Lecture 14
Proofs and Resolution
Compactness for Propositional Logic
Semantics for First-Order Logic
Normal Forms for Checking Satisfiability
Countable Models for First-Order Logic
Resolution for First-Order Logic

April 30, 2017

Objective

Present foundations for automated theorem provers, such as the ones used in program verification systems.

We will show results for propositional logic and then generalize them to first-order logic.

Procedures we present are sound (derive correct conclusion).
Less obviously, they are complete: if the theorem is true under all interpretations, they these techniques find them eventually

- this may be surprising, given that it applies even to mathematical statements about arbitrarily large sets, functions, and other complex structures
- we have no bound when they will be found, but the fact that for each theorem there exists a finite proof is remarkable

Proof Systems

- Proof rules are computatable relations on finite sequences formulas.
- Given some number of assumptions, a proof rule produces new conclusions.
- A proof tree describes the application of proof rules
- $\Gamma \vdash F$ means that there is a proof tree with leaves Γ that derives F
- Proof steps should be computable: must be able to decide whether a rule applies and can produce a given conclusion.
- A system of logical rules is **sound** iff every conclusion that it only derives is a consequence.
- A proof system is **complete** when it can prove all properties that are true.

Proof System for Propositional Logic

- Fix a countable set of propositional variables V e.g. $p_0, p_1, \ldots$.
  All formulas have variables from V
- Propositional interpretation is a map $I : V \to \mathbb{B}$, $\mathbb{B} = \{true, false\}$
- We write $I \models F$ if formula F is true in model I
- Let Γ be a set of formulas
- $I \models \Gamma$ means $\forall F \in \Gamma.\ I \models F$
- Γ is consistent (satisfiable) if there exists I for which $I \models \Gamma$, else it is contradictory
- $\Gamma \models F$ means $\forall I.\ (I \models \Gamma) \to (I \models F)$
- Proof system "$\vdash$" is **sound** iff $\Gamma \vdash F$ implies $\Gamma \models F$
- Proof system "$\vdash$" is **complete** iff $\Gamma \models F$ implies $\Gamma \vdash F$

Literals and Clauses

A literal is a propositional variable, e.g., p, or its negation $\neg p$.
We assume a countably infinite set of propositional variables e.g. p_i for each non-negative integer i

A clause is a disjunction of literals, like $\neg p \vee \neg q \vee r$.
Using associativity and idempotence of $\vee$, we represent clauses as finite sets of literals, e.g., $\{\neg p, \neg q, r\}$.

Each formula can be represented in disjunctive normal form (DNF) as conjunction of clauses.

By associativity and idempotence of $\wedge$, we can work with finite *sets* of clauses.

Formula $(p \to q) \wedge (q \to p)$ becomes $\{\{\neg p, q\}, \{\neg q, p\}\}$

We will show theorems that hold for both finite and infinite sets of clauses, but will always keep clauses finite.

Propositional Resolution

$$\frac{A \vee L \qquad \neg L \vee B}{A \vee B}$$

Soundness proof:

- Let I be an interpretation in which both $I(A \vee L) = \mathit{true}$ and $I(\neg L \vee B) = \mathit{true}$
- if $I(L) = \mathit{true}$ then from $I(\neg L \vee B) = \mathit{true}$ we conclude $I(B) = \mathit{true}$, so $I(A \vee B) = \mathit{true}$
- if $I(L) = \mathit{false}$ then from $I(A \vee L) = \mathit{true}$ we conclude $I(A) = \mathit{true}$, so $I(A \vee B) = \mathit{true}$
- In any case $I(A \vee B) = \mathit{true}$.

Propositional Resolution on Clauses

Rule on formulas:

$$\frac{A \vee L \qquad \neg L \vee B}{A \vee B}$$

When we represent disjunctions as sets of literals becomes:

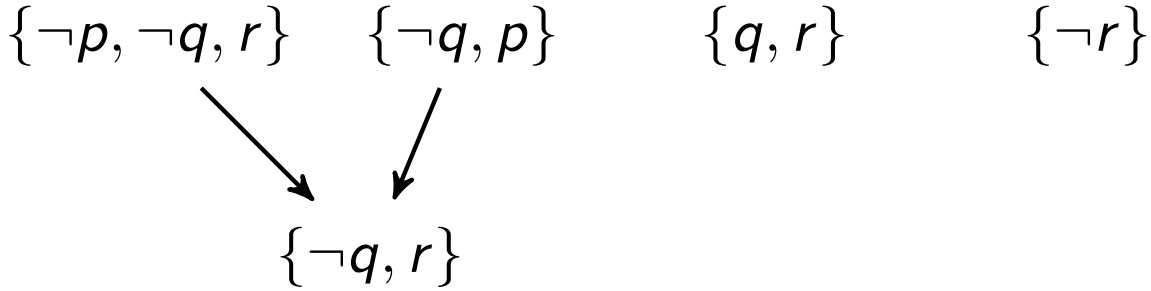
$$\frac{A \cup \{L\} \qquad \{\neg L\} \cup B}{A \cup B}$$

To prove that a formula is valid, we prove that its negation is contradictory by deriving an empty clause (which represents false).

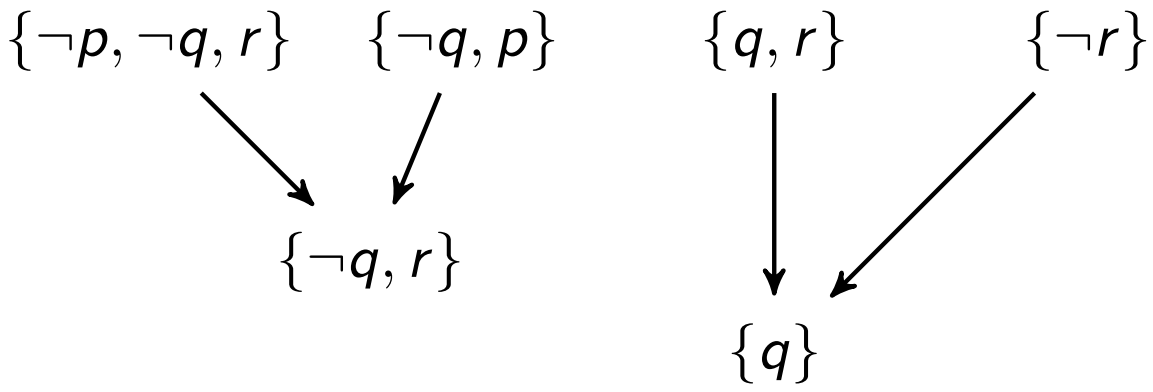
Example Proof of Contradiction by Resolution

$\{\neg p, \neg q, r\}$ $\{\neg q, p\}$ $\{q, r\}$ $\{\neg r\}$

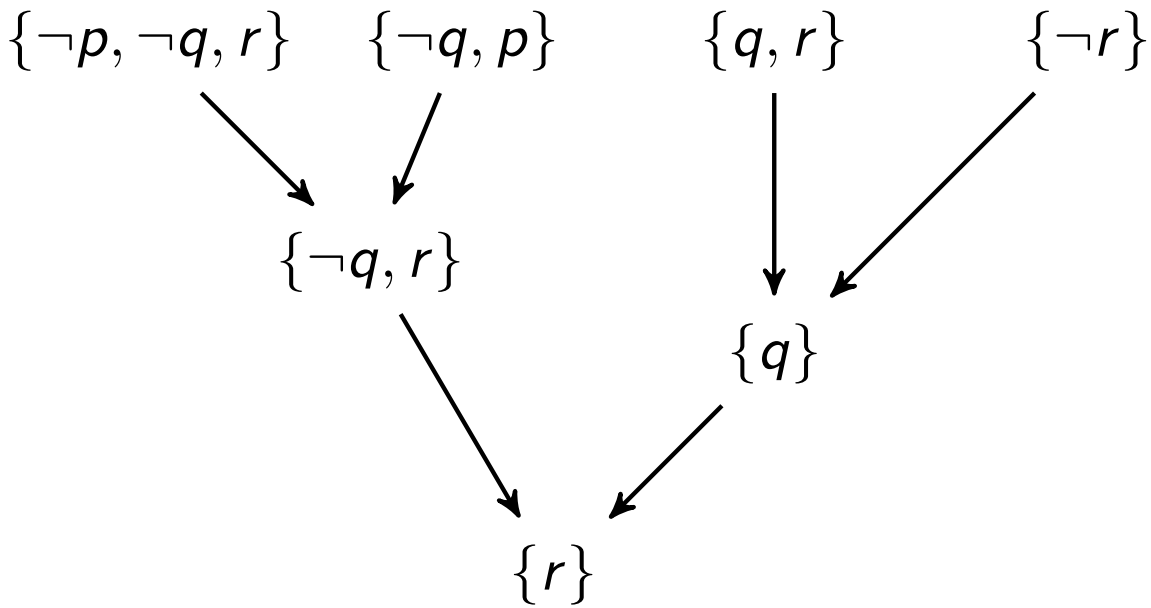
Example Proof of Contradiction by Resolution



