Truth and Justification in Knowledge Representation

Andrei Rodin (RAS/HSE), jww Serge Kovalyov (RAS)

Formal Concept Analysis for Knowledge Discovery 3, Moscow, HSE, June 7, 2019

JTB

Knowledge as Justified True Belief:

Subject S knows that p (where p is a proposition) just in case the following three conditions are satisfied:

JTB

Knowledge as Justified True Belief:

Subject S knows that p (where p is a proposition) just in case the following three conditions are satisfied:

1. p is true

JTB

Knowledge as Justified True Belief:

Subject S knows that p (where p is a proposition) just in case the following three conditions are satisfied:

1. p is true
2. S believes that p

JTB

Knowledge as Justified True Belief:

Subject S knows that p (where p is a proposition) just in case the following three conditions are satisfied:

1. p is true
2. S believes that p
3. S is justified in believing that p.

JTB: Feature 1

JTB identifies knowledge with knowledge of certain proposition or propositions.

This type of knowledge is conventionally referred to as propositional knowledge aka knowledge-that.

JTB: Feature 2

JTB assumes that the truth-value of a given proposition is determined wholly independently of one's knowledge of this proposition. Such an account of truth has a long tradition in logic and has been strongly defended, among other people, by Gottlob Frege. We shall see shortly show that this conception of truth is not commonly accepted.

JTB: Feature 3

According to JTB, a true belief, i.e., one's belief in certain true proposition, by itself does not constitute knowledge. A missing element is justification.

Assuming that a mathematical proof is a special form of justification, for a motivating example think of Bob who is able to state the Pythagorean theorem (provided she understands its meaning and believes it is true) and Alice who is also able to prove it. In terms of JTB theory Alice *knows* the theorem but Bob doesn't. What is at stake here is not the linguistic meaning of "know" but the difference between the two sorts of epistemic states, viz. knowledge and (true) belief (or however one may prefer to call them).

Gettier 1963

$$\frac{16}{64} = \frac{1\cancel{6}}{\cancel{6}4} = \frac{1}{4}$$

$$\frac{26}{65} = \frac{2\cancel{6}}{\cancel{6}5} = \frac{2}{5}$$

Constructive Logic

Constructive Logic

- Truth as the existence of proof; no tripartition of knowledge.

Constructive Logic

- Truth as the existence of proof; no tripartition of knowledge.
- Epistemically-laden conception of logic as the "determination of the best available evidence" (Cohen&Nagel 1934)

Constructive Logic

- Truth as the existence of proof; no tripartition of knowledge.
- Epistemically-laden conception of logic as the “determination of the best available evidence” (Cohen&Nagel 1934)
- Recent developments:
  Provability Logic, Justification Logic (Artemov&Fitting 2019), Proof-Theoretic Semantics, MLTT/HoTT

Under the constructive conception of logic *justification* and (justified) *knowledge* are essential intrinsic features of all logical reasoning rather than specific modalities added on the top of some core logical structures. The constructive approach to knowledge gives justification even more importance by taking it to be constitutive to truth and logic itself. Thus justification is a key element of knowledge both under the JTB and the constructive conceptions.

KR conceptions of knowledge

Ex.: Knowledge is the "whole body of data and information that people bring to bear to practical use in action" (after Jukus et al. 2013)

KR conceptions of knowledge

Ex.: Knowledge is the "whole body of data and information that people bring to bear to practical use in action" (after Jukus et al. 2013)

No mention of justification in the studied CS literature!

Ontology vs. Epistemology

Since late 1980-ies the concept of ontology has been widely used in KR as a formal semantic tool.

Ontology vs. Epistemology

Since late 1980-ies the concept of ontology has been widely used in KR as a formal semantic tool.

In philosophy *ontology* is a discipline that accounts for being/existence.

Ontology vs. Epistemology

Since late 1980-ies the concept of ontology has been widely used in KR as a formal semantic tool.

In philosophy *ontology* is a discipline that accounts for being/existence.

Epistemology accounts for knowledge.

Ontology vs. Epistemology

Since late 1980-ies the concept of ontology has been widely used in KR as a formal semantic tool.

In philosophy *ontology* is a discipline that accounts for being/existence.

Epistemology accounts for knowledge.

Puzzle: Why KR makes use of (formal) ontology but not of (formal) epistemology?

