From incompleteness of static theories to completeness of dynamic beliefs in people and in bots

Dusko Pavlovic   Temra Pavlovic

WoLLIC 2023

Outline

Outline

Static incompleteness and dynamic completeness

Universality

Static incompleteness

Dynamic completeness

World of states

String diagrams

Universal language

Self-confirming beliefs

Self-updating explanations

Frege, . . . , Tarski: soundness, completeness

$$\begin{tikzcd} \Theta \vdash \alpha \arrow[d, Rightarrow, bend right, "\text{sound}"'] \\ \mathcal{M} \models \alpha \arrow[u, Rightarrow, bend right, "\text{complete}"'] \end{tikzcd}$$

Frege, . . . , Tarski: soundness, completeness

$$\begin{tikzcd} \Theta \vdash \alpha \arrow[d, Rightarrow, bend right, "\text{sound}"'] \\ \mathcal{M} \models \alpha \arrow[u, Rightarrow, bend right, "\text{complete}"'] \end{tikzcd}$$

Hilbert: “Wir müssen wissen.”

Frege, . . . , Tarski: soundness, completeness

$$\begin{tikzcd}
\Theta \vdash \alpha \arrow[d, Rightarrow, bend right, "\text{sound}"'] \\
\mathcal{M} \models \alpha \arrow[u, Rightarrow, bend right, "\text{complete}"']
\end{tikzcd}$$

Hilbert: "Wir müssen wissen."

Brouwer: Mathematical logic is unsound for $\gamma \vee \neg\gamma$

Frege,…, Tarski: soundness, completeness

$$\begin{tikzcd} \Theta \vdash \alpha \arrow[d, Rightarrow, bend right, "\text{sound}"'] \\ \mathcal{M} \models \alpha \arrow[u, Rightarrow, bend right, "\text{complete}"'] \end{tikzcd}$$

Hilbert: "Wir müssen wissen."

Brouwer: Mathematical logic is unsound for $\gamma \vee \neg\gamma$

Gödel: Peano arithmetic is incomplete for $\gamma \vee \neg\gamma$

Gödel's incompleteness claim

$$\text{universal language } \mathbb{L} \quad \Longrightarrow \quad \exists\varphi.\ \mathcal{M} \models \varphi \ \wedge\ \Theta \not\vdash \varphi$$

Belief completeness claim

$$\left.\begin{array}{c}\text{universal language } \mathbb{L} \\ + \\ \text{belief updating}\end{array}\right\} \Longrightarrow \begin{cases} \forall\ X\text{-process } \mathcal{M} \\ \exists\ X\text{-beliefs } \Theta \text{ in } \mathbb{L} \\ \forall \text{ state } x \text{ in } X. \\ \mathcal{M}_x \models \alpha \iff \Theta_{\mathcal{M}_x} \vdash \alpha \end{cases}$$

Belief completeness claim

$$\left.\begin{array}{c}\text{universal language } \mathbb{L} \\ + \\ \text{belief updating}\end{array}\right\} \implies \left\{\begin{array}{c}\forall\, x : X \\ \forall X \xleftarrow{\mathcal{M}_x} A \xrightarrow{[x]-} B \\ \exists\, \Theta_x : \mathbb{L}. \\ \left[\mathcal{M}_x\right]\alpha \models \beta \\ \Updownarrow \\ \left\{\Theta_x\right\}''\alpha \vdash \beta \\ \wedge \\ \left\{\Theta_x\right\}'\alpha = \Theta_{\mathcal{M}_x\alpha}\end{array}\right.$$

Belief completeness claim

$$\left.\begin{array}{c}\text{universal language } \mathbb{L}\\ +\\ \text{belief updating}\end{array}\right\} \Longrightarrow \left\{\begin{tikzcd} X \arrow[d, dashed, "\Theta"'] & X \times A \arrow[l, "\mathcal{M}"'] \arrow[d, dashed, "\Theta \times id"] \arrow[dr, "{[-]-}"] & \\ \mathbb{L} & \mathbb{L} \times A \arrow[l, "\{\}'"] \arrow[r, "\{\}''"'] & B \end{tikzcd}\right.$$

