

Foundations of Data and Knowledge Systems

EPCL Basic Training Camp 2012

Part Three

Thomas Eiter and Reinhard Pichler

Institut für Informationssysteme
Technische Universität Wien

19 December, 2012

Outline

- 4. Foundations of Rule and Query Languages
- 4.1 Fragments of First-Order Predicate Logic
- 4.2 Assessment of Tarski Model Theory
- 4.3 Minimal Model Semantics of Definite Rules

Outline

- 4. Foundations of Rule and Query Languages
- 4.1 Fragments of First-Order Predicate Logic
- 4.2 Assessment of Tarski Model Theory
- 4.3 Minimal Model Semantics of Definite Rules

Fragments of First-Order Predicate Logic

Motivation

- Some fragments of first-order predicate logic are particularly well suited as query languages; in particular **rule languages**.
- We shall later see appropriate deviations from Tarski Model Theory.

Fragments of First-Order Predicate Logic

Motivation

- Some fragments of first-order predicate logic are particularly well suited as **query languages**; in particular **rule languages**.
- We shall later see appropriate deviations from Tarski Model Theory.

Notation

- A **rule** $\psi \leftarrow \varphi$ is a notation for a (not necessarily closed) formula $\varphi \Rightarrow \psi$.
- φ is called the **antecedent** or **body** and ψ the **consequent** or **head**.
- A rule $\psi \leftarrow \top$ may be written $\psi \leftarrow$ with empty antecedent. A rule $\perp \leftarrow \varphi$ may be written $\leftarrow \varphi$ with empty consequent.
- Implicit Quantification.** Typically, a rule is a shorthand notation for its universal closure: Let \vec{x} denote the free variables occurring in ψ (and possibly in φ) and \vec{y} the free variables occurring in φ but not in ψ . Then the **universal closure** $\forall \vec{x} \forall \vec{y} (\psi \leftarrow \varphi)$ is logically equivalent to $\forall \vec{x} (\psi \leftarrow \exists \vec{y} \varphi)$.

Logic Programming

- Logic programming considers a finite set of clauses with non-empty consequent as a **program** and clauses with empty consequent as **goals** used for program invocation. Unit clauses are also called **facts**.
- In a **definite program**, all clauses are definite. Together with definite goals, they represent a fragment of first-order predicate logic with especially nice semantic properties – cf. “**pure Prolog**” in the context of Prolog.

Logic Programming

Clause Classification

The following names are defined for special forms of clauses:

Name	Form	
definite clause	$A \leftarrow B_1 \wedge \dots \wedge B_n$	$k = 1, n \geq 0$
unit cl.	$A \leftarrow$	$k = 1, n = 0$
definite goal	$\leftarrow B_1 \wedge \dots \wedge B_n$	$k = 0, n \geq 0$
empty cl.	\leftarrow	$k = 0, n = 0$
normal clause	$A \leftarrow L_1 \wedge \dots \wedge L_n$	$k = 1, n \geq 0$
normal goal	$\leftarrow L_1 \wedge \dots \wedge L_n$	$k = 0, n \geq 0$
disjunctive clause	$A_1 \vee \dots \vee A_k \leftarrow B_1 \wedge \dots \wedge B_n$	$k \geq 0, n \geq 0$
general clause	$A_1 \vee \dots \vee A_k \leftarrow L_1 \wedge \dots \wedge L_n$	$k \geq 0, n \geq 0$

atoms A, A_j, B_i , literals L_i , $k, n \in \mathbb{N}$

Logic Programming

- Logic programming considers a finite set of clauses with non-empty consequent as a **program** and clauses with empty consequent as **goals** used for program invocation. Unit clauses are also called **facts**.
- In a **definite program**, all clauses are definite. Together with definite goals, they represent a fragment of first-order predicate logic with especially nice semantic properties – cf. “**pure Prolog**” in the context of Prolog.

Datalog: special case of logic programming

- Function symbols other than constants are excluded. Thus, the only terms are variables and constants.
- Relation symbols are partitioned into those that may occur in the data to be queried, called **extensional**, and those that may not, called **intensional**.
- Clauses are assumed to be **range restricted**, which essentially requires that all variables in the consequent of a clause also occur in its antecedent.

