Projecting temporal properties, events & actions

Tim Fernando

Gothenburg, 25 May 2019 (IWCS)

Projecting temporal properties, events & actions

Tim Fernando

Gothenburg, 25 May 2019 (IWCS)

- Labels in records and record-types

Projecting temporal properties, events & actions

Tim Fernando

Gothenburg, 25 May 2019 (IWCS)

- Labels in records and record-types
- Statives (Dowty aspect hypothesis)

Projecting temporal properties, events & actions

Tim Fernando

Gothenburg, 25 May 2019 (IWCS)

- Labels in records and record-types
- Statives (Dowty aspect hypothesis)
- Non-statives & Aktionsart (Moens & Steedman)

Projecting temporal properties, events & actions

Tim Fernando

Gothenburg, 25 May 2019 (IWCS)

- Labels in records and record-types
- Statives (Dowty aspect hypothesis)
- Non-statives & Aktionsart (Moens & Steedman $\rightsquigarrow$ episodes)
  Epilog (Schubert)
- Finite-state methods

Projecting temporal properties, events & actions

Tim Fernando

Gothenburg, 25 May 2019 (IWCS)

- Labels in records and record-types
- Statives (Dowty aspect hypothesis)
- Non-statives & Aktionsart (Moens & Steedman $\rightsquigarrow$ episodes)
  Epilog (Schubert)
- Finite-state methods
  – MSO under projections
  Allen interval networks and beyond

Timeline $\mathbb{R}$ (DRT, Kamp & Reyle)
unbounded linear order (Allen & Ferguson)

Finitization

Timeline $\mathbb{R}$ (DRT, Kamp & Reyle)
unbounded linear order (Allen & Ferguson)

$$X = \bigcup \mathit{Fin}(X) \text{ where } \mathit{Fin}(X) := \{A \subseteq X \mid A \text{ is finite}\}$$

Finitization

Timeline $\mathbb{R}$ (DRT, Kamp & Reyle)
unbounded linear order (Allen & Ferguson)

$$X = \bigcup Fin(X) \text{ where } Fin(X) := \{A \subseteq X \mid A \text{ is finite}\}$$

$$(X, <) \cong \lim_{\leftarrow} \{<_A\}_{A \in Fin(X)} \qquad \text{projections}$$

$$<_{\{a_1, \ldots, a_n\}} \quad \text{as string} \quad a_1 \cdots a_n \quad \text{where } a_1 < \cdots < a_n$$

Finitization

Timeline $\mathbb{R}$ (DRT, Kamp & Reyle)
unbounded linear order (Allen & Ferguson)

$$X = \bigcup Fin(X) \text{ where } Fin(X) := \{A \subseteq X \mid A \text{ is finite}\}$$

$$(X, <) \;\cong\; \lim_{\leftarrow} \{<_A\}_{A \in Fin(X)} \qquad \text{projections}$$

$$<_{\{a_1,\ldots,a_n\}} \;\text{ as string }\; a_1 \cdots a_n \;\text{ where } a_1 < \cdots < a_n$$

Trakhtenbrot's theorem *It is undecidable whether a first-order sentence with a binary relation has a finite model.*

Finitization & computability

Timeline $\mathbb{R}$ (DRT, Kamp & Reyle)
unbounded linear order (Allen & Ferguson)

$$X = \bigcup Fin(X) \text{ where } Fin(X) := \{A \subseteq X \mid A \text{ is finite}\}$$

$$(X, <) \;\cong\; \lim_{\leftarrow} \{<_A\}_{A \in Fin(X)} \qquad \text{projections}$$

$$<_{\{a_1,\ldots,a_n\}} \quad \text{as string} \quad a_1 \cdots a_n \quad \text{where } a_1 < \cdots < a_n$$

Trakhtenbrot's theorem *It is undecidable whether a first-order sentence with a binary relation has a finite model.*

Büchi-Elgot-Trakhtenbrot theorem (MSO = Reg)
For any finite set A and set L of strings over A,
an MSO_A-sentence defines L iff a finite automaton accepts L.

