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, ...)

Situations big and small

natural language semantics	world w modality Carnap, Montague	event e temporality 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

$$w \models \text{may-have-rained-yesterday} \quad \text{iff} \quad (\exists w' R w)\ w' \models \text{rained-yesterday}$$

$$w \models \text{rained-yesterday} \quad \text{iff} \quad (\exists e \sqsubseteq w)\ e : \text{rained-yesterday}$$

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 $<$

$$\begin{array}{rcl} i \in \text{past} & \text{iff} & i < 0 \\ i \in \text{future} & \text{iff} & 0 < i \end{array}$$

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

$$\begin{array}{rcl} x \models p & \text{iff} & p \in x(0) \\ x \models \text{next}(\varphi) & \text{iff} & x^1 \models \varphi \end{array}$$

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

$$x \models \varphi \text{ since } \psi \quad \text{iff} \quad (\exists i < 0)\ x^i \models \psi \text{ and } x^j \models \varphi \text{ for } i < j \leq 0$$

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

$$x \models p \wedge \mathrm{next}(q) \quad \text{iff} \quad p \in x(0) \text{ and } q \in x(1)$$

$$\text{iff} \quad \boxed{\mathrm{now},\ p}\boxed{q} \sqsubseteq x$$

$$x \models p \text{ since } q \quad \text{iff} \quad (\exists s \sqsubseteq x)\ s \in \underbrace{\boxed{q}\boxed{p}^{*}\boxed{\mathrm{now},\ p}}_{L(p \text{ since } q)}$$

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

$$x \models \varphi \quad \text{iff} \quad (\exists s \sqsubseteq x)\ s \in L(\varphi)$$

Conflating strings with languages,

$$L(p \wedge \mathrm{previous}(q)) \quad = \quad \boxed{q}\boxed{\mathrm{now},\ p}$$

$$L(p \text{ until } q) \quad = \quad \boxed{\mathrm{now},\ p}\boxed{p}^{*}\boxed{q}$$

Infinite strings via lazy evaluation

$$x \models \text{always}_>(\varphi) \quad \text{iff} \quad (\forall i \geq 0)\ x^i \models \varphi$$

$$L(\text{always}_>(p)) \approx \boxed{\text{now}, p}\boxed{p}\boxed{p}\boxed{p}\cdots$$

$$= \lim_{i \to \infty} \boxed{\text{now}, p}\boxed{p}^i$$

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

$$\boxed{\text{now}, \text{always}_>(p)} \leadsto \boxed{\text{now}, p}\boxed{\text{always}_>(p)}$$

$$\leadsto \boxed{\text{now}, p}\boxed{p}\boxed{\text{always}_>(p)}$$

$$\vdots$$

$$\leadsto \boxed{\text{now}, p}\boxed{p}^i\boxed{\text{always}_>(p)} \leadsto \cdots$$

Finite-state issues

$$\text{fut}(\varphi) \quad \rightsquigarrow \quad \Box^+ \boxed{\varphi}$$

$$\varphi \text{ until } \psi \quad \rightsquigarrow \quad \boxed{\varphi}^+ \boxed{\psi}$$

$$\varphi \wedge \psi \quad \rightsquigarrow \quad \boxed{\varphi, \psi}$$

$$(p \wedge \text{fut}(q)) \text{ until } r \quad \rightsquigarrow \quad \boxed{p, \text{fut}(q)}^+ \boxed{r}$$

$$\cdots \cap \boxed{p}^+ \boxed{r}\boxed{q}^+ \quad = \quad \sum_{i \geq 1} \sum_{1 \leq j \leq i} \boxed{p}^i \boxed{r}\boxed{q}^j$$

non-regular!