Universal language

is a type $\mathbb{L}$ equipped with for each pair A, B an *interpreter*

$$\big\{\big\} : \mathbb{L} \times A \to B$$

such that for every $f : X \times A \to B$ there is $\ulcorner f \urcorner : X \to \mathbb{L}$ with

$$f_x(a) \quad = \quad \big\{\ulcorner f \urcorner(x)\big\}(a)$$

Intuition

Think of

- the elements of $\mathbb{L}$ as *belief states*[1]
- the belief states $\ulcorner f \urcorner(x) : \mathbb{L}$ as *explanations* of f

[1] e.g., evolving scientific theories

Self-confirming beliefs

Every believer $g \colon \mathbb{L} \times A \to B$ has a self-confirming belief $\Gamma \colon \mathbb{L}$ with

$$g(\Gamma, a) \quad = \quad \{\Gamma\}\,(a)$$

Static incompleteness proof

- set $\mathbb{L} = \mathbb{N}$ with arithmetic.

Static incompleteness proof

- set $\mathbb{L} = \mathbb{N}$ with arithmetic.
- encode any $p \colon \mathbb{N} \to \mathbb{B}$ as $\ulcorner p \urcorner : \mathbb{N}$ such that $\{\ulcorner p \urcorner\} \; = \; p$

Static incompleteness proof

- set $\mathbb{L} = \mathbb{N}$ with arithmetic.
- encode any $p \colon \mathbb{N} \to \mathbb{B}$ as $\ulcorner p \urcorner : \mathbb{N}$ such that $\{\ulcorner p \urcorner\} = p$
- construct for any $g \colon \mathbb{N} \times A \to \mathbb{B}$ a predicate $\gamma \colon A \to \mathbb{B}$ with

$$g(\ulcorner \gamma \urcorner, a) = \{\ulcorner \gamma \urcorner\}(a) = \gamma(a) \qquad (*)$$

Static incompleteness proof

- set $\mathbb{L} = \mathbb{N}$ with arithmetic.
- encode any $p \colon \mathbb{N} \to \mathbb{B}$ as $\ulcorner p \urcorner : \mathbb{N}$ such that $\{\ulcorner p \urcorner\} \;=\; p$
- construct for any $g \colon \mathbb{N} \times A \to \mathbb{B}$ a predicate $\gamma \colon A \to \mathbb{B}$ with

$$g(\ulcorner \gamma \urcorner, a) = \{\ulcorner \gamma \urcorner\}(a) = \gamma(a) \qquad (*)$$

- specify $\P \colon \mathbb{N} \times A \to \mathbb{B}$ such that $\P(\ulcorner p \urcorner, a) \iff \vdash p(a)$

Static incompleteness proof

- set $\mathbb{L} = \mathbb{N}$ with arithmetic.
- encode any $p \colon \mathbb{N} \to \mathbb{B}$ as $\ulcorner p \urcorner : \mathbb{N}$ such that $\{\ulcorner p \urcorner\} \;=\; p$
- construct for any $g \colon \mathbb{N} \times A \to \mathbb{B}$ a predicate $\gamma \colon A \to \mathbb{B}$ with
$$g(\ulcorner \gamma \urcorner, a) = \{\ulcorner \gamma \urcorner\}(a) = \gamma(a) \qquad (*)$$
- specify $\P \colon \mathbb{N} \times A \to \mathbb{B}$ such that $\P(\ulcorner p \urcorner, a) \iff \vdash p(a)$
- soundness implies
$$\P(\ulcorner p \urcorner, a) = \{\ulcorner p \urcorner\}(a) = p(a) \qquad (\ddagger)$$

Static incompleteness proof

- set $\mathbb{L} = \mathbb{N}$ with arithmetic.
- encode any $p \colon \mathbb{N} \to \mathbb{B}$ as $\ulcorner p \urcorner : \mathbb{N}$ such that $\{\ulcorner p \urcorner\} \; = \; p$
- construct for any $g \colon \mathbb{N} \times A \to \mathbb{B}$ a predicate $\gamma \colon A \to \mathbb{B}$ with
$$g(\ulcorner \gamma \urcorner, a) = \{\ulcorner \gamma \urcorner\}(a) = \gamma(a) \tag{$*$}$$
- specify $\P \colon \mathbb{N} \times A \to \mathbb{B}$ such that $\P(\ulcorner p \urcorner, a) \iff \vdash p(a)$
- soundness implies
$$\P(\ulcorner p \urcorner, a) = \{\ulcorner p \urcorner\}(a) = p(a) \tag{$\ddagger$}$$
- instantiate $g(p, a) = \P(\ulcorner \neg p \urcorner, a)$ to get $\gamma \colon A \to \mathbb{B}$ with
$$\P(\ulcorner \neg\gamma \urcorner, a) \stackrel{(*)}{=} \gamma(a) \stackrel{(\ddagger)}{=} \P(\ulcorner \gamma \urcorner, a)$$

