Predication via Finite-State Methods

1. Introduction: *Less is More*

Tim Fernando
ESSLLI 2017, Toulouse

www.scss.tcd.ie/Tim.Fernando/FSM4SAS

Course themes: finite-state approximability  
bounded granularity

Today: extension/instance/node instantiation vs mechanism/string/label characterization
Predication via Finite-State Methods

1. Introduction: *Less is More*

Tim Fernando
ESSLLI 2017, Toulouse

www.scss.tcd.ie/Tim.Fernando/FSM4SAS

**Course themes:** finite-state approximability
bounded granularity

**Today:** extension/instance/node vs mechanism/string/label
instantiation characterization
1. Introduction: *Less is More*

Tim Fernando
ESSLLI 2017, Toulouse

www.scss.tcd.ie/Tim.Fernando/FSM4SAS

**Course themes:** finite-state approximability
bounded granularity

**Today:** extension/instance/node vs mechanism/string/label
instantiation vs characterization
### Predicates and instances

<table>
<thead>
<tr>
<th></th>
<th>Predicates and instances</th>
<th>Analysis</th>
<th>G. Carlson</th>
</tr>
</thead>
</table>
| **(1)** | Socrates is human  
Socrates is mortal  
*Socrates is widespread  
?Socrates is an omnivore | $S \in h$ | individual |
| **(2)** | Humans are mortal  
Humans are widespread  
Humans are omnivores | $h \subseteq m$  
h $\in w$  
generic | kind |
| **(3)** | Socrates was thirsty  
Socrates was walking his dog  
Socrates was running a mile | temporal | stage |

Semantic networks (Woods), frames (Fillmore, Barsalou/Düsseldorf), records (Cooper), attribute value structures . . . as finite automata (with strings for instances)
## Predicates and instances

<table>
<thead>
<tr>
<th>(1) Socrates is human</th>
<th>analysis</th>
<th>(2) Humans are mortal</th>
<th>analysis</th>
<th>(3) Socrates was thirsty</th>
<th>analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Socrates is mortal</td>
<td>S ∈ h</td>
<td>h ⊆ m</td>
<td>h ∈ w</td>
<td>temporal</td>
<td>stage</td>
</tr>
<tr>
<td>*Socrates is widespread</td>
<td></td>
<td>kind</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>?Socrates is an omnivore</td>
<td></td>
<td>generic</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Semantic networks (Woods), frames (Fillmore, Barsalou/Düsseldorf), records (Cooper), attribute value structures ... as finite automata (with strings for instances)
## Predicates and instances

<table>
<thead>
<tr>
<th>Analysis</th>
<th>G. Carlson</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicates</td>
<td>Individual</td>
</tr>
<tr>
<td>Instances</td>
<td></td>
</tr>
</tbody>
</table>

### (1) Socrates is human
- Socrates is mortal
- Socrates is widespread
- Socrates is an omnivore

<table>
<thead>
<tr>
<th>Analysis</th>
<th>G. Carlson</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicates</td>
<td>Individual</td>
</tr>
<tr>
<td>Instances</td>
<td></td>
</tr>
</tbody>
</table>

### (2) Humans are mortal
- Humans are widespread
- Humans are omnivores

<table>
<thead>
<tr>
<th>Analysis</th>
<th>G. Carlson</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicates</td>
<td>Kind</td>
</tr>
<tr>
<td>Instances</td>
<td></td>
</tr>
</tbody>
</table>

### (3) Socrates was thirsty
- Socrates was walking his dog
- Socrates was running a mile

<table>
<thead>
<tr>
<th>Analysis</th>
<th>G. Carlson</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temporal</td>
<td>Stage</td>
</tr>
<tr>
<td>Instances</td>
<td></td>
</tr>
</tbody>
</table>

Semantic networks (Woods), frames (Fillmore, Barsalou/Düsseldorf), records (Cooper), attribute value structures ... as finite automata (with strings for instances)
### Predicates and instances

<table>
<thead>
<tr>
<th>Analysis</th>
<th>Individual</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Socrates is human&lt;br&gt; Socrates is mortal&lt;br&gt; *Socrates is widespread&lt;br&gt; ?Socrates is an omnivore</td>
<td>$S \in h$</td>
</tr>
<tr>
<td>(2) Humans are mortal&lt;br&gt; Humans are widespread&lt;br&gt; Humans are omnivores</td>
<td>$h \subseteq m$&lt;br&gt; $h \in w$&lt;br&gt; generic</td>
</tr>
<tr>
<td>(3) Socrates was thirsty&lt;br&gt; Socrates was walking his dog&lt;br&gt; Socrates was running a mile</td>
<td>temporal&lt;br&gt; stage</td>
</tr>
</tbody>
</table>

Semantic networks (Woods), frames (Fillmore, Barsalou/Düsseldorf), records (Cooper), attribute value structures ... as finite automata (with strings for instances)
Semantic networks (Woods), frames (Fillmore, Barsalou/Düsseldorf), records (Cooper), attribute value structures \ldots as finite automata (with strings for instances)
## Predicates and instances

<table>
<thead>
<tr>
<th>Analysis</th>
<th>G. Carlson</th>
</tr>
</thead>
<tbody>
<tr>
<td>Socrates is human</td>
<td>S ∈ h</td>
</tr>
<tr>
<td>Socrates is mortal</td>
<td>individual</td>
</tr>
<tr>
<td>*Socrates is widespread</td>
<td></td>
</tr>
<tr>
<td>?Socrates is an omnivore</td>
<td></td>
</tr>
<tr>
<td>Humans are mortal</td>
<td>h ⊆ m</td>
</tr>
<tr>
<td>Humans are widespread</td>
<td>h ∈ w</td>
</tr>
<tr>
<td>Humans are omnivores</td>
<td>generic</td>
</tr>
<tr>
<td>Socrates was thirsty</td>
<td>temporal</td>
</tr>
<tr>
<td>Socrates was walking his dog</td>
<td>stage</td>
</tr>
<tr>
<td>Socrates was running a mile</td>
<td></td>
</tr>
</tbody>
</table>

Semantic networks (Woods), frames (Fillmore, Barsalou/Düsseldorf), records (Cooper), attribute value structures ... as finite automata (with strings for instances)
Semantic networks (Woods), frames (Fillmore, Barsalou/Düsseldorf), records (Cooper), attribute value structures ... as finite automata (with strings for instances)
## Attribute values and strings

<table>
<thead>
<tr>
<th>Name</th>
<th>Pat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nationality</td>
<td>Irish</td>
</tr>
<tr>
<td>DOB</td>
<td></td>
</tr>
<tr>
<td>Day</td>
<td>30</td>
</tr>
<tr>
<td>Month</td>
<td>June</td>
</tr>
<tr>
<td>Year</td>
<td>1990</td>
</tr>
</tbody>
</table>

- Name: Pat
- Nationality: Irish
- DOB: Day 30, Month June, Year 1990

Form strings over the alphabet

\[
\Sigma = \{\text{name, pat, nationality, \ldots, 1990}\} = \text{Attributes} \cup \text{Values}
\]

A domain \( D \) of objects \( d \in D \) described by \( L(d) \subseteq \Sigma^* \)

- a \( D \)-indexed family \( \mathcal{L} : D \rightarrow 2^{\Sigma^*} \) of \( \Sigma \)-languages.
Form strings over the alphabet

\[ \Sigma = \{ \text{name, pat, nationality}, \ldots, 1990 \} \]

\[ = \text{Attributes} \cup \text{Values} \]

A domain \( D \) of objects \( d \in D \) described by \( \mathcal{L}(d) \subseteq \Sigma^* \)

--- a \( D \)-indexed family \( \mathcal{L} : D \rightarrow 2^{\Sigma^*} \) of \( \Sigma \)-languages.
Attribute values and strings

\[
\begin{bmatrix}
\text{name} & \text{pat} \\
\text{nationality} & \text{irish} \\
\text{dob} & \begin{bmatrix}
\text{day} & 30 \\
\text{month} & \text{june} \\
\text{year} & 1990
\end{bmatrix}
\end{bmatrix}
\]

\[
\begin{bmatrix}
\text{name pat,} \\
\text{nationality irish,} \\
\text{dob day 30,} \\
\text{dob month june,} \\
\text{dob year 1990}
\end{bmatrix}
\]

Form strings over the alphabet

\[
\Sigma = \{\text{name, pat, nationality, \ldots, 1990}\} = \text{Attributes} \cup \text{Values}
\]

A domain \(D\) of objects \(d \in D\) described by \(\mathcal{L}(d) \subseteq \Sigma^*\) — a \(D\)-indexed family \(\mathcal{L} : D \rightarrow 2^{\Sigma^*}\) of \(\Sigma\)-languages.
Brzozowski automaton over $\Sigma$

A state is a subset of $\Sigma^*$, and for $L, L' \subseteq \Sigma^*$,

$$L \xrightarrow{a} L' \iff L' = \{ s \in \Sigma^* \mid as \in L \}$$

$L$ is accepting $\iff \epsilon \in L$

Brzozowski $s$-derivative of $L$

$$s^{-1}L := \{ s' \in \Sigma^* \mid ss' \in L \}$$

so that

$$\epsilon^{-1}L = L$$

$$(as)^{-1}L = s^{-1}(a^{-1}L)$$

$$L \xrightarrow{s} L' \iff L' = s^{-1}L$$

$$s \in L \iff \epsilon \in s^{-1}L$$

Myhill-Nerode: $L$ is regular iff $\{ s^{-1}L \mid s \in \Sigma^* \}$ is finite
A state is a subset of $\Sigma^*$, and for $L, L' \subseteq \Sigma^*$,

$$L \overset{a}{\rightarrow} L' \iff L' = \{ s \in \Sigma^* \mid as \in L \}$$

$L$ is accepting $\iff \epsilon \in L$.

**Brzozowski s-derivative of $L$**

$$s^{-1}L := \{ s' \in \Sigma^* \mid ss' \in L \}$$

so that

$$\epsilon^{-1}L = L$$

$$(as)^{-1}L = s^{-1}(a^{-1}L)$$

$$L \overset{s}{\rightarrow} L' \iff L' = s^{-1}L$$

$$s \in L \iff \epsilon \in s^{-1}L$$

**Myhill-Nerode:** $L$ is regular iff $\{ s^{-1}L \mid s \in \Sigma^* \}$ is finite.
Brzozowski automaton over $\Sigma$

A state is a subset of $\Sigma^*$, and for $L, L' \subseteq \Sigma^*$,

$$L \xrightarrow{a} L' \iff L' = \{ s \in \Sigma^* \mid as \in L \}$$

$L$ is accepting $\iff \epsilon \in L$

**Brzozowski $s$-derivative of $L$**

$$s^{-1}L := \{ s' \in \Sigma^* \mid ss' \in L \}$$

so that

$$\epsilon^{-1}L = L$$

$$(as)^{-1}L = s^{-1}(a^{-1}L)$$

$$L \xrightarrow{s} L' \iff L' = s^{-1}L$$

$$s \in L \iff \epsilon \in s^{-1}L$$

---

**Myhill-Nerode:** $L$ is regular $\iff \{ s^{-1}L \mid s \in \Sigma^* \}$ is finite
A state is a subset of $\Sigma^*$, and for $L, L' \subseteq \Sigma^*$,

$L \xrightarrow{a} L' \iff L' = \{s \in \Sigma^* \mid as \in L\}$

$L$ is accepting $\iff \epsilon \in L$

**Brzozowski s-derivative of $L$**

$s^{-1}L := \{s' \in \Sigma^* \mid ss' \in L\}$

so that

$\epsilon^{-1}L = L$

$(as)^{-1}L = s^{-1}(a^{-1}L)$

$L \xrightarrow{s} L' \iff L' = s^{-1}L$

$s \in L \iff \epsilon \in s^{-1}L$

**Myhill-Nerode:** $L$ is regular $\iff \{s^{-1}L \mid s \in \Sigma^*\}$ is finite
Relations, type symbols, extensions and records

$n$-ary relation $R$ with attributes $att_1, \ldots, att_n$

<table>
<thead>
<tr>
<th>$R$</th>
<th>$att_1$</th>
<th>$\cdots$</th>
<th>$att_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d$</td>
<td>$d_1$</td>
<td>$\cdots$</td>
<td>$d_n$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
</tbody>
</table>

For $d_1, \ldots, d_n, R \in \Sigma$, define extension

$$\text{ext}_\mathcal{L}(R(d_1 \cdots d_n)) := \{d \in D \mid \{R, att_1 d_1, \ldots, att_n d_n\} \subseteq \mathcal{L}(d)\}$$

$$\text{ext}_\mathcal{L}(R) := \bigcup_{d_1 \cdots d_n \in D^n} \text{ext}_\mathcal{L}(R(d_1 \cdots d_n))$$