Aktionsart

VENDLER

STATE	ACHIEVEMENT	ACCOMPLISHMENT
		ACTIVITY

Aktionsart

VENDLER, *Moens & Steedman* (Comrie)

	atomic	*extended*
+*conseq*	ACHIEVEMENT *culmination*	ACCOMPLISHMENT *culminated process*
STATE		
−*conseq*	(semelfactive) *point*	ACTIVITY *process*

Aktionsart in strings

VENDLER, *Moens & Steedman* (Comrie)

	atomic	*extended*
$+$*conseq* STATE a	ACHIEVEMENT *culmination* $\boxed{\bar{a}}\boxed{a}$	ACCOMPLISHMENT *culminated process* $\boxed{\bar{a},\mathrm{ap}(f)}\boxed{\bar{a},\mathrm{ap}(f),\mathrm{ef}(f)}\boxed{\mathrm{ef}(f),a}$
$-$*conseq* f	(semelfactive) *point* $\boxed{\mathrm{ap}(f)}\boxed{\mathrm{ef}(f)}$	ACTIVITY *process* $\boxed{\mathrm{ap}(f)}\boxed{\mathrm{ap}(f),\mathrm{ef}(f)}\boxed{\mathrm{ef}(f)}$

Aktionsart in strings

VENDLER, *Moens & Steedman* (Comrie)

	atomic	*extended*
	ACHIEVEMENT	ACCOMPLISHMENT
$+$*conseq*	*culmination*	*culminated process*
STATE a	$\boxed{\bar{a}}\boxed{a}$	$\boxed{\bar{a},\text{ap}(f)}\boxed{\bar{a},\text{ap}(f),\text{ef}(f)}\boxed{\text{ef}(f), a}$
	(semelfactive)	ACTIVITY
$-$*conseq*	*point*	*process*
f	$\boxed{\text{ap}(f)}\boxed{\text{ef}(f)}$	$\boxed{\text{ap}(f)}\boxed{\text{ap}(f),\text{ef}(f)}\boxed{\text{ef}(f)}$

force f	state a	
`hit`	`break`	Fillmore
manner	result	Levin & Rappaport Hovav

Variations on a theme of $\mathrm{S^{c}h_{u}b_{e}r^{t}}$

	Davidson (event)	Barwise & Perry (situation)
Episodic Logic	characterize **	true-in *

Variations on a theme of $\mathrm{S^{c}h_{u}b_{e}r^{t}}$

	Davidson (event)	Barwise & Perry (situation)
Episodic Logic	characterize **	true-in *
Here	project wrt $\ell \subseteq A$	$\models$ (MSO)

Variations on a theme of $\mathrm{S^c h_u b_e r^t}$

	Davidson (event)	Barwise & Perry (situation)
Episodic Logic	characterize **	true-in *
Here	project wrt $\ell \subseteq A$	$\models_A$ (MSO)
	$\{\langle \ell_i, s_i \rangle\}_{i \in I}$	$\langle A, L \rangle$

Vary finite set A (ℓ)

Variations on a theme of $\mathrm{S^c h_u b_e r^t}$

	Davidson (event)	Barwise & Perry (situation)
Episodic Logic	characterize **	true-in *
Here	project wrt $\ell \subseteq A$	$\models_A$ (MSO)
	$\{\langle \ell_i, s_i \rangle\}_{i \in I}$	$\langle A, L \rangle$

Vary finite set A (ℓ) of

- temporal properties, stative and non-stative

Variations on a theme of $\mathrm{S^ch_ub_er^t}$

	Davidson (event)	Barwise & Perry (situation)
Episodic Logic	characterize **	true-in *
Here	project wrt $\ell \subseteq A$	$\models_A$ (MSO)
	$\{\langle \ell_i, s_i \rangle\}_{i \in I}$	$\langle A, L \rangle$

Vary finite set A (ℓ) of

- temporal properties, stative and non-stative

- variables as in Constraint Satisfaction Problem *Var, Dom, Con*
$\approx$ *institution* (Goguen & Burstall) *Sign, Mod, sen*

Variations on a theme of $\mathrm{S^{c}h_{u}b_{e}r^{t}}$

	Davidson (event)	Barwise & Perry (situation)
Episodic Logic	characterize **	true-in *
Here	project wrt $\ell \subseteq A$	$\models_A$ (MSO)
	$\{\langle \ell_i, s_i \rangle\}_{i \in I}$	$\langle A, L \rangle$

Vary finite set A (ℓ) of

- temporal properties, stative and non-stative

- variables as in Constraint Satisfaction Problem *Var, Dom, Con*
  $\approx$ *institution* (Goguen & Burstall) *Sign, Mod, sen*

- random variables, or vertices in graphical model (cond independ)
  subset $\ell_i \approx$ clique in Markov network

Variations on a theme of $\mathrm{S}^{\mathrm{c}}\mathrm{h}_{\mathrm{u}}\mathrm{b}_{\mathrm{e}}\mathrm{r}^{\mathrm{t}}$