Static incompleteness proof

Since $\P(\ulcorner p \urcorner, a) \iff \vdash p(a)$, this means

$$\vdash \neg\gamma \iff \vdash \neg\gamma$$

and hence

$$\nvdash \neg\gamma \vee \gamma$$

Static incompleteness proof

Since $\P(\ulcorner p \urcorner, a) \iff \vdash p(a)$, this means

$$\vdash \neg\gamma \quad \iff \quad \vdash \neg\gamma$$

and hence

$$\nvdash \neg\gamma \vee \gamma$$

But in the standard model of arithmetic, all γ satisfy

$$\models \neg\gamma \vee \gamma$$

Dynamic completeness setting

Think of functions $f = \langle f', f'' \rangle : X \times A \to X \times B$ as *processes* over the state space X, with

- $f' : X \times A \to X$ is an X-*state update after* A
- $f'' : X \times A \to B$ is an X-*family of* B-*effects* of A

Self-confirming beliefs of families

Every X-family of believers $t\colon X \times \mathbb{L} \times A \to B$ has an X-family of self-confirming beliefs $\ulcorner t \urcorner : X \to \mathbb{L}$, i.e.

$$\left\{\ulcorner t \urcorner(x)\right\}(a) \quad = \quad t_x\left(\ulcorner t \urcorner(x), a\right) \qquad \boxed{*}$$

Self-updating explanations

Every process $f\colon X \times A \to X \times B$ has
a self-updating explanation $[\![f]\!]\colon X \to \mathbb{L}$, i.e.

$$\underbrace{\left\{[\![f]\!]_x\right\}'(a) = [\![f]\!]\left(f'_x(a)\right)}_{\boxed{\dfrac{f' : x \longmapsto f'_x}{\{\}' : [\![f]\!](x) \longmapsto [\![f]\!]\left(f'_x\right)}}} \qquad \underbrace{\left\{[\![f]\!](x)\right\}''(a) = f''_x(a)}_{\boxed{*}}$$

Outline

Wittgenstein,. . . , Kripke: states

(Sachverhalte, . . . , possible worlds)

A *state* is a model of a theory.

Wittgenstein,..., Kripke: states

(Sachverhalte, ..., possible worlds)

A *state* is a model of a theory.

A model of a theory assigns meanings to its expressions, making its theorems true.

Wittgenstein,..., Kripke: states

(Sachverhalte, ..., possible worlds)

A *state* is a model of a theory.

A model of a theory assigns meanings to its expressions, making its theorems true.

If a theory is expressed as a sketch $\Theta = \langle \Sigma, \Gamma \rangle$, then a state is a Γ-preserving functor $\mathcal{M}\colon \Sigma \to \mathrm{Set}$.

States and spaces

Definition

A *state space* is a pair $A = \langle \Theta_A, \mathcal{M}_A \rangle$ where

- $\Theta_A = \langle \Sigma_A, \Gamma_A \rangle$ is a theory, where
  - Σ_A is the signature and
  - Γ_A are the axioms
- $\mathcal{M}_A$ is a reference model of Θ_A, i.e. a Γ_A-preserving functor $\mathcal{M}_A \colon \Sigma_A \to \mathsf{Set}$.

States and spaces

Definition

A *state space* is a pair $A = \langle \Theta_A, \mathcal{M}_A \rangle$ where

- $\Theta_A = \langle \Sigma_A, \Gamma_A \rangle$ is a theory, where
  - Σ_A is the signature and
  - Γ_A are the axioms
- $\mathcal{M}_A$ is a reference model of Θ_A, i.e. a Γ_A-preserving functor $\mathcal{M}_A \colon \Sigma_A \to \mathsf{Set}$.

Remark

The states from a space A are the refinements of $\mathcal{M}_A$. They form an accessible category.