Some Versions of Datalog

Definition

Many (restricted or extended) versions of datalog have been studied because of their interesting expressive power and/or complexity or by their correspondence to classes of queries defined by other formalisation approaches.

Monadic datalog 1-ary intensional relation symbols

Nonrecursive datalog no (direct or indirect) recursion

Linear datalog at most one intensional atom per antecedent

Disjunctive datalog disjunctive clauses

Datalog[∇] normal clauses

Nonrecursive datalog[∇] normal clauses, no recursion

Disjunctive datalog[∇] general clauses

Conjunctive Queries

Definition (Conjunctive query)

A **conjunctive query** is a datalog rule

$$ans(\vec{u}) \leftarrow r_1(\vec{u}_1) \wedge \dots \wedge r_n(\vec{u}_n)$$

where $n \geq 0$, the r_i are extensional and ans is an intensional relation symbol, $\vec{u}, \vec{u}_1, \dots, \vec{u}_n$ are lists of terms of appropriate length, and the rule is range restricted, i.e., each variable in \vec{u} also occurs in at least one of $\vec{u}_1, \dots, \vec{u}_n$.

A **boolean conjunctive query** is a conjunctive query where \vec{u} is the empty list, i.e., the answer relation symbol ans is propositional.

Remark

Conjunctive queries correspond to the **SPJ subclass** (or **SPC subclass**) of **relational algebra queries** constructed with selection, projection, join (or, alternatively, cartesian product).

Conjunctive Queries

Definition (Conjunctive query)

A **conjunctive query** is a datalog rule

$$ans(\vec{u}) \leftarrow r_1(\vec{u}_1) \wedge \dots \wedge r_n(\vec{u}_n)$$

where $n \geq 0$, the r_i are extensional and ans is an intensional relation symbol, $\vec{u}, \vec{u}_1, \dots, \vec{u}_n$ are lists of terms of appropriate length, and the rule is range restricted, i.e., each variable in \vec{u} also occurs in at least one of $\vec{u}_1, \dots, \vec{u}_n$.

A **boolean conjunctive query** is a conjunctive query where \vec{u} is the empty list, i.e., the answer relation symbol ans is propositional.

Examples of Conjunctive Queries

Extensional relation symbols: *parent, male, female*

Is Mary a parent of Tom?
 Does Mary have children?
 Who are Tom's parents?
 Who are Tom's grandmothers?

Who are grandfathers and their grandchildren?

Examples of Conjunctive Queries

Extensional relation symbols: *parent, male, female*

$ans() \leftarrow parent(Mary, Tom)$	Is Mary a parent of Tom? Does Mary have children? Who are Tom's parents? Who are Tom's grandmothers?
	Who are grandfathers and their grandchildren?

Examples of Conjunctive Queries

Extensional relation symbols: *parent, male, female*

$ans() \leftarrow parent(Mary, Tom)$	Is Mary a parent of Tom?
$ans() \leftarrow parent(Mary, y)$	Does Mary have children? Who are Tom's parents? Who are Tom's grandmothers?
	Who are grandfathers and their grandchildren?

Examples of Conjunctive Queries

Extensional relation symbols: *parent, male, female*

$ans() \leftarrow parent(Mary, Tom)$	Is Mary a parent of Tom?
$ans() \leftarrow parent(Mary, y)$	Does Mary have children?
$ans(x) \leftarrow parent(x, Tom)$	Who are Tom's parents? Who are Tom's grandmothers?
	Who are grandfathers and their grandchildren?

Examples of Conjunctive Queries

Extensional relation symbols: *parent, male, female*

$ans() \leftarrow parent(Mary, Tom)$	Is Mary a parent of Tom?
$ans() \leftarrow parent(Mary, y)$	Does Mary have children?
$ans(x) \leftarrow parent(x, Tom)$	Who are Tom's parents? Who are Tom's grandmothers?
$ans(x) \leftarrow female(x) \wedge parent(x, y) \wedge parent(y, Tom)$	Who are grandfathers and their grandchildren?

Examples of Conjunctive Queries

Extensional relation symbols: *parent*, *male*, *female*

$ans() \leftarrow parent(Mary, Tom)$	Is Mary a parent of Tom?
$ans() \leftarrow parent(Mary, y)$	Does Mary have children?
$ans(x) \leftarrow parent(x, Tom)$	Who are Tom's parents?
$ans(x) \leftarrow female(x) \wedge parent(x, y) \wedge parent(y, Tom)$	Who are Tom's grandmothers?
$ans(x, z) \leftarrow male(x) \wedge parent(x, y) \wedge parent(y, z)$	Who are grandfathers and their grandchildren?

