Predications, fast & slow

Tim.Fernando@tcd.ie

Commonsense-2017, London

Daniel Kahneman, *Thinking, Fast & Slow*, 2011

Predications, fast & slow

Tim.Fernando@tcd.ie

Commonsense-2017, London

Daniel Kahneman, *Thinking, Fast & Slow*, 2011

	subject	predicate	Description Logic
`Tweety flies`	individual	concept	$flies(Tweety)$
`Birds fly`	concept	concept	$bird \sqsubseteq flies$

Predications, fast & slow

Tim.Fernando@tcd.ie

Commonsense-2017, London

Daniel Kahneman, *Thinking, Fast & Slow*, 2011

	subject	predicate	Description Logic
`Tweety flies`	individual	concept	*flies*(*Tweety*)
`Birds fly`	concept	concept	$bird \sqsubseteq flies$

William Woods, Meaning & Links, 2007

extensional *vs* intensional subsumption

Proposal

predication	subsumption
fast	intensional
slow	extensional

Proposal

predication	subsumption
fast	intensional
slow	extensional

1. path $\rightsquigarrow$ string

   $\approx$ model of *Monadic Second-Order Logic* (MSO)

   MSO-sentence $\approx$ regular language (Büchi, Elgot & Trakhtenbrot)

Proposal

predication	subsumption
fast	intensional
slow	extensional

1. path $\rightsquigarrow$ string

   $\approx$ model of *Monadic Second-Order Logic* (MSO)

MSO-sentence $\approx$ regular language (Büchi, Elgot & Trakhtenbrot)

Inheritance & inertia as: *No change without reason* (Principle of Sufficient Reason, Leibniz)

Proposal

predication	subsumption
fast	intensional
slow	extensional

1. path $\rightsquigarrow$ string

   $\approx$ model of *Monadic Second-Order Logic* (MSO)

   MSO-sentence $\approx$ regular language (Büchi, Elgot & Trakhtenbrot)

   Inheritance & inertia as: *No change without reason* (Principle of Sufficient Reason, Leibniz)

2. extensions approximated at bounded but refinable granularity

   What You See Is All There Is (WYSIATI, Kahneman)

   - *satisfaction condition* for *institution* (Goguen & Burstall 1992)

1. Intensions *vs* extensions

2. Paths & MSO

3. Granularity & institutions

Formal Concept Analysis (Wille, Ganter)

	subject	predicate	predication
Descr Logic	individual	concept	$\in$ (ABox)
FCA context	object	attribute	HAS

Formal Concept Analysis (Wille, Ganter)

	subject	predicate	predication
Descr Logic	individual	concept	$\in$ (ABox)
FCA context	object	attribute	HAS

FCA: Given a set D of objects and a set A of attributes,

$$\textsc{intent}(D) \;:=\; \{a \mid (\forall d \in D)\; d \;\textsc{has}\; a\}$$
$$\textsc{extent}(A) \;:=\; \{d \mid (\forall a \in A)\; d \;\textsc{has}\; a\}$$

a *concept* is a pair (D, A) s.t. $A = \textsc{intent}(D)$ &
$D = \textsc{extent}(A)$

Formal Concept Analysis (Wille, Ganter)

	subject	predicate	predication
Descr Logic	individual	concept	$\in$ (ABox)
FCA context	object	attribute	HAS

FCA: Given a set D of objects and a set A of attributes,

$$\begin{aligned} \textsc{intent}(D) &:= \{a \mid (\forall d \in D)\ d \ \textsc{has}\ a\} \\ \textsc{extent}(A) &:= \{d \mid (\forall a \in A)\ d \ \textsc{has}\ a\} \end{aligned}$$

a *concept* is a pair (D, A) s.t. $A = \textsc{intent}(D)$ &
$D = \textsc{extent}(A)$

