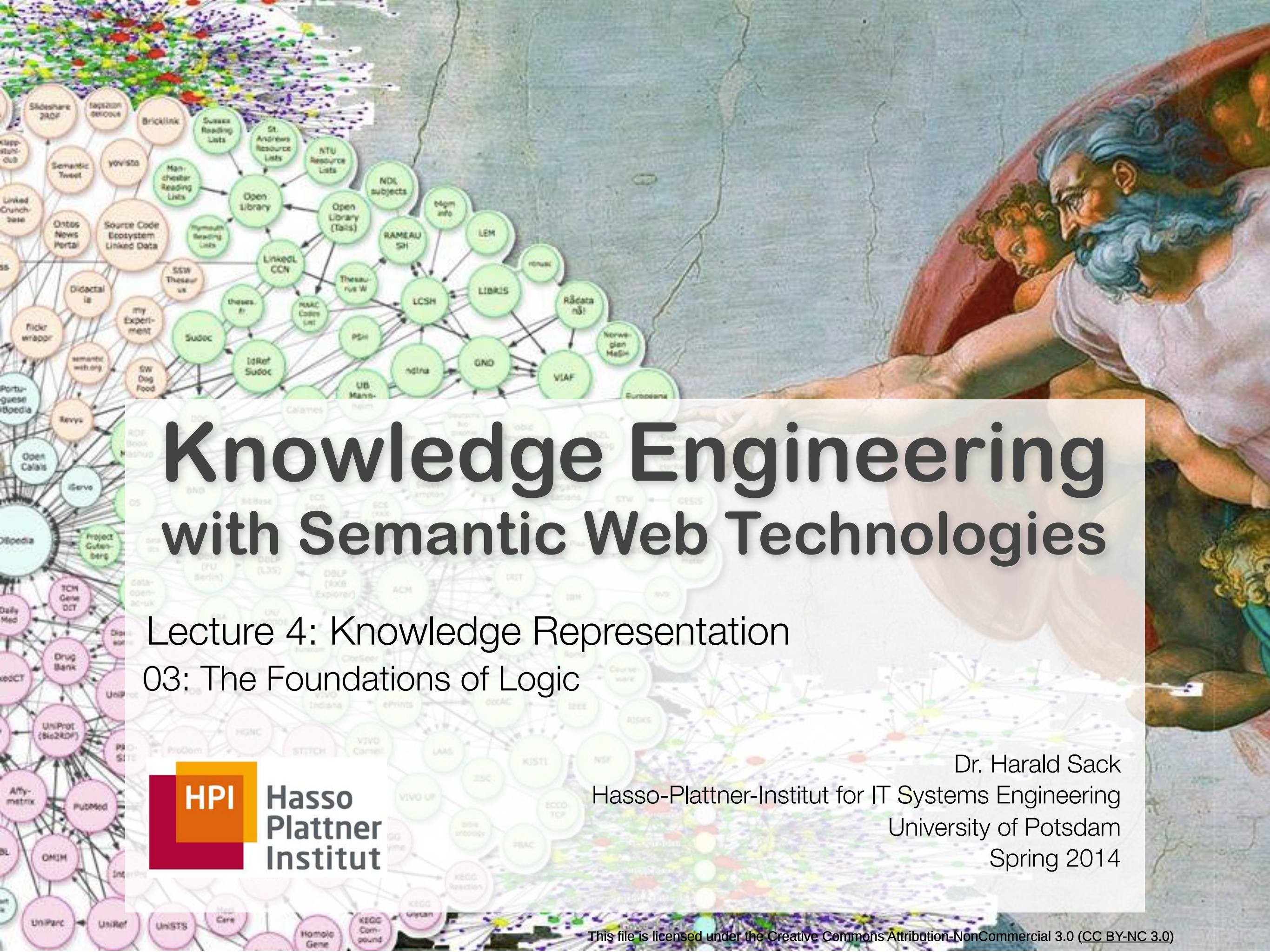


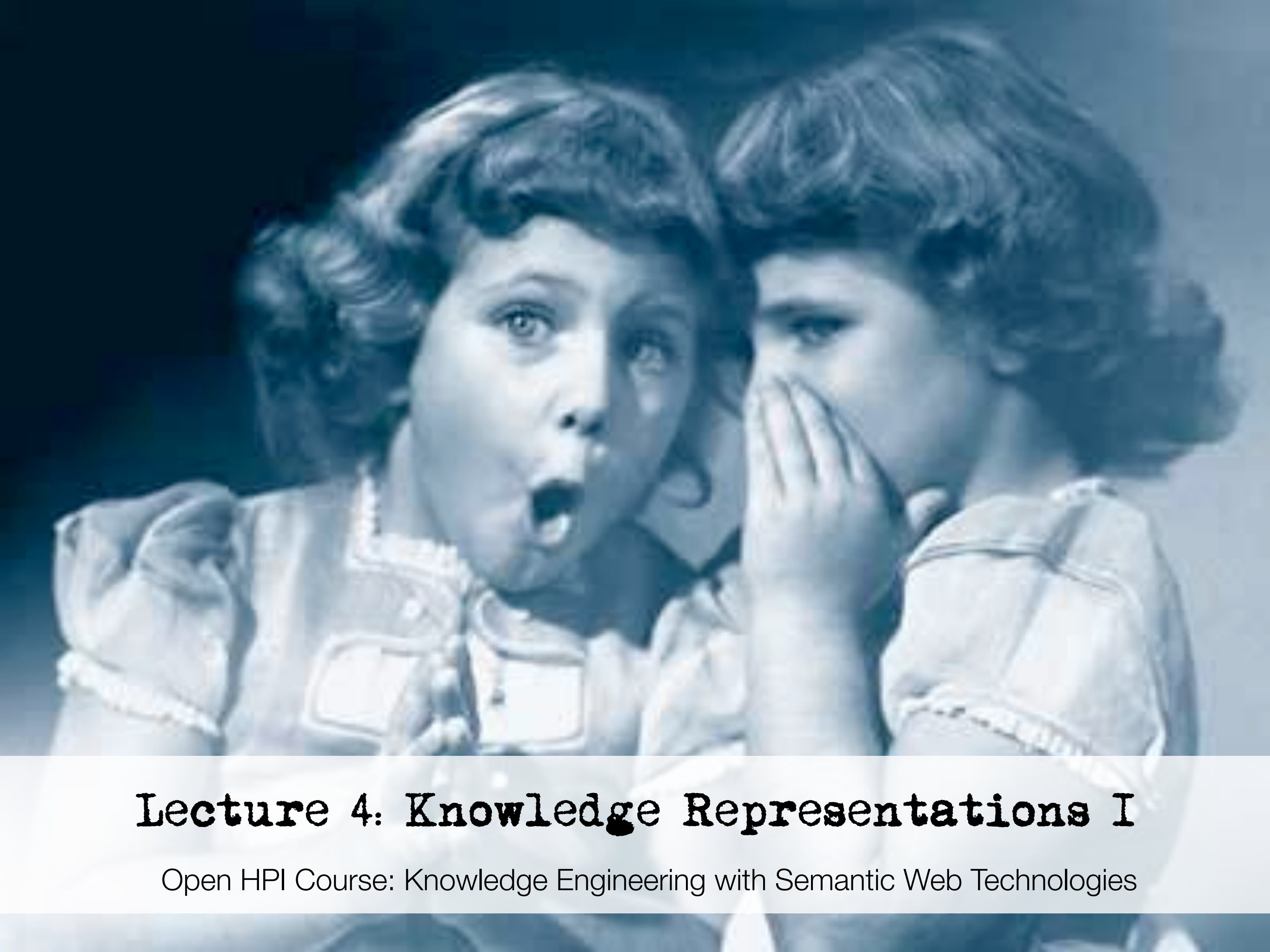


Knowledge Engineering with Semantic Web Technologies

Lecture 4: Knowledge Representation 03: The Foundations of Logic



Dr. Harald Sack
Hasso-Plattner-Institut for IT Systems Engineering
University of Potsdam
Spring 2014



