22 Jan 2005, Kyoto (Language under Uncertainty)

What is an inertia world?[a]

Tim.Fernando@tcd.ie

(1) I was giving my talk when ...

Imperfective Paradox (Dowty)

PROG(E) at I, w ~~imply~~ for some $I' \supseteq I$, E at I', w.

Replace w in rhs by (I, w)-*inertia world*.

Plan

Part 1.	What is inertia?	(worlds aside)
Part 2.	What is a world?	(events/temp log)
Part 3.	Putting 1 and 2 together	

[a] Revised: slide 16 below differs from the hand-out, taking up, as it does, three test cases Prof. Higginbotham described for the progressive.

Part 1. What is inertia?

Inertia is a property certain $\underbrace{\text{fluents}}$ have.

temporal formulas $\varphi, \ldots$

φ is *inertial* if it persists unless a force is applied on it.

E.g. φ = still-car (force = stop)

(2) Pat stopped the car before it hit the tree.

$\boxed{\overline{\text{still-car}}}\boxed{\text{still-car}}$??? $\boxed{\overline{\text{still-car}}}\cdots$

From (2), conclude

(†) the car did not hit the tree, unless some force intervened.

(3) But a bus behind it kept going.

Next: force-fluents and strings

Force and inertial flow

Given inertial φ, form fluent

$$\mathsf{F}\varphi \quad \approx \quad \text{`a force is applied on } \varphi\text{.'}$$

Fix a finite set Φ of fluents, and $\Sigma \subseteq \mathsf{Pow}(\Phi)$.

A string in Σ^* is φ-*inertially complete* (ic) if

whenever it says φ holds at i, it also says φ holds at $i+1$ or $\mathsf{F}\varphi$ holds at i

$$\text{(i)} \qquad \boxed{\varphi}\Box \;\Rightarrow\; \Box\boxed{\varphi} + \boxed{\mathsf{F}\varphi}\Box$$

and

whenever it says φ holds at $i+1$, it also says φ or $\mathsf{F}\varphi$ holds at i

$$\text{(ii)} \qquad \Box\boxed{\varphi} \;\Rightarrow\; \boxed{\varphi}\Box + \boxed{\mathsf{F}\varphi}\Box\,.$$

Inertial Hypothesis. Strings in NL interp are φ-ic, for each inertial φ — henceforth, ic.

Inertial completions and explanations

$\boxed{\varphi}\Box$ is not ic.

We can make $\boxed{\varphi}\Box$ ic two ways:

(a) let φ flow: add φ $\rightsquigarrow$ $\boxed{\varphi}\boxed{\varphi}$

- *inertial completion*

(b) block flow: add $\mathsf{F}\varphi$ $\rightsquigarrow$ $\boxed{\varphi, \mathsf{F}\varphi}\Box$

$$\mathsf{F}\varphi \quad \approx \quad \text{freeze } \varphi$$

- *inertial explanation*.

Back to (2): let sc = still-car

$$\boxed{\overline{\text{sc}}} \quad \boxed{\text{sc}} \quad \Box^n \quad \boxed{\overline{\text{sc}}}\cdots$$

$$\rightsquigarrow \underbrace{\boxed{\overline{\text{sc}}, \mathsf{F}\text{sc}}}_{\text{get Fsc from Pat-stop-car}} \quad \boxed{\text{sc}} \quad \boxed{\text{sc}}^n \quad \underbrace{\boxed{\overline{\text{sc}},\text{sc}}}_{\text{incoherent}}\cdots$$

assume $\mathsf{F}\varphi = \mathsf{F}\overline{\varphi}$

From strings to languages

$\boxed{\overline{\text{sc}},\text{Fsc}}\boxed{\text{sc}}$ is a very coarse rep of Pat-stop-car.

For simplicity, let's put Pat aside — and, for the moment, forces.

Generalizing from sc to φ, let

$$(\ddagger) \qquad \text{Become}_{telic}(\varphi) \;=\; \boxed{\overline{\varphi}}\boxed{\overline{\varphi},\varphi_{\uparrow}}^{*}\boxed{\varphi}$$

where (e.g.) given φ-degrees, $\varphi_{\uparrow}$ is

$$\exists d(\deg_{\varphi}(d) \wedge (\exists d' < d)\ \text{Previously } \deg_{\varphi}(d')).$$

Previously refers to previous position in string.