Justification as a practical issue

A regular user of KR system has no means to verify/justify an information obtained via the system provided by this very system. The existing verification technologies are not designed to be available to regular users. Hence the *Post-Truth* and related social/political issues.

Desiderata for KR:

Desiderata for KR:

- Support justification/verification in form of routine procedure available to all users;

Desiderata for KR:

- Support justification/verification in form of routine procedure available to all users;
- Support justification/verification in form, which is specific w.r.t. obtained information: specific evidences rather than general assurances.

A Reason Why Justification is not Supported:

Standard logical tools and gadgets such as Classical first-order logic with the Tarskian formal semantics realise a philosophical conception of logic that doesn't prioritise justification and doesn't support a satisfactory formal treatment of justification (Gettier).

This conception of logic had a strong impact on AI/KR in its early days.

A Solution:

Use alternative conceptions of logic along with their formal implementations for developing theoretical prototypes of KR systems.

Axiomatic Theory after Hilbert

Axiomatic Theory after Hilbert

- A theory is set T of formulas that are interpreted as true statements; such interpretations of formulas are called models of the given theory;

Axiomatic Theory after Hilbert

- A theory is set T of formulas that are interpreted as true statements; such interpretations of formulas are called models of the given theory;
- A theory has a subset of formulas $A \subset T$ called axioms of the given theory;

Axiomatic Theory (cont.)

- A theory comprises set R of syntactic rules, which, in particular, regulate derivations of new T-formulas from some given T-formulas. T-derivations preserve truth in the sense that given any model m of T, they derive from true sentences only true sentences (soundness). T comprises all formulas T-derivable from its axioms (deductive closure).

Axiomatic Theory (cont.)

- A theory comprises set R of syntactic rules, which, in particular, regulate derivations of new T-formulas from some given T-formulas. T-derivations preserve truth in the sense that given any model m of T, they derive from true sentences only true sentences (soundness). T comprises all formulas T-derivable from its axioms (deductive closure).
- T-formulas, which are T-derivable from the axioms (other than the axioms) are called theorems of T. A derivation of theorem from axioms (and by extension also from some intermediate theorems) is called a **proof** of a given theorem.

Axiomatic Theory (cont.)

- A theory comprises set R of syntactic rules, which, in particular, regulate derivations of new T-formulas from some given T-formulas. T-derivations preserve truth in the sense that given any model m of T, they derive from true sentences only true sentences (soundness). T comprises all formulas T-derivable from its axioms (deductive closure).
- T-formulas, which are T-derivable from the axioms (other than the axioms) are called theorems of T. A derivation of theorem from axioms (and by extension also from some intermediate theorems) is called a **proof** of a given theorem.
- The standard notation for the derivability of theorem B from axioms $A_1, \ldots, A_n$ in theory T is as follows: $A_1, \ldots, A_n \vdash_T B$.

Logical Consequence Relation (Tarski 1936)

T-formula B is called a *logical consequence* of T-formulas $A_1, \dots, A_n$, in symbols $A_1, \dots, A_n \models_T B$ just is case every interpretation m that interprets $A_1, \dots, A_n$ as true sentences also interprets B as true sentence.

Proofs as Formal Derivations from Axioms?

Proofs as Formal Derivations from Axioms?

Soundness (w.r.t. certain fixed class of interpretations): if B is derivable in T then B is true in every model (from the fixed class).
Is this a sufficient reason to qualify formal derivations as proofs?

Proofs as Formal Derivations from Axioms?

Soundness (w.r.t. certain fixed class of interpretations): if B is derivable in T then B is true in every model (from the fixed class). Is this a sufficient reason to qualify formal derivations as proofs?

Prawitz: NO! (emphatically). Such a derivation also needs to allow one to *see* that it preserves truth.

Are Axioms Self-Evident Truths?

Are Axioms Self-Evident Truths?

Not under the “modern” Hilbert’s conception of axiomatic theory!

Are Axioms Self-Evident Truths?

Not under the “modern” Hilbert’s conception of axiomatic theory!

Under this conception of the notion of evidence is qualified as psychological and on this basis ruled out of Logic.

Are Axioms Self-Evident Truths?

Not under the "modern" Hilbert's conception of axiomatic theory!

Under this conception of the notion of evidence is qualified as psychological and on this basis ruled out of Logic.

Along with the concept of evidence all epistemic considerations are neglected (Sundolm 2018).

