

Finite-state descriptions for temporal semantics

Tim Fernando

Computer Science Department, Trinity College, Dublin 2, Ireland

Abstract. Finite-state descriptions for temporal semantics are outlined through which to distinguish soft inferences reflecting manners of conceptualization from more robust semantic entailments defined over models. Just what descriptions are built (before being interpreted model-theoretically) and how they are grounded in models of reality explain (upon examination) why some inferences are soft.

Keywords: finite-state, descriptions, temporal semantics, underspecification

1. Introduction

The usual approach to inference in model-theoretic semantics is to restrict attention to models that satisfy certain formulas, among which are *meaning postulates* stating, for instance, that bachelors are unmarried. Computational costs aside, one might seek to deduce a proposition A from such formulas before simply stipulating A or, for that matter, devising some special-purpose mechanism that spits out A . The question for any given A , however, is whether such a deduction can be carried out with some generality whilst avoiding unwanted entailments.

Consider the pairs (1a,b) and (2a,b) from the critique in Ramsay (1994) of coercion in Moens and Steedman (1988) and co-specification in Pustejovsky (1991). (\models is understood here to denote entailment; $\not\models$ its denial.)

- (1) a. Harry was hiccuping \models Harry hiccupped twice
- b. Henriette was crossing the road $\not\models$ Henriette crossed the road [once, let alone twice]

- (2) a. John baked a cake \models that baking produced that cake
- b. John baked a potato $\not\models$ that baking produced that potato

Ramsay claims that suitable meaning postulates account for the a/b contrasts. Very briefly, road crossing is temporally extended “whereas hiccuping is (conceptually) instantaneous” (Ramsay, 1999); and cakes are produced by baking, potatoes are not. No rocket science here; only a modicum of common sense that underlies natural language understanding. But will an encoding by meaning postulates do?

Reflecting on (1a), it is not clear that the implication is unproblematic. Just as a bus may run over Henriette before she reaches the other

side of the road, so too might some irritable room-mate put Harry out of his misery before his second hiccup culminates.¹

- (3) Harry was hiccuping when Sally shot him dead.

Far-fetched or not, the possibility of killing off Harry in the middle of his second hiccup would seem to threaten (1a). We shall return to the progressive in §2.3 below. For now, suffice it to ask: do we want to infer (1a) by ordinary reasoning from meaning postulates?

I think we do not. (1a) holds only with a pesky qualification that is far from incontrovertible. Ramsay’s statement “hiccuping is (conceptually) instantaneous” is drawn out in Ramsay (1994) as

hiccuping events don’t [take time] (or rather ... we don’t think about the time they take).

Conceptualizations such as events can be very fragile. Example (4) is from Verkuyl (2000).

- (4) a. Piet warned him three times.
 b. What Piet did three times was warn him.
 c. What Piet did was to warn him three times.

Asking how many things Piet did (i.e. how many events Piet acted in) is as tricky as asking how many time units there are in a minute. What time *units*? If by time units, we mean seconds, then the answer is 60; if we mean hours, then the answer is 0 or rather, $\frac{1}{60}$. Talk of events in the abstract is slipperier still than talk of stretches of time, with units unspecified. The fragility of aspect is well-known; in Steedman (2000), we read

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

Although the extent to which language and reality can be separated is debatable, I think it is useful to say cakes and potatoes inhabit the world in a way that progressives do not. Surely, the change in (5) from cakes to potatoes has consequences that can be put down to worldly facts about cakes versus potatoes.

- (5) a. John baked a cake.
 b. John baked a potato.

¹ I am indebted to Martin Emms for this application of the *imperfective paradox* (Dowty, 1979).

The question about (5) brought out in (2) is how can a cake be produced in (5a), but a potato not in (5b)? Ramsay (1994) writes

It could be, of course, that *bake* is an ambiguous item and that the properties of *cake* and *potato* select between its possible interpretations. This is such an unattractive solution that I will not consider it further.

Ramsay’s solution is to add meaning postulates stating that cakes are produced by baking, and that

(†) no cake can get baked twice.

Notice that if we reject (†), then we deny the general principle from which to infer the creation of a cake in (5a). In this case, the contrast between (5a) and (5b) to explain is the inference that (5b) produces *no* potato. The focus then shifts to a meaning postulate ruling out the creation of a potato by baking.

So why should it be acceptable to apply meaning postulates in (5), but not in (1)? Because the issue of truth in (5) concerns stuff in the world (viz. cakes and potatoes) whereas that in (1) has to do with linguistic conventions reflecting conceptualization. Potatoes and cakes are *not*, in the main, the fragile conceptualizations that instantaneous events are. To say that potatoes are “ways of viewing a happening” is to invite a discussion of vagueness (borderline potatoes) and/or figurative uses (“John is a potato”) that (although extremely interesting) I trust we can, for the present, put aside. The distinction between the world and “ways of viewing a happening” is at the root of the tension between formal (model-theoretic) semantics, oriented around the external world, and conceptual (cognitive) semantics, centered around mental representations (e.g. Jackendoff, 1991). The former emphasizes the structures that model the external world, distinct from the descriptions/formulas that the mental representations of the latter come to. Cramming progressives alongside potatoes in one model-theoretic universe risks conflating structure with description.

But so what? Supposing we care only to draw inferences, and have no interest in classifying them, should we bother distinguishing description from structure? Important work in computational linguistics depends on such a distinction. Labelling some of that work the “Description Theory of linguistic representation,” Muskens (2001) argues that

The move from structures and truth conditions to descriptions of structures and their truth conditions offers a uniform and natural way to *underspecify* syntax and semantics . . .

Applying underspecification to the topic at hand, consider (6).

(6) a. It rained *for* an hour.

- b. It rained *in* an hour. [began to rain]

If treating “bake” in (5a) and (5b) differently is (as Ramsay claims) so “unattractive” then equally would not distinct analyses of “rain” in (6a) and (6b) be as well? Underspecification allows us to associate a single representation with “rain” that can then be fleshed out in opposing directions by the “for” and “in” temporal modifications in (6a) and (6b). This is shown in section 2 below, where the parallel with (5) is developed using regular languages as representations to account for well-known entailment patterns illustrated by (7) and (8).

