A Finite-State Approach to Event Semantics

Tim Fernando
Computer Science Department
Trinity College, Dublin 2, Ireland
Tim.Fernando@cs.tcd.ie

Abstract

Events employed in natural language semantics are characterized in terms of regular languages, each string in which can be regarded as a motion picture. The relevant finite automata then amount to movie cameras/projectors, or more formally, to finite Kripke structures with partial valuations. The usual regular constructs (concatenation, choice, etc) are supplemented with superposition of strings/automata/languages, realized model-theoretically as conjunction.

1. Introduction

Due in no small measure to [2], events of some form or another have become a common tool for semantically analyzing expressions of change in English (e.g. [15, 6, 9]). Under this approach, a sentence such as (1) is taken to describe an event of Mary swimming a mile, culminating in the past.

(1) Mary swam a mile.

Such events are formulated below as runs of machines that collectively constitute a causal order around which to explain temporality in natural language ([11, 18]). Similar ideas have been developed in [19, 13, 12, 17, 5], the distinctive feature of the present proposal being the use of a finite automaton for a declarative representation (as opposed to a procedural implementation) of a fragment of a first-order model. That fragment is given by strings accepted by the automaton — that is to say, by motion pictures taken by a movie camera (or, passing from accepting to generating devices, by films played by a movie projector).

2. Event-types as automata/languages

Formally, it is convenient to present the relevant automata as Kripke models over some finite set Φ of formulas (roughly the propositional fluents in [10]), defining

(i) a *(finite, 2-pointed) frame* $\langle N, A, 0, 1\rangle$ to be a finite set N of *nodes*, a set $A \subseteq N \times N$ of *arcs*, and distinguished nodes 0 and 1 (which often but not always are the numbers 0 and 1 ordinarily denote), and

(ii) a *(Φ-)event-automaton* E to be a frame $\langle N_E, A_E, 0_E, 1_E\rangle$ and *labeling* function $l_E : N_E \to \mathit{Pow}(\Phi)$ that maps a node $n \in N_E$ to a set $l_E(n) \subseteq \Phi$ of formulas.

To illustrate, the very rudimentary picture (2) of *die*(*Romeo*) is provided by the event-automaton $\langle\{0,1\}, \{(0,1)\}, 0, 1\rangle$, l where $\mathit{dom}(l) = \{0,1\}$, $l(0) = \{\textit{alive}(\textit{Romeo})\}$ and $l(1) = \{\textit{dead}(\textit{Romeo})\}$.

(2) $\boxed{\textit{alive}(\textit{Romeo})} \longrightarrow \boxed{\textit{dead}(\textit{Romeo})}$

Alternative analyses of *die*(*Romeo*) are, of course, possible, the idea being to

(i) generalize $(0,1)$ to a path $0 \cdots 1$ from 0 to 1, and

(ii) label a node n on that path by a set of formulas (true at n).

The *language* $\mathcal{L}(E)$ *of* E is

$$\mathcal{L}(E) \;=\; \{l_E(0_E) \cdots l_E(1_E) \mid 0_E \cdots 1_E \in \mathit{Path}(E)\}$$

where $\mathit{Path}(E)$ consists of strings $n_1 \cdots n_k \in {N_E}^+$ such that $n_1 = 0_E$, $n_k = 1_E$ and $(n_i, n_{i+1}) \in A_E$ for $1 \leq i < k$. Clearly, $\mathcal{L}(E)$ is just the language accepted by the finite automaton with initial node $\textsc{start} \notin N_E$, accepting node 1_E, and labeled transitions

$$n \stackrel{l_E(m)}{\longrightarrow} m \quad \text{for } (n,m) \in A_E$$

plus $\textsc{start} \stackrel{l_E(0_E)}{\longrightarrow} 0_E$. (That is, the finite automaton for E is obtained by moving node labels over to arcs that point to the node, throwing in an extra node START to point to 0_E.) Conversely, an obvious limitation on an event-automaton E is that it accepts only non-empty strings, any two of which begin with the same symbol and end with the same symbol. This limitation can be overcome by permitting the empty

string ϵ to label 0_E and/or 1_E, so as to implement non-deterministic choice $+$ as follows.

$$E + E' \quad = \quad \begin{array}{ccccc} & & E & & \\ & \nearrow & & \searrow & \\ [\epsilon] & & & & [\epsilon] \\ & \searrow & & \nearrow & \\ & & E' & & \end{array}$$

As it turns out, it suffices to allow $l_E(0_E) = l_E(1_E) = \epsilon$ in order to capture all regular languages.

Proposition 1. *For every regular language $L \subseteq Pow(\Phi)^*$, there is a finite set $\mathcal{E}[L]$ of Φ-event-automata, the sum of which accepts the non-empty strings in L*

$$L - \{\epsilon\} \quad = \quad \bigcup_{E \in \mathcal{E}[L]} \mathcal{L}(E) \, .$$

Proof. Working with regular expressions L, we define $\mathcal{E}[L]$ by induction. Let $\mathcal{E}[L + L']$ be $\mathcal{E}[L] \cup \mathcal{E}[L']$, and $\mathcal{E}[LL']$ be $\{EE' \mid E \in \mathcal{E}[L],\ E' \in \mathcal{E}[L']\}$ unioned with $\mathcal{E}[L]$ if $\epsilon \in L'$ and/or $\mathcal{E}[L']$ if $\epsilon \in L$ (defining EE' in the obvious way). As for L^*, form

$$\begin{array}{rcl} N & = & \{(n, E) \mid E \in \mathcal{E}[L],\ n \in N_E\} \\ A & = & \{((n, E), (m, E)) \mid E \in \mathcal{E}[L],\ (n, m) \in A_E\} \\ & & \cup \{((1_E, E), (0_{E'}, E')) \mid E, E' \in \mathcal{E}[L]\} \\ l(n, E) & = & l_E(n) \end{array}$$

and set $\mathcal{E}[L^*] = \{\langle N, A, (0_E, E), (1_E, E)\rangle, l \mid E \in \mathcal{E}[L]\}$. $\dashv$

