Compactness

Enumerability of Models

Finite Models

Size of Models

Definability of Models

Enumerability Theorem

Compactness

Last time

Review
Outline

Theories

 Congruence Closure

Interpretations Between Theories

Sources:

Sections 2.6 through 2.7 of Enderton.

http://theory.stanford.edu/~zm/new-papers.html


● Interprettations Between Theories

● Congruence Closure

● Theories
Size of Models

The cardinality \(|L|\) of a language \(L\) is the least infinite cardinal greater than or equal to the number of symbols in the signature of \(L\).

The cardinality \(|\mathcal{M}|\) of a model \(\mathcal{M}\) is the cardinality of its domain \(\text{dom}(\mathcal{M})\).

**Löwenheim-Skolem (LS) Theorem**

Let \(\Gamma\) be a satisfiable set of formulas in a language \(L\), then \(\Gamma\) is satisfiable in some model of cardinality \(\kappa \leq |L|\).

**Proof**

By soundness, \(\Gamma\) is consistent, and is thus satisfiable by the model constructed in the proof of the completeness theorem. But the domain of that model is \(\mathcal{M}/E\) which has cardinality \(\leq |\mathcal{M}|\), and \(|\mathcal{M}| = |L|\).
How is this possible?

that there are "uncountably" many sets.

is a countable model. But we can prove, starting with the axioms of set theory,

have a model. Because the signature of the language of set theory is finite, there
here

Let $A$ be your favorite set of axioms for set theory. If they are consistent, they

"Skolem's paradox"
Let your favorite set of axioms for set theory, $\mathcal{A}$, be your favorite set of axioms for set theory. If they are consistent, they have a model. Because the signature of the language of set theory is finite, there is a countable model, $M$, of $\mathcal{A}$. But we can prove, starting with the axioms of set theory, that there are "uncountably" many sets. How is this possible?

The answer is that in the countable model of set theory, things do not correspond to what we normally think of as the model of set theory. Thus, the model of the "natural numbers" in the countable model cannot be put in one-to-one correspondence with all of the elements of the model. But this does not mean that the size of the model is truly uncountable.
Theorem

Let \( \mathbf{L} \) be a set of formulas in a language \( \mathcal{L} \), and assume that \( \mathbf{L} \) is satisfiable. Then for every cardinal \( \kappa \), there is a model \( M \) of cardinality \( \kappa \). But any model must have at least cardinality \( \kappa \). Thus there is a cardinal of cardinality \( \kappa \). By compactness, \( \bigwedge \mathbf{L} \) is satisfiable. By the LSTheorem, there is a model of \( \mathbf{L} \) is satisfiable in some infinite model. Then, any finite subset \( \mathbf{L} \) is satisfiable in \( M \). By the LSTheorem, there is a model of cardinality \( \kappa \). But any model must have at least cardinality \( \kappa \). Thus there is a model of cardinality \( \kappa \).

Proof

Let \( \mathbf{L} \) be a set of formulas in a language \( \mathcal{L} \), and assume that \( \mathbf{L} \) is satisfiable. Then for every cardinal \( \kappa \), there is a model of cardinality \( \kappa \). By compactness, \( \bigwedge \mathbf{L} \) is satisfiable. By the LSTheorem, there is a model of cardinality \( \kappa \). But any model must have at least cardinality \( \kappa \). Thus there is a model of cardinality \( \kappa \).
Corollary

If \( M \) is an infinite model for a countable language, then for any infinite cardinal \( \kappa \), there is a model \( M_0 \) of cardinality \( \kappa \) such that \( M \equiv M_0 \).

Proof

Let \( \mathcal{L} \) be the set of all sentences true in \( M \). By the corollary above, \( \mathcal{L} \) has a model of cardinality \( \kappa \). But note that for every sentence \( \varphi \), either \( \mathcal{L} \in \varphi \) or \( \mathcal{L} \not\in \varphi \) (why?). Thus, \( \mathcal{L} \not\in \varphi \) or \( \mathcal{L} \in \varphi \).

Two models \( M \) and \( M' \) are elementarily equivalent if for any sentence \( \varphi \), \( M \vDash \varphi \) if and only if \( M' \vDash \varphi \). Two models, \( M \) and \( M' \), are elementarily equivalent if and only if they have the same set of sentences in a countable language. Thus, if \( M \) has some model of every infinite cardinality, then \( M \) has some model of cardinality \( \kappa \).
Theories

Last time, we defined a theory as a set of first-order sentences. Why?