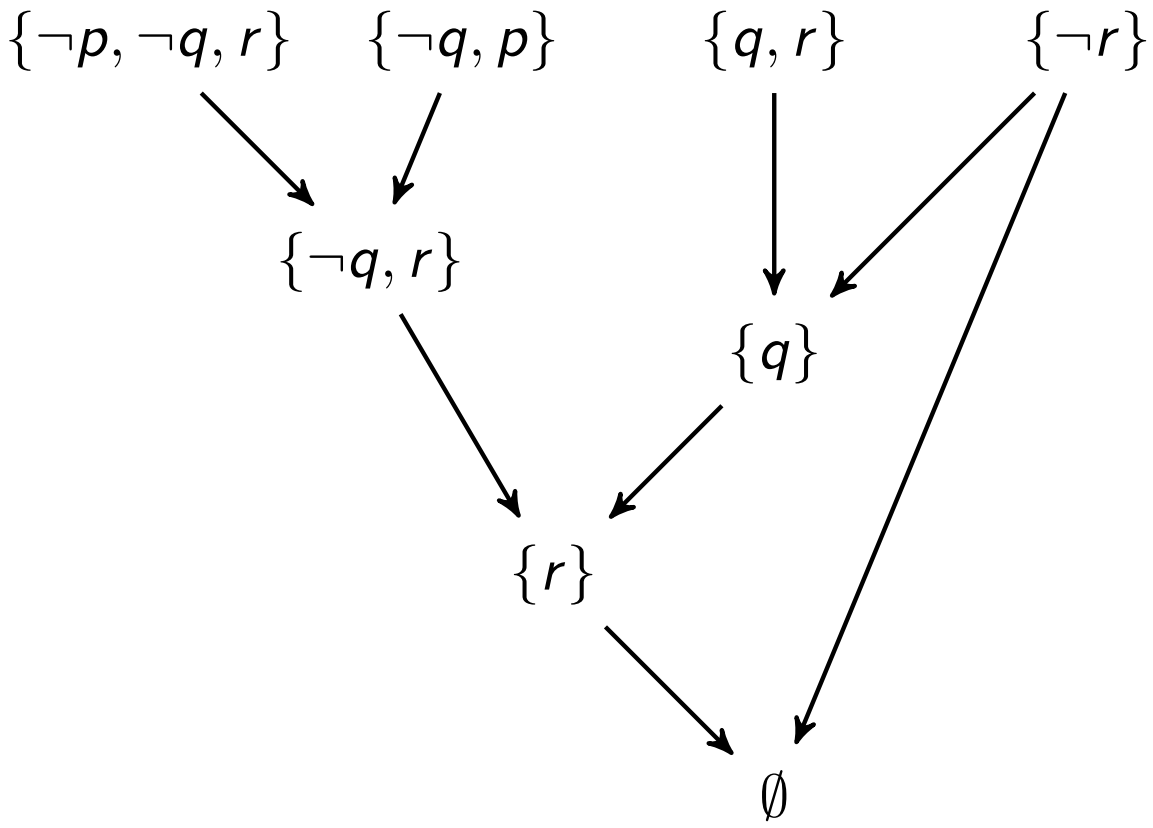
Example Proof of Contradiction by Resolution



Example Proof of Contradiction by Resolution

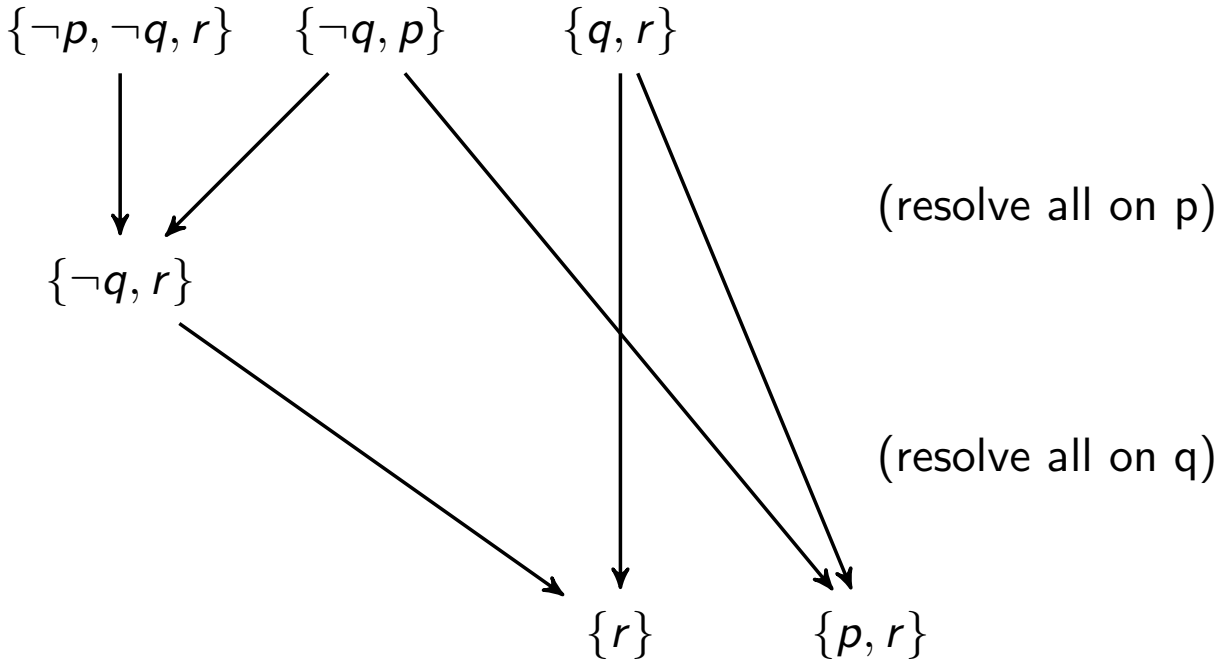


Example Proof of Contradiction by Resolution



Consistency by Absence of Contradiction

Conversely, if the set is contradictory, then existentially quantifying over all variables yields false, so applying resolution exhaustively also yields false.
Resolution is **complete**.
Therefore, if resolution does not yield false, the set is consistent.



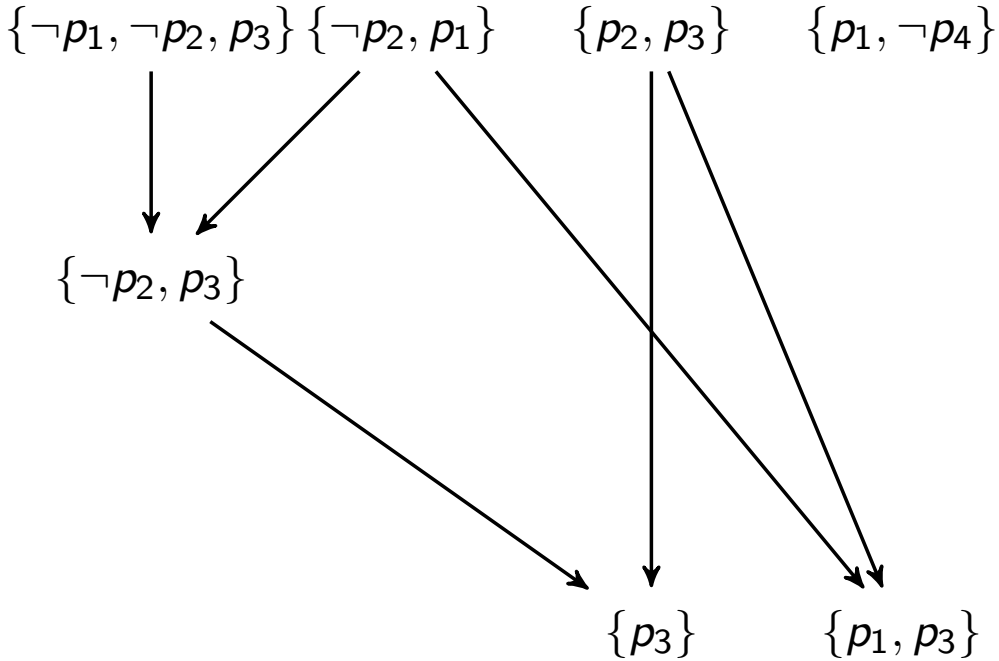
Further resolution attempts would only yield clauses that are subsumed (their subsets, which are stronger, are already derived). No empty clause is generated, so the original set is consistent (a model: $p \mapsto \mathit{true}, r \mapsto \mathit{true}$)

Compactness

Infinite set of Formulas

Suppose that we have a countably infinite set of formulas, with countably many propositional variables

Apply resolution exhaustively to larger and larger prefixes of this infinite set



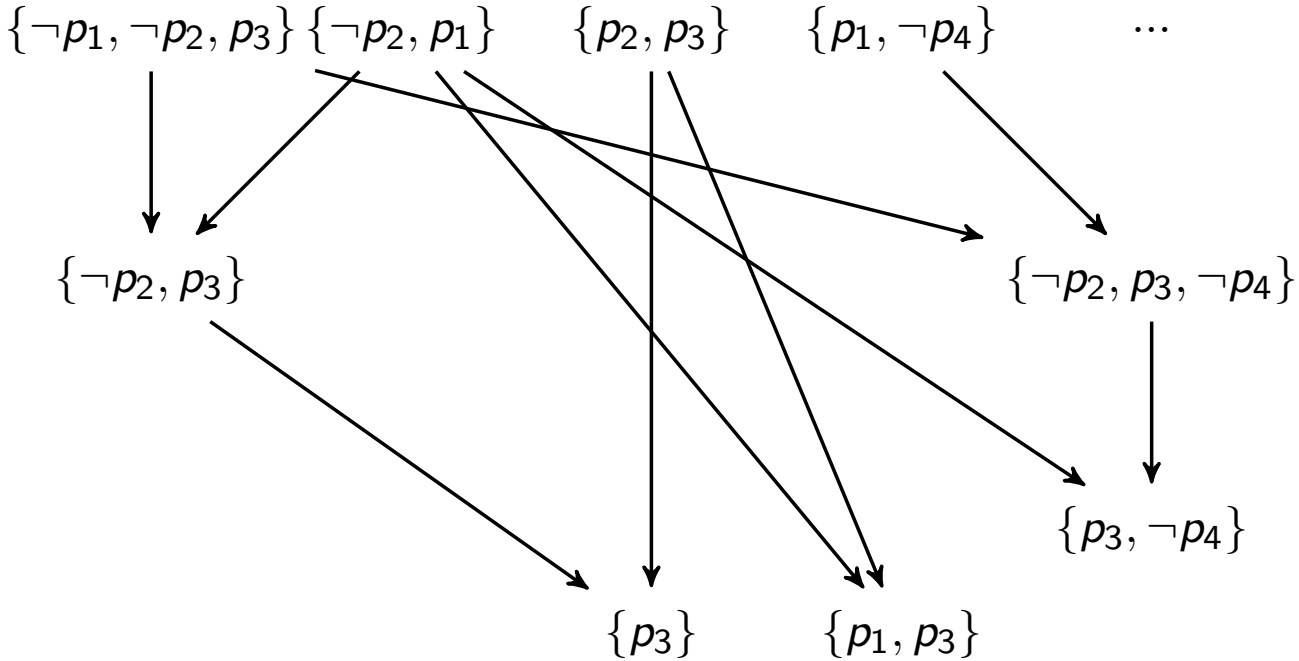
Suppose we are not finding a contradiction in such way. Is the entire infinite set consistent?

