# We want answers!!

$$KB \models \alpha$$

Given $\beta[x_1, \ldots, x_n]$ where $x_1, \ldots, x_n$ are free variables, we want to find the terms (ground) $t_1, \ldots, t_n$ such that:

$$KB \models \beta[t_1, \ldots, t_n]$$

# Some Observations

$$KB \models \alpha$$

$$iff$$

$$\models [(\alpha_1 \wedge \ldots \wedge \alpha_n) \rightarrow \alpha]$$

$$iff$$

$$KB \cup \{\neg\alpha\} \text{ is not satisfiable}$$

$$iff$$

$$KB \cup \{\neg\alpha\} \models \neg TRUE$$

# Propositional Logic

All predicates of 0-arity.

$\Im = \mathcal{I}$

$\mathcal{I}[P] = o$ or $\mathcal{I}[P] = 1$ or

# Propositional Logic: Conjunctive Normal Form

$$(\mathrm{P} \vee \neg \mathrm{Q}) \wedge (\mathrm{Q} \vee \mathrm{R})$$

Clausal Form

$$\{[\mathrm{P}, \neg \mathrm{Q}], [\mathrm{Q}, \mathrm{R}]\}$$

# Transformation to Conjunctive Normal Form

1. eliminate $\rightarrow$ and $\equiv$ by making use of the fact that they are abbreviations for formulas using only $\wedge$, $\vee$ and $\neg$.

2. move $\neg$ inwards so that it applies to only atoms, by using the following equivalences:

$$\models \neg\neg\alpha \equiv \alpha$$
$$\models \neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$$
$$\models \neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta)$$

3. distribute $\wedge$ over $\vee$ using the following equivalences:

$$\models (\alpha \vee (\beta \wedge \gamma)) \equiv ((\beta \wedge \gamma) \vee \alpha)$$

$$\models ((\alpha \vee \beta) \wedge (\alpha \vee \gamma)) \equiv ((\beta \wedge \gamma) \vee \alpha)$$

4. collapse identical atoms, using the following equivalences:

$$(\alpha \vee \alpha) \equiv \alpha$$

$$(\alpha \wedge \alpha) \equiv \alpha$$

# Clauses

**literal** An atom or the negation of an atom.

**clause** A finite set of literals.

$$[\neg \mathrm{R}]$$

$$[\mathrm{P}, \neg \mathrm{Q}, \mathrm{R}]$$

**clausal formula** A finite set of clauses.

$$[\mathrm{P}, \neg \mathrm{Q}, \mathrm{R}], [\mathrm{S}]$$

**empty clause** [] False

# Some Notation

**complement** if L is any literal, then $\bar{L}$ is the complement of L.

**unit clause**

$$[\neg Q]$$

$$[Q]$$

# Example

$(P \wedge (Q \rightarrow R) \rightarrow S)$

# Our Approach

We want to know whether or not $KB \models \alpha$

1. Put the sentences in $KB$ and $\neg\alpha$ into clausal form.
2. Determine whether or not the resulting set of clauses is satisfiable.

# Resolution Inference Rule

$$\frac{c_1 \cup \{l\}, c_2 \cup \{\bar{l}\}}{c_1 \cup c_2}$$

**resolvent**

# Soundness and Completeness

**Sound**

$$\text{If } S \vdash C$$
$$\text{then } S \models c$$

**Complete**

$$\text{If } S \models C$$
$$\text{then } S \vdash c$$

# Refutational Completeness Completeness

We do have:

$$\text{If } S \vdash []$$
$$\text{iff } S \models []$$

$$S \text{ is unsatisfiable}$$
$$\text{iff } S \vdash []$$

# Resolution Derivation

- A *Resolution Derivation* of a clause $c$ from a set of clauses $S$ is a sequence of clauses $c_1, \ldots, c_n$ where the last clause $c_n$ is $c$ and where each $c_i$ is either an element of $S$ or a resolvent of earlier clauses in the derivation.
- $S \vdash c$

# A Resolution Procedure

**Input:** a finite set S of propositional clauses
**Output:** `satisfiable` or `unsatisfiable`

1. Check if [] $\in S$; if so, return `unsatisfiable`.
2. Otherwise, check if there are two clauses in $S$, such that they resolve to produce another clause not already in $S$; if not, return `satisfiable`
3. Otherwise, add the new resolvent clause to $S$, and go back to step 1.