Lecture 4: Knowledge Representations I

Open HPI Course: Knowledge Engineering with Semantic Web Technologies



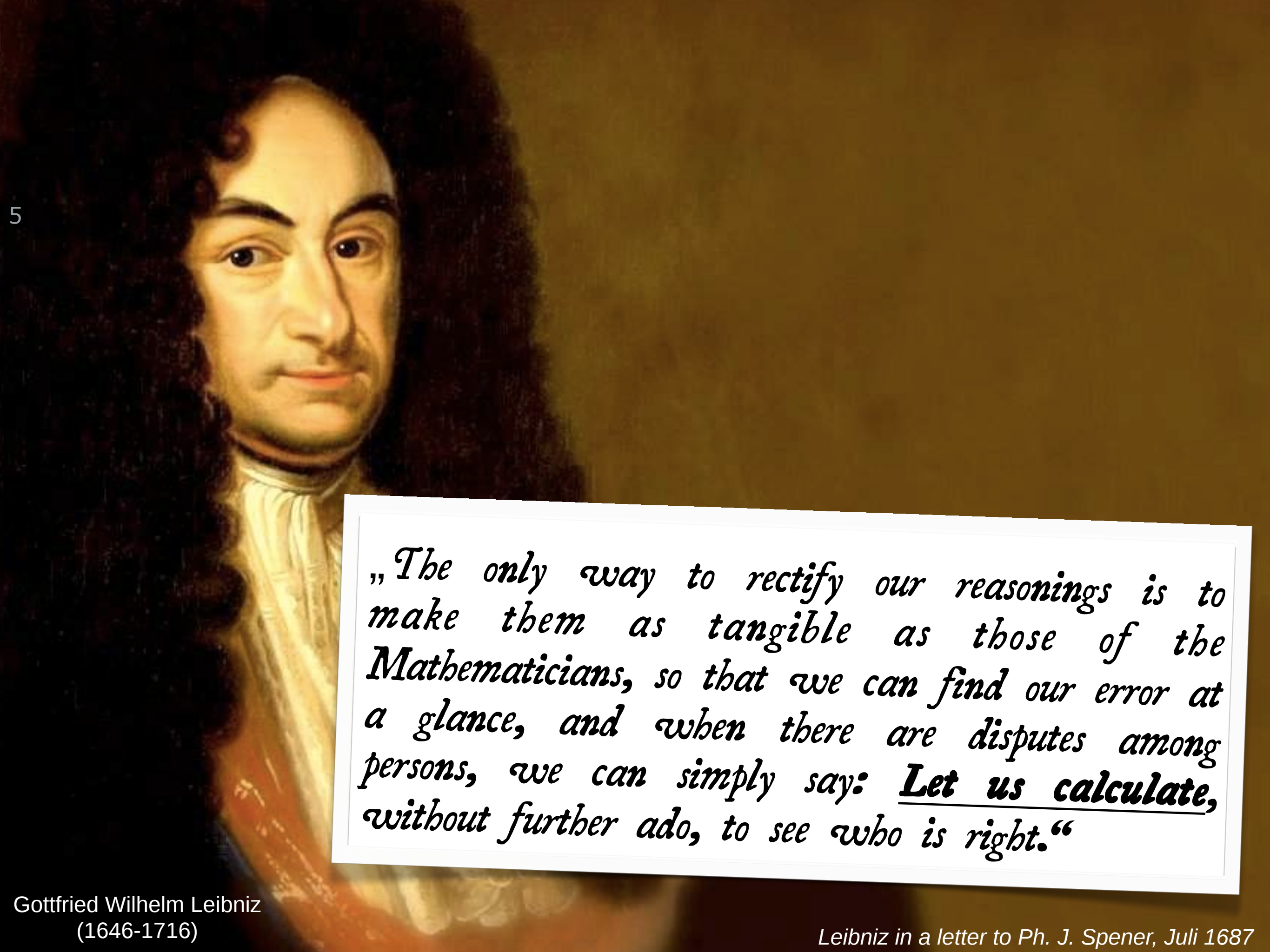
03 Foundations of Logic

Open HPI Course: Knowledge Engineering with Semantic Web Technologies
Lecture 04: Knowledge Representations Pt.1

Foundations of Logic

- 4
- Definition (for our lecture):

Logic is the study of how to make formal correct deductions and inferences.

A portrait of Gottfried Wilhelm Leibniz, a German philosopher, mathematician, and scientist. He is shown from the chest up, wearing a dark, curly wig and a white cravat. The background is a plain, light-colored wall.