Kleene $\cdot^{*}$ in ($\ddagger$) allows for different granularities

$$\text{Become}_{telic}(\varphi) \;=\; \sum_{n \geq 0} \boxed{\overline{\varphi}}\boxed{\overline{\varphi},\varphi_{\uparrow}}^{n}\boxed{\varphi}$$

Let n in slide 4 vary (just as above) — step up from strings to languages.

In addition to temporal granularity, there's also the descriptive detail within a fixed snapshot.

Slide 5

information $\trianglerighteq$ information

Subsumption (explicit entailment) $\trianglerighteq$ on strings

$$\boxed{p,q} \;\trianglerighteq\; \boxed{p}$$

- generalize $\supseteq$ ('more is more')

$$\alpha_1 \cdots \alpha_n \;\trianglerighteq\; \beta_1 \cdots \beta_n \;\text{ iff }\; \alpha_i \supseteq \beta_i \text{ for } 1 \leq i \leq n$$

Over languages: 'less is more'

$$\boxed{p} \;\trianglerighteq\; \boxed{p} + \boxed{q}$$

- generalize $\subseteq$

$$L \;\trianglerighteq\; L' \quad \text{iff} \quad (\forall \mathsf{s} \in L)(\exists \mathsf{s}' \in L')\ \mathsf{s} \trianglerighteq \mathsf{s}'$$

N.B. $\trianglerighteq$ is reflexive and transitive.

L is *φ-telic* if $L \trianglerighteq \boxed{\overline{\varphi}}^{*}\boxed{\varphi}$.

φ is a *precondition for* L if $L \trianglerighteq \boxed{\varphi}\square^{*}$.

φ is a *postcondition for* L if $L \trianglerighteq \square^{*}\boxed{\varphi}$.

Slide 6

Superposition & (Vendler)

rain for two-days

$$\boxed{\text{rain, } 0(\tau)}\boxed{\text{rain}}^{+}\boxed{\text{rain, 2days}(\tau)}$$

$$= \boxed{\text{rain}}^{+} \;\&\; \underbrace{\boxed{0(\tau)}\square^{+}\boxed{\text{2days}(\tau)}}_{\text{two-days}}$$

$$L\&L' \;=\; \bigcup_{n \geq 1} \{(\alpha_1 \cup \alpha_1') \cdots (\alpha_n \cup \alpha_n') \mid \alpha_1 \cdots \alpha_n \in L \text{ and } \alpha_1' \cdots \alpha_n' \in L'\}\ .$$

$$\begin{array}{ccccc} \boxed{b}\boxed{\overline{b}}^{+} & \& & \boxed{\overline{e}}^{+}\boxed{e} & = & \boxed{b,\overline{e}}\boxed{\overline{b},\overline{e}}^{*}\boxed{\overline{b},e} \\ \text{activity} & \& & \text{achiev} & = & \text{accomplishment} \end{array}$$

Facts. (i) $L \trianglerighteq L'$ iff $L \subseteq L\&L'$

(ii) $L \trianglerighteq L'\&L''$ iff $L \trianglerighteq L'$ and $L \trianglerighteq L''$

(iii) $L\&L'$ is regular if L and L' are.

Slide 7

R marks a stage/position in L

(4) Pat left Dublin but is back (in Dublin).

(5) Pat has left Dublin ?but is back.

