Justifications and Wrong Judgements

Giuseppe Primiero

FWO - Research Foundation Flanders
Centre for Logic and Philosophy of Science, Ghent University

Giuseppe.Primiero@Ugent.be
http://www.philosophy.ugent.be/giuseppeprimiero/

Judgement & Justification - 24 September 2012, University of Tampere

(Constructive) Knowledge

A constructive theory of knowledge is based on first-persons acts construing justifications for true propositions:

See e.g. [Martin-Löf, 1984], [Martin-Löf, 1987] [Sundholm, 1997], [Sundholm, 1998], [Sundholm, 1994], [Primiero, 2008], [Schaar v.d., 2009]

Blind Knowledge

In this setting, a certain amount of attention has been dedicated to the explanation of "blind knowledge", the epistemic state referring to a judgment which is correct not in virtue of a proper justification, rather only by chance (derived from Brentano, see e.g. [Sundholm, 2004])

> *"the number of windows-panes in the Leyden City Hall is 8548"*

Wrong Judgements

Valid Justification: Knowledge

$\Downarrow$

Luckily Correct Justification: Blind Knowledge

$\Downarrow$

Wrong Justification: Error (missing!)

$\Downarrow$

Missing Justification: Ignorance

Wrong Justification and Wrong Judgements

The only tentative approach is due to [Sundholm, 2012]:

- errors: ground failures preventing knowledge to be attained;
- mistakes: easily fixable deviations in the epistemic process.

...a lot more!

- Psychology: a very large literature on practical errors, see e.g. [Reason, 1990], [Woods, 2010], [Dekker, 2011];
- Epistemology&Philosophy of Science: error detection and resolution has a crucial importance in paradigm definition and change (Popper, Lakatos, Kuhn, Bayesian epistemology); see e.g. [Mayo, 1996], [Allchin, 2001], [Mayo and Spanos, 2010];
- Logic: defeasible conditions and bounded resources for knowledge can be interpreted as approximations to errors; see e.g. [Williamson, 1992]; [Williamson, 2002]; [Woods, 2004]; [Sundholm, 2012]; [Bonnay and Egre', 2011];
- Applications: error determination in designing principles of specification correctness and technological malfunctioning; see e.g. [Turner, 2011].

Tasks

1. formulate conditional (possible) constructive knowledge;
2. formulate a full characterization of error states for semantics with justifications;
3. provide a formal model of logical processes with error states.

The first task was met with a modal type theory in [Primiero, 2012]. We focus here on the second task. The third task is left to a next stage of this project.

Informational Semantics

We extend the purely constructive semantics of CTT, referring to a more abstract procedural approach

1. judgements express states (intermediary and final);
2. justifications are included in processes regulated by rules;
3. sets of justifications are refereed to as strategies;
4. set of rules are referred to as instructions;
5. it adds *access* and *use* of information to the standard constructive setting; cf. [Allo and Mares, 2011];

Computational Systems with Informational Semantics

Definition

A system $\mathcal{S}$ that processes a procedure $\mathcal{P} = \{S, \Sigma\}$ is composed by :

- a finite set of states $S = \{s_1, \ldots, s_n\}$ (aka situations);
- a finite set of strategies $\Sigma = \{\sigma_1, \ldots, \sigma_n\}$.
- a *strategy* $\Sigma \ni \sigma_i = \{i_1, \ldots, i_n\}$ is the collection of instructions that are used by the system to reach states.
- an *instruction* $i_i \in \sigma_i$ is characterized by a finite set of *rules* $r_1, r_2, \ldots, r_n$ applying to non-terminal states.
- the final state $s_n \in S(\mathcal{S})$ of $\mathcal{P}$ is the goal for the system $\mathcal{G}(\mathcal{S})$.

Computational Systems with Informational Semantics

Definition

A goal $\mathcal{G} := (A\ valid)$ expresses a valid specification in the form of true information A and constitutes the final state of a process $\mathcal{P} := \{p_1, \ldots, p_n\}$ of processes holding at states $s_1, \ldots, s_n$ for contents $A_1, \ldots, A_n$.

$$\frac{\mathcal{P} \text{ is a procedure for } A}{A\ valid}$$

$$\frac{p_1 \ldots p_n \text{ are processes for } A_1, \ldots, A_n}{A\ valid}$$

$$\frac{\text{Information } A_1 \text{ holds} \qquad \text{Use } A_1 \text{ to access } A_2 \qquad \text{Use } A_{n-1} \text{ to access } A_n}{\text{Information } A \text{ holds}}$$

Computational Systems with Informational Semantics

Correspondingly, information *inaccessibility* generates a state of ignorance:

$$\frac{\text{Information } A_1, \ldots, A_{n-1} \text{ holds} \qquad \text{Information } A \text{ cannot be accessed at } n}{A \text{ is not known to hold at states } 1, \ldots, n}$$

Uncertainty

A level of uncertainty is coupled to each error state:

- Total uncertainty on $\mathcal{G}$: a missing procedure $\mathcal{P}$ for $\mathcal{G}$;
- Partial uncertainty on $\mathcal{G}$: a malfunctioning procedure $\mathcal{P}$ for $\mathcal{G}$;
- Wrong Certainty on $\mathcal{G}$: an inappropriate procedure $\mathcal{P}$ for $\mathcal{G}$.