Or add $A_1, \ldots, A_n \in \Sigma$ with $A_i \in \mathcal{L}(d_i)$ whenever $\text{ext}_\mathcal{L}(R(d_1 \cdots d_n)) \neq \emptyset$

$$\text{ext}_\mathcal{L}(R_{A_1 \cdots A_n}) := \{d \in D \mid \{R, att_1 A_1, \ldots, att_n A_n\} \subseteq \mathcal{L}(d)\}$$

Record type $[att_1 : A_1, \ldots, att_n : A_n]$ with extension

$$\text{ext}_\mathcal{L}([att_1 : A_1, \ldots, att_n : A_n]) := \{d \in D \mid \{att_1 A_1, \ldots, att_n A_n\} \subseteq \mathcal{L}(d)\}$$
$n$-ary relation $R$ with attributes $att_1, \ldots, att_n$

<table>
<thead>
<tr>
<th>$R$</th>
<th>$att_1$</th>
<th>$\cdots$</th>
<th>$att_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d$</td>
<td>$d_1$</td>
<td>$\cdots$</td>
<td>$d_n$</td>
</tr>
<tr>
<td>$: $</td>
<td>$: $</td>
<td>$: $</td>
<td>$: $</td>
</tr>
</tbody>
</table>

For $d_1, \ldots, d_n$, $R \in \Sigma$, define extension

$$\text{ext}_L(R(d_1 \cdots d_n)) := \{d \in D \mid \{R, att_1 d_1, \ldots, att_n d_n\} \subseteq \mathcal{L}(d)\}$$

$$\text{ext}_L(R) := \bigcup_{d_1 \cdots d_n \in D^n} \text{ext}_L(R(d_1 \cdots d_n))$$

Or add $A_1, \ldots, A_n \in \Sigma$ with $A_i \in \mathcal{L}(d_i)$ whenever $\text{ext}_L(R(d_1 \cdots d_n)) \neq \emptyset$

$$\text{ext}_L(R_{A_1 \cdots A_n}) := \{d \in D \mid \{R, att_1 A_1, \ldots, att_n A_n\} \subseteq \mathcal{L}(d)\}$$

Record type $[att_1 : A_1, \ldots, att_n : A_n]$ with extension

$$\text{ext}_L([att_1 : A_1, \ldots, att_n : A_n]) := \{d \in D \mid \{att_1 A_1, \ldots, att_n A_n\} \subseteq \mathcal{L}(d)\}$$
$n$-ary relation $R$ with attributes $\text{att}_1, \ldots, \text{att}_n$

\[
\begin{array}{|c|c|c|}
\hline
R & \text{att}_1 & \cdots & \text{att}_n \\
\hline
d & d_1 & \cdots & d_n \\
\vdots & \vdots & \ddots & \vdots \\
\hline
\end{array}
\]

For $d_1, \ldots, d_n$, $R \in \Sigma$, define extension

\[
\text{ext}_L(R(d_1 \cdots d_n)) := \{ d \in D \mid \{R, \text{att}_1 d_1, \ldots, \text{att}_n d_n\} \subseteq L(d) \}
\]

\[
\text{ext}_L(R) := \bigcup_{d_1 \cdots d_n \in D^n} \text{ext}_L(R(d_1 \cdots d_n))
\]

Or add $A_1, \ldots, A_n \in \Sigma$ with $A_i \in L(d_i)$ whenever $\text{ext}_L(R(d_1 \cdots d_n)) \neq \emptyset$

\[
\text{ext}_L(R_{A_1 \cdots A_n}) := \{ d \in D \mid \{R, \text{att}_1 A_1, \ldots, \text{att}_n A_n\} \subseteq L(d) \}
\]

Record type $[\text{att}_1 : A_1, \ldots, \text{att}_n : A_n]$ with extension

\[
\text{ext}_L([\text{att}_1 : A_1, \ldots, \text{att}_n : A_n]) := \{ d \in D \mid \{\text{att}_1 A_1, \ldots, \text{att}_n A_n\} \subseteq L(d) \}
\]
Relations, type symbols, extensions and records

\( n \)-ary relation \( R \) with attributes \( att_1, \ldots, att_n \)

\[
\begin{array}{|c|c|c|}
\hline
R & att_1 & \cdots & att_n \\
\hline
d & d_1 & \cdots & d_n \\
\vdots & \vdots & \vdots & \vdots \\
\hline
\end{array}
\]

For \( d_1, \ldots, d_n, R \in \Sigma \), define extension

\[
\text{ext}_L(R(d_1 \cdots d_n)) := \{ d \in D \mid \{R, att_1 d_1, \ldots, att_n d_n\} \subseteq L(d) \}
\]

\[
\text{ext}_L(R) := \bigcup_{d_1 \cdots d_n \in D^n} \text{ext}_L(R(d_1 \cdots d_n))
\]

Or add \( A_1, \ldots, A_n \in \Sigma \) with \( A_i \in L(d_i) \) whenever \( \text{ext}_L(R(d_1 \cdots d_n)) \neq \emptyset \)

\[
\text{ext}_L(R_{A_1 \cdots A_n}) := \{ d \in D \mid \{R, att_1 A_1, \ldots, att_n A_n\} \subseteq L(d) \}
\]

Record type \([att_1 : A_1, \ldots, att_n : A_n]\) with extension

\[
\text{ext}_L([att_1 : A_1, \ldots, att_n : A_n]) := \{ d \in D \mid \{att_1 A_1, \ldots, att_n A_n\} \subseteq L(d) \}\]
Relations, type symbols, extensions and records

\[ n \text{-ary relation } R \text{ with attributes } att_1, \ldots, att_n \]

\[
\begin{array}{|c|c|c|}
\hline
R & att_1 & \cdots & att_n \\
\hline
d & d_1 & \cdots & d_n \\
\vdots & \vdots & \ddots & \vdots \\
\hline
\end{array}
\]

For \( d_1, \ldots, d_n, R \in \Sigma \), define extension

\[
\text{ext}_\mathcal{L}(R(d_1 \cdots d_n)) := \{ d \in D \mid \{ R, att_1 d_1, \ldots, att_n d_n \} \subseteq \mathcal{L}(d) \}
\]

\[
\text{ext}_\mathcal{L}(R) := \bigcup_{d_1 \cdots d_n \in D^n} \text{ext}_\mathcal{L}(R(d_1 \cdots d_n))
\]

Or add \( A_1, \ldots, A_n \in \Sigma \) with \( A_i \in \mathcal{L}(d_i) \) whenever \( \text{ext}_\mathcal{L}(R(d_1 \cdots d_n)) \neq \emptyset \)

\[
\text{ext}_\mathcal{L}(R_{A_1 \cdots A_n}) := \{ d \in D \mid \{ R, att_1 A_1, \ldots, att_n A_n \} \subseteq \mathcal{L}(d) \}
\]

Record type \([ att_1 : A_1, \ldots, att_n : A_n ]\) with extension

\[
\text{ext}_\mathcal{L}([att_1 : A_1, \ldots, att_n : A_n]) := \{ d \in D \mid \{ att_1 A_1, \ldots, att_n A_n \} \subseteq \mathcal{L}(d) \}
\]
Subsets of $D$ from languages over $\Sigma$

A language $L \subseteq \Sigma^*$ describes the $D$-subset

$$L_L := \{ d \in D \mid L \subseteq \mathcal{L}(d) \}$$

E.g.

$$\{ \text{att}_1 \ A_1, \ldots, \text{att}_n \ A_n \} \mathcal{L} = \text{ext}_L([\text{att}_1 : A_1, \ldots, \text{att}_n : A_n])$$

$$\{ \text{R}, \text{att}_1 \ A_1, \ldots, \text{att}_n \ A_n \} \mathcal{L} = \text{ext}_L(\text{R}_{A_1 \cdots A_n})$$

$$\{ \text{R}, \text{att}_1 \ d_1, \ldots, \text{att}_n \ d_n \} \mathcal{L} = \text{ext}_L(\text{R}(d_1 \cdots d_n))$$

Clearly,

$$L \subseteq L' \implies L'_L \subseteq L_L$$

but $\Leftarrow$ may fail because $D$ is not large enough

$$L \subseteq \bigcap_{d \in L_L} \mathcal{L}(d) \subseteq \bigcap_{d \in L'_L} \mathcal{L}(d) \nsubseteq L'$$
Subsets of $D$ from languages over $\Sigma$

A language $L \subseteq \Sigma^*$ describes the $D$-subset

$$L_L := \{ d \in D \mid L \subseteq \mathcal{L}(d) \}$$

E.g.

$$\{ \text{att}_1 A_1, \ldots, \text{att}_n A_n \} \mathcal{L} = \text{ext}_\mathcal{L}(\text{[att}_1 : A_1, \ldots, \text{att}_n : A_n])$$

$$\{ R, \text{att}_1 A_1, \ldots, \text{att}_n A_n \} \mathcal{L} = \text{ext}_\mathcal{L}(R A_1 \cdots A_n)$$

$$\{ R, \text{att}_1 d_1, \ldots, \text{att}_n d_n \} \mathcal{L} = \text{ext}_\mathcal{L}(R(d_1 \cdots d_n))$$

Clearly,

$$L \subseteq L' \implies L' \mathcal{L} \subseteq L \mathcal{L}$$

but $\Leftarrow$ may fail because $D$ is not large enough

$$L \subseteq \bigcap_{d \in L} \mathcal{L}(d) \subseteq \bigcap_{d \in L'} \mathcal{L}(d) \nsubseteq L'$$
Languages from subsets of $D$

A subset $E$ of $D$ is described by the $\Sigma$-language

$$E^L := \bigcap_{d \in E} \mathcal{L}(d)$$

with (antitone) Galois connection

$$L \subseteq E^L \iff E \subseteq L^L$$

**Formal Concept Analysis**: a concept is a pair $(E, L)$ s.t.

$$L = E^L \text{ (intent)} \quad \text{and} \quad E = L^L \text{ (extent)}.$$ 

∴ concepts $(E, L)$ and $(E', L')$ can be partially ordered

$$ E \subseteq E' \iff L' \subseteq L$$

$E$ consists of objects in $D$

$L$ consists of strings in $\Sigma^*$, not just symbols in $\Sigma$.

NEXT: focus on concepts given by objects form strings sharpening partial order on concepts
Languages from subsets of $D$

A subset $E$ of $D$ is described by the $\Sigma$-language

$$E^L := \bigcap_{d \in E} L(d)$$

with (antitone) Galois connection

$$L \subseteq E^L \iff E \subseteq L_L$$

Formal Concept Analysis: a concept is a pair $(E, L)$ s.t.

$$L = E^L \text{ (intent)} \quad \text{and} \quad E = L_L \text{ (extent)}.$$ 

$\therefore$ concepts $(E, L)$ and $(E', L')$ can be partially ordered

$$E \subseteq E' \iff L' \subseteq L$$

$E$ consists of objects in $D$
$L$ consists of strings in $\Sigma^*$, not just symbols in $\Sigma$.

NEXT: focus on concepts given by objects form strings sharpening partial order on concepts
Languages from subsets of $D$

A subset $E$ of $D$ is described by the $\Sigma$-language

$$E^\mathcal{L} := \bigcap_{d \in E} \mathcal{L}(d)$$

with (antitone) Galois connection

$$L \subseteq E^\mathcal{L} \iff E \subseteq L^\mathcal{L}$$

*Formal Concept Analysis*: a concept is a pair $(E, L)$ s.t.

$$L = E^\mathcal{L} \text{ (intent) and } E = L^\mathcal{L} \text{ (extent)}.$$  

∴ concepts $(E, L)$ and $(E', L')$ can be partially ordered

$$E \subseteq E' \iff L' \subseteq L$$

$E$ consists of objects in $D$

$L$ consists of strings in $\Sigma^*$, not just symbols in $\Sigma$.

*NEXT*: focus on concepts given by objects

form strings sharpening partial order on concepts
Concepts from objects in $D$

An object $d \in D$ names the concept $(\mathcal{L}(d)_\mathcal{L}, \mathcal{L}(d))$.

An $\mathcal{L}$-extensional account of IS-A

\[
    d \text{ IS-}_\mathcal{L} d' \iff d \in \mathcal{L}(d')_\mathcal{L} \iff \mathcal{L}(d)_\mathcal{L} \subseteq \mathcal{L}(d')_\mathcal{L} \iff \mathcal{L}(d') \subseteq \mathcal{L}(d)
\]

supporting inheritance along IS-A

\[
    s \in \mathcal{L}(d') \quad \frac{d \text{ IS-}_\mathcal{L} d'}{s \in \mathcal{L}(d)} \quad \text{e.g.} \quad \text{bird flies} \quad \frac{\text{Tweety IS-}_\mathcal{L} \text{ bird}}{\text{Tweety flies}}
\]

Complications

1. defeasibility: penguins are birds that don’t fly
2. other sorts of predication: kind-level, stage-level
3. dependence on choice of $\mathcal{L} : D \rightarrow 2^\Sigma^*$
Concepts from objects in $D$

An object $d \in D$ names the concept $(\mathcal{L}(d), \mathcal{L}(d))$. An $\mathcal{L}$-extensional account of IS-A

$$d \text{ IS-A}_\mathcal{L} d' \iff d \in \mathcal{L}(d')$$

$$\iff \mathcal{L}(d) \subseteq \mathcal{L}(d')$$

supporting inheritance along IS-A

$$s \in \mathcal{L}(d') \quad d \text{ IS-A}_\mathcal{L} d' \quad e.g. \quad \frac{\text{bird flies}}{s \in \mathcal{L}(d)}$$

Complications

1. defeasibility: penguins are birds that don’t fly
2. other sorts of predication: kind-level, stage-level
3. dependence on choice of $\mathcal{L} : D \rightarrow 2^{\Sigma^*}$
An object $d \in D$ names the concept $(\mathcal{L}(d)_\mathcal{L}, \mathcal{L}(d))$.

