Predication via Finite-State Methods

4/5. Finite-state truthmaking

Tim Fernando
ESSLLI 2017, Toulouse

Key phrases:
- events vs statives vs forces
- *ceteris paribus*: explicit vs derived
- pathfinding
- granularity and institutions
(7) Amundsen flew to the North Pole in May 1926
\[ \exists x \ Amundsen\text{-}flew\text{-}to\text{-}the\text{-}North\text{-}Pole(x) \land \text{In}(\text{May}1926, x) \]

I find entirely persuasive . . . Reichenbach’s proposal that ordinary action sentences have, in effect, an existential quantifier binding the action variable. When we were tempted into thinking a sentence like (7) describes a single event we were misled: it does not describe any event at all. But if (7) is true, then there is an event that makes it true.

This unrecognized element of generality in action sentences is, I think, of the utmost importance in understanding the relation between actions and desires.
Temporal extent: events vs statives

Events: *in* as *within* (Pratt-H 2005, Beaver & Condoravdi 2007)

\[ I \models A \text{ and } I \sqsubseteq I' \implies I' \models A \]

(1) Amundsen flew to the North Pole and swam a mile the same year but not at the same time.

Statives: homogeneous (Taylor 1977, Dowty 1979)

\[ I \models A \text{ and } I' \sqsubseteq I \implies I' \models A \]

(2) Amundsen stayed home in July 1926.
Temporal extent: events vs statives

Events: *in* as *within* (Pratt-H 2005, Beaver & Condoravdi 2007)

\[ I \models A \text{ and } I \subseteq I' \implies I' \models A \]

(1) Amundsen flew to the North Pole and swam a mile the same year but not at the same time.

Statives: homogeneous (Taylor 1977, Dowty 1979)

\[ I \models A \text{ and } I' \subseteq I \implies I' \models A \]

(2) Amundsen stayed home in July 1926.
Temporal extent: events vs statives

Events: *in* as *within* (Pratt-H 2005, Beaver & Condoravdi 2007)

\[ I \models A \text{ and } I \sqsubseteq I' \implies I' \models A \]

(1) Amundsen flew to the North Pole and swam a mile the same year but not at the same time.

Statives: homogeneous (Taylor 1977, Dowty 1979)

\[ I \models A \text{ and } I' \sqsubseteq I \implies I' \models A \]

(2) Amundsen stayed home in July 1926.
### Statives $\neq$ Events $\neq$ Forces

<table>
<thead>
<tr>
<th>stative</th>
<th>event</th>
<th>action/force</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varphi$</td>
<td>$s$</td>
<td>$f$</td>
</tr>
<tr>
<td>label</td>
<td>string</td>
<td>automaton</td>
</tr>
</tbody>
</table>

Dowty’s Aspect hypothesis (1979)

Statives + operators BECOME, DO, CAUSE, ...

\[ \text{BECOME}(\varphi) \leadsto \neg \varphi | \varphi \]
\[ \text{DO}(f) \leadsto \text{ap}(f) | \text{ef}(f) \]

- $\text{ap}(f)$: force $f$ is applied
- $\text{ef}(f)$: a previous application of $f$ is effectual
Statives \neq Events \neq Forces

<table>
<thead>
<tr>
<th>stative</th>
<th>event</th>
<th>action/force</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \varphi )</td>
<td>( s )</td>
<td>( f )</td>
</tr>
<tr>
<td>label</td>
<td>string</td>
<td>automaton</td>
</tr>
</tbody>
</table>

Dowty’s Aspect hypothesis (1979)

statives + operators BECOME, DO, CAUSE, …

\[
\text{BECOME}(\varphi) \sim \begin{array}{c}
\neg \varphi \\
\varphi
\end{array}
\]

\[
\text{DO}(f) \sim \begin{array}{c}
ap(f) \\
ef(f)
\end{array}
\]

\( ap(f) \): force \( f \) is applied

\( ef(f) \): a previous application of \( f \) is effectual
Dowty’s Aspect hypothesis (1979)

statives + operators BECOME, DO, CAUSE, ⋯

\[
\begin{align*}
\text{BECOME}(\varphi) & \sim [\neg \varphi, \varphi] \\
\text{DO}(f) & \sim [ap(f), ef(f)]
\end{align*}
\]

\(ap(f)\) : force \(f\) is applied

\(ef(f)\) : a previous application of \(f\) is effectual
Durativity and culmination

