Predicate logic - Structures (models)

Dr. Son P. Nguyen

UEL
VNU-HCMC

March 14, 2016

Structures

Definition

Let S be a signature. An S-structure $\mathcal{M}$ consists of:

- a non-empty set D (called the domain of $\mathcal{M}$);
- a set $P^{\mathcal{M}} \subseteq D^k$, for each k-ary predicate symbol P in S;
- a function $F^{\mathcal{M}} : D^k \to D$, for each k-ary function symbol F in S;
- an element $c^{\mathcal{M}} \in D$, for each constant symbol c in S.

Thus, an S-structure specifies a domain (of discourse),
and meanings for all predicate/function/constant symbols in S.

Example: Students and Instructors

Signature $S = \{\mathit{Student}, \mathit{Younger}, \mathit{Instructor}\}$:

- *Student* and *Instructor* are unary (i.e., 1-ary or of arity 1)
- *Younger* is binary (i.e., 2-ary or of arity 2)

A possible S-structure $\mathcal{M}$:

- Domain of $\mathcal{M}$: $D = \{\text{Alice}, \text{Bob}, \text{Carol}\}$
- $\mathit{Student}^{\mathcal{M}} = \{\text{Alice}, \text{Bob}\}$
- $\mathit{Younger}^{\mathcal{M}} = \{(\text{Alice}, \text{Bob}),\ (\text{Bob}, \text{Carol}),\ (\text{Alice}, \text{Carol})\}$
- $\mathit{Instructor}^{\mathcal{M}} = \{\text{Carol}\}$

Example: Students and Instructors

Signature $S = \{Student, Younger, Instructor\}$:

- $Student$ and $Instructor$ are unary (i.e., 1-ary or of arity 1)
- $Younger$ is binary (i.e., 2-ary or of arity 2)

A different S-structure $\mathcal{M}'$:

- Domain of $\mathcal{M}'$: the set D of all the people living in the UK
- $Student^{\mathcal{M}'} = \{d \in D \mid d \text{ is a student}\}$
- $Younger^{\mathcal{M}'} = \{(d, d') \in D^2 \mid d \text{ is younger than } d'\}$
- $Instructor^{\mathcal{M}'} = \{d \in D \mid d \text{ is an instructor at some university}\}$

Example: Arithmetic

Signature $S = \{F, \underline{2}\}$:

- F is a binary function symbol
- $\underline{2}$ is a constant symbol

A possible S-structure $\mathcal{M}$:

- Domain of $\mathcal{M}$: $D = \{0, 1, 2, 3, \dots\}$
- $F^{\mathcal{M}} : D^2 \to D$ is such that for all $d, d' \in D$:

$$F^{\mathcal{M}}(d, d') = d + d'$$

- $\underline{2}^{\mathcal{M}} = 2$ (the number, not to be confused with the symbol $\underline{2}$!)

Example: Kinship Relations

Signature $S = \{Female, Parent, alice\}$:

- *Female* and *Parent* are predicate symbols of arity 1 and 2, respectively,
- *alice* is a constant symbol.

A possible S-structure $\mathcal{M}$:

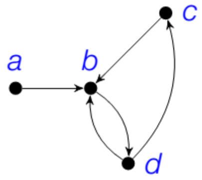
- Domain of $\mathcal{M}$: the set of all the people on earth
- $Female^{\mathcal{M}} = \{d \in D \mid d \text{ is female}\}$
- $Parent^{\mathcal{M}} = \{(d, d') \in D^2 \mid d \text{ is a parent of } d'\}$
- $alice^{\mathcal{M}} = \text{Alice}$

Example: Graphs

A (directed) graph consists of:

- points
  (called nodes, or vertices)
- directed lines connecting the points
  (called (directed) edges, or arcs).

Example:



A graph can be represented as a $\{E\}$-structure G, where E is a binary predicate symbol:

- The domain of G are the graph's nodes: $D = \{a, b, c, d\}$.
- $E^G = \{(a, b), (b, d), (c, b), (d, b), (d, c)\}$
  represents the graph's edges, including their directions.