	Davidson (event)	Barwise & Perry (situation)
Episodic Logic	characterize **	true-in *
Here	project wrt $\ell \subseteq A$	$\models_A$ (MSO)
	$\{\langle \ell_i, s_i \rangle\}_{i \in I}$	$\langle A, L \rangle$

Vary finite set A (ℓ) of

- temporal properties, stative and non-stative

- variables as in Constraint Satisfaction Problem *Var, Dom, Con*
  $\approx$ *institution* (Goguen & Burstall) *Sign, Mod, sen*

- random variables, or vertices in graphical model (cond independ)
  subset $\ell_i \approx$ clique in Markov network

> ... *causally or otherwise contingently related sequences of events, which we might call* **episodes**
>
> – Moens & Steedman

Reducts

ℓ-reduct $\rho_\ell(s)$ of s sees only symbols in ℓ

$$\rho_\ell(\alpha_1 \cdots \alpha_n) \;:=\; (\alpha_1 \cap \ell) \cdots (\alpha_n \cap \ell)$$

$$\rho_{\{a,a'\}}(\boxed{}\boxed{a}\boxed{a, a'}\boxed{a, a', {\color{red}a''}}\boxed{a', {\color{red}a''}}\boxed{{\color{red}a''}}) \;=\; \boxed{}\boxed{a}\boxed{a, a'}\boxed{a, a'}\boxed{a'}\boxed{}$$

Reducts & the border translation

ℓ-reduct $\rho_\ell(s)$ of s sees only symbols in ℓ

$$\rho_\ell(\alpha_1 \cdots \alpha_n) \;:=\; (\alpha_1 \cap \ell) \cdots (\alpha_n \cap \ell)$$

$$\rho_{\{a,a'\}}\left(\begin{array}{|c|c|c|c|c|c|}\hline \; & a & a, a' & a, a', a'' & a', a'' & a'' \\ \hline\end{array}\right) \;=\; \begin{array}{|c|c|c|c|c|c|}\hline \; & a & a, a' & a, a' & a' & \; \\ \hline\end{array}$$

$$b\left(\begin{array}{|c|c|c|c|c|}\hline \; & a & a, a' & a, a' & a' \\ \hline\end{array}\right) \;=\; \begin{array}{|c|c|c|c|c|}\hline l(a) & l(a') & \; & r(a) & r(a') \\ \hline\end{array}$$

$b : (2^A)^* \rightarrow (2^{A_\bullet})^*$

$$A_\bullet \;:=\; \{l(a) \mid a \in A\} \cup \{r(a) \mid a \in A\}$$

Reducts & the border translation

ℓ-reduct $\rho_\ell(s)$ of s sees only symbols in ℓ

$$\rho_\ell(\alpha_1 \cdots \alpha_n) \ := \ (\alpha_1 \cap \ell) \cdots (\alpha_n \cap \ell)$$

$$\rho_{\{a,a'\}}\left(\begin{array}{|c|c|c|c|c|c|}\hline \ & a & a, a' & a, a', a'' & a', a'' & a'' \\ \hline\end{array}\right) \ = \ \begin{array}{|c|c|c|c|c|c|}\hline \ & a & a, a' & a, a' & a' & \ \\ \hline\end{array}$$

$$b\left(\begin{array}{|c|c|c|c|c|}\hline \ & a & a, a' & a, a' & a' \\ \hline\end{array}\right) \ = \ \begin{array}{|c|c|c|c|c|}\hline l(a) & l(a') & \ & r(a) & r(a') \\ \hline\end{array}$$

$b : (2^A)^* \to (2^{A_\bullet})^*, \ \alpha_1 \cdots \alpha_n \mapsto \beta_1 \cdots \beta_n$

$$A_\bullet \ := \{l(a) \mid a \in A\} \cup \{r(a) \mid a \in A\}$$

$$\beta_n \ := \ \{r(a) \mid a \in \alpha_n\}$$

Reducts & the border translation

ℓ-reduct $\rho_\ell(s)$ of s sees only symbols in ℓ

$$\rho_\ell(\alpha_1 \cdots \alpha_n) \;:=\; (\alpha_1 \cap \ell) \cdots (\alpha_n \cap \ell)$$

$$\rho_{\{a,a'\}}(\boxed{}\boxed{a}\boxed{a, a'}\boxed{a, a', a''}\boxed{a', a''}\boxed{a''}) \;=\; \boxed{}\boxed{a}\boxed{a, a'}\boxed{a, a'}\boxed{a'}\boxed{}$$

$$b(\boxed{}\boxed{a}\boxed{a, a'}\boxed{a, a'}\boxed{a'}) \;=\; \boxed{l(a)}\boxed{l(a')}\boxed{}\boxed{r(a)}\boxed{r(a')}$$

