# Automated Theorem Proving

## Lecture 5: Resolution

**Prof. Dr. Jasmin Blanchette**

**based on slides by Dr. Uwe Waldmann**

**Winter Term 2024/25**

# 3.5 Normal Forms and Skolemization

Study of normal forms motivated by

- reduction of logical concepts,
- efficient data structures for theorem proving.

The main problem in first-order logic is the treatment of quantifiers. The subsequent normal form transformations are intended to eliminate many of them.

# Prenex Normal Form (Traditional)

Prenex formulas have the form

$$\mathrm{Q}_1 x_1 \ldots \mathrm{Q}_n x_n \; F,$$

where $F$ is quantifier-free and $\mathrm{Q}_i \in \{\forall, \exists\}$;
we call $\mathrm{Q}_1 x_1 \ldots \mathrm{Q}_n x_n$ the quantifier prefix and $F$ the matrix of the formula.

# Prenex Normal Form (Traditional)

Computing prenex normal form by the reduction system $\Rightarrow_P$:

$$
\begin{array}{rcl}
H[(F \leftrightarrow G)]_p & \Rightarrow_P & H[(F \to G) \wedge (G \to F)]_p \\
H[\neg \mathrm{Q} x\, F]_p & \Rightarrow_P & H[\overline{\mathrm{Q}} x\, \neg F]_p \\
H[((\mathrm{Q} x\, F) \circ G)]_p & \Rightarrow_P & H[\mathrm{Q} y\, (F\{x \mapsto y\} \circ G)]_p, \\
 & & \circ \in \{\wedge, \vee\} \\
H[((\mathrm{Q} x\, F) \to G)]_p & \Rightarrow_P & H[\overline{\mathrm{Q}} y\, (F\{x \mapsto y\} \to G)]_p, \\
H[(F \circ (\mathrm{Q} x\, G))]_p & \Rightarrow_P & H[\mathrm{Q} y\, (F \circ G\{x \mapsto y\})]_p, \\
 & & \circ \in \{\wedge, \vee, \to\}
\end{array}
$$

Here $y$ is always assumed to be some fresh variable and $\overline{\mathrm{Q}}$ denotes the quantifier dual to Q, i.e., $\overline{\forall} = \exists$ and $\overline{\exists} = \forall$.

# Skolemization

**Intuition:** replacement of $\exists y$ by a concrete choice function computing $y$ from all the arguments $y$ depends on.

Transformation $\Rightarrow_S$
(to be applied outermost, *not* in subformulas):

$$\forall x_1, \ldots, x_n \exists y\, F \quad \Rightarrow_S \quad \forall x_1, \ldots, x_n\, F\{y \mapsto f(x_1, \ldots, x_n)\}$$

where $f/n$ is a new function symbol (Skolem function).

# Skolemization

**Together:** $F \Rightarrow_P^* \underbrace{G}_{\text{prenex}} \Rightarrow_S^* \underbrace{H}_{\text{prenex, no } \exists}$

Theorem 3.5.1:
Let $F$, $G$, and $H$ as defined above and closed. Then

(i) $F$ and $G$ are equivalent.

(ii) $H \models G$ but the converse is not true in general.

(iii) $G$ satisfiable (w.r.t. $\Sigma$-Alg) $\Leftrightarrow$ $H$ satisfiable (w.r.t. $\Sigma'$-Alg)
where $\Sigma' = (\Omega \cup SKF, \Pi)$ if $\Sigma = (\Omega, \Pi)$.

# The Complete Picture

$$
\begin{aligned}
F \quad &\Rightarrow_P^* \quad Q_1 y_1 \ldots Q_n y_n \, G && (G \text{ quantifier-free}) \\
&\Rightarrow_S^* \quad \forall x_1, \ldots, x_m \, H && (m \leq n,\ H \text{ quantifier-free}) \\
&\Rightarrow_{CNF}^* \quad \underbrace{\underbrace{\forall x_1, \ldots, x_m}_{\text{leave out}} \bigwedge_{i=1}^{k} \underbrace{\bigvee_{j=1}^{n_i} L_{ij}}_{\text{clauses } C_i}}_{F'}
\end{aligned}
$$

$N = \{C_1, \ldots, C_k\}$ is called the clausal (normal) form of $F$.

Note: The variables in the clauses are implicitly universally quantified.

