Formal Proofs and their Lengths V

Propositional sequent calculus

Definition 1. Let $A_1, \ldots, A_n$ and $B_1, \ldots, B_m$ be propositional formulas. A sequent is a symbol of the form

$$A_1, \ldots, A_n \longrightarrow B_1, \ldots, B_m.$$

The semantics for a sequent are the same as for the formula

$$\bigwedge_i A_i \rightarrow \bigvee_i B_i,$$

which is semantically equivalent to

$$\bigvee_i \neg A_i \vee \bigvee_i B_i.$$

Definition 2. The Sequent calculus is a propositional proof system (which proves sequents), whose proves are given as follows.

A proof of a sequent S is a sequence of sequents, $S_1, \ldots, S_k$, where $S_k = S$ and each S_i is either an *initial sequent*

$$x \longrightarrow x,$$

where x is a propositional variable or was derived from $S_j, S_l, 1 \leq j \leq l \leq$ by one of the following rules.

Weak Structural Rules

$$\text{(Exchange:L)}\frac{\Gamma, A, B, \Pi \longrightarrow \Delta}{\Gamma, B, A, \Pi \longrightarrow \Delta} \qquad \text{(Exchange:R)}\frac{\Gamma \longrightarrow \Delta, A, B, \Lambda}{\Gamma \longrightarrow \Delta, B, A, \Lambda}$$

$$\text{(Contraction:L)}\frac{\Gamma, A, A, \Pi \longrightarrow \Delta}{\Gamma, A, \Pi \longrightarrow \Delta} \qquad \text{(Contraction:R)}\frac{\Gamma \longrightarrow \Delta, A, A, \Lambda}{\Gamma \longrightarrow \Delta, A, \Lambda}$$

$$\text{(Weakening:L)}\frac{\Gamma \longrightarrow \Delta}{A, \Gamma \longrightarrow \Delta} \qquad \text{(Weakening:R)}\frac{\Gamma \longrightarrow \Delta}{\Gamma \longrightarrow \Delta, A}$$

The Cut Rule

$$\text{(Cut)}\frac{\Gamma \longrightarrow \Delta, A \qquad \Gamma, A \longrightarrow \Delta}{\Gamma \longrightarrow \Delta}$$

The Propositional Rules

$$\text{($\neg$:L)}\frac{\Gamma \longrightarrow \Delta, A}{\Gamma, \neg A \longrightarrow \Delta} \qquad \text{($\neg$:R)}\frac{\Gamma, A \longrightarrow \Delta}{\Gamma \longrightarrow \Delta, \neg A}$$

$$\text{($\wedge$:L)}\frac{\Gamma, A, B \longrightarrow \Delta}{\Gamma, A \wedge B \longrightarrow \Delta} \qquad \text{($\wedge$:R)}\frac{\Gamma \longrightarrow \Delta, A \qquad \Gamma \longrightarrow \Delta, B}{\Gamma \longrightarrow \Delta, A \wedge B}$$

$$\text{($\vee$:L)}\frac{\Gamma, A \longrightarrow \Delta \qquad \Gamma, B \longrightarrow \Delta}{\Gamma, A \vee B \longrightarrow \Delta} \qquad \text{($\vee$:R)}\frac{\Gamma \longrightarrow \Delta, A, B}{\Gamma \longrightarrow \Delta, A \vee B}$$

Sequent calculus is denoted LK (for Logischer Kalkülus) or PK for the propositional version.

Fact 3. $LK \equiv_p F$

Definition 4. LK^- is the subsystem of LK, which forbids the use of the cut rule.

Exercise 5. Prove $LK^- \vdash \longrightarrow A \vee \neg A$

Fact 6. LK^- is complete.

Exercise 7. Is LK^- implicationally complete?

Refutation systems

Definition 8. We call a poly-time predicate $R(x, y)$ a refutation system iff for any propositional formula ϕ there is π_ϕ such that

$$R(\phi, \pi_\phi) \text{ accepts } \iff \pi \text{ is unsatisfiable.}$$

Exercise 9. Show that any refutation system is efficiently transformable into a proof system and vice versa.

Limited extension

Resolution is a refutation system which works with CNFs only. This may seem as a crucial flaw, since there are boolean formulas whose equivalent CNF formulas are of exponential size. The way to overcome this is to consider *equisatisfiability* instead of equivalence.

Definition 10. Two formulas φ and ψ are **equisatisfiable** iff

$$\phi \text{ is satisfiable } \iff \psi \text{ is satisfiable.}$$

Exercise 11. How hard is the problem of determining whether two given formulas are equivalent, or equisatisfiable?

Theorem 12 (Tseitin)**.** There is a poly-time algorithm L which on input a Boolean formula ϕ produces an equisatisfiable CNF formula $L(\phi)$.

Resolution

From now on we identify CNF formulas with set of *clauses* C_i, where each C_i is a set of *literals* l_j and a literal is either a variable or its negation.

Definition 13. Given a CNF $\mathcal{C} = \{C_1^{init}, ..., C_m^{init}\}$ a **resolution refutation** of $\mathcal{C}$ is a sequence of clauses $C_1, ..., C_k$ so that

- C_k is an empty set,

- each C_j is either an initial clause C_i^{init} or is derived from the previous clauses via the **resolution rule**

$$\frac{C \cup \{p\} \quad C' \cup \{\neg p\}}{C \cup C'}$$

For a general formula ϕ its resolution refutation is defined as a resolution refutation of $L(\phi)$.

Exercise 14. Show that resolution is a sound refutation system, i.e. if ϕ has a resolution refutation, then ϕ is unsatisfiable.

Exercise 15. Show that resolution is a complete refutation system, i.e. if ϕ is unsatisfiable, then there is a resolution refutation of ϕ.

Exercise 16. Conclude that resolution is a Cook–Reckhow proof system.

Exercise 17. Express the statement "there is a linear ordering on n elements with no least element" as a CNF. Derive a resolution refutation of this statement.

Exercise 18. Using resolution, show that 2-CNF-SAT is in **P**.