States and spaces

Examples

- Θ = a scientific theory
  $\mathcal{M}$ = settled measurements

States and spaces

Examples

- Θ = a scientific theory
  $\mathcal{M}$ = settled measurements
- Θ = a software specification
  $\mathcal{M}$ = reference implementation

States and spaces

Examples

- Θ = a scientific theory
  $\mathcal{M}$ = settled measurements
- Θ = a software specification
  $\mathcal{M}$ = reference implementation
- Θ = situation (Popper, Barwise)
  $\mathcal{M}$ = observations

States and spaces

Examples continued

- Θ = actor network (Latour et al.)
  $\mathcal{M}$ = protocol

States and spaces

Examples continued

- Θ = actor network (Latour et al.)
  $\mathcal{M}$ = protocol
- Θ = protocol
  $\mathcal{M}$ = session

States and spaces

Examples continued

- Θ = actor network (Latour et al.)
  $\mathcal{M}$ = protocol
- Θ = protocol
  $\mathcal{M}$ = session
- Θ = network layer
  $\mathcal{M}$ = stack interfaces

State transitions

A *state transition* $f: A \to B$ is a pair $f = \langle f_\Theta, f_\Gamma \rangle$ where

- $f_\Gamma : \Gamma_A \leftarrow \Gamma_B$ is a function, whereas
- $f_\Theta : \Theta_A \leftarrow \Theta_B$ is a functor such that

$$\begin{tikzcd}
\Theta_A \arrow[dr, dashed, "\overline{\mathcal{M}}_A"'] & & \Theta_B \arrow[ll, bend right, "f"'] \arrow[dl, dashed, "\overline{\mathcal{M}}_B"] \\
& \mathrm{Set} &
\end{tikzcd}$$

State transitions

A *state transition* $f\colon A \to B$ is a pair $f = \langle f_\Theta, f_\Gamma \rangle$ where

- $f_\Gamma : \Gamma_A \leftarrow \Gamma_B$ is a function, whereas
- $f_\Theta : \Theta_A \leftarrow \Theta_B$ is a functor such that

$$\begin{tikzcd}
\Theta_A \arrow[dr, dashed, "\overline{\mathcal{M}}_A"'] & & \Theta_B \arrow[ll, bend right, "f"'] \arrow[dl, dashed, "\overline{\mathcal{M}}_B"] \\
& \mathrm{Set} &
\end{tikzcd}$$

where

- Θ_A is a Γ_A-completion of Σ_A;

State transitions

A *state transition* $f \colon A \to B$ is a pair $f = \langle f_\Theta, f_\Gamma \rangle$ where

- $f_\Gamma \colon \Gamma_A \leftarrow \Gamma_B$ is a function, whereas
- $f_\Theta \colon \Theta_A \leftarrow \Theta_B$ is a functor such that

$$\begin{tikzcd}
\Theta_A \arrow[dr, dashed, "\overline{\mathcal{M}}_A"'] & & \Theta_B \arrow[ll, bend right, "f"'] \arrow[dl, dashed, "\overline{\mathcal{M}}_B"] \\
& \mathrm{Set} &
\end{tikzcd}$$

where

- Θ_A is a Γ_A-completion of Σ_A; ditto Θ_B

State transitions

A *state transition* $f \colon A \to B$ is a pair $f = \langle f_\Theta, f_\Gamma \rangle$ where

- $f_\Gamma \colon \Gamma_A \leftarrow \Gamma_B$ is a function, whereas
- $f_\Theta \colon \Theta_A \leftarrow \Theta_B$ is a functor such that

$$\begin{tikzcd}
\Theta_A \arrow[dr, "\overline{\mathcal{M}}_A"'] & & \Theta_B \arrow[ll, bend right, "f"'] \arrow[dl, "\overline{\mathcal{M}}_B"] \\
& \mathsf{Set} &
\end{tikzcd}$$

where

- Θ_A is a Γ_A-completion of Σ_A; ditto Θ_B
- $\overline{\mathcal{M}}_A$ is the Γ_A-preserving extension of $\mathcal{M}_A$;

State transitions

A *state transition* $f: A \to B$ is a pair $f = \langle f_\Theta, f_\Gamma \rangle$ where

- $f_\Gamma : \Gamma_A \leftarrow \Gamma_B$ is a function, whereas
- $f_\Theta : \Theta_A \leftarrow \Theta_B$ is a functor such that

$$\begin{tikzcd}
\Theta_A \arrow[dr, dashed, "\overline{\mathcal{M}}_A"'] & & \Theta_B \arrow[ll, bend right, "f"'] \arrow[dl, dashed, "\overline{\mathcal{M}}_B"] \\
& \mathrm{Set} &
\end{tikzcd}$$

where

- Θ_A is a Γ_A-completion of Σ_A; ditto Θ_B
- $\overline{\mathcal{M}}_A$ is the Γ_A-preserving extension of $\mathcal{M}_A$; ditto $\overline{\mathcal{M}}_B$