# The Complete Picture

Theorem 3.5.2:

Let $F$ be closed. Then $F' \models F$.

(The converse is not true in general.)

Theorem 3.5.3:

Let $F$ be closed. Then $F$ is satisfiable if and only if $F'$ is satisfiable if and only if $N$ is satisfiable.

# The Complete Picture

Example 3.5.4:

We clausify $\neg\exists x\,(\forall y\,(P(x,y) \vee Q(y,x)))$:

$$
\begin{aligned}
& \neg\exists x\,(\forall y\,(P(x,y) \vee Q(y,x))) \\
\Rightarrow_P\ & \forall x\,(\neg\forall y\,(P(x,y) \vee Q(y,x))) \\
\Rightarrow_P\ & \forall x\,\exists y\,(\neg(P(x,y) \vee Q(y,x))) \\
\Rightarrow_S\ & \forall x\,(\neg(P(x,f_1(x)) \vee Q(f_1(x),x))) \\
\Rightarrow_{CNF}\ & \forall x\,(\neg P(x,f_1(x)) \wedge \neg Q(f_1(x),x))
\end{aligned}
$$

Thus $N = \{\neg P(x,f_1(x)),\ Q(f_1(x),x)\}$.

# Optimization

The normal form algorithm described so far leaves lots of room for optimization. Note that we only can preserve satisfiability anyway due to Skolemization.

- the size of the clausal normal form is exponential when done naively; the transformations we already introduced for propositional logic avoid this exponential growth;
- we want to preserve the original formula structure;
- we want small arity of Skolem functions.

See Nonnengart and Weidenbach 2001 for details.

# 3.6 Herbrand Interpretations

From now on we will consider first-order logic without equality.
We assume that $\Omega$ contains at least one constant symbol.

An Herbrand interpretation (over $\Sigma$) is a $\Sigma$-algebra $\mathcal{A}$ such that

- $U_{\mathcal{A}} = \mathrm{T}_{\Sigma}$ (= the set of ground terms over $\Sigma$)
- $f_{\mathcal{A}} : (s_1, \ldots, s_n) \mapsto f(s_1, \ldots, s_n),\ \ f/n \in \Omega$

In other words, *values are fixed* to be ground terms and *functions are fixed* to be the term constructors. Only predicate symbols $P/m \in \Pi$ may be freely interpreted as relations $P_{\mathcal{A}} \subseteq \mathrm{T}_{\Sigma}^{m}$.

# Herbrand Interpretations

Proposition 3.6.1:

Every set of ground atoms $I$ uniquely determines an Herbrand interpretation $\mathcal{A}$ via

$$(s_1, \ldots, s_n) \in P_{\mathcal{A}} \quad \text{if and only if} \quad P(s_1, \ldots, s_n) \in I$$

Thus we will identify Herbrand interpretations (over $\Sigma$) with sets of $\Sigma$-ground atoms.

# Existence of Herbrand Models

An Herbrand interpretation $I$ is called an Herbrand model of $F$ if $I \models F$.

The importance of Herbrand models lies in the following theorem, which we will prove later in this lecture:

Let $N$ be a set of (universally quantified) $\Sigma$-clauses. Then the following properties are equivalent:

(1) $N$ has a model.
(2) $N$ has an Herbrand model (over $\Sigma$).
(3) $G_\Sigma(N)$ has an Herbrand model (over $\Sigma$).

where $G_\Sigma(N) = \{C\sigma \text{ ground clause} \mid (\forall \vec{x}\, C) \in N,\ \sigma : X \to \mathsf{T}_\Sigma\}$ is the set of ground instances of $N$.

# 3.7 Inference Systems and Proofs

Inference systems $\Gamma$ (proof calculi) are sets of tuples

$$(F_1, \ldots, F_n, F_{n+1}), \ \ n \geq 0,$$

called inferences, and written

$$\frac{\overbrace{F_1 \quad \cdots \quad F_n}^{\text{premises}}}{\underbrace{F_{n+1}}_{\text{conclusion}}} \quad \textit{side condition}$$

Clausal inference system: Premises and conclusions are clauses. One also considers inference systems over other data structures.