Limitations of Conjunctive Queries

The following queries cannot be expressed as Conjunctive Queries:

- 1 *who are parents of Tom or Mary?*
- 2 *who are parents, but not of Tom?*
- 3 *who are women all of whose children are sons?*
- 4 *who are ancestors of Tom?*

Limitations of Conjunctive Queries

The following queries cannot be expressed as Conjunctive Queries:

- 1 *who are parents of Tom or Mary?*
requires disjunction in rule antecedents or more than a single rule.
- 2 *who are parents, but not of Tom?*
- 3 *who are women all of whose children are sons?*
- 4 *who are ancestors of Tom?*

Limitations of Conjunctive Queries

The following queries cannot be expressed as Conjunctive Queries:

- 1 *who are parents of Tom or Mary?*
requires disjunction in rule antecedents or more than a single rule.
- 2 *who are parents, but not of Tom?*
requires negation in rule antecedents.
- 3 *who are women all of whose children are sons?*
- 4 *who are ancestors of Tom?*

Limitations of Conjunctive Queries

The following queries cannot be expressed as Conjunctive Queries:

- 1 *who are parents of Tom or Mary?*
requires disjunction in rule antecedents or more than a single rule.
- 2 *who are parents, but not of Tom?*
requires negation in rule antecedents.
- 3 *who are women all of whose children are sons?*
requires universal quantification in rule antecedents.
Note that variables occurring only in the antecedent of a conjunctive query are interpreted as if *existentially* quantified in the antecedent.
- 4 *who are ancestors of Tom?*

Limitations of Conjunctive Queries

The following queries cannot be expressed as Conjunctive Queries:

- 1 *who are parents of Tom or Mary?*
requires disjunction in rule antecedents or more than a single rule.
- 2 *who are parents, but not of Tom?*
requires negation in rule antecedents.
- 3 *who are women all of whose children are sons?*
requires universal quantification in rule antecedents.
Note that variables occurring only in the antecedent of a conjunctive query are interpreted as if *existentially* quantified in the antecedent.
- 4 *who are ancestors of Tom?*
requires recursion, i.e., intensional relation symbols in rule antecedents.

Outline

4. Foundations of Rule and Query Languages
 - 4.1 Fragments of First-Order Predicate Logic
 - 4.2 Assessment of Tarski Model Theory
 - 4.3 Minimal Model Semantics of Definite Rules

Tarski Model Theory for Logic and Mathematics in General

Important Characteristics

- domain of an interpretation may be any nonempty set:
first-order predicate logic can model statements about any arbitrary application domain
- excellent clarification of relationship syntax/semantics
- simple recursive definition of semantics
- rich body of results
- quite successful for mathematics

Inadequacy for Query Languages

1: Unique name assumption

- different constants to be interpreted differently
- frequent requirement in applications
a mechanism making it available by default would be useful
- not supported by Tarski model theory
explicit formalisation is cumbersome

3: Closed world assumption

- nothing holds unless explicitly specified
- tacit understanding in many applications
(transportation timetables)
- cannot be expressed in first-order predicate logic
with Tarski model theory

Inadequacy for Query Languages

1: Unique name assumption

- different constants to be interpreted differently
- frequent requirement in applications
a mechanism making it available by default would be useful
- not supported by Tarski model theory
explicit formalisation is cumbersome

2: Function symbols as term constructors

- grouping pieces of data that belong together
- makes sense in many applications
- terms as compound data structures
- not supported by Tarski model theory

3: Closed world assumption

- nothing holds unless explicitly specified
- tacit understanding in many applications
(transportation timetables)
- cannot be expressed in first-order predicate logic
with Tarski model theory

4: Restrictions on the cardinality of models

- Real-world applications are often finite (e.g., databases); in this case infinite domains should be excluded from further consideration.
- Finiteness cannot be expressed in first-order predicate logic with Tarski model theory.
- Likewise, the restriction to domains with a given cardinality, with odd cardinality, etc. cannot be expressed in first-order predicate logic with Tarski model theory.

