Situations as strings

Tim Fernando

Trinity College Dublin

WoLLIC, Stanford July 2006

Starting point A proposition is a set of worlds

Idea Sharpen world to situation (Barwise, ...)

	·	
natural language semantics	world <i>w</i>	event <i>e</i>
	modality	temporality
	Carnap, Montague	Davidson,
L. Schubert on situations	S is true in w	e is of type S
logical semantics	truth-conditional set-theoretic models	proof-theoretic constructive types
topology	point-set	point-less

 $w \models \mathsf{may-have-rained-yesterday}$ $w \models \mathsf{rained-yesterday}$ iff $(\exists w' R w) w' \models$ rained-yesterday iff $(\exists e \sqsubseteq w) e$: rained-yesterday

Below *Propositions-as-types* (PaT)

PaT says nothing special about

- what proofs of atomic formulas are
- time or change (inertia/frame problem: McCarthy and Hayes)

Subatomic semantics (T. Parsons)

the study of those "formulas in English" that are treated as atomic formulas in most logical investigations of English. The main hypothesis to be investigated is that simple sentences of English contain subatomic quantification over events.

Tense and aspect

- (1) Pat had been gaining weight.
- (2) Pat read the newspaper for/in an hour.

Case Study: Linear Temporal Logic (LTL)

Kripke frame ($\mathbb{Z},<$) with present 0 and temporal precedence <

$i \in past$	iff	i < 0
$i \in future$	iff	0 < <i>i</i>

Valuation $x : \mathbb{Z} \to 2^P$ given a set P of atomic propositions

$$x \models p$$
 iff $p \in x(0)$
 $x \models next(\varphi)$ iff $x^1 \models \varphi$

where $x^i = (\lambda n \in \mathbb{Z}) x(i + n)$

 $x \models \varphi$ since ψ iff ($\exists i < 0$) $x^i \models \psi$ and $x^j \models \varphi$ for $i < j \le 0$

From valuations (points) to strings (basic open sets)

$$\begin{aligned} x &\models p \land \mathsf{next}(q) & \text{iff} \quad p \in x(0) \text{ and } q \in x(1) \\ & \text{iff} \quad \boxed{\mathsf{now}, p \mid q} \sqsubseteq x \\ x &\models p \text{ since } q \quad \text{iff} \quad (\exists s \sqsubseteq x) \ s \in \underbrace{q \mid p}^* \underbrace{\mathsf{now}, p}_{L(p \text{ since } q)} \end{aligned}$$

Analyze φ as a language $\mathit{L}(\varphi) \subseteq (2^{\mathit{P} \cup \{\mathrm{now}\}})^*$ so that

$$x\models arphi \quad (\exists s\sqsubseteq x) \ s\in L(arphi)$$

Conflating strings with languages,

$$L(p \land \text{previous}(q)) = \boxed{q \mid \text{now}, p}$$
$$L(p \text{ until } q) = \boxed{\text{now}, p \mid p}^* \boxed{q}$$

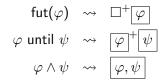
Infinite strings via lazy evaluation

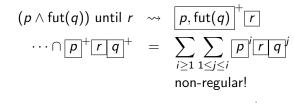
$$x \models \mathsf{always}_{>}(arphi) \quad \mathsf{iff} \quad (\forall i \ge 0) \; x^i \models arphi$$

$$L(\text{always}_{>}(p)) \approx \boxed{\operatorname{now}, p \mid p \mid p} \cdots$$
$$= \lim_{i \to \infty} \boxed{\operatorname{now}, p \mid p}^{i}$$

Finite approximations $s \in (2^{\Phi})^*$ where $\Phi \supseteq P \cup \{\mathsf{now}\}$

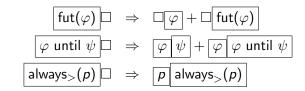
Finite-state issues





Regular sublanguage p^+ r q and for p, fut $(q)^+$ r, p^+ $(q, r) + r \square^* q)$

Constraints and subsumption



 $s \in A \Rightarrow B$ iff any stretch of s that contains a string in A contains one in B