Two cases

An error is a non-realizable procedure $\mathcal{P}$ for accessing an information content $A \in \mathcal{G}$:

Two cases

An error is a non-realizable procedure $\mathcal{P}$ for accessing an information content $A \in \mathcal{G}$:

- wrong coupling:
  - specification side: $\mathcal{P}$ is invalid for A in $\mathcal{G}$;
  - procedure side: $\mathcal{P}$ is inappropriate (though possibly correct) to validate A in $\mathcal{G}$;

Two cases

An error is a non-realizable procedure $\mathcal{P}$ for accessing an information content $A \in \mathcal{G}$:

- wrong coupling:
  - specification side: $\mathcal{P}$ is invalid for A in $\mathcal{G}$;
  - procedure side: $\mathcal{P}$ is inappropriate (though possibly correct) to validate A in $\mathcal{G}$;
- malfunctioning: $\mathcal{P}$ is an incorrect procedure for $\mathcal{G}$ (but when executed correctly, $\mathcal{P}$ is indeed a procedure for accessing content A in $\mathcal{G}$).

Three Main Categories

Definition

Errors are defined according to three main categories:

1. Conceptual Validity: related to the description and design of the goal;
2. Procedural Correctness: related to the procedural aspect;
3. Contextual Admissibility: related to the environment in which the goal is designed and the procedure executed.

Two Main Levels

Definition

.... and two main levels:

1. Internal Level: definitional or structural problem;
2. External Level: execution or environment-based problem.

The General Schema

	Conceptual	**Procedural**	**Contextual**
Internal Level	Goal Description	Process Construction	Dependency Recursion
External Level	Goal design	Data retrieval	Dependency accessibility

Three Types of Error

Type of Error	**Conceptual**	**Material**
Mistakes	Goal Description: Categorization	Goal design: Category Structuring
Failures	Procedure Definition: Form of main process	Procedure Construction: Accessibility of dependent processes
Slips	Algorithm Design: Efficiency	Algorithm execution: Performance

Mistakes or Planning Errors (I)

Definition (Conceptual Mistake)

The pair $(\mathcal{P}, \mathcal{G})$ contains or refers to a ill-defined category:

- incorrectly defined $A \in \mathcal{G}$ in environment, with special case of contradiction;
- non-freshly defined category for $p \in \mathcal{P}$;

Mistakes or Planning Errors (I)

Definition (Conceptual Mistake)

The pair $(\mathcal{P}, \mathcal{G})$ contains or refers to a ill-defined category:

- incorrectly defined $A \in \mathcal{G}$ in environment, with special case of contradiction;
- non-freshly defined category for $p \in \mathcal{P}$;

Definition (Material Mistake)

A pair $(\mathcal{P}, \mathcal{G})$ is given that does not constitute a strategical (sub-)goal.

Conceptual Failures

Definition (Execution Errors)

Errors in the *selection* and *formulation* of rules or strategies:

1. selection of bad rules: an illegal (but possibly correct) execution of the wrong rule r for the current pair p, A is given; EXAMPLE: conjunction elimination rule for the resolution of $A \vee B$;
2. mis-formulation of good rules: a faulty formulation of a valid process; EXAMPLE: application of conjunction elimination from $A \wedge B$ to generate $(A \wedge B) \vee C$ [also accounted for as slip, when occasional].

Material Failures (I)

Definition (Storage Errors)

Errors in the access of data:

1. misaddressed resources: required resources are possibly available in the current environment but are addressed by incorrect or insufficient instructions;
2. non-reachable resources: resources are well-defined but beyond the scope of the procedure, i.e. not available in the current environment.

Material Failures (II)

Definition (Encoding Errors)

Errors due to insufficient data encoding:

- selection of wrong goals;
- selection of rule or procedure with not enough computational depth;
- selection of construction or context with wrong sub-categorization;
- selection of strategy or language with insufficient rules-set.