- (7) a. Carl drank beer for an hour.
 b. Carl was drinking beer \models Carl drank beer
- (8) a. Carl drank six pints *in*/*for an hour.
 b. Carl was drinking six pints $\not\models$ Carl drank six pints

The account in section 2 is an ontologically innocent alternative to mereological approaches such as Krifka (1989).

As for the entailment (1a) from “Harry was hiccuping” to “Harry hiccuped twice,” even if we grant that the entailment there is soft (in view of counter-examples such as (3)), the tendency to accept it cries out for some explanation. An attempt at this is made in section 3, where issues of granularity and iteration are taken up. Temporal instants are turned into intervals, and iterations are analyzed in what Westerstahl (1989) calls “logic with free quantifier variables” (which, as far as axiomatization goes, is essentially no harder than first-order logic). The application here of generalized quantifiers (albeit underspecified) is consonant with the objection in Verkuyl (2000) to Davidson’s “wish to restrict his formalism to a first-order language.” Verkuyl questions the usefulness of Davidsonian events in compositionally piecing together a verb and its arguments (what he calls “inner aspectuality”), a problem he attacks instead through certain successor, path and participancy functions (having set aside any qualms about higher-order objects). Jointly, these functions may yield not just one event (that might be existentially quantified à la Davidson, 1967) but any number of events (taking us into the realm of pluralities).² Verkuyl’s claim that “it is only after the construal of the V[erb]P[hrase] information that one may begin to speak about events” suggests that during VP assembly, events are underspecified.

Ideas of Verkuyl’s have played an important role in work by Naumann (e.g. 2001), a finite-state form of which is developed in Fernando

² The same move to sets of events is advocated in Ramsay (1994).

(2002), drawing heavily on Steedman (2000). The present paper elaborates features of Fernando (2002) that pertain to underspecification and to more general computational concerns. The label “finite-state” reflects an interest in computational methods short of predicate logic. (This explains, in part, my reluctance above to embrace meaning postulates at every turn.) The claim is that useful finite-state descriptions can be constructed *without* arbitrarily complicated semantic inferencing. Section 2 below can be read as exploring some form of a *Grammatically Relevant Subsystem* hypothesis (Pinker, 1989). To appreciate the full semantic content of these finite-state descriptions (irrespective of grammatical relevance), section 3 grounds these descriptions in models of reality, uncovering inferential properties outside the scope of section 2. Underlying the division between sections 2 and 3 is the contention that so long as we distinguish the construction of logical forms from open-ended inference on them, a computational linguist need not worry if the logical forms s/he constructs go beyond first-order logic. Davidson’s extensional first-order bias is (I claim) not as computationally compelling as it may, at first blush, seem.

2. Building finite-state descriptions

The present section develops finite-state descriptions given by finite strings of observations in chronological order. Viewed as motion pictures, they are accepted by cameras that can be formulated as finite automata or finite Kripke models. Related mechanistic conceptions have been described in Tojo (1999), where models more general than finite automata are used, and in Chang et al. (1998), where a cognitive processing picture with “simulative inference” is presented. As interesting as these directions are, we shall focus here on the languages (ie string sets) at stake³ — a more abstract perspective insofar as any number of computational devices may capture the same language.

The regularity of these languages can be made evident by presenting them through regular expressions. The symbols in our alphabet are the still-pictures that, strung together, form our motion pictures. More formally, given some finite set Φ of formulas such as *dawn*, *rain*, and *dusk*, a symbol is a subset of Φ such as \emptyset , $\{\textit{rain}\}$ and $\{\textit{rain}, \textit{dawn}\}$ that enumerates propositions the associated still depicts. To distinguish sets-as-symbols from sets-as-languages (or, for that matter, from sets-as-strings), let us draw boxes instead of braces for the former, writing

³ These languages encode different information than those considered in ter Meulen (1990)’s application of van Benthem’s semantic automata to aspectual verbs.

\square for \emptyset -as-symbol. We may then translate English to regular languages via a mapping \mathcal{L} satisfying (9).

$$(9) \quad \mathcal{L}(\text{rain from dawn to dusk}) = \boxed{\text{rain, dawn}} \boxed{\text{rain}}^* \boxed{\text{rain, dusk}}$$

For every integer $k \geq 0$, the string $\boxed{\text{rain, dawn}} \boxed{\text{rain}}^k \boxed{\text{rain, dusk}}$ from (9) is a movie that starts with *rain* and *dawn*, ends with *rain* and *dusk*, and in between shows k instances of *rain*. In a similar vein, we have the associations (10)-(11), illustrating general translations of statives and of the prepositions “from” and “to” (used temporally), where r^+ is the concatenation rr^* of r with the Kleene star r^* of r .

$$(10) \quad \mathcal{L}(\text{rain}) = \boxed{\text{rain}}^+$$

$$(11) \quad \begin{aligned} \mathcal{L}(\text{from dawn}) &= \boxed{\text{dawn}} \square^+ \\ \mathcal{L}(\text{to dusk}) &= \square^+ \boxed{\text{dusk}} \end{aligned}$$

Can we derive (9) from (10) and (11)? The next subsection describes an associative binary operation $\&$ on languages such that (10) and (11) yield (9), provided

$$\mathcal{L}(\text{rain from dawn to dusk}) = \mathcal{L}(\text{rain}) \& \mathcal{L}(\text{from dawn}) \& \mathcal{L}(\text{to dusk}).$$