2.1. Moens-Steedman and the Vendler classes

Applying [11], let 0 and 1 be preparatory and consequent states respectively, and let a durative event(-type) E come with an inceptive event E^i and a culminative event E^c, the consequent state of E^i being the preparatory state of E^c, termed the progressive state p_E of E

$$p_E \quad = \quad 1_{E^i} \quad = \quad 0_{E^c} \, .$$

Extracting a loop $p_E \rightarrow p_E$ from the equality $1_{E^i} = 0_{E^c}$ and taking $0_E = 0_{E^i}$ and $1_E = 1_{E^c}$, we get the transitions

$$A_E \quad = \quad 0_E \rightarrow p_E \rightarrow p_E \rightarrow 1_E \, .$$

From this, Vendler's well-known aspectual classifications drop out once p_E is related to 1_E and 0_E. For activities E like "swim," $p_E = 1_E$ inasmuch as every subtransition of a swim counts as a swim. By contrast, for accomplishments E such as "swim a mile," no proper subtransition from 0_E can end at 1_E. (This is the "directed" analog of *quantization* [9]). The contrasting factive entailments of activities

$$\text{Mary was swimming} \quad \models \quad \text{Mary swam}$$

and accomplishments

$$\text{Mary was swimming a mile} \quad \not\models \quad \text{Mary swam a mile}$$

can then be linked to the test: is $p_E = 1_E$? Identifying statives E with the equations $0_E = p_E = 1_E$, the oddness (or markedness) of the progressive form of stative verbs might be blamed on the equality of progressive and simple forms (making the progressive operator, as it were, semantically redundant) or, focusing on 0_E, the lack of progress from 0_E (insofar as $0_E = 1_E$). Pushing this line further, let $p_E = 0_E$ for achievements E such as "win" and "begin," reducing the difference between achievements and accomplishments mentioned, for example, in [6], pp 560-561, to whether or not $p_E = 0_E$ ($p_E = 1_E$ holding in neither case).

Path(E)	Vendler-class(E)	$p_E = 1_E$	$p_E = 0_E$
01^+	activity	$+$	$-$
$0p^+1$	accomplishment	$-$	$-$
1^+	stative $[0 = 1]$	$+$	$+$
0^+1	achievement	$-$	$+$

Table 1. A first stab.

Is Table 1 faithful to a reading of 0_E and 1_E as the preparatory and consequent states of E? Do we want to confuse the progressive state of "swimming" with the consequent state for "have swum"? The answer to both questions must surely be: *no*. Which is *not* to say that Table 1 is all wrong. But rather that it ought to be re-interpreted with the identifications '0=preparatory' and '1=consequent' relaxed. Table 1 gives only a partial picture — namely, that concerning some formula φ expressing culmination of (an event-occurrence of type) E. Indeed, the entries 01^+ and 0^+1 in Table 1 are reminiscent of the constructs *Con-BEC*(φ) and *Min-BEC*(φ) in [13], where the Vendler classes are characterized by binary features [$\pm$ for] and [$\pm$ Prog] that Table 1 interprets according to (3).

(3) E is [+ for] iff $p_E = 1_E$.
E is [+ Prog] iff $p_E \neq 0_E$.

(3) leaves out the labeling l_E that turns a frame into a Kripke model, suggesting that aspectual properties we ascribe to E might be reducible to the frame of E.

2.2. Framing and/or simulating aspect?

Testing the hypothesis that aspect can be confined to frames calls for a fuller account of aspect than that offered by Table 1. [19] provides an elegant analysis of aspect, with arcs in a frame beefed up to *arrows* ([20]). Most (if not all) the ideas in [19] can, I suspect, be reformulated in terms of event-automata, with a single arrow blown up to a (sub)frame (of a monster frame). To my knowledge, such

a reformulation has yet to be carried out, or shown conclusively to be impossible (or a step backwards). Whether or not [19] provides, on balance, evidence for the reducibility of aspect to frames (be they irreducibly populated by arrows or not), I think it is fair to describe the focus of [19] as being a frame (drawn there as figure 12, page 98) wherein to locate certain formulas (written there $\varphi, Cl(\varphi), Pf(\varphi)$, etc). The attention paid to the frame is well-deserved inasmuch as it fleshes out a temporal ontology for aspect, with arrows for the progressive, the perfect, etc.

Further afield, "an active computational representation for verb semantics called **x-schemas**" is presented in [1] that analyzes aspect "in terms of the context-sensitive interaction between verb-specific x-schemas and a **controller** x-schema that captures important regularities in the evolution of events." The controller x-schema goes beyond the temporal ontology of [19] in recognizing points at which events are suspended, resumed, etc. Indeed, x-schemas go beyond much of formal natural language semantics in offering a cognitive processing picture with "simulative inference," as opposed to a model-theoretic account oriented around (not so much a mind that processes language as) an external reality that language describes. That said, there is a growing appreciation within model-theoretic semantics of the importance of cognitive considerations (e.g. [18, 5]). A model-theoretic account that says nothing about cognitive mechanisms can hardly be a complete theory of language. But this does not render the admittedly incomplete pictures offered by traditional model-theoretic analyses irrelevant. If programming languages have denotational semantics distinct from their operational semantics, why not natural languages? It is precisely to understand what computational accounts such as [12] come to — and the proliferation of concurrency models suggests there are bound to be many such accounts — that one abstracts away as much of their computational details as one can usefully get away with. (Exactly what is useful is, alas, a matter of taste.)

Getting back to the specifics at hand, we have from page 9 of [18] the following claim:

> aspectual categories like activity and accomplishment are *ways of viewing* a happenning, rather than intrinsic properties of verbs and associated propositions, or of objective reality and the external world.

Keeping in mind the motion picture-camera/projector metaphor previously mentioned, it is natural to associate

(i) "ways of viewing" with an event-automaton E (or, more narrowly, the frame of E)

and, on the other hand,

(ii) the "associated propositions," "objective reality and the external world" with what E is about.

But what is E about? Notice that Table 1 falls short of a model-theoretic analysis of say, [$\pm$ for] and [$\pm$ Prog] under (3). In particular, it is natural to ask: how can we make precise what information the nodes 1, 0 and p encode in Table 1? A brief answer is: apply the labeling l_E of E to 1, 0 and p. Passing from *Path*(E) to $\mathcal{L}(E)$, our emphasis shifts from the mechanism E to the description $\mathcal{L}(E)$ of "the external world" that E contributes (as §3 below spells out).