Reminder

- Formulae of predicate logic are strings without any meaning, so how do we make sense of, say,

$$\exists x \, \exists y \, \big(\mathit{Sibling}(\mathit{alice}, x) \wedge \mathit{Mother}(x) = y \big) \quad ?$$

- Last lecture: We can use structures to specify
    - a domain of discourse and
    - a meaning for each of the symbols in a formulae.

Example: Kinship Relations

Signature $S = \{\mathit{Sibling}, \mathit{Mother}, \mathit{alice}\}$:

- *Sibling* is a binary predicate symbol,
- *Mother* is a unary function symbol,
- *alice* is a constant symbol.

S-structure $\mathcal{M}$:

- Domain of $\mathcal{M}$: $D = \{\text{Alice}, \text{Bob}, \text{Carol}, \text{Unknown}\}$
- $\mathit{Sibling}^{\mathcal{M}} = \{(\text{Alice}, \text{Bob})\}$
- $\mathit{Mother}^{\mathcal{M}}(\text{Alice}) = \mathit{Mother}^{\mathcal{M}}(\text{Bob}) = \text{Carol}$,
  $\mathit{Mother}^{\mathcal{M}}(d) = \text{Unknown}$ for all $d \in D \setminus \{\text{Alice}, \text{Bob}\}$
- $\mathit{alice}^{\mathcal{M}} = \text{Alice}$

Reminder

- Formulae of predicate logic are strings without any meaning, so how do we make sense of, say,

$$\exists x\, \exists y\, \big(\, \mathit{Sibling}(\mathit{alice}, x) \wedge \mathit{Mother}(x) = y \,\big) \quad ?$$

- Last lecture: We can use structures to specify
  - a domain of discourse and
  - a meaning for each of the symbols in a formulae.

- Now: How to interpret formulae in structures?

Example: Students and Instructors (cont'd)

- Signature $S = \{Student, Younger, alice\}$
- S-structure $\mathcal{M}$:
  - Domain of $\mathcal{M}$: $D = \{\text{Alice}, \text{Bob}, \text{Carol}\}$
  - $Student^{\mathcal{M}} = \{\text{Alice}, \text{Bob}\}$
  - $Younger^{\mathcal{M}} = \{(\text{Alice}, \text{Bob}), \ (\text{Bob}, \text{Carol}), \ (\text{Alice}, \text{Carol})\}$
  - $alice^{\mathcal{M}} = \text{Alice}$

Example: Students and Instructors (cont'd)

In the context of the structure $\mathcal{M}$:

- $\mathit{Student}(\mathit{alice})$ means: $\mathit{alice}^{\mathcal{M}} \in \mathit{Student}^{\mathcal{M}}$.
- $\exists x\ \mathit{Younger}(\mathit{alice}, x)$ means:
  There exists an object $d \in D$ such that $(\mathit{alice}^{\mathcal{M}}, d) \in \mathit{Younger}^{\mathcal{M}}$.
- $\neg\exists x\ \mathit{Student}(x)$ means:
  There does not exist a $d \in D$ such that $d \in \mathit{Student}^{\mathcal{M}}$.
- What does $\mathit{Younger}(\mathit{alice}, x)$ mean?

  We have to associate the free variable x with a concrete object to make sense of this.

Notation

- The domain of a structure $\mathcal{M}$ is denoted by $\mathrm{dom}(\mathcal{M})$.
- The set of all variables is denoted by Var.

Assignments

Definition

Let S be a signature, and $\mathcal{M}$ an S-structure.

- An assignment in $\mathcal{M}$ is a function $a\colon \mathrm{Var} \to \mathrm{dom}(\mathcal{M})$.
- Given a variable x and an element $d \in \mathrm{dom}(\mathcal{M})$, we define the assignment $a[x \mapsto d]$ in $\mathcal{M}$ by:

$$a[x \mapsto d](y) := \begin{cases} d, & \text{if } y = x, \\ a(y), & \text{otherwise.} \end{cases}$$

- An assignment a assigns to each variable x an object in $\mathcal{M}$'s domain, namely $a(x)$.
- $a[x \mapsto d]$ changes a slightly by assigning to x the object d (but otherwise, it's the same as a).