- equivalently, $A = \textsc{intent}(\textsc{extent}(A))$

- for concepts A and A',

$$\textsc{extent}(A) \subseteq \textsc{extent}(A') \iff A' \subseteq A$$

Formal Concept Analysis (Wille, Ganter)

	subject	predicate	predication
Descr Logic	individual	concept	$\in$ (ABox)
FCA context	object	attribute	HAS

FCA: Given a set D of objects and a set A of attributes,

$$\textsc{intent}(D) := \{a \mid (\forall d \in D)\ d \textsc{ has } a\}$$
$$\textsc{extent}(A) := \{d \mid (\forall a \in A)\ d \textsc{ has } a\}$$

a *concept* is a pair (D, A) s.t. $A = \textsc{intent}(D)$ &
$D = \textsc{extent}(A)$

- equivalently, $A = \textsc{intent}(\textsc{extent}(A))$

- for concepts A and A',

$$\textsc{extent}(A) \subseteq \textsc{extent}(A') \iff A' \subseteq A$$

- for each object d, $\textsc{intent}(\{d\})$ is a concept

Inheritance

$$d \sqsubseteq d' \iff \textsc{intent}(\{d'\}) \subseteq \textsc{intent}(\{d\})$$

$$\frac{d' \ \textsc{has}\ a \qquad d \sqsubseteq d'}{d \ \textsc{has}\ a}$$

$$\textsc{intent}(D) \ := \ \{a \mid (\forall d \in D)\ d \ \textsc{has}\ a\}$$

Inheritance qualified

$$d \sqsubseteq d' \iff \textsc{intent}(\{d'\}) \subseteq \textsc{intent}(\{d\})$$

$$\frac{d' \ \textsc{has}\ a \qquad d \sqsubseteq d'}{d \ \textsc{has}\ a} \qquad \textsc{intent}(D) \ := \ \{a \mid (\forall d \in D)\ d \ \textsc{has}\ a\}$$

– exceptions: `birds fly` but not `penguins` . . .

Inheritance qualified

$$d \sqsubseteq d' \iff \textsc{intent}(\{d'\}) \subseteq \textsc{intent}(\{d\})$$

$$\frac{d' \ \textsc{has}\ a \qquad d \sqsubseteq d'}{d \ \textsc{has}\ a} \qquad \textsc{intent}(D) \ := \ \{a \mid (\forall d \in D)\ d \ \textsc{has}\ a\}$$

– exceptions: `birds fly` but not `penguins` ...

$$\frac{d' \ \textsc{has}\ a \qquad d \ \textsc{is}\ d' \qquad \text{not}(d \ \textsc{has}\ \overline{a})}{d \ \textsc{has}\ a}$$

$$d \ \textsc{has}\ \overline{a} \ \neq \text{not}(d \ \textsc{has}\ a)$$

every penguin is flightless $\neq$ not(every penguin flies)

Inheritance qualified

$$d \sqsubseteq d' \iff \textsc{intent}(\{d'\}) \subseteq \textsc{intent}(\{d\})$$

$$\frac{d' \text{ HAS } a \qquad d \sqsubseteq d'}{d \text{ HAS } a} \qquad \textsc{intent}(D) := \{a \mid (\forall d \in D)\ d \text{ HAS } a\}$$

– exceptions: `birds fly` but not `penguins` ...

$$\frac{d' \text{ HAS } a \qquad d \text{ IS } d' \qquad \text{not}(d \text{ HAS } \overline{a})}{d \text{ HAS } a} \text{ in}(a)$$

$$d \text{ HAS } \overline{a} \neq \text{not}(d \text{ HAS } a)$$

every penguin is flightless $\neq$ not(every penguin flies)

– category mistake: `widespread birds` but `*Tweety` ...