Equivalently: if a countable set is contradictory, is there always a finite subset that is contradictory? (Note: there are ∞ many variables.)

Infinite set of Formulas

Suppose that we have a countably infinite set of formulas, with countably many propositional variables

Apply resolution exhaustively to larger and larger prefixes of this infinite set



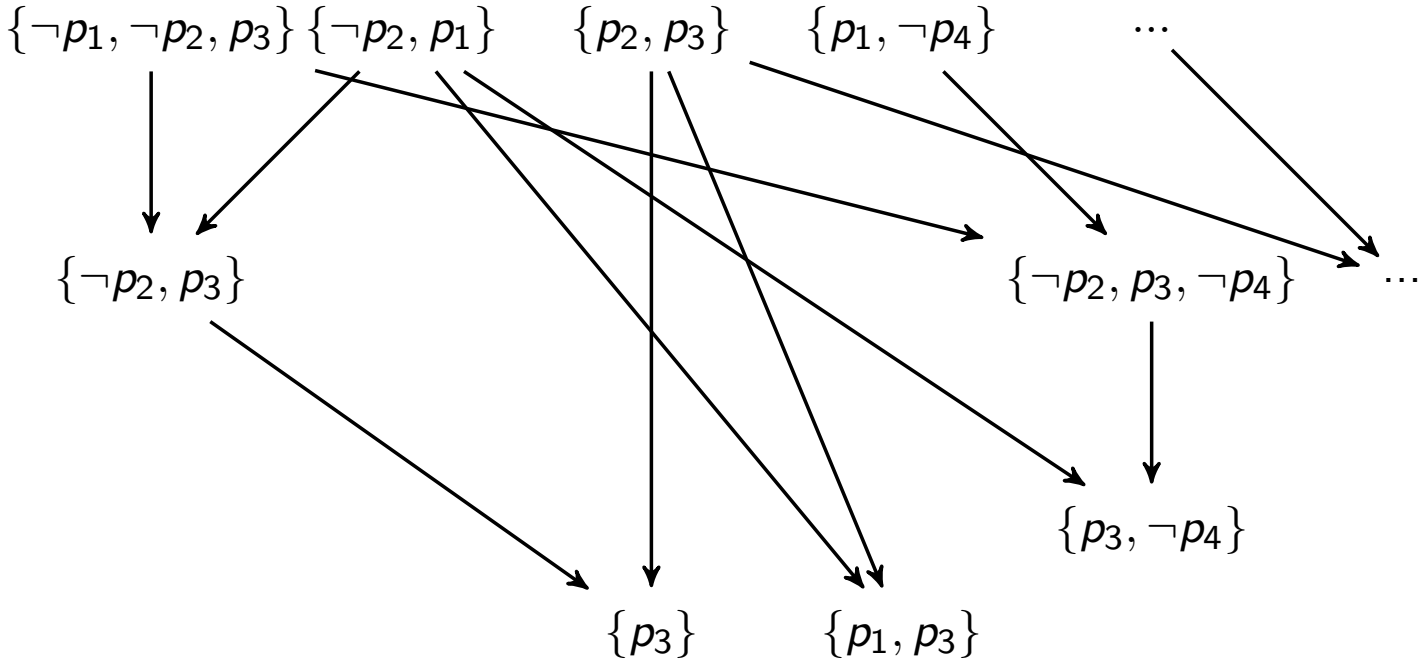
Suppose we are not finding a contradiction in such way. Is the entire infinite set consistent?

Equivalently: if a countable set is contradictory, is there always a finite subset that is contradictory? (Note: there are ∞ many variables.)

Infinite set of Formulas

Suppose that we have a countably infinite set of formulas, with countably many propositional variables

Apply resolution exhaustively to larger and larger prefixes of this infinite set



Suppose we are not finding a contradiction in such way. Is the entire infinite set consistent?

Equivalently: if a countable set is contradictory, is there always a finite subset that is contradictory? (Note: there are ∞ many variables.)

Compactness

Theorem (Compactness for Propositional Logic.)

Let S be a set of propositional formulas. Then S is satisfiable iff every finite subset of S is satisfiable.

Equivalently: S is contradictory iff some finite subset of S is contradictory

Remark: Compactness is a non-trivial property. In logic with infinite disjunctions it does not hold. In such *infinitary logic* we could take $S = \{D, p_1, p_2, p_3, \ldots\}$ where $D = \bigvee_{i=1}^{\infty} \neg p_i$, that is, D is equivalent to $\exists i \geq 0. \neg p_i$. In this example, every finite subset of S is satisfiable, but S itself is not.

Proof of Compactness

One direction is trivial: if S is satisfiable then there exists I such that $I \models S$. Then for every finite subset $T \subseteq S$ we have $I \models T$, so T is satisfiable. So, the point is to show the converse.

Intuition: A finitely satisfiable set has "all finite pieces" satisfiable (using potentially different interpretations). The question is whether we can somehow assemble interpretations for all finite pieces T into one large interpretation for the entire infinite set S. We will define such interpretation by extending it, variable by variable, while preserving finite satisfiability for interepretations that begin with values for propositional variables chosen so far.

Let S be finitely satisfiable. Let $V = \{p_1, p_2, \ldots\}$ be the sequence of all propositional variables for formulas in S (this set is countable by our assumption on syntax of formulas, but can be infinite).

Given a sequence of boolean values $u_1, u_2, \ldots, u_n \in \mathcal{B}$ of length $n \geq 0$, by an $(u_1, u_2, \ldots, u_n)$-interpretation we mean an interpretation $I : V \to \mathcal{B}$ such that $I(p_1) = u_1, \ldots, I(p_n) = u_n$.

Proof: Constructing Interpretation

We will define interpretation $I^*(p_k) = v_k$ where the sequence of values $v_1, v_2, \ldots$ is given as follows:

$$v_{k+1} = \begin{cases} \mathit{false}, & \text{if for every finite } T \subseteq S, \text{ there exists a} \\ & (v_1, \ldots, v_k, \mathit{false}) - \text{interpretation } I \text{ such that } I \models T \\ \mathit{true}, & \text{otherwise} \end{cases}$$

We next show by induction the following.

FIRST PART.
Claim: For every non-negative integer k, every finite subset $T \subseteq S$ has a $(v_1, \ldots, v_k)$-interpretation I such that $I \models T$.
Base case: For $k = 0$ the statement reduces to claim that every finite subset of S is satisfiable, which is an assumption of the theorem.

Inductiveness and the Model

Inductive step: Assume the claim for k: every finite subset $T \subseteq S$ has a $(v_1, \ldots, v_k)$-interpretation I such that $I \models T$, we show that the statement holds for $k+1$.
If $v_{k+1} = \mathit{false}$, the inductive statement holds by definition of v_{k+1}. Let $v_{k+1} = \mathit{true}$.
Then by definition of v_{k+1}, there exists a finite set $A \subseteq S$ that has no $(v_1, \ldots, v_k, \mathit{false})$ interpretation. We wish to show that every finite set $B \subseteq T$ has a $(v_1, \ldots, v_k, \mathit{true})$-interpretation such that $I \models B$. Take any such set B. Consider the set $A \cup B$. This is a finite set, so by inductive hypothesis, it has a $(v_1, \ldots, v_k)$-interpretation I. Because $I \models A$, which has no $(v_1, \ldots, v_k, \mathit{false})$-interpretation, we have $I(p_{k+1}) = \mathit{true}$. Therefore, I is a $(v_1, \ldots, v_k, \mathit{true})$-interpretation for $A \cup B$, and therefore for B. This completes the inductive proof.

From Sequence of Interpretations to One

We have shown that for every non-negative integer k, every finite subset $T \subseteq S$ has a $(v_1, \ldots, v_k)$-interpretation I such that $I \models T$. We have defined $I^*(p_k) = v_k$.

SECOND PART.
We finally show that $I^* \models S$. Let $F \in S$. Let $FV(F) = \{p_{i_1}, \ldots, p_{i_k}\}$ and $M = \max(i_1, \ldots, i_k)$. Then $FV(F) \subseteq \{p_1, \ldots, p_M\}$. The set $\{F\}$ is finite, so, by the Claim, it has a $v_1, \ldots, v_M$-interpretation I such that $I \models F$. Because I^* is also a $v_1, \ldots, v_M$-interpretation, and it agrees with I on all variables in F, we have $I^* \models F$.

We have therefore shown that I^* makes all formulas in S true, as desired.

Why did this work?

How does this proof break if we allow infinite disjunctions? Consider the above example $S = \{D, p_1, p_2, p_3, \ldots\}$ where $D = \bigvee_{i=1}^{\infty} \neg p_i$. The inductively proved claim still holds, and the sequence defined must be $\mathit{true}, \mathit{true}, \mathit{true}, \ldots$. Here is why the claim holds for every k. Let k be arbitrary and $T \subseteq S$ be finite. Define

$$m = \max(k, \max\{i \mid p_i \in T\})$$

Then consider interpretation that assigns to true all p_j for $j \leq m$ and sets the rest to false. Such interpretation makes D true, so if it is in the set T, then interpretation makes it true. Moreover, all other formulas in T are propositional variables set to true, so the interpretation makes T true. Thus, we see that the inductively proved statement holds even in this case. What the infinite formula D breaks is the second part, which, from the existence of interpretations that agree on an arbitrarily long finite prefix derives an interpretation for infinitely many variables. Indeed, this part explicitly refers to a finite number of variables in the formula.

Resolution for First-Order Logic

Automating First-Order Logic

First-order logic allows arbitrary relations and functions (they are defined only through their axioms)

Useful for modeling all of math (e.g. through set theory axioms), and thus in principle applies to all program verification problems as well.

To prove whether a property holds:

- describe the property using a formula F
- describe the functions and relations in F using a sequence of axioms S

Check if the sequence $(\neg F; S)$ is contradictory. If yes, then F follows from S

Completeness: if F follows from S, then there is a procedure that will in finite time find this (in general we do not know how long it will take).