$s$ is *durative* if $\text{length}(s) \geq 3$

<table>
<thead>
<tr>
<th>$-\text{telic}$</th>
<th>$-\text{durative}$</th>
<th>$+\text{durative}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>semelfactive</td>
<td>activity</td>
</tr>
<tr>
<td></td>
<td>$ap(f)$ $ef(f)$</td>
<td>$ap(f)$ $ap(f)$, $ef(f)$ $ef(f)$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$+\text{telic}$</th>
<th>achievement</th>
<th>accomplishment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\neg \varphi$ $\varphi$</td>
<td>$\neg \varphi$, $ap(f)$ $\neg \varphi$, $ap(f)$, $ef(f)$ $\varphi$, $ef(f)$</td>
</tr>
</tbody>
</table>

$s$ is $\varphi$-telic if $s \triangleright \neg \varphi$ $\triangleright \varphi$

$\alpha_1 \cdots \alpha_n \triangleright \beta_1 \cdots \beta_m \iff n = m$ and $\beta_i \subseteq \alpha_i$ for $1 \leq i \leq n$

$s \triangleright L \iff (\exists s' \in L) s \triangleright s'$
Durativity and culmination

$s$ is *durative* if $\text{length}(s) \geq 3$

<table>
<thead>
<tr>
<th>$\neg$telic</th>
<th>$\neg$durative</th>
<th>$+$durative</th>
</tr>
</thead>
<tbody>
<tr>
<td>semelfactive</td>
<td>$\text{ap}(f)$</td>
<td>$\text{ef}(f)$</td>
</tr>
<tr>
<td>$\text{ap}(f)$</td>
<td>$\text{ef}(f)$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$+$telic</th>
<th>achievement</th>
<th>$\neg$telic</th>
<th>accomplishment</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\neg \varphi$</td>
<td>$\varphi$</td>
<td>$\neg \varphi$</td>
<td>$\neg \varphi$, $\text{ap}(f)$, $\text{ef}(f)$</td>
</tr>
</tbody>
</table>

$s$ is $\varphi$-*telic* if $s \supseteq \neg \varphi$ $\supseteq \varphi$

$\alpha_1 \cdots \alpha_n \supseteq \beta_1 \cdots \beta_m$ $\iff$ $n = m$ and $\beta_i \subseteq \alpha_i$ for $1 \leq i \leq n$

$s \supseteq L$ $\iff$ $(\exists s' \in L) \; s \supseteq s'$
Inertia and negation

**Inertia**: a stative persists unless something happens to it

\[ f(\varphi) = ap(f) \text{ where } ef(f) = \varphi \text{ subject to} \]

\[
\begin{array}{c|c}
\neg \varphi & \varphi \\
\hline
\varphi & \neg \varphi
\end{array} \Rightarrow \begin{array}{c} f(\varphi) \\
f(-\varphi)\end{array}
\]

\[ L \Rightarrow L' := \{ s \mid (\forall s' \in \text{factor}(s)) \ s' \triangleright L \text{ implies } s' \triangleright L' \} \]

\[ \text{factor}(s) := \{ s' \mid s = us'v \text{ for some strings } u, v \} \]

\[
\begin{array}{c}
\varphi, \neg \varphi \\
\hline
\Rightarrow \emptyset
\end{array}
\]

\[
\begin{array}{c}
\varphi \\
\hline
\Rightarrow \Box(\varphi + \neg \varphi)
\end{array}
\]

\[
\begin{array}{c}
\varphi \\
\hline
\Rightarrow (\varphi + \neg \varphi)
\end{array}
\]
Inertia: a stative persists unless something happens to it

\[ f(\varphi) = ap(f) \text{ where } ef(f) = \varphi \] subject to

\[
\begin{align*}
\neg \varphi & \quad \varphi & \Rightarrow & \quad f(\varphi) \\
\varphi & \quad \neg \varphi & \Rightarrow & \quad f(\neg \varphi)
\end{align*}
\]

\[
L \Rightarrow L' := \{ s \mid (\forall s' \in \text{factor}(s)) s' \triangleright L \text{ implies } s' \triangleright L' \}
\]

\[
\text{factor}(s) := \{ s' \mid s = us'v \text{ for some strings } u, v \}
\]

\[
\begin{align*}
\varphi, \neg \varphi & \Rightarrow \emptyset \\
\varphi & \Rightarrow (\varphi + \neg \varphi) \\
\varphi & \Rightarrow (\varphi + \neg \varphi)
\end{align*}
\]
Inertia and negation