# Inference Systems

Inference systems $\Gamma$ are shorthands for reduction systems over sets of formulas. If $N$ is a set of formulas, then

$$\frac{\overbrace{F_1 \quad \cdots \quad F_n}^{\text{premises}}}{\underbrace{F_{n+1}}_{\text{conclusion}}} \quad \textit{side condition}$$

is a shorthand for

$$N \cup \{F_1, \ldots, F_n\} \quad \Rightarrow_\Gamma \quad N \cup \{F_1, \ldots, F_n\} \cup \{F_{n+1}\}$$
$$\text{if } \textit{side condition}$$

# Proofs

A proof in $\Gamma$ of a formula $F$ from a set of formulas $N$ (called assumptions) is a sequence $F_1, \ldots, F_k$ of formulas where

(i) $F_k = F$,

(ii) for all $1 \leq i \leq k$: $F_i \in N$ or there exists an inference

$$\frac{F_{m_1} \quad \cdots \quad F_{m_n}}{F_i}$$

in $\Gamma$, such that $0 \leq m_j < i$, for $1 \leq j \leq n$.

# Soundness and Completeness

Provability $\vdash_\Gamma$ of $F$ from $N$ in $\Gamma$:

$$N \vdash_\Gamma F \text{ if there exists a proof in } \Gamma \text{ of } F \text{ from } N.$$

$\Gamma$ is called sound if

$$\frac{F_1 \quad \cdots \quad F_n}{F} \in \Gamma \text{ implies } F_1, \ldots, F_n \models F$$

$\Gamma$ is called complete if

$$N \models F \text{ implies } N \vdash_\Gamma F$$

$\Gamma$ is called refutationally complete if

$$N \models \bot \text{ implies } N \vdash_\Gamma \bot$$

# Soundness and Completeness

Proposition 3.7.1:

(i) Let $\Gamma$ be sound. Then $N \vdash_\Gamma F \Rightarrow N \models F$.

(ii) If $N \vdash_\Gamma F$ then there exist finitely many $F_1, \ldots, F_n \in N$ such that $F_1, \ldots, F_n \vdash_\Gamma F$.

# Reduced Proofs

The definition of a proof of $F$ given above admits sequences $F_1, \ldots, F_k$ of formulas where some $F_i$ are not ancestors of $F_k = F$ (i.e., some $F_i$ are not actually used to derive $F$).

A proof is called reduced if every $F_i$ with $i < k$ is an ancestor of $F_k$.

We obtain a reduced proof from a proof by marking first $F_k$ and then recursively all the premises used to derive a marked conclusion, and by deleting all nonmarked formulas in the end.

# Reduced Proofs as Trees

markings = formulas

leaves = assumptions and axioms

other nodes = inferences: conclusion = parent node

premises = child nodes

$$\dfrac{\dfrac{P(f(c)) \lor Q(b) \quad \neg P(f(c)) \lor \neg P(f(c)) \lor Q(b)}{\neg P(f(c)) \lor Q(b) \lor Q(b)}}{\dfrac{P(f(c)) \lor Q(b) \qquad \neg P(f(c)) \lor Q(b)}{Q(b) \lor Q(b)}}$$

$$\dfrac{\dfrac{Q(b) \qquad \neg P(f(c)) \lor \neg Q(b)}{\neg P(f(c))} \qquad P(f(c))}{\bot}$$

# Mandatory vs. Admissible Inferences

It is useful to distinguish between two kinds of inferences:

- Mandatory (required) inferences:

  Must be performed to ensure refutational completeness.

  The fewer, the better.

- Optional (admissible) inferences:

  May be performed if useful.

We will first consider only mandatory inferences.

# 3.8 Ground (or Propositional) Resolution

We observe that propositional clauses and ground clauses are essentially the same, as long as we do not consider equational atoms.

In this section we deal only with ground clauses.

Unlike in Part 2 we admit duplicated literals in clauses,
i.e., we treat clauses as multisets of literals, not as sets.

# The Resolution Calculus *Res*

Resolution inference rule:

$$\frac{D \vee A \qquad C \vee \neg A}{D \vee C}$$

Terminology: $D \vee C$: resolvent; $A$: resolved atom

(Positive) factorization inference rule:

$$\frac{C \vee A \vee A}{C \vee A}$$

# The Resolution Calculus *Res*

These are schematic inference rules; for each substitution of the schematic variables $C$, $D$, and $A$, by ground clauses and ground atoms, respectively, we obtain an inference.

