逐次一貫性下の知識伝達を表す直観主義様相論理

Yoichi Hirai

2010-03-03

Overview

Contribution 0: intuitionistic epistemic logic

deduction system, semantics.

soundness, strong completeness, disjunction property, finite model property, decidability.

Contribution 1: formalising sequential consistency

Sequential consistency as $(K_{\mathsf{m}}\varphi \supset K_{\mathsf{m}}\psi) \vee (K_{\mathsf{m}}\psi \supset K_{\mathsf{m}}\varphi)$.

Unexpected application of an intermediate logic.

Contribution 2: decidable abstraction of waitfreely solvable tasks

Is a task waitfreely solvable or not?

Original task: undecidable

Abstract task: decidable

Contribution 1: Formalising Sequential Consistency

Shared memory consistencies [Steinke04]

linearizability

sequential

PRAM

loose

$\Longrightarrow$

A lattice of intermediate logics

$(\varphi \supset \psi) \vee (\psi \supset \varphi)$

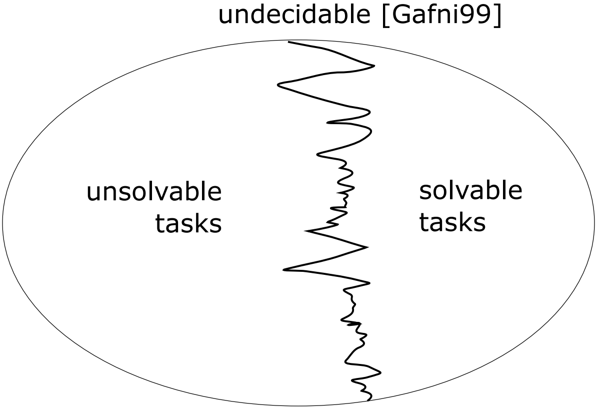
$(K_m\varphi \supset K_m\psi) \vee (K_m\psi \supset K_m\varphi)$

$(K_pK_m\varphi \supset K_pK_m\psi)$
$\vee (K_pK_m\psi \supset K_pK_m\varphi)$

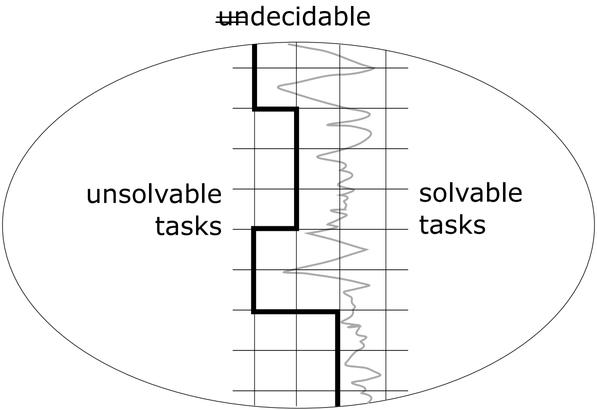
$\emptyset$

m: memory. p: process.

Contribution 2: Decidable abstraction of waitfreely solvable tasks



Contribution 2: Decidable abstraction of waitfreely solvable tasks



Contribution 0

Intuitionistic Epistemic Logic

New informal reading of $K_p\varphi$

Formula $\varphi ::= \bot \mid I \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \supset \varphi \mid K_{\boldsymbol{p}}\varphi$.
Same as classical epistemic logic [Hintikka, 1962].

$K_p\varphi$: p knows φ. (What does "know" mean?)

Classical	In all p's possible worlds, φ is true.
This work	p has received a proof of φ.

c.f. justified, true belief as in Plato: *Theaetetus*.

New informal reading of $K_q K_p \varphi$: COMMUNICATION

$K_q K_p \varphi$: q knows that p knows φ.