$b : (2^A)^* \to (2^{A_\bullet})^*,\; \alpha_1 \cdots \alpha_n \mapsto \beta_1 \cdots \beta_n$

$$A_\bullet \;:=\; \{l(a) \mid a \in A\} \cup \{r(a) \mid a \in A\}$$

$$\beta_n \;:=\; \{r(a) \mid a \in \alpha_n\}$$

$$\beta_i \;:=\; \{l(a) \mid a \in \alpha_{i+1} - \alpha_i\} \cup \{r(a) \mid a \in \alpha_i - \alpha_{i+1}\} \text{ for } i < n$$

Reducts & the border translation

ℓ-reduct $\rho_\ell(s)$ of s sees only symbols in ℓ

$$\rho_\ell(\alpha_1 \cdots \alpha_n) \;:=\; (\alpha_1 \cap \ell) \cdots (\alpha_n \cap \ell)$$

$$\rho_{\{a,a'\}}(\boxed{}\boxed{a}\boxed{a, a'}\boxed{a, a', a''}\boxed{a', a''}\boxed{a''}) \;=\; \boxed{}\boxed{a}\boxed{a, a'}\boxed{a, a'}\boxed{a'}\boxed{}$$

$$b(\boxed{}\boxed{a}\boxed{a, a'}\boxed{a, a'}\boxed{a'}) \;=\; \boxed{l(a)}\boxed{l(a')}\boxed{}\boxed{r(a)}\boxed{r(a')}$$

$b : (2^A)^* \to (2^{A_\bullet})^*,\; \alpha_1 \cdots \alpha_n \mapsto \beta_1 \cdots \beta_n$

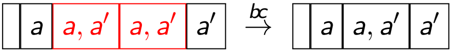
$$A_\bullet \;:=\; \{l(a) \mid a \in A\} \cup \{r(a) \mid a \in A\}$$

$$\beta_n \;:=\; \{r(a) \mid a \in \alpha_n\}$$

$$\beta_i \;:=\; \{l(a) \mid a \in \alpha_{i+1} - \alpha_i\} \cup \{r(a) \mid a \in \alpha_i - \alpha_{i+1}\} \;\text{ for } i < n$$

$$P_{l(a)}(x) \;\equiv\; \neg P_a(x) \wedge (\exists y)(xSy \wedge P_a(y))$$

Compression



no stutters

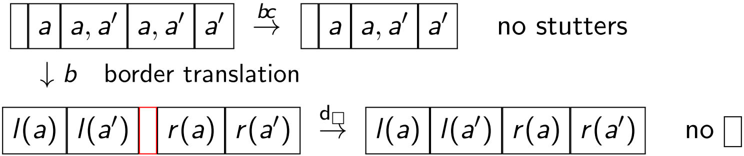
Compression

	a	a, a'	a, a'	a'	$\xrightarrow{bc}$		a	a, a'	a'		no stutters

$\downarrow b$ border translation

$l(a)$	$l(a')$		$r(a)$	$r(a')$

Compression two ways



$$\mathrm{d}_{\square}(s) \; := s \text{ without } \square$$

Compression two ways & projection

$$\boxed{}\boxed{a}\boxed{a,a'}\boxed{a,a'}\boxed{a'} \overset{bc}{\to} \boxed{}\boxed{a}\boxed{a,a'}\boxed{a'} \qquad \text{no stutters}$$

$\downarrow b$ border translation

$$\boxed{l(a)}\boxed{l(a')}\boxed{}\boxed{r(a)}\boxed{r(a')} \overset{\mathrm{d}_\Box}{\to} \boxed{l(a)}\boxed{l(a')}\boxed{r(a)}\boxed{r(a')} \qquad \text{no } \Box$$