**Inertia**: a stative persists unless something happens to it.

\[ f(\varphi) = ap(f) \text{ where } ef(f) = \varphi \text{ subject to} \]

\[
\begin{array}{cc}
\neg \varphi & \varphi \\
\varphi & \neg \varphi
\end{array}
\Rightarrow
\begin{array}{c}
f(\varphi) \\
f(\neg \varphi)
\end{array}
\]

\[
L \Rightarrow L' := \{ s \mid (\forall s' \in \text{factor}(s)) s' \not\sqsupset L \text{ implies } s' \not\sqsupset L' \}
\]

\[
\text{factor}(s) := \{ s' \mid s = us'v \text{ for some strings } u, v \}
\]

\[
\begin{array}{c}
\varphi, \neg \varphi \\
\varphi
\end{array}
\Rightarrow
\begin{array}{c}
\emptyset \\
(\varphi + \neg \varphi)
\end{array}
\]
Inertia and negation

**Inertia:** a stative persists unless something happens to it

\[
f(\varphi) = ap(f) \text{ where } ef(f) = \varphi \text{ subject to }
\]

\[
\begin{array}{c|c}
\varphi & \varphi \\
\hline
\neg \varphi & \varphi \\
\hline
\varphi & \neg \varphi \\
\end{array}
\Rightarrow
\begin{array}{c|c}
f(\varphi) & f(\varphi) \\
\hline
f(\neg \varphi) & f(\neg \varphi) \\
\end{array}
\]

\[
L \Rightarrow L' := \{ s \mid (\forall s' \in \text{factor}(s)) s' \supset L \text{ implies } s' \supset L' \}
\]

\[
\text{factor}(s) := \{ s' \mid s = us'v \text{ for some strings } u, v \}
\]

\[
\begin{array}{c|c}
\varphi, \neg \varphi & \emptyset \\
\hline
\varphi & (\varphi + \neg \varphi) \\
\hline
\varphi & (\varphi + \neg \varphi) \\
\end{array}
\]
This principle of “inertia” in the interpretation of statives in discourse applies to many kinds of statives but of course not to all of them . . . there must be a graded hierarchy of the likelihood that various statives will have this kind of implicature, depending on the nature of the state, the agent, and our knowledge of which states are long-lasting and which decay or reappear rapidly. Clearly, an enormous amount of real-world knowledge and expectation must be built into any system which mimics the understanding that humans bring to the temporal interpretations of statives in discourse, so no simple non-pragmatic theory of discourse interpretation is going to handle them very effectively.

Inertia and force constraints above are non-defeasible.

Left open: forces at play and which win out ...
This principle of “inertia” in the interpretation of statives in discourse applies to many kinds of statives but of course not to all of them . . . there must be a graded hierarchy of the likelihood that various statives will have this kind of implicature, depending on the nature of the state, the agent, and our knowledge of which states are long-lasting and which decay or reappear rapidly. Clearly, an enormous amount of real-world knowledge and expectation must be built into any system which mimics the understanding that humans bring to the temporal interpretations of statives in discourse, so no simple non-pragmatic theory of discourse interpretation is going to handle them very effectively.

Inertia and force constraints above are non-defeasible.

Left open: forces at play and which win out ...
This principle of “inertia” in the interpretation of statives in discourse applies to many kinds of statives but of course not to all of them . . . there must be a graded hierarchy of the likelihood that various statives will have this kind of implicature, depending on the nature of the state, the agent, and our knowledge of which states are long-lasting and which decay or reappear rapidly. Clearly, an enormous amount of real-world knowledge and expectation must be built into any system which mimics the understanding that humans bring to the temporal interpretations of statives in discourse, so no simple non-pragmatic theory of discourse interpretation is going to handle them very effectively.

Inertia and force constraints above are non-defeasible.