5: Definability of transitive closure

- relevant in many query answering applications
 - e.g., traffic application
 - r represents direct connections between junctions
 - t represents indirect connections
 - t should be interpreted as the transitive closure of r
- cannot be expressed in first-order predicate logic with Tarski model theory

$$\forall x \forall z (t(x, z) \Leftrightarrow (r(x, z) \vee \exists y [t(x, y) \wedge t(y, z)]))$$

does **not** do it! Why?

5: Definability of transitive closure

- relevant in many query answering applications
 - e.g., traffic application
 - r represents direct connections between junctions
 - t represents indirect connections
 - t should be interpreted as the transitive closure of r
- cannot be expressed in first-order predicate logic with Tarski model theory

$$\forall x \forall z (t(x, z) \Leftrightarrow (r(x, z) \vee \exists y [t(x, y) \wedge t(y, z)]))$$

does **not** do it! Why?

- For instance, consider the following interpretation $\mathcal{I} = (D, I)$:
 - $D = \{0, 1, 2\}$
 - $r^{\mathcal{I}} = \{(0, 1), (1, 2)\}$
 - $t^{\mathcal{I}} = \{(0, 1), (1, 2), (0, 2), (0, 0)\}$

Alternative Semantics Definitions

Alternative Approaches

Several approaches aim at overcoming some of these problems 1 to 5, e.g.:

- **Herbrand Model Theory.** Considering only *Herbrand interpretations* and *Herbrand models* instead of general interpretations addresses points 1 and 2.
- **Minimal model semantics.** Considering only *minimal Herbrand models* addresses point 3. Applying the minimal model semantics to (definite) rules addresses point 5.
- **Finite Model Theory.** Considering only *finite* interpretations and models addresses point 4.

Outline

- 4. Foundations of Rule and Query Languages
 - 4.1 Fragments of First-Order Predicate Logic
 - 4.2 Assessment of Tarski Model Theory
 - 4.3 Minimal Model Semantics of Definite Rules

Minimal Model Semantics of Definite Rules

Motivation

- Recall: **Definite programs** are finite sets of **definite clauses**, also called **definite rules**: $A \leftarrow B_1 \wedge \dots \wedge B_n$ with $n \geq 0$.
- Definite programs admit a very natural semantics definition:
 - Each program Π is satisfiable.
 - The intersection of all its Herbrand models is a model of Π .
 - This is the *minimal model* of Π .
 - Precisely the atoms implied by Π are true in the minimal model.

Universal and Inductive Formulas

Definition (Universal and Inductive Formulas)

Recall the transformation of any formula into **prenex form**.

- A formula is called **universal** if it can be transformed into a prenex form with **universal quantifiers only**.
- A formula is called **inductive** if it can be transformed into a prenex form with the following properties:
 - The quantifier prefix starts with **universal quantifiers** for all variables in the consequent followed by arbitrary quantifiers for the remaining variables.
 - The quantifier-free part is of the form $(A_1 \wedge \dots \wedge A_n) \leftarrow \varphi$, where $n \geq 0$ and φ is a **positive formula** (i.e., it contains no negation).
- An inductive formula is either a **generalised definite rule** (if $n \geq 1$) or a **generalised definite goal** (if $n = 0$).

Minimal Model Semantics of Definite Rules

Motivation

- Recall: **Definite programs** are finite sets of **definite clauses**, also called **definite rules**: $A \leftarrow B_1 \wedge \dots \wedge B_n$ with $n \geq 0$.
- Definite programs admit a very natural semantics definition:
 - Each program Π is satisfiable.
 - The intersection of all its Herbrand models is a model of Π .
 - This is the *minimal model* of Π .
 - Precisely the atoms implied by Π are true in the minimal model.
- Definite rules are a special case of **universal** and **inductive** formulas.
- The interesting model-theoretic properties of definite rules are inherited from these more general classes of formulas.

Outline of the Subsection

Roadmap

- Definition: compatible interpretations, intersection of interpretations
- Definition: intersection of Herbrand models $HI(Mod_{\cap}(S))$
- Definition: order on models, minimal (Herbrand) model

Outline of the Subsection

Roadmap

- Definition: compatible interpretations, intersection of interpretations
- Definition: intersection of Herbrand models $HI(Mod_{\cap}(S))$
- Definition: order on models, minimal (Herbrand) model
- Theorem: For **universal formulas** S , $Mod_{\cap}(S) = \{A \in HB \mid S \models A\}$.
- Observation: $HI(Mod_{\cap}(S))$ is not necessarily a model of S .