Classical	In all q's possible worlds, in all p's possible worlds, φ is true.
This work	q has received a proof of the fact that p has received a proof of φ. Communication from p to q

Formal Semantics = Intuitionistic Logic (+ Knowledge)

model $\langle W, \preceq, (f_p)_{p \in A} \rangle$

$f_p \colon W \to W$: idempotent, descending, monotonic

valuation $\rho \colon \mathrm{PVar} \to \mathcal{P}(W)$ $\qquad$ $\rho(I)$: downward-closed

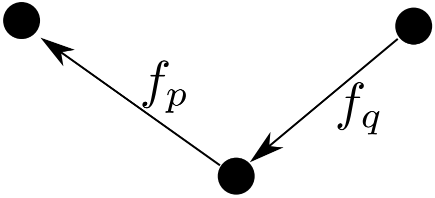
Define $w \models \varphi$ for a state $w \in W$ and a formula φ:

$$
\begin{array}{rcl}
w \models \bot & \Leftrightarrow & \text{never} \\
w \models I & \Leftrightarrow & w \in \rho(I) \\
w \models K_p \psi & \Leftrightarrow & f_p(w) \models \psi \\
w \models \psi_0 \wedge \psi_1 & \Leftrightarrow & \text{both } w \models \psi_0 \text{ and } w \models \psi_1 \text{ hold} \\
w \models \psi_0 \vee \psi_1 & \Leftrightarrow & \text{either } w \models \psi_0 \text{ or } w \models \psi_1 \text{ holds} \\
w \models \psi_0 \supset \psi_1 & \Leftrightarrow & v \models \psi_0 \text{ implies } v \models \psi_1 \text{ for any } v \succeq w.
\end{array}
$$

Formal Semantics = Intuitionistic Logic (+ Knowledge)

model $\langle W, \preceq, (f_p)_{p \in A} \rangle$
$f_p \colon W \to W$: idempotent, descending, monotonic

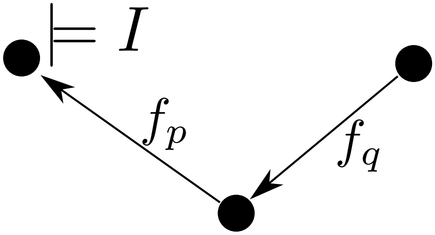
past $\preceq$ future



Formal Semantics = Intuitionistic Logic (+ Knowledge)

model $\langle W, \preceq, (f_p)_{p \in A} \rangle$
$f_p \colon W \to W$: idempotent, descending, monotonic

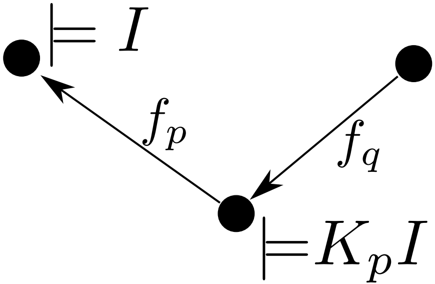
past $\preceq$ future



Formal Semantics = Intuitionistic Logic (+ Knowledge)

model $\langle W, \preceq, (f_p)_{p \in A} \rangle$
$f_p \colon W \to W$: idempotent, descending, monotonic

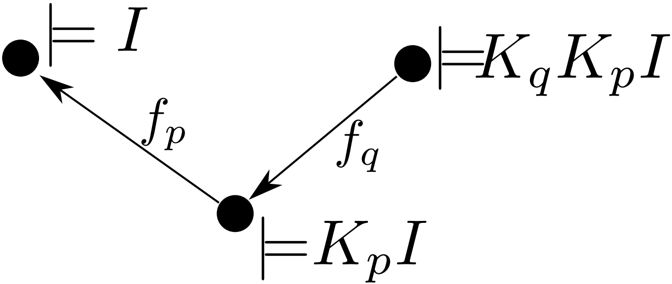
past $\preceq$ future



Formal Semantics = Intuitionistic Logic (+ Knowledge)

model $\langle W, \preceq, (f_p)_{p \in A} \rangle$
$f_p \colon W \to W$: idempotent, descending, monotonic