We treat "$\vee$" as associative and commutative, hence $A$ and $\neg A$ can occur anywhere in the clauses; moreover, when we write $C \vee A$, etc., this includes unit clauses, that is, $C = \bot$.

# An Example Refutation

| | | |
|---|---|---|
| 1 | $\neg P(f(c)) \vee \neg P(f(c)) \vee Q(b)$ | (given) |
| 2 | $P(f(c)) \vee Q(b)$ | (given) |
| 3 | $\neg P(g(b,c)) \vee \neg Q(b)$ | (given) |
| 4 | $P(g(b,c))$ | (given) |
| 5 | $\neg P(f(c)) \vee Q(b) \vee Q(b)$ | (Res. 2 into 1) |
| 6 | $\neg P(f(c)) \vee Q(b)$ | (Fact. 5) |
| 7 | $Q(b) \vee Q(b)$ | (Res. 2 into 6) |
| 8 | $Q(b)$ | (Fact. 7) |
| 9 | $\neg P(g(b,c))$ | (Res. 8 into 3) |
| 10 | $\bot$ | (Res. 4 into 9) |

# Soundness of Resolution

Theorem 3.8.1:
Ground first-order resolution is sound.

Note: In ground first-order logic we have (like in propositional logic):

1. $\mathcal{B} \models L_1 \vee \cdots \vee L_n$ if and only if there exists $i$: $\mathcal{B} \models L_i$.
2. $\mathcal{B} \models A$ or $\mathcal{B} \models \neg A$.

This does *not* hold for formulas with variables.

# 3.9 Refutational Completeness of Resolution

How to show refutational completeness of ground resolution:

- We have to show: $N \models \bot \;\Rightarrow\; N \vdash_{Res} \bot$,
  or equivalently: If $N \nvdash_{Res} \bot$, then $N$ has a model.
- Idea: Suppose that we have computed sufficiently many inferences (and not derived $\bot$).
- Now order the clauses in $N$ according to some appropriate ordering, inspect the clauses in ascending order, and construct a series of Herbrand interpretations.
- The limit interpretation can be shown to be a model of $N$.

# Closure of Clause Sets under *Res*

$$Res(N) = \{C \mid C \text{ is conclusion of an inference in } Res \text{ with premises in } N\}$$

$$Res^0(N) = N$$

$$Res^{n+1}(N) = Res(Res^n(N)) \cup Res^n(N), \text{ for } n \geq 0$$

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

$N$ is called saturated (w.r.t. resolution) if $Res(N) \subseteq N$.

# Closure of Clause Sets under *Res*

Proposition 3.9.1:

(i) $Res^*(N)$ is saturated.

(ii) *Res* is refutationally complete if and only if for each set $N$ of ground clauses:

$$N \models \bot \text{ implies } \bot \in Res^*(N)$$

# Orderings

Let $\succ$ be a strict partial ordering on $M$; let $M'$ be a multiset over $M$.

$a \in M'$ is called strictly maximal in $M'$ if there is no $b \in M' - \{a\}$ with $a \preceq b$.

The notions of maximal and strictly maximal elements coincide except that a maximal element can have duplicates, whereas a strictly maximal element cannot.

# Clause Orderings

1. We assume that $\succ$ is any fixed ordering on ground atoms that is *total* and *well-founded*. (There exist many such orderings, e.g., the length-based ordering on atoms when these are viewed as words over a suitable alphabet.)
2. Extend $\succ$ to an ordering $\succ_{\mathsf{L}}$ on ground literals:

$$
\begin{array}{llll}
A & \succ_{\mathsf{L}} & B & \text{if } A \succ B \\
A & \succ_{\mathsf{L}} & \neg B & \text{if } A \succ B \\
\neg A & \succ_{\mathsf{L}} & B & \text{if } A \succ B \\
\neg A & \succ_{\mathsf{L}} & \neg B & \text{if } A \succ B \\
\neg A & \succ_{\mathsf{L}} & A &
\end{array}
$$

3. Extend $\succ_{\mathsf{L}}$ to an ordering $\succ_{\mathsf{C}}$ on ground clauses:
   $\succ_{\mathsf{C}} = (\succ_{\mathsf{L}})_{\mathsf{mul}}$, the multiset extension of $\succ_{\mathsf{L}}$.