Theories

Examples

For a given signature, the smallest possible theory consists of exactly the valid sentences over that signature.

Theories

Thus, if \( \emptyset \) is a theory iff \( \emptyset \) is a set of sentences and \( \emptyset \) is closed under logical implication.

Thus, \( \emptyset \) is a theory iff \( \emptyset \) is a set of sentences and if \( \emptyset \) is closed under logical implication.

For this lecture, we will refine our definition to be a set of first-order sentences.

Theories

Theories

• The largest theory for a given signature is the set of all sentences. It is the only unsatisfiable theory. Why?
For a class $\mathcal{K}$ of models over a given signature, define the theory of $\mathcal{K}$ as

$$\{ \varphi \mid \varphi \text{ is a } \mathcal{K}\text{-sentence which is true in every model in } \mathcal{K} \}$$

Theorem

Then $\mathcal{C}n = \text{ThMod} I$. Define the set $\mathcal{C}n$ of consequences of $I$ to be $\{ \varphi \mid I \models \varphi \}$.

Suppose $I$ is a set of sentences.

Proof

The $\mathcal{C}n$ is indeed a theory.

Then $\mathcal{C}n = \text{ThMod} I$. We know that $I \models \varphi$ for each $\varphi$ in $\mathcal{K}$, and thus $\varphi \in \text{Th} \mathcal{K}$ for each $\varphi$ in $\mathcal{K}$. It follows that $\mathcal{C}n = \text{Th} \mathcal{K}$.
A theory $T$ is complete if for every sentence $\phi$, either $T \models \phi$ or $T \models \neg \phi$. It follows that

\[ \text{If } T \models \phi \text{ and since } T \models \psi \text{, then } T \models (\phi \lor \psi). \]

Thus, $T \models \top$. By completeness, we have that there exists $T \models \psi$ such that

Clearly, $T \models \psi$. Therefore, there exists an $T \models \psi$ such that

\[ \text{If } T \models \phi \text{ is initially axiomatizable, then for some sentence } \psi \text{, } T \models \phi = T \models \psi. \]

\[ \text{Proof} \]

\[ \text{If } T \models \phi \text{ is initially axiomatizable, then for some finite set of sentences } \phi \text{, } T \models \phi = T \models \psi. \]

\[ \text{Therefore, } T \models \phi \text{ is initially axiomatizable iff there exists a finite set } \phi \text{, such that } T \models \phi. \]

\[ \text{A theory } T \text{ is initially axiomatizable iff there exists a decidable set of sentences such that } T \models \phi. \]

\[ A \text{ theory } T \text{ is axiomatizable iff there exists a decidable set of sentences such that } T \models \phi. \]

\[ \text{Note that if } T \text{ is a model, then } T\{N\} \text{ is complete. In fact, for a class of models, } \forall \text{ if } T \text{ is complete, then for every sentence } \phi, \text{ either } \phi \text{ or } \neg \phi \text{, such that } T \models \phi \text{, or } T \models \neg \phi. \]

\[ \text{A theory } T \text{ is complete iff for every sentence } \phi, \text{ either } \phi \text{ or } \neg \phi \text{, such that } T \models \phi \text{, or } T \models \neg \phi. \]
Theories

Using the above terminology, we can restate our earlier results as follows:

A complete axiomatizable theory (in a reasonable language) is decidable.

A finitely axiomatizable theory (in a reasonable language) is effectively enumerable.

Our results about theories can be summarized in the following diagram:

Axiomatizable  Effectively Enumerable

Decidable  Finitely axiomatizable

Complete  If
Theorem. Isomorphic models are elementarily equivalent. But these models must be isomorphic, and by the homomorphism theorem, if they are infinite, there exist (by LST) elementarily equivalent models of cardinality $\aleph_0$. Since $M_\omega \equiv M_\omega$, $M$ and $M_0$ are isomorphic, and by the homomorphism theorem, isomorphic models are elementarily equivalent.

**Proof.**

Then $T$ is complete.

All models of $T$ are infinite.