past $\preceq$ future

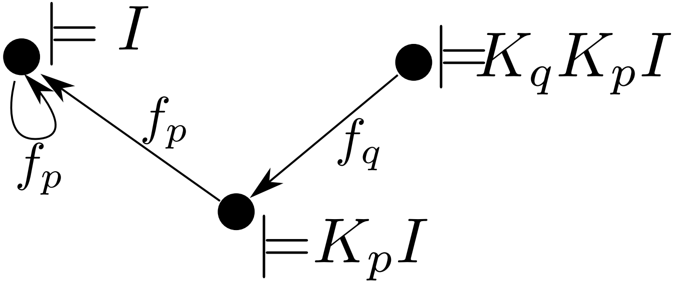


Formal Semantics = Intuitionistic Logic (+ Knowledge)

model $\langle W, \preceq, (f_p)_{p \in A} \rangle$
$f_p \colon W \to W$: idempotent, descending, monotonic

past $\preceq$ future

p's state.

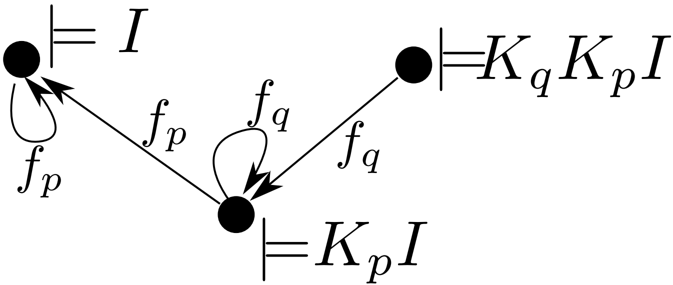


Formal Semantics = Intuitionistic Logic (+ Knowledge)

model $\langle W, \preceq, (f_p)_{p \in A} \rangle$
$f_p \colon W \to W$: idempotent, descending, monotonic

past $\preceq$ future

p's state. q's state.



Deduction System

$$(\mathrm{T})\ \frac{\Gamma \vdash K_p\varphi}{\Gamma \vdash \varphi} \qquad (\text{introspection})\ \frac{\Gamma \vdash K_p\varphi}{\Gamma \vdash K_pK_p\varphi}$$

$$(\text{necessitation})\ \frac{\Gamma \vdash \varphi}{K_p\Gamma \vdash K_p\varphi} \qquad (\vee K)\ \frac{\Gamma \vdash K_p(\varphi \vee \psi)}{\Gamma \vdash K_p\varphi \vee K_p\psi}$$

$$(\mathrm{ax})\ \frac{}{\varphi \vdash \varphi} \qquad (\mathrm{w})\ \frac{\Gamma \vdash \varphi}{\psi,\ \Gamma \vdash \varphi} \qquad (\mathrm{c})\ \frac{\varphi,\ \varphi,\ \Gamma \vdash \varphi'}{\varphi,\ \Gamma \vdash \varphi'} \qquad (\mathrm{e})\ \frac{\Gamma, \varphi, \psi,\ \Gamma' \vdash \varphi'}{\Gamma,\ \psi, \varphi,\ \Gamma' \vdash \varphi'}$$

$$(\wedge\text{-E}_0)\ \frac{\Gamma \vdash \varphi \wedge \psi}{\Gamma \vdash \varphi} \qquad (\wedge\text{-I})\ \frac{\Gamma \vdash \varphi \qquad \Gamma' \vdash \psi}{\Gamma, \Gamma' \vdash \varphi \wedge \psi} \qquad (\wedge\text{-E}_1)\ \frac{\Gamma \vdash \varphi \wedge \psi}{\Gamma \vdash \psi}$$

$$(\vee\text{-I}_0)\ \frac{\Gamma \vdash \varphi}{\Gamma \vdash \varphi \vee \psi} \qquad (\vee\text{-I}_1)\ \frac{\Gamma \vdash \varphi}{\Gamma \vdash \psi \vee \varphi}$$