$$\mathrm{d}_\Box(s) := s \text{ without } \Box$$

$$d_\ell(s) := \mathrm{d}_\Box(\rho_\ell(s))$$

s *projects to* s' if $s' = d_{voc(s')}(s)$ where

$$voc(\alpha_1 \cdots \alpha_n) := \bigcup_{i=1}^{n} \alpha_i.$$

a is an *s-interval* if $b(s)$ projects to $\boxed{l(a)}\boxed{r(a)}$

Leibniz's law: identity of indiscernibles

$$x \neq y \ \supset \ (\exists P)\neg(P(x) \equiv P(y)) \qquad \text{(LL)}$$

Leibniz's law: identity of indiscernibles

$$x \neq y \;\supset\; (\exists P)\neg(P(x) \equiv P(y)) \tag{LL}$$

replace $\neq$ by adjacency S

$$x\textcolor{red}{S}y \;\supset\; x \not\equiv_A y \tag{LL$_A$}$$

Leibniz's law: identity of indiscernibles

$$x \neq y \;\supset\; (\exists P)\neg(P(x) \equiv P(y)) \qquad \text{(LL)}$$

replace $\neq$ by adjacency S

$$xSy \;\supset\; x \not\equiv_{A} y \qquad (\mathrm{LL}_A)$$

and take P from a finite set A

$$\begin{aligned} x \not\equiv_A y \;&:=\; \bigvee_{a \in A} \neg(P_a(x) \equiv P_a(y)) \\ &\equiv \bigvee_{a \in A} \underbrace{(\neg P_a(x) \wedge P_a(y))}_{P_{l(a)}(x)} \vee \underbrace{(P_a(x) \wedge \neg P_a(y))}_{P_{r(a)}(x)} \end{aligned}$$

Leibniz's law: identity of indiscernibles relativized

$$x \neq y \ \supset \ (\exists P)\neg(P(x) \equiv P(y)) \qquad \text{(LL)}$$

replace $\neq$ by adjacency S "time steps$_S$ only with change$_A$"

$$xSy \ \supset \ x \not\equiv_A y \qquad (\text{LL}_A)$$

and take P from a finite set A

$$x \not\equiv_A y \ := \ \bigvee_{a\in A} \neg(P_a(x) \equiv P_a(y)) \qquad bc$$

$$\equiv \ \bigvee_{a\in A} \underbrace{(\neg P_a(x) \wedge P_a(y))}_{P_{l(a)}(x)} \vee \underbrace{(P_a(x) \wedge \neg P_a(y))}_{P_{r(a)}(x)} \qquad d_\square$$

Projecting

$$\mathcal{L}_A := \{\mathrm{d}_\square(s) \mid s \in (2^A)^+\} \qquad \text{Schwer } S\text{-words}$$

$$\mathcal{L}_A(s) := \{s' \in \mathcal{L}_A \mid s' \text{ projects to } s\}$$

Projecting more than once

$$\mathcal{L}_A \;:=\; \{\mathrm{d}_\Box(s) \mid s \in (2^A)^+\} \qquad \text{Schwer } S\text{-words}$$

$$\mathcal{L}_A(s) \;:=\; \{s' \in \mathcal{L}_A \mid s' \text{ projects to } s\}$$

$$\mathcal{L}_A(\boxed{l(a)}\boxed{r(a)}) \cap \mathcal{L}_A(\boxed{l(a')}\boxed{r(a')}) = \overbrace{\mathrm{Allen}(a, a')}^{\text{13 strings}} \quad \text{for } A = \{a, a'\}.$$

Projecting more than once

$$\mathcal{L}_A \;:=\; \{\mathrm{d}_\square(s) \mid s \in (2^A)^+\} \qquad \text{Schwer } S\text{-words}$$

$$\mathcal{L}_A(s) \;:=\; \{s' \in \mathcal{L}_A \mid s' \text{ projects to } s\}$$

$$\mathcal{L}_A(\boxed{l(a)}\boxed{r(a)}) \cap \mathcal{L}_A(\boxed{l(a')}\boxed{r(a')}) = \underbrace{\text{Allen}(a, a')}_{\text{13 strings}} \quad \text{for } A = \{a, a'\}.$$

Define superposition $s\&s'$ such that

$$\mathcal{L}_A(s) \cap \mathcal{L}_A(s') \;=\; \mathcal{L}_A(s\&s')$$

Projecting more than once

$$\mathcal{L}_A \; := \; \{\mathrm{d}_\square(s) \mid s \in (2^A)^+\} \qquad \text{Schwer } S\text{-words}$$

$$\mathcal{L}_A(s) \; := \; \{s' \in \mathcal{L}_A \mid s' \text{ projects to } s\}$$

$$\mathcal{L}_A(\boxed{l(a)}\boxed{r(a)}) \cap \mathcal{L}_A(\boxed{l(a')}\boxed{r(a')}) = \overbrace{\mathrm{Allen}(a, a')}^{\text{13 strings}} \quad \text{for } A = \{a, a'\}.$$