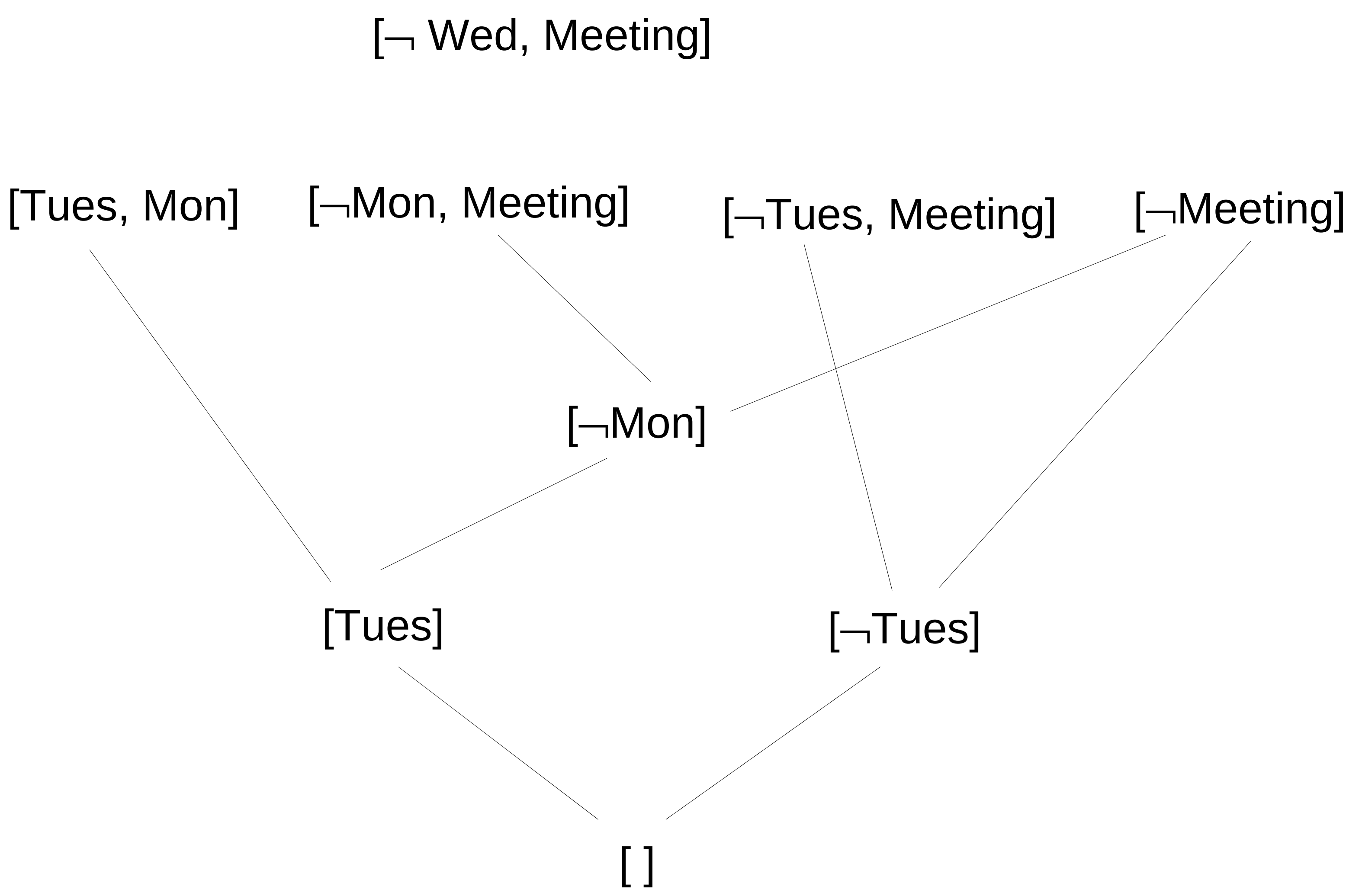
# Example

$$\textsc{Mon} \rightarrow \textsc{Meeting}$$

$$(\textsc{Tues} \vee \textsc{Wed}) \rightarrow \textsc{Meeting}$$

$$\textsc{Mon} \vee \textsc{Tues}$$

# Example (cont)



# Clausal form for 1'st Order Logic

$$\{[\mathrm{P}(x), \neg\mathrm{R}(\textsc{a}, \textsc{f}(\textsc{b}, x))], [\mathrm{Q}(z, y)]\}$$