„The only way to rectify our reasonings is to make them as tangible as those of the Mathematicians, so that we can find our error at a glance, and when there are disputes among persons, we can simply say: Let us calculate, without further ado, to see who is right.“

- 6
- **Syntax:** *symbols without meaning*
defines rules, how to construct well-formed and valid sequences of symbols (strings)
 - **Semantic:** *meaning of syntax*
defines rules how the meaning of complex sequences of symbols can be derived from atomic sequences of symbols.

Syntax

```
If (i<0) then display ("negative account!")
```

*assignment of
meaning*

*print the message "negative account!", if
the account balance is negative*

Variants of Semantics

- E.g. programming languages

```
FUNCTION f(n:natural):natural;  
BEGIN  
    IF n=0 THEN f:=1  
    ELSE f:=n*f(n-1);  
END;
```

computation of the factorial

intentional semantics

Syntax

- „the meaning intended by the user“
- restricts the set of all possible models (meanings) to the meaning intended by the (human) user

Variants of Semantics

- E.g. programming languages

```
FUNCTION f(n:natural):natural;  
BEGIN  
    IF n=0 THEN f:=1  
    ELSE f:=n*f(n-1);  
END;
```

computation of the factorial

intentional semantics

$$f : n \rightarrow n!$$

formal semantics

Syntax

- aims to express the meaning of symbol sequences (programs) in a **formal language**, in a way that assertions over the symbol sequences (programs) can be proven by the application of deduction rules.

Variants of Semantics

- E.g. programming languages

```
FUNCTION f(n:natural):natural;  
BEGIN  
    IF n=0 THEN f:=1  
    ELSE f:=n*f(n-1);  
END;
```

computation of the factorial

intentional semantics

$f : n \rightarrow n!$

formal semantics

behaviour of the program at execution

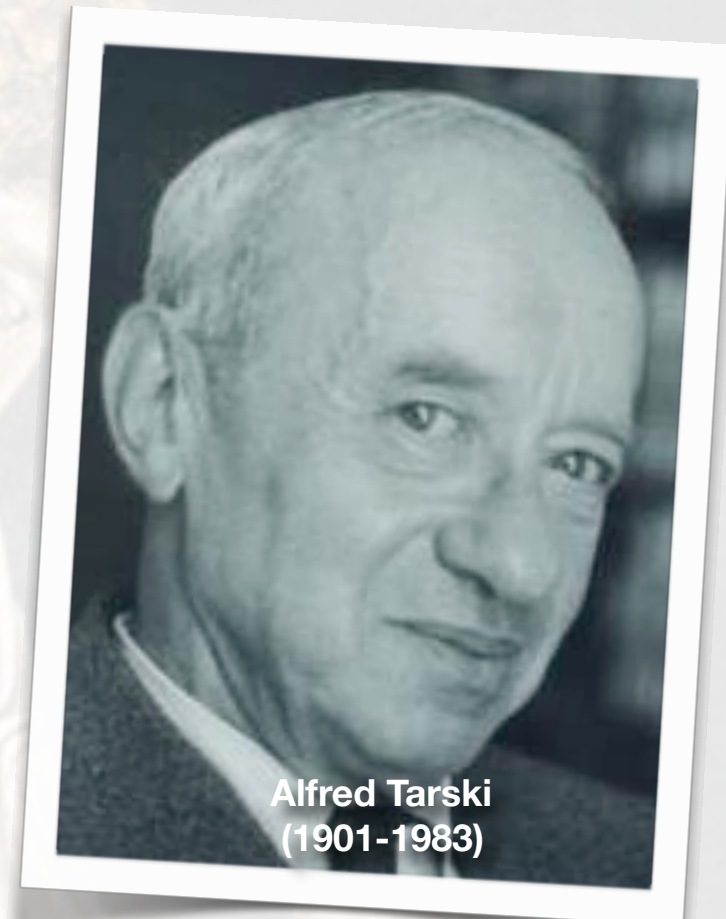
procedural semantics

- the meaning of a language expression (program) is the procedure that takes place internally, whenever the expression does occur.