$T$ is $\aleph_0$-categorical for some infinite cardinal $\aleph_0$.

Let $T$ be a theory in a countable language such that having cardinality $\aleph_0$ are isomorphic. For a theory $T$ and a cardinal $\aleph_0$, say that $T$ is $\aleph_0$-categorical if all models of $T$.
Validity and Satisfiability Modulo Theories

Given a $\mathcal{L}$-theory $T$, a $\mathcal{L}$-formula $\phi$ is $T$-valid if $\phi[M[s]] = true$ for all models $M$ of $T$ and all variable assignments $s$.

2. $T$-satisfiable if there exists some model $M$ of $T$ and variable assignment $s$ such that $\phi[M[s]] = true$.

3. $T$-unsatisfiable if for all models $M$ of $T$ and all variable assignments $s$, $\phi[M[s]] \neq true$.

The validity problem for $T$ is the problem of deciding, for each $\mathcal{L}$-formula $\phi$, whether $\phi$ is $T$-valid.

The satisfiability problem for $T$ is the problem of deciding, for each $\mathcal{L}$-formula $\phi$, whether $\phi$ is $T$-satisfiable.

Similarly, one can define the quantifier-free validity problem and the quantifier-free satisfiability problem for a $\mathcal{L}$-theory $T$ by restricting the formula to be quantifier-free.
Validity and Satisfiability Modulo Theories

A decision problem is decidable if there exists an effective procedure which always terminates with an answer for any given instance of the problem.

For example, the validity problem for a $L$-theory $T$ is decidable if there exists an effective procedure which always terminates with an answer for any given instance of the problem.

We will consider a few examples of theories which are of particular interest in

Verification applications.

Note that validity problems can always be reduced to satisfiability problems:

$\phi$ is valid iff $L$-unsatisfiable.

For example, the validity problem for a $L$-theory $T$ is decidable if there exists an effective procedure which always terminates with an answer for any given instance of the problem.

A decision problem is decidable if there exists an effective procedure which always terminates with an answer for any given instance of the problem.

Validity and Satisfiability Modulo Theories
The theory of equality is the theory \( \Gamma \).

Note that the exact set of sentences in \( \Gamma \) depends on the signature in question.

The theory does not restrict the possible values of symbols in any way. For this reason, it is sometimes called the theory of equality with uninterpreted functions (EUF).

The satisfiability problem for conjunctions of literals in the theory of equality with uninterpreted functions is just the satisfiability problem for first-order logic, which is undecidable.

The satisfiability problem for \( \Gamma \) is decidable in polynomial time using congruence closure.

\( \Gamma \) is sometimes called the theory of equality with uninterpreted functions (EUF).
Let $\mathbb{Z}$ be the standard model of the integers with domain $\mathbb{Z}$.

Then $T_\mathbb{Z}$ is defined to be $\text{Th}_\mathbb{A}\mathbb{Z}$.

Let $\mathbb{Z}$ be the standard model of the integers with domain $\mathbb{Z}$.

Then $T_\mathbb{Z}$ is defined to be $\text{Th}_\mathbb{A}\mathbb{Z}$.

As showed by Presburger in 1929, the validity problem for $T_\mathbb{Z}$ is undecidable, but its complexity is triply-exponential.

The satisfiability problem for $T_\mathbb{Z}$ is "only" $\mathbb{NP}$-complete.

In fact, even the quantifier-free satisfiability problem for $T_\mathbb{Z}$ is undecidable: $\mathbb{Z}_\mathbb{L}$ is undecidable (a consequence of Gödel's incompleteness theorem).

The satisfiability problem for $T_\mathbb{Z}$ is undecidable (a consequence of Gödel's incompleteness theorem).
The Theory of Reals

Let $\mathcal{L}_R$ be the signature $(0, 1, +, \times, \leq)$. Let $\mathcal{A}_R$ be the standard model of the reals with domain $\mathbb{R}$. Then $T_R$ is defined to be $\mathcal{L}_R$. The satisfiability problem for $T_R$ is decidable, but the complexity is doubly-exponential.

In contrast to the theory of integers, the satisfiability problem for $\mathcal{L}_R$ is decidable.

Let $\mathcal{L}_R$ and $\mathcal{A}_R$ be defined in the obvious way. Let $\mathcal{L}_R$ and $\mathcal{A}_R$ be defined in the obvious way.