G. CARLSON: individual/kind/stage-level predication

```
Tweety flies
Birds are widespread
Tweety was thirsty
```

G. CARLSON: individual/kind/stage-level predication

```
Tweety flies
Birds are widespread
Tweety was thirsty
```

G. Carlson: individual/kind/stage-level predication

```
Tweety flies
Birds are widespread
Tweety was thirsty
```

G. CARLSON: individual/kind/stage-level predication

```
Tweety flies
Birds are widespread
Tweety was thirsty
```

- generics are less about instances than about

rules & regulations, "causal forces behind instances" (1995)

Carlson & Steedman causes

G. Carlson: individual/kind/stage-level predication

```
Birds are widespread
Tweety was thirsty
```

- generics are less about instances than about

rules & regulations, "causal forces behind instances" (1995)

M. Steedman 2005: temporality is about "causality & goal-directed action"

die(*Tweety*) contra inertial *alive*(*Tweety*)

CARLSON & STEEDMAN causes

G. CARLSON: individual/kind/stage-level predication

```
Birds are widespread
Tweety was thirsty
```

- generics are less about instances than about

rules & regulations, "causal forces behind instances" (1995)

M. STEEDMAN 2005: temporality is about "causality & goal-directed action"

$die(Tweety)$ contra inertial $alive(Tweety)$

$$\frac{alive(Tweety)@t \qquad tSt' \qquad \text{not } opp(alive(Tweety)@t)}{alive(Tweety)@t'}$$

Intensions from instances/extensions

From $\textsc{intent}(\{d\}) = A$ with

$$\begin{aligned} A &= \textsc{intent}(\textsc{extent}(A)) \\ A \sqsubseteq A' &\iff \textsc{extent}(A) \subseteq \textsc{extent}(A') \end{aligned}$$

Intensions from instances/extensions to strings/causes

From $\textsc{intent}(\{d\}) = A$ with

$$A \;=\; \textsc{intent}(\textsc{extent}(A))$$
$$A \sqsubseteq A' \iff \textsc{extent}(A) \subseteq \textsc{extent}(A')$$

to strings

$$A_1 \cdots A_n \;\text{ with }\; \begin{array}{l} A_n = A \\ A_{n-1} \approx \textsc{intent}(\{d'\}) \text{ for } d'Sd \\ \cdots \end{array}$$

S from top/past for inferences such as

$$\frac{a \in A_i \qquad \overline{a} \notin A_{i+1}}{a \in A_{i+1}} \; 1 \leq i < n$$

for $d'Sd$ saying d IS d'.

Attributes in strings

FCA	string $A_1 \cdots A_n$	
d HAS a	$a \in A_i$	
object d	position i	
attribute a		

Attributes in strings as predicates

FCA	string $A_1 \cdots A_n$	MSO
d HAS a	$a \in A_i$	$i \in [\![P_a]\!]$
object d	position i	$i \in \{1, \ldots, n\}$
attribute a		unary predicate P_a

$$[\![P_a]\!] \; = \; \{i \in \{1, \ldots, n\} \mid a \in A_i\}$$

Attributes in strings as predicates

FCA	string $A_1 \cdots A_n$	$\mathrm{MSO}_{\mathcal{A}}$
d HAS a	$a \in A_i$	$i \in [\![P_a]\!]$
object d	position i	$i \in \{1, \ldots, n\}$
attribute $a \in \mathcal{A}$	$A_i \subseteq \mathcal{A}$	unary predicate P_a

$$
\begin{aligned}
[\![P_a]\!] &= \{i \in \{1, \ldots, n\} \mid a \in A_i\} \\
A_i &= \{a \in \mathcal{A} \mid i \in [\![P_a]\!]\}
\end{aligned}
$$

Attributes in strings as predicates

FCA	string $A_1 \cdots A_n$	$\mathrm{MSO}_{\mathcal{A}}$
d HAS a	$a \in A_i$	$i \in [\![P_a]\!]$
object d	position i	$i \in \{1, \ldots, n\}$
attribute $a \in \mathcal{A}$	$A_i \subseteq \mathcal{A}$	unary predicate P_a

$$
\begin{aligned}
[\![P_a]\!] &= \{i \in \{1, \ldots, n\} \mid a \in A_i\} \\
A_i &= \{a \in \mathcal{A} \mid i \in [\![P_a]\!]\} \\
[\![S]\!] &= \{(1,2), \ldots, (n-1, n)\}
\end{aligned}
$$

$$\mathrm{MSO}_{\mathcal{A}}\text{-model} \;=\; \text{string over the alphabet } 2^{\mathcal{A}}$$

$$A_1 \cdots A_n \models \exists x(P_a x \wedge \forall y \neg y S x) \iff a \in A_1$$

MSO-sentence = regular language (BÜCHI ...)

