

Propositional Logic Resolution

Resolution — The idea

Input: Set of clauses F

Question: Is F unsatisfiable?

Algorithm:

Keep on “resolving” two clauses from F and adding the result to F until the empty clause is found

Correctness:

If the empty clause is found, the initial F is unsatisfiable

Completeness:

If the initial F is unsatisfiable, the empty clause can be found.

Correctness/Completeness of **syntactic** procedure (**resolution**)
w.r.t. **semantic** property (**unsatisfiability**)

Resolvent

Definition

Let L be a literal. Then \bar{L} is defined as follows:

$$\bar{L} = \begin{cases} \neg A_i & \text{if } L = A_i \\ A_i & \text{if } L = \neg A_i \end{cases}$$

Definition

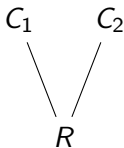
Let C_1, C_2 be clauses and let L be a literal such that $L \in C_1$ and $\bar{L} \in C_2$. Then the clause

$$(C_1 - \{L\}) \cup (C_2 - \{\bar{L}\})$$

is a **resolvent** of C_1 and C_2 .

The process of deriving the resolvent is called a **resolution step**.

Graphical representation of resolvent:



If $C_1 = \{L\}$ and $C_2 = \{\bar{L}\}$ then the empty clause is a resolvent of C_1 and C_2 . The special symbol \square denotes the empty clause.

Recall: \square represents \perp .

Resolution proof

Definition

A **resolution proof** of a clause C from a set of clauses F is a sequence of clauses C_0, \dots, C_n such that

- ▶ $C_i \in F$ or C_i is a resolvent of two clauses C_a and C_b , $a, b < i$,
- ▶ $C_n = C$

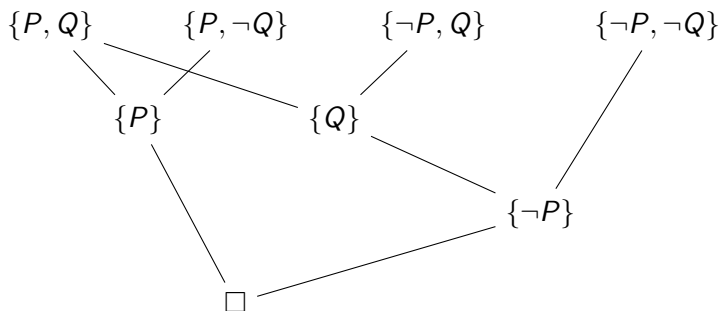
Then we can write $F \vdash_{Res} C$.

Note: F can be finite or infinite!

Resolution proof as DAG

A resolution proof can be shown as a DAG with the clauses in F as the leaves and C as the root:

Example



A linear resolution proof

0: $\{P, Q\}$

1: $\{P, \neg Q\}$

2: $\{\neg P, Q\}$

3: $\{\neg P, \neg Q\}$

4: $\{P\}$ (0, 1)

5: $\{Q\}$ (0, 2)

6: $\{\neg P\}$ (3, 5)

7: \square (4, 6)

Correctness of resolution

Lemma (Resolution Lemma)

Let R be a resolvent of two clauses C_1 and C_2 . Then $C_1, C_2 \models R$.

Proof By definition $R = (C_1 - \{L\}) \cup (C_2 - \{\bar{L}\})$ (for some L).

Assume $\mathcal{A} \models C_1$ and $\mathcal{A} \models C_2$. We show $\mathcal{A} \models R$.

There are two cases:

- ▶ $\mathcal{A} \models L$. Then $\mathcal{A} \models C_2 - \{\bar{L}\}$ (because $\mathcal{A} \models C_2$), thus $\mathcal{A} \models R$.
- ▶ $\mathcal{A} \not\models L$. Then $\mathcal{A} \models C_1 - \{L\}$ (because $\mathcal{A} \models C_1$), thus $\mathcal{A} \models R$.