The quantifier-free satisfiability problem for conjunctions of literals (atomic formulas or their negations) in $T_R$ is solvable in polynomial time, though exponential methods (like Simplex or Fourier-Motzkin) often perform better in practice.
Let $A$ be the signature $(\text{read}, \text{write})$.

Let $\forall A$ be the following axioms:

\[
\begin{align*}
(q = a & \iff ((\neg q, q) \text{ read} = (\neg q, q) \text{ read}) \iff A)) A \land A \land A \\
((\neg q) \text{ read} = (q, q) \text{ read}) & \iff \neg q \iff A \land A \land A \\
(a = ((\neg q, q) \text{ write}) \iff q \iff A) & \iff A \land A \\
\text{Then } V_A & = \forall A.
\end{align*}
\]

The satisfiability problem for $\forall A$ is undecidable, but the quantifier-free satisfiability problem for $\forall A$ is decidable (the problem is NP-complete).
An inductive data type (IDT) defines one or more constructors, and possibly also selectors and testers. Therefore, inductively defined types (IDTs) include all the types in the examples above.

The theory of an inductive data type associates a function symbol with each constructor and selector, and a predicate symbol with each tester.

For IDTs with a single constructor, a conjunction of literals is decidable in polynomial time. For more general IDTs, the problem is NP-complete, but reasonably efficient algorithms exist in practice.

Example:

\[
\begin{align*}
\text{List} & : \text{list} \\
\text{Cons} & : \text{int} \times \text{list} \\
\text{Car} & : \text{list} \rightarrow \text{int} \\
\text{Cdr} & : \text{list} \rightarrow \text{list} \\
\text{IsCons} & : \text{list} \rightarrow \text{bool} \\
\text{IsNil} & : \text{list} \rightarrow \text{bool}
\end{align*}
\]
Some other interesting theories include:

- Theories of bit-vectors
- Fragments of set theory
Let $G = (V; E)$ be a directed graph such that for each vertex $v$ in $G$, the successors of $v$ are ordered.

Let $C$ be any equivalence relation on $V$. The congruence closure $\mathcal{C}$ of $C$ is the finest (most discriminating) equivalence relation on $V$ that contains $C$ and satisfies the following property for all vertices $v$ and $w$.

If $v \mathrel{\text{precedes}} w$, and for $1 \leq i \leq m$, $(v_i, w_i) \in C$, then $(v, w) \in \mathcal{C}$.

Often, the vertices are labeled by some labeling function $\lambda$. In this case, the property becomes:

Let $v$ and $w$ have successors $v_1, \ldots, v_k$ and $w_1, \ldots, w_l$ respectively. Let $\lambda$ be any labeling function. If $v \mathrel{\text{precedes}} w$, and for $1 \leq i \leq m$, $(v_i, w_i) \in C$, then $(v, w) \in \mathcal{C}$.

The congruence closure $\mathcal{C}$ of $C$ is the coarsest (least discriminating) equivalence relation on $V$ that contains $C$ and satisfies the following property for all vertices $v$ and $w$.

Let $\mathcal{C}$ be any equivalence relation on $V$. Let $\mathcal{C} \mathrel{\mathcal{E}} (\Lambda)$ be a directed graph such that for each vertex $v$ in $\mathcal{C}$, the successors of $v$ are ordered. If $v \mathrel{\text{precedes}} w$, and for $1 \leq i \leq m$, $(v_i, w_i) \in C$, then $(v, w) \in \mathcal{C}$. The congruence closure $\mathcal{C}$ of $C$ is the finest (most discriminating) equivalence relation on $V$ that contains $C$ and satisfies the following property for all vertices $v$ and $w$.

If $v \mathrel{\text{precedes}} w$, and for $1 \leq i \leq m$, $(v_i, w_i) \in C$, then $(v, w) \in \mathcal{C}$.
A Simple Algorithm

1. Number the equivalence classes in $C_i$ consecutively from 1.
2. Assign to each vertex $v$ the number $(v)$ of the equivalence class containing $v$.
3. For each vertex $v$ construct a signature $s(v) = (v)(v_1)(v_2)...(v_k)$, where $v_1, v_2, ..., v_k$ are the successors of $v$.
4. Group the vertices into classes of vertices having equal signatures.
5. Let $C_{i+1}$ be the finest equivalence relation on $V$ such that two vertices equivalent under $C_i$ or having the same signature are equivalent under $C_{i+1}$.
6. If $C_{i+1} = C_i$, let $C = C_i$; otherwise increment $i$ and repeat.

