Propositional resolution

JR

Motivation

- sequent calculus with cut is a very strong system, but a cut-formula can be much more complicated than the theorem, so it is problematic to suggest a proof-search algorithm
- a similar problem arises in Hilbert-style systems with modus ponens
- propositional cut-free sequent calculus offers a rather straightforward way to search for proofs, but sizes of cut-free proofs can be unnecessarily long
- so we want to find a system
    1. whose proofs are not too long and
    2. which can search for proofs effectively

Resolution - basic notions

- a *literal* is a variable p_i or its negation (complement) $\overline{p_i}$,
  if x is a negated variable $\overline{p_i}$, define $\overline{x}$ to be p_i
- a *clause* is a finite set of literals, interpretated as a disjuction
- a clause is *positive* (*negative*) if it contains only positive (negative) literals, other clauses are *mixed*
- a non-empty set of clauses Γ represents the conjunction of its members
- Γ is *satisfiable* if there is a valuation making all its members true.

Convention

No clause contains a literal together with its complement.

Resolution - basic notions

Definition

For clauses C and D and literals $x \in C$ and $\overline{x} \in D$, the *resolution rule* is the following inference:

$$\frac{C \cup \{x\} \qquad D \cup \{\overline{x}\}}{C \cup D}$$

C and D are assumed not to contain x and $\overline{x}$. The set $C \cup D$ is the *resolvent* of C and D with respect to x.

Definition

A *resolution refutation* of a set of clauses Γ is a derivation of the empty clause $\emptyset$ from Γ using only the resolution rule.

Notes

- resolution inference can be seen as cut on atomic formulas; for example if $C = \{p_1, p_2, \overline{p_3}\}$, $D = \{p_4, \overline{p_5}\}$ and the variable resolved on is p_6, the rule can be rewriten as

$$\frac{\overline{p_1}, \overline{p_2}, p_3 \to p_6 \qquad p_6 \to p_4, \overline{p_5}}{\overline{p_1}, \overline{p_2}, p_3 \to p_4, \overline{p_5}}$$

- if there is a valuation satisfying the clause $C \cup D$, the same valuation also satisfies the resolvent of C and D
- so if there is a derivation of $\emptyset$ from Γ, then Γ is not satisfiable
- so the principal interpretation of resolution is that we try to *refute* the satisfiability of Γ

Resolution - a proof method

To *prove* a formula φ means to construct a set of clauses Γ_φ such that Γ_φ is not satisfiable iff φ is a tautology. Then it suffices to refute Γ_φ.

Such a Γ_φ can be arrived at in two ways:

1. convert $\neg\varphi$ into CNF and take Γ_φ to be the corresponding set of clauses; but the CNF of $\neg\varphi$ can be exponentially longer than φ
2. (Tsejtin's extension method) introduce new variables to denote subformulas of φ, encode the meaning of these variables by clauses, construct Γ_φ from these clauses together with $\{\overline{x_\varphi}\}$. Γ_φ has size linear to the size of φ, it corresponds to the negation of a formula equisatisfiable with φ but one with a different structure

Example

Prove the formula $p \wedge q \supset q \wedge p$:

- the CNF of the negated formula is $p \wedge q \wedge (\neg q \vee \neg p)$
- so the clauses are $\{p\}, \{q\}$ and $\{\overline{q}, \overline{p}\}$
- two applications of resolution yield the empty clause

$$\dfrac{p \qquad \dfrac{q \qquad \overline{q}, \overline{p}}{\overline{p}}}{\emptyset}$$

Completeness of resolution

Theorem

If Γ is an unsatisfiable set clauses, then there is a resolution refutation of Γ.

Buss sketches two different proofs, a direct proof based on the David-Putnam procedure and an indirect one that reduces to completeness of the free-cut free sequent calculus.

Sketch of the first proof

- by compactness we may work with a finite Γ and use induction on the number of distinct variables in Γ
- for $n = 0$ we must have $\{\emptyset\} \in \Gamma$
- for a fixed p from Γ, define Γ' to contain the following clauses
  (i) the resolvents of all C, D from Γ such that $p \in C$ and $\overline{p} \in D$
  (ii) every $C \in \Gamma$ such that C contains neither p nor $\overline{p}$
- by the above convention no clause in Γ' contains p
- now prove that Γ is satisfiable iff Γ' is, by IH this concludes the proof

Sketch of the second proof

- clauses can be identified with sequents consisting of atomic formulas only and a cut inference with all three sequents consisitng of atoms only can be identified with a resolution inference
- example: the clause $\{p_1, p_2, \overline{p_3}\}$ translates as $p_3 \rightarrow p_1, p_2$
- given Γ, for any $C \in \Gamma$ denote by Π_C (Δ_C) the cedent consisting of variables that occur negatively (positively) in C; then the sequents $G = \{\Pi_C \rightarrow \Delta_C \; ; C \in \Gamma\}$ form the additional non-logical axioms

Let Γ be unsatisfiable. By the completeness of free-cut free sequent calculus there is a free-cut free proof P of the empty sequent from G. Every cut-formula in P must be atomic, and hence so is every formula in P. So P can be translated as a resolution refutation of Γ.