An $\mathcal{L}$-extensional account of IS-A

$$d \text{ IS-A}_\mathcal{L} d' \iff d \in \mathcal{L}(d')_\mathcal{L}$$

$$\iff \mathcal{L}(d)_\mathcal{L} \subseteq \mathcal{L}(d')_\mathcal{L} \iff \mathcal{L}(d') \subseteq \mathcal{L}(d)$$

supporting inheritance along IS-A

$$s \in \mathcal{L}(d') \quad d \text{ IS-A}_\mathcal{L} d' \quad \frac{s \in \mathcal{L}(d)}{\text{e.g.} \quad \text{bird flies}} \quad \frac{\text{Tweety IS-A}_\mathcal{L} \text{ bird}}{\text{Tweety flies}}$$

Complications

1. defeasibility: penguins are birds that don’t fly
2. other sorts of predication: kind-level, stage-level
3. dependence on choice of $\mathcal{L} : D \rightarrow 2^{\Sigma^*}$
An object \( d \in D \) names the concept \((\mathcal{L}(d)_\mathcal{L}, \mathcal{L}(d))\).

An \( \mathcal{L} \)-extensional account of IS-A

\[
d \text{ IS-}A_{\mathcal{L}} \ d' \iff d \in \mathcal{L}(d')_{\mathcal{L}}
\]

\[
\iff \mathcal{L}(d)_{\mathcal{L}} \subseteq \mathcal{L}(d')_{\mathcal{L}} \iff \mathcal{L}(d') \subseteq \mathcal{L}(d)
\]

supporting inheritance along IS-A

\[
s \in \mathcal{L}(d') \quad d \text{ IS-}A_{\mathcal{L}} \ d' \quad \text{e.g.} \quad \text{bird flies} \quad \text{Tweety IS-}A_{\mathcal{L}} \ 	ext{bird}
\]

\[
s \in \mathcal{L}(d) \quad s \in \mathcal{L}(d') \quad \text{e.g.} \quad \text{Tweety flies}
\]

Complications

1. defeasibility: penguins are birds that don’t fly
2. other sorts of predication: kind-level, stage-level
3. dependence on choice of \( \mathcal{L} : D \to 2^{\Sigma^*} \)
Reconstruing IS-A: defeasibility & strings

Replace $L(d)$ by a set $Φ[d]$ of predicates that hold of $d$

\[
\frac{\psi \in Φ[d']} \quad d \text{ IS-A } d' \quad \overline{\psi} \notin Φ[d] \quad \psi \in Ψ
\]

1. defeasibility: introduce negation ‘not-opposed($ψ, d$)’
2. other sorts: $Ψ \subseteq Φ$
3. dependence on $L$: form $Φ[d]$ *institutionally* (Goguen)

Ensure $\{ψ, \overline{ψ}\} \not\subseteq Φ[d]$ by taking IS-A to be a successor relation on finitely many $d$’s for a string

\[
Φ[d_1] Φ[d_2] \cdots Φ[d_n] \quad \text{with } d_i \text{ IS-A } d_{i+1}.
\]

Similar strings for stage-level predication, with PREV (and inertia) in place of IS-A (and inheritance).

Woods 2007 - *intensional* subsumption (contra IS-A $L$)
- more on Thursday (Lecture 4)
Reconstruing IS-A: defeasibility & strings

Replace $\mathcal{L}(d)$ by a set $\Phi[d]$ of predicates that hold of $d$

\[
\begin{array}{c}
\psi \in \Phi[d'] \quad \quad \text{d IS-A d'} \quad \psi \notin \Phi[d] \\
\hline
\psi \in \Phi[d]
\end{array}
\]

1. defeasibility: introduce negation ‘not-opposed($\psi$, $d$)’

2. other sorts: $\Psi \subseteq \Phi$

3. dependence on $\mathcal{L}$: form $\Phi[d]$ institutionally (Goguen)

Ensure $\{\psi, \overline{\psi}\} \not\subseteq \Phi[d]$ by taking IS-A to be a successor relation on finitely many $d$’s for a string

\[
\Phi[d_1] \Phi[d_2] \cdots \Phi[d_n] \quad \text{with } d_i \text{ IS-A } d_{i+1}.
\]

Similar strings for stage-level predication, with PREV (and inertia) in place of IS-A (and inheritance).

Woods 2007 - intensional subsumption (contra IS-A$_{\mathcal{L}}$)
- more on Thursday (Lecture 4)
Reconstruing IS-A: defeasibility & strings

Replace $\mathcal{L}(d)$ by a set $\Phi[d]$ of predicates that hold of $d$

$$\begin{align*}
\psi \in \Phi[d'] & \quad d \text{ IS-A } d' & \quad \overline{\psi} \notin \Phi[d] \\
\psi \in \Phi[d] & \\
\end{align*}$$

1. defeasibility: introduce negation ‘not-opposed($\psi, d$)’
2. other sorts: $\Psi \subseteq \Phi$
3. dependence on $\mathcal{L}$: form $\Phi[d]$ institutionally (Goguen)

Ensure $\{\psi, \overline{\psi}\} \not\subseteq \Phi[d]$ by taking IS-A to be a successor relation on finitely many $d$’s for a string

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

Similar strings for stage-level predication, with PREV (and inertia) in place of IS-A (and inheritance).

Woods 2007 - *intensional* subsumption (contra IS-A$_\mathcal{L}$) - more on Thursday (Lecture 4)
Reconstruing IS-A: defeasibility & strings

Replace \( \mathcal{L}(d) \) by a set \( \Phi[d] \) of predicates that hold of \( d \)

\[
\frac{\psi \in \Phi[d'] \quad d \text{ IS-A } d' \quad \overline{\psi} \notin \Phi[d]}{\psi \in \Phi[d]} \quad \psi \in \psi
\]

1. defeasibility: introduce negation ‘not-opposed(\( \psi, d \))’
2. other sorts: \( \Psi \subseteq \Phi \)
3. dependence on \( \mathcal{L} \): form \( \Phi[d] \) *institutionally* (Goguen)

Ensure \( \{\psi, \overline{\psi}\} \not\subseteq \Phi[d] \) by taking IS-A to be a successor relation on finitely many \( d \)'s for a string

\[
\Phi[d_1] \Phi[d_2] \cdots \Phi[d_n] \quad \text{with } d_i \text{ IS-A } d_{i+1}.
\]

Similar strings for stage-level predication, with PREV (and inertia) in place of IS-A (and inheritance).

Woods 2007 - *intensional* subsumption (contra IS-A\( _\mathcal{L} \))
- more on Thursday (Lecture 4)
Replacing $\mathcal{L}(d)$ by a set $\Phi[d]$ of predicates that hold of $d$

\[
\psi \in \Phi[d'] \quad d \text{ IS-A } d' \quad \overline{\psi} \notin \Phi[d] \quad \psi \in \psi
\]

1. defeasibility: introduce negation ‘not-opposed($\psi, d$)’
2. other sorts: $\Psi \subseteq \Phi$
3. dependence on $\mathcal{L}$: form $\Phi[d]$ *institutionally* (Goguen)

Ensure $\{\psi, \overline{\psi}\} \not\subseteq \Phi[d]$ by taking IS-A to be a successor relation on finitely many $d$’s for a string

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

Similar strings for stage-level predication, with PREV (and inertia) in place of IS-A (and inheritance).

Woods 2007 - *intensional* subsumption (contra IS-A$_\mathcal{L}$)
- more on Thursday (Lecture 4)
Reconstruing IS-A: defeasibility & strings

Replace $\mathcal{L}(d)$ by a set $\Phi[d]$ of predicates that hold of $d$

$$
\psi \in \Phi[d'] \quad d \text{ IS-A } d' \quad \overline{\psi} \notin \Phi[d] \quad \psi \in \Phi[d]
$$

1. defeasibility: introduce negation ‘not-opposed($\psi, d$)’
2. other sorts: $\Psi \subseteq \Phi$
3. dependence on $\mathcal{L}$: form $\Phi[d]$ institutionally (Goguen)

Ensure $\{\psi, \overline{\psi}\} \not\subseteq \Phi[d]$ by taking IS-A to be a successor relation on finitely many $d$’s for a string

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

Similar strings for stage-level predication, with PREV (and inertia) in place of IS-A (and inheritance).

Woods 2007 - intensional subsumption (contra IS-A$_\mathcal{L}$) - more on Thursday (Lecture 4)
the study of those ‘formulas of English’ that are treated as atomic formulas in most logical investigations of English

including predication/modification (Davidson 1967)

Jones did it slowly, deliberately, in the bathroom, with a knife, at midnight

and tense & aspect

- Reichenbach relates R to \{ S for tense, E for aspect \}
- Inside E: e.g. (non-)entailments (Aristotle . . . Dowty 1979)

\[
\begin{align*}
\text{Al was running (towards home)} & \quad \therefore \text{Al ran (towards home)} \\
\end{align*}
\]

\[
\begin{align*}
\text{Al was running home} & \quad \neq \text{Al ran home}
\end{align*}
\]
the study of those 'formulas of English' that are treated as atomic formulas in most logical investigations of English

including predication/modification (Davidson 1967)

Jones did it slowly, deliberately, in the bathroom, with a knife, at midnight

and tense & aspect

- Reichenbach relates R to \( \{ \text{S for tense} \cup \text{E for aspect} \) 

- Inside E: e.g. (non-)entailments (Aristotle ... Dowty 1979)

\[
\begin{align*}
\text{Al was running (towards home)} & \quad \text{\therefore Al ran (towards home)} \\
\text{Al was running home} & \quad \neq \text{Al ran home}
\end{align*}
\]
the study of those 'formulas of English' that are treated as atomic formulas in most logical investigations of English

including predication/modification (Davidson 1967)

Jones did it slowly, deliberately, in the bathroom, with a knife, at midnight

and tense & aspect

- Reichenbach relates R to \( \{ S \text{ for tense} \), \( E \text{ for aspect} \)

- Inside E: e.g. (non-)entailments (Aristotle ... Dowty 1979)

\[
\text{Al was running (towards home)} \quad \therefore \quad \text{Al ran (towards home)} \\
\text{Al was running home} \quad \not\therefore \quad \text{Al ran home}
\]
the study of those 'formulas of English' that are treated as atomic formulas in most logical investigations of English

including predication/modification (Davidson 1967)

Jones did it slowly, deliberately, in the bathroom, with a knife, at midnight

and tense & aspect

- Reichenbach relates $R$ to $\begin{cases} S \text{ for tense} \\ E \text{ for aspect} \end{cases}$
- Inside $E$: e.g. (non-)entailments (Aristotle ... Dowty 1979)

\[
\begin{align*}
\text{Al was running (towards home)} & \quad \therefore \text{Al ran (towards home)} \\
\text{Al was running home} & \quad \therefore \text{Al ran home}
\end{align*}
\]
Subatomic semantics (Parsons 1990)

the study of those 'formulas of English' that are treated as atomic formulas in most logical investigations of English

including predication/modification (Davidson 1967)

Jones did it slowly, deliberately, in the bathroom, with a knife, at midnight

and tense & aspect

- Reichenbach relates R to \{ S for tense, E for aspect \}
- Inside E: e.g. (non-)entailments (Aristotle . . . Dowty 1979)

\[
\begin{align*}
\text{Al was running (towards home)} & \quad \therefore \text{Al ran (towards home)} \\
\text{Al was running home} & \quad \therefore \text{Al ran home}
\end{align*}
\]
Subatomic semantics (Parsons 1990)

the study of those 'formulas of English' that are treated as atomic formulas in most logical investigations of English

including predication/modification (Davidson 1967)

Jones did it slowly, deliberately, in the bathroom, with a knife, at midnight

and tense & aspect

  - Reichenbach relates R to \( \{ S \text{ for tense} \) \) \( E \text{ for aspect} \)
  - Inside E: e.g. (non-)entailments (Aristotle . . . Dowty 1979)

\[
\begin{align*}
\text{Al was running (towards home)} & \quad \text{Al was running home} \\
\therefore \text{Al ran (towards home)} & \quad \therefore \text{Al ran home}
\end{align*}
\]
The expression of time in natural languages relates a clause-internal temporal structure to a clause-external temporal structure. The latter may shrink to a single interval, for example, the time at which the sentence is uttered; but this is just a special case. The clause-internal temporal structure may also be very simple – it may be reduced to a single interval without any further differentiation, the 'time of the situation'; but if this ever happens, it is only a borderline case. As a rule, the clause-internal structure is much more complex.
The expression of time in natural languages relates a clause-internal temporal structure to a clause-external temporal structure. The latter may shrink to a single interval, for example, the time at which the sentence is uttered; but this is just a special case. The clause-internal temporal structure may also be very simple – it may be reduced to a single interval without any further differentiation, the 'time of the situation'; but if this ever happens, it is only a borderline case.