Left open: forces at play and which win out . . . PRAGMATICS
Formal concept analysis (Galois connection)
given $\models D \times A$, 
a concept is a pair $(E, I) \in 2^D \times 2^A$ s.t.

$$
E = \{ d \in D \mid (\forall a \in I) d \models a \} \quad \text{(the extent of } I) 
$$

$$
I = \{ a \in A \mid (\forall d \in E) d \models a \} \quad \text{(the intent of } E) 
$$

E.g. object $d$ describes the concept $(d_\models, d_\models^\models)$ where

$$
d_\models^\models = \{ a \in A \mid d \models a \} \quad \text{(the intent of } \{d\}) 
$$

$$
d_\models = \{ d' \in D \mid (\forall a \in d_\models^\models) d' \models a \} \quad \text{(the extent of } d_\models^\models) 
$$

$$
d \ IS-\!A\models d' : \iff \ d \in d_\models' 
$$

$$
\iff \ d_\models \subseteq d_\models' \iff \ d_\models ^\models \subseteq d_\models^\models \quad \text{(member of type)} 
$$

$$
\iff \ d_\models' \subseteq d_\models \iff \ d_\models ^\models \subseteq d_\models^\models \quad \text{(subtype)} 
$$

for concepts $(E, I)$ and $(E', I')$, $E \subseteq E' \iff I' \subseteq I$
Back to predication

Formal concept analysis (Galois connection)

given \( D \times A \),
a concept is a pair \( (E, I) \in 2^D \times 2^A \) s.t.

\[
E = \{ d \in D \mid (\forall a \in I) \ d \models a \} \quad \text{(the extent of } I) \\
I = \{ a \in A \mid (\forall d \in E) \ d \models a \} \quad \text{(the intent of } E) 
\]

E.g. object \( d \) describes the concept \( (d_\models, d_{\models}) \) where

\[
d_{\models} = \{ a \in A \mid d \models a \} \quad \text{(the intent of } \{d\}) \\
d_\models = \{ d' \in D \mid (\forall a \in d_{\models})d' \models a \} \quad \text{(the extent of } d_{\models})
\]

\[
d \text{ IS- } A \models d' : \iff d \in d_\models \quad \text{(member of type)} \\
\iff d_\models \subseteq d'_\models \iff d'_{\models} \subseteq d_{\models} \quad \text{(subtype)}
\]

for concepts \( (E, I) \) and \( (E', I') \), \( E \subseteq E' \iff I' \subseteq I \)
Back to predication

**Formal concept analysis** (Galois connection)

given $\models D \times A$,
a *concept* is a pair $(E, I) \in 2^D \times 2^A$ s.t.

\[
E = \{d \in D \mid (\forall a \in I) \, d \models a\} \quad \text{(the extent of } I\text{)}
\]

\[
I = \{a \in A \mid (\forall d \in E) \, d \models a\} \quad \text{(the intent of } E\text{)}
\]

E.g. object $d$ describes the concept $(d\models, d\models)$ where

\[
d\models = \{a \in A \mid d \models a\} \quad \text{(the intent of } \{d\}\text{)}
\]

\[
d\models = \{d' \in D \mid (\forall a \in d\models) d' \models a\} \quad \text{(the extent of } d\models\text{)}
\]

\[
d \text{ IS-A}\models d' : \iff d \in d'\models \quad \text{(member of type)}
\]

\[
\iff d\models \subseteq d'\models \iff d'\models \subseteq d\models \quad \text{(subtype)}
\]

for concepts $(E, I)$ and $(E', I')$, $E \subseteq E' \iff I' \subseteq I$
Reconstruing IS-A: defeasible inheritance

birds fly  Tweety IS-A bird

\[ \text{Tweety flies} \]

Complications

1. defeasibility: penguins are birds that don’t fly
   negations: not opposed (Reiter: normal default rule)

2. other sorts of predication: kind-level, stage-level
   other sorts: \( \Psi \subseteq \Phi \) (widespread?)

3. dependence on choice of \( \models \)
   form \( \Phi[d] \) “institutionally”

\[
\psi \in \Phi[d'] \quad d \text{ IS-A } d' \quad \neg \psi \not\in \Phi[d] \quad \psi \in \psi \quad (\dagger)
\]

Problem: how do we ensure \( \{\psi, \neg \psi\} \not\subseteq \Phi[d] \) and implement (\dagger)?
Reconstruing IS-A: defeasible inheritance

\[
\begin{align*}
\text{birds fly} & \quad \text{Tweety IS-A } \equiv \text{ bird} \\
\hline 
\text{Tweety flies} 
\end{align*}
\]

Complications

1. defeasibility: penguins are birds that don’t fly
   negations: not opposed (Reiter: normal default rule)

2. other sorts of predication: kind-level, stage-level
   other sorts: \( \Psi \subseteq \Phi \) (widespread?)