Paths back up S

$$\frac{a \in A_i \qquad \overline{a} \notin A_{i+1}}{a \in A_{i+1}}$$

$$(P_a y \wedge \neg P_{\overline{a}} x \wedge ySx) \supset P_a x$$

Paths back up S

$$\frac{a \in A_i \qquad \overline{a} \notin A_{i+1}}{a \in A_{i+1}} \qquad\qquad (P_a y \wedge \neg P_{\overline{a}} x \wedge ySx) \supset P_a x$$

$$P_a x \quad \mapsto \quad \exists X(Xx \wedge \mathsf{path}_a(X))$$

$$\mathsf{path}_a(X) \;:=\; \underbrace{\forall x(Xx \supset \exists y(ySx \wedge Xy) \vee P_a x)}_{X \text{ backs up}^S \text{ until } a} \;\wedge\; \underbrace{\neg\exists x(Xx \wedge P_{\overline{a}} x)}_{X \text{ avoids } \overline{a}}$$

Paths back up S

$$\frac{a \in A_i \qquad \overline{a} \notin A_{i+1}}{a \in A_{i+1}} \qquad\qquad (P_a y \wedge \neg P_{\overline{a}} x \wedge ySx) \supset P_a x$$

$$P_a x \quad \mapsto \quad \exists X(Xx \wedge \text{path}_a(X))$$

$$\text{path}_a(X) \;:=\; \underbrace{\forall x(Xx \supset \exists y(ySx \wedge Xy) \vee P_a x)}_{X \text{ backs up}^S \text{ until } a} \;\wedge\; \underbrace{\neg\exists x(Xx \wedge P_{\overline{a}} x)}_{X \text{ avoids } \overline{a}}$$

$$\frac{a \in A_i \qquad o(a) \notin A_{\color{red}{i}}}{a \in A_{i+1}} \qquad\qquad (P_a y \wedge \neg P_{o(a)} {\color{red}{y}} \wedge ySx) \supset P_a x$$

Paths back up S

$$\frac{a \in A_i \qquad \overline{a} \notin A_{i+1}}{a \in A_{i+1}} \qquad\qquad (P_a y \wedge \neg P_{\overline{a}} x \wedge ySx) \supset P_a x$$

$$P_a x \quad \mapsto \quad \exists X (Xx \wedge \mathrm{path}_a(X))$$

$$\mathrm{path}_a(X) \ := \ \underbrace{\forall x (Xx \supset \exists y (ySx \wedge Xy) \vee P_a x)}_{X \text{ backs up}^S \text{ until } a} \ \wedge \ \underbrace{\neg \exists x (Xx \wedge P_{\overline{a}} x)}_{X \text{ avoids } \overline{a}}$$

$$\frac{a \in A_i \qquad o(a) \notin A_i}{a \in A_{i+1}} \qquad\qquad (P_a y \wedge \neg P_{o(a)} y \wedge ySx) \supset P_a x$$

$$P_a x \quad \mapsto \quad \exists X (Xx \wedge \mathrm{path}^o_a(X))$$

$$\mathrm{path}^o_a(X) \ := \ \forall x (Xx \supset \exists y (ySx \wedge Xy \wedge \neg P_{o(a)} y) \vee P_a x)$$

Finite state transducers down S

Fix a finite set In of inheritable/inertial attributes.

$$\frac{a \in A_i \qquad \overline{a} \notin A_{i+1}}{a \in A_{i+1}}$$

state = subset q of In in previous position (initially $\emptyset$)

$$q \xrightarrow{A:A'} q' \quad \text{where} \quad \begin{aligned} A' &:= A \cup \{a \in q \mid \overline{a} \notin A\} \\ q' &:= A' \cap In \end{aligned}$$