- semantic notion $S \models F$ (in all interpretation of axioms S formulas F is true) can, in first-order logic, too, be replaced with syntactic notion $S \vdash F$ (F can be derived from S)

We give a complete syntactic inference procedure for first-order logic

First-Order Logic

Set of first-order variables $x_1, x_2, \ldots$

Set of function symbols $f \in \mathcal{L}$ of arity $ar(f_i)$. Constants are of arity zero.
Used to build terms. If $ar(f) = n$ and $t_1, \ldots, t_n$ are terms, then
$f(t_1, \ldots, t_n)$ is a term

Set of relation symbols $R \in \mathcal{L}$ of arity $ar(R_i)$
Used to build artomic formulas. If $ar(R) = n$ and $t_1, \ldots, t_n$ are terms, then
$R(t_1, \ldots, t_n)$ is an atomic formula.

From atomic formulas we build quantifier-free formulas using $\wedge, \vee, \neg$

From quantifier-free formulas we build quantified formulas by quantifying over first-order variables using $\forall x_i, \exists x_i$

Example and Models

We will look at the language $\mathcal{L} = \{P, R, a, f\}$ where

- P is relation symbol of arity one
- R is relation symbol of arity two
- a is a constant
- f is a function symbol of two arguments

Consider this formula in $\mathcal{L}$:

$$
\begin{aligned}
&(\forall x.\exists y.\ R(x, y)) \wedge \\
&(\forall x.\forall y.\ (R(x, y) \rightarrow \forall z.\ R(x, f(y, z)))) \wedge \\
&(\forall x.\ (P(x) \vee P(f(x, a)))) \\
&\rightarrow \forall x.\exists y.\ (R(x, y) \wedge P(y))
\end{aligned}
$$

An interpretation is a pair (D, α) of

- the domain, D, which is a non-empty (finite or infinite) set
- the interpretation α that maps:
  - each function symbol f of arity n into a function $\alpha(f) : D^n \rightarrow D$
  - each predicate symbol P of arity n into a relation $\alpha(D) \subseteq D^n$

In the above example, consider: D the set of non-negative integers $\{0, 1, 2, 3, \ldots\}$, a is 1, $R(x, y)$ to hold iff $x < y$, so $\alpha(R) = \{(x, y) \mid x < y\}$, $P(x)$ to be a predicate that holds whenever x is even, and $f(x, y)$ be $x + y$.

Normal Forms for First-Order Logic through Example

We will look at the language $\mathcal{L} = \{P, R, a, f\}$ where

- P is relation symbol of arity one and R rel. symbol of arity two
- a is a constant
- f is a function symbol of two arguments

Consider this formula in $\mathcal{L}$:

$$
\begin{aligned}
&(\forall x.\exists y.\ R(x,y)) \wedge \\
&(\forall x.\forall y.\ R(x,y) \rightarrow \forall z.\ R(x, f(y,z))) \wedge \\
&(\forall x.\ P(x) \vee P(f(x,a))) \\
&\rightarrow \forall x.\exists y.\ R(x,y) \wedge P(y)
\end{aligned}
$$

We are interested in checking the *validity* of this formula (is it true in all interpretations). We will check the *satisfiability* of the negation of this formula (does it have a model):

$$
\begin{aligned}
\neg\Big(&((\forall x.\exists y.\ R(x,y)) \wedge \\
&(\forall x.\forall y.\ R(x,y) \rightarrow \forall z.\ R(x, f(y,z))) \wedge \\
&(\forall x.\ P(x) \vee P(f(x,a))) \ ) \ \rightarrow \forall x.\exists y.\ R(x,y) \wedge P(y) \Big)
\end{aligned}
$$

Negation Normal Form

In negation normal form of formula the negation applies only to atomic formulas.
Every FOL formula can be transformed in NNF using the formulas used for the same purpose in PL extended by two new ones:

- $\neg\neg F \Leftrightarrow F$
- $\neg\bot \Leftrightarrow \top$
- $\neg\top \Leftrightarrow \bot$
- $\neg(F_1 \wedge F_2) \Leftrightarrow \neg F1 \vee \neg F2$
- $\neg(F_1 \vee F_2) \Leftrightarrow \neg F1 \wedge \neg F2$
- $F1 \rightarrow F2 \Leftrightarrow \neg F1 \vee F2$
- $F1 \leftrightarrow F2 \Leftrightarrow (F1 \rightarrow F2) \wedge (F2 \rightarrow F1)$
- $\neg\forall x.F[x] \Leftrightarrow \exists x.\neg F[x]$
- $\neg\exists x.F[x] \Leftrightarrow \forall x.\neg F[x]$

NNF of Example

$$\neg\Big(\big((\forall x.\exists y.\ R(x,y)) \wedge (\forall x.\forall y.\ R(x,y) \rightarrow \forall z.\ R(x,f(y,z))) \wedge (\forall x.\ P(x) \vee P(f(x,a)))\big) \rightarrow \forall x.\exists y.\ R(x,y) \wedge P(y)\Big)$$

becomes:

$$\begin{array}{l}(\forall x.\exists y.\ R(x,y)) \wedge \\ (\exists x.\exists y.\ \neg R(x,y) \vee \forall z.\ R(x,f(y,z))) \wedge \\ (\forall x.\ P(x) \vee P(f(x,a))) \wedge \\ (\exists x.\forall y.\ \neg R(x,y) \vee \neg P(y))\end{array}$$

Prenex Normal Form

Prenex normal form has all quantifiers in front.
Prenex normal form (PNF) is a formula of the form

$$Q_1 x_1 . Q_2 x_2 . \ldots Q_n x_n . G$$

where $Q_i \in \{\forall, \exists\}$ and G has no quantifiers.
Any FOL formula can be transformed to PNF. First convert it to NNF, then if several quantified variables or free variables have the same name rename them to fresh names, and finaly use the following formulas:

- $(\forall x.F) \vee G \Leftrightarrow \forall x.(F \vee G)$
- $(\forall x.F) \wedge G \Leftrightarrow \forall x.(F \wedge G)$
- $(\exists x.F) \vee G \Leftrightarrow \exists x.(F \vee G)$
- $(\exists x.F) \wedge G \Leftrightarrow \exists x.(F \wedge G)$

PNF of Example

$$
\begin{aligned}
&(\forall x.\exists y.\ R(x,y)) \wedge \\
&(\exists x.\exists y.\ \neg R(x,y) \vee \forall z.\ R(x,f(y,z))) \wedge \\
&(\forall x.\ P(x) \vee P(f(x,a))) \wedge \\
&(\exists x.\forall y.\ \neg R(x,y) \vee \neg P(y))
\end{aligned}
$$

becomes (applying to each conjunct separately):

$$
\begin{aligned}
&(\forall x_1.\exists y_1.\ R(x_1,y_1)) \wedge \\
&(\exists x_2.\exists y_2.\ \forall z.\ \neg R(x_2,y_2) \vee R(x_2,f(y_2,z))) \wedge \\
&(\forall x_3.\ P(x_3) \vee P(f(x_3,a))) \wedge \\
&(\exists x_4.\forall y_4.\ \neg R(x_4,y_4) \vee \neg P(y_4))
\end{aligned}
$$

Skolem Normal Form

Let $P : D \times D \to \{true, false\}$ be a predicate with two arguments.
Note that

$$\exists x.\forall y.P(y, x) \to \forall u.\exists v.P(u, v)$$

but converse implication does not hold (take as P relation $\leq$ or $>$ on natural numbers).
In general, we have this theorem:

$$\forall u.\exists v.P(u, v) \leftrightarrow \exists g.\forall u.P(u, g(u))$$

where $g : D \to D$ is a function.
Proof:
($\leftarrow$): For each u we take $f(u)$ as the witness v.
($\rightarrow$): We know there exists a witness v for each u. We define f to map u to one such witness v. (To prove that this is possible requires //axiom of choice// from set theory.)
Note also that satisfiability of formula F expresses existential quantification over function symbols and relation symbols.

Skolemization

Definition: Skolemization is the result of applying this transformation

$$\forall x_1, \ldots, x_n.\exists y.F \rightsquigarrow \forall x_1, \ldots, x_n.subst(\{y \mapsto g(x_1, \ldots, x_n)\})(F)$$

to the entire PNF formula to eliminate all existential quantifiers. Above, g is a fresh function symbol. Denote $snf(F)$ the result of applying skolemization to formula F.

Lemma: A set of formulas S in prenex normal form is satisfiable iff the set $\{snf(F) \mid F \in S\}$ is satisfiable.

SNF for Example

$$
\begin{aligned}
&(\forall x_1.\exists y_1.\ R(x_1, y_1)) \wedge \\
&(\exists x_2.\exists y_2.\ \forall z.\ \neg R(x_2, y_2) \vee R(x_2, f(y_2, z))) \wedge \\
&(\forall x_3.\ P(x_3) \vee P(f(x_3, a))) \wedge \\
&(\exists x_4.\forall y_4.\ \neg R(x_4, y_4) \vee \neg P(y_4))
\end{aligned}
$$

becomes:

$$
\begin{aligned}
&(\forall x.R(x, g(x))) \wedge \\
&(\forall x.\forall y.\forall z.\ \neg R(x, y) \vee R(x, f(y, z))) \wedge \\
&(\forall x.\ P(x) \vee P(f(x, a))) \wedge \\
&(\forall y.\ \neg R(c, y) \vee \neg P(y))
\end{aligned}
$$

Note: it is better to do PNF and SNF //for each conjunct independently//.

CNF and Sets of Clauses

Let $snf(F)$ be $\forall x_1, \ldots, x_n.F$. Convert F to conjunctive normal form $C_1 \wedge \ldots \wedge C_m$. Then $snf(F)$ is equivalent to

$$(\forall x_1, \ldots, x_n.C_1) \wedge \ldots \wedge (\forall x_1, \ldots, x_n.C_m)$$

where each C_i is a disjunction of first-order literals. We call C_i //(first-order) clause//. For a given formula F, denote the set of such clauses in conjunctive normal form of $snf(pnf(F))$ by $clauses(F)$.
We omit universal quantifiers because all variables are universally quantified. We use a convention to denote variables by $x, y, z, \ldots$ and constants by $a, b, c, \ldots$.
Theorem: The set S is satisfiable iff the set

$$\bigcup_{F \in S} clauses(F)$$

is satisfiable.

