T is valid in T. □

Remark *The completeness holds as well, i.e. $T \models \varphi \Rightarrow T \vdash_H \varphi$.*