Propositional and Predicate Logic - III

Petr Gregor

KTIML MFF UK

WS 2025/2026

Properties of theories

A propositional theory T over $\mathbb{P}$ is *(semantically)*

- *inconsistent* (*unsatisfiable*) if $T \models \perp$, otherwise is *consistent* (*satisfiable*),
- *complete* if it is consistent, and $T \models \varphi$ or $T \models \neg\varphi$ for every $\varphi \in \mathrm{PF}_{\mathbb{P}}$, i.e. no proposition over $\mathbb{P}$ is independent in T,
- an *extension* of a theory T' over $\mathbb{P}'$ if $\mathbb{P}' \subseteq \mathbb{P}$ and $\theta^{\mathbb{P}'}(T') \subseteq \theta^{\mathbb{P}}(T)$; we say that an extension T of a theory T' is *simple* if $\mathbb{P} = \mathbb{P}'$; and *conservative* if $\theta^{\mathbb{P}'}(T') = \theta^{\mathbb{P}}(T) \cap \mathrm{PF}_{\mathbb{P}'}$,
- *equivalent* with a theory T' if T is an extension of T' and vice-versa,

Observation *Let T and T' be theories over $\mathbb{P}$. Then T is (semantically)*

(1) *consistent if and only if it has a model,*

(2) *complete if and only if it has a single model,*

(3) *extension of T' if and only if $M^{\mathbb{P}}(T) \subseteq M^{\mathbb{P}}(T')$,*

(4) *equivalent with T' if and only if $M^{\mathbb{P}}(T) = M^{\mathbb{P}}(T')$.*

Algebra of propositions

Let T be a consistent theory over $\mathbb{P}$. On the quotient set $\mathrm{PF}_{\mathbb{P}}/\sim_T$ we define operations $\neg$, $\wedge$, $\vee$, $\bot$, $\top$ (correctly) by use of representatives, e.g.

$$[\varphi]_{\sim_T} \wedge [\psi]_{\sim_T} = [\varphi \wedge \psi]_{\sim_T}$$

Then $AV^{\mathbb{P}}(T) = \langle \mathrm{PF}_{\mathbb{P}}/\sim_T, \neg, \wedge, \vee, \bot, \top \rangle$ is *algebra of propositions* for T.

Since $\varphi \sim_T \psi \iff M(T,\varphi) = M(T,\psi)$, it follows that $h([\varphi]_{\sim_T}) = M(T,\varphi)$ is a (well-defined) injective function $h\colon \mathrm{PF}_{\mathbb{P}}/\sim_T \to \mathcal{P}(M(T))$ and

$$\begin{aligned}
h(\neg[\varphi]_{\sim_T}) &= M(T) \setminus M(T,\varphi)\\
h([\varphi]_{\sim_T} \wedge [\psi]_{\sim_T}) &= M(T,\varphi) \cap M(T,\psi)\\
h([\varphi]_{\sim_T} \vee [\psi]_{\sim_T}) &= M(T,\varphi) \cup M(T,\psi)\\
h([\bot]_{\sim_T}) = \emptyset, &\quad h([\top]_{\sim_T}) = M(T)
\end{aligned}$$

Moreover, h is *surjective* if $M(T)$ is *finite*.

Corollary If T is a consistent theory over a finite $\mathbb{P}$, then $AV^{\mathbb{P}}(T)$ is a Boolean algebra *isomorphic* via h to the (finite) algebra of sets $\underline{\mathcal{P}}(M(T))$.

Analysis of theories over finite languages

Let T be a consistent theory over $\mathbb{P}$ where $|\mathbb{P}| = n \in \mathbb{N}^+$ and $m = |M^{\mathbb{P}}(T)|$.
Then the number of (mutually) nonequivalent

- propositions (or theories) over $\mathbb{P}$ is 2^{2^n},
- propositions over $\mathbb{P}$ that are valid (contradictory) in T is $2^{2^n - m}$,
- propositions over $\mathbb{P}$ that are independent in T is $2^{2^n} - 2.2^{2^n - m}$,
- simple extensions of T is 2^m, out of which 1 is inconsistent,
- complete simple extensions of T is m.

And the number of (mutually) T-nonequivalent

- propositions over $\mathbb{P}$ is 2^m,
- propositions over $\mathbb{P}$ that are valid (contradictory) (in T) is 1,
- propositions over $\mathbb{P}$ that are independent (in T) is $2^m - 2$.

Proof By the bijection of $\mathrm{PF}_{\mathbb{P}}/\sim$ resp. $\mathrm{PF}_{\mathbb{P}}/\sim_T$ with $\mathcal{P}(M(\mathbb{P}))$ resp. $\mathcal{P}(M^{\mathbb{P}}(T))$ it suffices to determine the number of appropriate subsets of models. □

SAT problem and solvers

- Problem SAT: Is φ in CNF satisfiable?
- *Example Is it possible to perfectly cover the chessboard without two diagonally removed corners using the domino tiles?*

  We can easily form a propositional formula that is satisfiable, if and only if the answer is yes. Then we can test its satisfiability by a SAT solver.
- Best SAT solvers: `www.satcompetition.org`.
- SAT solver in the demo: `Glucose`, CNF format: `DIMACS`.
- *Can all the mathematics be translated into logical formulas?*
  AI, theorem proving, Peano: *Formulario* (1895-1908), Mizar, LEAN
- How can we solve it more *elegantly*? What is our approach based on?