3. dependence on choice of \( \models \)
   form \( \Phi[d] \) “institutionally”

\[
\begin{align*}
\psi \in \Phi[d'] \\
\text{d IS-A} \\
\text{d'} \\
\neg \psi \not\in \Phi[d'] \\
\psi \in \Phi[d] \\
\psi \in \Psi \quad (\dagger)
\end{align*}
\]

Problem: how do we ensure \( \{\psi, \neg \psi\} \not\subseteq \Phi[d] \) and implement (\( \dagger \))?
Reconstruing IS-A: defeasible inheritance

\[
\begin{align*}
\text{birds fly} & \quad \text{Tweety IS-A bird} \\
\hline
\text{Tweety flies}
\end{align*}
\]

Complications

1. defeasibility: penguins are birds that don’t fly
   negations: not opposed (Reiter: normal default rule)

2. other sorts of predication: kind-level, stage-level
   other sorts: \(\Psi \subseteq \Phi\) (widespread?)

3. dependence on choice of \(\models\)
   form \(\Phi[d]\) “institutionally”

\[
\frac{\psi \in \Phi[d']} \quad d \text{ IS-A } d' \quad \neg \psi \notin \Phi[d] \quad \psi \in \Psi}{\psi \in \Phi[d]} \quad (†)
\]

Problem: how do we ensure \(\{\psi, \neg \psi\} \not\subseteq \Phi[d]\) and implement (†)?
Reconstruing IS-A: defeasible inheritance

birds fly \quad Tweety IS-A \models bird

\hline
Tweety flies

Complications

1. defeasibility: penguins are birds that don’t fly
   negations: not opposed (Reiter: normal default rule)

2. other sorts of predication: kind-level, stage-level
   other sorts: $\Psi \subseteq \Phi$ (widespread?)

3. dependence on choice of $\models$
   form $\Phi[d]$ “institutionally”

$$
\begin{array}{c}
\psi \in \Phi[d'] \\
\quad d \text{ IS-A } d' \\
\quad \neg \psi \not\in \Phi[d] \\
\hline
\psi \in \Psi \\
\end{array}
$$

Problem: how do we ensure $\{\psi, \neg \psi\} \not\subseteq \Phi[d]$ and implement $(†)$?
Reconstruing IS-A: defeasible inheritance

\[
\begin{array}{c}
\text{birds fly} \\
\text{Tweety IS-A bird} \\
\hline
\text{Tweety flies}
\end{array}
\]

Complications

1. defeasibility: penguins are birds that don’t fly
   negations: not opposed (Reiter: normal default rule)

2. other sorts of predication: kind-level, stage-level
   other sorts: \( \Psi \subseteq \Phi \) (widespread?)

3. dependence on choice of \( \models \)
   form \( \Phi[d] \) “institutionally”

\[
\frac{\psi \in \Phi[d'] \quad d \text{ IS-A } d' \quad \neg \psi \not\in \Phi[d]}{\psi \in \Phi[d]} \quad \psi \in \Psi
\] (†)

Problem: how do we ensure \( \{\psi, \neg \psi\} \not\subseteq \Phi[d] \) and implement (†)?
A solution based on strings

**Step 1.** Distinguish a subset $\Phi_\circ[d]$ of predicates that hold *explicitly* in $d$

$$
\varphi \in \Phi_\circ[d] \quad \psi \in \Phi[d'] \quad d \text{ IS-A } d' \quad \neg \psi \not\in \Phi_\circ[d] \quad \psi \in \Psi
$$

forbidding $\{\psi, \neg \psi\} \not\subseteq \Phi_\circ[d]$ for all $\psi$ and $d$

**Step 2.** Restrict IS-A to $n$ distinct $d_1, \ldots, d_n \in D$ for a string

$$
\Phi_\circ[d_1] \Phi_\circ[d_2] \cdots \Phi_\circ[d_n]
$$

with $d_i$ IS-A $d_{i+1}$

so that $\Phi[d]$ is $\Phi_\circ[d]$ and any $\psi \in \Psi$ s.t. for some $i \leq n$,

$$
\psi \in \Phi_\circ[d_i] \text{ and some IS-A-path links } d_i \text{ to } d \text{ avoiding } \neg \psi.
$$