How and Why Ontologies Pop In?

Truthmaker Realism (Smith 1999):

Given a true statement there exists an entity (or entities) that make(s) this statement true.

Once one accepts TMR the notion of formal ontology readily suggests itself as a useful formal semantic tool, which helps one to supplement, and in many applications even to replace, the talk of models and interpretations by talks about some familiar entities that a given theory is supposed to account for. The Tarski-style formal semantics helps to make this ontological talk formal and rigorous.

This is a pragmatic reason to accept some form of TMR and the notion of formal ontology, which may convince even those people, including KR developers, who are not interested in traditional philosophical debates about being and existence. One does not need to explore deep philosophical questions about being in order to use formal ontologies as semantic tools. This explains why the notion of formal ontology became useful and popular in the AI research.

MLTT: Syntax

- 4 basic forms of judgement:
  (i) $A : TYPE$;
  (ii) $A \equiv_{TYPE} B$;
  (iii) $a : A$;
  (iv) $a \equiv_A a'$
- Context : $\Gamma \vdash$ judgement (of one of the above forms)
- no axioms (!)
- rules for contextual judgements; Ex.: dependent product :
  If $\Gamma, x : X \vdash A(x) : TYPE$, then $\Gamma \vdash (\Pi x : X)A(x) : TYPE$

MLTT: Semantics of $t : T$ (Martin-Löf 1984)

- t is an element of set T
- t is a proof (construction) of proposition T ("propositions-as-types")
- t is a method of fulfilling (realizing) the intention (expectation) T
- t is a method of solving the problem (doing the task) T (BHK-style semantics)

MLTT: Definitional aka Judgmental Equality/Identity

$x, y : A$ (in words: x, y are of type A)

$x \equiv_A y$ (in words: x is y by definition)

MLTT: Propositional Equality/Identity

$p : x =_A y$ (in words: x, y are (propositionally) equal as this is evidenced by proof p)

Definitional eq. entails Propositional eq.

$$\frac{x \equiv_A y}{refl_x : x =_A y}$$

Equality Reflection Rule (ER)

$$\frac{p : x =_A y}{x \equiv_A y}$$

ER is not a theorem in the (intensional) MLTT (Streicher & Hofmann 1995).

Extension and Intension in MLTT

- MLTT + ER is called *extensional* MLTT
- MLTT w/out ER is called *intensional*
  (notice that according to this definition intensionality is a negative property!)

Higher Identity Types

- $x', y' : x =_A y$
- $x'', y'' : x' =_{x =_A y} y'$
- $\ldots$

HoTT: the Idea

Types in MLTT are modelled by spaces (up to homotopy equivalence) in Homotopy theory, or equivalently, by higher-dimensional groupoids in Category theory (in which case one thinks of *n*-groupoids as higher homotopy groupoids of an appropriate topological space).

The Homotopical Interpretation

- $x, y : A$
  x, y are points in space A
- $x', y' : x =_A y$
  x', y' are paths between points x, y; $x =_A y$ is the space of all such paths
- $x'', y'' : x' =_{x =_A y} y'$
  x'', y'' are homotopies between paths x', y'; $x' =_{x =_A y} y'$ is the space of all such homotopies
- ...

Point

Definition

Space S is called contractible or space of h-level (-2) when there is point $p : S$ connected by a path with each point $x : A$ in such a way that all these paths are homotopic (i.e., there exists a homotopy between any two such paths).

Homotopy Levels

Definition

We say that S is a space of h-level $n+1$ if for all its points x, y path spaces $x =_S y$ are of h-level n.

Cummulative Hierarchy of Homotopy Types

- -2-type: single point *pt*;
- -1-type: the empty space $\emptyset$ and the point *pt*: truth-values aka (mere) propositions
- 0-type: sets: points in space with no (non-trivial) paths
- 1-type: flat groupoids: points and paths in space with no (non-trivial) homotopies
- 2-type: 2-groupoids: points and paths and homotopies of paths in space with no (non-trivial) 2-homotopies
- ...

Which types are propositions?

Def.: Type P is a *mere proposition* if $x, y : P$ entails $x \equiv y$ (definitionally).

(Internal criterion of logicality)

Truncation

Each type is transformed into a (mere) proposition when one ceases to distinguish between its terms, i.e., *truncates* its higher-order homotopical structure.