Correctness of resolution

Theorem (Correctness of resolution)

Let F be a set of clauses. *If $F \vdash_{Res} C$ then $F \models C$.*

Proof Assume there is a resolution proof $C_0, \dots, C_n = C$. We show $F \models C_i$ by induction on i . IH: $F \models C_j$ for all $j < i$. There are two cases:

- ▶ $C_i \in F$.
Then $F \models C_i$ by definition.
- ▶ C_i is a resolvent of C_a and C_b for $a, b < i$.
Then $F \models C_a$ and $F \models C_b$ by IH, and $C_a, C_b \models C_i$ by the resolution lemma. Thus $F \models C_i$.

Corollary

Let F be a set of clauses. *If $F \vdash_{Res} \square$ then F is unsatisfiable.*

Completeness of resolution

Theorem

Let F be a finite set of clauses. If F is unsatisfiable then $F \vdash_{Res} \square$.

Theorem (Completeness of resolution)

Let F be a set of clauses. If F is unsatisfiable then $F \vdash_{Res} \square$.

Proof If F is infinite, there must be a finite unsatisfiable subset of F (by the Compactness Theorem); in that case let F be that finite subset and apply the previous theorem.

Corollary

A set of clauses F is unsatisfiable iff $F \vdash_{Res} \square$.

Completeness proof

Corollary

(of the Boole-Shannon expansion) F is unsatisfiable iff $F[\perp/A]$ and $F[\top/A]$ are unsatisfiable.

Idea for completeness proof:

If A is an atom of F , then both $F[\perp/A]$ and $F[\top/A]$ have fewer atoms than F .

Use Boole-Shannon to prove completeness by induction on the number of atoms of the unsatisfiable formula F :

- ▶ construct inductively resolution proofs for $F[\perp/A]$ and $F[\top/A]$, and
- ▶ “combine” them into a resolution proof for F .

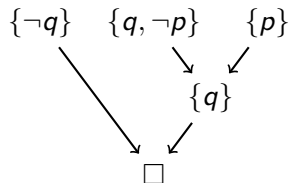
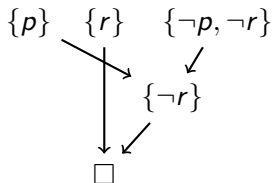
Inductive construction of resolution proofs

$$F = \{ \{\neg q, s\}, \{\neg p, q, s\}, \{p\}, \{r, \neg s\}, \{\neg p, \neg r, \neg s\} \}$$

- Compute inductively proofs for $F[\top/s]$ and $F[\perp/s]$.

$$F[\top/s] \equiv \{ \{p\}, \{r\}, \{\neg p, \neg r\} \}$$

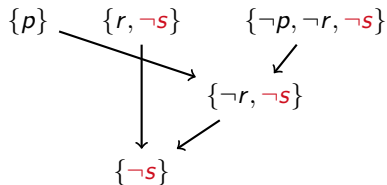
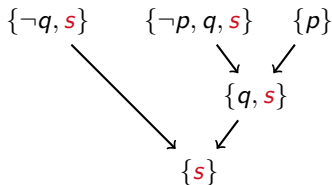
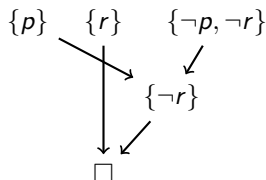
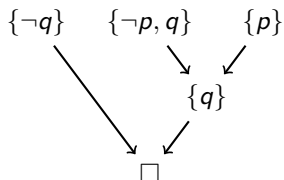
$$F[\perp/s] \equiv \{ \{\neg q\}, \{\neg p, q\}, \{p\} \}$$



