Term Models for First-Order Logic

Example Formula in First-Order Logic

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

$$
\begin{aligned}
\neg\Big(\quad & (\forall x.\exists y.\ R(x,y)) \land \\
& (\forall x.\forall y.\ (R(x,y) \to \forall z.\ R(x,f(y,z)))) \land \\
& (\forall x.\ (P(x) \lor P(f(x,a)))) \\
& \to \forall x.\exists y.\ (R(x,y) \land P(y))\Big)
\end{aligned}
$$

Example Formula in First-Order Logic

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

$$
\begin{aligned}
\neg(\quad & (\forall x. \exists y.\ R(x,y)) \land \\
& (\forall x. \forall y.\ (R(x,y) \to \forall z.\ R(x, f(y,z)))) \land \\
& (\forall x.\ (P(x) \lor P(f(x,a)))) \\
& \to \forall x. \exists y.\ (R(x,y) \land P(y)))
\end{aligned}
$$

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

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

- 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

$$
\begin{array}{ll}
\neg(& (\forall x.\exists y.\ R(x,y)) \land \\
 & (\forall x.\forall y.\ (R(x,y) \rightarrow \forall z.\ R(x, f(y,z)))) \land \\
 & (\forall x.\ (P(x) \lor P(f(x,a)))) \\
 & \rightarrow \forall x.\exists y.\ (R(x,y) \land P(y)))
\end{array}
$$

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

$$
\begin{array}{l}
R(x, g_1(x)) \\
\neg R(x,y) \lor R(x, f(y,z)) \\
P(x) \lor P(f(x,a)) \\
\neg R(c_0, y) \lor \neg P(y)
\end{array}
$$

- 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?

Remark on Notation

We can view n ary relation on D either as a function

$$r_f : D^n \to \{0, 1\}$$

which is how we defined it, so far, or, more conventionally, as the subets

$$r \subseteq D^n$$

through isomorphism such that

$$(d_1, \dots, d_n) \in r \iff r_f(d_1, \dots, d_n) = 1$$

In the following slides we will use the view $r \subseteq D^n$

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 $D'_1 = \{1, 2, \dots\}$ 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) $x, y \in \mathbb{Z} \to x + y \in \mathbb{Z}$.

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^*. $\square$

Theorem (Downward Löwenheim-Skolem)

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. $\square$

Example: Dense Orders

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

$$
\begin{array}{l}
\forall x. \neg T(x,x) \\
\forall x \forall y \forall z. T(x,y) \land T(y,z) \rightarrow T(x,z) \\
\forall x \forall y. (T(x,y) \rightarrow \exists z. (T(x,z) \land T(z,y))) \\
\forall x \exists y. T(x,y)
\end{array}
$$

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:

$$
\begin{array}{l}
\forall x. \neg T(x,x) \\
\forall x \forall y \forall z. T(x,y) \wedge T(y,z) \rightarrow T(x,z) \\
\forall x \forall y. (T(x,y) \rightarrow \exists z. (T(x,z) \wedge T(z,y))) \\
\forall x \exists y. T(x,y)
\end{array}
$$

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 $h(x)$:

$$
\begin{array}{l}
\neg T(x,x) \\
\neg T(x,y) \vee \neg T(y,z) \vee T(x,z) \\
\neg T(x,y) \vee (T(x,g(x,y)) \wedge T(g(x,y),y)) \\
T(x,h(x))
\end{array}
$$

Finding Non-Empty Countable Model

$$
\begin{aligned}
&\neg T(x,x)\\
&\neg T(x,y) \lor \neg T(y,z) \lor T(x,z)\\
&\neg T(x,y) \lor (T(x,g(x,y)) \land T(g(x,y),y))\\
&T(x,h(x))
\end{aligned}
$$

Theorem ensures we can find interpretation of g, h.
One possibility:

Finding Non-Empty Countable Model

$$
\begin{aligned}
&\neg T(x,x)\\
&\neg T(x,y) \vee \neg T(y,z) \vee T(x,z)\\
&\neg T(x,y) \vee (T(x,g(x,y)) \wedge T(g(x,y),y))\\
&T(x,h(x))
\end{aligned}
$$

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

$$
\begin{array}{l}
\neg T(x,x) \\
\neg T(x,y) \vee \neg T(y,z) \vee T(x,z) \\
\neg T(x,y) \vee (T(x,g(x,y)) \wedge T(g(x,y),y)) \\
T(x,h(x))
\end{array}
$$

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

$$\begin{array}{l} \neg T(x,x) \\ \neg T(x,y) \vee \neg T(y,z) \vee T(x,z) \\ \neg T(x,y) \vee (T(x,g(x,y)) \wedge T(g(x,y),y)) \\ T(x,h(x)) \end{array}$$

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 $GT_{\mathcal{L}}^n \to GT_{\mathcal{L}}$

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

$$
\begin{array}{l}
\neg T(x,x) \\
\neg T(x,y) \vee \neg T(y,z) \vee T(x,z) \\
\neg T(x,y) \vee (T(x,g(x,y)) \wedge T(g(x,y),y)) \\
T(x,h(x))
\end{array}
$$

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.
Propositional resolution derives an empty clause iff rules with ground instantiation and resolution derive an empty clause.

Theorem. A set of FOL formulas is unsatisfiable if and only if it is possible to derive empty clause from it using resolution with instantiation.

We can also show: ground instantiation with resolution on ground clauses derives an empty clause if and only iff resolution with unification does.

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

The principles behind Skolemization, transformation to clauses lead to automated theorem provers (ATPs).

Example: E prover. Try E by downloading and building it from:
`https://github.com/eprover/eprover`

To try an example file example.p, use e.g.

```
eprover --auto --proof-object example.p
```

Theorem proving problems, links to competition, other provers:

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

Let us Give Our Example to E

Our example in math:

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

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))).
```

$\wedge$	$\vee$	$\neg$	$\rightarrow$	$\leftrightarrow$	$\forall$	$\exists$
&	|	~	=>	<=>	!	?