As a rule, the clause-internal structure is much more complex.
The expression of time in natural languages relates a clause-internal temporal structure to a clause-external temporal structure. The latter may shrink to a single interval, for example, the time at which the sentence is uttered; but this is just a special case. The clause-internal temporal structure may also be very simple – it may be reduced to a single interval without any further differentiation, the 'time of the situation'; but if this ever happens, it is only a borderline case.

As a rule, the clause-internal structure is much more complex.

Ed exhaled

H. Reichenbach

it rained
it has rained
W. Klein

The expression of time in natural languages relates a clause-internal temporal structure to a clause-external temporal structure. The latter may shrink to a single interval, for example, the time at which the sentence is uttered; but this is just a special case. The clause-internal temporal structure may also be very simple – it may be reduced to a single interval without any further differentiation, the 'time of the situation'; but if this ever happens, it is only a borderline case.

As a rule, the clause-internal structure is much more complex.

Ed exhaled

H. Reichenbach

it rained
it has rained
The expression of time in natural languages relates a clause-internal temporal structure to a clause-external temporal structure. The latter may shrink to a single interval, for example, the time at which the sentence is uttered; but this is just a special case. The clause-internal temporal structure may also be very simple – it may be reduced to a single interval without any further differentiation, the 'time of the situation'; but if this ever happens, it is only a borderline case.

As a rule, the clause-internal structure is much more complex.

Ed exhaled

H. Reichenbach

it rained

it has rained
The expression of time in natural languages relates a clause-internal temporal structure to a clause-external temporal structure. The latter may shrink to a single interval, for example, the time at which the sentence is uttered; but this is just a special case. The clause-internal temporal structure may also be very simple – it may be reduced to a single interval without any further differentiation, the 'time of the situation'; but if this ever happens, it is only a borderline case. As a rule, the clause-internal structure is much more complex.

Ed exhaled

H. Reichenbach

it rained

it has rained
Open-endedness & institutions

- more than one symbol in a box — alphabet as powerset
  \[ \Sigma \sim 2^\Sigma := \{ A \mid A \subseteq \Sigma \} \]

- add symbols, possibly lengthening string
  e.g. real line \( \mathbb{R} \) from finite sets of rational numbers

Formulate \( \Sigma \) as a *signature* in an *institution* (Goguen & Burstall)

\[ M \models \Sigma \varphi \]

model \( M \) given by a functor \( \text{Mod} \)
signature \( \Sigma \) in a category \( \text{Sign} \)
sentence \( \varphi \) given by a functor \( \text{sen} \)

Mark Steedman: temporality is not just about timelines
Greg Carlson: *rules & regulations*

Tess is eating dal \( \neq \) Tess eats dal

From episodes/particulars to generics/types (frames)
models-as-strings \( \sim \) models-as-finite automata
Open-endedness & institutions

- more than one symbol in a box — alphabet as powerset
  \[ \Sigma \sim 2^\Sigma := \{ A \mid A \subseteq \Sigma \} \]

- add symbols, possibly lengthening string
  e.g. real line \( \mathbb{R} \) from finite sets of rational numbers

Formulate \( \Sigma \) as a signature in an institution (Goguen & Burstall)

\[ M \models \Sigma \varphi \]

model \( M \) given by a functor \( \text{Mod} \)
signature \( \Sigma \) in a category \( \text{Sign} \)
sentence \( \varphi \) given by a functor \( \text{sen} \)

Mark Steedman: temporality is not just about timelines
Greg Carlson: rules & regulations

\[ \text{Tess is eating dal} \neq \text{Tess eats dal} \]

From episodes/particulars to generics/types (frames)

models-as-strings \( \sim \) models-as-finite automata
Open-endedness & institutions

- more than one symbol in a box — alphabet as powerset
  \[ \Sigma \leadsto 2^\Sigma := \{ A \mid A \subseteq \Sigma \} \]
- add symbols, possibly lengthening string
  e.g. real line \( \mathbb{R} \) from finite sets of rational numbers

Formulate \( \Sigma \) as a signature in an institution (Goguen & Burstall)

\[ M \models \Sigma \varphi \]

- model \( M \) given by a functor \( \text{Mod} \)
- signature \( \Sigma \) in a category \( \text{Sign} \)
- sentence \( \varphi \) given by a functor \( \text{sen} \)

Mark Steedman: temporality is not just about timelines
Greg Carlson: rules & regulations

Tess is eating dal \( \neq \) Tess eats dal

From episodes/particulars to generics/types (frames)

models-as-strings \( \leadsto \) models-as-finite automata
Open-endedness & institutions

- more than one symbol in a box — alphabet as powerset
  \[ \Sigma \rightsquigarrow 2^\Sigma := \{ A \mid A \subseteq \Sigma \} \]

- add symbols, possibly lengthening string
  e.g. real line \( \mathbb{R} \) from finite sets of rational numbers

Formulate \( \Sigma \) as a *signature* in an *institution* (Goguen & Burstall)

\[
M \models_\Sigma \varphi \quad \left\{ \begin{array}{l}
\text{model } M \text{ given by a functor } Mod \\
\text{signature } \Sigma \text{ in a category } \Sign \\
\text{sentence } \varphi \text{ given by a functor } sen
\end{array} \right.
\]

Mark Steedman: temporality is not just about timelines
Greg Carlson: *rules & regulations*

Tess is eating dal \( \neq \) Tess eats dal

From episodes/particulars to generics/types (frames)

models-as-strings \( \rightsquigarrow \) models-as-finite automata
Open-endedness & institutions

• more than one symbol in a box — alphabet as powerset

\[ \Sigma \leadsto 2^\Sigma := \{ A \mid A \subseteq \Sigma \} \]

• add symbols, possibly lengthening string
  e.g. real line \( \mathbb{R} \) from finite sets of rational numbers

Formulate \( \Sigma \) as a *signature* in an *institution* (Goguen & Burstall)

\[ M \models_{\Sigma} \varphi \left\{ \begin{array}{l}
\text{model } M \text{ given by a functor } \text{Mod} \\
\text{signature } \Sigma \text{ in a category } \text{Sign} \\
\text{sentence } \varphi \text{ given by a functor } \text{sen}
\end{array} \right. \]

Mark Steedman: temporality is not just about timelines
Greg Carlson: *rules & regulations*

Tess is eating dal \( \neq \) Tess eats dal

From episodes/particulars to generics/types (frames)

models-as-strings \( \leadsto \) models-as-finite automata
Strings built from fluents representing states

<table>
<thead>
<tr>
<th>it rained</th>
<th>tense</th>
<th>aspect</th>
</tr>
</thead>
<tbody>
<tr>
<td>E,R,S</td>
<td>R,S</td>
<td>E,R</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>it has rained</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>E,R,S</td>
<td>R,S</td>
<td>E,R</td>
</tr>
</tbody>
</table>

E,R,S as temporal propositions or fluents (for short)

Interpret a fluent \( \varphi \) over an interval \( I \)

\[
I \models \varphi
\]

leading to filmstrips such as

<table>
<thead>
<tr>
<th>months in a year</th>
<th>Jan</th>
<th>Feb</th>
<th>\cdots</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ d_1, d_2, \ldots, d_{31}</td>
<td>Jan, d_1</td>
<td>Jan, d_2</td>
<td>\cdots</td>
<td>Jan, d_{31}</td>
</tr>
</tbody>
</table>

where, for example, Jan represents a state

\[
I \models \text{Jan} \iff (\forall t \in I) \{t\} \models \text{Jan}
\]
Strings built from fluents representing states

<table>
<thead>
<tr>
<th>It rained</th>
<th>E,R,S</th>
<th>E,R,S</th>
</tr>
</thead>
<tbody>
<tr>
<td>It has rained</td>
<td>E,R,S</td>
<td>E,R,S</td>
</tr>
</tbody>
</table>

E,R,S as temporal propositions or *fluents* (for short)

Interpret a fluent $\varphi$ over an interval $I$

$$ I \models \varphi $$

leading to filmstrips such as

months in a year

+ $d_1,d_2,\ldots,d_{31}$

where, for example, Jan represents a state

$$ I \models \text{Jan} \iff (\forall t \in I) \{t\} \models \text{Jan} $$
Strings built from fluents representing states

<table>
<thead>
<tr>
<th>it rained</th>
<th>E, R, S</th>
<th>tense</th>
<th>aspect</th>
</tr>
</thead>
<tbody>
<tr>
<td>it has rained</td>
<td>E, R, S</td>
<td>R, S</td>
<td>E, R</td>
</tr>
</tbody>
</table>

E, R, S as temporal propositions or *fluents* (for short)

Interpret a fluent $\varphi$ over an interval $I$

$$I \models \varphi$$

leading to filmstrips such as

months in a year

<table>
<thead>
<tr>
<th>Jan</th>
<th>Feb</th>
<th>...</th>
<th>Dec</th>
</tr>
</thead>
</table>

$+$ d1, d2, ... d31

| Jan, d1 | Jan, d2 | ... | Jan, d31 | Feb, d1 | ... | Dec, d31 |

where, for example, Jan represents a state

$$I \models \text{Jan} \iff (\forall t \in I) \{ t \} \models \text{Jan}$$
Strings built from fluents representing states

<table>
<thead>
<tr>
<th>E,R,S</th>
<th>R,S</th>
<th>E,R</th>
</tr>
</thead>
<tbody>
<tr>
<td>it rained</td>
<td>it has rained</td>
<td></td>
</tr>
</tbody>
</table>

E,R,S as temporal propositions or *fluents* (for short)

Interpret a fluent $\varphi$ over an interval $I$

$$I \models \varphi$$

leading to filmstrips such as

<table>
<thead>
<tr>
<th>Jan</th>
<th>Feb</th>
<th>···</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan,d1</td>
<td>Jan,d2</td>
<td>···</td>
<td>Jan,d31</td>
</tr>
</tbody>
</table>

where, for example, Jan represents a state

$$I \models \text{Jan} \iff (\forall t \in I) \{t\} \models \text{Jan}$$
Strings built from fluents representing states

<table>
<thead>
<tr>
<th>It rained</th>
<th>E,R,S</th>
<th>tense</th>
<th>aspect</th>
</tr>
</thead>
<tbody>
<tr>
<td>It has rained</td>
<td>E,R,S</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

E,R,S as temporal propositions or *fluents* (for short)

Interpret a fluent $\varphi$ over an interval $I$

$$I \models \varphi$$

leading to filmstrips such as

months in a year

<table>
<thead>
<tr>
<th>Jan</th>
<th>Feb</th>
<th>...</th>
<th>Dec</th>
</tr>
</thead>
</table>
+ d1, d2, ... d31

| Jan,d1 | Jan,d2 | ... | Jan,d31 | Feb,d1 | ... | Dec,d31 |

where, for example, Jan represents a state

$$I \models \text{Jan} \iff (\forall t \in I) \{ t \} \models \text{Jan}$$
There is apparently a strong tendency to think that states are somehow basic, a sort of filmstrip view of reality which I do not share. If anything quite the opposite seems to be true. It took about two millennia to come up with satisfactory ways of coping with Zeno’s questions about what it could possibly mean to be in a state of motion at an instant or how you could possibly add together dimensionless instants to get changes (you can’t).

the “filmstrip” model of change . . . arguably is not the way movement and change is conceptualized (cf. Jackendoff 1996).
Against such strings

Bach 1986 (NLM, page 587)

There is apparently a strong tendency to think that states are somehow basic, a sort of filmstrip view of reality which I do not share. If anything quite the opposite seems to be true. It took about two millennia to come up with satisfactory ways of coping with Zeno’s questions about what it could possibly mean to be in a state of motion at an instant or how you could possibly add together dimensionless instants to get changes (you can’t).

Krifka 1998 (page 98)

the “filmstrip” model of change . . . arguably is not the way movement and change is conceptualized (cf. Jackendoff 1996).
I wish to reject this 'snapshot' conceptualization, on the grounds that it misrepresents the essential continuity of events of motion. For one thing, aside from the beginning and end points, the choice of a finite set of subevents is altogether arbitrary. How many subevents are there, and how is one to choose them? Notice that to stipulate the subevents as equally spaced, for instance one second or 3.5 milliseconds apart, is as arbitrary and unmotivated as any other choice.