Define superposition $s\&s'$ such that

$$\mathcal{L}_A(s) \cap \mathcal{L}_A(s') \; = \; \mathcal{L}_A(s\&s') \;\; \text{where}$$

$$\mathcal{L}_A(L) \; := \; \bigcup_{s \in L} \mathcal{L}_A(s)$$

Allen relations projected

$$s \models aRa' \iff b(s) \text{ projects to } \mathfrak{s}_R(a, a')$$

R	aRa'	$\mathfrak{s}_R(a, a')$	R^{-1}	$\mathfrak{s}_{R^{-1}}(a, a')$
$<$	a before a'	$\boxed{l(a)}\boxed{r(a)}\boxed{l(a')}\boxed{r(a')}$	$>$	$\boxed{l(a')}\boxed{r(a')}\boxed{l(a)}\boxed{r(a)}$
m	a meets a'	$\boxed{l(a)}\boxed{r(a), l(a')}\boxed{r(a')}$	mi	$\boxed{l(a')}\boxed{r(a'), l(a)}\boxed{r(a)}$
o	a overlaps a'	$\boxed{l(a)}\boxed{l(a')}\boxed{r(a)}\boxed{r(a')}$	oi	$\boxed{l(a')}\boxed{l(a)}\boxed{r(a')}\boxed{r(a)}$
s	a starts a'	$\boxed{l(a), l(a')}\boxed{r(a)}\boxed{r(a')}$	si	$\boxed{l(a), l(a')}\boxed{r(a')}\boxed{r(a)}$
d	a during a'	$\boxed{l(a')}\boxed{l(a)}\boxed{r(a)}\boxed{r(a')}$	di	$\boxed{l(a)}\boxed{l(a')}\boxed{r(a')}\boxed{r(a)}$
f	a finishes a'	$\boxed{l(a')}\boxed{l(a)}\boxed{r(a), r(a')}$	fi	$\boxed{l(a)}\boxed{l(a')}\boxed{r(a), r(a')}$
$=$	a equal a'	$\boxed{l(a), l(a')}\boxed{r(a), r(a')}$	$=$	

Allen(a, a')

Allen relations projected

$$s \models aRa' \iff b(s) \text{ projects to } \mathfrak{s}_R(a, a')$$

R	aRa'	$\mathfrak{s}_R(a, a')$	R^{-1}	$\mathfrak{s}_{R^{-1}}(a, a')$
$<$	a before a'	$l(a) \mid r(a) \mid l(a') \mid r(a')$	$>$	$l(a') \mid r(a') \mid l(a) \mid r(a)$
m	a meets a'	$l(a) \mid r(a), l(a') \mid r(a')$	mi	$l(a') \mid r(a'), l(a) \mid r(a)$
o	a overlaps a'	$l(a) \mid l(a') \mid r(a) \mid r(a')$	oi	$l(a') \mid l(a) \mid r(a') \mid r(a)$
s	a starts a'	$l(a), l(a') \mid r(a) \mid r(a')$	si	$l(a), l(a') \mid r(a') \mid r(a)$
d	a during a'	$l(a') \mid l(a) \mid r(a) \mid r(a')$	di	$l(a) \mid l(a') \mid r(a') \mid r(a)$
f	a finishes a'	$l(a') \mid l(a) \mid r(a), r(a')$	fi	$l(a) \mid l(a') \mid r(a), r(a')$
$=$	a equal a'	$l(a), l(a') \mid r(a), r(a')$	$=$	

$\text{Allen}(a, a')$

From 2 intervals to 3

$$\frac{a < a' \qquad a' < a''}{a < a''}$$

	$<$	o	d	$\cdots$
$<$	$<$	$<$	$<$ d m o s	$\cdots$
o	$<$	$<$ m o	d o s	$\cdots$
d	$<$	$<$ d m o s	d	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$

From 2 intervals to 3

$$\frac{a < a' \qquad a' < a''}{a < a''}$$

	$<$	o	d	$\cdots$
$<$	$<$	$<$	$<$ d m o s	$\cdots$
o	$<$	$<$ m o	d o s	$\cdots$
d	$<$	$<$ d m o s	d	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$

$$\mathfrak{s}_<(a, a') \;\&\; \mathfrak{s}_<(a', a'') \;=\; \boxed{l(a)}\boxed{r(a)}\boxed{l(a')}\boxed{r(a')}\boxed{l(a'')}\boxed{r(a'')}$$

From 2 intervals to 3

$$\frac{a < a' \qquad a' < a''}{a < a''} \qquad \frac{a \text{ o } a' \qquad a' \text{ d } a''}{a \ \{\text{d,o,s}\} \ a''}$$

	$<$	o	d	$\cdots$
$<$	$<$	$<$	$<$ d m o s	$\cdots$
o	$<$	$<$ m o	d o s	$\cdots$
d	$<$	$<$ d m o s	d	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$