Containment as subsumption \triangleright

 $\begin{array}{rl} a_1 \cdots a_n \trianglerighteq b_1 \cdots b_m & \text{iff} \quad n = m \text{ and } a_i \supseteq b_i \text{ for } 1 \le i \le n \\ & \boxed{p,q} \trianglerighteq \boxed{p} \trianglerighteq \boxed{p} + \boxed{q} \\ & L \trianglerighteq L' & \text{iff} \quad (\forall s \in L)(\exists s' \in L') \ s \trianglerighteq s' \end{array}$

Regularity of constraints and conciseness

No $(A \land \neg B)$ -counter-examples

$$A \Rightarrow B = (2^{\Phi})^* (A^{\unrhd} \cap \overline{B^{\trianglerighteq}})(2^{\Phi})^*$$

where

$$L^{\unrhd} = \{s \mid (\exists s' \in L) \ s \trianglerighteq s'\}$$
.

Apply the constraint $A \Rightarrow B$ to L

$$(A \Rightarrow B) \cap L^{\succeq}$$

and take \geq -minimal strings.

$$L_{\supseteq} = \{s \in L \mid (\forall s' \in L) \ s \supseteq s' \text{ implies } s = s'\}$$
$$= L - \{s \mid (\exists s' \in L - \{s\}) \ s \supseteq s'\}$$

Fact. L^{\succeq} and L_{\succeq} are regular if L is.

Inertia and force (*always* finitarily)

 φ is $\mathit{inertial}$ if it persists unless forced not to



- $f\overline{\varphi} \approx there is a force against \varphi$ $f\varphi \approx there is a force for \varphi$
- (3) Pat stopped the car before it hit the tree. $\cdots still(car) \qquad still(car) \cdots$
- (4) Pat left Dublin but is back. Pat has left Dublin but is back.

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. in the workings of a digital computer.

Discreteness

- in computation
- in planning
- from finiteness

To show: we can take time to be the real line

$$L(\text{rain from dawn to dusk}) = \overline{\text{rain, dawn rain}^+ \text{rain, dusk}}$$

Is each string rain, dawn rainⁱ rain, dusk a distinct event?
For $i \ge 1$, reduce to rain, dawn rain rain, dusk
— "no time without change"

interval reduction ir(s) of s

$$\mathsf{ir}(s) = \begin{cases} s & \text{if } \mathsf{length}(s) \le 1 \\ \mathsf{ir}(as') & \text{if } s = aas' \\ a\,\mathsf{ir}(a's') & \text{if } s = aa's' \text{ where } a \neq a' \end{cases}$$

$$ir(\Box \Box p p \Box \Box q \Box) = \Box p \Box q \Box$$

Events as inverse limits

Turn any finite sequence of real numbers

ſ

$$r_1 < r_2 < \cdots < r_n$$

into the string

$$r_1 \square r_2 \square \cdots \square r_n \square$$

to approximate the real line $(\Re, <)$ by finite subsets X of \Re .

$$\bigcap_{X}(a_{1}\cdots a_{n}) = (a_{1}\cap X)\cdots(a_{n}\cap X)$$

$$\bigcap_{\{r_{2},r_{4}\}}(\square \underline{r_{1}} \square \underline{r_{2}} \square \underline{r_{3}} \square \underline{r_{4}} \square) = \square \square \underline{r_{2}} \square \square \underline{r_{4}} \square$$

$$\operatorname{ir}_{X}(s) = \operatorname{ir}(\cap_{X}(s))$$

$$\operatorname{ir}_{\{r_{2},r_{4}\}}(\square \underline{r_{1}} \square \underline{r_{2}} \square \underline{r_{3}} \square \underline{r_{4}} \square) = \square \underline{r_{2}} \square \underline{r_{4}} \square$$

$$\underbrace{\lim_{K} (2^{X})^{*}}_{K} = \{(s_{X})_{X \in Fin(\Re)} \in \prod_{X \in Fin(\Re)} (2^{X})^{*} \mid$$

$$s_{X'} = \operatorname{ir}_{X'}(s_{X}) \text{ for } X' \subseteq X \in Fin(\Re) \}$$

Dynamic semantics (DRT, DPL): not strictly computational (negation = complement of the halting problem)

logical semantics	truth-conditional	proof-theoretic
topology	point-set	point-less (\sqsubseteq)
formal verification	model-checking	theorem-proving

In nl semantics, access to suitable model/point is problematic.

No compiler for English.

Partiality is crucial — there's nothing partial about a point.

Regard proofs as hypothetical (from a context of variable typings).