State transitions

A *state transition* $f: A \to B$ is a pair $f = \langle f_\Theta, f_\Gamma \rangle$ where

- $f_\Gamma : \Gamma_A \leftarrow \Gamma_B$ is a function, whereas
- $f_\Theta : \Theta_A \leftarrow \Theta_B$ is a functor such that

$$\begin{tikzcd}
\Theta_A \arrow[dr, dashed, "\overline{\mathcal{M}}_A"'] & & \Theta_B \arrow[ll, bend right, "f"'] \arrow[dl, dashed, "\overline{\mathcal{M}}_B"] \\
& \mathrm{Set} &
\end{tikzcd}$$

where

- Θ_A is a Γ_A-completion of Σ_A; ditto Θ_B
- $\overline{\mathcal{M}}_A$ is the Γ_A-preserving extension of $\mathcal{M}_A$; ditto $\overline{\mathcal{M}}_B$
- f_Θ maps Γ_B-(co)limits into Γ_A-(co)limits according to f_Γ.

State transitions

Examples

- scientific theory updates
- software specification refinements
- events
- protocol options, suites, stacks, attacks...
- network integration...

Monoidal universe

Let $\mathcal{U}$ be the category of

- state spaces as objects and
- state transitions as morphisms.

with the monoidal product

$$A \otimes B \;=\; \left\langle \Sigma_A + \Sigma_B \,,\, \Gamma_A + \Gamma_B \,,\, [\mathcal{M}_A + \mathcal{M}_B] \right\rangle$$

Monoidal universe with comonoids

Every state A comes with a comonoid

$$A \otimes A \xleftarrow{\;\Delta\;} A \xrightarrow{\qquad} I$$

$$\Sigma_A + \Sigma_A \xrightarrow{[id,id]} \Sigma_A \xleftarrow{\;\perp\;} \perp$$

Cartesian transitions

Let $\mathcal{U}^\bullet$ be the category of

- state spaces and
- comonoid homomorphisms.

Cartesian transitions

Let $\mathcal{U}^{\bullet}$ be the category of

- state spaces and
- comonoid homomorphisms.

Remarks

- $\mathcal{U}^{\bullet}$ is the largest cartesian subcategory of $\mathcal{U}$.
- A transition $f = \langle f_{\Theta}, f_{\Gamma} \rangle : A \to B$ is in $\mathcal{U}^{\bullet}$ if and only if $f_{\Theta} : \Theta_A \leftarrow \Theta_B$ restricts to $f_{\Sigma} : \Sigma_A \leftarrow \Sigma_B$.

Outline

Logical inference

static: Hoare:

$A \vdash B$ $A \left[p \right] B$

Logical inference

static: dynamic:

$A \vdash B$ $\quad\quad$ $\left[p\right] A \models B$

Logical inference

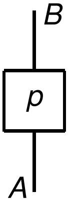
static:	dynamic:	evolving:
$A \vdash B$	$\left[p\right] A \models B$	$q\colon X \times A \to X \times B$

Logical inference in string diagrams

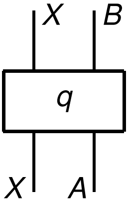
static:

$$\frac{A}{B}$$

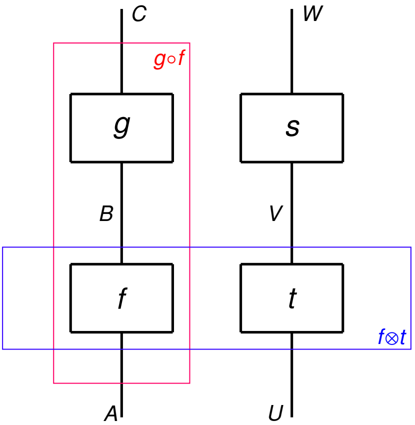
dynamic:



evolving:



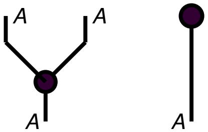
String diagrams are 2-dimensional text



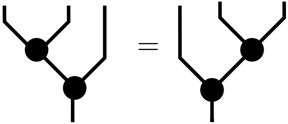
$$(g \circ f) \otimes (s \circ t) = (g \otimes s) \circ (f \otimes t)$$