Syntax

Model-theoretic Semantics

- **Model-theoretic semantics** performs the semantic interpretation of artificial and natural languages by
„identifying meaning with an exact and formally defined interpretation with a model“
- = formal Interpretation with a model
- e.g. model-theoretic semantics of propositional logic
 - assignment of truth values „*true*“ and „*false*“ to atomic assertions and
 - description of logical connectives with *truth tables*



11 ■ **Example:** the language of arithmetics

■ **Syntax:**

■ $x+2 \geq y$ is a formula

■ $x^2+y \geq$ is not a formula

■ **Semantics:**

■ $x+2 \geq y$ is **true** under the interpretation $x = 7, y = 1$

■ $x+2 \geq y$ is **not true** under the interpretation $x = 1, y = 7$

■ Inference/Entailment:

■ **Gaussian Elimination** is an algorithm for arithmetics.

Model-theoretic Semantics

12

- Any logic $\mathbf{L} := (\mathbf{S}, \models)$ consists of
 - (1) a **set of statements** $\mathbf{S}$ and
 - (2) an **entailment relation** $\models$
- Let $\Phi \subseteq \mathbf{S}$ and $\varphi \in \mathbf{S}$: $\Phi \models \varphi$
- „ φ is a logical consequence of Φ “ or
„from the assertions of Φ follows the assertion φ “
- If for 2 assertions $\varphi, \psi \in \mathbf{S}$
both $\{\varphi\} \models \psi$ and $\{\psi\} \models \varphi$,
then both assertions φ and ψ are logically equivalent

$$\varphi \equiv \psi$$

Propositional Logic

logical connective	Name	intentional meaning
$\neg$	Negation	„not“
$\wedge$	Conjunction	„and“
$\vee$	Disjunction	„or“
$\rightarrow$	Implication	„if – then“
$\leftrightarrow$	Equivalence	„if, and only if, then“

- **Logical connectives:** $Op = \{ \neg, \wedge, \vee, \rightarrow, \leftrightarrow, (,) \}$, a set of symbols Σ with $\Sigma \cap Op = \emptyset$ and $\{\text{true}, \text{false}\}$
- Production rules for propositional formulae (propositions):
 - all atomic formulas are propositions (*all elements of Σ*)
 - if φ is a proposition, then also $\neg\varphi$
 - if φ and ψ are propositions, then also $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \rightarrow \psi$, $\varphi \leftrightarrow \psi$
- Priority: $\neg$ prior to $\wedge, \vee$ prior to $\rightarrow, \leftrightarrow$

- How to model facts?

Simple Assertions	Modeling
The moon is made of green cheese	g
It rains	r
The street is getting wet.	n

Composed Assertions	Modeling
if it rains, then the street will get wet.	$r \rightarrow n$
If it rains and the street does not get wet, then the moon is made of green cheese.	$(r \wedge \neg n) \rightarrow g$

■ Interpretation **I**:

Mapping of all atomic propositions to $\{t, f\}$.

- If F is a formula and I an Interpretation, then $I(F)$ is a truth value computed from F and I via truth tables.

$I(p)$	$I(q)$	$I(\neg p)$	$I(p \vee q)$	$I(p \wedge q)$	$I(p \rightarrow q)$	$I(p \leftrightarrow q)$
f	f	t	f	f	t	t
f	t	t	t	f	t	f
t	f	f	t	f	f	f
t	t	f	t	t	t	t

16

- We write $\mathbf{I} \models \mathbf{F}$, if $\mathbf{I}(\mathbf{F}) = \mathbf{t}$,
and call Interpretation $\mathbf{I}$ a **Model** of formula $\mathbf{F}$.
- Rules of Semantics:
 - $\mathbf{I}$ is model of $\neg\varphi$, iff $\mathbf{I}$ is not a model of φ
 - $\mathbf{I}$ is model of $(\varphi \wedge \psi)$, iff $\mathbf{I}$ is a model of φ AND of ψ
 - ...
- Basic concepts:
 - tautology
 - satisfiable
 - refutable
 - unsatisfiable (contradiction)

First Order Logic - FOL

Quantifier	Name	Intentional Meaning
$\exists$	Existential Quantifier	„it exists“
$\forall$	Universal Quantifier	„for all“