$\mathfrak{s}_<(a,a')$ & $\mathfrak{s}_<(a',a'')$ = | $l(a)$ | $r(a)$ | $l(a')$ | $r(a')$ | $l(a'')$ | $r(a'')$ |

$\mathfrak{s}_\text{o}(a,a')$ & $\mathfrak{s}_\text{d}(a',a'')$ = | $l(a'')$ | $l(a)$ | $l(a')$ | $r(a)$ | $r(a')$ | $r(a'')$ | — a d a''

\+ | $l(a)$ | $l(a'')$ | $l(a')$ | $r(a)$ | $r(a')$ | $r(a'')$ | — a o a''

\+ | $l(a), l(a'')$ | $l(a')$ | $r(a)$ | $r(a')$ | $r(a'')$ | — a s a''

Superposition

$$\frac{}{\&^\circ(\epsilon, \epsilon, \epsilon)} \qquad \frac{\&^\circ(s, s', s'')}{\&^\circ(\alpha s, \alpha' s, (\alpha \cup \alpha') s'')}$$

Superposition

$$\frac{}{\&^\circ(\epsilon, \epsilon, \epsilon)} \quad \frac{\&^\circ(s, s', s'')}{\&^\circ(\alpha s, \alpha' s, (\alpha \cup \alpha') s'')} \quad \frac{\&^\circ(s, s', s'')}{\&^\circ(\alpha s, s', \alpha s'')} \quad \frac{\&^\circ(s, s', s'')}{\&^\circ(s, \alpha' s', \alpha' s'')}$$

Superposition

$$\frac{}{\&^\circ(\epsilon,\epsilon,\epsilon)} \quad \frac{\&^\circ(s,s',s'')}{\&^\circ(\alpha s,\alpha' s,(\alpha\cup\alpha')s'')} \quad \frac{\&^\circ(s,s',s'')}{\&^\circ(\alpha s,s',\alpha s'')} \quad \frac{\&^\circ(s,s',s'')}{\&^\circ(s,\alpha' s',\alpha' s'')}$$

Constrain through A, A'

$$\frac{\&_{A,A'}(s,s',s'') \qquad \alpha\cap A'\subseteq\alpha' \qquad \alpha'\cap A\subseteq\alpha}{\&_{A,A'}(\alpha s,\alpha' s,(\alpha\cup\alpha')s'')}$$

$$\frac{\&_{A,A'}(s,s',s'') \quad \alpha\cap A'=\emptyset}{\&_{A,A'}(\alpha s,s',\alpha s'')} \qquad \frac{\&_{A,A'}(s,s',s'') \quad \alpha'\cap A=\emptyset}{\&_{A,A'}(s,\alpha' s',\alpha' s'')}$$

Superposition

$$\frac{}{\&^\circ(\epsilon,\epsilon,\epsilon)} \quad \frac{\&^\circ(s,s',s'')}{\&^\circ(\alpha s,\alpha' s,(\alpha\cup\alpha')s'')} \quad \frac{\&^\circ(s,s',s'')}{\&^\circ(\alpha s,s',\alpha s'')} \quad \frac{\&^\circ(s,s',s'')}{\&^\circ(s,\alpha' s',\alpha' s'')}$$

Constrain through A, A'

$$\frac{\&_{A,A'}(s,s',s'') \qquad \alpha\cap A'\subseteq\alpha' \qquad \alpha'\cap A\subseteq\alpha}{\&_{A,A'}(\alpha s,\alpha' s,(\alpha\cup\alpha')s'')}$$

$$\frac{\&_{A,A'}(s,s',s'') \quad \alpha\cap A'=\emptyset}{\&_{A,A'}(\alpha s,s',\alpha s'')} \qquad \frac{\&_{A,A'}(s,s',s'') \quad \alpha'\cap A=\emptyset}{\&_{A,A'}(s,\alpha' s',\alpha' s'')}$$

$\&_{voc(s),voc(s')}(s,s',s'') \iff \&^\circ(s,s',s'')$ and s'' projects to s and s'

Episodes in Moens & Steedman

Rather than a homogeneous database of dated points or intervals, we should partition it into distinct sequences of causally or otherwise contingently related sequences of events, which we might call **episodes** [MS, p 26]

Episodes in Moens & Steedman as record types

> Rather than a homogeneous database of dated points or intervals, we should partition it into distinct sequences of causally or otherwise contingently related sequences of events, which we might call **episodes** [MS, p 26]

$$[\![\{\langle \ell_i, L_i \rangle\}_{i \in I}]\!]_A \;:=\; \{s \in \mathcal{L}_A \mid (\forall i \in I)\; d_{\ell_i}(s) \in L_i\}$$