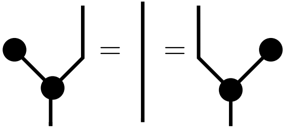
Comonoids are data services

$$A \otimes A \xleftarrow{\;\Delta\;} A \xrightarrow{\quad} I$$

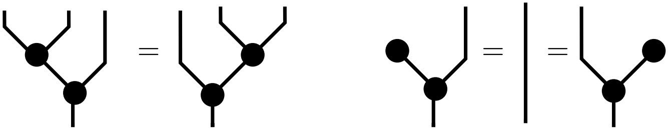


Comonoid equations

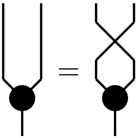




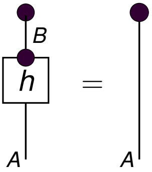
Comonoid equations

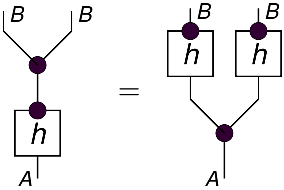


. . . also commutative:



Comonoid homomorphism





Outline

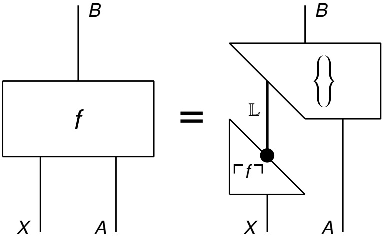
Universal language

is a type $\mathbb{L}$ equipped with for each pair A, B an *interpreter*

$$\{\}: \mathbb{L} \times A \to B$$

such that for every $f: X \times A \to B$ there is $\ulcorner f \urcorner : X \to \mathbb{L}$ with

$$f_x(a) \quad = \quad \{\ulcorner f \urcorner(x)\}(a)$$



Universal language

Example 1

If the states are $A = \langle \Theta_A, \mathcal{M}_A \rangle$ where

- $\Theta_A = \langle \Sigma_A, \Gamma_A \rangle$ is a sketch of a first-order theory
- $\mathcal{M}_A \colon \Sigma_A \to \mathsf{Set}$ is a Γ_A-preserving functor

then $\mathbb{L} = \langle \Theta_\mathbb{L}, \mathcal{M}_\mathbb{L} \rangle$ is

- $\Theta_\mathbb{L}$ = sketch of state transitions
- $\mathcal{M}_\mathbb{L} = 1$

Universal language

Example 2

If the states *A* are computably enumerable types

- (enumerations = computable projections)

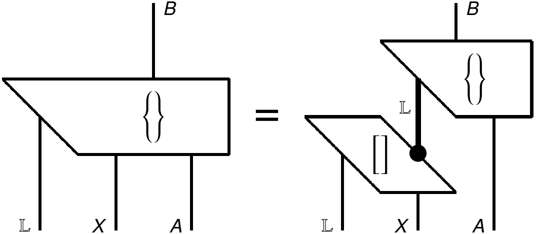
then $\mathbb{L}$ is a universal programming language

- (interpreters = interpreters)

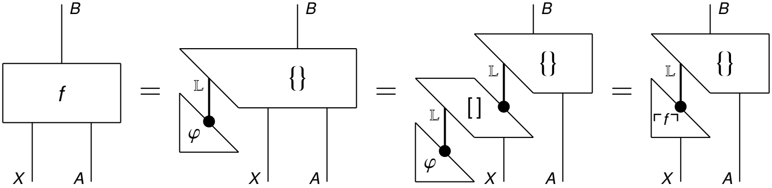
Specializer

Lemma

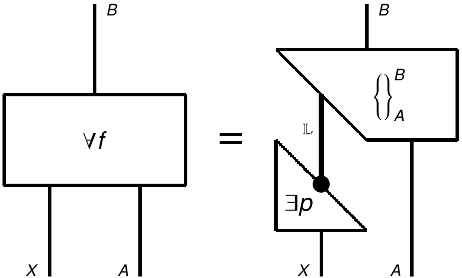
For all $X, A, B \in \mathcal{U}$ there is a strict transition $[]_X^{AB} \in \mathcal{U}^\bullet(\mathbb{L} \times X, \mathbb{L})$ such that