Another difficulty with a 'snapshot' conceptualization concerns the representation of nonbounded events (activities) such as John ran along the river (for hours). A finite sequence of subevents necessarily has a specified beginning and ending, so it cannot encode the absence of endpoints. And excluding the specified endpoints simply exposes other specified subevents, which thereby become new endpoints. Thus encoding nonbounded events requires major surgery in the semantic representation.
I wish to reject this 'snapshot' conceptualization, on the grounds that it misrepresents the essential continuity of events of motion. For one thing, aside from the beginning and end points, the choice of a finite set of subevents is altogether arbitrary. How many subevents are there, and how is one to choose them? Notice that to stipulate the subevents as equally spaced, for instance one second or 3.5 milliseconds apart, is as arbitrary and unmotivated as any other choice.

Another difficulty with a 'snapshot' conceptualization concerns the representation of nonbounded events (activities) such as John ran along the river (for hours). A finite sequence of subevents necessarily has a specified beginning and ending, so it cannot encode the absence of endpoints. And excluding the specified endpoints simply exposes other specified subevents, which thereby become new endpoints. Thus encoding nonbounded events requires major surgery in the semantic representation.
Calendar & block compression

days in a year $\sim$ months in a year

<table>
<thead>
<tr>
<th>Jan,d1</th>
<th>Jan,d2</th>
<th>···</th>
<th>Dec,d31</th>
</tr>
</thead>
</table>

$\rho_{\text{months}} : \begin{align*}
&\begin{array}{cccc}
\text{Jan} & \text{Feb} & \cdots & \text{Dec} \\
31 & 28 & \cdots & 31
\end{array} \\
\end{align*}$

$\sim$

| Jan | Feb | ··· | Dec |

$\rho_{A}$ “see only $A$”

$\rho_{A}(\alpha_1 \alpha_2 \cdots \alpha_n) := (\alpha_1 \cap A)(\alpha_2 \cap A) \cdots (\alpha_n \cap A)$

$bc$ compress $\alpha^+$ to $\alpha$  
[ no identical adjacent boxes ]

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

$bc(\alpha \alpha') := \begin{cases} 
bc(\alpha s) & \text{if } \alpha = \alpha' \\
\alpha bc(\alpha' s) & \text{otherwise}
\end{cases}$

$\alpha_1 \cdots \alpha_n$ is stutterless if $\alpha_i \neq \alpha_{i+1}$ for $1 \leq i < n$  
(fixed pt of $bc$)

$bc_A$ is $\rho_{A}$; $bc$  
[ vocabulary ; ontology ]
Calendar & block compression

days in a year \sim \rightarrow \text{months in a year}

\begin{align*}
\text{Jan, d1} & \quad \text{Jan, d2} & \cdots & \quad \text{Dec, d31} \\
\end{align*}

\(\rho_{\text{months}}\)

\begin{align*}
\text{Jan}^{31} & \quad \text{Feb}^{28} & \cdots & \quad \text{Dec}^{31} \\
\end{align*}

\(\sim_{\text{bc}}\)

\begin{align*}
\text{Jan} & \quad \text{Feb} & \cdots & \quad \text{Dec} \\
\end{align*}

\(\rho_{A}\) “see only \(A\)”

\[
\rho_{A}(\alpha_1\alpha_2\cdots\alpha_n) := (\alpha_1 \cap A)(\alpha_2 \cap A) \cdots (\alpha_n \cap A)
\]

\(\text{bc}\) compress \(\alpha^+\) to \(\alpha\) [no identical adjacent boxes]

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

\[
\text{bc}(\alpha\alpha') := \begin{cases} 
\text{bc}(\alpha s) & \text{if } \alpha = \alpha' \\
\alpha \text{bc}(\alpha' s) & \text{otherwise}
\end{cases}
\]

\(\alpha_1 \cdots \alpha_n\) is stutterless if \(\alpha_i \neq \alpha_{i+1}\) for \(1 \leq i < n\) (fixed pt of \(bc\))

\(bc_{\mathcal{A}}\) is \(\rho_{\mathcal{A}}\); \(bc\) [vocabulary; ontology]
Calendar & block compression

days in a year $\sim$ months in a year

<table>
<thead>
<tr>
<th>Jan, d1</th>
<th>Jan, d2</th>
<th>...</th>
<th>Dec, d31</th>
</tr>
</thead>
</table>

$\rho_{months}$

<table>
<thead>
<tr>
<th>Jan</th>
<th>Feb</th>
<th>...</th>
<th>Dec</th>
</tr>
</thead>
</table>

$\rho_A$ “see only $A$”

$$\rho_A(\alpha_1 \alpha_2 \cdots \alpha_n) := (\alpha_1 \cap A)(\alpha_2 \cap A) \cdots (\alpha_n \cap A)$$

$bc$ compress $\alpha^+$ to $\alpha$ [ no identical adjacent boxes ]

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

$$bc(\alpha \alpha^s) := \begin{cases} 
bc(\alpha s) & \text{if } \alpha = \alpha' \\
\alpha \ bc(\alpha^s) & \text{otherwise} 
\end{cases}$$

$\alpha_1 \cdots \alpha_n$ is stutterless if $\alpha_i \neq \alpha_{i+1}$ for $1 \leq i < n$ (fixed pt of $bc$)

$bc_A$ is $\rho_A$; $bc$ [ vocabulary; ontology ]
Calendar & block compression

days in a year \sim months in a year

\begin{array}{ccc}
    \text{Jan,}d1 & \text{Jan,}d2 & \cdots & \text{Dec,}d31 \\
\end{array}
\sim
\begin{array}{ccc}
    \text{Jan}^{31} & \text{Feb}^{28} & \cdots & \text{Dec}^{31} \\
\end{array}

\rho_{\text{months}}

\begin{array}{ccc}
    \text{Jan} & \text{Feb} & \cdots & \text{Dec} \\
\end{array}

\rho_{\text{A}} \text{ “see only } A’’

\rho_{\text{A}}(\alpha_1 \alpha_2 \cdots \alpha_n) := (\alpha_1 \cap A)(\alpha_2 \cap A) \cdots (\alpha_n \cap A)

\rho_{\text{A}} (\alpha_1 \alpha_2 \cdots \alpha_n) := (\alpha_1 \cap A)(\alpha_2 \cap A) \cdots (\alpha_n \cap A)

\begin{array}{ccc}
    \text{bc} & \text{compress } \alpha^+ \text{ to } \alpha & [ \text{ no identical adjacent boxes } ] \\
\end{array}

\begin{align*}
\text{bc}(s) & := s \quad \text{if } \text{length}(s) \leq 1 \\
\text{bc}(\alpha \alpha’ s) & := \begin{cases} 
\text{bc}(\alpha s) & \text{if } \alpha = \alpha’ \\
\alpha \text{ bc}(\alpha’ s) & \text{otherwise}
\end{cases}
\end{align*}

\alpha_1 \cdots \alpha_n \text{ is stutterless if } \alpha_i \neq \alpha_{i+1} \text{ for } 1 \leq i < n \text{ (fixed pt of bc)}

\text{bc}_A \text{ is } \rho_{\text{A}}; \text{ bc} \quad [ \text{ vocabulary ; ontology } ]
For any set \( X \), let \( \text{Fin}(X) \) be the set of finite subsets of \( X \).

Let \( \lim_{\leftarrow} (b_{CA})_{A \in \text{Fin}(Q)} \) be the set of \( f : \text{Fin}(Q) \rightarrow \text{Fin}(Q)^{+} \) s.t.

\[
(\forall A \in \text{Fin}(Q))(\forall B \subseteq A) f(B) = b_{CB}(f(A)).
\]

E.g. \( str : \text{Fin}(Q) \rightarrow \text{Fin}(Q)^{+} \) s.t.

\[
str(\{a_1, \ldots, a_n\}) := \underbrace{a_1, \ldots, a_n}_{\text{where } a_1 < \cdots < a_n}
\]

and also for \( C \subseteq Q \), \( str_C : \text{Fin}(Q) \rightarrow \text{Fin}(C)^{+} \) s.t.

\[
str_C(A) := str(A \cap C).
\]
The real line $\mathbb{R}$ & inverse limits

$\mathbb{R}$ and numbers $a' < a < a'' < \cdots$

For any set $X$, let $\text{Fin}(X)$ be the set of finite subsets of $X$.

Let $\lim_{\leftarrow} \,(bc_A)_{A \in \text{Fin}(\mathbb{Q})}$ be the set of $f : \text{Fin}(\mathbb{Q}) \to \text{Fin}(\mathbb{Q})^*$ s.t.

$$(\forall A \in \text{Fin}(\mathbb{Q}))(\forall B \subseteq A) \; f(B) = bc_B(f(A)).$$

E.g. $str : \text{Fin}(\mathbb{Q}) \to \text{Fin}(\mathbb{Q})^+$ s.t.

$$str(\{a_1, \ldots, a_n\}) := \begin{array}{c} a_1 \\ \vdots \\ a_n \end{array} \quad \text{where} \; a_1 < \cdots < a_n$$

and also for $C \subseteq \mathbb{Q}$, $str_C : \text{Fin}(\mathbb{Q}) \to \text{Fin}(C)^+$ s.t.

$$str_C(A) := str(A \cap C).$$
The real line $\mathbb{R}$ & inverse limits

$\mathbb{R}$ and numbers $a' < a < a'' < \ldots$

\[
\begin{array}{cccc}
\varnothing & \sim & \{a\} & \sim \{a', a\} & \sim \{a', a, a''\} & \sim \ldots
\end{array}
\]

For any set $X$, let $\operatorname{Fin}(X)$ be the set of finite subsets of $X$.

Let $\varprojlim (bc_A)_{A \in \operatorname{Fin}(\mathbb{Q})}$ be the set of $f : \operatorname{Fin}(\mathbb{Q}) \to \operatorname{Fin}(\mathbb{Q})^*$ s.t.

\[
(\forall A \in \operatorname{Fin}(\mathbb{Q}))(\forall B \subseteq A) f(B) = bc_B(f(A)).
\]

E.g. $\operatorname{str} : \operatorname{Fin}(\mathbb{Q}) \to \operatorname{Fin}(\mathbb{Q})^+$ s.t.

\[
\operatorname{str} \left( \{a_1, \ldots, a_n\} \right) := \{a_1\} \cdots \{a_n\} \quad \text{where } a_1 < \cdots < a_n
\]

and also for $C \subseteq \mathbb{Q}$, $\operatorname{str}_C : \operatorname{Fin}(\mathbb{Q}) \to \operatorname{Fin}(C)^+$ s.t.

\[
\operatorname{str}_C(A) := \operatorname{str}(A \cap C).
\]
The real line $\mathbb{R}$ & inverse limits

$\mathbb{R}$ and numbers $a' < a < a'' < \cdots$

$$\begin{array}{cccccc}
\emptyset & \sim & \{ a \} & \sim & \{ a', a \} & \sim & \{ a', a, a'' \} & \sim & \cdots
\end{array}$$

For any set $X$, let $\text{Fin}(X)$ be the set of finite subsets of $X$.

Let $\varprojlim (bc_A)_{A \in \text{Fin}(\mathbb{Q})}$ be the set of $f : \text{Fin}(\mathbb{Q}) \to \text{Fin}(\mathbb{Q})^*$ s.t.

$$(\forall A \in \text{Fin}(\mathbb{Q}))(\forall B \subseteq A) f(B) = bc_B(f(A)).$$

E.g. $str : \text{Fin}(\mathbb{Q}) \to \text{Fin}(\mathbb{Q})^+$ s.t.

$$str(\{ a_1, \ldots, a_n \}) := \{ a_1 \} \cdots \{ a_n \} \text{ where } a_1 < \cdots < a_n$$

and also for $C \subseteq \mathbb{Q}$, $str_C : \text{Fin}(\mathbb{Q}) \to \text{Fin}(C)^+$ s.t.

$$str_C(A) := str(A \cap C).$$
The real line \( \mathbb{R} \) & inverse limits

\( \mathbb{R} \) and numbers \( a' < a < a'' < \cdots \)

\[
\begin{array}{c}
\emptyset \\
\sim \quad a \\
\sim \quad a' \\
\sim \quad a' a \\
\sim \quad a' a a'' \\
\sim \quad \cdots
\end{array}
\]

For any set \( X \), let \( \text{Fin}(X) \) be the set of finite subsets of \( X \).