2.3. Superposition: from N and $Pow(\Phi)$ to *Cn*

Apart from the usual regular constructs composing finite automata with each other, the particular alphabet *Pow*(Φ) suggests aligning two strings $\alpha_1 \cdots \alpha_k$ and $\alpha'_1 \cdots \alpha'_k$ of equal length against each other to form a string

$$(\alpha_1 \cup \alpha'_1) \cdots (\alpha_k \cup \alpha'_k)$$

of length k, where the ith-symbol $\alpha_i \cup \alpha'_i$ is α_i and α'_i superimposed on each other. For instance, taking

$$\begin{aligned} \alpha &= \{\textit{swim}(\textit{Mary})\} \\ \alpha'_1 &= \{\textit{in}(\textit{Mary,Ireland})\} \\ \alpha'_2 &= \{\textit{in}(\textit{Mary,IrishSea})\} \\ \alpha'_3 &= \{\textit{in}(\textit{Mary,Wales})\}\ , \end{aligned}$$

the string $\alpha\alpha\alpha$ portraying Mary swimming can be componentwise superimposed on $\alpha'_1\alpha'_2\alpha'_3$ to give the string

$$(\alpha \cup \alpha'_1)\ (\alpha \cup \alpha'_2)\ (\alpha \cup \alpha'_3)$$

depicting Mary swimming from Ireland to Wales. Now, stepping from strings up to languages L and L', let

$$\begin{aligned} L \,\&_\cup\, L' &= \{(\alpha_1 \cup \alpha'_1) \cdots (\alpha_k \cup \alpha'_k) \mid k \geq 1, \\ &\qquad \alpha_1 \cdots \alpha_k \in L,\ \alpha'_1 \cdots \alpha'_k \in L'\}\ , \end{aligned}$$

while over event automata E and E', define an event-automaton $E \times_\cup E'$ by $N_{E\times_\cup E'} = N_E \times N_{E'}$, $0_{E\times_\cup E'} = (0_E, 0_{E'})$, $1_{E\times_\cup E'} = (1_E, 1_{E'})$, and

$$\begin{aligned} A_{E\times_\cup E'} &= \{((n,n'),(m,m')) \mid (n,m) \in A_E \\ &\qquad \text{and } (n',m') \in A_{E'}\} \\ l_{E\times_\cup E'}(n,n') &= l_E(n) \cup l_{E'}(n')\ . \end{aligned}$$

$E \times_\cup E'$ is the (unconstrained) concurrent composition of E and E', with language

$$\mathcal{L}(E \times_\cup E') = \mathcal{L}(E)\ \&_\cup\ \mathcal{L}(E')\ .$$

But should we be allowed to superimpose any two pictures α and α' on each other? If pictures are assumed to be complete descriptions (as the sets labeling a Kripke model

ordinarily are), then they can be superimposed only on themselves, suggesting that we restrict the nodes of $E \times_\cup E'$ to the pullback

$$\{(n, n') \in N_E \times N_{E'} \mid l_E(n) = l_{E'}(n')\}.$$

This restriction, call it $P(E, E')$, yields the usual construction intersecting languages $\mathcal{L}(E)$ and $\mathcal{L}(E')$

$$\mathcal{L}(P(E, E')) \quad = \quad \mathcal{L}(E) \cap \mathcal{L}(E') \; .$$

On the other hand, if pictures are understood as *in*complete (as we shall), then some middle ground between $E \times_\cup E'$ and $P(E, E')$ might be sought, allowing the superposition of some but not all pairs of pictures. (Take 'Mary-swimming' and 'two-ticks-of-an-hour clock' versus 'Mary-sleeping' and 'Mary-wide-awake.') Accordingly, let us weaken the requirement on acceptable node pairs n, n' from $l_E(n) = l_{E'}(n')$ to $l_E(n) \cup l_{E'}(n')$ being, in a precise sense, legitimate.

To pick out what pictures an event-automaton can frame, let us henceforth assume Φ comes equipped with a non-empty family *Cn* $\subseteq$ *Pow*(Φ) that is $\subseteq$-closed (ie for all $\alpha \in$ *Cn*, *Pow*$(\alpha) \subseteq$ *Cn*), with the intended reading

$$\alpha \in \textit{Cn} \quad \text{iff} \quad \alpha \text{ is } \textit{con}\text{sistent/}\textit{con}\text{ceivable/a } \textit{c}\text{artoo}\textit{n}$$

for every $\alpha \subseteq \Phi$. *Cn* induces the refinement

$$\begin{aligned} L \,\&_{Cn}\, L' \quad &= \quad (L \,\&_\cup\, L') \cap \textit{Cn}^+ \\ &= \quad \{(\alpha_1 \cup \alpha'_1) \cdots (\alpha_k \cup \alpha'_k) \mid k \geq 1, \\ &\qquad \alpha_1 \cdots \alpha_k \in L,\ \alpha'_1 \cdots \alpha'_k \in L' \\ &\qquad \text{and } \alpha_i \cup \alpha'_i \in \textit{Cn} \text{ for } 1 \leq i \leq k\} \end{aligned}$$

of $\&_\cup$. As for $E \times_\cup E'$, let $E \times_{Cn} E'$ be the restriction of $E \times_\cup E'$ to the set of nodes

$$\{(n, n') \in N_E \times N_{E'} \mid l_E(n) \cup l_{E'}(n') \in \textit{Cn}\}$$

provided this set includes both $(0_E, 0_{E'})$ and $(1_E, 1_{E'})$; otherwise, let $E \times_{Cn} E'$ be the event-automaton

$$\langle \{0, 1\}, \emptyset, 0, 1 \rangle, l \text{ where } l(0) = l(1) = \emptyset$$

with empty language.

Proposition 2. *For all event-automata E and E',*

$$\mathcal{L}(E \times_{Cn} E') \quad = \quad \mathcal{L}(E) \,\&_{Cn}\, \mathcal{L}(E') \; .$$

Henceforth, we focus on regular languages over *Cn*, as opposed to event-automata (linked to these languages according to Propositions 1 and 2). This has a technical advantage illustrated by the ease in formulating

Proposition 3. *For all languages L, L' and $L'' \subseteq$ Pow$(\Phi)^*$,*

(a) $L \,\&_{Cn}\, L' = L' \,\&_{Cn}\, L$

(b) $(L \,\&_{Cn}\, L') \,\&_{Cn}\, L'' = L \,\&_{Cn}\, (L' \,\&_{Cn}\, L'')$

(c) $L \,\&_{Cn}\, \emptyset^+ = L$ *iff* $L \subseteq \textit{Cn}^+$.