2.1. SUPERPOSITION AND SUBSUMPTION (UNCONSTRAINED)

Given languages $L, L' \subseteq \text{Power}(\Phi)^*$, let us define their *superposition* $L\&L'$ to be

$$L\&L' = \bigcup_{k \geq 1} \{(\sigma_1 \cup \sigma'_1) \cdots (\sigma_k \cup \sigma'_k) \mid \sigma_1 \cdots \sigma_k \in L, \sigma'_1 \cdots \sigma'_k \in L'\},$$

the intuition being that the union $\sigma_i \cup \sigma'_i$ is the superposition of the two stills σ_i and σ'_i . Like intersection, $\&$ is *not* so readily expressible as a function on regular expressions, but nonetheless maps regular languages to regular languages. To see this, suppose M and M' are finite automata that accept the languages L and L' . Then we can form a finite automata that accepts $L\&L'$ by taking

- (i) as its set of states the (Cartesian) product $Q \times Q'$ of the state sets of M and M' (respectively)
- (ii) as its initial state the pair (q_0, q'_0) of initial states of M and M'
- (iii) as its set of final states the product $F \times F'$ of those of M and M'

(iv) as its transition set the transitions

$$(q, q') \xrightarrow{\sigma \cup \sigma'} (r, r')$$

for all M -transitions $q \xrightarrow{\sigma} r$ and M' -transitions $q' \xrightarrow{\sigma'} r'$.

(i)-(iv) above are identical to the construction for intersection $L \cap L'$ except that in (iv), $L \cap L'$ would require $\sigma = \sigma'$. Like \cap , $\&$ is associative and commutative.

Superposition induces a useful pre-order \supseteq on languages L, L' defined as follows. L *subsumes* L' , written $L \supseteq L'$, if $L \subseteq L \& L'$

$$L \supseteq L' \quad \text{iff} \quad L \subseteq L \& L' .$$

It is not difficult to see that \supseteq is reflexive and transitive, and that

$$L \supseteq L' \& L'' \quad \text{iff} \quad L \supseteq L' \quad \text{and} \quad L \supseteq L'' .$$

That is, $L \supseteq L'$ can be understood intuitively as saying: L has at least as much information content as L' .⁴

To make this intuition precise, we shall impose constraints on superposition that we shall then apply to capture certain restrictions implicit in temporal modifications. More specifically, consider the verb phrase “rain for an hour,” which we might handle according to (12), where (the underspecified constructs) x and y are variables (subject to instantiation described in section 3).

$$(12) \quad \mathcal{L}(\text{rain for an hour}) = \boxed{\text{rain, time}(x)} \boxed{\text{rain}}^* \boxed{\text{rain, time}(y), \text{hour}(x, y)}$$

An obvious derivation of (12) would appeal to (13)-(14) and (10).

$$(13) \quad \mathcal{L}(\text{an hour}) = \boxed{\text{time}(x)} \square^+ \boxed{\text{time}(y), \text{hour}(x, y)}$$

$$(14) \quad \mathcal{L}(\text{rain for an hour}) = \mathcal{L}(\text{rain}) \& \mathcal{L}(\text{an hour})$$

The question arises: what about $\mathcal{L}(\text{for})$?

The semantic content of “for” is often illustrated by contrasting it with “in,” as in (8a), “Carl drank six pints in/*for an hour.” The oddness of “for” in (8a) is based on a so-called *telic* reading of “Carl drank six pints” under which a proposition such as *six-pints-drunk*

⁴ An obvious alternative to $L \supseteq L'$ would be to strengthen the requirement $L \subseteq L \& L'$ to $L = L \& L'$. But then we would lose reflexivity (consider $L = \boxed{p} + \boxed{q}$) as well as the property that L subsumes all L' and L'' such that $L = L' \& L''$.

holds only at the end of the hour in question (and not before). That is, once the six pints Carl drank are drunk, the event described is over. Now, generalizing from (8a), a first stab at defining the language of a sentence S modified by a temporal interval I using “in” and “for” is (15).

$$(15) \quad \begin{aligned} \mathcal{L}(S \text{ in } I) &\approx \begin{cases} \mathcal{L}(S) \ \& \ \mathcal{L}(I) & \text{if } \mathcal{L}(S) \text{ is 'telic'} \\ \emptyset & \text{otherwise} \end{cases} \\ \mathcal{L}(S \text{ for } I) &\approx \begin{cases} \mathcal{L}(S) \ \& \ \mathcal{L}(I) & \text{if } \mathcal{L}(S) \text{ is not 'telic'} \\ \emptyset & \text{otherwise} \end{cases} \end{aligned}$$

Implicit in (15) is the idea that S is odd if its language $\mathcal{L}(S)$ is empty, $\mathcal{L}(S) = \emptyset$. Leaving aside exactly what “telic” means, notice that (15) cannot account for cases where both “in” and “for” modification are acceptable, albeit with different meanings, as in (6ab), “rain for/in an hour.”

2.2. CONSTRAINTS AND “IN”/“FOR” MODIFICATION

A variant of (15) that we shall develop in this subsection is (16), which depends on the following ingredients:

- (i1) a refinement $\&_{\Sigma}$ of $\&$, relative to some subfamily $\Sigma \subseteq \text{Power}(\Phi)$
- (i2) certain functions *telic* and *iter* (associated with “in” and “for” respectively) on languages.

$$(16) \quad \begin{aligned} \mathcal{L}(S \text{ in } I) &= \mathcal{L}(S) \ \&_{\Sigma} \ \text{telic}(\mathcal{L}(S)) \ \&_{\Sigma} \ \mathcal{L}(I) \\ \mathcal{L}(S \text{ for } I) &= \mathcal{L}(S) \ \&_{\Sigma} \ \text{iter}(\mathcal{L}(S)) \ \&_{\Sigma} \ \mathcal{L}(I) \end{aligned}$$

Let us describe (i1) and (i2) in turn.