Let \( \lim_{\leftarrow} (bc_A)_{A \in \text{Fin}(\mathbb{Q})} \) be the set of \( f : \text{Fin}(\mathbb{Q}) \to \text{Fin}(\mathbb{Q})^* \) s.t.

\[
(\forall A \in \text{Fin}(\mathbb{Q}))(\forall B \subseteq A) \ f(B) = bc_B(f(A)).
\]

E.g. \( str : \text{Fin}(\mathbb{Q}) \to \text{Fin}(\mathbb{Q})^+ \) s.t.

\[
str(\{a_1, \ldots, a_n\}) := a_1 \cdots a_n \text{ where } a_1 < \cdots < a_n
\]

and also for \( C \subseteq \mathbb{Q} \), \( str_C : \text{Fin}(\mathbb{Q}) \to \text{Fin}(C)^+ \) s.t.

\[
str_C(A) := str(A \cap C).
\]
The real line $\mathbb{R}$ & inverse limits

$\mathbb{R}$ and numbers $a' < a < a'' < \cdots$

$$
\emptyset \sim \begin{array}{c} a \\ a \\ a' \\ a' \\ a'' \\
\end{array} \sim \cdots
$$

For any set $X$, let $\text{Fin}(X)$ be the set of finite subsets of $X$.

Let $\underset{\leftarrow}{\lim} \ (b_{c_A})_{A \in \text{Fin}(\mathbb{Q})}$ be the set of $f : \text{Fin}(\mathbb{Q}) \rightarrow \text{Fin}(\mathbb{Q})^*$ s.t.

$$(\forall A \in \text{Fin}(\mathbb{Q}))(\forall B \subseteq A) \ f(B) = b_{c_B}(f(A)).$$

E.g. $\text{str} : \text{Fin}(\mathbb{Q}) \rightarrow \text{Fin}(\mathbb{Q})^+$ s.t.

$$\text{str}(\{a_1, \ldots, a_n\}) := \begin{array}{c} a_1 \\ \cdots \\ a_n \\
\end{array} \quad \text{where } a_1 < \cdots < a_n$$

and also for $C \subseteq \mathbb{Q}$, $\text{str}_C : \text{Fin}(\mathbb{Q}) \rightarrow \text{Fin}(C)^+$ s.t.

$$\text{str}_C(A) := \text{str}(A \cap C).$$
The real line $\mathbb{R}$ & inverse limits

$\mathbb{R}$ and numbers $a' < a < a'' < \cdots$

\[
\begin{align*}
\emptyset & \leadsto \{a\} \leadsto \{a', a\} \leadsto \{a', a, a''\} \leadsto \cdots
\end{align*}
\]

For any set $X$, let $\text{Fin}(X)$ be the set of finite subsets of $X$.

Let $\lim\left(\mathbb{b}_A\right)_{A \in \text{Fin}(\mathbb{Q})}$ be the set of $f : \text{Fin}(\mathbb{Q}) \to \text{Fin}(\mathbb{Q})^*$ s.t.

\[(\forall A \in \text{Fin}(\mathbb{Q}))(\forall B \subseteq A) f(B) = \mathbb{b}_B(f(A)).\]

E.g. $str : \text{Fin}(\mathbb{Q}) \to \text{Fin}(\mathbb{Q})^+$ s.t.

\[str(\{a_1, \ldots, a_n\}) := \{a_1\} \cdots \{a_n\} \quad \text{where } a_1 < \cdots < a_n\]

and also for $C \subseteq \mathbb{Q}$, $str_C : \text{Fin}(\mathbb{Q}) \to \text{Fin}(C)^+$ s.t.

\[str_C(A) := str(A \cap C).\]
Map a real number to the set of rational numbers less than it

\[ \text{left}_Q : \mathbb{R} \rightarrow 2^\mathbb{Q}, \ r \mapsto \{ q \in \mathbb{Q} \mid q < r \} \]

Let \( \text{Cut} \) be the set of \( C \subseteq \mathbb{Q} \) s.t.

\[
\begin{align*}
\emptyset &\neq C \neq \mathbb{Q} \\
(\forall a \in C)(\forall a' \in \mathbb{Q}) \ a' < a &\implies a' \in C \quad \text{(lower half)} \\
(\forall a \in C)(\exists a' \in C) \ a < a' &\quad \text{(else 2 copies of } \mathbb{Q})
\end{align*}
\]

Fact. \( \text{left}_Q : \langle \mathbb{R}, < \rangle \cong \langle \text{Cut}, \subset \rangle \) (extends to ordered field)

\[
\cong \langle \{ \text{str}_C \mid C \in \text{Cut} \}, < \rangle
\]

where for \( f, f' \in \lim_{\leftarrow} (bc_A)_{A \in \text{Fin}(\Phi)} \),

\[ f < f' \iff f \neq f' \text{ and } (\forall A \in \text{Fin}(\Phi)) \ f(A) \leq_{\text{prefix}} f'(A) \]

and for strings \( s, s' \),

\[ s \leq_{\text{prefix}} s' \iff (\exists x) \ sx = s'. \]
Dedekind cuts & prefixes

Map a real number to the set of rational numbers less than it

$$\text{left}_Q : \mathbb{R} \to 2^\mathbb{Q}, \ r \mapsto \{ q \in \mathbb{Q} \mid q < r \}$$

Let \( \textbf{Cut} \) be the set of \( C \subseteq \mathbb{Q} \) s.t.

- \( \emptyset \neq C \neq \mathbb{Q} \)
- \( (\forall a \in C)(\forall a' \in \mathbb{Q}) \ a' < a \) implies \( a' \in C \) (lower half)
- \( (\forall a \in C)(\exists a' \in C) \ a < a' \) (else 2 copies of \( \mathbb{Q} \))

**Fact.** \( \text{left}_Q : \langle \mathbb{R}, < \rangle \cong \langle \textbf{Cut}, \subset \rangle \) (extends to ordered field)

\[ \cong \langle \{ \text{str}_C \mid C \in \textbf{Cut} \}, \prec \rangle \]

where for \( f, f' \in \lim_{\leftarrow} (bc_A)_{A \in \text{Fin}(\Phi)} \),

\[ f \prec f' \iff f \neq f' \text{ and } (\forall A \in \text{Fin}(\Phi)) f(A) \leq_{\text{prefix}} f'(A) \]

and for strings \( s, s' \),

\[ s \leq_{\text{prefix}} s' \iff (\exists x)\ sx = s'. \]
Map a real number to the set of rational numbers less than it

\[
\text{left}_\mathbb{Q} : \mathbb{R} \to 2^\mathbb{Q}, \quad r \mapsto \{ q \in \mathbb{Q} \mid q < r \}
\]

Let **Cut** be the set of \( C \subseteq \mathbb{Q} \) s.t.

\[
\emptyset \neq C \neq \mathbb{Q} \\
(\forall a \in C)(\forall a' \in \mathbb{Q}) \text{ } a' < a \text{ implies } a' \in C \quad \text{(lower half)}
\]
\[
(\forall a \in C)(\exists a' \in C) \text{ } a < a' \quad \text{(else 2 copies of } \mathbb{Q})
\]

**Fact.** \( \text{left}_\mathbb{Q} : \langle \mathbb{R}, < \rangle \cong \langle \textbf{Cut}, \subseteq \rangle \) (extends to ordered field)

\[
\cong \langle \{ \text{str}_C \mid C \in \textbf{Cut} \}, < \rangle
\]

where for \( f, f' \in \varprojlim (bc_A)_{A \in \text{Fin}(\Phi)} \),

\[
f < f' \iff f \neq f' \text{ and } (\forall A \in \text{Fin}(\Phi)) \text{ } f(A) \leq_{\text{prefix}} f'(A)
\]

and for strings \( s, s' \),

\[
s \leq_{\text{prefix}} s' \iff (\exists x) \text{ } sx = s'.
\]
Dedekind cuts & prefixes

Map a real number to the set of rational numbers less than it

\[
\text{left}_\mathbb{Q} : \mathbb{R} \to 2^\mathbb{Q}, \ r \mapsto \{ q \in \mathbb{Q} \mid q < r \}
\]

Let \textbf{Cut} be the set of \( C \subseteq \mathbb{Q} \) s.t.

\[
\emptyset \neq C \neq \mathbb{Q}
\]

\[
(\forall a \in C)(\forall a' \in \mathbb{Q}) \ a' < a \text{ implies } a' \in C \quad \text{(lower half)}
\]

\[
(\forall a \in C)(\exists a' \in C) \ a < a' \quad \text{(else 2 copies of } \mathbb{Q})
\]

\textbf{Fact.} \quad \text{left}_\mathbb{Q} : \langle \mathbb{R}, < \rangle \cong \langle \textbf{Cut}, \subseteq \rangle \quad \text{(extends to ordered field)}

\[
\cong \langle \{ \text{str}_C \mid C \in \textbf{Cut} \}, \prec \rangle
\]

where for \( f, f' \in \lim_{\leftarrow} (bc_A)_{A \in \text{Fin}(\Phi)} \),

\[
f \prec f' \iff f \neq f' \text{ and } (\forall A \in \text{Fin}(\Phi)) f(A) \leq_{\text{prefix}} f'(A)
\]

and for strings \( s, s' \),

\[
s \leq_{\text{prefix}} s' \iff (\exists x) sx = s'.
\]
Where are we?

We formed the continuum $\mathbb{R}$ from stutterless strings in $\text{Fin}(\mathbb{Q})^+$

$\{bc(s) \mid s \in \text{Fin}(\mathbb{Q})^+\}$

Next: consider intervals

- Reichenbachian progressive R inside E
- subinterval property (Bennett and Partee 1972/8)

Ed slept from 3 to 6

$\varphi \models I \models \varphi$

$I \models \varphi$ and $I' \subseteq I$ implies $I' \models \varphi$

Ed slept from 3 to 5pm, Ed slept from 4 to 6pm

$\vdash I$ Ed slept from 3 to 6pm
Where are we?

We formed the continuum $\mathbb{R}$ from stutterless strings in $\text{Fin}(\mathbb{Q})^+$

$$\{ bc(s) \mid s \in \text{Fin}(\mathbb{Q})^+ \}$$

Next: consider intervals

- Reichenbachian progressive $R$ inside $E$
  - subinterval property (Bennett and Partee 1972/8)

$$\text{Ed slept from 3 to 6} \quad \varnothing \quad I \quad \text{Ed slept from 3 to 5} \quad \varnothing \quad I'$$

$$I \models \varnothing \text{ and } I' \subseteq I \quad \text{implies} \quad I' \models \varnothing$$

Ed slept from 3 to 5pm, \quad Ed slept from 4 to 6pm

\[ \vdash \quad \text{Ed slept from 3 to 6pm} \]
Where are we?

We formed the continuum $\mathbb{R}$ from stutterless strings in $\text{Fin}(\mathbb{Q})^+$

\[ \{ b \varepsilon(s) \mid s \in \text{Fin}(\mathbb{Q})^+ \} \]

**Next**: consider intervals

- Reichenbachian progressive $R$ inside $E$
- subinterval property (Bennett and Partee 1972/8)

\[
\begin{align*}
\text{Ed slept from 3 to 6} & \quad \vdash \quad \text{Ed slept from 3 to 5} \\
\varphi & \quad I & \quad \varphi & \quad I' \\
& \quad \models \varphi \text{ and } I' \subseteq I & \Rightarrow & \quad I' \models \varphi
\end{align*}
\]

Ed slept from 3 to 5pm, Ed slept from 4 to 6pm

\[ \vdash \quad \text{Ed slept from 3 to 6pm} \]
We formed the continuum $\mathbb{R}$ from stutterless strings in $\text{Fin}(\mathbb{Q})^+$

\[
\{ b\mathcal{C}(s) \mid s \in \text{Fin}(\mathbb{Q})^+ \}
\]

**Next:** consider intervals
- Reichenbachian progressive $\mathcal{R}$ inside $E$
- subinterval property (Bennett and Partee 1972/8)

\[
\begin{align*}
\text{Ed slept from 3 to 6} & \models \quad \text{Ed slept from 3 to 5} \\
\varphi & \quad I & \varphi & \quad I'
\end{align*}
\]

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

Ed slept from 3 to 5pm, \quad Ed slept from 4 to 6pm
\[
\models \quad \text{Ed slept from 3 to 6pm}
\]

...
Where are we?

We formed the continuum \( \mathbb{R} \) from stutterless strings in \( \text{Fin}(\mathbb{Q})^+ \)
\[ \{ bc(s) \mid s \in \text{Fin}(\mathbb{Q})^+ \} \]

Next: consider intervals
- Reichenbachian progressive \( R \) inside \( E \)
- subinterval property (Bennett and Partee 1972/8)

\[
\text{Ed slept from 3 to 6} \vdash \text{Ed slept from 3 to 5}
\]

\[
\varnothing \vdash \varnothing \text{ and } I' \subseteq I \text{ implies } I' \vdash \varnothing
\]

Ed slept from 3 to 5pm,  Ed slept from 4 to 6pm
\[
\vdash \text{Ed slept from 3 to 6pm}
\]

\[ \vdots \]
From points to intervals

Fix a (strict) linear order $<$ on some non-empty set $T$ of points.