   *Notation:* $\succ$ also for $\succ_{\mathsf{L}}$ and $\succ_{\mathsf{C}}$.

# Example

Suppose $A_5 \succ A_4 \succ A_3 \succ A_2 \succ A_1 \succ A_0$. Then:

$$
\begin{aligned}
& A_0 \vee A_1 \\
\prec \quad & A_1 \vee A_1 \vee A_2 \\
\prec \quad & \neg A_1 \vee A_2 \\
\prec \quad & A_1 \vee \neg A_2 \\
\prec \quad & A_1 \vee \neg A_2 \vee \neg A_2 \\
\prec \quad & \neg A_1 \vee A_3 \vee A_4 \\
\prec \quad & A_3 \vee \neg A_4 \\
\prec \quad & A_1 \vee \neg A_5
\end{aligned}
$$

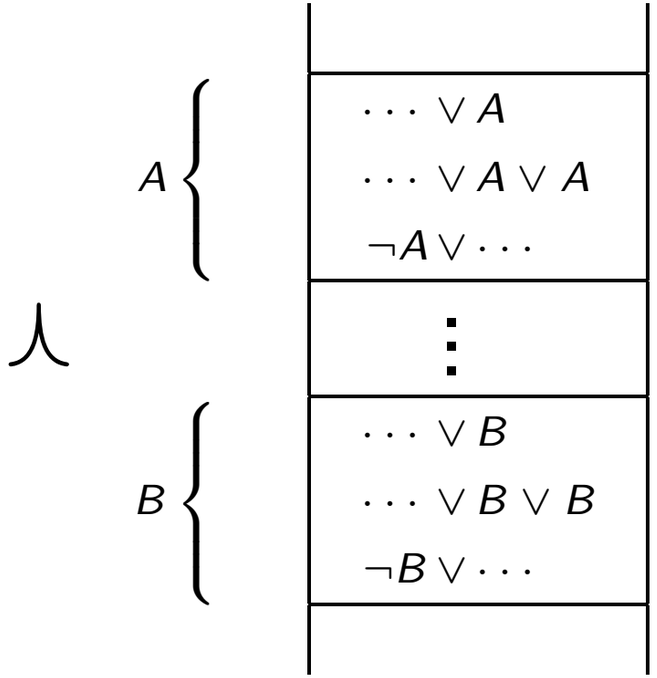
# Properties of the Clause Ordering

Proposition 3.9.2:

1. The orderings on literals and clauses are total and well-founded.
2. Let $C$ and $D$ be clauses with
   $A = \text{maxatom}(C)$, $B = \text{maxatom}(D)$,
   where maxatom($C$) denotes the maximal atom in $C$.
   (i) If $A \succ B$ then $C \succ D$.
   (ii) If $A = B$ and $A$ occurs negatively in $C$ but only positively in $D$, then $C \succ D$.

# Stratified Structure of Clause Sets

Let $B \succ A$. Clause sets are then stratified in this form:




all clauses $C$ with maxatom$(C) = A$

all clauses $C$ with maxatom$(C) = B$

# Construction of Interpretations

Given: set $N$ of ground clauses, atom ordering $\succ$.

Wanted: Herbrand interpretation $I$ such that

$$I \models N \quad \text{if } N \text{ is saturated and } \bot \notin N$$

Construction according to $\succ$, starting with the smallest clause.

# Main Ideas of the Construction

- Clauses are considered in the order given by $\succ$.
- When considering $C$, one already has an interpretation so far available ($I_C$). Initially $I_C = \emptyset$.
- If $C$ is true in this interpretation, nothing needs to to be changed.
- Otherwise, one would like to change the interpretation such that $C$ becomes true.

# Main Ideas of the Construction

- Changes should, however, be *monotone*. One never deletes atoms from the interpretation, and the truth value of clauses smaller than $C$ should not change from true to false.
- Hence, one adds $\Delta_C = \{A\}$ if and only if $C$ is false in $I_C$, if $A$ occurs positively in $C$ (*adding $A$ will make $C$ become true*) and if this occurrence in $C$ is strictly maximal in the ordering on literals (*changing the truth value of $A$ has no effect on smaller clauses*). Otherwise, $\Delta_C = \emptyset$.