$$(\vee\text{-E})\ \frac{\Gamma \vdash \psi_0 \vee \psi_1 \qquad \Gamma,\ \psi_0 \vdash \varphi \qquad \Gamma,\ \psi_1 \vdash \varphi}{\Gamma \vdash \varphi}$$

$$(\supset\text{-I})\ \frac{\varphi,\ \Gamma \vdash \psi}{\Gamma \vdash \varphi \supset \psi} \qquad (\supset\text{-E})\ \frac{\Gamma \vdash \psi_0 \supset \psi_1 \qquad \Gamma \vdash \psi_0}{\Gamma \vdash \psi_1} \qquad (\bot\text{-E})\ \frac{\Gamma \vdash \bot}{\Gamma \vdash \varphi}$$

Theoretical Results

Soundness and strong completeness $\Gamma \models \varphi \quad \Longleftrightarrow \quad \Gamma \vdash \varphi$.

Disjunction property $\vdash \varphi \vee \psi \quad \Longrightarrow \quad \vdash \varphi$ or $\vdash \psi$
by extending Aczel's slash relation.

Defining $f_p(\Gamma)$: agent p's view on a set of formulas Γ.
$g_p(\Gamma) = \{\varphi \in \mathsf{Fml} \mid (K_p)^+\varphi \in \Gamma$ and φ does not begin with $K_p\}$,
$f_p(\Gamma) = g_p(\Gamma) \cup K_p g_p(\Gamma) \cup \{\varphi \in \mathsf{Fml} \mid \Gamma \vdash \bot\}$.

(revising, seeking for more general approach.)

Finite model property $M \models \varphi$ for all finite $M \Longleftrightarrow M \models \varphi$ for all M
by modifying subformula relation in [Sato, 1977]

Decidability It is decidable whether $\vdash \varphi$.

Contribution 1

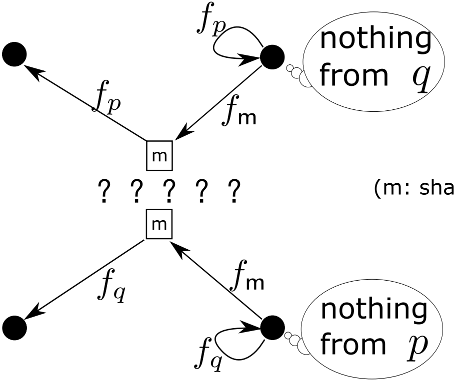
Modelling Sequential Consistency

Need for shared memory consistency

Assumption: full-information

- A message contains all knowledge of its sender.
- Nothing is ever forgotten.

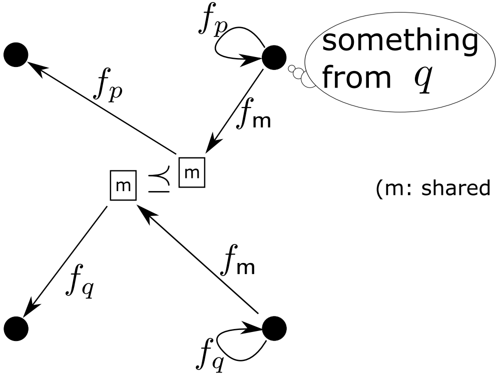
Even under this assumption, no communication is guaranteed between processes.



(m: shared memory)

Essence of Sequential Consistency

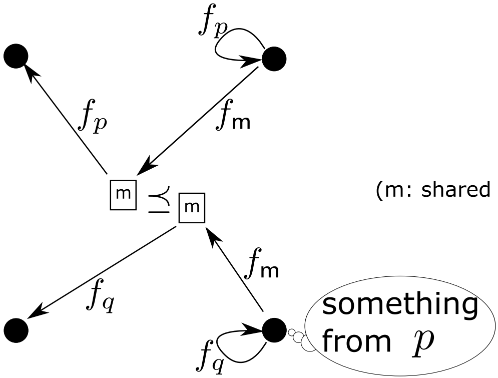
For two memory states, either $\preceq$ or $\succeq$ holds.



(m: shared memory)

Essence of Sequential Consistency

For two memory states, either $\preceq$ or $\succeq$ holds.