We lift $<$ to non-empty subsets $I, I'$ of $T$ via $\forall$

$$I < I' \iff (\forall t \in I)(\forall t' \in I') \; t < t'$$

for the left and right sides of $I$

$$\begin{align*}
\text{left}(I) & := \{t \in T \mid \{t\} < I\} \\
\text{right}(I) & := \{t \in T \mid I < \{t\}\}.
\end{align*}$$

The set $\mathbf{lvl}$ of intervals (wrt $<$) is

$$\mathbf{lvl} := \{I \subseteq T \mid T \subseteq \text{left}(I) \cup I \cup \text{right}(I)\} - \{\emptyset\}$$

and a satisfaction relation $|=_{\circ}$ between $T$ and a set $\Phi$ of fluents $\varphi$ can be lifted to intervals $I$ via $\forall$

$$I |=_{\varphi_{\forall}} \iff (\forall t \in I) \; t |=_{\circ} \varphi.$$
From points to intervals

Fix a (strict) linear order $<$ on some non-empty set $T$ of points. We lift $<$ to non-empty subsets $I, I'$ of $T$ via $\forall$

$$I < I' \iff (\forall t \in I)(\forall t' \in I') t < t'$$

for the left and right sides of $I$

$$\text{left}(I) := \{ t \in T \mid \{t\} < I \}$$
$$\text{right}(I) := \{ t \in T \mid I < \{t\} \}.$$ 

The set $\mathbf{I}_v\mathbf{l}$ of intervals (wrt $<$) is

$$\mathbf{I}_v\mathbf{l} := \{ I \subseteq T \mid T \subseteq \text{left}(I) \cup I \cup \text{right}(I) \} - \{\emptyset\}$$

and a satisfaction relation $|=\circ$ between $T$ and a set $\Phi$ of fluents $\varphi$ can be lifted to intervals $I$ via $\forall$

$$I |= \varphi_\forall \iff (\forall t \in I) t |=\circ \varphi.$$
From points to intervals

Fix a (strict) linear order $<$ on some non-empty set $T$ of points.

We lift $<$ to non-empty subsets $I, I'$ of $T$ via $\forall$

$$I < I' \iff (\forall t \in I)(\forall t' \in I') t < t'$$

for the left and right sides of $I$

$$\text{left}(I) := \{ t \in T \mid \{t\} < I \}$$
$$\text{right}(I) := \{ t \in T \mid I < \{t\} \}.$$

The set $\mathbf{Ivl}$ of intervals (wrt $<$) is

$$\mathbf{Ivl} := \{ I \subseteq T \mid T \subseteq \text{left}(I) \cup I \cup \text{right}(I) \} - \{\emptyset\}$$

and a satisfaction relation $|=_{\circ}$ between $T$ and a set $\Phi$ of fluents $\varphi$ can be lifted to intervals $I$ via $\forall$

$$I |=_{\varphi_{\forall}} \iff (\forall t \in I) t |=_{\circ} \varphi.$$
From points to intervals

Fix a (strict) linear order \(<\) on some non-empty set \(T\) of points.

We lift \(<\) to non-empty subsets \(I, I'\) of \(T\) via \(\forall\)

\[
I < I' \iff (\forall t \in I)(\forall t' \in I') \ t < t'
\]

for the left and right sides of \(I\)

\[
\text{left}(I) := \{t \in T \mid \{t\} < I\}
\]

\[
\text{right}(I) := \{t \in T \mid I < \{t\}\}.
\]

The set \(\text{lvl}\) of intervals (wrt \(<\) is

\[
\text{lvl} := \{I \subseteq T \mid T \subseteq \text{left}(I) \cup I \cup \text{right}(I)\} - \{\emptyset\}
\]

and a satisfaction relation \(\models_\circ\) between \(T\) and a set \(\Phi\) of fluents \(\varphi\)
can be lifted to intervals \(I\) via \(\forall\)

\[
I \models_\circ \varphi_\forall \iff (\forall t \in I) \ t \models_\circ \varphi.
\]
From points to intervals

Fix a (strict) linear order $<$ on some non-empty set $T$ of points.

We lift $<$ to non-empty subsets $I, I'$ of $T$ via $\forall$

$$I < I' \iff (\forall t \in I)(\forall t' \in I') \ t < t'$$

for the left and right sides of $I$

$$\text{left}(I) := \{ t \in T \mid \{t\} < I \}$$
$$\text{right}(I) := \{ t \in T \mid I < \{t\} \}.$$ 

The set $\text{lvl}$ of intervals (wrt $<$) is

$$\text{lvl} := \{ I \subseteq T \mid T \subseteq \text{left}(I) \cup I \cup \text{right}(I) \} - \{\emptyset\}$$

and a satisfaction relation $|=_{\circ}$ between $T$ and a set $\Phi$ of fluents $\varphi$ can be lifted to intervals $I$ via $\forall$

$$I |=_{\circ} \varphi_{\forall} \iff (\forall t \in I) \ t |=_{\circ} \varphi.$$
Fluents over intervals & Dowty’s aspect hypothesis

Interpret fluents from a set $\Phi$ over a $\Phi$-timeline $\mathcal{A} = \langle T, \prec, \models \rangle$

- a linear order $\prec$ on a non-empty set $T$ of points
- a binary relation $\models$ between intervals $I$ and fluents in $\Phi$

Given $\models$ between $\text{Ivl}$ and $\Phi$, call $\varphi$ pointwise if

$$I \models \varphi \iff (\forall t \in I) \{t\} \models \varphi.$$

Widespread assumption (Taylor 1977, Dowty 1979, ...). Fluents expressing statives (predicates over states) are pointwise.

Dowty’s Aspect Hypothesis (1979)

the different aspectual properties of the various kinds of verbs can be explained by postulating a single homogeneous class of predicates – stative predicates – plus three or four sentential operators and connectives
Interpret fluents from a set $\Phi$ over a $\Phi$-timeline $\mathcal{A} = \langle T, \prec, \mid \rangle$
- a linear order $\prec$ on a non-empty set $T$ of points
- a binary relation $\mid$ between intervals $I$ and fluents in $\Phi$

Given $\mid$ between $I$vl and $\Phi$, call $\varphi$ pointwise if

$$I \mid \varphi \iff (\forall t \in I) \{t\} \mid \varphi.$$ 

Widespread assumption (Taylor 1977, Dowty 1979, ...). Fluents expressing statives (predicates over states) are pointwise.

Dowty’s Aspect Hypothesis (1979)

the different aspectual properties of the various kinds of verbs can be explained by postulating a single homogeneous class of predicates – stative predicates – plus three or four sentential operators and connectives
Interpret fluents from a set $\Phi$ over a $\Phi$-timeline $\mathcal{A} = \langle T, \prec, \models \rangle$

- a linear order $\prec$ on a non-empty set $T$ of points
- a binary relation $\models$ between intervals $I$ and fluents in $\Phi$

Given $\models$ between $IvI$ and $\Phi$, call $\varphi$ pointwise if

$$I \models \varphi \iff (\forall t \in I) \{t\} \models \varphi.$$ 

**Widespread assumption** (Taylor 1977, Dowty 1979, ...). Fluents expressing statives (predicates over states) are pointwise.

**Dowty’s Aspect Hypothesis** (1979)

the different aspectual properties of the various kinds of verbs can be explained by postulating a single homogeneous class of predicates – stative predicates – plus three or four sentential operators and connectives
Fluents over intervals & Dowty’s aspect hypothesis

Interpret fluents from a set $\Phi$ over a $\Phi$-timeline $\mathcal{A} = \langle T, \prec, \models \rangle$
- a linear order $\prec$ on a non-empty set $T$ of points
- a binary relation $\models$ between intervals $I$ and fluents in $\Phi$

Given $\models$ between $Ivl$ and $\Phi$, call $\varphi$ pointwise if

$$I \models \varphi \iff (\forall t \in I) \{t\} \models \varphi.$$ 

Widespread assumption (Taylor 1977, Dowty 1979, ...).
Fluents expressing statives (predicates over states) are pointwise.

Dowty’s Aspect Hypothesis (1979)

the different aspectual properties of the various kinds of verbs can be explained by postulating a single homogeneous class of predicates – stative predicates – plus three or four sentential operators and connectives
The negation $\neg \varphi$ of a pointwise $\varphi$ need not be pointwise

$$I \models \neg \varphi \iff \text{not } I \models \varphi$$

An interval $I$ is $\varphi$-homogeneous if $\varphi$ is satisfied by either all or none of the subintervals of $I$

$$(\forall I' \subseteq I) \quad I' \models \varphi \iff (\exists I' \subseteq I) \quad I' \models \varphi$$

(where the subinterval relation $\subseteq$ is $\subseteq$ restricted to $\text{ivl}$).

N.B. $I$ is $\varphi$-homogeneous $\iff I$ is $\neg \varphi$-homogeneous

Wikipedia on Ramsey’s theorem:

one will find monochromatic cliques in any edge colouring of a sufficiently large complete graph.

Plan. Segment an interval into homogeneous subintervals $\leadsto$ string
Negation & homogeneity

The negation $\neg \varphi$ of a pointwise $\varphi$ need not be pointwise

$$ I \models \neg \varphi \iff \neg I \models \varphi $$

An interval $I$ is $\varphi$-homogeneous if $\varphi$ is satisfied by either all or none of the subintervals of $I$

$$ (\forall I' \subseteq I) \ I' \models \varphi \iff (\exists I' \subseteq I) \ I' \models \varphi $$

(where the subinterval relation $\subseteq$ is $\subseteq$ restricted to $|I|$).

N.B. $I$ is $\varphi$-homogeneous $\iff$ $I$ is $\neg \varphi$-homogeneous

Wikipedia on Ramsey’s theorem:

one will find monochromatic cliques in any edge colouring of a sufficiently large complete graph.

Plan. Segment an interval into homogeneous subintervals $\leadsto$ string
Negation & homogeneity

The negation $\neg \varphi$ of a pointwise $\varphi$ need not be pointwise

$$I \models \neg \varphi \iff \neg I \models \varphi$$

An interval $I$ is $\varphi$-homogeneous if $\varphi$ is satisfied by either all or none of the subintervals of $I$

$$(\forall I' \subseteq I) \; I' \models \varphi \iff (\exists I' \subseteq I) \; I' \models \varphi$$

(where the subinterval relation $\subseteq$ is $\subseteq$ restricted to $\mathbf{lvI}$).

N.B. $I$ is $\varphi$-homogeneous $\iff I$ is $\neg \varphi$-homogeneous

Wikipedia on Ramsey’s theorem:

*one will find monochromatic cliques in any edge colouring of a sufficiently large complete graph.*

**Plan.** Segment an interval into homogeneous subintervals $\leadsto$ string
Negation & homogeneity

The negation $\neg \varphi$ of a pointwise $\varphi$ need not be pointwise

$$ I \models \neg \varphi \iff \neg \exists I \models \varphi $$

An interval $I$ is $\varphi$-homogeneous if $\varphi$ is satisfied by either all or none of the subintervals of $I$

$$ (\forall I' \subseteq I) \quad I' \models \varphi \iff (\exists I' \subseteq I) \quad I' \models \varphi $$

(where the subinterval relation $\subseteq$ is $\subseteq$ restricted to $\text{lv} I$).

N.B. $I$ is $\varphi$-homogeneous $\iff$ $I$ is $\neg \varphi$-homogeneous

Wikipedia on Ramsey’s theorem:

one will find monochromatic cliques in any edge colouring of a sufficiently large complete graph.
Negation & homogeneity

The negation $\neg \varphi$ of a pointwise $\varphi$ need not be pointwise

$$I \models \neg \varphi \iff \neg \left( I \models \varphi \right)$$

An interval $I$ is $\varphi$-homogeneous if $\varphi$ is satisfied by either all or none of the subintervals of $I$

$$(\forall I' \sqsubseteq I) \quad I' \models \varphi \iff (\exists I' \sqsubseteq I) \quad I' \models \varphi$$

(where the subinterval relation $\sqsubseteq$ is $\subseteq$ restricted to $|I|$).

N.B. $I$ is $\varphi$-homogeneous $\iff$ $I$ is $\neg \varphi$-homogeneous

Wikipedia on Ramsey’s theorem:

one will find monochromatic cliques in any edge colouring of a sufficiently large complete graph.

Plan. Segment an interval into homogeneous subintervals $\rightsquigarrow$ string
A segmentation of $I$ is a $<-\text{-increasing sequence}$ $I_1 \cdots I_n$ covering $I$

$$I_1 \cdots I_n \nearrow I \iff \bigcup_{i=1}^n I_i = I \text{ and for } 1 \leq i < n, \ I_i < I_{i+1}$$

— i.e. a finite partition of $I$ ordered by $<$. 

A seg is a segmentation $I_1 \cdots I_n$ of $\bigcup_{i=1}^n I_i$.

A $\varphi$-segmentation is a seg $I_1 \cdots I_n$ s.t. each $I_i$ is $\varphi$-homogeneous.