# Converting to Clausal Form

1. eliminate $\rightarrow$ and $\equiv$ as before.
2. move $\neg$ inwards so that it applies to only atoms, by using the previous equivalences and also:

   $$\models \neg\forall x.\alpha \equiv \exists x.\neg\alpha$$
   $$\models \neg\exists x.\alpha \equiv \forall x.\neg\alpha$$

3. standardize variables apart by renaming as necessary

   $$\models \forall y.\alpha \equiv \forall x.\alpha^y_x$$
   $$\models \exists y.\alpha \equiv \exists x.\alpha^y_x$$

4. Eliminate existentials through Skolemization.

# Converting to Clausal Form (cont)

5. move universals outside the scope of $\wedge$ and $\vee$ using the following equivalences:

$$\models (\alpha \wedge \forall x.\beta) \equiv \forall x.(\alpha \wedge \beta)$$
$$\models (\alpha \vee \forall x.\beta) \equiv \forall x.(\alpha \vee \beta)$$

6. distribute $\wedge$ over $\vee$ as before.
7. collapse identical atoms, as before.

# Substitution

A *substitution* $\theta$ is a finite set of pairs

$$\{x_1/t_1, \ldots, x_n/t_n\}$$

where the $x_i$ are distinct variables and the $t_i$s are arbitrary terms.

Example:

$$\theta \;=\; \{x/\text{A}, y/\text{G}(x, \text{B}, z)\}$$

$$\text{P}(x, z, \text{F}(x, y))\theta$$

*ground clause*, *ground literal*, *ground term*

# First-Order Resolution Rule

$$\frac{c_1 \cup \{l_1\}, c_2 \cup \{l_2\}}{(c_1 \cup c_2)\theta}$$

As long as there is a substitution $\theta$ such that $l_1\theta = \bar{l_2}\theta$

*unifier*, *unifies*

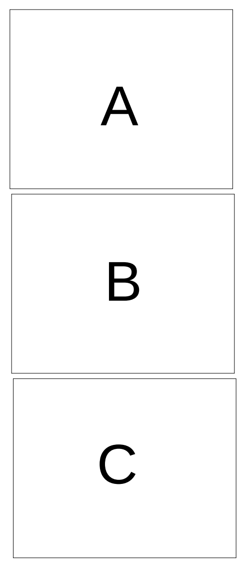
**resolvent**

# Example

Three Blocks Stacked

Top one is green.

Bottom one is not green.