Let $C_0 = 0$ and $C = 0$. Otherwise increment $i$ and repeat.
Recall that $T$ is the empty theory with equality over some signature containing only function symbols.

If $\mathfrak{S}$ is a set of ground-equalities and $\mathfrak{L}$ is a set of ground-disequalities, then the satisfiability of $\mathfrak{S} \cup \mathfrak{L}$ can be determined as follows. Let $G$ be a graph which corresponds to the abstract syntax trees of terms in $\mathfrak{S}$, and let $\tau_t$ denote the vertex of $G$ associated with the term $t$. Let $C$ be the equivalence relation on the vertices of $G$ induced by $\mathfrak{L}$. Let $\mathfrak{S} \nvdash \tau_s, \tau_t$, for each $s \neq t$. 

- $\mathfrak{S} \cup \mathfrak{L}$ is satisfiable if for each $s \neq t$, $(\tau_s, \tau_t) \notin C$. 

Recall that $T$ is the empty theory with equality over some signature containing $\mathfrak{L}$.
T union and end are abstract operations for manipulating equivalence classes.

union \((x; y)\) merges the equivalence classes of \(x\) and \(y\).

end \((x)\) returns a unique representative of the equivalence class of \(x\).

\(\forall \in J\) returns a unique representative of the equivalence classes of \(\forall \in J\).

\[\begin{align*}
\text{return true if } & (q) \text{ and } (v) = (n) \text{ for some } a \in I \\
\text{return false if } & (q,v) = (n) \\
\text{if } & \text{merge}(q,v) \\
\text{then } & \forall \in J \text{ from terms in } I \text{ and } \\
\text{remove some equality } & q = a \text{ from } I \\
\text{while } & J \neq \emptyset \\
\text{construct } & G(V; E) \text{ from terms in } I \\
\text{return true if } & \forall \text{ from } I, J \text{ and } \\
\text{find}(x) \text{ returns a unique representative of the equivalence class of } x. \\
\text{union}(x; y) \text{ merges the equivalence classes of } x \text{ and } y. \\
\end{align*}\]
An Algorithm for $\mathcal{T}_\varepsilon$

$\text{Merge}(a, b)$

\begin{align*}
\text{if } \text{find}(a) = \text{find}(b) \text{ then return;}
\end{align*}

Let $A$ be the set of all predecessors of all vertices equivalent to $a$;

Let $B$ be the set of all predecessors of all vertices equivalent to $b$;

\begin{align*}
\text{union}(a, b); \\
\text{foreach } x \in A \text{ and } y \in B \\
\text{if } \text{signature}(x) = \text{signature}(y) \text{ then } \text{Merge}(x, y);
\end{align*}
The Downey-Sethi-Tarjan Congruence Closure algorithm is more efficient. It makes use of some additional data structures and methods.

Additional Helper Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>union</td>
<td>Unions the list of vertices with at least one successor in equivalence class e.</td>
</tr>
<tr>
<td>delete</td>
<td>Removes (v, signature(v)) from the signature table if it is there.</td>
</tr>
<tr>
<td>enter</td>
<td>Stores (v, signature(v)) in a signature table.</td>
</tr>
<tr>
<td>query</td>
<td>If there is an entry (w, signature(w)) in the signature table and signature(w) = signature(v), then return w; otherwise, return -1.</td>
</tr>
<tr>
<td>list</td>
<td>Returns the list of vertices with at least one successor in equivalence class e.</td>
</tr>
</tbody>
</table>

The Downey-Sethi-Tarjan Congruence Closure algorithm is more efficient.
DST Algorithm

Construct $G(V; E)$ from terms in $\land$.