To understand (i1), recall that a symbol from which we form our strings is a set of propositions — i.e. a subset of Φ — that describes a still-picture. But can any $\sigma \subseteq \Phi$ describe a still? To get sensible motion pictures, we had better require that all of the formulas in σ can be realized simultaneously. This would rule out sets such as $\boxed{\text{dawn}, \text{dusk}}$ (unless *dawn* is allowed to coincide with *dusk*). Now, the idea is to let Σ be some subcollection of subsets of Φ from which we can string together motion pictures; that is, Σ constitutes a (static) notion of consistency. Beyond requiring that $\boxed{\text{dawn}, \text{dusk}} \notin \Sigma$, let us assume that Φ is closed under negations $\sim \cdot$ such that (17) holds, and that Σ is closed under subsets, as stated in (18).

$$(17) \quad (\forall \varphi \in \Phi) \quad \boxed{\varphi, \sim \varphi} \notin \Sigma$$

$$(18) \quad (\forall \sigma \in \Sigma)(\forall \sigma' \subseteq \sigma) \sigma' \in \Sigma$$

Notice that if $\Sigma \neq \text{Power}(\Phi)$, then the superposition $L\&L'$ may fail to be a subset of Σ^* . To be assured of a subset of Σ^* , let us intersect $L\&L'$ with Σ^+ , defining

$$L \&_{\Sigma} L' = (L\&L') \cap \Sigma^+ .$$

Since regular languages are closed under intersection, it follows that $\&_{\Sigma}$ maps regular languages to regular languages. A more economical construction of automata for $L\&L'$ would draw on the idea that two stills σ and σ' can be superposed precisely if $\sigma \cup \sigma' \in \Sigma$. The equation

$$L \&_{\Sigma} L' = \bigcup_{k \geq 1} \{(\sigma_1 \cup \sigma'_1) \cdots (\sigma_k \cup \sigma'_k) \mid \sigma_1 \cdots \sigma_k \in L, \\ \sigma'_1 \cdots \sigma'_k \in L' \text{ and for } 1 \leq i \leq k, \sigma_i \cup \sigma'_i \in \Sigma\}$$

suggests forming an automata for $L\&_{\Sigma}L'$ from automata M and M' for L and L' (respectively) according to clauses (i)-(iii) above, replacing (iv) by a transition relation $\rightsquigarrow \subseteq (Q \times Q') \times \Sigma \times (Q \times Q')$ given by

$$(q, q') \rightsquigarrow (r, r') \quad \text{iff} \quad (\exists \tau, \tau') \sigma = \tau \cup \tau' \text{ and } q \xrightarrow{\tau} r \text{ and } q' \xrightarrow{\tau'} r'$$

for all $\sigma \in \Sigma$, $q, r \in Q$ and $q', r' \in Q'$. Observe that $\&_{\Sigma}$ is associative, commutative, and that if $\Sigma' \subseteq \Sigma$ then

$$(L\&_{\Sigma}L')\&_{\Sigma'}L'' = (L\&L'\&L'') \cap \Sigma'^*$$

for all $L, L', L'' \subseteq \text{Power}(\Phi)^+$.

Turning to (i2), for every language L , let $\omega(L)$ be the set of propositions that must hold at the end of L ; that is,

$$\omega(L) = \{\varphi \in \Phi \mid (\forall s \in L - \{\epsilon\}) \varphi \in \text{end}(s)\}$$

where ϵ is the empty string, and for every non-empty string s , $\text{end}(s)$ is s 's last symbol. For example,

$$\omega(\boxed{\text{rain}}^+) = \boxed{\text{rain}} = \omega(L\boxed{\text{rain}}) \quad \text{for any } L$$

and we may expect something like

$$\text{six-pints-drunk} \in \omega(\mathcal{L}(\text{Carl drank six pints})).$$

Moreover, the telicity of ‘‘Carl drank six pints’’ discussed above suggests that for any symbol σ that occurs before the end of $\mathcal{L}(\text{Carl drank six pints})$,

$$\text{six-pints-drunk} \notin \sigma.$$

To move the negation on \notin over to propositions and symbols (i.e. sets of propositions), let us define the *negation* $\bar{\sigma}$ of a subset $\sigma \subseteq \Phi$ by

$$\begin{aligned} \bar{\square} &= \Phi \quad (\notin \Sigma) \\ \overline{\varphi_1, \dots, \varphi_n} &= \boxed{\sim \varphi_1} + \dots + \boxed{\sim \varphi_n} \quad \text{for } n \geq 1 \end{aligned}$$

(writing $+$ for non-deterministic choice; i.e. union). Now, let

$$\text{telic}(L) = \overline{\omega(L)}^+ \square$$

and let us agree to call L *telic* if L subsumes $\text{telic}(L)$. It follows that if a language L subsumes $\boxed{\sim \varphi}^+ \boxed{\varphi}$ for some proposition $\varphi \in \Phi$, then L is telic. For the case $L = \mathcal{L}(\text{Carl drank six pints})$, the obvious candidate for φ is *six-pints-drunk*. In general, for any S such that $\omega(\mathcal{L}(S)) = \boxed{\varphi}$, (16) gives

$$\begin{aligned} \mathcal{L}(S \text{ in } I) &= \mathcal{L}(S) \ \&\Sigma \ \boxed{\sim \varphi}^+ \square \ \&\Sigma \ \mathcal{L}(I) \\ &= \mathcal{L}(S) \ \&\Sigma \ \mathcal{L}(I) \quad \text{if } \mathcal{L}(S) \text{ is telic.} \end{aligned}$$

If we drop the assumption $\omega(\mathcal{L}(S)) = \boxed{\varphi}$ then (16) yields

$$\mathcal{L}(S) \ \&\Sigma \ \mathcal{L}(I) \subseteq \mathcal{L}(S \text{ in } I) \quad \text{if } \mathcal{L}(S) \text{ is telic}$$

where \subseteq above is as good as $=$, given (19).

(19) If $L \subseteq L \ \&\Sigma \ L'$ then L and $L \ \&\Sigma \ L'$ are equivalent up to Σ -entailment.

Without getting into the details of Σ -entailment mentioned in (19),⁵ let us make (19) plausible. That $L \ \&\Sigma \ L'$ should Σ -entail L is a consequence of the conjunctive character of $\&\Sigma$. The converse (L Σ -entails $L \ \&\Sigma \ L'$) depends on the subset relation ($L \subseteq L \ \&\Sigma \ L'$) assumed, and on a construal of languages as sets of possibilities that become more informative as possibilities get ruled out.