N.B. For fixed $\Phi[\cdot]$, $\Phi_\circ[\cdot]$ varies with IS-A

IS-A as PREV

$$
\begin{align*}
\varphi & \Rightarrow (\varphi + \neg \varphi) \\
\neg \varphi & \Rightarrow (\varphi + \neg \varphi)
\end{align*}
$$

for stages but not kinds
A solution based on strings

**Step 1.** Distinguish a subset $\Phi_\circ[d]$ of predicates that hold explicitly in $d$

$\varphi \in \Phi_\circ[d] \quad \psi \in \Phi[d'] \quad d \text{ IS-A } d' \quad \neg \psi \notin \Phi_\circ[d] \quad \psi \in \Psi$

$\varphi \in \Phi[d] \quad \psi \in \Phi[d]$ forbidding $\{\psi, \neg \psi\} \not\subseteq \Phi_\circ[d]$ for all $\psi$ and $d$

**Step 2.** Restrict IS-A to $n$ distinct $d_1, \ldots, d_n \in D$ for a string

$\Phi_\circ[d_1] \Phi_\circ[d_2] \cdots \Phi_\circ[d_n]$ with $d_i \text{ IS-A } d_{i+1}$

so that $\Phi[d]$ is $\Phi_\circ[d]$ and any $\psi \in \Psi$ s.t. for some $i \leq n$,

$\psi \in \Phi_\circ[d_i]$ and some IS-A-path links $d_i$ to $d$ avoiding $\neg \psi$.

N.B. For fixed $\Phi[\cdot]$, $\Phi_\circ[\cdot]$ varies with IS-A

**IS-A as PREV**

$\varphi \Rightarrow \left(\varphi + \neg \varphi\right)$

$\neg \varphi \Rightarrow \left(\varphi + \neg \varphi\right)$ for stages but not kinds
A solution based on strings

Step 1. Distinguish a subset $\Phi_\circ[d]$ of predicates that hold explicitly in $d$

$$
\varphi \in \Phi_\circ[d] \quad \psi \in \Phi[d'] \quad d \bowtie d' \quad -\psi \notin \Phi_\circ[d] \quad \psi \in \Psi
$$

forbidding $\{\psi, -\psi\} \not\subseteq \Phi_\circ[d]$ for all $\psi$ and $d$

Step 2. Restrict IS-A to $n$ distinct $d_1, \ldots, d_n \in D$ for a string

$$
\Phi_\circ[d_1] \Phi_\circ[d_2] \cdots \Phi_\circ[d_n]
$$

so that $\Phi[d]$ is $\Phi_\circ[d]$ and any $\psi \in \Psi$ s.t. for some $i \leq n$,

$$
\psi \in \Phi_\circ[d_i] \text{ and some IS-A-path links } d_i \text{ to } d \text{ avoiding } -\psi.
$$

N.B. For fixed $\Phi[\cdot], \Phi_\circ[\cdot]$ varies with IS-A

IS-A as PREV

$$
\begin{align*}
\varphi & \Rightarrow (\varphi + -\varphi) \\
\varphi & \Rightarrow (\varphi + -\varphi)
\end{align*}
$$

for stages but not kinds
A solution based on strings

Step 1. Distinguish a subset $\Phi_\circ[d]$ of predicates that hold explicitly in $d$

$\varphi \in \Phi_\circ[d] \quad \psi \in \Phi[d'] \quad d$ IS-A $d' 
\varphi \in \Phi[d] \quad \psi \notin \Phi_\circ[d] 
\psi \in \Psi$

forbidding $\{\psi, \neg\psi\} \not\subseteq \Phi_\circ[d]$ for all $\psi$ and $d$

Step 2. Restrict IS-A to $n$ distinct $d_1, \ldots, d_n \in D$ for a string

$\Phi_\circ[d_1] \Phi_\circ[d_2] \cdots \Phi_\circ[d_n]$ with $d_i$ IS-A $d_{i+1}$

so that $\Phi[d]$ is $\Phi_\circ[d]$ and any $\psi \in \Psi$ s.t. for some $i \leq n$,

$\psi \in \Phi_\circ[d_i]$ and some IS-A-path links $d_i$ to $d$ avoiding $\neg\psi$.