# Main Ideas of the Construction

- We say that the construction fails for a clause $C$ if $C$ is false in $I_C$ and $\Delta_C = \emptyset$.
- We will show: If there are clauses for which the construction fails, then some inference with the smallest such clause (the so-called "minimal counterexample") has not been computed. Otherwise, the limit interpretation is a model of all clauses.

# Construction of Candidate Interpretations

Let $N, \succ$ be given. We define sets $I_C$ and $\Delta_C$ for all ground clauses $C$ over the given signature inductively over $\succ$:

$$I_C := \bigcup_{C \succ D} \Delta_D$$

$$\Delta_C := \begin{cases} \{A\}, & \text{if } C \in N,\ C = C' \vee A,\ A \succ C',\ I_C \not\models C \\ \emptyset, & \text{otherwise} \end{cases}$$

We say that $C$ produces $A$ if $\Delta_C = \{A\}$.

Note that the definitions satisfy the conditions of Thm. 1.3.7;
so they are well-defined even if $\{D \mid C \succ D\}$ is infinite.

# Construction of Candidate Interpretations

The candidate interpretation for $N$ (w.r.t. $\succ$) is given as $I_N^{\succ} := \bigcup_C \Delta_C$.
(We also simply write $I_N$ or $I$ for $I_N^{\succ}$ if $\succ$ is either irrelevant or known from the context.)

# Example

Let $A_5 \succ A_4 \succ A_3 \succ A_2 \succ A_1 \succ A_0$ (max. literals in red).

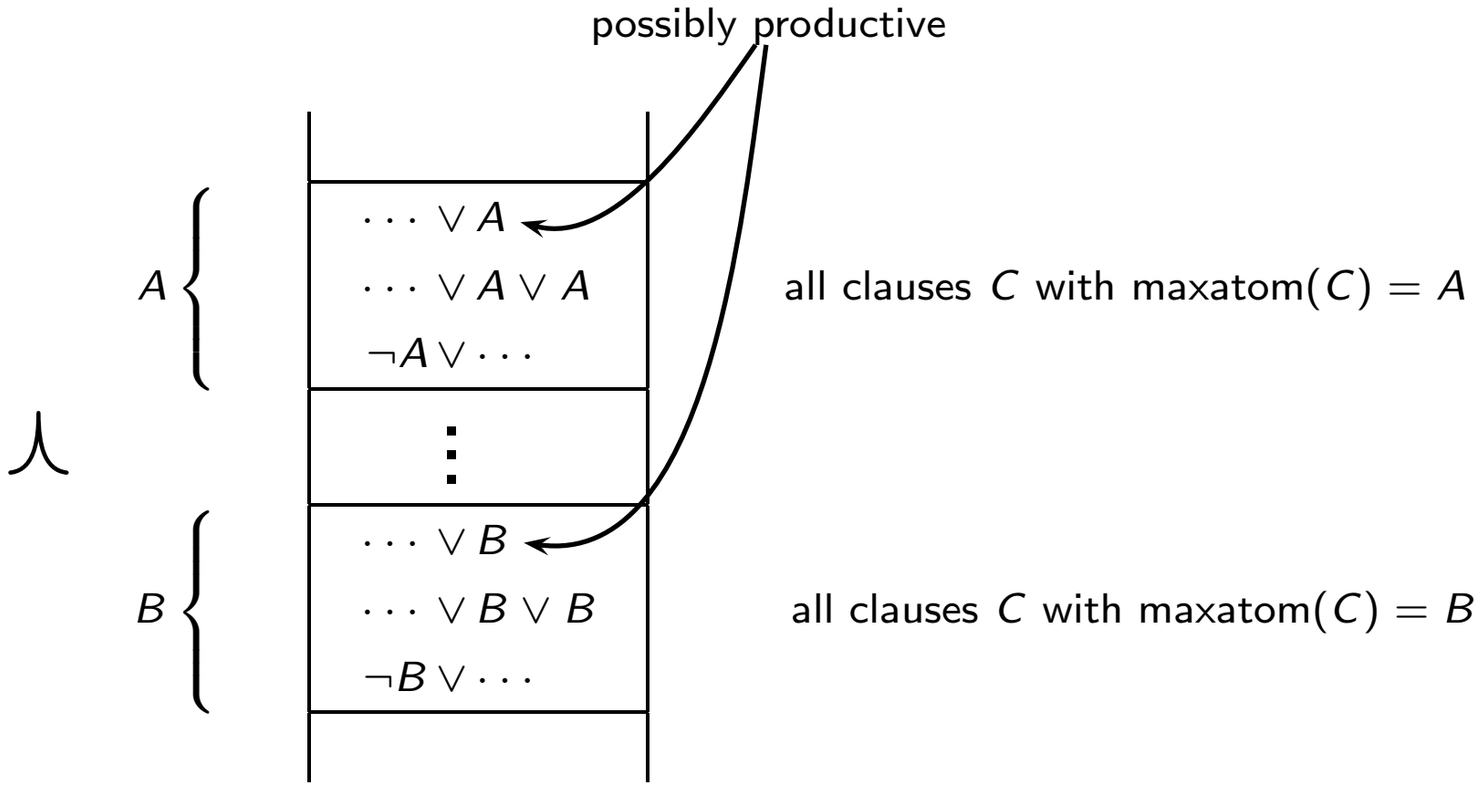
| Iter. | Clause $C$ | $I_C$ | $\Delta_C$ | Remarks |
|---|---|---|---|---|
| 0 | $\neg A_0$ | $\emptyset$ | $\emptyset$ | true in $I_C$ |
| 1 | $A_0 \vee A_1$ | $\emptyset$ | $\{A_1\}$ | $A_1$ maximal |
| 2 | $A_1 \vee A_2$ | $\{A_1\}$ | $\emptyset$ | true in $I_C$ |
| 3 | $\neg A_1 \vee A_2$ | $\{A_1\}$ | $\{A_2\}$ | $A_2$ maximal |
| 4 | $A_0 \vee \neg A_1 \vee A_3 \vee A_4$ | $\{A_1, A_2\}$ | $\{A_4\}$ | $A_4$ maximal |
| 5 | $\neg A_1 \vee A_3 \vee \neg A_4$ | $\{A_1, A_2, A_4\}$ | $\emptyset$ | max. lit. $\neg A_4$ neg.; *min. counter-ex.* |
| 6 | $\neg A_1 \vee A_5$ | $\{A_1, A_2, A_4\}$ | $\{A_5\}$ | |