Next, “for” modification under (16) presupposes a function iter, which we can now identify

$$\text{iter}(L) = \omega(L)^+ .$$

It is immediate from (17) and (18) that

$$\text{telic}(L) \ \&\Sigma \ \text{iter}(L) = \emptyset .$$

Thus, if we call a language L that subsumes $\text{iter}(L)$ *iterative*, and write $L = L' \text{ mod } \Sigma$ for “ L is equivalent to L' up to Σ -entailment” then (16) sharpens (15) to (20).

⁵ Beyond the hints from section 3 below, this notion is fleshed out in www.cs.tcd.ie/Tim.Fernando/jlc.pdf.

$$(20) \quad \begin{aligned} \mathcal{L}(S \text{ in } I) &= \begin{cases} \mathcal{L}(S) \ \&_{\Sigma} \ \mathcal{L}(I) \ \text{mod } \Sigma & \text{if } \mathcal{L}(S) \text{ is telic} \\ \emptyset & \text{if } \mathcal{L}(S) \text{ is iterative} \end{cases} \\ \mathcal{L}(S \text{ for } I) &= \begin{cases} \mathcal{L}(S) \ \&_{\Sigma} \ \mathcal{L}(I) \ \text{mod } \Sigma & \text{if } \mathcal{L}(S) \text{ is iterative} \\ \emptyset & \text{if } \mathcal{L}(S) \text{ is telic} \end{cases} \end{aligned}$$

Notice that there are languages that are neither telic nor iterative. Take, for example, $\Box^* \boxed{\text{rain}}$. Clearly, for every language L ,

$$\text{rain} \in \omega(L) \quad \text{iff} \quad L \supseteq \Box^* \boxed{\text{rain}}.$$

Moreover, if we re-set $\mathcal{L}(\text{rain})$ to $\Box^* \boxed{\text{rain}}$ (revising (10) to (10)'), then not only does (12) still follow from (16) but (16) also equates $\mathcal{L}(\text{rain in an hour})$ with

$$\boxed{\sim \text{rain}, \text{time}(x)} \ \boxed{\sim \text{rain}}^* \ \boxed{\text{rain}, \text{time}(y), \text{hour}(x, y)}$$

which can be read “begin to rain in an hour.”

$$(10)' \quad \mathcal{L}(\text{rain}) = \Box^* \boxed{\text{rain}}$$

Attacking (6) through (10)' is analogous to analyzing “bake OBJ” in (2) by asserting the existence of an OBJ at the end of the baking, but leaving to OBJ the matter of whether or not one exists beforehand. (Yes, if OBJ is potato; no, if OBJ is cake.) (10)' leaves the iterativity/telicity of “rain PREP an hour” to PREP. (Iterative, if PREP is for; telic, if PREP is in.)

Revisiting “rain from dawn to dusk,” we can preserve our original analysis (9) under the revision (10)' of (10) if we re-analyze $\mathcal{L}(\text{rain from dawn to dusk})$ according to

$$\mathcal{L}(S \text{ from } B \text{ to } E) = \mathcal{L}(S) \ \&_{\Sigma} \ \text{iter}(\mathcal{L}(S)) \ \&_{\Sigma} \ \mathcal{L}(\text{from } B) \ \&_{\Sigma} \ \mathcal{L}(\text{to } E)$$

where *iter* is triggered by the interval modification “from B to E .” I confess that it is not clear to me if this re-analysis is an improvement. Or, for that matter, if (10) ought to be revised to (10)'. Be that as it may, let us step back a bit in the next subsection, and explore more generally how $\mathcal{L}(S)$ might be defined. But first, we look more closely at drinking six pints/beer, spelling out what the proposition *six-pints-drunk* comes to, with tense abstracted away.

2.3. FINISHING, STARTING AND A BIT ON THE PROGRESSIVE

The tense-less phrase “Carl drink six pints” describes an accomplishment (in the sense of Vendler, 1967) that, as is well-known in the

literature (e.g. Tenny and Pustejovsky, 2000), can be tracked by the action of the verb on *parts* of a particular argument a . If we write

$$\begin{aligned} v(u) & \text{ for a proposition saying "Carl drink } u\text{"} \\ u \preceq a & \text{ for a proposition saying } u \text{ is a non-null part of } a \\ u \prec a & \text{ for the proposition } u \preceq a \wedge u \neq a \end{aligned}$$

then we may expect that $\mathcal{L}(\text{Carl drink six pints})$ subsumes

$$\boxed{\sim(\exists u \preceq a) v(u)} \quad \boxed{(\exists u \prec a) v(u)}^* \quad \boxed{v(a)}$$

and $\boxed{6\text{pints}(a)}^+$. By contrast, the language for the activity “Carl drink beer” would subsume

$$\boxed{\sim\exists u(\text{beer}(u) \wedge v(u))} \quad \boxed{\exists u(\text{beer}(u) \wedge v(u))}^+$$

with no well-defined termination condition. In other cases of (Vendlerian) achievements (e.g. “Carl arrive”), it is an end $e \in \Phi$ that is supplied, and the stages along the way that are abstracted out, leading to $\boxed{\sim e}^+ \boxed{e}$. In general, given a language $L \subseteq \text{Power}(\Phi)^+$ and a symbol $\sigma \subseteq \Phi$, let us say that σ *finishes* L if L subsumes $\bar{\sigma}^+ \sigma$. Thus, L is telic iff some σ finishes L . Dually, we say σ *starts* L if L subsumes $\sigma \bar{\sigma}^+$. It is reasonable to assume that $\boxed{v(a)}$ finishes $\mathcal{L}(\text{Carl drink six pints})$ ⁶ while $\boxed{\sim(\exists u \preceq a) v(u)}$ starts it. More generally, we might approximate a language L by a precondition $b \in \Phi$ and a postcondition $e \in \Phi$ as in Table I (on page 13), forming an accomplishment by &-conjoining an achievement and an activity

$$\boxed{b} \boxed{\sim b}^+ \& \boxed{\sim e}^+ \boxed{e} = \boxed{b, \sim e} \boxed{\sim b, \sim e}^* \boxed{\sim b, e}$$

as well as a state (lest we forget that $\mathcal{L}(\text{Carl drink six pints})$ also subsumes $\boxed{6\text{pints}(a)}^+$).