2.4. Table 1 and (3) revisited with superposition

Returning to §2.1, with an event-automaton E replaced by a language L, let us define analogs of $0_E, p_E$ and 1_E as pictures $\alpha(L), \delta(L)$ and $\omega(L) \subseteq$ *Pow*(Φ) of the inceptive, progressive and culminative stages of L given by

$$\begin{aligned} \alpha(L) \quad &= \quad \bigcap \{\sigma_1 \mid \sigma \in L\} \\ \delta(L) \quad &= \quad \bigcap \{\sigma_i \mid i > 1 \text{ and } \sigma \in L \text{ with length} > i\} \\ \omega(L) \quad &= \quad \bigcap \{\sigma_i \mid i \geq 1 \text{ and } \sigma \in L \text{ with length } i\} \end{aligned}$$

where σ_i is the ith-symbol of the infinite string $\sigma\emptyset^\infty$ obtained by concatenating σ to the left of the infinite string $\emptyset^\infty$ of $\emptyset$'s. Next, instead of forming $0_E \to p_E \to p_E \to 1_E$, define the "Moens-Steedman" language

$$\text{MS}(L) \quad = \quad \alpha(L)\, \delta(L)^+\, \omega(L)$$

(which differs from Table 1 in yielding strings only with length ≥ 3) and in place of (3), let

$$\begin{aligned} \text{for}(L) \quad &= \quad \emptyset\, \omega(L)^+\, \emptyset \\ \text{prog}(L) \quad &= \quad \emptyset\, \overline{\alpha(L)}^+\, \emptyset \end{aligned}$$

where $\bar{\cdot}$ is a negation operation on subsets β of Φ such that $\beta \cup \overline{\beta} \notin$ *Cn* and for all $\gamma \in$ *Cn*,

$$\gamma \cup \beta \in \textit{Cn} \quad \text{or} \quad \gamma \cup \overline{\beta} \in \textit{Cn} \; .$$

Finally, we turn Table 1 into the definitions

$$\begin{aligned} \text{Activ}(L) \quad &= \quad \text{MS}(L) \,\&_{Cn}\, \text{for}(L) \,\&_{Cn}\, \text{prog}(L) \\ \text{Accmp}(L) \quad &= \quad \text{MS}(L) \,\&_{Cn}\, \text{for}_-(L) \,\&_{Cn}\, \text{prog}(L) \\ \text{Stat}(L) \quad &= \quad \text{MS}(L) \,\&_{Cn}\, \text{for}(L) \,\&_{Cn}\, \text{prog}_-(L) \\ \text{Achie}(L) \quad &= \quad \text{MS}(L) \,\&_{Cn}\, \text{for}_-(L) \,\&_{Cn}\, \text{prog}_-(L) \end{aligned}$$

where

$$\begin{aligned} \text{for}_-(L) \quad &= \quad \emptyset\, \overline{\omega(L)}^+\, \emptyset \\ \text{prog}_-(L) \quad &= \quad \emptyset\, \alpha(L)^+\, \emptyset \; . \end{aligned}$$

Having used the function MS to motivate the definitions of for(L), prog(L), for$_-(L)$ and prog$_-(L)$, notice that MS contributes nothing to differentiating $v(L)$ for $v \in$ {Activ, Accmp, Stat, Achie}. And instead of defining for$_-(L)$ and prog(L) in terms of negation $\bar{\cdot}$ on subsets of Φ, we might specify how a function on languages classifies languages. Given an arbitrary function f on languages, let us say

(i) L *is* $[-f]$ if $L \& _{Cn} f(L) = \emptyset$

(ii) L is *f-acceptable* if $L \& _{Cn} f(L) \neq \emptyset$, and

(iii) L *is* $[+f]$ if L is f-acceptable and $L = L \& _{Cn} f(L)$.

Notice that if L is Activ-acceptable, then L is for- and prog-acceptable, and Activ(L) is $[+$ for$]$ and $[+$ prog$]$. Similar remarks can be made about the other $v(L)$'s.

Under the definitions above, $[+$ for$]$ amounts to a sortal/aspectual restriction that 'for' imposes on the verb phrase with which it combines. (Furthermore, that restriction is treated along the lines of the approach to presupposition in [7], with L satisfying the presuppositions of 'for' precisely if L is $[+$ for$]$.) Passing again to arbitrary functions f on languages, let us call f *Cn-conjunctive* if

$$\begin{aligned} f(L) &= f(f(L)) \\ f(L) &= f(L)\ \&_{Cn}\ f(L) \\ f(L) &= f(L\ \&_{Cn}\ f(L)) \end{aligned}$$

for all languages L that are f-acceptable.

Proposition 4.

(a) *If f is Cn-conjunctive, then for every f-acceptable language L, both $f(L)$ and $L\&_{Cn}f(L)$ are $[+f]$.*

(b) *Each of* Activ, Accmp, Stat, Achie *is Cn-conjunctive, as are the functions sending L to $\emptyset\ \omega(L)^+\omega(L)$ and $\alpha(L)\ \alpha(L)^+\emptyset$ (which are slight variants of* for(L) *and* prog$_-(L)$*, respectively).*

Pausing for an example, consider the following pair from [1].

(4) a. She read the book for an hour.

b. She read the book in an hour.