Outline of the Subsection

Roadmap

- Definition: compatible interpretations, intersection of interpretations
- Definition: intersection of Herbrand models $HI(Mod_{\cap}(S))$
- Definition: order on models, minimal (Herbrand) model
- Theorem: For **universal formulas** S , $Mod_{\cap}(S) = \{A \in HB \mid S \models A\}$.
- Observation: $HI(Mod_{\cap}(S))$ is not necessarily a model of S .
- Theorem: Satisfiability of **definite inductive formulas**.
- Theorem: For **inductive formulas** S , the intersection of compatible models is a model.
- Main result:
Minimal Herbrand Model $HI(Mod_{\cap}(\Pi))$ of a Definite Program Π .

Outline of the Subsection

Roadmap

- Definition: compatible interpretations, intersection of interpretations
- Definition: intersection of Herbrand models $HI(Mod_{\cap}(S))$
- Definition: order on models, minimal (Herbrand) model
- Theorem: For **universal formulas** S , $Mod_{\cap}(S) = \{A \in HB \mid S \models A\}$.
- Observation: $HI(Mod_{\cap}(S))$ is not necessarily a model of S .
- Theorem: Satisfiability of **definite inductive formulas**.
- Theorem: For **inductive formulas** S , the intersection of compatible models is a model.

Intersection of (Compatible) Interpretations

Definition (Compatible set of interpretations)

A set $\{\mathcal{I}_i \mid i \in I\}$ of interpretations with index set I is called **compatible**, iff

- $I \neq \emptyset$.
- $D = \bigcap \{dom(\mathcal{I}_i) \mid i \in I\} \neq \emptyset$.
- all interpretations of a function symbol coincide on the common domain:
 $f^{\mathcal{I}_i}(d_1, \dots, d_n) = f^{\mathcal{I}_j}(d_1, \dots, d_n)$ for each n -ary ($n \geq 0$) function symbol f , for all $i, j \in I$, and for all $d_1, \dots, d_n \in D$.
- each variable is identically interpreted in all interpretations:
 $x^{\mathcal{I}_i} = x^{\mathcal{I}_j}$ for each variable x and all $i, j \in I$.

Definition (Intersection of a compatible set of interpretations)

Let $\{\mathcal{I}_i \mid i \in I\}$ be a compatible set of interpretations. Then $\bigcap\{\mathcal{I}_i \mid i \in I\}$ is defined as the interpretation \mathcal{I} with

- $dom(\mathcal{I}) = D = \bigcap\{dom(\mathcal{I}_i) \mid i \in I\}$.
- a function symbol is interpreted as in each interpretation \mathcal{I}_i :
 $f^{\mathcal{I}}(d_1, \dots, d_n) = f^{\mathcal{I}_i}(d_1, \dots, d_n)$ for each n -ary ($n \geq 0$) function symbol f , for an arbitrary $i \in I$, and for all $d_1, \dots, d_n \in D$.
- a relation symbol is interpreted as the intersection of its interpretations:
 $p^{\mathcal{I}} = \bigcap_{i \in I} p^{\mathcal{I}_i}$ for each relation symbol p .
- a variable is interpreted like in all given interpretations:
 $x^{\mathcal{I}} = x^{\mathcal{I}_i}$ for each variable x and an arbitrary $i \in I$.

Intersection of Herbrand Models

Lemma

Let $\{B_i \mid i \in I\}$ be a set of sets of ground atoms, i.e., $B_i \subseteq HB$ for each $i \in I$. If this set is nonempty, then

- $\{HI(B_i) \mid i \in I\}$ is a compatible set of interpretations.
- $\bigcap\{HI(B_i) \mid i \in I\} = HI(\bigcap\{B_i \mid i \in I\})$ i.e., its intersection is the Herbrand interpretation induced by the intersection of inducers.

Intersection of Herbrand Models

Lemma

Let $\{B_i \mid i \in I\}$ be a set of sets of ground atoms, i.e., $B_i \subseteq HB$ for each $i \in I$. If this set is nonempty, then

- $\{HI(B_i) \mid i \in I\}$ is a compatible set of interpretations.
- $\bigcap\{HI(B_i) \mid i \in I\} = HI(\bigcap\{B_i \mid i \in I\})$ i.e., its intersection is the Herbrand interpretation induced by the intersection of inducers.