Clauses for Example

We obtain that satisfiability of the original formula reduces to the satisfiability of the set of clauses $\{C_1, C_2, C_3, C_4\}$
where

- C_1 denotes $\{R(x, g(x))\}$
- C_2 denotes $\{R(x, y), R(x, f(y, z)))\}$
- C_3 denotes $\{P(x), P(f(x, a))\}$
- C_4 denotes $\{\neg R(c, y), \neg P(y)\}$

Another Example: Irreflexive Dense Linear Orders

Let $\mathcal{L} = \{less\}$ be binary relation ("strictly less"). We consider the following axioms for irreflexive partial order that is total and dense:

$$
\begin{array}{rcl}
IRef & \equiv & \forall x.\ \neg less(x, x) \\
Tra & \equiv & \forall x.\ \forall y.\ \forall z.\ less(x, y) \wedge less(y, z) \rightarrow less(x, z) \\
Total & \equiv & \forall x. \forall y.\ x \neq y \rightarrow less(x, y) \vee less(y, x) \\
Dense & \equiv & \forall x. \forall y.\ less(x, y) \rightarrow \exists z.\ less(x, z) \wedge less(z, y)
\end{array}
$$

Another Example: Irreflexive Dense Linear Orders

Let $\mathcal{L} = \{less\}$ be binary relation ("strictly less"). We consider the following axioms for irreflexive partial order that is total and dense:

$$\begin{aligned}
IRef &\equiv \forall x.\ \neg less(x,x) \\
Tra &\equiv \forall x.\ \forall y.\ \forall z.\ less(x,y) \wedge less(y,z) \rightarrow less(x,z) \\
Total &\equiv \forall x. \forall y.\ x \neq y \rightarrow less(x,y) \vee less(y,x) \\
Dense &\equiv \forall x. \forall y.\ less(x,y) \rightarrow \exists z.\ less(x,z) \wedge less(z,y)
\end{aligned}$$

Clauses for these axioms are (one set per line):

$$\begin{array}{l}
\neg less(x_1, x_1) \\
\neg less(x_2, y_2), \neg less(y_2, z_2), less(x_2, z_2) \\
x_3 = y_3, less(x_3, y_3), less(y_3, x_3) \\
\neg less(x_4, y_4),\ \ less(x_4, f(x_4, y_4)) \\
\neg less(x_4, y_4),\ \ less(f(x_4, y_4), y_4)
\end{array}$$

Example Formula in First-Order Logic

model of a formula = interpretation (structure) that makes a formula true

$$
\begin{aligned}
\neg(\quad & (\forall x.\exists y.\ R(x,y)) \wedge \\
& (\forall x.\forall y.\ (R(x,y) \Rightarrow \forall z.\ R(x, f(y,z)))) \wedge \\
& (\forall x.\ (P(x) \vee P(f(x,a)))) \\
& \Rightarrow \forall x.\exists y.\ (R(x,y) \wedge P(y)))
\end{aligned}
$$

Example Formula in First-Order Logic

model of a formula = interpretation (structure) that makes a formula true

$$\begin{aligned}\neg(\quad & (\forall x.\exists y.\ R(x,y)) \wedge \\ & (\forall x.\forall y.\ (R(x,y) \Rightarrow \forall z.\ R(x,f(y,z)))) \wedge \\ & (\forall x.\ (P(x) \vee P(f(x,a)))) \\ & \Rightarrow \forall x.\exists y.\ (R(x,y) \wedge P(y)))\end{aligned}$$

After normal form and Skolemization we obtain these first-order clauses:

$$\begin{aligned}& R(x, g_1(x)) \\ & \neg R(x,y) \vee R(x, f(y,z)) \\ & P(x) \vee P(f(x,a)) \\ & \neg R(c_0, y) \vee \neg P(y)\end{aligned}$$

- variables are implicitly $\forall$ quantified; there are no $\exists$ quantifiers
- each clause is disjunction of literals (atomic formulas or their negation)
- from any model of these clauses we can obtain model for the original formula (just ignore interpretation of Skolem constants g_1, c_0)

Example Formula in First-Order Logic

model of a formula = interpretation (structure) that makes a formula true

$$\begin{aligned}\neg(\quad & (\forall x.\exists y.\ R(x,y)) \wedge \\ & (\forall x.\forall y.\ (R(x,y) \Rightarrow \forall z.\ R(x,f(y,z)))) \wedge \\ & (\forall x.\ (P(x) \vee P(f(x,a)))) \\ & \Rightarrow \forall x.\exists y.\ (R(x,y) \wedge P(y)))\end{aligned}$$

After normal form and Skolemization we obtain these first-order clauses:

$$\begin{aligned}& R(x, g_1(x)) \\ & \neg R(x,y) \vee R(x, f(y,z)) \\ & P(x) \vee P(f(x,a)) \\ & \neg R(c_0, y) \vee \neg P(y)\end{aligned}$$

- variables are implicitly $\forall$ quantified; there are no $\exists$ quantifiers
- each clause is disjunction of literals (atomic formulas or their negation)
- from any model of these clauses we can obtain model for the original formula (just ignore interpretation of Skolem constants g_1, c_0)

Do given universally quantified formulas have a model?

Finding a Smaller Model

Small model theorems in logic: "if a given set of formulas has a model, then it has a model of a particular kind (e.g. small)"

- First place to look for smaller models: **substructures**

Given a structure (interpretation) (D, α) a substructure is (D', α') where

- $D' \subseteq D$
- for elements in D', α' defines the relations and functions in the same way, so $\alpha'(R) = \alpha(R) \cap (D')^n$ for $n = ar(R)$, and $\alpha'(f)(x_1, ..., x_n) = \alpha(f)(x_1, ..., x_n)$ for $n = ar(f)$
- (D', α') is a valid interpretation, in particular, it maps function symbols of arity n to total functions on $(D')^n \to D'$

Observation: Given (D, α), a substructure is uniquely given by its domain $D' \subseteq D$. The domain D' defines a substructure if and only if it is closed under the interpretation of all function symbols f:

$$\bigwedge_{f \in \mathcal{L}_F} \forall x_1, ..., x_n \in D'.\ \alpha(f)(x_1, ..., x_n) \in D'$$

Examples of Substructures

$\mathcal{L} = \{f, a, b, T\}$ where

- f, a, b are functions symbols of arity $2, 0, 0$, respectively; $\mathcal{L}_F = \{f, a, b\}$
- T is a binary relation symbol

(D, α) is given by $D = \mathbb{R}$ (real numbers) and

- $\alpha(a) = 0$, $\alpha(b) = 1$
- $\alpha(f)(x, y) = x + y$
- $\alpha(T) = \{(x, y) | x \leq y\}$

How do substructures look like?

Examples of Substructures

$\mathcal{L} = \{f, a, b, T\}$ where

- f, a, b are functions symbols of arity $2, 0, 0$, respectively; $\mathcal{L}_F = \{f, a, b\}$
- T is a binary relation symbol

(D, α) is given by $D = \mathbb{R}$ (real numbers) and

- $\alpha(a) = 0$, $\alpha(b) = 1$
- $\alpha(f)(x, y) = x + y$
- $\alpha(T) = \{(x, y) | x \leq y\}$

How do substructures look like?

- The set $D'_1 = \{1, 2, \ldots\}$ is not a substructure because $\alpha(a) \notin D'$.

Examples of Substructures

$\mathcal{L} = \{f, a, b, T\}$ where

- f, a, b are functions symbols of arity $2, 0, 0$, respectively; $\mathcal{L}_F = \{f, a, b\}$
- T is a binary relation symbol

(D, α) is given by $D = \mathbb{R}$ (real numbers) and

- $\alpha(a) = 0$, $\alpha(b) = 1$
- $\alpha(f)(x, y) = x + y$
- $\alpha(T) = \{(x, y) | x \leq y\}$

How do substructures look like?

- The set $D'_1 = \{1, 2, \ldots\}$ is not a substructure because $\alpha(a) \notin D'$.
- Then the set $D'_2 = \{0, 1, 2\}$ does not form a substructure because it is not closed under addition, e.g. $1 + 2 \notin D'_2$.

Examples of Substructures

$\mathcal{L} = \{f, a, b, T\}$ where

- f, a, b are functions symbols of arity $2, 0, 0$, respectively; $\mathcal{L}_F = \{f, a, b\}$
- T is a binary relation symbol

(D, α) is given by $D = \mathbb{R}$ (real numbers) and

- $\alpha(a) = 0$, $\alpha(b) = 1$
- $\alpha(f)(x, y) = x + y$
- $\alpha(T) = \{(x, y) | x \leq y\}$

How do substructures look like?

- The set $D'_1 = \{1, 2, \ldots\}$ is not a substructure because $\alpha(a) \notin D'$.
- Then the set $D'_2 = \{0, 1, 2\}$ does not form a substructure because it is not closed under addition, e.g. $1 + 2 \notin D'_2$.
- The set of integers $D'_3 = \mathbb{Z}$ induces a substructure because: (i) $\alpha(a) \in \mathbb{Z}$, (ii) $\alpha(b) \in \mathbb{Z}$, and (iii) $x, y \in \mathbb{Z} \Rightarrow x + y \in \mathbb{Z}$.

Universal Formulas Stay True in Substructures

Consider a **universal** formula, with only universal quantifiers (e.g. after Skolemization)

$$\forall x_1, ..., x_n.\ G(x_1, ..., x_n)$$

where G is quantifier free. Suppose this formula is true in (D, α). This means

$$\forall e_1, ..., e_n \in D.\ [\![G(x_1, ..., x_n)]\!]^{\alpha[x_i := e_i]_{i=1}^n}$$

Let (D', α) be a substructure of (D, α). Then from $D' \subseteq D$ follows also

$$\forall e_1, ..., e_n \in D'.\ [\![G(x_1, ..., x_n)]\!]^{\alpha[x_i := e_i]_{i=1}^n}$$

so the formula remains true in substructure.