Material Failures (III)

Definition (Encoding Errors)

Errors due to inaccurate data encoding:

- by inattention: omitting checks, including action on the wrong path of a branching tree is selected, under-use of rule (e.g. missing to go through any branch of a disjunctive rule), missing search for (sub-)goals space and wrong (sub-)typing by accident;

- by over-attention: inappropriate checks, including missing to execute a novel variable declaration, establish a wrong level of abstraction and the overuse of rule (e.g. acting on both branches of a disjunctive rule).

Slips

Material, rule-based errors generated by wrong *applications* of correct rules:

Slips

Material, rule-based errors generated by wrong *applications* of correct rules:

- Exceptions: the rule is applied within a category that accommodates it, but with respect to a construction that represents an exception;
- Rule strength: the rule is applied admitting its global validity, whereas the current context allows only a local validation;
- Redundancy: a rule or strategy is selected on the basis of its previous validity; a rule or strategy is selected that incur in a number of unnecessary steps to reach a goal;
- Rigidity: a fixed set of data or rules is selected for different tasks.

Error Probing Method

The error probing method consists in analyzing the value of the (possibly newly generated) data, according to the typology given above, with two conditions:

1. the test procedure must validate processes on a large account of the environment, i.e. the environment has to be sufficiently large for the validity conditions to be considered robust;
2. the test procedure must be well-defined to establish valid processes; moreover, the test procedure must be itself independent from resources or conditions of the environment it checks.

Defined in Coq (not included here, ask for the code!).

Further work

- A procedural semantics with error-states, based on [Primiero, 2011]
  - failure and error states already designed
  - slip states?
  - including the `check` and `resolve` algorithms

Further work

- A procedural semantics with error-states, based on [Primiero, 2011]
  - failure and error states already designed
  - slip states?
  - including the `check` and `resolve` algorithms
- Applications:
  - currently: errors in computing systems (with Nir Fresco)
  - future: errors generating distrustful networks (based on [Primiero and Taddeo, 2012])
  - future: unsafe programs

References I

Allchin, D. (2001).
Error types.
Perspectives on Science, 9:38–59.

Allo, P. and Mares, E. (2011).
Informational semantics as a third alternative?
Erkenntnis.

Bonnay, D. and Egre', P. (2011).
Knowing One's Limits - An analysis in Centered Dynamic Epistemic Logic.
Synthese, Springer.

Dekker, S. (2011).
Drift into Failure.
Ashgate.

Martin-Löf, P. (1984).
Intuitionistic Type Theory.
Bibliopolis.

References II

Martin-Löf, P. (1987).
Truth of a proposition, evidence of a judgement, validity of a proof.
Synthese, 73(3):407–420.

Mayo, D. (1996).
Error and the Growth of Experimental Knowledge.
Chicago University Press.

Mayo, D. and Spanos, A., editors (2010).
Error and Inference.
Cambridge University Press.

Primiero, G. (2008).
Information & Knowledge - A Constructive Type-Theoretical Apporach, volume 10 of *Logic, Epistemology and the Unity of Sciences*.
Springer.

References III

Primiero, G. (2011).
A multi-modal type system and its procedural semantics for safe distributed programming.
In *Intuitionistic Modal Logic and Applications Workshop (IMLA11)*, Nancy.

Primiero, G. (2012).
A contextual type theory with judgemental modalities for reasoning from open assumptions.
Logique & Analyse, 220.

Primiero, G. and Taddeo, M. (2012).
A modal type theory for formalizing trusted communications.
Journal of Applied Logic, 10:92–114.

Reason, J. (1990).
Human Error.
Cambridge University Press.

References IV

Schaar v.d., M. (2009).
The cognitive act and the first-person perspective: an epistemology for constructive type theory.
Synthese.
10.1007/s11229-009-9708-4.

Sundholm, B. (1994).
Existence, proof and truth-making.
Topoi, 13:117–126.

Sundholm, B. (1997).
Implicit epistemic aspects of constructive logic.
Journal of Logic, Language and Information, 6:191–212.

Sundholm, B. (1998).
Inference, consequence, implication: a constructivist's approach.
Philosophia Mathematica, 6:178–194.

References V

Sundholm, B. (2012).
Error.
Topoi.

Sundholm, G. (2004).
Handbook of Epistemology, chapter Antirealism and the Roles of Truth, pages 437–466.
Kluwer.

Turner, R. (2011).
Specification.
Minds & Machines, 21(2).:135–152.

Williamson, T. (1992).
Inexact knowledge.
Mind, 101(402).:217–241.

Williamson, T. (2002).
Knowledge and its Limits.
Oxford University Press.

References VI

Woods, D.D, D. S. C. R. J. L. S. N., editor (2010).
Behind Human Error.
Ashgate.

Woods, H. (2004).
The Death of Argument: Fallacies in Agent-based Reasoning.
luwer Academic Publishers.