Definition (Set of inducers of Herbrand models of a set of formulas)

For a set S of formulas, the set of inducers of its Herbrand models is

$$Mod_{HB}(S) = \{B \subseteq HB \mid HI(B) \models S\}.$$

Intersection of Herbrand Models

Lemma

Let $\{B_i \mid i \in I\}$ be a set of sets of ground atoms, i.e., $B_i \subseteq HB$ for each $i \in I$. If this set is nonempty, then

- $\{HI(B_i) \mid i \in I\}$ is a compatible set of interpretations.
- $\bigcap\{HI(B_i) \mid i \in I\} = HI(\bigcap\{B_i \mid i \in I\})$ i.e., its intersection is the Herbrand interpretation induced by the intersection of inducers.

Definition (Set of inducers of Herbrand models of a set of formulas)

For a set S of formulas, the set of inducers of its Herbrand models is

$$Mod_{HB}(S) = \{B \subseteq HB \mid HI(B) \models S\}.$$

Notation

For a set S of formulas:

$$Mod_{\cap}(S) = \begin{cases} \bigcap Mod_{HB}(S) & \text{if } Mod_{HB}(S) \neq \emptyset \\ HB & \text{if } Mod_{HB}(S) = \emptyset \end{cases}$$

Order on Models

Definition (Order on Models)

$\mathcal{I}_1 \leq \mathcal{I}_2$ for interpretations \mathcal{I}_1 and \mathcal{I}_2 if

- $dom(\mathcal{I}_1) = dom(\mathcal{I}_2)$.
- the interpretations of a function symbol coincide on the common domain:
 $f^{\mathcal{I}_1}(d_1, \dots, d_n) = f^{\mathcal{I}_2}(d_1, \dots, d_n)$ for each n -ary ($n \geq 0$) function symbol f and all $d_1, \dots, d_n \in dom(\mathcal{I}_1)$.
- the “smaller” interpretation of a relation symbol is a restriction of the other:
 $p^{\mathcal{I}_1} \subseteq p^{\mathcal{I}_2}$ for each n -ary ($n \geq 0$) relation symbol p .
- each variable is identically interpreted in the interpretations:
 $x^{\mathcal{I}_1} = x^{\mathcal{I}_2}$ for each variable x .

Minimal Model

Definition (Minimal model)

A **minimal model** of a set of formulas is a \leq -minimal member \mathcal{I} of the set of all its models with domain $dom(\mathcal{I})$.

Proposition

Let S be a set of formulas. An Herbrand model $HI(B)$ of S is minimal iff there is no proper subset $B' \subset B$ such that $HI(B')$ is also a model of S .

Minimal Model

Definition (Minimal model)

A **minimal model** of a set of formulas is a \leq -minimal member \mathcal{I} of the set of all its models with domain $dom(\mathcal{I})$.

Minimal Model

Definition (Minimal model)

A **minimal model** of a set of formulas is a \leq -minimal member \mathcal{I} of the set of all its models with domain $dom(\mathcal{I})$.

Proposition

Let S be a set of formulas. An Herbrand model $HI(B)$ of S is minimal iff there is no proper subset $B' \subset B$ such that $HI(B')$ is also a model of S .

Lemma

Let S be a set of formulas.

- An Herbrand model $HI(B)$ of S is minimal iff B is a \subseteq -minimal member of $Mod_{HB}(S)$.
- If $HI(Mod_{\cap}(S))$ is a model of S , then it is the **unique minimal Herbrand model** of S .

Theorem

If S is universal, then $Mod_{\cap}(S) = \{A \in HB \mid S \models A\}$.

Proof

If S is unsatisfiable, both sides are equal to HB . So suppose that S is satisfiable:

“ \subseteq ”: Let $A \in Mod_{\cap}(S)$, thus $A \in B$ for each $B \subseteq HB$ with $HI(B) \models S$.

We have to show $S \models A$.

Let \mathcal{I} be an arbitrary model of S . By the correspondence of satisfiability and Herbrand-satisfiability for *universal formulas*, $HI(B') \models S$ where $B' = \{A' \in HB \mid \mathcal{I} \models A'\}$. Hence, $A \in B'$ and, therefore $\mathcal{I} \models A$.