The analysis of “Carl drink six pints” above is widely applicable. For instance, we may assume that $\mathcal{L}(\text{Pat swim a mile})$ subsumes $\boxed{\text{mile}(m)}^+$ and

$$\boxed{\sim(\exists u \preceq m) \text{swim}(p, u)} \quad \boxed{(\exists u \prec m) \text{swim}(p, u)}^* \quad \boxed{\text{swim}(p, m)}.$$

The present set-up applies also to accomplishments measured out *not* by a partial order \preceq on the direct object but by some other gradable

⁶ Or, at least, some language equivalent to $\mathcal{L}(\text{Carl drink six pints})$ up to Σ -entailment. Note that $\sim(\exists u \preceq a) v(u)$ implies $\sim v(a)$, and $v(a)$ implies $\sim\sim(\exists u \preceq a) v(u)$.

Table I. Vendler classes induced by beginnings b and endings e ($b, e \in \Phi$)

	L subsumes \dots	Vendler class of L
no change within L	\boxed{b}^+ (b is e)	state (e.g. hate)
\boxed{b} starts L	$\boxed{b} \boxed{\sim b}^+$ (e is $\sim b$)	activity (e.g. swim)
\boxed{e} finishes L	$\boxed{\sim e}^+ \boxed{e}$ (b is $\sim e$)	achievement (e.g. arrive)
\boxed{b} starts and \boxed{e} finishes L	$\boxed{b, \sim e} \boxed{\sim b, \sim e}^* \boxed{\sim b, e}$ ($\boxed{\sim b, \sim e} \in \Sigma$)	accomplishment (e.g. swim a mile)

condition that may involve other arguments (e.g. “push the cart to the park”).

It is instructive to contrast the present proposal with Krifka (1989), where a unary predicate P is defined to be *quantized* if

$$\forall x \forall y (P(x) \wedge y \prec x \supset \sim P(y)).$$

The variables x and y above are understood to range over *objects/tokens*. What is *quantized* is not a token, but a type P , which is comparable to our notion of a telic language L . In place of a mereology on event-tokens x, y over predicate logic, our analysis employs descriptions L, L' , subject to a notion of subsumption that, when beefed up to Σ -entailment, is essentially propositional (as opposed to predicate) logic. As indicated in the next section, the languages L can be interpreted over predicate logic models, but these models need not include event-tokens as objects.⁷ We may, if we wish, introduce event-tokens into our ontology; but our account (so far) does *not* depend on any such commitment. In this sense, our approach is ontologically innocent, compared to Krifka (1989). Rather than appealing to some partial order \preceq on event-tokens, we build on a pre-order \trianglerighteq that supports a measure of underspecification, illustrated above by the analysis of “rain for/in an hour” based on (10)'.⁸

⁷ Exactly what an event-object is, I am not prepared to say. But it certainly is *not* just a string in a language L , and so I hesitate to refer to a language as an event-type (given the common practice of identifying tokens with objects, and types with the set of their tokens). Languages are descriptions, as are the strings in languages. Just what they describe is the subject of section 3 below.

⁸ The underspecification at stake here comes down to the *incompleteness* of Σ -entailment. See www.cs.tcd.ie/Tim.Fernando/jlc.pdf, where the Σ -*complement* $\neg L$ of a language L is defined such that neither L nor $\neg L$ may be Σ -entailed by a language L' .

And speaking of that analysis, Table I suggests some minor amendments. Besides a beginning and an end, an interval may have a middle (which will prove useful below for accommodating the progressive). Accordingly, let us revise **telic** ever so slightly to

$$\begin{aligned} \text{telic}(L) &= \overline{\omega(L)}^+ \square \ \& \ \square \square^+ \square \\ &= \overline{\omega(L)} \ \overline{\omega(L)}^+ \square . \end{aligned}$$

Next, for activities to admit “for”-modification as easily as states, let us redefine **ite** so as to leave the first symbol (i.e. observation) unconstrained before iterating $\omega(L)$

$$\text{ite}(L) = \square \ \omega(L)^+ \ \omega(L) .$$

The boundary conditions on interval modification are, it seems to me, underspecified, which is why we drop the requirement that the very first symbol of a string in an iterative language contain $\omega(L)$. The functions **telic** and **ite** are, as before, complementary: $\text{telic}(L) \ \&_{\Sigma} \ \text{ite}(L) = \emptyset$ for Σ satisfying (17) and (18).

Turning next to the progressive, let us move the spotlight from the last to the initial symbol of a string, defining a function **prog** on languages L by

$$\text{prog}(L) = \square \ \overline{\alpha(L)}^+ \ \overline{\alpha(L)}$$

where

$$\alpha(L) = \{\varphi \in \Phi \mid (\forall s \in L - \{\epsilon\}) \ \varphi \in \text{begin}(s)\}$$

(with $\text{begin}(s)$ equal to the initial symbol of s). Note that

$$L \text{ subsumes } \text{prog}(L) \text{ iff } \alpha(L) \text{ starts } L$$

and that $\alpha(L)$, **prog**(L) and the notion of starting are duals to $\omega(L)$, **telic**(L) and finishing (respectively), obtained by reversing languages. The dual of the function **ite** for iterative languages is the function

$$\text{reten}(L) = \alpha(L) \ \alpha(L)^+ \ \square$$

for languages that are *retentive* insofar as the initial condition $\alpha(L)$ is preserved by the middle. The characterization of Vendler classes by binary features $[\pm\text{for}]$ and $[\pm\text{Prog}]$ in Naumann (2001), reproduced here as Table II, can be formulated in terms of subsumption

$$\begin{aligned} L \text{ is } [+for] & \text{ iff } L \text{ subsumes } \text{iter}(L) \\ L \text{ is } [-for] & \text{ iff } L \text{ subsumes } \text{telic}(L) \\ L \text{ is } [+Prog] & \text{ iff } L \text{ subsumes } \text{prog}(L) \\ L \text{ is } [-Prog] & \text{ iff } L \text{ subsumes } \text{reten}(L) . \end{aligned}$$