On the surface, (4) poses a challenge to the prohibition against 'for' and 'in' being able to fill the same holes (ie $[+$ for$]$=$[-$ in$]$). But, as pointed out in [1], (4b) entails that 'she finished reading the book' whereas (4a) does not. That is, 'read the book' amounts in (4a) to 'read *parts* of the book' and in (4b) to 'read the *entire* book.' It is well-known (e.g. [9, 13]) that the argument of a verb can shape the aspectual property of the verb phrase, so that, in particular, 'read parts of the book' is naturally conceptualized as an activity, whereas 'read the book' (or especially, 'read the entire book') is an accomplishment (that culminates with all unread parts of the book consumed). Thus, 'for an hour' combines easily with 'read parts of the book,' while 'in an hour' modifies 'read the entire book.' But what does $\&_{Cn}$ have to do with any of this? Following the widespread use of conjunction in event semantics, we can apply $\&_{Cn}$ to combine not only 'for an hour' with 'read parts of the book' but also the argument 'parts of the book' (or 'the entire book') with 'read.' That is, $\&_{Cn}$ is offered here as a tool for the logical investigation that lexical semantics richly deserves (e.g. [3, 15]). Focusing on time, let us work out a simple example — 'read for an hour,' analyzed as

$$\begin{aligned} & (\emptyset\ \{\textit{read}\}^+)\ \&_{Cn}\ \text{for}(\emptyset\ \{\textit{read}\}^+)\ \&_{Cn} \\ & (\{\textit{time}(x)\}\ \emptyset^+\{\textit{time}(y),\ \textit{hour}(x,y)\}) \\ = \ & \{\textit{time}(x)\}\ \{\textit{read}\}^+\{\textit{read},\ \textit{time}(y),\ \textit{hour}(x,y)\} \end{aligned}$$

with parameters x and y, and restrictions *time*(x), *time*(y), *hour*(x,y) that we will return to in the next section.

3. Event-tokens and models

Our attention in this section shifts from event-*types* to event-*tokens*, embedded in a model of reality that the formulas in Φ describe. An important part of that model is a temporal frame $(\mathit{Ot}, \mathsf{S}_\tau)$ consisting of a "successor" relation $\mathsf{S}_\tau \subseteq \mathit{Ot} \times \mathit{Ot}$ on a set *Ot* of "observation times."

Examples.

(i) *Ot* is the set of integers, and S_τ is the usual successor $(+1)$ function.

(ii) $\mathit{Ot} \subseteq \mathit{Pow}(\mathit{Pt}) - \{\emptyset\}$ for some set *Pt* of points linearly ordered by $<_\tau$, and S_τ is the set of all pairs $(t, t') \in \mathit{Ot} \times \mathit{Ot}$ such that

$$\begin{aligned} & (\forall x \in t)(\forall y \in t')\ x <_\tau y \ \text{ and} \\ & \text{not } (\exists t'' \in \mathit{Ot})\ t'' \subseteq \mathit{gap}(t, t') \end{aligned}$$

where $\mathit{gap}(t,t')$ is $\{z \in \mathit{Pt} - (t \cup t') \mid (\exists x \in t)(\exists y \in t')\ x <_\tau z <_\tau y\}$. (The second conjunct excludes gaps containing observation times, guarding against insertion anomalies.) More concretely, if *Pt* is the set $\Re$ of real numbers, δ is some positive number (fixing a level of granularity/degree of error-tolerance) and *Ot* is the set of non-empty open intervals

$$o(p,q) \quad = \quad \{r \in \Re \mid p < r < q\}$$

with rational end-points p, q such that $q - p > \delta$, then for all $o(p,q), o(p',q') \in \mathit{Ot}$,

$$o(p,q)\ \mathsf{S}_\tau\ o(p',q') \quad \text{iff} \quad 0 \leq p' - q \leq \delta\ .$$

Note that *Ot* is countable and S_τ is not functional.

It will suffice throughout this section to equate an event-type with a language $L \subseteq \mathit{Cn}^+$, the strings from which we anchor in a temporal frame as follows. An *event(-token) of event-type* L is a function e from some finite subset $\{t_1, t_2, \ldots, t_k\}$ of *Ot* to *Cn* such that

$$t_1\ \mathsf{S}_\tau\ t_2\ \mathsf{S}_\tau\ \cdots\ \mathsf{S}_\tau\ t_k \quad \text{ and } \quad e(t_1)e(t_2)\cdots e(t_k) \in L\ .$$

To state that e is an event of event-type L, we write

$$e : L\ .$$

For typical choices of S_τ, at most one S_τ-chain can be arranged from a finite subset $\mathit{dom}(e)$ of *Ot*, whence e picks out at a unique string in L. Also, the definition of '$e : L$' can be relativized to a binary relation S_τ that depends on L, but for simplicity, we will leave S_τ fixed in the background. Assuming an observation time can occur at most once in an S_τ-chain (ie $t_1\ \mathsf{S}_\tau\ t_2\ \mathsf{S}_\tau\ \cdots t_k$ and $i \neq j$ imply $t_i \neq t_j$), the totality of events of event-type L is essentially $L\&_{Cn}\mathcal{L}(\mathsf{S}_\tau)$ where $\mathcal{L}(\mathsf{S}_\tau)$ is the set of finite S_τ-chains. In general, however, $\mathcal{L}(\mathsf{S}_\tau)$ is not a regular language, and the special role it plays in time-stamping L makes it natural to present an event as a function whose domain is a finite subset of *Ot*.

To form a model from events, we must spell out what contribution an event e makes. A simple answer is that e contributes the set

$$\Delta(e) \quad = \quad \{@(\varphi, t) \mid t \in \mathit{dom}(e) \text{ and } \varphi \in e(t)\}$$

of formulas, where '$@(\varphi, t)$' is some formula saying: φ holds at t. Recalling the formulas $\mathit{time}(x)$, $\mathit{time}(y)$ and $\mathit{hour}(x, y)$ from the end of section 2, we might equate $@(\mathit{time}(x), t)$ with '$x = t$', $@(\mathit{time}(y), t)$ with '$y = t$' and $@(\mathit{hour}(x, y), t)$ with '$\mathit{hour}(x, y)$' (the dependence on t of $\mathit{hour}(x, y)$ being spurious). By contrast, for *read* (and many other Φ-formulas), it is useful to construe *@*(*read*,*t*) as literally the string '$@(\mathit{read}, t)$'. In view of these differences, let us partition Φ into three sets

$$\Phi \quad = \quad \Phi_= \ \cup\ \Phi_g \ \cup\ \Phi_l$$

where $\Phi_=$ consists of Φ-formulas such as $\mathit{time}(x)$ that translate to equations, Φ_g consists of "global" Φ-formulas such as $\mathit{hour}(x, y)$ that are independent of t, and Φ_l consists of "local" Φ-formulas such as *read*. From $\Phi_=$-formulas in e, we form the *substitution*