$\forall \in q \neq a$ for some $a \in \text{find}(q) = \text{find}(a)$.

if $\text{merge}(I_1)$:

return false!

else:

$\forall \in R$ and $\forall$

return true!
Merge

\[ \emptyset = \text{merge} \]

\[ ((q) \cup \text{query}(v)) \cup \text{query}(n) \]

\[ \text{delete} \ n \ \text{to pending} \]

\[ ((q) \cup \text{query}(v)) \cup \text{query}(n) \]

\[ \text{foreach n } \in \text{list} \]

\[ |((q) \cup \text{query}(v)) \cup \text{query}(n)| > |((q) \cup \text{query}(v))| \]

\[ \text{if } |((q) \cup \text{query}(v)) \cup \text{query}(n) \neq |((q) \cup \text{query}(v))| \]

\[ \text{foreach } (a, q) \in \text{combine} \]

\[ \emptyset = \text{pending} \]

\[ \text{delete } (n, \text{query}(v)) \ \text{to combine} \]

\[ \text{else add } (n, \text{query}(v)) \ \text{to combine} \]

\[ \text{if } \text{query}(v) \text{ then } \top = (n) \text{query}(v) \]

\[ \text{foreach n } \in \text{pending} \]

\[ \emptyset \neq \text{pending} \]

while \[ \emptyset \neq \text{pending} \]

\[ \text{pending} = \text{set of all vertices} \]

\[ \text{merge} \]

\[ \text{DST Algorithm} \]
Interpretations Between Theories

Given two theories, $T_0$ and $T_1$, sometimes it is possible to show that one of the theories is at least as powerful as the other.

We will consider more general cases in which the languages of the two theories differ. A simple case is when language $L_1$ and language $L_0$ are in the same language and one is a subset of the other. Given two theories, $T_0$ and $T_1$, sometimes it is possible to show that one of the theories is at least as powerful as the other.
A common practice in mathematical reasoning is to introduce a new piece of notation, defining it in terms of a formula not containing the new notation. Formally, defining it in terms of a formula not containing the new notation.

A function definition is a formula containing a function symbol and is a function symbol not in a signature. Technically, it is an abbreviation for

\[ \phi \Rightarrow \exists \gamma \, \exists \zeta \, \forall \alpha \, \alpha \in \mathcal{L} \wedge \gamma \vdash \phi \Rightarrow \exists \zeta \]
Proof

To show that (1) implies (2), note that \( \emptyset \models \emptyset \). Thus, if we take \( \emptyset \models \emptyset \)
There are more general ways in which one theory can be as strong as another theory in another language.

Example

The second theory is at least as strong as the first. To show this, we make the following observations:

The successor relation is defined by

\[ 0 = z + z \]

The set \( \{0\} \) is defined by \( 0 + 0 = 0 \).

An integer is nonnegative if it is the sum of four squares.

Consider the theory of natural numbers with 0 and successor (and on

The other hand the theory of \( \mathbb{Z} \) (integers) and the theory of \( \mathbb{N} \) (natural numbers with 0 and successor) and on

Thus, for example, the sentence

\[ 0 \neq x S x A \]

can be translated as

\[ 0 = z + x \leftarrow z \neq z + z \vee z = z \times z \]
Suppose \(0\) and \(1\) are signatures and \(1\) is a \(1\)-theory. An interpretation of \(0\) into \(1\) consists of the following three items:

1. A \(1\)-formula in which at most \(1\) variables occur free, such that

\[
A \models (x)^1 \nu
\]

2. A \(1\)-formula for each \(u\)-ary predicate symbol \(d\nu \in \mathcal{F}\) in which at most \(u\) variables occur free.

\[
A \models u \nu \leftarrow \cdots \leftarrow (1^n \nu)^u \nu; \cdots; (1^n \nu)A \Rightarrow 1_L \quad (!)
\]

3. A \(1\)-formula for each \(f\)-ary function symbol \(f \nu \in \mathcal{F}\) in which at most \(f\) variables occur free such that

\[
A \models \left( (x = 1 + u \nu \leftrightarrow (1 + u \nu, \cdots, 1^n \nu) f \nu) 1 + u \nu A \vee (x)^1 \nu \right) x \models (u \nu)^1 \nu \leftarrow \cdots \leftarrow (1^n \nu)^u \nu; \cdots; (1^n \nu)A \Rightarrow 1_L \quad (!!)
\]

For our previous example, we have

\[
\begin{align*}
1. (x = 1 + u \nu & \leftrightarrow (1 + u \nu, \cdots, 1^n \nu) f \nu) 1 + u \nu A \vee (x)^1 \nu \\
& x \models (x)^1 \nu \leftarrow \cdots \leftarrow (1^n \nu)^u \nu; \cdots; (1^n \nu)A \Rightarrow 1_L \quad (!!) \end{align*}
\]

Interpretations
Interpretations

Assume that \( \mathcal{L} \) is an interpretation and let \( \mathcal{M} \) be a model of \( \mathcal{L} \). There is a natural way to extract from \( \mathcal{M} \) a model of \( 0^\mathcal{M} \), as shown above.