(m: shared memory)

Logical Background: logic **Lin** for linear models

Lin $=$ Intuitionistic logic $+ (\varphi \supset \psi) \vee (\psi \supset \varphi)$:

Intuitionistic logic $\subsetneq$ **Lin** $\subsetneq$ Classical logic

Well-known property:

Lin $\vdash \theta \Longleftrightarrow M \models \theta$ for all linear model M

(Linear model: for any two states, either $\preceq$ or $\succeq$ holds.)

A logic **SC** for Sequential Consistency

$$\textbf{SC} = \text{Int. Epistemic logic} + (K_{\mathsf{m}}\varphi \supset K_{\mathsf{m}}\psi) \vee (K_{\mathsf{m}}\psi \supset K_{\mathsf{m}}\varphi):$$

$$\text{Intuitionistic epistemic logic} \subsetneq \textbf{SC} \subsetneq \text{Classical logic}$$

A result:

$$\textbf{SC} \vdash \theta \iff M \models \theta \text{ for all sequential model } M$$

(Sequential model: for any two memory states, $\leq$ or $\geq$ holds.)

An example theorem under sequential consistency

$$\vdash ((K_p K_m K_p I) \wedge K_q K_m K_q J) \supset ((K_q K_p I) \vee K_p K_q J)$$

Informal reading

- p sends a proof of I to m, then m replies to p.
- q sends a proof of J to m, then m replies to q.
- then, p's knowledge has been transmitted to q, or q's knowledge has been transmitted to p.

A proof of this contains 55 steps (cf. it was 1 step).

Ongoing work: finite model property of **SC**

Trying to avoid

logically possible but computationally impossible schedules like

$$\overbrace{t_0 \precnsim t_1 \precnsim t_2 \precnsim \cdots t_n \ldots}^{\text{infinite}} \precnsim t'$$

Status

1. Find a proof strategy.
2. A gap found. Fix or abort.
3. No gaps found. Write and revise until find one. ← current

Contribution 2

Decidable Abstraction of Waitfreely Solvable Tasks

Waitfree Computation

Sequential consistency: restriction on schedules.

Waitfreedom: restriction on programs.

Informal meaning

No process waits for another process.

Formal meaning k exists

each process accesses the shared memory at most k times in any scheduling.

Undecidability and decidability

It is undecidable whether a task is waitfreely solvable [Gafni and Koutsoupias, 1999].
task: a set of allowed (input values, output values) pairs.

It is decidable whether a communication is waitfreely attainable (new)
Waitfree task description ψ is defined with the BNF:

$$\psi ::= K_p\psi \mid \psi \wedge \psi \mid \psi \vee \psi \mid I_p$$

where a proof of I_p represents initial knowledge of p.

yes $(K_q K_p I_p) \vee K_p K_q I_q$

no $(K_q K_p I_p) \wedge K_p K_q I_q$

Future Work

Extending program extraction to concurrent/distributed computation.

- Modelling other memory consistencies: especially PRAM consistency, cache consistency and processor consistency
- typed lambda calculus
  - For multi-core?
  - Type-safe Paxos [Lamport, 1997] implimentation
- Quantify agents $\exists x K_x \varphi$ for program extraction with mobility.
  - Knowledge of π-calculus terms?
- Knowledge of forking and merging agents

This work has been accepted to LPAR-16 held in Dakar, Senegal.

Gafni, E. and Koutsoupias, E. (1999).
Three-processor tasks are undecidable.
SIAM Journal on Computing, 28(3):970–983.

Hintikka, J. (1962).
Knowledge and belief: an introduction to the logic of the two notions.
Cornell University Press.

Lamport, L. (1997).
How to make a correct multiprocess program execute correctly on a multiprocessor.
IEEE Transactions on Computers, 46(7):779–782.

Sato, M. (1977).
A study of Kripke-type models for some modal logics by Gentzen's sequential method.
Publications of the research institute for mathematical sciences, 13:381.