Inductive construction of resolution proofs

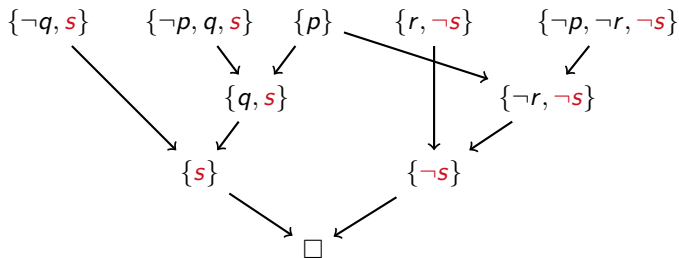
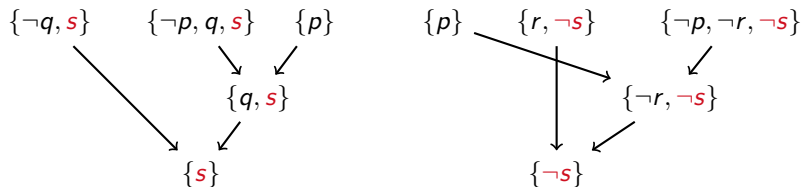
- ▶ Reintroduce s and $\neg s$.

$$F = \{ \{\neg q, s\}, \{\neg p, q, s\}, \{p\}, \{r, \neg s\}, \{\neg p, \neg r, \neg s\} \}$$



Inductive construction of resolution proofs

- Combine the graphs for $\{s\}$ and $\{\neg s\}$.



Completeness proof

Theorem

Let F be a finite set of clauses. *If F is unsatisfiable then $F \vdash_{Res} \square$.*

Proof By induction on the number n of distinct atoms in F .

Basis: If $n = 0$ then $F = \{\}$ (but F is unsat.) or $F = \{\square\}$.

Step:

IH: For every unsat. set of clauses F with n dist. atoms, $F \vdash_{Res} \square$.

Let F contain $n + 1$ distinct atoms. Pick some atom A in F .

$F[\top/A] \equiv$ take F , remove all clauses with A , remove all $\neg A$.

$F[\perp/A] \equiv$ take F , remove all clauses with $\neg A$, remove all A .

Completeness proof

By IH: there are res. proofs $C_0, \dots, C_m = \square$ from $F[\perp/A]$ and $D_0, \dots, D_n = \square$ from $F[\top/A]$.

Now transform C_0, \dots, C_m into a proof C'_0, \dots, C'_m from F by adding A back into the clauses it was removed from. Then:

- ▶ either $C'_m = \{A\}$
- ▶ or $C'_m = \square$ (and we are done).

Similarly we transform D_0, \dots, D_n into a proof D'_0, \dots, D'_n from F by adding $\neg A$ back in. Then:

- ▶ either $D'_n = \{\neg A\}$
- ▶ or $D'_n = \square$ (and we are done).

If $C'_m = \{A\}$ and $D'_n = \{\neg A\}$ then $F \vdash_{Res} A$ and $F \vdash_{Res} \neg A$ and thus $F \vdash_{Res} \square$.

Resolution is only refutation complete

Not everything that is a consequence of a set of clauses
can be derived by resolution.

Exercise

Find F and C such that $F \models C$ but not $F \vdash_{Res} C$.

How to prove $F \models C$ by resolution?

Prove $F \cup \{\neg C\} \vdash_{Res} \square$

A resolution algorithm

Input: A CNF formula F , i.e. a finite set of clauses

while there are clauses $C_a, C_b \in F$ and resolvent R of C_a and C_b
such that $R \notin F$
do $F := F \cup \{R\}$

Lemma

The algorithm terminates.

Proof There are only finitely many clauses over a finite set of atoms.

Theorem

The initial F is unsatisfiable iff \square is in the final F

Proof F_{init} is unsat. iff $F_{init} \vdash_{Res} \square$ iff $\square \in F_{final}$ because the algorithm enumerates all R such that $F_{init} \vdash_{Res} R$.

The algorithm is a decision procedure for unsat. of CNF formulas.