Universal Formulas Stay True in Substructures

Consider a **universal** formula, with only universal quantifiers (e.g. after Skolemization)

$$\forall x_1, ..., x_n.\ G(x_1, ..., x_n)$$

where G is quantifier free. Suppose this formula is true in (D, α). This means

$$\forall e_1, ..., e_n \in D.\ [\![G(x_1, ..., x_n)]\!]^{\alpha[x_i := e_i]_{i=1}^n}$$

Let (D', α) be a substructure of (D, α). Then from $D' \subseteq D$ follows also

$$\forall e_1, ..., e_n \in D'.\ [\![G(x_1, ..., x_n)]\!]^{\alpha[x_i := e_i]_{i=1}^n}$$

so the formula remains true in substructure.

Theorem

If a set of universal first-order formulas is true in a structure, then it is true in each of its substructures.

Universal Formulas Stay True in Substructures

Consider a **universal** formula, with only universal quantifiers (e.g. after Skolemization)

$$\forall x_1, ..., x_n.\ G(x_1, ..., x_n)$$

where G is quantifier free. Suppose this formula is true in (D, α). This means

$$\forall e_1, ..., e_n \in D.\ [\![G(x_1, ..., x_n)]\!]^{\alpha[x_i := e_i]_{i=1}^n}$$

Let (D', α) be a substructure of (D, α). Then from $D' \subseteq D$ follows also

$$\forall e_1, ..., e_n \in D'.\ [\![G(x_1, ..., x_n)]\!]^{\alpha[x_i := e_i]_{i=1}^n}$$

so the formula remains true in substructure.

Theorem

If a set of universal first-order formulas is true in a structure, then it is true in each of its substructures.

Our goal: find a small substructure

Smallest Substructure

(D, α) is given by $D = \mathbb{R}$ (real numbers) and

- $\alpha(a) = 0$, $\alpha(b) = 1$
- $\alpha(f)(x, y) = x + y$
- $\alpha(T) = \{(x, y) | x \leq y\}$

Let D' be a substructure. Which elements must it contain?

Smallest Substructure

(D, α) is given by $D = \mathbb{R}$ (real numbers) and

- $\alpha(a) = 0$, $\alpha(b) = 1$
- $\alpha(f)(x, y) = x + y$
- $\alpha(T) = \{(x, y) | x \leq y\}$

Let D' be a substructure. Which elements must it contain?

- 0, 1 (interpretations of constants)

Smallest Substructure

(D, α) is given by $D = \mathbb{R}$ (real numbers) and

- $\alpha(a) = 0$, $\alpha(b) = 1$
- $\alpha(f)(x, y) = x + y$
- $\alpha(T) = \{(x, y) | x \leq y\}$

Let D' be a substructure. Which elements must it contain?

- 0, 1 (interpretations of constants)
- $0 + 1$, $1 + 0$, $1 + 1$ (adding up constants), so $2 \in D'$

Smallest Substructure

(D, α) is given by $D = \mathbb{R}$ (real numbers) and

- $\alpha(a) = 0$, $\alpha(b) = 1$
- $\alpha(f)(x, y) = x + y$
- $\alpha(T) = \{(x, y) | x \leq y\}$

Let D' be a substructure. Which elements must it contain?

- 0, 1 (interpretations of constants)
- $0 + 1$, $1 + 0$, $1 + 1$ (adding up constants), so $2 \in D'$
- $2 + 1 = 3 \in D'$

Smallest Substructure

(D, α) is given by $D = \mathbb{R}$ (real numbers) and

- $\alpha(a) = 0$, $\alpha(b) = 1$
- $\alpha(f)(x, y) = x + y$
- $\alpha(T) = \{(x, y) | x \leq y\}$

Let D' be a substructure. Which elements must it contain?

- 0, 1 (interpretations of constants)
- $0 + 1$, $1 + 0$, $1 + 1$ (adding up constants), so $2 \in D'$
- $2 + 1 = 3 \in D'$
- every non-negative integer

Smallest Substructure

(D, α) is given by $D = \mathbb{R}$ (real numbers) and

- $\alpha(a) = 0$, $\alpha(b) = 1$
- $\alpha(f)(x, y) = x + y$
- $\alpha(T) = \{(x, y) | x \leq y\}$

Let D' be a substructure. Which elements must it contain?

- 0, 1 (interpretations of constants)
- $0 + 1$, $1 + 0$, $1 + 1$ (adding up constants), so $2 \in D'$
- $2 + 1 = 3 \in D'$
- every non-negative integer

Define: $D_0 = \emptyset$, $D_{i+1} = \{0, 1\} \cup \{x + y | x, y \in D_i\}$ i.e.

Smallest Substructure

(D, α) is given by $D = \mathbb{R}$ (real numbers) and

- $\alpha(a) = 0$, $\alpha(b) = 1$
- $\alpha(f)(x, y) = x + y$
- $\alpha(T) = \{(x, y) | x \leq y\}$

Let D' be a substructure. Which elements must it contain?

- 0, 1 (interpretations of constants)
- $0 + 1$, $1 + 0$, $1 + 1$ (adding up constants), so $2 \in D'$
- $2 + 1 = 3 \in D'$
- every non-negative integer

Define: $D_0 = \emptyset$, $D_{i+1} = \{0, 1\} \cup \{x + y | x, y \in D_i\}$ i.e. $D_{i+1} = \{\alpha(a), \alpha(b)\} \cup \{\alpha(f)(x, y) | x, y \in D_i\}$.

Smallest Substructure

(D, α) is given by $D = \mathbb{R}$ (real numbers) and

- $\alpha(a) = 0$, $\alpha(b) = 1$
- $\alpha(f)(x, y) = x + y$
- $\alpha(T) = \{(x, y) | x \leq y\}$

Let D' be a substructure. Which elements must it contain?

- 0, 1 (interpretations of constants)
- $0 + 1$, $1 + 0$, $1 + 1$ (adding up constants), so $2 \in D'$
- $2 + 1 = 3 \in D'$
- every non-negative integer

Define: $D_0 = \emptyset$, $D_{i+1} = \{0, 1\} \cup \{x + y | x, y \in D_i\}$ i.e. $D_{i+1} = \{\alpha(a), \alpha(b)\} \cup \{\alpha(f)(x, y) | x, y \in D_i\}$. Let $D^* = \bigcup_{i \geq 0} D_i$

Smallest Substructure

(D, α) is given by $D = \mathbb{R}$ (real numbers) and

- $\alpha(a) = 0$, $\alpha(b) = 1$
- $\alpha(f)(x, y) = x + y$
- $\alpha(T) = \{(x, y) | x \leq y\}$

Let D' be a substructure. Which elements must it contain?

- 0, 1 (interpretations of constants)
- $0 + 1$, $1 + 0$, $1 + 1$ (adding up constants), so $2 \in D'$
- $2 + 1 = 3 \in D'$
- every non-negative integer

Define: $D_0 = \emptyset$, $D_{i+1} = \{0, 1\} \cup \{x + y | x, y \in D_i\}$ i.e.
$D_{i+1} = \{\alpha(a), \alpha(b)\} \cup \{\alpha(f)(x, y) | x, y \in D_i\}$. Let $D^* = \bigcup_{i \geq 0} D_i$

Least fixpoint of function $H(D_k) = \{\alpha(a), \alpha(b)\} \cup \{\alpha(f)(x, y) | x, y \in D_k\}$

Smallest Substructure

(D, α) is given by $D = \mathbb{R}$ (real numbers) and

- $\alpha(a) = 0$, $\alpha(b) = 1$
- $\alpha(f)(x, y) = x + y$
- $\alpha(T) = \{(x, y) | x \leq y\}$

Let D' be a substructure. Which elements must it contain?

- 0, 1 (interpretations of constants)
- $0 + 1$, $1 + 0$, $1 + 1$ (adding up constants), so $2 \in D'$
- $2 + 1 = 3 \in D'$
- every non-negative integer

Define: $D_0 = \emptyset$, $D_{i+1} = \{0, 1\} \cup \{x + y | x, y \in D_i\}$ i.e.
$D_{i+1} = \{\alpha(a), \alpha(b)\} \cup \{\alpha(f)(x, y) | x, y \in D_i\}$. Let $D^* = \bigcup_{i \geq 0} D_i$

Least fixpoint of function $H(D_k) = \{\alpha(a), \alpha(b)\} \cup \{\alpha(f)(x, y) | x, y \in D_k\}$
Evey set D_i is finite. D^* is countable: can enumerate elements of D_1, followed by the elements of D_2, D_3,... establishing bijection with $\mathbb{N}$

Definition of Smallest Substructure

Language $\mathcal{L}$ with function symbols $\mathcal{L}_F \subseteq \mathcal{L}$.

$$D_0 = \emptyset$$

$$D_{i+1} = \bigcup_{f \in \mathcal{L}_F} \{\alpha(f)(x_1, ..., x_n) | x_1, ..., x_n \in D_i\}$$

$$D^* = \bigcup_{i \geq 0} D_i$$

Note: D_i for $i \geq 1$ includes the interpretations of all constants, which are functions of arity $n = 0$

Theorem

- *D^* is the domain of the smallest substructure of (D, α)*
- *D^* is*
  - *always countable*
  - *non-empty $\Leftrightarrow$ $\mathcal{L}$ contains at least one constant symbol*
  - *finite when $\mathcal{L}$ has no function symbols except for constants*

Countable Model Theorem

Lemma

A set of **universal** *first-order formulas has a model if and only if it has a countable model.*

Proof.

Let (D, α) be a model. Then D^* induces a countable sub-structure. Because all formulas are universal, they remain true in D^*. □

Theorem

A set of first-order formulas has a model if and only if it has a countable model.

Proof.

Let the set of formulas have a model. Transform the formulas into normal form and skolemize them to eliminate existential quantifiers, which introduces a countable number of skolem functions. Then there is a model for the resulting set of universal formulas as well. By previous lemma, then there is also a countable model. Ignoring the interpretation of Skolem constants, we obtain a countable model for the original formula. □

Example: Dense Orders

Consider these axioms, which define *dense linear orders* without upper bound:

$$\forall x.\neg T(x,x)$$
$$\forall x \forall y \forall z.\, T(x,y) \wedge T(y,z) \Rightarrow T(x,z)$$
$$\forall x \forall y.(T(x,y) \Rightarrow \exists z.(T(x,z) \wedge T(z,y)))$$
$$\forall x \exists y.\, T(x,y)$$

Real numbers with strict inequality $<$ interpreting relation symbol T are a model of these axioms. Find one countable non-empty model using our construction.