Example: Students and Instructors (cont'd)

- Signature $S = \{\mathit{Student}, \mathit{Younger}, \mathit{alice}\}$
- S-structure $\mathcal{M}$:
  - Domain of $\mathcal{M}$: $D = \{\text{Alice}, \text{Bob}, \text{Carol}\}$
  - $\mathit{Student}^{\mathcal{M}} = \{\text{Alice}, \text{Bob}\}$
  - $\mathit{Younger}^{\mathcal{M}} = \{(\text{Alice}, \text{Bob}), (\text{Bob}, \text{Carol}), (\text{Alice}, \text{Carol})\}$
  - $\mathit{alice}^{\mathcal{M}} = \text{Alice}$
- A possible assignment in $\mathcal{M}$ is the function $a \colon \text{Var} \to D$ with $a(x) = \text{Alice}$, $a(y) = \text{Bob}$, and $a(z) = \text{Carol}$ for all other variables z.
- $a[x \mapsto \text{Bob}]$ is the assignment such that $a[x \mapsto \text{Bob}](x) = \text{Bob}$, $a[x \mapsto \text{Bob}](y) = \text{Bob}$, and $a[x \mapsto \text{Bob}](z) = \text{Carol}$ for all other variables z.

The Value of a Term

Recall: terms are names for objects

The object pointed to by a term can be calculated after "plugging in" the meanings of variables, constants, functions:

Definition

Let $\mathcal{M}$ be an S-structure, and a an assignment in $\mathcal{M}$.

The value of an S-term t in $(\mathcal{M}, a)$, written $t^{\mathcal{M},a}$, is obtained by:

1. replacing each variable x in t by $a(x)$;
2. replacing each constant symbol c in t by $c^{\mathcal{M}}$;
3. replacing each function symbol F in t by $F^{\mathcal{M}}$;
4. computing the value of the resulting expression.

Example: Kinship Relations

- Signature $S = \{Sibling, Mother, alice\}$
- S-structure $\mathcal{M}$:
  - Domain of $\mathcal{M}$: $D = \{\text{Alice}, \text{Bob}, \text{Carol}, \text{Unknown}\}$
  - $Sibling^{\mathcal{M}} = \{(\text{Alice}, \text{Bob})\}$
  - $Mother^{\mathcal{M}}(\text{Alice}) = Mother^{\mathcal{M}}(\text{Bob}) = \text{Carol}$,
    $Mother^{\mathcal{M}}(d) = \text{Unknown}$ for all $d \in D \setminus \{\text{Alice}, \text{Bob}\}$
  - $alice^{\mathcal{M}} = \text{Alice}$
- Assignment a defined by $a(x) = \text{Bob}$, and $a(y) = \text{Unknown}$ for all variables $y \in \text{Var} \setminus \{x\}$

S-term t	the corresponding value $t^{\mathcal{M},a}$
$alice$	Alice
x	Bob
$Mother(alice)$	Carol
$Mother(x)$	Carol
$Mother(Mother(alice))$	Unknown

Satisfaction Relation

Let $\mathcal{M}$ be an S-structure, and a an assignment in $\mathcal{M}$.

An S-formula φ is satisfied in $(\mathcal{M}, a)$, denoted by $(\mathcal{M}, a) \models \varphi$, if the following holds:

- If $\varphi = P(t_1, \ldots, t_k)$, then $(\mathcal{M}, a) \models \varphi$ if $(t_1^{\mathcal{M},a}, \ldots, t_k^{\mathcal{M},a}) \in P^{\mathcal{M}}$.
- If φ has the form $t_1 = t_2$, then $(\mathcal{M}, a) \models \varphi$ if $t_1^{\mathcal{M},a} = t_2^{\mathcal{M},a}$.
- If $\varphi = \neg\psi$, then $(\mathcal{M}, a) \models \varphi$ if $(\mathcal{M}, a) \not\models \psi$ ("not $(\mathcal{M}, a) \models \psi$").
- If $\varphi = \psi \wedge \chi$, then $(\mathcal{M}, a) \models \varphi$ if $(\mathcal{M}, a) \models \psi$ and $(\mathcal{M}, a) \models \chi$.
- If $\varphi = \psi \vee \chi$, then $(\mathcal{M}, a) \models \varphi$ if $(\mathcal{M}, a) \models \psi$ or $(\mathcal{M}, a) \models \chi$.
- If $\varphi = \exists x\, \psi$, then $(\mathcal{M}, a) \models \varphi$ if there exists a $d \in \mathrm{dom}(\mathcal{M})$ such that $(\mathcal{M}, a[x \mapsto d]) \models \psi$.
- If $\varphi = \forall x\, \psi$, then $(\mathcal{M}, a) \models \varphi$ if for all $d \in \mathrm{dom}(\mathcal{M})$ we have $(\mathcal{M}, a[x \mapsto d]) \models \psi$.