Episodes in Moens & Steedman as record types

Rather than a homogeneous database of dated points or intervals, we should partition it into distinct sequences of causally or otherwise contingently related sequences of events, which we might call **episodes** [MS, p 26]

$$\begin{aligned}
[\![\{\langle \ell_i, L_i \rangle\}_{i \in I}]\!]_A &:= \{s \in \mathcal{L}_A \mid (\forall i \in I)\ d_{\ell_i}(s) \in L_i\} \\
&= \bigcap_{i \in I} \mathcal{L}_A(L_i) \quad \text{if } \ell_i = voc(s_i) \text{ for } i \in I
\end{aligned}$$

Episodes in Moens & Steedman as record types

> Rather than a homogeneous database of dated points or intervals, we should partition it into distinct sequences of causally or otherwise contingently related sequences of events, which we might call **episodes** [MS, p 26]

$$
\begin{aligned}
[\![\{\langle \ell_i, L_i \rangle\}_{i \in I}]\!]_A &:= \{s \in \mathcal{L}_A \mid (\forall i \in I)\ d_{\ell_i}(s) \in L_i\} \\
&= \bigcap_{i \in I} \mathcal{L}_A(L_i) \quad \text{if } \ell_i = voc(s_i) \text{ for } i \in I
\end{aligned}
$$

E.g. interval network arc labeling $\lambda : (Ivl \times Ivl) \rightarrow 2^{AR}$

$$
\text{for } \ell_i = \{a, a'\},\ L_i = \{\mathfrak{s}_R(a, a') \mid R \in \lambda(a, a')\}
$$

Episodes in Moens & Steedman as record types

> Rather than a homogeneous database of dated points or intervals, we should partition it into distinct sequences of causally or otherwise contingently related sequences of events, which we might call **episodes** [MS, p 26]

$$
\begin{aligned}
[\![\{\langle \ell_i, L_i\rangle\}_{i\in I}]\!]_A &:= \{s \in \mathcal{L}_A \mid (\forall i \in I)\ d_{\ell_i}(s) \in L_i\} \\
&= \bigcap_{i\in I} \mathcal{L}_A(L_i) \quad \text{if } \ell_i = voc(s_i) \text{ for } i \in I
\end{aligned}
$$

E.g. interval network arc labeling $\lambda : (Ivl \times Ivl) \rightarrow 2^{AR}$

$$\text{for } \ell_i = \{a, a'\},\ L_i = \{\mathfrak{s}_R(a, a') \mid R \in \lambda(a, a')\}$$

determinate labeling $L_i = \{s_i\}$ for record

$$[\![\{\langle \ell_i, s_i\rangle\}_{i\in I}]\!]_A := \{s \in \mathcal{L}_A \mid (\forall i \in I)\ d_{\ell_i}(s) = s_i\}$$

J.A. Wheeler: *it from bit*

every *it* – every particle, every field of force, even the space-time continuum itself – derives its function, its meaning, its very existence entirely – even if in some contexts indirectly – from the apparatus-elicited answers to yes-or-no questions, binary choices, bits.

. . .

all things physical are information-theoretic in origin

J.A. Wheeler: *it from bit*

every *it* – every particle, every field of force, even the space-time continuum itself – derives its function, its meaning, its very existence entirely – even if in some contexts indirectly – from the apparatus-elicited answers to yes-or-no questions, binary choices, bits.

…

all things physical are information-theoretic in origin

$$\begin{array}{rcl} it & \approx & \text{value/string } v_i \text{ (or type/language } T_i\text{)} \\ & & \text{linked by } \ell_i \text{ in records (or record types)} \end{array}$$

J.A. Wheeler: *it from bit*

every *it* – every particle, every field of force, even the space-time continuum itself – derives its function, its meaning, its very existence entirely – even if in some contexts indirectly – from the apparatus-elicited answers to yes-or-no questions, binary choices, bits.

. . .

all things physical are information-theoretic in origin

it $\approx$ value/string v_i (or type/language T_i)
linked by ℓ_i in records (or record types)

MSO:

J.A. Wheeler: *it from bit*

every *it* – every particle, every field of force, even the space-time continuum itself – derives its function, its meaning, its very existence entirely – even if in some contexts indirectly – from the apparatus-elicited answers to yes-or-no questions, binary choices, bits.

...

all things physical are information-theoretic in origin

it $\approx$ value/string v_i (or type/language T_i)
linked by ℓ_i in records (or record types)

MSO: yes-or-no P_a-questions answered in S-steps

Thank You

☺