Finite state transducers down S

Fix a finite set In of inheritable/inertial attributes.

$$\frac{a \in A_i \qquad \overline{a} \notin A_{i+1}}{a \in A_{i+1}}$$

state = subset q of In in previous position (initially $\emptyset$)

$$q \xrightarrow{A:A'} q' \quad \text{where} \quad \begin{array}{l} A' := A \cup \{a \in q \mid \overline{a} \notin A\} \\ q' := A' \cap In \end{array}$$

$$\frac{a \in A_i \qquad o(a) \notin A_i}{a \in A_{i+1}}$$

$$q \xrightarrow{A:A'} q' \quad \text{where} \quad \begin{array}{l} A' := A \cup q \\ q' := \{a \in A' \cap In \mid o(a) \notin A\} \end{array}$$

A causal ontology

Trade GALOIS connection

$$D \subseteq \textsc{extent}(A) \iff A \subseteq \textsc{intent}(D)$$

for an ontology based on S-change

$$(P_a y \wedge ySx) \supset (P_a x \vee yR_a x) \qquad yR_a x := \begin{cases} P_{\overline{a}} x & \text{for kinds} \\ P_{o(a)} y & \text{for time} \end{cases}$$

A causal ontology

Trade GALOIS connection

$$D \subseteq \textsc{extent}(A) \iff A \subseteq \textsc{intent}(D)$$

for an ontology based on S-change

$$(P_a y \wedge ySx) \supset (P_a x \vee yR_a x) \qquad yR_a x := \begin{cases} P_{\overline{a}} x & \text{for kinds} \\ P_{o(a)} y & \text{for time} \end{cases}$$

Principle of Sufficient Reason (LEIBNIZ) $\begin{cases} \text{differentia } \overline{a} \\ \text{force } o(a) \end{cases}$

bias for $P_a x \approx$ a domain minimisation assumption

A causal ontology based on attributes

Trade Galois connection

$$D \subseteq \text{EXTENT}(A) \iff A \subseteq \text{INTENT}(D)$$

for an ontology based on S-change $\quad (a \in In)$

$$(P_a y \wedge ySx) \supset (P_a x \vee yR_a x) \qquad yR_a x := \begin{cases} P_{\overline{a}} x & \text{for kinds} \\ P_{o(a)} y & \text{for time} \end{cases}$$

Principle of Sufficient Reason (Leibniz) $\begin{cases} \text{differentia } \overline{a} \in In \\ \text{force } o(a) \notin In \end{cases}$

bias for $P_a x \approx$ a domain minimisation assumption

$$\frac{\text{individual}}{\text{kind}} \approx \frac{\text{instant}}{\text{interval}} \approx \frac{\text{stative}}{\text{eventive}} \approx \frac{\text{persistent}}{\text{altering}} \approx \frac{\forall \text{ (homogeneous)}}{\exists \text{ (ontological)}}$$

Reducts

Given a set $\mathcal{A}$ of attributes and $A \subseteq \mathcal{A}$,
A-reduct of $\langle \{1,\ldots,n\}, S_n, [\![P_a]\!]_{a\in\mathcal{A}} \rangle$ is $\langle \{1,\ldots,n\}, S_n, [\![P_a]\!]_{a\in A} \rangle$

$$\rho_A(A_1 \cdots A_n) \ := \ (A_1 \cap A) \cdots (A_n \cap A) \qquad \text{“see only } A\text{”}$$

$$\rho_{\{a,\overline{a}\}}(\boxed{a,b}\boxed{a}\boxed{\overline{a},c}) \ = \ \boxed{a}\boxed{a}\boxed{\overline{a}}$$