- **Operators (logical connectives)** as in propositional logic
- **Variables**, e.g., $X, Y, Z, \dots$
- **Constants**, e.g., $a, b, c, \dots$
- **Functions**, e.g., $f, g, h, \dots$ (incl. arity)
- **Relations / Predicates**, e.g., $p, q, r, \dots$ (incl. arity)

$$(\forall X)(\exists Y) ((p(X) \vee \neg q(f(X), Y)) \rightarrow r(X))$$

FOL: Syntax

- „correct“ formulation of **Terms** from Variables, Constants and Functions:

□ $f(X), g(a, f(Y)), s(a), .(H, T), x_location(Pixel)$

- „correct“ formulation of **Atoms** from Relations with Terms as arguments

□ $p(f(X)), q(s(a), g(a, f(Y))), add(a, s(a), s(a)),$
 $greater_than(x_location(Pixel), 128)$

- „correct“ formulation of **Formulas** from Atoms, Operators and Quantifiers:

□ $(\forall Pixel) (greater_than(x_location(Pixel), 128) \rightarrow red(Pixel))$

- If in doubt, use brackets!
- All Variables should be quantified!

- How to model facts?

- „All kids love icecream.”

- $\forall X: \text{Child}(X) \rightarrow \text{lovesIcecreme}(X)$

- „the father of a person is a male parent.”

- $\forall X \forall Y: \text{isFather}(X,Y) \leftrightarrow (\text{Male}(X) \wedge \text{isParent}(X,Y))$

- „There are (one or more) interesting lectures.”

- $\exists X: \text{Lecture}(X) \wedge \text{Interesting}(X)$

- „The relation ‚isNeighbor‘ is symmetric.”

- $\forall X \forall Y: \text{isNeighbor}(X,Y) \rightarrow \text{isNeighbor}(Y,X)$

■ Structure:

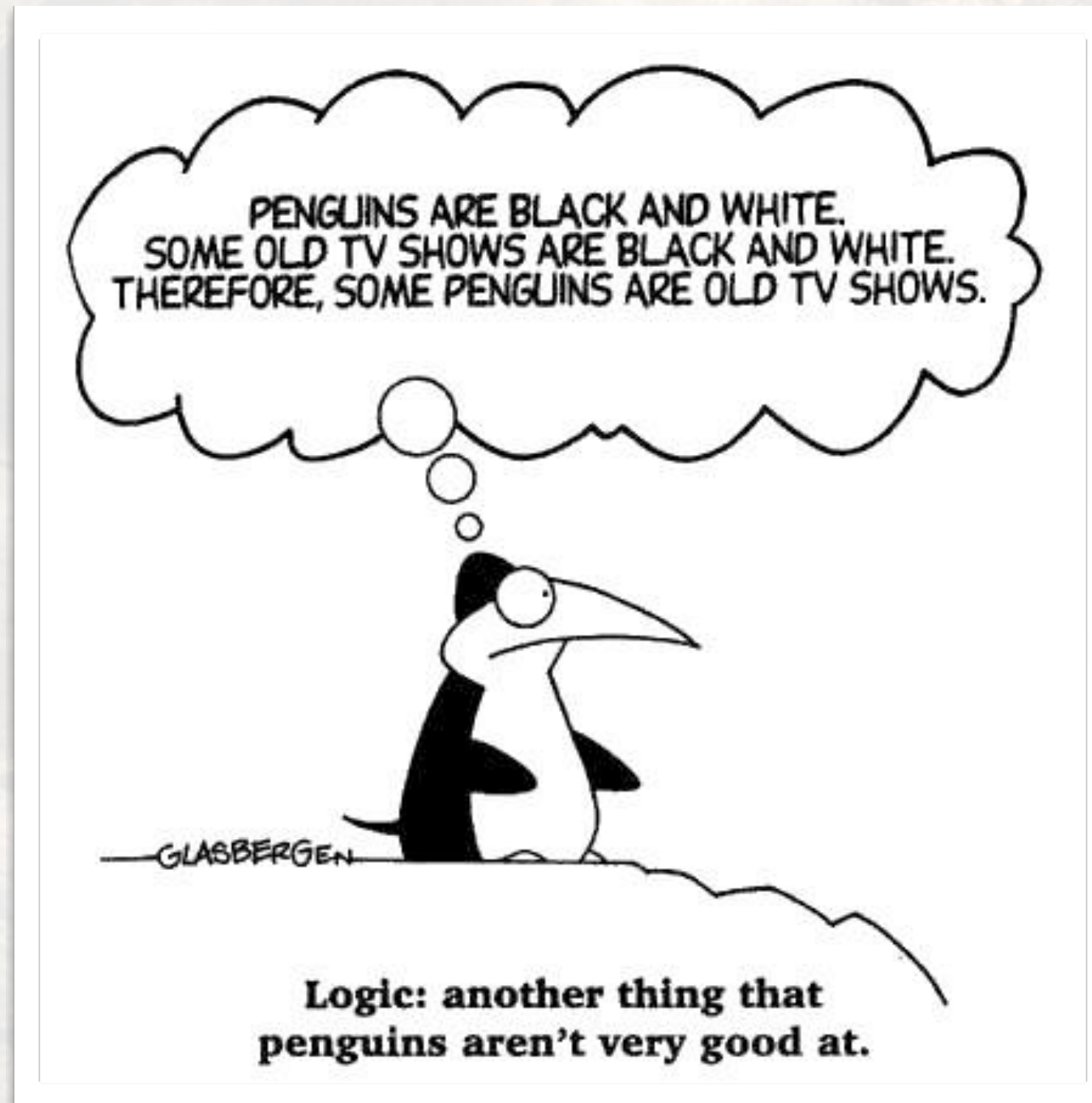
- Definition of a **domain D**.
- **Constant symbols** are mapped to elements of D.
- **Function symbols** are mapped to functions in D.
- **Relation symbols** are mapped to relations over D.