Regular sublanguage $\boxed{p}^+ \boxed{r}\boxed{q}$ and for $\boxed{p, \text{fut}(q)}^+ \boxed{r}$,

$$\boxed{p}^+ (\boxed{q, r} + \boxed{r}\Box^* \boxed{q})$$

Constraints and subsumption

$$\boxed{\text{fut}(\varphi)}\boxed{} \;\Rightarrow\; \boxed{}\boxed{\varphi} + \boxed{}\boxed{\text{fut}(\varphi)}$$

$$\boxed{\varphi \text{ until } \psi}\boxed{} \;\Rightarrow\; \boxed{\varphi}\boxed{\psi} + \boxed{\varphi}\boxed{\varphi \text{ until } \psi}$$

$$\boxed{\text{always}_{>}(p)}\boxed{} \;\Rightarrow\; \boxed{p}\boxed{\text{always}_{>}(p)}$$

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

Containment as subsumption $\trianglerighteq$

$$a_1 \cdots a_n \trianglerighteq b_1 \cdots b_m \quad \text{iff} \quad n = m \text{ and } a_i \supseteq b_i \text{ for } 1 \leq i \leq n$$

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

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

Regularity of constraints and conciseness

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

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

where

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

Apply the constraint $A \Rightarrow B$ to L

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

and take $\trianglerighteq$-minimal strings.

$$\begin{aligned} L_{\trianglerighteq} \;\; &= \;\; \{s \in L \mid (\forall s' \in L)\ s \trianglerighteq s' \text{ implies } s = s'\} \\ &= \;\; L - \{s \mid (\exists s' \in L - \{s\})\ s \trianglerighteq s'\} \end{aligned}$$

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

Inertia and force (*always* finitarily)

φ is *inertial* if it persists unless forced not to

$$\boxed{\varphi}\,\Box \;\Rightarrow\; \Box\,\boxed{\varphi} + \boxed{\mathsf{f}\overline{\varphi}}\,\Box$$

$$\Box\,\boxed{\varphi} \;\Rightarrow\; \boxed{\varphi}\,\Box + \boxed{\mathsf{f}\varphi}\,\Box$$

$\mathsf{f}\overline{\varphi}$ $\approx$ there is a force against φ

$\mathsf{f}\varphi$ $\approx$ there is a force for φ

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

$$\cdots\,\boxed{\text{still(car)}} \qquad\qquad \boxed{\overline{\text{still(car)}}}\,\cdots$$

(4) Pat left Dublin but is back.

?Pat has left Dublin but is back.

Coming to terms with discreteness

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

Events as strings, reduced

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

Is each string $\boxed{\text{rain, dawn}}\,\boxed{\text{rain}}^{i}\,\boxed{\text{rain, dusk}}$ a distinct event?

For $i \geq 1$, reduce to $\boxed{\text{rain, dawn}}\,\boxed{\text{rain}}\,\boxed{\text{rain, dusk}}$

— "no time without change"

interval reduction $\mathrm{ir}(s)$ of s

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

$$\mathrm{ir}(\Box\Box\boxed{p}\boxed{p}\boxed{p}\Box\Box\boxed{q}\Box) \quad = \quad \Box\boxed{p}\Box\boxed{q}\Box$$

Events as inverse limits

Turn any finite sequence of real numbers

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

into the string

$$\square\boxed{r_1}\square\boxed{r_2}\square\cdots\square\boxed{r_n}\square$$

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

$$\begin{aligned}
\cap_X(a_1 \cdots a_n) &= (a_1 \cap X) \cdots (a_n \cap X) \\
\cap_{\{r_2,r_4\}}(\square\boxed{r_1}\square\boxed{r_2}\square\boxed{r_3}\square\boxed{r_4}\square) &= \square\square\square\boxed{r_2}\square\square\square\boxed{r_4}\square \\
\mathrm{ir}_X(s) &= \mathrm{ir}(\cap_X(s)) \\
\mathrm{ir}_{\{r_2,r_4\}}(\square\boxed{r_1}\square\boxed{r_2}\square\boxed{r_3}\square\boxed{r_4}\square) &= \square\boxed{r_2}\square\boxed{r_4}\square
\end{aligned}$$

$$\varprojlim (2^X)^* = \{(s_X)_{X \in Fin(\Re)} \in \prod_{X \in Fin(\Re)} (2^X)^* \mid s_{X'} = \mathrm{ir}_{X'}(s_X) \text{ for } X' \subseteq X \in Fin(\Re)\}$$

Natural versus programming languages

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).