Resolution

For every formula F:

$$(F \vee A) \wedge (F' \vee \neg A) \equiv (F \vee A) \wedge (F' \vee \neg A) \wedge (F \vee F')$$

Or in clause form

$$\{\ \{♣, A\}\ ,\ \{♠, \neg A\}\ \}\ \equiv\ \{\ \{♣, A\}\ ,\ \{♠, \neg A\}\ ,\ \{♣, ♠\}\ \}$$

If $F \vee F'$ is the "empty disjunction" ($=$ empty clause) then the formula is unsatisfiable.

- Is it always possible to derive the empty clause from any unsatisfiable formula?
  (completeness)
- Can we represent derivations in more compact form?
  (without carrying always all clauses around)

Clause representation of CNF formulas

- Clause: set of literals (disjunction).
  $\{A, B\}$ stands for $A \vee B$.
- Formula: set of clauses (conjunction).
  $\{\{A, B\}, \{\neg A, B\}\}$ stands for $(A \vee B) \wedge (\neg A \vee B)$.
- Block: set of formulas (disjunction).
  $\{F, G\}$ stands for $F \vee G$.

The empty clause stands for **false** or 0.

The empty formula stands for **true** or 1.

The empty block stands for **false**.

Advantages of the clause form

We get "for free":

- Commutativity:
  $A \vee B \equiv B \vee A$, both represented by $\{A, B\}$
- Associativity:
  $(A \vee B) \vee C \equiv A \vee (B \vee C)$, both represented by $\{A, B, C\}$
- Idempotence:
  $(A \vee A) \equiv A$, both represented by $\{A\}$

Resolvent (I)

Definition: Let C_1, C_2 and R be clauses. R is a resolvent of C_1 and C_2 if there is a literal L such that $L \in C_1$, $\overline{L} \in C_2$ and

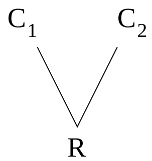
$$R = (C_1 - \{L\}) \cup (C_2 - \{\overline{L}\})$$

where $\overline{L}$ is defined by

$$\overline{L} = \begin{cases} \neg A_i & \text{falls } L = A_i \\ A_i & \text{falls } L = \neg A_i \end{cases}$$

Resolvent (II)

Graphical representation:



If $C_1 = \{L\}$ and $C_2 = \{\overline{L}\}$ then the empty clause is a resolvent of C_1 and C_2. We represent it with the special symbol $\square$.

Recall: $\square \equiv \textbf{false}$.

Resolution Lemma

Resolution Lemma: Let F be a formula in **CNF**, represented as a set of clauses, and let R be a resolvent of two clauses C_1 and C_2 in F. Then the formulas F and $F \cup \{R\}$ are equivalent.

Proof: Follows immediately from

$$\underbrace{(F_1 \vee A)}_{C_1} \wedge \underbrace{(F_2 \vee \neg A)}_{C_2} \equiv \underbrace{(F_1 \vee A)}_{C_1} \wedge \underbrace{(C_2 \vee \neg A)}_{C_2} \wedge \underbrace{(F_1 \vee F_2)}_{R}$$

Resolution calculus

A calculus is a set of syntactic transformation rules allowing to decide semantic properties.

- Syntactic rules: resolution, halt when the empty clause is derived.
- Semantic property: unsatisfiabilty.

Example

We wish to prove that

$$((AB \vee BB) \wedge (AB \rightarrow BB) \wedge (BB \wedge RL \rightarrow \neg AB) \wedge RL) \rightarrow (\neg AB \wedge BB)$$

is valid. This is the case iff

$$(AB \vee BB) \wedge (\neg AB \vee BB) \wedge (\neg BB \vee \neg RL \vee \neg AB) \wedge RL \wedge (AB \vee \neg BB)$$

is unsatisfiable. (Recall: $F \rightarrow G$ valid iff $F \wedge \neg G$ unsatisfiable.)

Desirable properties of a calculus

- Correctness (or consistency): If the application of the syntactic rules say that the semantic property holds, then this is indeed the case.
  If the empty clause can be derived from F then F is unsatisfiable.
- Completeness: If the semantic property holds, then this can be shown with the help of the syntactic rules.
  If F is unsatisfiable then the empty clause can be derived from F.

Definition of $Res(F)$

Definition: Let F be a set of clauses. The formula $Res(F)$ is defined as follows:

$$Res(F) = F \cup \{R \mid R \text{ ist a resolvent of two clauses in } F\}.$$

Furthermore, define

$$\begin{aligned} Res^0(F) &= F \\ Res^{n+1}(F) &= Res(Res^n(F)) \qquad \text{für } n \geq 0 \end{aligned}$$

and finally let

$$Res^*(F) = \bigcup_{n \geq 0} Res^n(F).$$

Exercise

Assume n atomic formulas occur in F. Then:

A	$\lvert Res^*(F)\rvert \leq 2^n$	**B**	$\lvert Res^*(F)\rvert \leq 4^n$
C	$\lvert Res^*(F)\rvert$ can be arbitrarily large		

Resolution Theorem

We prove that resolution is correct and complete:

Resolution Theorem (of propositional logic):
A set of clauses F is unsatisfiable iff $\Box \in Res^*(F)$.