Aspect: Reichenbach *reference time* $R \in \Phi - \mathsf{Inr}$

$$\begin{array}{rcll} \textsc{Simp}(L,R) & = & L \;\&\; \square^{*}\boxed{R} & \%\ L \text{ is complete} \\ \textsc{Prog}(L,R) & = & L \;\&\; \square^{+}\boxed{R}\square^{+} & \%\ L \text{ is in progress} \\ \textsc{Perf}_{\circ}(L,R) & = & L\square^{*}\boxed{R} & \%\ L \text{ is history} \end{array}$$

E.g. For $L = \boxed{\text{rain, } 0(\tau)}\boxed{\text{rain}}^{+}\boxed{\text{rain, 2days}(\tau)}$

$$\boxed{\text{rain, } 0(\tau)}\boxed{\text{rain}}^{+}\boxed{\text{rain,2days}(\tau),R}$$

$$\boxed{\text{rain, } 0(\tau)}\boxed{\text{rain}}^{*}\boxed{\text{rain, } R}\boxed{\text{rain}}^{*}\boxed{\text{rain,2days}(\tau)}$$

$$\boxed{\text{rain, } 0(\tau)}\boxed{\text{rain}}^{+}\boxed{\text{rain,2days}(\tau)}\square^{*}\boxed{R}$$

If L is ic, so are $\textsc{Simp}(L,R)$ and $\textsc{Prog}(L,R)$.

But *not* necessarily $\textsc{Perf}_{\circ}(L,R)$.

Slide 8

R as a freezer

Let φ^{F} abbreviate '$\varphi, \mathsf{F}\varphi$'

Pat-leave-Dublin $\boxed{\mathrm{in}(p,d)^{\mathsf{F}}}\boxed{\overline{\mathrm{in}(p,d)}}$ (†)

PERF$_\circ$(PleaveD) $\boxed{\mathrm{in}(p,d)^{\mathsf{F}}}\boxed{\overline{\mathrm{in}(p,d)}}\square^*\boxed{R}$

$\overset{inr}{\rightsquigarrow}$ $\boxed{\mathrm{in}(p,d)^{\mathsf{F}}}\boxed{\overline{\mathrm{in}(p,d)}}\boxed{\overline{\mathrm{in}(p,d)}}^*\boxed{\overline{\mathrm{in}(p,d)}, R}$

PRESENT(PERF(Pat-leave-Dublin))

$\boxed{\mathrm{in}(p,d)^{\mathsf{F}}}\boxed{\overline{\mathrm{in}(p,d)}}^+\boxed{\overline{\mathrm{in}(p,d)}, R, S}$

speech time S

(5) is ok in reply to: Has Pat *ever* left Dublin?

Ever questions license *existential* readings.

Freeze $\overline{\mathrm{in}(p,d)}$ by adding $\mathsf{F}\overline{\mathrm{in}(p,d)}$ to (†).

Pat was in Dublin $\not\Rightarrow$ Pat is in Dublin

$\boxed{\mathrm{in}(p,d), R}\square^*\boxed{S}$ $\boxed{\mathrm{in}(p,d), R, S}$

Part 2. What is a world?

A world is the totality of states and events in it.

We will consider the case of linear temporal logic with discrete future operators, LTL.

In LTL, time is *discrete*: 'next moment' is well-defined.

Prior 1967

> the usefulness of systems of this sort does not depend on any serious metaphysical assumption that time *is* discrete; they are applicable in limited fields of discourse in which we are concerned with what happens in a sequence of discrete states.

E.g. φ-telicity depends on bounded granularity: we cannot pin down the precise moment of change over the real numbers.

LTL over infinite strings

Given a set P of *atomic propositions*,

a P-*world* is an infinite string

$$\sigma \quad = \quad \sigma(0)\ \sigma(1)\ \sigma(2)\cdots \ \in \mathsf{Pow}(P)^{\mathbb{N}}.$$

$$\begin{array}{rll} \sigma \models p & \text{iff} & p \in \sigma(0) \qquad (\text{for } p \in P) \\ \sigma \models \varphi \wedge \psi & \text{iff} & \sigma \models \varphi \text{ and } \sigma \models \psi \\ \sigma \models \varphi \vee \psi & \text{iff} & \sigma \models \varphi \text{ or } \sigma \models \psi \\ \sigma \models \mathit{next}(\varphi) & \text{iff} & \sigma^1 \models \varphi \end{array}$$

where $\sigma^k = \sigma(k)\ \sigma(k+1)\cdots \in \mathsf{Pow}(P)^{\mathbb{N}}$

$$\begin{array}{rll} \sigma \models \varphi\ \mathit{until}\ \psi & \text{iff} & (\exists n \geq 0)\ (\sigma^n \models \psi \text{ and} \\ & & \quad (\forall m < n)\ \sigma^m \models \varphi) \\ \sigma \models \varphi\ \mathit{releases}\ \psi & \text{iff} & (\forall n \geq 0)\ (\sigma^n \models \psi \text{ or} \\ & & \quad (\exists m < n)\ \sigma^m \models \varphi)\ . \end{array}$$