Restricted resolution systems

- searching for refutations in restricted systems of resolution requires less space, in one way or another they restrict the number of possible search paths (and/or clauses) that need to be considered when trying to refute a formula
- they can also be lifted to first-order logic

Example

- if Γ contains a clause C with a (*pure*) literal x such that $\overline{x}$ does not occur anywhere in Γ, we may discard C and repeat this process (this may give rise to new pure literals)
- the convention that clauses containing complementary litarals are not assumed can be rephrased as a rule to begin each proof-search - first delete tautological clauses

Subsumption

C *subsumes* D if $C \subseteq D$. The reason for this definition lies in the following theorem which states that the removal of subsumed clauses from an unsatisfiable set preserves the unsatisfiability.

Theorem

If Γ is not satisfiable and $C \subseteq D$, then $\Gamma' = (\Gamma \setminus \{D\}) \cup \{C\}$ is also unsatisfiable and has a refutation which is no longer than the shortest refutation of Γ.

Positive resolution

A resolution inference is *positive* if one of the premises is a positive clause.

Theorem (Completeness)

If Γ *is not satisfiable, it has a refutation with only positive resolution inferences.*

Positive resolution is an important stepping stone for hyperresolution.

Hyperresolution

- multiple resolution inferences are combined into a single one with positive conclusion
- justification: every positive resolution refutation can be uniquely partitioned into subderivations of the form

$$\dfrac{A_2 \qquad \dfrac{A_1 \qquad B_1}{B_2}}{B_3}$$

$$\cdot^{\cdot^{\cdot}}$$

$$\dfrac{A_n \qquad\qquad B_n}{A_{n+1}}$$

where the clauses $A_1, \ldots, A_{n+1}$ are positive.

Hyperresolution

Such a subderivation induces the following hyperresolution inference:

$$\frac{A_1 \qquad A_2 \qquad \dots \qquad A_n \qquad B_1}{A_{n+1}}$$

Notes

- by the above theorem hyperresolution is complete
- its usefulness lies in that only positive clauses need to be saved for future use as possible premises.

Semantic resolution

- let v be a fixed valuation. A resolution inference is *v-supported* if v falsifies one its premises. A refutation P is *v-supported* if each resolution inference is v-supported
- if v_F assignes every variable the value 0, then a v_F-supported refutation is the same as positive resolution refutation
- conversely, given Γ and v we can form Γ' by complementing every variable which is assigned 1 by v. Then a v-supported refutation of Γ is isomorphic to a positive refutation of Γ'
- so semantic resolution can be viewed as a generalization of positive resolution and we have the following theorem:

Theorem

For any Γ and v, Γ is not satisfiable iff Γ has a v-supported refutation.

Set-of-support resolution

For $\Pi \subset \Gamma$, if $\Gamma \setminus \Pi$ is satisfiable, then Π is a *set of support* for Γ. A refutation P of Γ is *supported* by Π if every inference in P uses (possibly indirectly) at least one clause from Π.

Theorem

If Γ is not satisfiable and Π is a set of support for Γ, then Γ has a refutation supported by Π.

Proof.

This follows from the completeness theorem of semantic resolution. If v is any truth assignment that satisfies $\Gamma \setminus \Pi$, any v-supported refutation is also supported by Π. □

Contrary to semantic resolution, in set-of-support resolution we do not need to know a satisfying assignment for $\Gamma \setminus \Pi$.

Unit and input resolution

- a *unit clause* contains exactly one literal. A resolution inference is a *unit resolution inference* if at least of its premises is a unit clause. A *unit resolution refutation* is a refutation containing only unit resolutions
- if there is a unit clause $C = \{x\}$ in Γ, we can reduce the number and size of clauses in Γ by eliminating each clause which contains x (subsumption) and removing $\overline{x}$ from all other clauses; this preserves unsatisfiability
- an *input resolution refutation* of Γ is a refutation in which every inference has a premise from Γ
- unit and input refutations are not complete and refute exactly the same sets

Linear resolution

- a *linear resolution refutation* is a sequence $A_1, \ldots, A_n = \emptyset$ such that each A_i is either from Γ or it is the conclusion of A_{i-1} and A_j for $j < i - 1$
- it is a generalization of input resolution, it allows to use intermediate clauses which are not in Γ multiple times
- linear resolution is complete, every unsatisfiable Γ has a linear refutation

Horn clauses

- a *Horn* clause contains at most one positive literal; deciding the satisfiability of sets of Horn clauses is more effective than deciding the satisfiability of arbitrary clauses
- a *positive unit* resolution inference is one whose one premise is a unit clause containing a positive literal, a *positive unit* refutation contains only positive unit resolution inferences

Theorem (Completeness)

Every unsatisfiable set of Horn clauses Γ has a positive unit refutation.

Proof.

Γ must contain a positive unit clause $\{p\}$. Resolve $\{p\}$ against all other clauses containing $\overline{p}$ and remove all clauses containing $\overline{p}$ or p. This operation yields a smaller unsatisfiable set of Horn clauses and its iteration yields the empty clause. □