Previously:

events as $\text{Pow}(\Phi)$-strings, subject to inertia
with executions tracked by $R$

Today’s Plan:

ground string + introduce background

schedule $p \subseteq T \times \Phi$  $P \subseteq \text{Pow}(T \times \Phi)$

Intuition:

ground/schedule $\approx$ execute/run machine $\approx$ realize in world

world $\approx$ $P$-generic set (w.r.t. $\parallel_p$)

= $P$ if extensional

background : implicit info (vs explicit $\triangleright$)

Applications: the progressive, causation

R as realis marker and $L$ mod $\kappa$

The set of $R$-continuations of $s$ is

$$\kappa_R(s) = \{ s' \in \text{Pow}(\Phi)^* \mid s' \in s_R \} .$$

Fix a (contextually given) map

$$\kappa : \text{Pow}(\Phi)^* \rightarrow \text{Pow}(\text{Pow}(\Phi)^*)$$

such that

(c1) $\kappa(s) \subseteq \kappa_R(s)$ for every $s \in \text{Pow}(\Phi)^*$.

Instead of $s \in L$, consider

$$(\exists s' \in \kappa(s)) \ s' \in L$$

stepping from $L$ to $L$ mod $\kappa$

$$L/\kappa = \{ s \in \text{Pow}(\Phi)^* \mid \kappa(s) \cap L \neq \emptyset \} .$$

In addition to (c1),

(c2) $s \in \kappa(s)$ for every $s \in \text{Pow}(\Phi)^*$.

Facts about $L/\kappa$

(F1) $L/\{\cdot\} = L$ where $\{\cdot\} : s \mapsto \{s\}$.

(F2) $L/\kappa \subseteq L/\kappa'$ if $(\forall s \in \text{Pow}(\Phi)^*) \ \kappa(s) \subseteq \kappa'(s)$.

(F3) $L \subseteq L/\kappa$ assuming (c2).

(F4) $(L/\kappa)_R = L_R$ assuming (c1) and (c2).

(F3) validates Landman 1992’s claim

if an accomplishment manages to get completed, it is unproblematic to assume (in retrospect) that the progressive is true during the development stage . . . even if the event gets completed against all odds.

Failure of converse is the imperfective paradox.

$$\kappa(s) - \{s\} = \text{what makes PROG problematic : if } \kappa(s) = \{s\}, \text{ there is no branching from } s.$$
Ground fluents from $\Phi$ in a set $T_i$ of times via relations $p \subseteq T_i \times \Phi$ construed as schedules
$p(t, \varphi)$ read as $p$ schedules $\varphi$ at $t$.

To get a string $\in \text{Pow}(\Phi)^*$ from a schedule $p$, fix
$succ \subseteq T_i \times T_i$ (with irreflexive $\text{succ}^+$)
$\text{ch}(\text{succ}) = \{t_1 \cdots t_n \in T_i^+ | \text{succ}(t_i, t_{i+1})$
for $1 \leq i < n\}$

Call $p$ a succ-strip if domain$(p) = \{t_1, \ldots, t_n\}$ for
some $t_1 \cdots t_n \in \text{ch}(\text{succ})$. (For irreflexive $\text{succ}^+$, $t_1 \cdots t_n$ is unique. Add $\top \in \Phi$ for $\Box$ as $\Box$)

Let
$\text{str}(p) = [\varphi \mid p(t_1, \varphi)] \cdots [\varphi \mid p(t_n, \varphi)]$

with $\text{str}(p) = \varepsilon$ if $p$ is not a strip, and let
$p : L, t$ iff $p(t, R)$ and $\text{str}(p) \in L$.

A string $s$ occurs in $p$ if $(\exists p' \subseteq p) s = \text{str}(p')$.

### Strips and $L$-events (grounded in $T_i, \text{succ}$)

Strip = time-stamped string
$p \approx \text{str}(p) \sqcup 0(t_1) \cdots 0(t_n)$

Expand alphabet with $0(t) \in \Phi - \lnr$
$p(t', 0(t))$ implies $t' = t$.

But factor out $\infty T_i$ for a finite-state approach.

Allow $\Phi$ to be infinite, with the understanding that when we speak of a regular language $L$, there is a finite $\Phi_0 \subseteq \Phi$ for which $L \subseteq \text{ Pow}(\Phi_0)^*$.

A strip $p$ is an $L$-event if
$[\varphi \mid p(t_1, \varphi)] \cdots [\varphi \mid p(t_n, \varphi)] \in L$.

That is, an $L$-event is a string $\alpha_1 \cdots \alpha_n \in L$
time-stamped in $\text{ch}(\text{succ})$.

$p$ and $\{(t, \varphi)\}$ are $\text{Pow}(\Phi)^+$-events.

### Schedules and strings

Fix a set $P \subseteq \text{Pow}(T_i \times \Phi)$ of possible schedules.

Define $\models_P$ with domain $\subseteq P$
$p \models_P L, t$ iff $(\exists p' \subseteq p) p' : L/K, t$
$p \models_P \neg A$ iff not $(\exists p' \supseteq p) p' \models_P A$

where
$p' \supseteq p$ iff $p' \supseteq p$ and $p, p' \in P$.