For $n \geq 1$, let $\sigma[n]$ be the string (in $\mathsf{Pow}(\Phi)^n$)

$$\boxed{\psi \mid \sigma \models \psi}\boxed{\psi \mid \sigma^1 \models \psi}\cdots\boxed{\psi \mid \sigma^{n-1} \models \psi}\ .$$

From stills to movies

$$\begin{array}{rcl} p & \rightsquigarrow & \boxed{p} \\ p \wedge q & \rightsquigarrow & \boxed{p, q} \\ p \vee q & \rightsquigarrow & \boxed{p} + \boxed{q} \\ \mathit{next}(p) & \rightsquigarrow & \square\boxed{p} \\ p\ \mathit{until}\ q & \rightsquigarrow & \boxed{p}^*\boxed{q} \end{array}$$

$\boxed{p}^*\boxed{q}$ portrays 'p *until* q' and $\square\boxed{p}\ldots$

$L \subseteq \mathsf{Pow}(\Phi)^*$ *portrays* φ if for all P-worlds σ,

$$\begin{array}{rll} \sigma \models \varphi & \text{iff} & (\exists \mathsf{s} \in L)\ \sigma[\mathrm{length}(\mathsf{s})] \trianglerighteq \mathsf{s} \\ & [\ \text{iff} & (\exists n \geq 0)(\exists \alpha_1 \cdots \alpha_n \in L) \\ & & (\forall i \in \{1,\ldots,n\})(\forall \psi \in \alpha_i)\ \sigma^{i-1} \models \psi\ ] \end{array}$$

— the P-worlds satisfying φ are the infinite $\trianglerighteq$-extensions of strings in L.

Instead of reducing φ to σ's such that $\sigma \models \varphi$, we can reduce φ to $L \subseteq \mathsf{Pow}(\Phi)^*$.

Strings as events

Every $\varphi \in \Phi$ is portrayed by $\boxed{\varphi}$.

But $\boxed{p}^*\boxed{q}$ provides better drawn strings $\boxed{p}^n\boxed{q}$ for ‘*p until q*’ than does $\boxed{p\ until\ q}$:

> $\boxed{p}^n\boxed{q}$ has a well-delineated temporal extent that $\boxed{p\ until\ q}$ does not.

Event as sequence of snapshots (e.g. Tenny 87)

Problem with

$$p\ releases\ q \quad \rightsquigarrow \quad \boxed{q}^*\boxed{p,q} + \underbrace{\boxed{q}^\infty}$$

$$\boxed{q}^n\boxed{\bot\ releases\ q}$$

where $\sigma \not\models \bot$.

Theorem. *For all $\varphi \in \Phi$ and $n \geq 1$, there is an $\mathcal{L}(\varphi) \subseteq \mathsf{Pow}(B(\varphi) \cap P)^n \mathsf{Pow}(B(\varphi))^*$ that portrays φ, where $B(\varphi)$ is a set of subformulas of φ that are, for ‘releases’-free φ, all atomic.*

& as ∧ and irregularity

$$\begin{aligned}
\mathcal{L}(p) &= \boxed{p}\Box^{n-1}\Box^* \\
\mathcal{L}(\varphi \wedge \psi) &= \mathcal{L}(\varphi)\ \&\ \mathcal{L}(\psi) \\
\mathcal{L}(\varphi \vee \psi) &= \mathcal{L}(\varphi) + \mathcal{L}(\psi) \\
\mathcal{L}(next(\varphi)) &= \Box\mathcal{L}(\varphi)
\end{aligned}$$