N.B. For fixed $\Phi[\cdot]$, $\Phi_\circ[\cdot]$ varies with IS-A

IS-A as PREV

$\begin{align*}
\varphi \iff (\varphi + \neg\varphi) \\
\neg\varphi \iff (\varphi + \neg\varphi)
\end{align*}$

for stages but not kinds

<table>
<thead>
<tr>
<th>+univ</th>
<th>kind</th>
<th>−subs</th>
<th>attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>+subs</td>
<td>↑</td>
<td>−subs</td>
<td>↑</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>−univ</th>
<th>object</th>
<th>−subs</th>
<th>trope</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑</td>
<td>↓</td>
<td>↑</td>
<td>↓</td>
</tr>
</tbody>
</table>

$\uparrow$ instantiation $\rightsquigarrow$ characterisation

$\models, \Phi[d] \models \Sigma$

$\models \Sigma$ graded: relativize to $\Sigma$

Sub-atomic semantics (T. Parsons)

Institutions $(\text{Mod}_\Sigma, \text{sen}_\Sigma, \models_\Sigma)_\Sigma$

$M'_\sigma \models_\Sigma \varphi \iff M' \models_\Sigma \varphi^\sigma$ given $\Sigma \xrightarrow{\sigma} \Sigma'$

- abstract model theory (Barwise)

<table>
<thead>
<tr>
<th>+univ</th>
<th>kind</th>
<th>( \leadsto )</th>
<th>attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>+subs</td>
<td>↑</td>
<td></td>
<td></td>
</tr>
<tr>
<td>−univ</td>
<td>object</td>
<td>( \leadsto )</td>
<td>trope</td>
</tr>
<tr>
<td></td>
<td>↑</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ \uparrow \text{ instantiation} \quad \downarrow \text{ characterisation} \]

\[ \models, \Phi[d] \quad \models \Sigma \quad \models \Sigma \]

+subs: node, \( d \)  semantics, contra syntax

\( \pm \text{univ} \): graded: relativize to \( \Sigma \)

Sub-atomic semantics (T. Parsons)

\[ \text{Institutions } (\text{Mod}_\Sigma, \text{sen}_\Sigma, \models_\Sigma)_\Sigma \]

\[ M'_\sigma \models_\Sigma \varphi \iff M' \models_\Sigma \varphi^\sigma \text{ given } \Sigma \xrightarrow{\sigma} \Sigma' \]

- abstract model theory (Barwise)
E.J. Lowe and Goguen & Burstall


<table>
<thead>
<tr>
<th>+univ</th>
<th>+subs</th>
<th>−subs</th>
</tr>
</thead>
<tbody>
<tr>
<td>kind</td>
<td></td>
<td>attribute</td>
</tr>
<tr>
<td>↑</td>
<td></td>
<td>↑</td>
</tr>
<tr>
<td>−univ</td>
<td>object</td>
<td>trope</td>
</tr>
</tbody>
</table>

↑ instantiation IS-A, PREV (Mod_Σ, sen_Σ)

∼ characterisation |=, Φ[d] |=Σ

+subs node, d semantics, contra syntax

±univ graded: relativize to Σ

Sub-atomic semantics (T. Parsons)

Institutions (Mod_Σ, sen_Σ, |=Σ)_Σ

M'_σ |=Σ φ ⇔ M' |=Σ φ^σ given Σ → σ Σ'

- abstract model theory (Barwise)

| +univ | kind | ↓ | attribute |
| | | | |
| +subs | | | |
| −subs | | | |
| −univ | object | ↓ | trope |

| instantiation | IS-A, PREV | (Mod_Σ, sen_Σ) |
| characterisation | =, Φ[d] | =_Σ |
| +subs | node, d | semantics, contra syntax |
| ±univ | graded: relativize to Σ |

Sub-atomic semantics (T. Parsons)

Institutions (Mod_Σ, sen_Σ, =_Σ)_Σ

\[ M' |σ|=_Σ ϕ ⇔ M' |σ|=_Σ ϕ^σ \]

given \[ Σ \xrightarrow{σ} Σ' \]

- abstract model theory (Barwise)

<table>
<thead>
<tr>
<th>+univ</th>
<th>kind</th>
<th>−subs</th>
<th>attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>+univ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>−univ</td>
<td>object</td>
<td></td>
<td>trope</td>
</tr>
</tbody>
</table>

↑ instantiation
\sim characterisation
\models, \Phi[d] \models \Sigma

+subs
node, d
semantics, contra syntax

±univ
graded: relativize to \Sigma

Sub-atomic semantics (T. Parsons)

Institutions \((\text{Mod}_\Sigma, \text{sen}_\Sigma, \models_\Sigma)\)

\[ M'_\sigma \models_\Sigma \varphi \iff M' \models_\Sigma \varphi^\sigma \]

- abstract model theory (Barwise)