Example: Students and Instructors (cont'd)

4. $(\mathcal{M}, a) \models \mathit{Younger}(\mathit{alice}, y)$:
   This is true since $\mathit{alice}^{\mathcal{M},a} = \text{Alice}$, $y^{\mathcal{M},a} = a(y) = \text{Bob}$, and $(\text{Alice}, \text{Bob}) \in \mathit{Younger}^{\mathcal{M}}$.

5. $(\mathcal{M}, a) \models \exists x\, \mathit{Younger}(\mathit{alice}, x)$:
   First of all, note that $(\mathcal{M}, a[x \mapsto \text{Bob}]) \models \mathit{Younger}(\mathit{alice}, x)$. This can be shown as above. Just observe that $\mathit{alice}^{\mathcal{M},a[x\mapsto\text{Bob}]} = \text{Alice}$ and $x^{\mathcal{M},a[x\mapsto\text{Bob}]} = \text{Bob}$. Now, by the sixth condition in the definition of the satisfaction relation, we have $(\mathcal{M}, a) \models \exists x\, \mathit{Younger}(\mathit{alice}, x)$.

6. $(\mathcal{M}, a) \models \forall x\, (\mathit{Younger}(\mathit{alice}, x) \vee x = \mathit{alice})$?

Assignments and Free Variables

Observation

Let $\mathcal{M}$ be an S-structure, and a an assignment in $\mathcal{M}$.

In order to check whether an S-formula φ is satisfied in $(\mathcal{M}, a)$, the values $a(x)$ for all variables x that do not occur free in φ are irrelevant.

More precisely: If a' is another assignment in $\mathcal{M}$ such that $a'(x) = a(x)$ for all $x \in \mathrm{free}(\varphi)$, then

$$(\mathcal{M}, a) \models \varphi \ \textit{ if and only if } \ (\mathcal{M}, a') \models \varphi.$$

Conclusion: It suffices to specify $a(x)$ for all $x \in \mathrm{free}(\varphi)$.

Conventions

- Instead of "φ is satisfied in $(\mathcal{M}, a)$", we also say that φ is satisfied in $\mathcal{M}$ under a.

- If φ does not have any free variables, we omit a and say: φ is satisfied in $\mathcal{M}$.

Notation

For formulae φ, we use the notation $\varphi(x_1, \ldots, x_n)$ to indicate that the free variables of φ are precisely $x_1, \ldots, x_n$.

(We omit the brackets if there are no free variables.)

Examples:

- $\varphi(x) = \exists y\, R(x, y)$
- $\psi = \forall x \exists y\, R(x, y)$
- $\chi(x, y, z) = (\exists y\, R(x, y) \vee Q(y)) \wedge P(z)$

Note: The order of the free variables in the list is arbitrary. We can choose any order we like.

Notation

Let $\varphi(x_1, \ldots, x_n)$ be an S-formula.

Given an S-structure $\mathcal{M}$ and elements $d_1, \ldots, d_n \in \mathrm{dom}(\mathcal{M})$, we use the notation

$$\mathcal{M} \models \varphi(d_1, \ldots, d_n)$$

to indicate that $(\mathcal{M}, a) \models \varphi$, where a is any assignment in $\mathcal{M}$ with $a(x_1) = d_1, \ldots, a(x_n) = d_n$.

(We omit the brackets if the list $x_1, \ldots, x_n$ is empty.)

Examples:

- $\mathcal{M} \models \varphi(\text{Bob}, \text{Carol})$ with $\varphi(x, y) = \mathit{Younger}(\mathit{alice}, x) \wedge \neg \mathit{Student}(y)$
- $\mathcal{M} \models \varphi$ with $\varphi = \exists x\, \mathit{Younger}(\mathit{alice}, x)$.