2-SAT

- A proposition in CNF is in *k-CNF* if every its clause has at most k literals.
- *k-SAT* is the problem of satisfiability of a given proposition in k-CNF.

Although for $k = 3$ it is an NP-complete problem, we show that 2-SAT can be solved in *linear* time (with respect to the length of φ).

We neglect implementation details (computational model, representation in memory) and we use the following fact, see [ADS I].

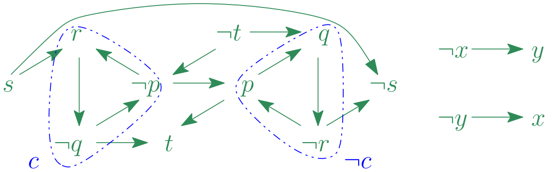
Proposition *A partition of a directed graph (V, E) to strongly connected components can be found in time $\mathcal{O}(|V| + |E|)$.*

- A directed graph G is *strongly connected* if for every two vertices u and v there are directed paths in G both from u to v and from v to u.
- A strongly connected *component* of a graph G is a maximal strongly connected subgraph of G.

Implication graphs

An *implication graph* of a proposition φ in 2-CNF is a directed graph G_φ s.t.

- vertices are all the propositional letters in φ and their negations,
- a clause $l_1 \vee l_2$ in φ is represented by a pair of edges $\overline{l_1} \to l_2$, $\overline{l_2} \to l_1$,
- a clause l_1 in φ is represented by an edge $\overline{l_1} \to l_1$.



$$p \wedge (\neg p \vee q) \wedge (\neg q \vee \neg r) \wedge (p \vee r) \wedge (r \vee \neg s) \wedge (\neg p \vee t) \wedge (q \vee t) \wedge \neg s \wedge (x \vee y)$$

Proposition *φ is satisfiable if and only if no strongly connected component of G_φ contains a pair of complementary literals.*

Proof Every satisfying assignment assigns the same value to all the literals in a same component. Thus the implication from left to right holds (necessity).

Satisfying assignment

For the implication from right to left (sufficiency), let G^*_φ be the graph obtained from G_φ by contracting strongly connected components to single vertices.

Observation *G^*_φ is acyclic, and therefore has a topological ordering $<$.*

- A directed graph is *acyclic* if it is has no directed *cycles*.
- A linear ordering $<$ of vertices of a directed graph is *topological* if $p < q$ for every edge from p to q.

Now for every unassigned component in an increasing order by $<$, assign 0 to all its literals and 1 to all literals in the complementary component.

It remains to show that such assignment v satisfies φ. If not, then G^*_φ contains edges $p \to q$ and $\overline{q} \to \overline{p}$ with $v(p) = 1$ and $v(q) = 0$. But this contradicts the order of assigning values to components since $p < q$ and $\overline{q} < \overline{p}$. $\square$

Corollary *2-SAT can be solved in a linear time.*

Horn-SAT

- A *unit clause* is a clause containing a single literal,
- a *Horn clause* is a clause containing at most one positive literal,

$$\neg p_1 \vee \cdots \vee \neg p_n \vee q \quad \sim \quad (p_1 \wedge \cdots \wedge p_n) \to q$$

- a *Horn formula* is a conjunction of Horn clauses,
- *Horn-SAT* is the problem of satisfiability of a given Horn formula.

Algorithm

(1) *if φ contains a pair of unit clauses l and $\overline{l}$, then it is not satisfiable,*

(2) *if φ contains a unit clause l, then assign 1 to l, remove all clauses containing l, remove $\overline{l}$ from all clauses, and repeat from the start,*

(3) *if φ does not contain a unit clause, then it is satisfied by assigning 0 to all remaining propositional variables.*

Step (2) is called *unit propagation*.

Unit propagation

$$(\neg p \vee q) \wedge (\neg p \vee \neg q \vee r) \wedge (\neg r \vee \neg s) \wedge (\neg t \vee s) \wedge s \qquad v(s) = 1$$
$$(\neg p \vee q) \wedge (\neg p \vee \neg q \vee r) \wedge \neg r \qquad v(\neg r) = 1$$
$$(\neg p \vee q) \wedge (\neg p \vee \neg q) \qquad v(p) = v(q) = v(t) = 0$$

Observation *Let φ^l be the proposition obtained from φ by unit propagation. Then φ^l is satisfiable if and only if φ is satisfiable.*

Corollary *The algorithm is correct (it solves Horn-SAT).*

Proof The correctness in Step (1) is obvious, in Step (2) it follows from the observation, in Step (3) it follows from the *Horn form* since every remaining clause contains at least one negative literal.

Note A direct implementation requires quadratic time, but with an appropriate representation in memory, one can achieve linear time (w.r.t. the length of φ).

DPLL algorithm

A literal l is *pure* in a CNF formula φ if l occurs in φ and $\overline{l}$ does not occur in φ.

Algorithm DPLL(φ)

(1) *while φ contains a unit clause l, assign 1 to l, remove all clauses containing l, remove $\overline{l}$ from all clauses, and repeat, (unit propagation)*

(2) *while φ contains a pure literal l, assign 1 to l, remove all clauses containing l and repeat, (pure literal elimination)*

(3) *if φ contains an empty clause, then it is not satisfiable,*

(4) *if φ does not contain any clause, then it is satisfiable,*

(5) *choose an unassigned propositional letter p and run DPLL($\varphi \wedge p$) and DPLL($\varphi \wedge \neg p$). (branching)*

Note The algoritm runs in exponentional time in the worst case. Its correctness is easy to verify.