$I = \{A_1, A_2, A_4, A_5\}$ is not a model of the clause set

$\Rightarrow$ there exists a counterexample.

# Structure of $N, \succ$

Let $B \succ A$. Note that producing a new atom does not change the truth value of smaller clauses.



# Some Properties of the Construction

Proposition 3.9.3:

(i) If $D = D' \vee \neg A$, then no $C \succeq D$ produces $A$.

(ii) If $I_D \models D$, then $I_C \models D$ for every $C \succeq D$ and $I_N^{\succ} \models D$.

(iii) If $D = D' \vee A$ produces $A$,
then $I_C \models D$ for every $C \succ D$ and $I_N^{\succ} \models D$.

(iv) If $D = D' \vee A$ produces $A$,
then $I_C \not\models D'$ for every $C \succeq D$ and $I_N^{\succ} \not\models D'$.

(v) If for every clause $C \in N$, $C$ is productive or $I_C \models C$, then $I_N^{\succ} \models N$.

# Model Existence Theorem

Proposition 3.9.4:
Let $\succ$ be a clause ordering.
If $N$ is saturated w.r.t. *Res* and $\bot \notin N$,
then for every clause $C \in N$, $C$ is productive or $I_C \models C$.

Theorem 3.9.5 (Bachmair and Ganzinger 1990):
Let $\succ$ be a clause ordering.
If $N$ is saturated w.r.t. *Res* and $\bot \notin N$, then $I_N^{\succ} \models N$.

Corollary 3.9.6:
Let $N$ be saturated w.r.t. *Res*.
Then $N \models \bot$ if and only if $\bot \in N$.

# Compactness of Propositional Logic

Lemma 3.9.7:
Let $N$ be a set of propositional (or first-order ground) clauses.
Then $N$ is unsatisfiable if and only if some finite subset $N' \subseteq N$ is unsatisfiable.

Theorem 3.9.8 (Compactness for Propositional Formulas):
Let $S$ be a set of propositional (or first-order ground) formulas.
Then $S$ is unsatisfiable if and only if some finite subset $S' \subseteq S$ is unsatisfiable.