Relational Databases

A relational database consists of one or more tables, e.g.:

Films

id	title	running_time
1	Her	126 min
2	Gravity	91 min
3	Pompeii	105 min
⋮	⋮	⋮

Actors

id	name
1	Joaquin Phoenix
2	Amy Adams
3	Scarlett Johansson
4	Sandra Bullock
⋮	⋮

Cast

film_id	actor_id
1	1
1	2
1	3
2	4
⋮	⋮

- Relational model
- proposed 1970 by Edgar F. Codd

Relational Databases and Structures

A relational database D can be seen as a structure $\mathcal{M}$:

- A table T in D corresponds to a predicate $T^{\mathcal{M}}$ in $\mathcal{M}$ whose arity matches the number of columns of T.
- A row in T corresponds to a tuple in $T^{\mathcal{M}}$.

Films

id	**title**	**running_time**
1	Her	126 min
2	Gravity	91 min
3	Pompeii	105 min
$\vdots$	$\vdots$	$\vdots$

$$
\mathit{Films}^{\mathcal{M}} = \{(1,\ \text{Her},\ 126\text{ min}),\ (2,\ \text{Gravity},\ 91\text{ min}),\ (3,\ \text{Pompeii},\ 105\text{ min}),\ \ldots\}
$$

- The domain of $\mathcal{M}$ consists of all the entries in the database $(1, 2, 3, \ldots, \text{Her}, \text{Gravity}, \ldots)$.

A Note on the Signature

For this translation of databases D into structures $\mathcal{M}$ to work:

- The signature S of $\mathcal{M}$ has to contain a predicate symbol for each table in D.
- The arity of each predicate symbol has to match the number of columns of the corresponding table.

Example: For our film database, the signature is

$$S = \{Films, Actors, Cast\},$$

where

- $Films$ has arity 3,
- $Actors, Cast$ have arity 2.

The Film Database as a Structure

Our film database as an S-structure $\mathcal{M}$:

- Signature $S = \{Films, Actors, Cast\}$
- $\mathrm{dom}(\mathcal{M}) = \{1, 2, 3, \ldots, \text{Her}, \text{Gravity}, \ldots\}$
- Predicates:

$$
\begin{aligned}
Films^{\mathcal{M}} = \{&(1, \text{Her}, 126 \text{ min}),\\
&(2, \text{Gravity}, 91 \text{ min}),\\
&(3, \text{Pompeii}, 105 \text{ min}), \ldots\}
\end{aligned}
\qquad
\begin{aligned}
Cast^{\mathcal{M}} = \{&(1, 1),\\
&(1, 2),\\
&(1, 3),\\
&(2, 4), \ldots\}
\end{aligned}
$$

$$
\begin{aligned}
Actors^{\mathcal{M}} = \{&(1, \text{Joaquin Phoenix}),\\
&(2, \text{Amy Adams}),\\
&(3, \text{Scarlett Johansson}),\\
&(4, \text{Sandra Bullock}), \ldots\}
\end{aligned}
$$

Formulae as Queries

Queries to a database correspond to formulae
(1-1 correspondence between relational algebra and predicate logic)

Example: Consider our film database

The SQL query

```
select title from Films
```

"Output all the entries in the 'title' column of the 'Film' table."

corresponds to

$$\varphi(y) = \exists x \, \exists z \, \mathit{Films}(x, y, z)$$

"Output all y for which $\exists x \, \exists z \, \mathit{Films}(x, y, z)$ is true."

Note: These are the titles returned by the above query.

Example 2

```
select running_time
from Films
where title='Gravity'
```

"Output the running time of Gravity."

$$\varphi(z) = \exists x \, \mathit{Films}(x, \mathit{gravity}, z)$$

Note: This assumes that we also have a constant symbol *gravity* in our signature, and that *gravity* is interpreted by "Gravity". Such constant symbols are typically added to the signature.

Evaluating a Formula in a Structure

Definition

Let $\varphi(x_1, \ldots, x_k)$ be an S-formula, and let $\mathcal{M}$ be an S-structure.

The result of φ in $\mathcal{M}$ is defined as:

$$\varphi(\mathcal{M}) = \{(d_1, \ldots, d_k) \in \mathrm{dom}(\mathcal{M}) \mid \mathcal{M} \models \varphi(d_1, \ldots, d_k)\}.$$

Note: $\varphi(\mathcal{M})$ is sensitive to the order of the variables in the list $x_1, \ldots, x_k$. We will always make sure that this order is clear from the context.