$$\theta_e \quad = \quad \{(x, t) \mid t \in \mathit{dom}(e) \text{ and } \mathit{time}(x) \in e(t)\}$$

which we then apply to the rest of the Φ-formulas in e to get

$$\begin{aligned} \Delta_g(e) \quad &= \quad \{\varphi[\theta_e] \mid (\exists t \in \mathit{dom}(e))\ \varphi \in e(t) \cap \Phi_g\} \\ \Delta_l(e) \quad &= \quad \{@(\varphi[\theta_e], t) \mid t \in \mathit{dom}(e) \text{ and } \varphi \in e(t) \cap \Phi_l\} \end{aligned}$$

with the understanding that $\varphi[\theta_e]$ is falsum $\perp$ if θ_e is not functional, else $\varphi[\theta_e]$ is φ with every variable $x \in \mathit{dom}(\theta_e)$ replaced by $\theta_e(x)$.[1]

[1] A substitution pairs variables (or parameters) with terms — the terms in this case being observation times. θ_e provides a means of binding variables $x \in \mathit{dom}(\theta_e)$ locally to e, allowing multiple instantiations of x to co-exist (in a model). That is, instead of proliferating alphabetic variants of an event-type, the event-type is construed as *parametric*, with x as a temporal parameter that an event e instantiates as $\theta_e(x)$. The thematic arguments of an event-type (e.g. agent, patient) can also be presented as parameters, provided an event specifies instantiations of these parameters. For the sake of simplicity, parameters and substitutions are confined here to observation times.

3.1. Lumping events into forcing-conditions

Next, generalizing from event(-token)s to the set $\mathit{Ot} \rightharpoonup \mathit{Cn}$ of partial functions $(p, q, \ldots)$ from *Ot* to *Cn*, let $\preceq$ be the partial order on $\mathit{Ot} \rightharpoonup \mathit{Cn}$ comparing information content as follows

$$\begin{aligned} p \preceq q \quad \text{iff} \quad & \mathit{dom}(p) \subseteq \mathit{dom}(q) \text{ and} \\ & (\forall t \in \mathit{dom}(p))\ p(t) \subseteq q(t)\ . \end{aligned}$$

(The intuition is that $p \preceq q$ says q is at least as informative as p.) Given a collection **ET** of event-types and a partial function $p : \mathit{Ot} \rightharpoonup \mathit{Cn}$, let $\mathbf{ET}(p)$ be the set of **ET**-events $\preceq$-contained in p

$$\mathbf{ET}(p) \quad = \quad \{e \mid e \preceq p \text{ and } (\exists L \in \mathbf{ET})\ e : L\}\ ,$$

this being the bit of reality **ET** carves up from p. Fix an expansion $\mathbf{Time} = (\mathit{Ot}, \mathsf{S}_\tau, \ldots)$ of $(\mathit{Ot}, \mathsf{S}_\tau)$ to the *vocabulary* (aka signature, language, set of non-logical symbols) of Φ_g so that every $\psi \in \Phi_g$ can be judged to be true or false in **Time**. Also, to extract a model from a partial function $p : \mathit{Ot} \rightharpoonup \mathit{Cn}$, we have to be careful about overlapping observation times, which we henceforth assume is given by a family $\mathbf{O}_\tau \subseteq \mathit{Pow}(\mathit{Ot}) - \{\emptyset\}$. (For example, if $\mathit{Ot} \subseteq \mathit{Pow}(\mathit{Pt}) - \{\emptyset\}$, then $\mathbf{O}_\tau$ consists of all non-empty families $T \subseteq \mathit{Ot}$ such that $\bigcap T \neq \emptyset$.) Let

(i) $P(\mathbf{ET}, \mathbf{Time}, \mathbf{O}_\tau)$ be the set of partial functions $p : \mathit{Ot} \rightharpoonup \mathit{Cn}$ such that for all $e \in \mathbf{ET}(p)$ and $\psi \in \Delta_g(e)$,

$$\mathbf{Time} \ \models\ \psi\ ,$$

and for all $T \in \mathbf{O}_\tau$ where $T \subseteq \mathit{dom}(p)$,

$$\bigcup_{t \in T} p(t) \ \in\ \mathit{Cn}$$

(ii) $\mathsf{v}(\Phi)$ be the vocabulary consisting of unary relation symbols $@(\psi, \cdot)$ for every $\psi \in \Phi_l$ (with '$@(\varphi, t)$' to be read: 'φ holds at t'), and

(iii) $f_{\mathbf{ET}}$ be the function that maps a partial function $p : \mathit{Ot} \rightharpoonup \mathit{Cn}$ to

$$f_{\mathbf{ET}}(p) \quad = \quad \bigcup_{e \in \mathbf{ET}(p)} \Delta_l(e)\ .$$

Let us define a $\Phi(\mathbf{Time})$-*model* to be an expansion of **Time** to the vocabulary $\mathsf{v}(\Phi, \mathit{Ot})$ $(= \mathsf{v}(\Phi) \cup \mathit{Ot})$ where the constants t (in *Ot*) are interpreted as t. (To simplify notation, we do not distinguish between t qua constant, and t qua semantic interpretation.) Now, given an element $\hat{p}$ of $P(\mathbf{ET}, \mathbf{Time}, \mathbf{O}_\tau)$, how might we build a $\Phi(\mathbf{Time})$-model that contains $\mathbf{ET}(\hat{p})$? Allowing for $\Phi(\mathbf{Time})$-models that

may or may not contain events beyond those in $\mathbf{ET}(\hat{p})$, let us pass from $\hat{p}$ to a set $P \subseteq P(\mathbf{ET}, \mathbf{Time}, \mathbf{O}_\tau)$ with $\preceq$-least element $\hat{p}$. (To restrict events to those in $\mathbf{ET}(\hat{p})$, take $P = \{\hat{p}\}$.) Applying the "forcing" machinery in, for example, [8],[2] let us define a *forcing* predicate $||\!\vdash_P$ (which we simplify to $||\!\vdash$) that relates *forcing-conditions* $p \in P$ to $\mathsf{v}(\Phi, \mathit{Ot})$-formulas (closed under $\neg$, $\vee$ and $\exists$) as follows:

(i) basing $||\!\vdash$ on $f_{\mathbf{ET}}$,

$$p \,||\!\vdash @(\psi, t) \quad \text{iff} \quad @(\psi, t) \in f_{\mathbf{ET}}(p)$$

for all $\psi \in \Phi_l$ and $t \in \mathit{Ot}$

(ii) confining our search for $\neg\varphi$-counterexamples to q's in P such that $p \preceq q$,

$$p \,||\!\vdash \neg\varphi \quad \text{iff} \quad \text{not } (\exists q \in P)\ (p \preceq q \text{ and } q \,||\!\vdash \varphi)$$

(iii) externalizing $\vee$ to non-deterministic choice $+$,

$$p \,||\!\vdash \varphi \vee \varphi' \quad \text{iff} \quad p \,||\!\vdash \varphi \text{ or } p \,||\!\vdash \varphi'$$

(iv) restricting $\exists$ to *Ot*,

$$p \,||\!\vdash \exists x\varphi \quad \text{iff} \quad p \,||\!\vdash \varphi[x/t] \text{ for some } t \in \mathit{Ot}\ .$$

3.2. Between events and worlds

Applied twice, $\neg$ yields a notion of *weak forcing* $||\!\vdash^w$

$$p \,||\!\vdash^w \varphi \quad \text{iff} \quad p \,||\!\vdash \neg\neg\varphi$$

that extends $||\!\vdash$ in a manner that can be characterized by $\Phi(\mathbf{Time})$-models generated by certain subsets of P. More specifically, a subset G of P is *(P-)generic* if for all $p, q \in P$,

(i) whenever $p \in G$ and $q \preceq p$, $q \in G$

(ii) whenever $p, q \in G$, there exists $r \in G$ such that $p \preceq r$ and $q \preceq r$

(iii) for every $\mathsf{v}(\Phi, \mathit{Ot})$-formula φ, there exists $r \in G$ such that $r \,||\!\vdash \varphi$ or $r \,||\!\vdash \neg\varphi$.

Let us record fundamental results explained in [8] as

Fact 5.

(a) A generic set G induces a $\Phi(\mathbf{Time})$-model $\mathbf{Time}(G)$ such that for every $\mathsf{v}(\Phi, \mathit{Ot})$-formula φ,

$$\mathbf{Time}(G) \models \varphi \quad \textit{iff} \quad (\exists p \in G)\ p \,||\!\vdash \varphi\ .$$

(b) Assuming Φ_l and Ot are countable,

$$p \,||\!\vdash^w \varphi \quad \textit{iff} \quad \textit{for every generic } G \textit{ s.t. } p \in G,\ \mathbf{Time}(G) \models \varphi\ .$$

Forcing-conditions p span the divide between events $e \in \mathbf{ET}(p)$ and $\Phi(\mathbf{Time})$-models $\mathbf{Time}(G)$, for generic $G \ni p$. Given $P \subseteq P(\mathbf{ET}, \mathbf{Time}, \mathbf{O}_\tau)$, let $\mathbf{MOD}(P)$ be the set of $\Phi(\mathbf{Time})$-models generated by P-generic sets

$$\mathbf{MOD}(P) \quad = \quad \{\mathbf{Time}(G) \mid G \text{ is } P\text{-generic}\}\ ,$$

and (going down $\preceq$, rather than [as is the case for generic sets] up), let $P_{\mathbf{ET}}$ be P with all its **ET**-events

$$P_{\mathbf{ET}} \quad = \quad P \,\cup\, \bigcup_{p \in P} \mathbf{ET}(p)\ .$$

The following proposition is easily proved.

Proposition 6. *If $P \subseteq P(\mathbf{ET}, \mathbf{Time}, \mathbf{O}_\tau)$, then*

$$\begin{aligned} P_{\mathbf{ET}} \quad &\subseteq \quad P(\mathbf{ET}, \mathbf{Time}, \mathbf{O}_\tau) \\ \mathbf{MOD}(P_{\mathbf{ET}}) \quad &= \quad \mathbf{MOD}(P) \end{aligned}$$

and $||\!\vdash_P$ is the restriction of $||\!\vdash_{P_{\mathbf{ET}}}$ to P.

Why bother forming $P_{\mathbf{ET}}$? Because in $P_{\mathbf{ET}}$, events $e \in \bigcup_{p \in P} \mathbf{ET}(p)$ count as forcing-conditions, allowing us to ask of a $\mathsf{v}(\Phi, \mathit{Ot})$-formula φ whether or not e forces φ. But beyond φ of the form $@(\psi, t)$, what else is there to '$e \,||\!\vdash \varphi$'? Simplifying notation, let us henceforth assume $P = P_{\mathbf{ET}}$. Observe that for every **ET**-event $e \in P$, $\psi \in \Phi_l$ and $t \in \mathit{Ot}$,

$$e \,||\!\vdash @(\psi, t) \quad \text{iff} \quad t \in \mathit{dom}(e) \text{ and } \psi \in e(t)$$

and if Φ is closed under negations $\sim\psi$ (with $\{\psi, \sim\psi\} \notin \mathit{Cn}$),

$$e \,||\!\vdash @(\sim\psi, t) \quad \text{implies} \quad e \,||\!\vdash \neg @(\psi, t)\ ,$$

the converse of which does not, in general, hold. Readers familiar with [14] might liken the discrepancy here to that between constructible falsity ($\sim$) and intuitionistic negation ($\neg$). More specifically, $\neg$ brings the full space P of forcing-conditions into the picture, denying the existence of a $\preceq$-extension p of e in P such that $p \,||\!\vdash @(\psi, t)$, whereas $\sim$ requires local, positive evidence e. Indeed, the double negation translation $\neg\neg$ underlying $||\!\vdash^w$ weakens the requirement of local, positive evidence to

$$e \,||\!\vdash \neg\neg\varphi \quad \text{iff} \quad (\forall p \in P \text{ s.t. } e \preceq p)\ (\exists q \in P \text{ s.t. } p \preceq q)\ q \,||\!\vdash \varphi\ ,$$

[2] In the terminology of [8], we get a *forcing property* $\langle P, \leq, f\rangle$ (over the base vocabulary $\mathsf{v}(\Phi)$ and set *Ot* of constants) where $\leq$ is the restriction of $\preceq$ to P, and f is a function with domain P mapping $p \in P$ to $f(p) = f_{\mathbf{ET}}(p) \cup \{\text{'}t = t\text{'} \mid t \in \mathit{Ot}\}$. As our only constants are those from *Ot* (no two of which are to be semantically identified), equality is trivial and is accordingly left out above.