Table II. From Naumann (2001)

Vendler class	for	Prog[ress]
state	+	-
activity	+	+
achievement	-	-
accomplishment	-	+

Now, suppose the progressive $\text{PROG}(L)$ of a language L were defined, and that every string in $\text{PROG}(L)$ is a prefix of a non-empty string in L going beyond the first symbol but falling short of the last, as specified in (21).

- (21) For every string $s \in \text{PROG}(L)$, $\text{length}(s) \geq 2$ and $ss' \in L$ for some non-empty string $s' \in \text{Power}(\Phi)^+$.

While (21) provides only a very sketchy picture of the progressive, it is sufficient to yield

$$\text{PROG}(L) \text{ subsumes } \Box^* \omega(L) \quad \text{if } L \text{ is iterative}$$

but *not* so if L is telic. Inasmuch as $\omega(L)$ signals the culmination of L , we have the basis here of an explanation for the entailment in (7b)

$$\text{ing}(\text{Carl-drink-beer}) \models \text{Carl-drink-beer}$$

and the non-entailment in (8b)

$$\text{ing}(\text{Carl-drink-six-pints}) \not\models \text{Carl-drink-six-pints} .$$

But what about (1a)

$$\text{ing}(\text{Harry-hiccup}) \models \text{twice}(\text{Harry-hiccup}) ?$$

The argument for (1a) in Ramsay (1994,1999) is based on the assumption that the progressive $\text{ing}(\text{Harry-hiccup})$ describes a temporal interval that has at least two points, enough to implicate two instantaneous Harry-hiccup events. Now, while (21) says that every string s in $\text{PROG}(L)$ has $\text{length} \geq 2$, it also says that s must be a proper prefix of some string in L . That is, if all strings in a language L for instantaneous events have length 1, then $\text{PROG}(L) = \emptyset$ by (21). But we agreed in §2.1 that an equality $\mathcal{L}(S) = \emptyset$ is a symptom that S is odd; and presumably “Harry was hiccuping” is not. So we had better not apply (21) to a language L for instantaneous events, construed as

a set of strings of length 1. Let us restrict (21) to languages L that subsume $\square\square^+\square$. Fortunately, this includes the Vendler classes.

That still leaves (1a) unresolved. And it also raises the larger question of how to reconcile talk of temporal intervals, points, and instantaneous events with the languages L we have been working with. The next section fashions a partial answer.

3. Grounding the descriptions

Can we assume (following Ramsay 1994,1999) that “Harry was hiccuping” describes a temporal interval with at least two points, and thus conclude that “Harry hiccuped twice”? Line (3) above (“Harry was hiccuping when Sally shot him dead”) invokes the *imperfective paradox* (Dowty, 1979), suggesting that a hiccup might be interrupted. Parsons (1990) side-steps the imperfective paradox by calling on partial events, or rather, in-progress states. But can we (truthfully) say “Harry hiccuped twice” when the second hiccup, in fact, failed to culminate?

It is not altogether clear (to me) that the iteration Ramsay is after depends on the progressive. Consider the progressive-free sentences (22) and (23), essentially from Jackendoff (1991).

(22) Harry hiccupped all night.

(23) The light flashed all night.

If (22) and (23) are true, then we can be sure that there were more than two complete hiccups/flashes that night. A night is simply too long, and hiccups/flashes short. Similarly, the reason, I think, we would normally infer “Harry hiccuped twice” from “Harry was hiccuping” is that the former is *normally used* to describe a sufficiently long time span. Surely, however, we may utter (3) to report an incident where Harry had no time to hiccup twice.

The possibility of counting Harry’s hiccups, any two of which are separated by a temporal interval, points to a certain objection that might be raised against the application of Kleene iteration \cdot^+ to characterize activities and states in §2.3. From Tenny and Pustejovsky (2000), for example, we have

An activity or a state can be considered a homogeneous event because it may be divided into any number of temporal slices, and one will still have an event of the same kind (i.e. if *Boris walked along the road* is true for ten seconds, then a one-second slice of that walking is still an event of walking along the road). There are obvious problems relating to the granularity analysis of homogeneity that we will ignore for discussion’s sake.

Obvious or not, the present section is all about granularity. For granularity is crucial to the basic claim of this paper: that *events (whatever they are) are not so much about temporal intervals but rather about samplings of such intervals — and very selective samplings at that, given by strings of observations made with bounded granularity.*

That said, let us turn to $\mathcal{L}(\text{an hour})$, defined in (13) as

$$\boxed{\text{time}(x)} \square^+ \boxed{\text{time}(y), \text{hour}(x, y)} .$$

The variables x and y above are evidently intended to range over times, stamping positions in strings. But what times? For the sake of concreteness, let us identify temporal instants/points with real numbers, writing \mathfrak{R} for the set of reals. Real numbers embody a certain infinite precision that our observations can only finitely approximate with *non-empty* open intervals

$$(a, b) = \{r \in \mathfrak{R} \mid a < r < b\}$$

for $a, b \in \mathfrak{R}$ with $a < b$. Identifying a level of granularity with a choice of a real number $\delta > 0$, our best δ -approximation of a real number r is the open interval

$$\delta_r = \left(r - \frac{\delta}{2}, r + \frac{\delta}{2}\right)$$

of size δ with center r . (The intuition is that δ is a bound on the precision of observations of that granularity.) Accordingly, we define the set \mathcal{O}_δ of δ -points to be

$$\mathcal{O}_\delta = \{\delta_r \mid r \in \mathfrak{R}\} .$$

Now, while we may take the variables x and y in $\mathcal{L}(\text{an hour})$ above to range over real numbers, we should be careful to fix a level $\delta > 0$ of granularity, treating a real number r as the δ -point δ_r . The slogan is *real observations are δ -points* (for some δ). The remainder of this paper uses sequences of δ -points to ground strings from the languages L considered in section 2.