Since \mathcal{I} was arbitrary, we have shown $S \models A$.

Theorem

If S is universal, then $Mod_{\cap}(S) = \{A \in HB \mid S \models A\}$.

Proof

If S is unsatisfiable, both sides are equal to HB . So suppose that S is satisfiable:

“ \subseteq ”: Let $A \in Mod_{\cap}(S)$, thus $A \in B$ for each $B \subseteq HB$ with $HI(B) \models S$.

We have to show $S \models A$.

Let \mathcal{I} be an arbitrary model of S . By the correspondence of satisfiability and Herbrand-satisfiability for *universal formulas*, $HI(B') \models S$ where $B' = \{A' \in HB \mid \mathcal{I} \models A'\}$. Hence, $A \in B'$ and, therefore $\mathcal{I} \models A$.

Since \mathcal{I} was arbitrary, we have shown $S \models A$.

“ \supseteq ”: Let $A \in HB$ with $S \models A$, i.e., each model of S satisfies A .

Then for each $B \subseteq HB$ with $HI(B) \models S$ holds $HI(B) \models A$ and thus $A \in B$. Hence $A \in Mod_{\cap}(S)$. \square

Motivation

The above theorem shows that $HI(Mod_{\cap}(S))$ has an interesting property for universal formulas. However, there remain two concerns:

- S may be unsatisfiable:
 $HI(Mod_{\cap}(S))$ is the Herbrand interpretation induced by those atoms which are implied by S . This is non-trivial only if $Mod_{HB}(S) \neq \emptyset$. We shall see that for sets of definite inductive formulas, $Mod_{HB}(S) \neq \emptyset$ is guaranteed.
- $HI(Mod_{\cap}(S))$ is not necessarily a model of S :
 This may be the case even if S is satisfiable (and universal). We shall see that for sets of inductive formulas, $HI(Mod_{\cap}(S))$ is always a model of S .

Motivation

The above theorem shows that $HI(Mod_{\cap}(S))$ has an interesting property for universal formulas. However, there remain two concerns:

- S may be unsatisfiable:
 $HI(Mod_{\cap}(S))$ is the Herbrand interpretation induced by those atoms which are implied by S . This is non-trivial only if $Mod_{HB}(S) \neq \emptyset$. We shall see that for sets of definite inductive formulas, $Mod_{HB}(S) \neq \emptyset$ is guaranteed.
- $HI(Mod_{\cap}(S))$ is not necessarily a model of S :
 This may be the case even if S is satisfiable (and universal). We shall see that for sets of inductive formulas, $HI(Mod_{\cap}(S))$ is always a model of S .

Example

Assume a signature consisting of a unary relation symbol p and constants a, b . Let $S = \{p(a) \vee p(b)\}$. Then $Mod_{HB}(S) = \{ \{p(a)\}, \{p(b)\}, \{p(a), p(b)\} \}$. But $HI(Mod_{\cap}(S)) = HI(\emptyset)$ is not a model of S .

Important Properties of Inductive Formulas

Theorem

For each set S of generalised definite rules, $HI(HB) \models S$.

Proof

Let S be a set of generalised definite rules. Thus each member of S is equivalent to a formula of the form $\forall \vec{x}[(A_1 \wedge \dots \wedge A_n) \leftarrow \varphi]$ where \vec{x} are the variables occurring in $A_1 \dots A_n$.

Clearly, for every ground instance $A_i\sigma$ of each atom A_i in the conclusion, we have $HI(HB) \models A_i\sigma$. Thus $HI(HB) \models (A_1 \wedge \dots \wedge A_n)\sigma$ and, therefore, also $HI(HB) \models [(A_1 \wedge \dots \wedge A_n) \leftarrow \varphi]\sigma$ for every ground substitution σ .

Hence, $HI(HB)$ satisfies each member of S . \square

Theorem

Let S be a set of inductive formulas. If $\{\mathcal{I}_i \mid i \in I\}$ is a set of compatible models of S with the same domain D , then $\mathcal{I} = \bigcap \{\mathcal{I}_i \mid i \in I\}$ is also a model of S .