Reducts & compression

Given a set $\mathcal{A}$ of attributes and $A \subseteq \mathcal{A}$,
A-reduct of $\langle \{1, \ldots, n\}, S_n, [\![P_a]\!]_{a \in \mathcal{A}} \rangle$ is $\langle \{1, \ldots, n\}, S_n, [\![P_a]\!]_{a \in A} \rangle$

$$\rho_A(A_1 \cdots A_n) \;:=\; (A_1 \cap A) \cdots (A_n \cap A) \qquad \text{“see only } A\text{”}$$

$$\rho_{\{a,\overline{a}\}}(\boxed{a, b}\boxed{a}\boxed{\overline{a}, c}) \;=\; \boxed{a}\boxed{a}\boxed{\overline{a}}$$

$$bc(\rho_{\{a,\overline{a}\}}(\boxed{a, b}\boxed{a}\boxed{\overline{a}, c})) \;=\; \boxed{a}\boxed{\overline{a}}$$

Compress $A_1 \cdots A_n$ to eliminate stutters $A_i A_{i+1}$ with $A_i = A_{i+1}$

$$bc(A_1 \cdots A_n) \;:=\; \begin{cases} A_1 & \text{if } n = 1 \\ bc(A_2 \cdots A_n) & \text{else if } A_1 = A_2 \\ A_1 \; bc(A_2 \cdots A_n) & \text{otherwise} \end{cases}$$

Reducts & compression

Given a set $\mathcal{A}$ of attributes and $A \subseteq \mathcal{A}$,

A-reduct of $\langle \{1, \ldots, n\}, S_n, [\![P_a]\!]_{a \in \mathcal{A}} \rangle$ is $\langle \{1, \ldots, n\}, S_n, [\![P_a]\!]_{a \in A} \rangle$

$$\rho_A(A_1 \cdots A_n) \ := \ (A_1 \cap A) \cdots (A_n \cap A) \qquad \text{“see only } A\text{”}$$

$$\rho_{\{a, \bar{a}\}}(\boxed{a, b}\boxed{a}\boxed{\bar{a}, c}) \ = \ \boxed{a}\boxed{a}\boxed{\bar{a}}$$

$$\mathit{bc}(\rho_{\{a, \bar{a}\}}(\boxed{a, b}\boxed{a}\boxed{\bar{a}, c})) \ = \ \boxed{a}\boxed{\bar{a}}$$

Compress $A_1 \cdots A_n$ to eliminate stutters $A_i A_{i+1}$ with $A_i = A_{i+1}$

$$\mathit{bc}(A_1 \cdots A_n) \ := \ \begin{cases} A_1 & \text{if } n = 1 \\ \mathit{bc}(A_2 \cdots A_n) & \text{else if } A_1 = A_2 \\ A_1 \ \mathit{bc}(A_2 \cdots A_n) & \text{otherwise} \end{cases}$$

Base ontology on granularity

$$\mathit{bc}_A(s) \ := \ \mathit{bc}(\rho_A(s))$$

Institutionalisation

An $\mathrm{MSO}_{\mathcal{A}}$-formula φ has finite $voc(\varphi) \subseteq \mathcal{A}$ with all attributes in φ

$$A_1 \cdots A_n \models \varphi \iff \rho_{voc(\varphi)}(A_1 \cdots A_n) \models \varphi$$

Institutionalisation

An $\text{MSO}_{\mathcal{A}}$-formula φ has finite $voc(\varphi) \subseteq \mathcal{A}$ with all attributes in φ

$$A_1 \cdots A_n \models \varphi \iff \rho_{voc(\varphi)}(A_1 \cdots A_n) \models \varphi$$

satisfaction condition (Goguen & Burstall) for an

$$\textit{institution} \begin{cases} \text{signature } A = \text{finite subset of } \mathcal{A} \\ A\text{-model} = \text{string over the alphabet } 2^{A} \\ A\text{-sentence} = \text{MSO}_{A}\text{-sentence} \end{cases}$$