<table>
<thead>
<tr>
<th>+univ</th>
<th>-univ</th>
</tr>
</thead>
<tbody>
<tr>
<td>+subs</td>
<td>−subs</td>
</tr>
<tr>
<td>kind</td>
<td>attribute</td>
</tr>
<tr>
<td>↑</td>
<td>↑</td>
</tr>
<tr>
<td>object</td>
<td>trope</td>
</tr>
</tbody>
</table>

↑ instantiation  IS-A, PREV  \((\text{Mod}_\Sigma, \text{sen}_\Sigma)\)

\(\sim\) characterisation  \(\models, \Phi[d]\)  \(\models\Sigma\)

+subs  node, \(d\)  semantics, contra syntax

±univ  graded: relativize to \(\Sigma\)

Sub-atomic semantics (T. Parsons)

**Institutions** \((\text{Mod}_\Sigma, \text{sen}_\Sigma, \models_\Sigma)\_\Sigma\)

\(M'_\sigma \models_\Sigma \varphi \iff M' \models_\Sigma \varphi^\sigma\)  given \(\Sigma \overset{\sigma}{\rightarrow} \Sigma'\)

- abstract model theory (Barwise)
Loose ends

Stuttering licensed by a set $A$ of punctual fluents

$$bc_A(s) := s \quad \text{if } \text{length}(s) \leq 1$$

$$bc_A(\alpha\alpha's) := \begin{cases} bc_A(\alpha s) & \text{if } \alpha = \alpha' \text{ and } \alpha \cap A = \emptyset \\ \alpha bc_A(\alpha's) & \text{otherwise} \end{cases}$$

<table>
<thead>
<tr>
<th>$\Sigma$</th>
<th>$D$</th>
<th>$\mathcal{L} : D \to 2^{\Sigma^*}$</th>
<th>$\text{FCA}$</th>
<th>$\text{Constraint Sat Prob}$</th>
<th>$\text{institution}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>variables</td>
<td>possible values</td>
<td>incidence extent intent</td>
<td>$\models \subseteq Mod_\Sigma \times sen_\Sigma$</td>
<td>$Mod_\Sigma(\Phi_L)$</td>
<td>$sen_\Sigma(E)$</td>
</tr>
</tbody>
</table>

MANY THANKS!
Stuttering licensed by a set $A$ of *punctual* fluents

$$bc_A(s) := s \quad \text{if } \text{length}(s) \leq 1$$

$$bc_A(\alpha\alpha's) := \begin{cases} 
bc_A(\alpha s) & \text{if } \alpha = \alpha' \text{ and } \alpha \cap A = \emptyset \\
\alpha \ bc_A(\alpha' s) & \text{otherwise}
\end{cases}$$

<table>
<thead>
<tr>
<th>$\Sigma$</th>
<th>$\Delta$</th>
<th>$L : D \rightarrow 2^{\Sigma^*}$</th>
<th>$L_L$</th>
<th>$E_L$</th>
<th>FCA</th>
<th>Constraint Sat Prob</th>
<th>institution</th>
</tr>
</thead>
<tbody>
<tr>
<td>variables</td>
<td>possible values</td>
<td>signature models</td>
<td>$\models_{\Sigma} \subseteq Mod_{\Sigma} \times sen_{\Sigma}$</td>
<td>$Mod_{\Sigma}(\Phi_L)$</td>
<td>$sen_{\Sigma}(E)$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

MANY THANKS!
Stuttering licensed by a set \( A \) of punctual fluents

\[
bc_A(s) := s \quad \text{if } \text{length}(s) \leq 1
\]

\[
b_A(\alpha \alpha') := \begin{cases} 
bc_A(\alpha s) & \text{if } \alpha = \alpha' \text{ and } \alpha \cap A = \emptyset \\
\alpha bc_A(\alpha' s) & \text{otherwise}
\end{cases}
\]

<table>
<thead>
<tr>
<th>( \Sigma )</th>
<th>( D )</th>
<th>( L : D \to 2^{\Sigma^*} )</th>
<th>( L_L )</th>
<th>( E^L )</th>
<th>Constraint Sat Prob</th>
<th>institution</th>
</tr>
</thead>
<tbody>
<tr>
<td>variables</td>
<td>possible values</td>
<td>signature</td>
<td>models</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \models \subseteq Mod_\Sigma \times sens_\Sigma )</td>
<td>( Mod_\Sigma(\Phi_L) )</td>
<td>( sens_\Sigma(E) )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

MANY THANKS!