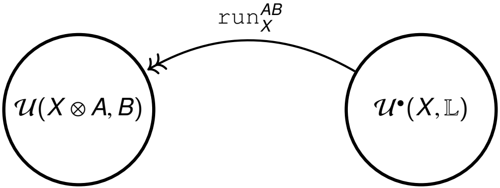
Slicing state changes



Universality = polymorphism



induces program-closure $\mathtt{run}_{\mathbb{P}}^{AB}(p) = \{p\}_A^B$



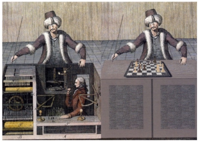
Single-instruction programming language

Programs as Diagrams

From Categorical Computability to Computable Categories

Dusko Pavlovic

20230333



4 What can be computed

20230333

Springer, August 2023
(or dusko.org)

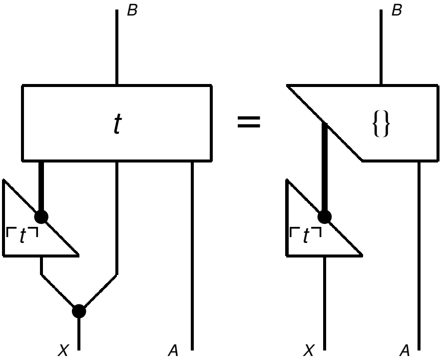
Logic as an evolutionary process

$$\frac{\text{program construction}}{\text{program evaluation}} = \frac{\text{induction}}{\text{deduction}} = \frac{\text{learning}}{\text{prediction}}$$

Outline

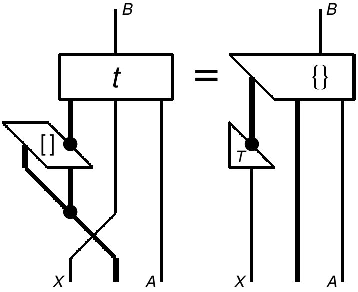
Self-confirming beliefs theorem

Every family of believers $t \in \mathcal{U}(\mathbb{P} \otimes X \otimes A, B)$ has a family of self-confirming beliefs $\ulcorner t \urcorner \in \mathcal{U}^\bullet(X, \mathbb{P})$

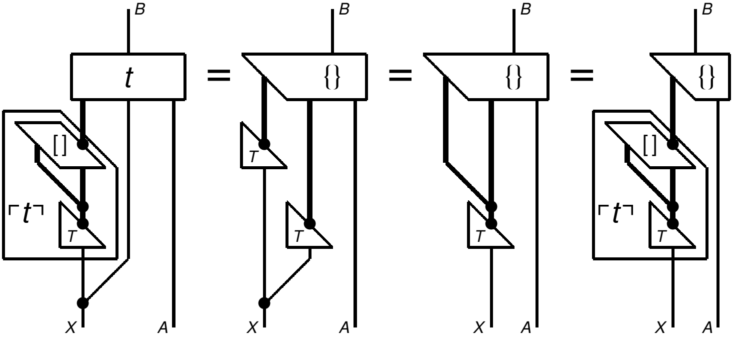


$\boxed{*}$

Self-confirming beliefs construction



Self-confirming beliefs construction



Self-confirming beliefs examples

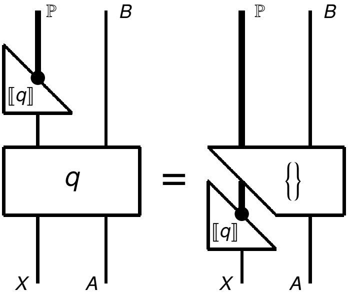
- observer's bias in scientific theories
  - sampling bias (R. Fisher: controls)
  - design bias (J. Neyman: causal inference, allocation)
  - anthropic assumptions (F. Hoyle!)
  - cultural relativism (M. Mead, Levi-Strauss, . . .)
- self-validating software (Knuth, Blum, Rubinsfeld. . .)
- "Credo quia absurdum"
- . . .

Outline

Self-updating explanation theorem

Every process $q\colon X \times A \to X \times B$ has a self-updating explanation $[\![q]\!]\colon X \to \mathbb{L}$, consistent with state changes:

$$[\![q]\!]\,(q'_x(a)) = \left\{[\![q]\!]_x)\right\}'(a) \qquad\qquad q''_x(a) = \left\{[\![q]\!](x)\right\}''(a)$$



Complete beliefs

D. & T. Pavlovic

(In)completeness
World of states
String diagrams
Universal
Self-confirming
Self-updating
Bot beliefs