Institutionalisation

An $\mathrm{MSO}_{\mathcal{A}}$-formula φ has finite $voc(\varphi) \subseteq \mathcal{A}$ with all attributes in φ

$$A_1 \cdots A_n \models \varphi \iff \rho_{voc(\varphi)}(A_1 \cdots A_n) \models \varphi$$

satisfaction condition (Goguen & Burstall) for an

$$\textit{institution} \begin{cases} \text{signature } A = \text{finite subset of } \mathcal{A} \\ A\text{-model} = \text{string over the alphabet } 2^A \\ A\text{-sentence} = \mathrm{MSO}_A\text{-sentence} \end{cases}$$

What You See Is All There Is (WYSIATI, Kahneman)

Institutionalisation

An $\mathrm{MSO}_{\mathcal{A}}$-formula φ has finite $voc(\varphi) \subseteq \mathcal{A}$ with all attributes in φ

$$A_1 \cdots A_n \models \varphi \iff \rho_{voc(\varphi)}(A_1 \cdots A_n) \models \varphi$$

satisfaction condition (Goguen & Burstall) for an

$$\textit{institution} \begin{cases} \text{signature } A = \text{finite subset of } \mathcal{A} \\ A\text{-model} = \text{string over the alphabet } 2^A \\ A\text{-sentence} = \mathrm{MSO}_A\text{-sentence} \end{cases}$$

What You See Is All There Is (WYSIATI, Kahneman)

For finite-state transducer T for inheritance,

$$bc_{In}(T(A_1 \cdots A_n)) = bc(T(bc_{In}(A_1 \cdots A_n)))$$

and similarly for inertia.

Institutionalisation

An $\text{MSO}_{\mathcal{A}}$-formula φ has finite $voc(\varphi) \subseteq \mathcal{A}$ with all attributes in φ

$$A_1 \cdots A_n \models \varphi \iff \rho_{voc(\varphi)}(A_1 \cdots A_n) \models \varphi$$

satisfaction condition (Goguen & Burstall) for an

$$\textit{institution} \begin{cases} \text{signature } A = \text{finite subset of } \mathcal{A} \\ A\text{-model} = \text{string over the alphabet } 2^A \\ A\text{-sentence} = \text{MSO}_A\text{-sentence} \end{cases}$$

What You See Is All There Is (WYSIATI, Kahneman)

For finite-state transducer T for inheritance,

$$bc_{In}(T(A_1 \cdots A_n)) = bc(T(bc_{In}(A_1 \cdots A_n)))$$

and similarly for inertia.

Multiple A-models — bound search by reducing A
but additional constraints may expand A and change institution

Conclusion

1. strings/causes in place of instances/extensions

   - strings as MSO-models
   - expect finite automata to be fast

Conclusion

1. strings/causes in place of instances/extensions

   - strings as MSO-models
   - expect finite automata to be fast

2. top-down, contra bottom-up

   - given xSy $\begin{cases} x \text{ is more general than } y \\ x \text{ is before } y \end{cases}$

     draw inference from x to y

   - avoid fixing an extension

Conclusion

1. strings/causes in place of instances/extensions

   - strings as MSO-models
   - expect finite automata to be fast

2. top-down, contra bottom-up

   - given xSy $\begin{cases} x \text{ is more general than } y \\ x \text{ is before } y \end{cases}$

     draw inference from x to y

   - avoid fixing an extension

3. from $\underbrace{\text{known unknowns}}_{\substack{A\text{-models} \\ (\text{signature } A)}}$ to $\underbrace{\text{unknown unknowns}}_{\substack{\text{change of} \\ \text{signature, institution}}}$

Conclusion

1. strings/causes in place of instances/extensions

   - strings as MSO-models
   - expect finite automata to be fast

2. top-down, contra bottom-up

   - given xSy $\begin{cases} x \text{ is more general than } y \\ x \text{ is before } y \end{cases}$

     draw inference from x to y

   - avoid fixing an extension

3. from $\underbrace{\text{known unknowns}}_{\substack{A\text{-models} \\ (\text{signature } A)}}$ to $\underbrace{\text{unknown unknowns}}_{\substack{\text{change of} \\ \text{signature, institution}}}$

THANK YOU