To see $\mathcal{L}(\varphi\ until\ \psi) \neq \mathcal{L}(\varphi)^*\mathcal{L}(\psi)$, note

$$\begin{aligned}
next(p)\ until\ p \quad &\rightsquigarrow \quad \boxed{next(p)}^*\boxed{p} \\
&= \boxed{p} + \boxed{next(p)}\boxed{p} + \cdots \\
&= \boxed{p} + \underbrace{\Box\boxed{p}} + \cdots \\
&\qquad\qquad \not\in (\Box\boxed{p})^*\boxed{p}
\end{aligned}$$

Consider $\hat{\varphi} = (p \wedge (\top\ until\ q))\ until\ r$

$$\hat{\varphi} \rightsquigarrow \boxed{p, \top\ until\ q}^*\boxed{r}$$

$$\mathcal{L}(\hat{\varphi}) \cap \boxed{p}^+\boxed{r}\boxed{q}^+ = \sum_{i \geq 1} \boxed{p}^i\boxed{r} \sum_{1 \leq j \leq i} \boxed{q}^j$$

- non-regular by Pumping Lemma argument.

Event-types as regular languages

For regularity, unwind incrementally [lazy evaln]

$$\begin{aligned}
\boxed{\varphi\ until\ \psi} &\quad\to\quad \boxed{\psi} + \boxed{\varphi}\boxed{\varphi\ until\ \psi} \\
\boxed{\varphi\ releases\ \psi} &\quad\to\quad \boxed{\varphi, \psi} + \boxed{\psi}\boxed{\varphi\ releases\ \psi}
\end{aligned}$$

Compare to LTL tableaux

See Fer4 (‘Events from temporal logic to …’)

(i) in slide 3 as ‘$p \supset (\mathsf{F}p\ releases\ p)$’

$$\mathsf{F}p\ releases\ p \quad \approx \quad \boxed{p}^*\boxed{p, \mathsf{F}p} + \boxed{p}^\infty$$

Facts. (a) Entailments (i) and (ii) in slide 3 define regular languages (FeNa).

(b) If L is regular, then so are its inertial completion and explanation (Fer3).

Call L *stative* if $L = ic(\Box^* L \Box^*)$. E.g. $\boxed{\varphi}^+$.

Part 3. Putting 1 and 2 together

If events are ic and worlds are sums of events, then worlds are ic.

From Fer2: R as realis marker [(c2)]

$$\text{PROG}(L) = \text{Prog}(L, R)_c$$

where $c : \Sigma^* \to \mathsf{Pow}(\Sigma^*)$ gives set $c(s)$ of *continutations of s from R* satisfying

(c1) $s \in c(s)$

(c2) $(\forall s' \in c(s))\ {s'}_R = s_R$ [same up to R]

$$s \in {L'}_c \quad \text{iff} \quad (\exists s' \in c(s))\ s' \in L'$$

(Hig1) Coin landing heads: $c(s) = \{s\}$ (Ockhamist)

(Hig2) Flying to NY or Algeria but NOT both $|c(s)| \geq 2$ with $\boxed{\text{in-NY,in-Algeria}} \not\in \Sigma$

(Hig3) Dorothy turning to sand, crossing the desert $L \trianglerighteq \Box^* \boxed{\overline{\text{sand}}, \text{in-des}}\boxed{\text{sand}, \overline{\text{in-des}}}$

Branching in BEFORE

(2)′ The car stopped before it hit the tree.

$\boxed{\overline{\mathrm{sc}}}^*\boxed{\overline{\mathrm{sc}}^{\mathsf{F}}}\boxed{\mathrm{sc}}$ $\quad$ $\boxed{\overline{\mathrm{sc}},\overline{\mathrm{ct}}}^*\boxed{\overline{\mathrm{sc}},\overline{\mathrm{ct}}^{\mathsf{F}}}\boxed{\mathrm{ct}}$

t_1 $\quad$ t_2

sc = still-car $\quad$ ct = car-tree-contact

BEFORE $t_1 < t_2$

$$\boxed{\overline{\mathrm{sc}}}^*\boxed{\overline{\mathrm{sc}}^{\mathsf{F}}}\ \boxed{\mathrm{sc}}$$