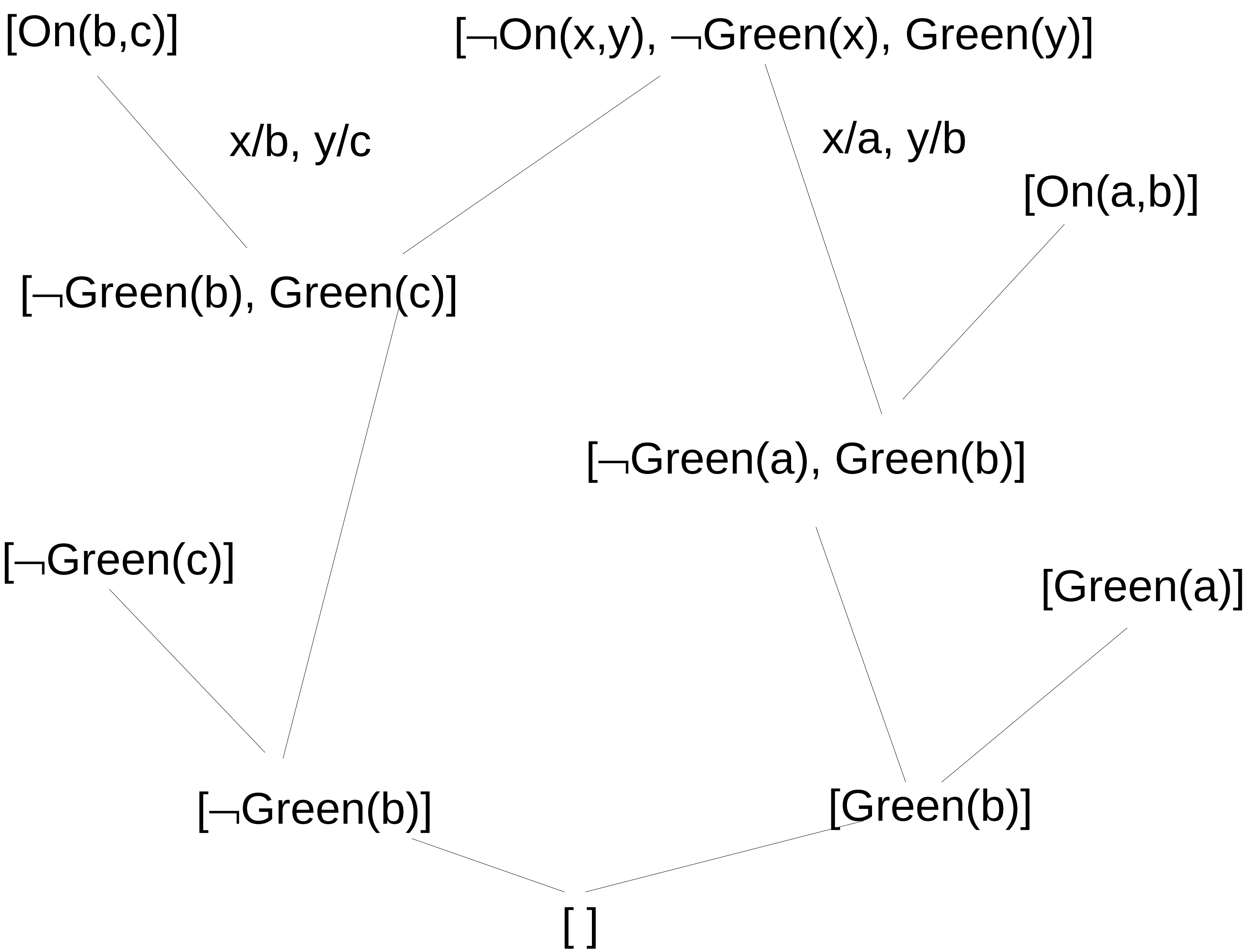



green

non-green

Is there a green block directly on top of a non-green block?

# Example (cont)



# Skolemization

Replace existentials by new function symbols.

$\exists x\ \textsc{Red}(x)$ is replaced by $\textsc{Red}(\textsc{a})$

where $\textsc{a}$ is a new constant symbol that does not occur anywhere else in our database.

In general:

$$\forall x_1(\ldots \forall x_2(\ldots \forall x_3(\ldots \exists y[\ldots y \ldots] \ldots) \ldots) \ldots)$$

$$\forall x_1(\ldots \forall x_2(\ldots \forall x_3(\ldots [\ldots \textsc{f}(x_1, x_2, x_3) \ldots] \ldots) \ldots) \ldots)$$

# Herbrand Theorem

Given a set $S$ of clauses, the *Herbrand universe* of $S$ is the set of all ground terms formed with the function symbols (including constants) in $S$.

Assume we have the 0-arity function symbols A, B, and the unary function symbol G, what is the Herbrand Universe.

# Herbrand Theorem (cont)

The *Herbrand base* of $S$ is the set of all ground clauses $\textsc{c}\theta$ where $\textsc{c} \in S$ and $\theta$ assigns the variables in $\textsc{c}$ to terms in the Herbrand universe.