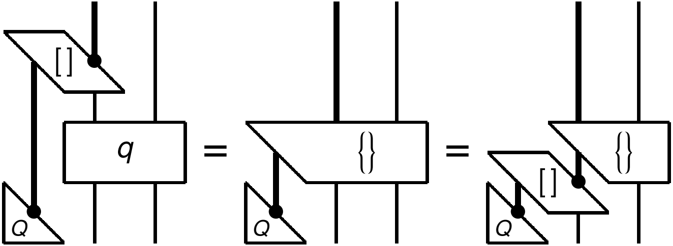
Self-updating explanation **explanation**

Consistency with the state changes means that $[\![q]\!]$ explains q

- not only at the state x by $[\![q]\!]_x$
- but also at the state $q'_x(a)$ by $[\![q]\!]_{q'_x(a)} = \left\{[\![q]\!]_x\right\}'(a)$

for all a in A and x in X.

Self-updating explanation construction



Self-updating explanation example

- X = uses of a social network
  - $u(x) = x$ $\leftsquigarrow$ utility
  - $v(x) = 1 - x$ $\leftsquigarrow$ demand
  - $w(x) = u(x) \cdot v(x) = x - x^2$ $\leftsquigarrow$ valuation
- A = costs of a social network
  - $p =$ price as a supply-demand equilibrium of providers' *influencing* investments and users' *influnced* payments
- B = price update
  - $d =$ price difference

Self-updating explanation example

- adoption process (Brian Arthur)

$$\begin{aligned} q: X \times A &\to X \times B \\ \langle x, p \rangle &\longmapsto \langle x', p' \rangle \\ & \quad x' = x + \delta(p, \tau) \\ & \quad p' = p - d(p, \tau) \end{aligned}$$

where

  - $\tau =$ investment distribution
  - $\delta(p, \tau) =$ part of x believing that $p < x - x^2$
  - $d(p, \tau) =$ adoption discount

Self-updating explanation example

- deployment strategy

$$
\begin{aligned}
[\![q]\!] : X &\to \mathbb{L} \\
x &\longmapsto \begin{cases} \text{advertize}(\tau) & \text{if } p(\tau) \geq x - x^2 \\ \text{monetize}(\tau) & \text{otherwise} \end{cases}
\end{aligned}
$$

Outline

Static incompleteness and dynamic completeness

World of states

String diagrams

Universal language

Self-confirming beliefs

Self-updating explanations

Training and constructing bot beliefs

Empiric problem in machine learning

Why do bots hallucinate?

Foundational problem in belief logic

How do false beliefs evolve:

- myths, legends, religions
- lies, deceptions
- delusions

?

Foundational problem in belief logic

How do false beliefs evolve:

- myths, legends, religions
- lies, deceptions
- delusions

?

How do they survive the reality checks?

General approaches

- induction $\rightsquigarrow$ formal epistemology
  - Bayes, Fisher, Carnap, Popper, Jaynes, Dempster-Shafer, Vapnik, ...
- incomplete information $\rightsquigarrow$ interactive epistemology
  - Harsanyi, Aumann, Brandenburger, DEL (Baltag, van Benthem, van Ditmarsch, ...)

Insight

Belief engineering (campaigning) combines

- belief logic
- algorithm design (programming)
- universal (programming) languages.

Insight

Belief engineering (campaigning) combines

- belief logic
- algorithm design (programming)
- universal (programming) languages.

Universality allows

- unfalsifiable claims
- dynamic completeness.

Tentative answers

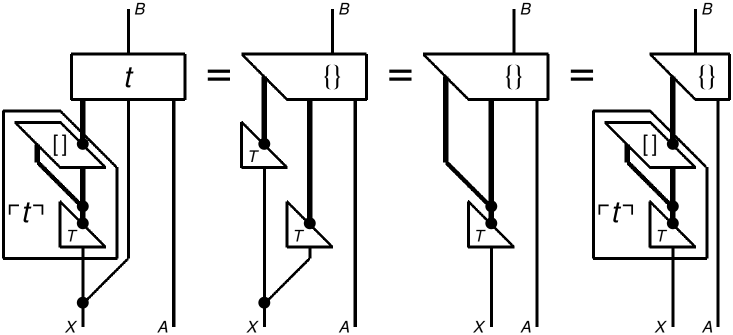
Bots hallucinations arise through

- *data processing*:
  dreaming without sleeping
- **logical constructions**:
  self-confirming explanations

False beliefs persist through

- *data processing*:
  ~~concept amplification~~
- **logical constructions**:
  self-updating (unfalsifiable complete) explanations

“Credo quia absurdum”



"Your friends are my followers"