$$\downarrow^b$$

$$\boxed{\overline{\mathrm{sc}},\overline{\mathrm{ct}}}^*\boxed{\overline{\mathrm{sc}},\overline{\mathrm{ct}}^{\mathsf{F}}}\ \boxed{\mathrm{ct}} \quad = L_2$$

$$= \quad b(\boxed{\overline{\mathrm{sc}}}^*\boxed{\overline{\mathrm{sc}}^{\mathsf{F}}}, L_2)\ \boxed{\mathrm{sc}}$$

$$\overset{inr}{\rightsquigarrow} \quad b(\boxed{\overline{\mathrm{sc}},\overline{\mathrm{ct}}}^*\boxed{\overline{\mathrm{sc}}^{\mathsf{F}},\overline{\mathrm{ct}}}, L_2)\ \boxed{\mathrm{sc},\overline{\mathrm{ct}}}$$

(6) Had it not stopped, the car may have hit the tree.

(Heinämäki 1972, Beaver & Condoravdi 2003)

Slide 17

B(ranching)-strings and b-automata

$b(s, s_1)s_2$ says $\begin{cases} s_1 \text{ } may \text{ follow } s \\ s_2 \text{ follows } s \text{ [no qualification]} \end{cases}$

$b(s, s')$ weakens ss' — suggests an approach to the Imperfective paradox (Fer2).

Σ^b is the set of (non-empty) *b-strings* s over Σ

$$\mathsf{s} \quad :== \quad \alpha \mid \mathsf{ss'} \mid b(\mathsf{s}, \mathsf{s'}) \qquad [\alpha \in \Sigma]$$

A *finite b-automaton* = fa $(Q, \to, q_0, F)$ plus $\xrightarrow{b} \subseteq Q \times Q$ (realis and temporal distance)

Extend $\xrightarrow{\alpha} \subseteq Q \times Q$ to $\xrightarrow{\mathsf{s}} \subseteq Q \times Q$ for $\mathsf{s} \in \Sigma^b$

$$q \xrightarrow{\mathsf{ss'}} q' \quad \text{iff} \quad (\exists q'' \xleftarrow{\mathsf{s}} q)\ q'' \xrightarrow{\mathsf{s'}} q'$$

$$q \xrightarrow{b(\mathsf{s},\mathsf{s'})} q' \quad \text{iff} \quad q \xrightarrow{\mathsf{s}} q' \text{ and } (\exists q_1 \xleftarrow{b} q')(\exists q_2 \in F)\ q_1 \xrightarrow{\mathsf{s'}} q_2$$

and *accept* s if for some $q \in F$, $q_0 \overset{\mathsf{s}}{\Rightarrow} q$.

Slide 18

B-forms and $\trianglerighteq$ extended to Σ^b

Just as we can picture *next*(p) as $\Box\boxed{p}$, we have

$may(p) \rightsquigarrow b(\Box, \boxed{p})$ b-string

$\Box \xrightarrow{b} \boxed{p}$ b-automata fragment

$(\Box, \boxed{p})$ b-form

B-forms r and b-possibilities A

$$r \quad :== \quad (\alpha, A) \mid rr'$$
$$A \quad :== \quad \emptyset \mid r + A$$

We can encode b-strings as b-forms — e.g. $b(\alpha\alpha', \beta\beta')\gamma$ as $(\alpha, \emptyset)(\alpha', \beta\beta')(\gamma, \emptyset)$ — and vv.

$\trianglerighteq$ on strings extends to b-forms

$$(\alpha_1, A_1) \cdots (\alpha_n, A_n) \ \trianglerighteq\ (\beta_1, B_n) \cdots (\beta_n, B_n)$$
$$\text{iff} \quad \alpha_i \supseteq \beta_i \text{ and } A_i \supseteq B_i \text{ for } 1 \leq i \leq n$$

and then to b-languages as before.

Slide 19

Choice: b vs $+$

Condoravdi 2002: metaphysical vs epistemic

(7) Pat might have celebrated/slept/died yesterday [but did not]/[for all we know].