In other words, \( \mathcal{L} \) is the set of all \( 0^\mathcal{M} \)-sentences true in every model \( \mathcal{M} \).

\[
\{ \mathcal{M} \in \mathcal{L} \text{ Mod } \mid \mathcal{M} \models \phi \} \text{ for } \phi \in 0^\mathcal{M}
\]

Define the set of \( 0^\mathcal{L} \)-sentences as

By condition (i), the definition of \( \cdot \) makes \( \mathcal{M} \models f \) sense.

By condition (ii), \( \emptyset \neq (\mathcal{M}^\mathcal{L}) \text{ dom } f \).

\[
(\mathcal{M}^\mathcal{L}) \text{ dom } f = (u_0, \ldots, u_n) \text{ where } u_0, \ldots, u_n \text{ are in dom } \mathcal{M}.
\]

\[
[\{ q_0, \ldots, q_n \}] f = \mathcal{M} \models q_0 = (u_0, \ldots, u_n) \in \mathcal{M}.
\]

- The relation defined in \( \mathcal{M} \) by \( \mathcal{M} \) restricted to \( \mathcal{M} \) is \( \mathcal{M} \).
- The set defined in \( \mathcal{M} \) by \( \mathcal{M} \) is \( \mathcal{M} \).
- The set defined in \( \mathcal{M} \) by \( \mathcal{M} \) is \( \mathcal{M} \).

There is a natural way to extract from a model of a model of \( \mathcal{L} \).

Interpretations
Given a formula \( \phi \) and an interpretation \( \nu \), we can find a formula which in some sense corresponds exactly to \( \nu \phi \) into \( \nu \). This is possible for atomic formulas, and it is possible for non-atomic formulas as well, as illustrated in the obvious ways:

\[
(\forall x \phi \leftarrow (x)^f \psi) z A = (\forall x (x)^f \psi) z A = (\forall z (x)^f \psi) z A = (\forall f (x)^f \psi) z A = (x)^f \psi.
\]

For example, given a predicate symbol \( P \) (it is not equal to) by \( f \).

The interpretation of non-atomic formulas are defined in the obvious way:

\[
(\forall x \phi \leftarrow (x)^f \psi), (\forall x (x)^f \psi) z A = (\forall x (x)^f \psi) z A = (\forall z (x)^f \psi) z A = (\forall f (x)^f \psi) z A = (x)^f \psi.
\]

Intefpretations
Interpretations

Lemma

Let \( \mathcal{I} \models \phi \) iff \( \mathcal{I} \models \phi \).

For a faithful interpretation, we have \( \mathcal{I} \models \phi \) iff \( \mathcal{I} \models \phi \).

If \( \mathcal{I} \models \phi \), then \( \mathcal{I} \models \phi \).

\[ [\mathcal{I}]_{\phi} \subseteq 0_{\mathcal{I}} \subseteq 0_{\mathcal{I}} \]

An interpretation \( \mathcal{I} \) of a theory \( \mathcal{I} \) is an interpretation of the signature of \( \mathcal{I} \) into \( \mathcal{I} \).

For a \( 0_{\mathcal{I}} \)-sentence \( \phi \), we have \( \mathcal{I} \models \phi \) iff \( \mathcal{I} \models \phi \).

Corollary

The proof is by induction on \( \phi \) and is omitted.

\[ [s]_{\phi} \models \phi \iff [s]_{\phi} \models \phi \]

\[ \left( I_{\phi} \right)_{\mathcal{I}} \text{ for any } s \text{ and any map of the variables into } \text{dom} \]

\[ \mathcal{I} \models \phi \]

Let be an interpretation of a model of \( \mathcal{I} \).

Lemma

Interpretations