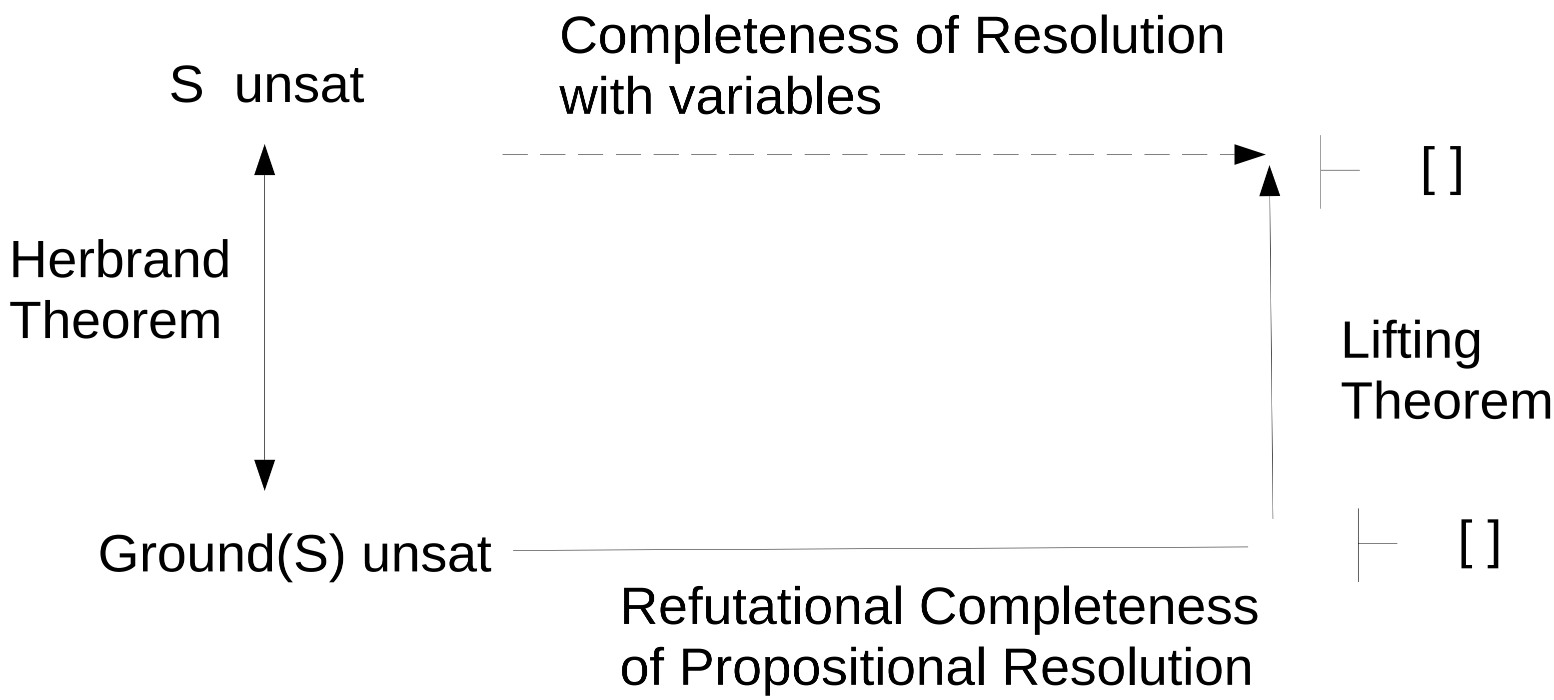
**Theorem**: A set of clauses is satisfiable iff its Herbrand base is satisfiable.

# Most General Unifier

A *most general unifier* (MGU) $\theta$ of literals $l_1$ and $l_2$ is a unifier that has the property that for any other unifier $\theta'$, there is a further substitution $\theta^*$ such that $\theta' = \theta\theta^*$

Can limit the resolution rule to MGUs and still maintain completeness.

# Completeness



# Decidability

- Entailment is decidable for propositional logic. It is not decidable for first-order logic. But it is semidecidable.
- Satisfiability for propositional logic is decidable.
- Satisfiability for first-order logic is not decidable, but is semidecidable.
- What do we do?

# SAT Solvers

Propositional Logic

1. Satisfiability is NP-complete
2. No polynomial algorithm is known. Yet in practice satisfiability testers get good performance.
3. Algorithms include Davis Putnam, GSAT.
4. Many practical applications.