Example: Dense Orders

Consider these axioms, which define *dense linear orders* without upper bound:

$$\forall x. \neg T(x, x)$$
$$\forall x \forall y \forall z. T(x, y) \wedge T(y, z) \Rightarrow T(x, z)$$
$$\forall x \forall y. (T(x, y) \Rightarrow \exists z. (T(x, z) \wedge T(z, y)))$$
$$\forall x \exists y. T(x, y)$$

Real numbers with strict inequality $<$ interpreting relation symbol T are a model of these axioms. Find one countable non-empty model using our construction.
Skolemizing the existential quantifier for density using $g(x, y)$ and for no-bound with $h(x)$:

$$\neg T(x, x)$$
$$\neg T(x, y) \vee \neg T(y, z) \vee T(x, z)$$
$$\neg T(x, y) \vee (T(x, g(x, y)) \wedge T(g(x, y), y))$$
$$T(x, h(x))$$

Finding Non-Empty Countable Model

$$\neg T(x, x)$$
$$\neg T(x, y) \vee \neg T(y, z) \vee T(x, z)$$
$$\neg T(x, y) \vee (T(x, g(x, y)) \wedge T(g(x, y), y))$$
$$T(x, h(x))$$

Theorem ensures we can find interpretation of g, h.

One possibility:

Finding Non-Empty Countable Model

$$\neg T(x,x)$$
$$\neg T(x,y) \vee \neg T(y,z) \vee T(x,z)$$
$$\neg T(x,y) \vee (T(x,g(x,y)) \wedge T(g(x,y),y))$$
$$T(x,h(x))$$

Theorem ensures we can find interpretation of g, h.
One possibility: $g(x,y) = (x+y)/2 \quad h(y) = y+1$
Since we have no constant and do not wish to have an empty domain, just pick any element as the starting point.

Finding Non-Empty Countable Model

$$\neg T(x,x)$$
$$\neg T(x,y) \vee \neg T(y,z) \vee T(x,z)$$
$$\neg T(x,y) \vee (T(x,g(x,y)) \wedge T(g(x,y),y))$$
$$T(x,h(x))$$

Theorem ensures we can find interpretation of g, h.
One possibility: $g(x,y) = (x+y)/2 \quad h(y) = y+1$
Since we have no constant and do not wish to have an empty domain, just pick any element as the starting point. Say, 0.
Apply closure under operations. Here they are all Skolem operations, but in general we use all operations we have, original or Skolem. Describe the set generated in this way.

Finding Non-Empty Countable Model

$$\neg T(x,x)$$
$$\neg T(x,y) \vee \neg T(y,z) \vee T(x,z)$$
$$\neg T(x,y) \vee (T(x,g(x,y)) \wedge T(g(x,y),y))$$
$$T(x,h(x))$$

Theorem ensures we can find interpretation of g, h.
One possibility: $g(x,y) = (x+y)/2 \quad h(y) = y+1$
Since we have no constant and do not wish to have an empty domain, just pick any element as the starting point. Say, 0.
Apply closure under operations. Here they are all Skolem operations, but in general we use all operations we have, original or Skolem. Describe the set generated in this way.
Answer: The set of all non-negative numbers representable in binary notation $\overline{b_1 \ldots b_p . d_1 \ldots d_q}$, that is:

$$\left\{ \frac{p}{2^k} \mid p, k \in \mathbb{N} \right\}$$

Note that this is a countable set.

Finding Non-Empty Countable Model

$$\neg T(x,x)$$
$$\neg T(x,y) \vee \neg T(y,z) \vee T(x,z)$$
$$\neg T(x,y) \vee (T(x,g(x,y)) \wedge T(g(x,y),y))$$
$$T(x,h(x))$$

Theorem ensures we can find interpretation of g, h.
One possibility: $g(x,y) = (x+y)/2 \quad h(y) = y+1$
Since we have no constant and do not wish to have an empty domain, just pick any element as the starting point. Say, 0.
Apply closure under operations. Here they are all Skolem operations, but in general we use all operations we have, original or Skolem. Describe the set generated in this way.
Answer: The set of all non-negative numbers representable in binary notation $\overline{b_1 \ldots b_p . d_1 \ldots d_q}$, that is:

$$\left\{ \frac{p}{2^k} \mid p, k \in \mathbb{N} \right\}$$

Note that this is a countable set. Try also $g(x,y) = x + 1/(1+y-x)$

Herbrand (Term) Model: A Generic Countable Model

Instead of looking at arbitrary countable domains and functions on them, we show we can consider a more special class of structures: *ground term models*.

In these models the domain the set of expressions (group terms) built from constants and function symbols, and operations as just constructors.

Remember (D, α) is given by $D = \mathbb{R}$ (real numbers) and

- $\alpha(a) = 0$, $\alpha(b) = 1$
- $\alpha(f)(x, y) = x + y$
- $\alpha(T) = \{(x, y) | x \leq y\}$

The smallest substructure is given by $D_0 = \emptyset$,
$D_{i+1} = \{0, 1\} \cup \{x + y | x, y \in D_i\}$, $D^* = \bigcup_{i \geq 0} D_i$.

This is precisely the set of values of all expressions built from $0, 1$ and $+$.

In general, the least substructure is the set of values of ground terms:

$$D^* = \{[\![t]\!]^\alpha \mid t \in GT_{\mathcal{L}}\}$$

$GT_{\mathcal{L}}$ is the set of all ground terms (terms without variables) in language $\mathcal{L}$

Values of Ground Terms Induce Smallest Substructure

$GT_{\mathcal{L}}$ is the least set such that if $f \in \mathcal{L}$, $ar(f) = n$ $(n \geq 0)$ and $t_1, ..., t_n \in GT_{\mathcal{L}}$ then $f(t_1, ..., t_n) \in GT_{\mathcal{L}}$.
In other words, define $GT^0 = \emptyset$ and

$$GT^{i+1} = \{f(t_1, ..., t_n) \mid f \in \mathcal{L} \wedge t_1, ..., t_n \in GT^i\}$$

Then the set of all ground terms is $\bigcup_{i \geq 0} GT^i$

- GT^i is the set of terms of height (depth) at most $i - 1$

Compare to: $D_0 = \emptyset$, $D_{i+1} = \bigcup_{f \in \mathcal{L}_F} \{\alpha(f)(x_1, ..., x_n) | x_1, ..., x_n \in D_i\}$

Values of Ground Terms Induce Smallest Substructure

$GT_{\mathcal{L}}$ is the least set such that if $f \in \mathcal{L}$, $ar(f) = n$ $(n \geq 0)$ and $t_1, ..., t_n \in GT_{\mathcal{L}}$ then $f(t_1, ..., t_n) \in GT_{\mathcal{L}}$.
In other words, define $GT^0 = \emptyset$ and

$$GT^{i+1} = \{f(t_1, ..., t_n) \mid f \in \mathcal{L} \wedge t_1, ..., t_n \in GT^i\}$$

Then the set of all ground terms is $\bigcup_{i \geq 0} GT^i$

- GT^i is the set of terms of height (depth) at most $i - 1$

Compare to: $D_0 = \emptyset$, $D_{i+1} = \bigcup_{f \in \mathcal{L}_F} \{\alpha(f)(x_1, ..., x_n) | x_1, ..., x_n \in D_i\}$
By induction we prove easily

$$D_i = \{[\![t]\!]^\alpha \mid t \in GT^i\}$$

Therefore, $D^* = \{[\![t]\!]^\alpha \mid t \in GT_{\mathcal{L}}\}$

Values of Ground Terms Induce Smallest Substructure

$GT_{\mathcal{L}}$ is the least set such that if $f \in \mathcal{L}$, $ar(f) = n$ $(n \geq 0)$ and $t_1, ..., t_n \in GT_{\mathcal{L}}$ then $f(t_1, ..., t_n) \in GT_{\mathcal{L}}$.
In other words, define $GT^0 = \emptyset$ and

$$GT^{i+1} = \{f(t_1, ..., t_n) \mid f \in \mathcal{L} \wedge t_1, ..., t_n \in GT^i\}$$

Then the set of all ground terms is $\bigcup_{i \geq 0} GT^i$

- GT^i is the set of terms of height (depth) at most $i - 1$

Compare to: $D_0 = \emptyset$, $D_{i+1} = \bigcup_{f \in \mathcal{L}_F} \{\alpha(f)(x_1, ..., x_n) | x_1, ..., x_n \in D_i\}$
By induction we prove easily

$$D_i = \{[\![t]\!]^\alpha \mid t \in GT^i\}$$

Therefore, $D^* = \{[\![t]\!]^\alpha \mid t \in GT_{\mathcal{L}}\}$
How to define meaning of $f \in \mathcal{L}$ as function $GT^n_{\mathcal{L}} \to GT_{\mathcal{L}}$

Interpreting Functions on Ground Terms

Given a language $\mathcal{L}$ we are defining an interpretation $(GT_{\mathcal{L}}, \alpha_H)$. If there are no constants, invent a fresh constant a_0 and add it into $\mathcal{L}$.
For function symbols f, we just let

$$\alpha_H(f)(t_1, ..., t_n) = f(t_1, ..., t_n)$$

because we can always build a larger term.
This definition does not depend on the original model (D, α).

We next want to define $\alpha_H(R)$ for each relation symbols $R \in \mathcal{L}$
Idea: define the truth value following the truth value in (D, α)

$$\alpha_H(R) = \{(t_1, ..., t_n) \mid ([\![t_1]\!]^\alpha, ..., [\![t_n]\!]^\alpha) \in \alpha(R)\}$$

To determine if relation holds on ground terms, just check if it holds on their values.
It is in this step that we used the original structure (D, α) to define the new structure $(GT_{\mathcal{L}}, \alpha_H)$. We postponed evaluation to relations.