Example 3

```
select A1.name, A2.name
from   Actors A1, Actors A2, Cast C1, Cast C2
where  A1.id = C1.actor_id and
       A2.id = C2.actor_id and
       C1.film_id = C2.film_id
```

"Output all pairs of actors who played in the same film."

$$\varphi(a_1, a_2) = \exists f\ \exists i_1\ \exists i_2\ \big(\ \mathit{Actors}(i_1, a_1) \wedge \mathit{Actors}(i_2, a_2) \wedge \mathit{Cast}(f, i_1) \wedge \mathit{Cast}(f, i_2)\ \big)$$

For our film database $\mathcal{M}$:

$$\varphi(\mathcal{M}) = \big\{(\text{Joaquin Phoenix}, \text{Joaquin Phoenix}), (\text{Joaquin Phoenix}, \text{Amy Adams}), \ldots, (\text{Scarlett Johansson}, \text{Joaquin Phoenix}), \ldots\big\}$$

Sentences

Definition

Let S be a signature. An S-sentence is an S-formula φ with $\text{free}(\varphi) = \emptyset$.

Examples:

- $\forall x \, \exists y \, R(x, y)$ is an $\{R\}$-sentence.
- $\exists y \, R(x, y)$ is *not* an $\{R\}$-sentence.

Recall: If φ is an S-sentence and $\mathcal{M}$ an S-structure, then $\mathcal{M} \models \varphi$ means that φ is satisfied in $\mathcal{M}$.

Semantic Consequence: Motivation

Assumption 1: $Instructor(john)$

Assumption 2: $\forall x \big(Instructor(x) \rightarrow \exists y\, Teaches(x, y)\big)$

Does $\exists z\, Teaches(john, z)$ follow from these two sentences?

- Intuitively: yes!
- But what does it mean precisely that a sentence follows from a set of sentences?

Semantic Consequence

Definition

Fix a signature S, a set F of S-sentences, and an S-sentence φ.

We say φ follows from F (or φ is a semantic consequence of F) if for all S-structures $\mathcal{M}$:

$$\text{If for all } \psi \in F \text{ we have } \mathcal{M} \models \psi, \text{ then } \mathcal{M} \models \varphi.$$

This is denoted by $F \models \varphi$.

Note: We use $\models$ both for the satisfaction relation and for the semantic consequence relation. The left hand side of $\models$ determines which of the two relations we mean.

Example

Signature: $S = \{P, c\}$

$$\{P(c)\} \models \exists x\, P(x)$$

Proof: We have to show that the following is true for all S-structures $\mathcal{M}$: If $P(c)$ is satisfied in $\mathcal{M}$, then $\exists x\, P(x)$ is satisfied in $\mathcal{M}$.

Let $\mathcal{M}$ be an S-structure, and assume that $\mathcal{M} \models P(c)$. Note that $(\mathcal{M}, a) \models P(x)$, where a is an assignment in $\mathcal{M}$ with $a(x) = c^{\mathcal{M}}$.

Since $a = a[x \mapsto c^{\mathcal{M}}]$, we have $(\mathcal{M}, a[x \mapsto c^{\mathcal{M}}]) \models P(x)$. The definition of the satisfaction relation thus yields $(\mathcal{M}, a) \models \exists x\, P(x)$. Since $\exists x\, P(x)$ is a sentence, we can write this as $\mathcal{M} \models \exists x\, P(x)$. $\square$

Example 2

Signature: $S = \{P, Q, c\}$

$$\{P(c), \forall x(P(x) \to Q(x))\} \models Q(c)$$

Proof: We have to show that the following is true for all S-structures $\mathcal{M}$: If the two sentences in the set on the left-hand side of $\models$ are satisfied in $\mathcal{M}$, then $Q(c)$ is satisfied in $\mathcal{M}$.