A $P$-generic $G \subseteq P$ induces a model $M[G]$ s.t.

$M[G] \models A$ iff $(\exists p \in G) p \models_P A$
$p \models_P \neg \neg A$ iff $(\forall P$-generic $G \ni p) M[G] \models A$

A $P$-world is $\bigcup G$ for some $P$-generic $G$ ($\approx P$ for extensional perspective).

$P = \bigcup\{G \mid G$ is $P$-generic\}

### Generic sets and worlds

Given $G \subseteq P$ and $p, p' \subseteq T_i \times \Phi$,

we write $p \sqsubseteq_G p'$ for $p$ is $G$-compatible with $p'$
$(\exists p'' \in G) p \cup p' \subseteq p''$.

$G$ is extensional if for all $p, p' \in G$, $p \sqsubseteq_G p'$.

A set $G \subseteq P$ is $P$-generic if

(i) $G$ is extensional

(ii) for all $p \in G$ and $p' \subseteq_P p, p' \in G$

(iii) for every $A$, there is a $p \in G$ such that
$p \models_P A$ or $p \models_P \neg A$.

A $P$-world is $\bigcup G$ for some $P$-generic $G$ ($\approx P$

for extensional perspective).
Background for causation

Max fell. John pushed him.

Fall before or after push?

SDRT: push-causes-fall + causes-precede-effects
soft (defeasible) hard

Cannot expect
\( L(push(j, m)) \supseteq \square^* \text{fall}(m). \)

Pass to set \( P \) of non-empty strips \( p \), writing \( \cdot \) first \( p \), for \( t \in \text{domain}(p) \) such that
\( \text{succ}^*(t, t') \) for all \( t' \in \text{domain}(p) \)
\( p|t \) for the \( t \)-history of \( p \) (\( \approx \) t-truncation)
\( p|t = \{ (t', \varphi) \in p \mid \text{succ}^*(t', t) \} \).

Causation against a background

Simple case involving \( p, \varphi, t, P \) illustrating
- counterfactuality (ii)
- no backward causation (iii),(iv)

A strip \( p \) causes \( \varphi, t \) against the background \( P \) if
(i) \( p \in P \) and \( p |\varphi, R \mid t \)
(ii) for some \( p' \in P, \; p' \not|\varphi, R \mid t \)
(iii) \( \text{succ}^+(\text{first}(p), t) \)
(iv) for all \( p' \in P, \; p' \supset_p p'|\text{first}(p) \)

N.B. (iv) precludes \( p \) from having any effect on \( \text{first}(p) \)-histories in \( P \).

Causes-precede-effects (iii),(iv) is hard, but
the choice of a salient \( P \) is soft and delicate
— already (i) may fail for different \( P \ni p \).

Subsumption \( \not\Rightarrow \) causation

Subsumption \( \supseteq \) need not imply causation: e.g.
\( L \supseteq \square^* \Rightarrow L \) presupposes \( \varphi \).

Nor need \( L \) be the force on \( \varphi \) when \( L \supseteq \square^* \varphi \).
Presuppositions may go beyond preconditions.

To ascribe agency, tag \( \varphi \) in \( L \) with \( c \) or \( p \)
\( \varphi_c \approx \varphi \) is contributed/caused
\( \varphi_p \approx \varphi \) is presupposed.

E.g. Recast \( \sim \varphi \varphi \) as \( (\sim \varphi)p \varphi_c \).
But in general, allow for \( L \supseteq \square^* \varphi_p \square^* \),
scheduling \( L \) with other \( L' \supseteq \square^* \varphi_c \square^* \).

Bring out background as \( P \subseteq \text{Pow}(Ti \times \Phi) \).
Accomplishment = [Activity CAUSE Achievement]

\[ p\text{-}walk\text{-}home \approx p\text{-}walk + p\text{-}arrive\text{-}at\text{-}home \]

Pat was walking home when . . .

\[ \text{PAST} (\text{PROG} (\text{Pat\text{-}walk\text{-}home})) \]

\[ \text{[PROG } \phi \text{] is true at a time interval } I \text{ iff } \phi \text{ is true at some end extension } I' \supseteq I \]

Given a world and time, an inertia world is exactly like the given world up to the time in question and in which the future course of events after this time develops in ways most compatible with the past course of events.

\[ \text{[PROG } \phi \text{] is true at } I, w \text{ iff } (\exists I' \supseteq I) (\forall w' \in \text{lnr}(I, w)) \phi \text{ is true at } I', w' \]

Conclusion: a computational semantics

A finite-state alternative to Prior -

& \cap, ic vs P, F, G, H, . . .

Make sense of E, R, S by constructing “reality” from runs of machines (regular languages)

- E as regular language/automaton/intension
- R marks position/computation-stage/realis

Inertial considerations induce different readings of the (Present) Perfect.

Claim (pace Montague). Intensions, not extensions are basic. Worlds arise from running machines under inertial laws that apply before S.

Nothing paradoxical in the Imperfective paradox if intensions (regular langs) are more basic than extensions (machine runs).

Context dependence of PROG can be pinned down to the choice \( \kappa(s) \) of continuations of \( s_R \).