Revisiting Example of Dense Orders

$$\neg T(x,x)$$
$$\neg T(x,y) \vee \neg T(y,z) \vee T(x,z)$$
$$\neg T(x,y) \vee (T(x,g(x,y)) \wedge T(g(x,y),y))$$
$$T(x,h(x))$$

Use the model $(\mathbb{R}, \alpha)$ in which T is $<$, $g(x,y) = (x+y)/2$, $h(y) = y+1$ to define Herbrand model $(GT_{\mathcal{L}}, \alpha_H)$. Add fresh constant c.
Define

- $\alpha_H(c)$
- $\alpha_H(g)$
- $\alpha_H(h)$
- $\alpha_H(T)$

Example: why a formula holds in the ground model

Now use this definition of $\alpha_H(T)$.
Take any formula, say

$$\neg T(x,y) \vee (T(x,g(x,y)) \wedge T(g(x,y),y))$$

We wonder if it holds in $(GT_{\mathcal{L}}, \alpha_H)$. Let $x, y, z \in GT_{\mathcal{L}}$. Say $x = c$, $y = h(c)$. Why does

$$\neg T(c,h(c)) \vee (T(c,g(c,h(c))) \wedge T(g(c,h(c)),h(c)))$$

hold?

Example: why a formula holds in the ground model

Now use this definition of $\alpha_H(T)$.
Take any formula, say

$$\neg T(x,y) \vee (T(x,g(x,y)) \wedge T(g(x,y),y))$$

We wonder if it holds in $(GT_{\mathcal{L}}, \alpha_H)$. Let $x, y, z \in GT_{\mathcal{L}}$. Say $x = c$, $y = h(c)$. Why does

$$\neg T(c,h(c)) \vee (T(c,g(c,h(c))) \wedge T(g(c,h(c)),h(c)))$$

hold?
Because the same formula holds in the original structure. We defined $[\![T]\!]^{\alpha_H}$ so that

$$(c,h(c)) \in [\![T]\!]^{\alpha_H} \quad \Leftrightarrow \quad ([\![c]\!]^{\alpha}, [\![h(c)]\!]^{\alpha}) \in [\![T]\!]^{\alpha}$$

Herbrand Model is a Model of Same Universal Formulas

Lemma

For every quantifier-free formula $G(x_1, ..., x_n)$, *if* $\alpha_H(x_i) = t_i$ *then*

$$[\![G(x_1, ..., x_n)]\!]^{\alpha_H} \Leftrightarrow [\![G(x_1, ..., x_n)]\!]^{\alpha[x_i := \alpha(t_i)]_{i=1}^n}$$

Proof by induction, using the definition of $\alpha_H(R)$ in the base cases.

Theorem (Herbrand)

Let (D, α) *be a model of a set* S *of universal first-order formulas in the language* $\mathcal{L}$ *containing at least one constant. Then* $(GT_{\mathcal{L}}, \alpha_H)$ *is also a model of these formulas.*

Proof. Let $F \in S$ be of the form $\forall x_1, ..., x_n . G(x_1, ..., x_n)$. Then F holds in (D, α). Let $t_1, ..., t_n \in GT_{\mathcal{L}}$ be arbitrary. Then by the above lemma,

$$[\![G(x_1, ..., x_n)]\!]^{\alpha_H[x_i := t_i]_{i=1}^n} \Leftrightarrow [\![G(x_1, ..., x_n)]\!]^{\alpha[x_i := \alpha(t_i)]}$$

Last formula is true because F holds in (D, α). So, F holds in $(GT_{\mathcal{L}}, \alpha_H)$.

Viewing Herbrand Model as Propositional Model

Set S of universal formulas. Suppose we write universal variables as free variables. There is a model (D, α) if and only if there is Herbrand model $(GT_{\mathcal{L}}, \alpha_H)$.
How do we check if a set S has some Herbrand model? Function symbol interpretations are fixed. Need to check if there exists interpretation of each relation symbol R such that

$$\forall G \in S. \forall t_1, ..., t_n \in GT_{\mathcal{L}}. \ [\![G[x_1 := t_1, ..., x_n := t_n]]\!]^{\alpha_H} = true$$

Expand all these universal quantifiers:

$$S' = \{G[x_1 := t_1, ..., x_n := t_n] \mid G \in S\}$$

Then S holds in $GT_{\mathcal{L}}$ if and only if S' holds in $GT_{\mathcal{L}}$. We have countable domain $GT_{\mathcal{L}}$ and allow countable sets, so we instantiated.
S' has no variables, so it is like a propositional model.

Propositions with Long Names

For each relation symbol R define Herbrand atoms (ground instances):

$$HA = \{R(t_1, ..., t_n) \mid ar(R) = n, t_1, ..., t_n \in GT_{\mathcal{L}}\}$$

Then S' is a set of propositional formulas over the countable set HA.
Moreover, S' has a model if and only if each finite subset of S' has a model (compactness).
A finite subset has a model if and only if propositional resolution does not derive empty clause.

A set of FOL formulas is unsatisfiable if and only if for its skolemization there is a finite subset of ground instances on which resolution derives empty clause.

Naive Semidecision Procedure for FOL Satisfiability

For increasingly large size $N = 0, 1, 2, \ldots$:

1. instantiate a set of clauses with all terms of size up to N
2. check if the resulting finite set of propositional formulas is satisfiable (can use resolution, or a SAT solver)

Resolution for FOL

Instead of instantiating and then doing resolution on all propositional (ground) instances, do resolution using **unification** on first-order clauses.

A Resolution-Based Prover: **E** by Stephan Schulz

The web page with easy installation instructions and manual:

- `http://www4.informatik.tu-muenchen.de/~schulz/E/E.html`

Theorem proving problems, links to competition, other provers:

- `http://www.tptp.org`

Give Our Example to Automated Prover

Our example in math:

$$\neg\Big(\quad (\forall x.\exists y.\ R(x,y)) \wedge (\forall x.\forall y.\ (R(x,y) \Rightarrow \forall z.\ R(x,f(y,z)))) \wedge (\forall x.\ (P(x) \vee P(f(x,a)))) \Rightarrow \forall x.\exists y.\ (R(x,y) \wedge P(y))\Big)$$

Our example in TPTP ASCII format:

```
fof(ax1,axiom, ![X]: ?[Y]: r(X,Y)).
fof(ax2,axiom, ![X]: ![Y]: (r(X,Y) => ![Z]: r(X,f(Y,Z)))).
fof(ax3,axiom, ![X]: (p(X) | p(f(X,a)))).
fof(c,conjecture, ![X]: ?[Y]: (r(X,Y) & p(Y))).
```

$\wedge$	$\vee$	$\neg$	$\Rightarrow$	$\Leftrightarrow$	$\forall$	$\exists$
&	|	$\sim$	=>	<=>	!	?

Example Formula in First-Order Logic

model of a formula = interpretation (structure) that makes a formula true

$$\begin{aligned}\neg(\quad & (\forall x.\exists y.\ R(x,y)) \wedge \\ & (\forall x.\forall y.\ (R(x,y) \Rightarrow \forall z.\ R(x,f(y,z)))) \wedge \\ & (\forall x.\ (P(x) \vee P(f(x,a)))) \\ & \Rightarrow \forall x.\exists y.\ (R(x,y) \wedge P(y)))\end{aligned}$$

After normal form and Skolemization we obtain these first-order clauses:

$$\begin{aligned}& R(x, g_1(x)) \\ & \neg R(x,y) \vee R(x, f(y,z)) \\ & P(x) \vee P(f(x,a)) \\ & \neg R(c_0, y) \vee \neg P(y)\end{aligned}$$

- variables are implicitly $\forall$ quantified; there are no $\exists$ quantifiers
- each clause is disjunction of literals (atomic formulas or their negation)
- from any model of these clauses we can obtain model for the original formula (just ignore interpretation of Skolem constants g_1, c_0)

Applying Resolution

1. $R(x, g_1(x))$
2. $\neg R(x, y) \vee R(x, f(y, z))$
3. $P(x) \vee P(f(x, a))$
4. $\neg R(c_0, y) \vee \neg P(y)$

Applying Resolution

1. $R(x, g_1(x))$
2. $\neg R(x, y) \vee R(x, f(y, z))$
3. $P(x) \vee P(f(x, a))$
4. $\neg R(c_0, y) \vee \neg P(y)$
5. (1,2): $R(x, f(g_1(x), z))$

Applying Resolution

1. $R(x, g_1(x))$
2. $\neg R(x, y) \vee R(x, f(y, z))$
3. $P(x) \vee P(f(x, a))$
4. $\neg R(c_0, y) \vee \neg P(y)$
5. (1,2): $R(x, f(g_1(x), z))$
6. (1,4): $\neg P(g_1(c_0))$

Applying Resolution

1. $R(x, g_1(x))$
2. $\neg R(x, y) \vee R(x, f(y, z))$
3. $P(x) \vee P(f(x, a))$
4. $\neg R(c_0, y) \vee \neg P(y)$
5. (1,2): $R(x, f(g_1(x), z))$
6. (1,4): $\neg P(g_1(c_0))$
7. (3,6): $P(f(g_1(c_0), a))$

Applying Resolution

1. $R(x, g_1(x))$
2. $\neg R(x, y) \vee R(x, f(y, z))$
3. $P(x) \vee P(f(x, a))$
4. $\neg R(c_0, y) \vee \neg P(y)$
5. (1,2): $R(x, f(g_1(x), z))$
6. (1,4): $\neg P(g_1(c_0))$
7. (3,6): $P(f(g_1(c_0), a))$
8. : $\neg R(c_0, f(g_1(c_0), a))$

Applying Resolution

1. $R(x, g_1(x))$
2. $\neg R(x, y) \vee R(x, f(y, z))$
3. $P(x) \vee P(f(x, a))$
4. $\neg R(c_0, y) \vee \neg P(y)$
5. (1,2): $R(x, f(g_1(x), z))$
6. (1,4): $\neg P(g_1(c_0))$
7. (3,6): $P(f(g_1(c_0), a))$
8. : $\neg R(c_0, f(g_1(c_0), a))$
9. : $\emptyset$

Applying Resolution

1. $R(x, g_1(x))$
2. $\neg R(x, y) \vee R(x, f(y, z))$
3. $P(x) \vee P(f(x, a))$
4. $\neg R(c_0, y) \vee \neg P(y)$
5. (1,2): $R(x, f(g_1(x), z))$
6. (1,4): $\neg P(g_1(c_0))$
7. (3,6): $P(f(g_1(c_0), a))$
8. : $\neg R(c_0, f(g_1(c_0), a))$
9. : $\emptyset$

Proof found!