To this end, let $\mathcal{M}$ be an S-structure, and assume that $\mathcal{M} \models P(c)$ and

$$\mathcal{M} \models \forall x\big(P(x) \to Q(x)\big).$$

The latter implies

$$(\mathcal{M}, a) \models P(x) \to Q(x) \qquad (\star)$$

for all assignments a in $\mathcal{M}$ and, in particular, for any assignment a with $a(x) = c^{\mathcal{M}}$. Fix such an assignment a. Since $\mathcal{M} \models P(c)$, we have $(\mathcal{M}, a) \models P(x)$. Together with $(\star)$, this implies $(\mathcal{M}, a) \models Q(x)$. Since $a(x) = c^{\mathcal{M}}$, we obtain $\mathcal{M} \models Q(c)$. $\square$

Example 3

Signature: $S = \{I, T, john\}$

$$\{I(john), \forall x\big(I(x) \rightarrow \exists y\, T(x,y)\big)\} \models \exists z\, T(john, z)$$

Proof: We have to show that the following is true for all S-structures $\mathcal{M}$: If the two sentences in the set on the left-hand side of $\models$ are satisfied in $\mathcal{M}$, then $\exists z\, T(john, z)$ is satisfied in $\mathcal{M}$.

To this end, let $\mathcal{M}$ be an S-structure, and assume that $\mathcal{M} \models I(john)$ and

$$\mathcal{M} \models \forall x\big(I(x) \rightarrow \exists y\, T(x,y)\big).$$

Note that the latter implies

$$(\mathcal{M}, a) \models I(x) \rightarrow \exists y\, T(x,y) \tag{$\star$}$$

for all assignments a in $\mathcal{M}$ and, in particular, for any assignment a with $a(x) = john^{\mathcal{M}}$. Fix such an assignment a. Since $\mathcal{M} \models I(john)$, we have $(\mathcal{M}, a) \models I(x)$. Together with ($\star$), this implies $(\mathcal{M}, a) \models \exists y\, T(x,y)$. Since $a(x) = \mathrm{john}^{\mathcal{M}}$, we obtain $\mathcal{M} \models \exists y\, T(john, y)$. $\square$

Example 4

Signature: $S = \{P, Q, c\}$

$$\{P(c) \vee Q(c)\} \not\models P(c)$$

"$P(c)$ does not follow from $P(c) \vee Q(c)$"

Proof: We provide a counter-example for $\{P(c) \vee Q(c)\} \models P(c)$, that is, an S-structure $\mathcal{M}$ such that $\mathcal{M} \models P(c) \vee Q(c)$, but $\mathcal{M} \not\models P(c)$.

One such counter-example is the S-structure $\mathcal{M}$ with $\mathrm{dom}(\mathcal{M}) = \{1\}$, $P^{\mathcal{M}} = \emptyset$, $Q^{\mathcal{M}} = \{1\}$, and $c^{\mathcal{M}} = 1$. Since $c^{\mathcal{M}} = 1 \in Q^{\mathcal{M}}$, we have $\mathcal{M} \models Q(c)$ and hence $\mathcal{M} \models P(c) \vee Q(c)$. On the other hand, since $c^{\mathcal{M}} = 1 \notin P^{\mathcal{M}}$, we have $\mathcal{M} \not\models P(c)$. $\square$

Semantic Equivalence

Definition

Let S be a signature, and φ, ψ two S-sentences.

We say φ and ψ are equivalent if they are satisfied in the same S-structures, i.e., if for all S-structures $\mathcal{M}$:

$$\mathcal{M} \models \varphi \text{ if and only if } \mathcal{M} \models \psi.$$

This is denoted by $\varphi \equiv \psi$.

Observation: $\varphi \equiv \psi$ if and only if $\{\varphi\} \models \psi$ and $\{\psi\} \models \varphi$.

Examples

- $\neg\exists x\, \varphi \equiv \forall x\, \neg\varphi$
  $\neg\forall x\, \varphi \equiv \exists x\, \neg\varphi$

- $\forall x\, \varphi \wedge \forall x\, \psi \equiv \forall x(\varphi \wedge \psi)$
  $\forall x\, \varphi \vee \forall x\, \psi \not\equiv \forall x(\varphi \vee \psi)$

- $\exists x\, \varphi \wedge \exists x\, \psi \not\equiv \exists x(\varphi \wedge \psi)$
  $\exists x\, \varphi \vee \exists x\, \psi \equiv \exists x(\varphi \vee \psi)$

- $\exists x\, \forall y\, \varphi \not\equiv \forall y\, \exists x\, \varphi$