Interpretation: Truncation reduces the higher-order structure to a single element, which is **truth-value**: for any non-empty type this value is **true** and for an empty type it is **false**.
The reduced structure is the structure of **proofs** of the corresponding proposition.
To treat a type as a proposition is to ask whether or not this type is instantiated without asking for more.

- "Merely" logical rules (i.e., rules for handling propositions) are instances of more general formal rules, which equally apply to non-propositional types.
- These general rules work as rules of building models of the given theory from certain basic elements which interpret primitive terms (= basic types) of this given theory.

Why HoTT?

Why HoTT?

- HoTT admits the constructive epistemically-laden proof-theoretic semantics intended by Martin-Löf's Type for MLTT (in a slightly modified form).

Why HoTT?

- HoTT admits the constructive epistemically-laden proof-theoretic semantics intended by Martin-Löf's Type for MLTT (in a slightly modified form).
- The cumulative *h*-hierarchy of types made explicit via the homotopical interpretation supports the distinction between propositional, set-level and higher-level types. This distinctive feature of HoTT supports formal constructive representation of objects (of various levels) and propositions "about" these objects within the same framework. Each such object serves as a witness/truthmaker for proposition obtained via the propositional truncation of type where the given object belongs.

Why HoTT? (cont.)

- HoTT comprises a system of formal rules, which are interpreted as logical rules at the propositional *h*-level and as rules for object-construction at all higher levels. This feature of HoTT supports representation various extra-logical procedures (such as material technological procedures) keeping track of the corresponding logical procedures at the propositional level of representation.

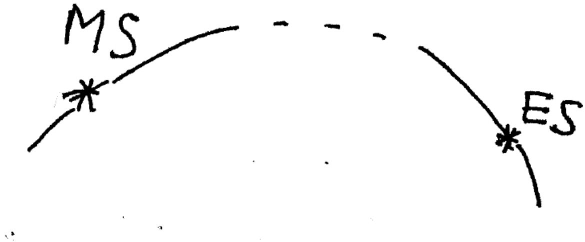
Why HoTT? (cont.)

- HoTT comprises a system of formal rules, which are interpreted as logical rules at the propositional *h*-level and as rules for object-construction at all higher levels. This feature of HoTT supports representation various extra-logical procedures (such as material technological procedures) keeping track of the corresponding logical procedures at the propositional level of representation.
- HoTT/MLTT is computer-friendly, i.e., computationally implementable. Fragments of HoTT/MLTT have been implemented in proof-assistant Coq, program languages AGDA, LEAN and some other products.

Why HoTT? (cont.)

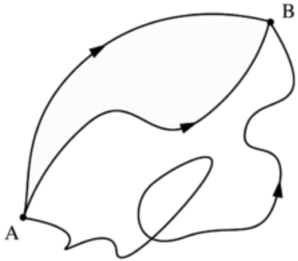
- HoTT comprises a system of formal rules, which are interpreted as logical rules at the propositional *h*-level and as rules for object-construction at all higher levels. This feature of HoTT supports representation various extra-logical procedures (such as material technological procedures) keeping track of the corresponding logical procedures at the propositional level of representation.
- HoTT/MLTT is computer-friendly, i.e., computationally implementable. Fragments of HoTT/MLTT have been implemented in proof-assistant Coq, program languages AGDA, LEAN and some other products.
- HoTT-constructions admit intuitive spatial (homotopical) interpretations that may be used for facilitating human-computer interactions.

The Morning Star is The Evening Star



Venus Homotopically http://philsci-archive.pitt.edu/12116/

also in the Quantum Realm



Remark 1

During the last decade KR technologies have been enriched with approaches based on the Big Data analysis, Machine Learning and artificial Neural Networks. According to a radical opinion, these new approaches make more traditional logical approaches based on the explicit representation of facts and rules hopelessly outdated and irrelevant. We disagree. Because of their possible unpredictable behaviour Neural Networks and other tools of the Big Data analysis can significantly enrich but not replace logical approaches and logical tools in KR.

Remark 2

At the same time we agree that standard logical architectures and formal ontologies, which are presently used in KR, don't provide a sufficient theoretical background for KR because they have no epistemological content. In this paper we explained the relevance of epistemological considerations in logic and KR and then pointed to some recent advances in mathematical logic, more specifically discussing the Homotopy Type theory, that may allow to use logical approaches in KR more effectively.

Thank You! Takk! Спасибо!