Proof Idea

Each member of S is (equivalent to) a formula $\forall \vec{x}[(A_1 \wedge \dots \wedge A_n) \leftarrow \varphi]$ with $n \geq 0$ where \vec{x} are the variables in A_1, \dots, A_n and φ is a positive formula.

Let V be an arbitrary variable assignment on \vec{x} . Clearly $\mathcal{I}[V] \leq \mathcal{I}_i[V]$ for each i .

If φ is false in $\mathcal{I}[V]$, then S is trivially true in $\mathcal{I}[V]$. Now suppose that $\mathcal{I}[V] \models \varphi$. Then clearly $\mathcal{I}_i[V] \models \varphi$ for each $i \in I$ (since φ is positive).

By assumption, $\mathcal{I}_i[V] \models (A_1 \wedge \dots \wedge A_n) \leftarrow \varphi$ holds. It follows that $\mathcal{I}_i[V] \models (A_1 \wedge \dots \wedge A_n)$. Thus $\mathcal{I}[V] \models (A_1 \wedge \dots \wedge A_n)$ and, therefore $\mathcal{I}[V] \models [(A_1 \wedge \dots \wedge A_n) \leftarrow \varphi]$. Hence, (since V is arbitrary), also $\mathcal{I} \models S$. \square

Theorem

Let S be a set of inductive formulas. If $\{\mathcal{I}_i \mid i \in I\}$ is a set of compatible models of S with the same domain D , then $\mathcal{I} = \bigcap \{\mathcal{I}_i \mid i \in I\}$ is also a model of S .

Theorem

Let S be a set of inductive formulas. If $\{\mathcal{I}_i \mid i \in I\}$ is a set of compatible models of S with the same domain D , then $\mathcal{I} = \bigcap \{\mathcal{I}_i \mid i \in I\}$ is also a model of S .

Proof Idea

Each member of S is (equivalent to) a formula $\forall \vec{x}[(A_1 \wedge \dots \wedge A_n) \leftarrow \varphi]$ with $n \geq 0$ where \vec{x} are the variables in A_1, \dots, A_n and φ is a positive formula.

Let V be an arbitrary variable assignment on \vec{x} . Clearly $\mathcal{I}[V] \leq \mathcal{I}_i[V]$ for each i .

If φ is false in $\mathcal{I}[V]$, then S is trivially true in $\mathcal{I}[V]$. Now suppose that $\mathcal{I}[V] \models \varphi$. Then clearly $\mathcal{I}_i[V] \models \varphi$ for each $i \in I$ (since φ is positive).

By assumption, $\mathcal{I}_i[V] \models (A_1 \wedge \dots \wedge A_n) \leftarrow \varphi$ holds. It follows that $\mathcal{I}_i[V] \models (A_1 \wedge \dots \wedge A_n)$. Thus $\mathcal{I}[V] \models (A_1 \wedge \dots \wedge A_n)$ and, therefore $\mathcal{I}[V] \models [(A_1 \wedge \dots \wedge A_n) \leftarrow \varphi]$. Hence, (since V is arbitrary), also $\mathcal{I} \models S$. \square

Corollary

If S is a set of inductive formulas and $\{B_i \subseteq HB \mid i \in I\}$ is a nonempty set with $HI(B_i) \models S$ for each $i \in I$, then $HI(\bigcap \{B_i \mid i \in I\}) \models S$.

Minimal Model of Definite Programs

Theorem

Each set S of definite rules (i.e., each definite program) has a unique minimal Herbrand model. This model is the intersection of all Herbrand models of S . It satisfies precisely those ground atoms that are logical consequences of S .

Minimal Model of Definite Programs

Theorem

Each set S of definite rules (i.e., each definite program) has a unique minimal Herbrand model. This model is the intersection of all Herbrand models of S . It satisfies precisely those ground atoms that are logical consequences of S .

Proof

- Every set S of inductive formulas is satisfiable. Hence, $HI(Mod_{\cap}(S))$ is the intersection of the Herbrand models of S .
- The intersection of models of a set S of inductive formulas is a model of S . Hence, $HI(Mod_{\cap}(S))$ is a model of S .
- If $HI(Mod_{\cap}(S))$ is a model of S then it is the unique minimal Herbrand model of S .
- For universal formulas S , $HI(Mod_{\cap}(S))$ satisfies precisely those ground atoms that are logical consequences of S .

□