Veltman-esque update of $\Box$ with MIGHT(p) –

b: metaphysical	$+$: epistemic
$\boxed{may(p)}$	$\boxed{p}+\boxed{\overline{p}}$
event (pointwise)	event-type (non-distrib)
narrow: ‘and’	wide: ‘or’

Compare (from slide 12)

$$p \vee q \quad \rightsquigarrow \quad \boxed{p}+\boxed{q}$$

to Zimmermann 2000 (*or* and *epi* poss); whereas two b-possib’s in the *one* b-string $b(b(\Box, \boxed{p}), \boxed{q})$

$$\boxed{p} \xleftarrow{b} \Box \xrightarrow{b} \boxed{q} \qquad [b\text{-form } (\Box, \boxed{p}+\boxed{q})]\ .$$

We may have many b/may’s, each less about truth/entailments than about coherence/connec

Slide 20

Maximizing coherence: the Zipper (Ouch!)

Systematize slide 17 as follows:

given two strings from Σ^+

$$s = \alpha_1 \cdots \alpha_n$$
$$s' = \beta_1 \cdots \beta_m$$

such that $\alpha_1 \cup \beta_1 \in \Sigma$ (*compat*), let

$$k = (\text{MAX } j \leq \min(n,m))(\forall i \leq j)\ \alpha_i \cup \beta_i \in \Sigma$$
$$s_0 = (\alpha_1 \cup \beta_1) \cdots (\alpha_k \cup \beta_k) \ \in \ \Sigma^k$$

and define the *(asym) superposition* $s \&^\Sigma s'$ as

$$\begin{cases} s_0 & \text{if } k = n = m \\ b(s_0, \beta_{k+1} \cdots \beta_m) & \text{if } k = n < m \\ s_0 \alpha_{k+1} \cdots \alpha_n & \text{if } k = m < n \\ b(s_0, \beta_{k+1} \cdots \beta_m)\alpha_{k+1} \cdots \alpha_n & \text{otherwise.} \end{cases}$$

$$L \&^\Sigma L' = \{s \&^\Sigma s' \mid s \in L,\ s' \in L' \text{ compat}\}\ .$$

Slide 21

Does branching require force?

(2)″ ?The car was stationary before it hit the tree.

Where to attach the precondition $\overline{\text{still-car}}$ for car-hit-tree?

Need to introduce force for $\overline{\text{still-car}}$. Before or after car-was-stationary? (Non-verid/veridical.)

Is that why interpreting (2)″ requires effort?

(7)′ Pat might have been celebrating/ sleeping/dead yesterday ?but wasn't.

Interpreting (7)′ requires conditionalization ("had things been different") or aspectual coercion ("be celebrating" ⇝ "celebrate"; "be dead" ⇝ "die") as statives, being inertial, cannot on their own lead to branching.

Is that so?

Slide 22

Summing up

What's interesting about inertia worlds are
the forces that overturn inertia, and
the events that form worlds.

Inertia applies to formulas, strung out.

Formula-as-event-type constitutes a disjunctive normal form:

- inner conjunctions are events (proofs/truth-makers; slide 12)
- outer disjunctions are constrained by finite automata (regular language; slide 15).

Reichenbach's R neutralizes inertial inference: from simple past (slides 8,9) to the progressive.

Forces lead to branching (away from simple linear orders) — e.g. counterfactual implications.

Distinguish b (various) from + (epistemic):
coherence/connectedness vs truth/entailment.

No forces, no branching?

Slide 23

Details

Available from `www.cs.tcd.ie/Tim.Fernando`:

Fer1 A finite-state approach to events in nl sem, *J. Logic & Computation* 14:79-92, 2004

Fer2 Reichenbach's E, R and S in a finite-state setting (Sinn und Bedeutung 2003)

Fer3 Inertia in temporal modification (SALT 14, 2004)

Fer4 Events from temporal logic to regular languages with branching (Draft: 2005)

FeNa with Rowan Nairn, Entailments in finite-state temporality (IWCS-6, 2005)

plus course webpage

`www.cs.tcd.ie/Tim.Fernando/NASSLLI/`

Nat Lang Semantic Representations as Types

to be updated for ESSLLI 05 (Edinburgh).

Slide 24