A seg $I_1 \cdots I_n$ tracks $\varphi$ if for all $I \subseteq \bigcup_{i=1}^n I_i,$

$$I \models \varphi \iff I \subseteq \bigcup \{I_i \mid 1 \leq i \leq n \text{ and } I_i \models \varphi\}.$$ 

Fact. For pointwise $\varphi$, a seg is a $\varphi$-segmentation iff it tracks $\varphi$. 

Segmentations
A segmentation of $I$ is a $\prec$-increasing sequence $l_1 \cdots l_n$ covering $I$

$$l_1 \cdots l_n \not\prec I \iff \bigcup_{i=1}^{n} l_i = I \text{ and for } 1 \leq i < n, \ l_i < l_{i+1}$$

— i.e. a finite partition of $I$ ordered by $\prec$.

A seg is a segmentation $l_1 \cdots l_n$ of $\bigcup_{i=1}^{n} l_i$.

A $\varphi$-segmentation is a seg $l_1 \cdots l_n$ s.t. each $l_i$ is $\varphi$-homogeneous.

A seg $l_1 \cdots l_n$ tracks $\varphi$ if for all $I \subseteq \bigcup_{i=1}^{n} l_i$,

$$I \models \varphi \iff I \subseteq \bigcup \{l_i \mid 1 \leq i \leq n \text{ and } l_i \models \varphi\}.$$ 

Fact. For pointwise $\varphi$, a seg is a $\varphi$-segmentation iff it tracks $\varphi$. 

---

**Segmentations**
A segmentation of $I$ is a $\prec$-increasing sequence $I_1 \cdots I_n$ covering $I$

$$I_1 \cdots I_n \overset{\succ}{\rightarrow} I \iff \bigcup_{i=1}^{n} I_i = I \text{ and for } 1 \leq i < n, I_i < I_{i+1}$$

— i.e. a finite partition of $I$ ordered by $\prec$.

A seg is a segmentation $I_1 \cdots I_n$ of $\bigcup_{i=1}^{n} I_i$.

A $\varphi$-segmentation is a seg $I_1 \cdots I_n$ s.t. each $I_i$ is $\varphi$-homogeneous.

A seg $I_1 \cdots I_n$ tracks $\varphi$ if for all $I \subseteq \bigcup_{i=1}^{n} I_i$,

$$I \models \varphi \iff I \subseteq \bigcup \{ I_i \mid 1 \leq i \leq n \text{ and } I_i \models \varphi \}.$$  

Fact. For pointwise $\varphi$, a seg is a $\varphi$-segmentation iff it tracks $\varphi$.  

A segmentation of \( I \) is a \(<\)-increasing sequence \( l_1 \cdots l_n \) covering \( I \)
\[
l_1 \cdots l_n \nearrow I \iff \bigcup_{i=1}^{n} l_i = I \quad \text{and for } 1 \leq i < n, \ l_i < l_{i+1}
\]
— i.e. a finite partition of \( I \) ordered by \(<\).

A seg is a segmentation \( l_1 \cdots l_n \) of \( \bigcup_{i=1}^{n} l_i \).

A \( \varphi \)-segmentation is a seg \( l_1 \cdots l_n \) s.t. each \( l_i \) is \( \varphi \)-homogeneous.

A seg \( l_1 \cdots l_n \) tracks \( \varphi \) if for all \( I \subseteq \bigcup_{i=1}^{n} l_i \),
\[
I \models \varphi \iff I \subseteq \bigcup \{ l_i \mid 1 \leq i \leq n \text{ and } l_i \models \varphi \}.
\]

**Fact.** For pointwise \( \varphi \), a seg is a \( \varphi \)-segmentation iff it tracks \( \varphi \).
A segmentation of $I$ is a $\prec$-increasing sequence $I_1 \cdots I_n$ covering $I$

\[ I_1 \cdots I_n \prec I \iff \bigcup_{i=1}^{n} I_i = I \text{ and for } 1 \leq i < n, \ I_i \prec I_{i+1} \]
— i.e. a finite partition of $I$ ordered by $\prec$.

A seg is a segmentation $I_1 \cdots I_n$ of $\bigcup_{i=1}^{n} I_i$.

A $\varphi$-segmentation is a seg $I_1 \cdots I_n$ s.t. each $I_i$ is $\varphi$-homogeneous.

A seg $I_1 \cdots I_n$ tracks $\varphi$ if for all $I \subseteq \bigcup_{i=1}^{n} I_i$,

\[ I \models \varphi \iff I \subseteq \bigcup \{I_i \mid 1 \leq i \leq n \text{ and } I_i \models \varphi \}. \]

**Fact.** For pointwise $\varphi$, a seg is a $\varphi$-segmentation iff it tracks $\varphi$. 
Alternations

When does an interval $I$ have a $\varphi$-segmentation?

A $(\varphi, n)$-alternation of $I$ is a string $t_1 \cdots t_n \in I^n$ s.t. for $1 \leq i < n$,

\[ t_i < t_{i+1} \quad \text{and} \quad \{t_i\} \models \varphi \iff \{t_{i+1}\} \not\models \varphi \]

(e.g. $\{t_1\} \models \varphi$, $\{t_2\} \not\models \varphi$, $\{t_3\} \models \varphi$, ...).

$I$ is $\varphi$-alternation bounded (a.b.) if there is an integer $n > 0$ s.t. no $(\varphi, n)$-alternation of $I$ exists.

**Fact.** For pointwise $\varphi$, $I$ has a $\varphi$-segmentation iff $I$ is $\varphi$-a.b.

Stepping up from a fluent $\varphi$ to a set $A$ of fluents, we call a segmentation an $A$-segmentation if it is a $\varphi$-segmentation for each $\varphi \in A$.

$I$ is $A$-segmentable if there is an $A$-segmentation of $I$.

**Fact.** Given a finite set $A$ of pointwise fluents and an interval $I$,

$I$ is $A$-segmentable $\iff$ $I$ is $\varphi$-a.b. for all $\varphi \in A$. 
Alternations

When does an interval $I$ have a $\varphi$-segmentation?

A $(\varphi, n)$-alternation of $I$ is a string $t_1 \cdots t_n \in I^n$ s.t. for $1 \leq i < n$,

$$t_i < t_{i+1} \quad \text{and} \quad \{t_i\} \models \varphi \iff \{t_{i+1}\} \not\models \varphi$$

(e.g. $\{t_1\} \models \varphi$, $\{t_2\} \not\models \varphi$, $\{t_3\} \models \varphi$, ...).

$I$ is $\varphi$-alternation bounded (a.b.) if there is an integer $n > 0$ s.t.

no $(\varphi, n)$-alternation of $I$ exists.

**Fact.** For pointwise $\varphi$, $I$ has a $\varphi$-segmentation iff $I$ is $\varphi$-a.b.

Stepping up from a fluent $\varphi$ to a set $A$ of fluents, we call a segmentation an $A$-segmentation if it is a $\varphi$-segmentation for each $\varphi \in A$.

$I$ is $A$-segmentable if there is an $A$-segmentation of $I$.

**Fact.** Given a finite set $A$ of pointwise fluents and an interval $I$,

$I$ is $A$-segmentable $\iff$ $I$ is $\varphi$-a.b. for all $\varphi \in A$. 
Alternations

When does an interval $I$ have a $\varphi$-segmentation?

A $(\varphi, n)$-alternation of $I$ is a string $t_1 \cdots t_n \in l^n$ s.t. for $1 \leq i < n$,

$$t_i < t_{i+1} \quad \text{and} \quad \{t_i\} \models \varphi \iff \{t_{i+1}\} \not\models \varphi$$

(e.g. $\{t_1\} \models \varphi$, $\{t_2\} \not\models \varphi$, $\{t_3\} \models \varphi$, ...).

$I$ is $\varphi$-alternation bounded (a.b.) if there is an integer $n > 0$ s.t. no $(\varphi, n)$-alternation of $I$ exists.

Fact. For pointwise $\varphi$, $I$ has a $\varphi$-segmentation iff $I$ is $\varphi$-a.b.

Stepping up from a fluent $\varphi$ to a set $A$ of fluents, we call a segmentation an $A$-segmentation if it is a $\varphi$-segmentation for each $\varphi \in A$.

$I$ is $A$-segmentable if there is an $A$-segmentation of $I$.

Fact. Given a finite set $A$ of pointwise fluents and an interval $I$,

$I$ is $A$-segmentable $\iff$ $I$ is $\varphi$-a.b. for all $\varphi \in A$. 


From $A$-segmentations to strings in $(2^A)^+$

The $A$-diagram of a seg $I_1 \cdots I_n$ is the string

$$\Delta_A(I_1 \cdots I_n) := \{ \varphi \in A \mid I_1 \models \varphi \} \cdots \{ \varphi \in A \mid I_n \models \varphi \}.$$  

**Fact.** If an interval $I$ is $A$-segmentable, there is a unique string $s^I_A$ s.t. for all $A$-segmentations $I_1 \cdots I_n$ of $I$,

$$bc(\Delta_A(I_1 \cdots I_n)) = s^I_A.$$  

Moreover, $s^I_A$ is the $A$-diagram of the shortest $A$-segmentation of $I$.

Recall: an $A$-segmentation $I_1 \cdots I_n$ tracks each $\varphi \in A$ — for all $I \subseteq \bigcup_{i=1}^n I_i$,

$$I \models \varphi \iff I \subseteq \bigcup \{ I_i \mid 1 \leq i \leq n \text{ and } I_i \models \varphi \}.$$  

Block compression $bc$ implements the Aristotelian insight

*No time without change.*

More tomorrow.
From $A$-segmentations to strings in $(2^A)^+$

The $A$-diagram of a seg $I_1 \cdots I_n$ is the string

$$\Delta_A(I_1 \cdots I_n) := \{ \varphi \in A \mid I_1 \models \varphi \} \cdots \{ \varphi \in A \mid I_n \models \varphi \}.$$ 

**Fact.** If an interval $I$ is $A$-segmentable, there is a unique string $s^I_A$ s.t. for all $A$-segmentations $I_1 \cdots I_n$ of $I$,

$$bc(\Delta_A(I_1 \cdots I_n)) = s^I_A.$$ 

Moreover, $s^I_A$ is the $A$-diagram of the shortest $A$-segmentation of $I$.

Recall: an $A$-segmentation $I_1 \cdots I_n$ tracks each $\varphi \in A$ — for all $I \subseteq \bigcup_{i=1}^n I_i$,

$$I \models \varphi \iff I \subseteq \bigcup \{ I_i \mid 1 \leq i \leq n \text{ and } I_i \models \varphi \}.$$ 

Block compression $bc$ implements the Aristotelian insight

*No time without change.*

More tomorrow.
From $A$-segmentations to strings in $(2^A)^+$

The $A$-diagram of a seg $l_1 \cdots l_n$ is the string

$$
\Delta_A(l_1 \cdots l_n) := \{ \varphi \in A \mid l_1 \models \varphi \} \cdots \{ \varphi \in A \mid l_n \models \varphi \}.
$$

**Fact.** If an interval $I$ is $A$-segmentable, there is a unique string $s^I_A$ s.t. for all $A$-segmentations $l_1 \cdots l_n$ of $I$,

$$
bc(\Delta_A(l_1 \cdots l_n)) = s^I_A.
$$

Moreover, $s^I_A$ is the $A$-diagram of the shortest $A$-segmentation of $I$.

Recall: an $A$-segmentation $l_1 \cdots l_n$ tracks each $\varphi \in A$ — for all $I \subseteq \bigcup_{i=1}^n l_i$,

$$
I \models \varphi \iff I \subseteq \bigcup \{ l_i \mid 1 \leq i \leq n \text{ and } l_i \models \varphi \}.
$$

Block compression $bc$ implements the Aristotelian insight

*No time without change.*

More tomorrow.
From $A$-segmentations to strings in $(2^A)^+$

The $A$-diagram of a seg $l_1 \cdots l_n$ is the string

$$\Delta_A(l_1 \cdots l_n) := \{ \varphi \in A \mid l_1 \models \varphi \} \cdots \{ \varphi \in A \mid l_n \models \varphi \}.$$

**Fact.** If an interval $I$ is $A$-segmentable, there is a unique string $s^l_A$ s.t. for all $A$-segmentations $l_1 \cdots l_n$ of $I$,

$$bc(\Delta_A(l_1 \cdots l_n)) = s^l_A.$$

Moreover, $s^l_A$ is the $A$-diagram of the shortest $A$-segmentation of $I$.

Recall: an $A$-segmentation $l_1 \cdots l_n$ tracks each $\varphi \in A$ — for all $I \subseteq \bigcup_{i=1}^n l_i$,

$$I \models \varphi \iff I \subseteq \bigcup \{ l_i \mid 1 \leq i \leq n \text{ and } l_i \models \varphi \}.$$

Block compression $bc$ implements the Aristotelian insight

*No time without change.*

More tomorrow.