It is perhaps also worth noting that [8] allows infinitary disjunctions $\bigvee$, which should come in handy for infinitary $+$. As it is, we can (in line with a finite-state perspective) limit the forcing-conditions in the present section to finite functions, provided we do not require that a generic set be represented by a single forcing-condition.

allowing for the possibility that $e \ ||\!\vdash^w \ @(\psi, t)$ but not $e \ ||\!\vdash \ @(\psi, t)$. That is, e may not be enough to settle '$e \ ||\!\vdash \neg\neg@(\psi, t)$', although arguably, if $e \ ||\!\vdash \ \neg\neg\varphi$, then, as $e \preceq e \in P$, there *is* positive evidence $q \succeq e$ in P for φ (except that it alone may not suffice). At stake between $\sim$ and $\neg$ is the distinction between *ex*plicit and *im*plicit information. '$e \ ||\!\vdash @(\sim\psi, t)$' says that explicit in e is information for $\sim\psi$ at t, whereas '$e \ ||\!\vdash \neg@(\psi, t)$' claims only that information against ψ at t can be *inferred* from e, possibly with the aid of P. The question arises: how do we pick P and/or **MOD**(P)? I hope to report on this matter elsewhere.

4. Conclusion

The finite-state approach to event semantics above is presented in two parts: one centered around event-*types*, formulated as finite automata, or more abstractly, regular languages; and the other around event-*tokens*, grounding strings from an event-type in a model. The strings are built from a set Φ of formulas, which correspond to the propositional fluents of [10]. It is natural to ask: where in the present approach are the situations? Given a generic set G and a time $t \in$ *Ot*, one might expect to reconstruct a situation, understood as "the complete state of the universe at an instant of time" [10], from the snapshot

$$\{\varphi \mid (\exists p \in G)\ p \ ||\!\vdash @(\varphi, t)\}.$$

Evidently, the presentation above is not oriented around situations. Much more prominent is partiality, embodied in the first part by $\times_{Cn}/\&_{Cn}$ (playing the role of conjunction in a Davidsonian approach to event modification), and in the second part by $\preceq$ (linking events to worlds, to give the pictures in Φ model-theoretic bite).

Speaking of partiality, it bears stressing that the present approach puts finite automata in the service of declarative, as opposed to operational, semantics, shying away from details of *how* language is processed. Despite this limitation, I do think that

(i) a useful temporal ontology reflecting "ways of viewing" ([18]) can be fashioned from finite automata (perhaps with help from [19, 13]), and that

(ii) the restriction to regular languages ought to have positive consequences for both the representational and inferential aspects of the *frame problem* ([10]) for natural language ([18]).

Obviously, there is much work to be done. As to what has been carried out, I close with the note that it was conceived as part of a model-theoretic re-interpretation of *propositions-as-types* (applied to natural language in [16]), pushing typed λ-calculi analyses of logical connectives down to the sub-atomic (lexical) level through finite-state methods. (The interested reader is referred to the *constructive eventuality assumption* in [4], §3.1.)

References

[1] N. Chang, D. Gildea, and S. Narayanan. A dynamic model of aspectual composition. In *Proc. CogSci 98*, 1998.

[2] D. Davidson. The logical form of action sentences. In N. Rescher, editor, *The Logic of Decision and Action*. University of Pittsburgh Press, 1967.

[3] D. R. Dowty. *Word Meaning and Montague Grammar*. Reidel, Dordrecht, 1979.

[4] T. Fernando. Conservative generalized quantifiers and presupposition. In *Proc. Semantics and Linguistic Theory XI*. Cornell University, 2001.

[5] F. Hamm and M. van Lambalgen. Event calculus, nominalization, and the progressive. Available from `www.semanticsarchive.net`, 2000.

[6] H. Kamp and U. Reyle. *From Discourse to Logic*. Kluwer Academic Publishers, Dordrecht, 1993.

[7] L. Karttunen. Presupposition and linguistic context. *Theoretical Linguistics*, pages 181–194, 1974.

[8] H. J. Keisler. Forcing and the omitting types theorem. In M. Morley, editor, *Studies in Model Theory*. The Mathematical Association of America, 1973.

[9] M. Krifka. Nominal reference, temporal constitution and quantification in event semantics. In R. Bartsch, J. van Benthem, and P. van Emde Boas, editors, *Semantics and Contextual Expressions*. Foris, Dordrecht, 1989.

[10] J. McCarthy and P. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In M. Meltzer and D. Michie, editors, *Machine Intelligence 4*. Edinburgh University Press, 1969.

[11] M. Moens and M. Steedman. Temporal ontology and temporal reference. *Computational Linguistics*, 14:15–28, 1988.

[12] S. Narayanan. Reasoning about actions in narrative understanding. In *Proceedings of IJCAI '99*. Morgan Kaufmann, San Francisco, 1999.

[13] R. Naumann. Aspects of changes: a dynamic event semantics. *J. Semantics*, 18:27–81, 2001.

[14] D. Nelson. Constructible falsity. *J. Symbolic Logic*, 14(1):16–26, 1949.

[15] T. Parsons. *Events in the Semantics of English: A Study in Subatomic Semantics*. MIT Press, Cambridge, MA, 1990.

[16] A. Ranta. *Type-Theoretical Grammar*. Oxford University Press, Oxford, 1994.

[17] R. Reiter. Narratives as programs. In *Principles of Knowlege Representation: Procedings of KR 2000*. Morgan Kaufmann, San Francisco, 2000.

[18] M. Steedman. *The Productions of Time*. Draft, `ftp://ftp.cogsci.ed.ac.uk/pub/steedman/temporality/temporality.ps.gz`, July 2000.

[19] S. Tojo. Event, state and process in arrow logic. *Minds and Machines*, 9:81–103, 1999.

[20] J. van Benthem. A note on dunamic arrow logic. In J. van Eijck and A. Visser, editors, *Logic and Information Flow*. MIT Press, Cambridge, MA, 1994.