Correctness: $\Box \in Res^*(F) \Rightarrow F$ is unsatisfiable follows immediately from the resolution lemma.

Completeness proof (I)

Completeness: F is unsatisfiable $\Rightarrow \Box \in Res^*(F)$
By induction on the number of atomic formulas in F.

Here: Induction step with $n+1=4$

$$F = \{\{A_1\},\{\neg A_2, A_4\},\{\neg A_1, A_2, A_4\},\{A_3,\neg A_4\},\{\neg A_1,\neg A_3,\neg A_4\}\}$$

Completeness proof (I)

Completeness: F is unsatisfiable $\Rightarrow \Box \in Res^*(F)$
By induction on the number of atomic formulas in F.

Here: Induction step with $n+1=4$

$$F = \{\{A_1\},\{\neg A_2, A_4\},\{\neg A_1, A_2, A_4\},\{A_3,\neg A_4\},\{\neg A_1,\neg A_3,\neg A_4\}\}$$

$$F_0 = \{\{A_1\},\{\neg A_2\},\{\neg A_1, A_2\}\}$$

Completeness proof (I)

Completeness: F is unsatisfiable $\Rightarrow \Box \in Res^*(F)$
By induction on the number of atomic formulas in F.

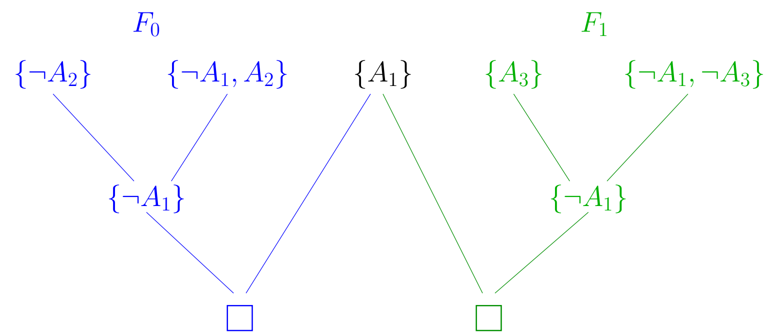
Here: Induction step with $n+1=4$

$$F = \{\{A_1\},\{\neg A_2, A_4\},\{\neg A_1, A_2, A_4\},\{A_3,\neg A_4\},\{\neg A_1,\neg A_3,\neg A_4\}\}$$

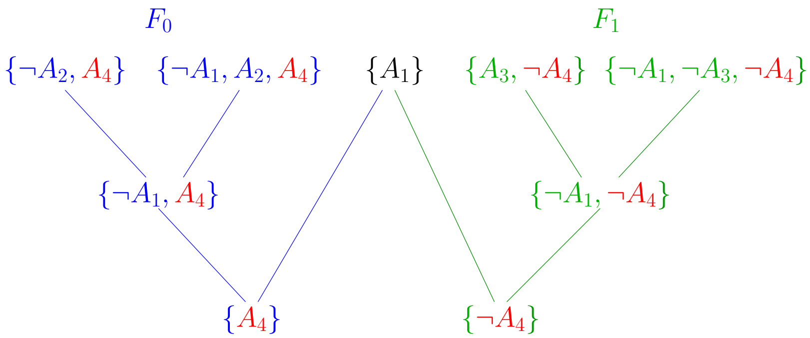
$$F_0 = \{\{A_1\},\{\neg A_2\},\{\neg A_1, A_2\}\}$$

$$F_1 = \{\{A_1\},\{A_3\},\{\neg A_1,\neg A_3\}\}$$

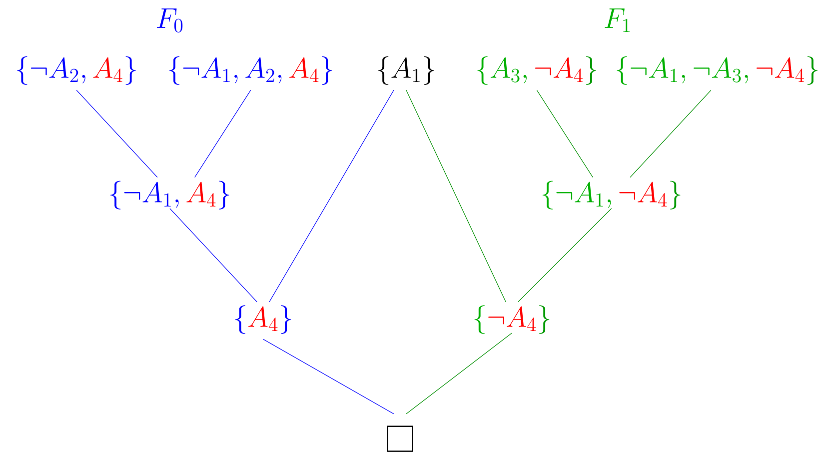
Completeness proof (II)



Completeness proof (II)



Completeness proof (II)



Definition

A derivation (or proof) of the empty clause from a set F of clauses is a sequence $C_1, C_2, \ldots, C_m$ of clauses such that:

> C_m is the empty clause and for every $i = 1, \ldots, m$ it holds that C_i is either a clause in F or a resolvent of two clauses C_a,C_b with $a, b < i$.

F is unsatisfiable iff a derivation of the empty clause from F exists.