■ Then:

- **Assertions** will become elements of D.
- **Relation symbols** with arguments will become true or false.
- **Logical connectives** and **quantifiers** are treated likewise.

The Penguin Example...

21



■ The Penguin Example

$$\begin{aligned} & ((\forall X)(\text{penguin}(X) \rightarrow \text{blackandwhite}(X)) \\ & \wedge (\exists X)(\text{oldTVshow}(X) \wedge \text{blackandwhite}(X)) \\ &) \rightarrow (\exists X)(\text{penguin}(X) \wedge \text{oldTVshow}(X)) \end{aligned}$$

■ Intentional Semantics?

■ The Penguin Example

$$\begin{aligned} & ((\forall X)(\text{penguin}(X) \rightarrow \text{blackandwhite}(X)) \\ & \wedge (\exists X)(\text{oldTVshow}(X) \wedge \text{blackandwhite}(X)) \\ &) \rightarrow (\exists X)(\text{penguin}(X) \wedge \text{oldTVshow}(X)) \end{aligned}$$

■ Interpretation **I**:

- **Domain**: a set M , containing elements a, b .
- ... no constant or function symbols ...
- We show: the formula is *refutable* (i.e. it is not a tautology):
- If $I(\text{penguin})(a)$, $I(\text{blackandwhite})(a)$,
 $I(\text{oldTVshow})(b)$, $I(\text{blackandwhite})(b)$ is **true**,
 $I(\text{oldTVshow})(a)$ and $I(\text{penguin})(b)$ is **false**,
- then the formula with Interpretation **I** is **false**, d.h. **I** $\neq$ **F**

24

- a **theory** T is a set of formulas.
- an interpretation I is a **model** of T ,
iff $I \models G$ for all formulas G in T .
- a formula F is a **logical consequence** of T ,
iff all models of T are also models of F .
- then we write $T \models F$.
- two formulas F, G are called **logically equivalent**,
iff $\{F\} \models G$ and $\{G\} \models F$.
- then we write $F \equiv G$

Theory $\triangleq$ Knowledge Base

Handwritten text in a stylized, cursive script, possibly representing a sequence of characters or symbols. The text is arranged in four rows, with the first row containing 8 characters, the second row 8 characters, the third row 8 characters, and the fourth row 8 characters. The characters are highly stylized and difficult to decipher.

Spil T 57 Mai 31.9.22

04 Tablaux Algorithm for Automated Reasoning

Open HPI Course: Knowledge Engineering with Semantic Web Technologies
Lecture 04: Knowledge Representations Pt.1