3.1. δ -SUCCESSORS AND δ -INSTANTIATIONS

Fix a real number $\delta > 0$. Let $<_\delta$ be the binary relation on the set \mathcal{O}_δ of δ -points o, o' given by

$$\begin{aligned} o <_\delta o' & \text{ iff } (\forall r \in o)(\forall r' \in o') r < r' \\ & \text{ [iff the right end-point of } o \text{ is } \leq \text{ the left end-point of } o'] \end{aligned}$$

and let $\text{succ}_\delta \subseteq \mathcal{O}_\delta \times \mathcal{O}_\delta$ be the successor relation induced by $<_\delta$

$$o \text{ succ}_\delta o' \text{ iff } o <_\delta o' \text{ and there is no } o'' \text{ such that } o <_\delta o'' <_\delta o' .$$

Notice that succ'_δ is *not* functional (i.e. δ -successors are not unique), and that while $o \text{ succ}_\delta o'$ precludes a gap between o and o' large enough for another δ -point, we might be able to squeeze a δ' -point in between o and o' , for some positive $\delta' < \delta$.

Returning now to our finite-state descriptions $L \subseteq \Sigma^+$, let us define a δ -instantiation of L to be a function e from some finite subset $\{o_1, o_2, \dots, o_k\}$ of \mathcal{O}_δ to Σ such that

$$e(o_1) e(o_2) \cdots e(o_k) \in L$$

and $o_i \text{ succ}_\delta o_{i+1}$ for $1 \leq i < k$. Given such a δ -instantiation e , let us define $\text{first}(e)$ to be the $<_\delta$ -least element o_1 in $\text{domain}(e)$, and $\text{last}(e)$ to be the $<_\delta$ -greatest element o_k in $\text{domain}(e)$. Now, the variables x and y in $\mathcal{L}(\text{an hour})$ can be assigned δ -points⁹ by a δ -instantiation of $\mathcal{L}(\text{an hour})$. More precisely, a δ -instantiation e of $\mathcal{L}(\text{an hour})$ assigns x the δ -point $\text{first}(e)$, and y the δ -point $\text{last}(e)$.

3.2. MODELS AND INTERVALS

What must we require of a model M in order to interpret a language L relative to M ? For interpretations of the variables x and y above, M must include among its objects some δ -points. And in order to make sense of propositions $\varphi \in \Phi$ that occur within L , we must translate these to formulas φ_z , where z is a (temporal) variable and the truth of φ_z can be tested against M , coupled with an assignment of z to a δ -point in M . For instance, $\text{rain}_z = \text{rain}(z)$ and so the vocabulary/signature of M must include a unary relation (symbol) rain . The unary relation time in $\mathcal{L}(\text{an hour})$ is special in that $\text{time}(u)^z$ will, for our purposes, be set to the equality $u = z$ (to force the assignment to x and y mentioned at the end of §3.1). Also special is the binary relation hour , which is atemporal in the sense that $\text{hour}(x, y)^z$ is just $\text{hour}(x, y)$. (That is, $\text{hour}(x, y)^z$ ignores the variable z .) Let $\mathbf{v}(\Phi)$ be a vocabulary such that for every $\varphi \in \Phi$, the translation φ_z is a $\mathbf{v}(\Phi)$ -formula. To interpret a language $L \subseteq \text{Power}(\Phi)^+$ relative to a $\mathbf{v}(\Phi)$ -model M , it will be useful to fix a 1-1 function from δ -points to variables that do not occur in Φ , writing z^o for the variable to which the δ -point o is mapped. Now, suppose we are given a $\mathbf{v}(\Phi)$ -model M , a variable assignment f into objects in M ,¹⁰ and a function e (such as a δ -instantiation of a

⁹ Or real numbers, via the inverse of the map from $r \in \mathfrak{R}$ to ${}^\delta r \in \mathcal{O}_\delta$.

¹⁰ In addition to the variables x and y in $\mathcal{L}(\text{an hour})$, section 2 uses terms that we may construe as variables, such as a for the six pints in $\mathcal{L}(\text{Carl drink six pints})$ and m for the mile in $\mathcal{L}(\text{Pat swim a mile})$.

language) from a finite subset of \mathcal{O}_δ to $\Sigma \subseteq \text{Power}(\Phi)$. Let us define M, f supports e to mean that for every $o \in \text{domain}(e)$,

$$f(z^o) = o \quad \text{and} \quad (\forall \varphi \in e(o)) \quad M, f \models \varphi_{z^o} .$$

For instance, if \hat{e} has domain $\{o_1, o_2, o_3\}$ and

$$\begin{aligned} \hat{e}(o_1) &= \boxed{\text{time}(x), \text{rain}} \\ \hat{e}(o_2) &= \boxed{\text{rain}} \\ \hat{e}(o_3) &= \boxed{\text{time}(y), \text{rain}, \text{hour}(x, y)} \end{aligned}$$

then M, f supports \hat{e} precisely if

$$\begin{aligned} f(x) &= f(z^{o_1}) \\ M &\models \text{rain}[o_1] \\ M &\models \text{rain}[o_2] \\ f(y) &= f(z^{o_3}) \\ M &\models \text{rain}[o_3] \\ M &\models \text{hour}[o_1, o_3] \end{aligned}$$

and, for good measure, $f(z^{o_1}) = o_1$, $f(z^{o_2}) = o_2$ and $f(z^{o_3}) = o_3$.

Next, passing from functions e to languages L and open intervals (r, r') over \mathfrak{R} , we define

$$M, f \models L @_\delta (r, r') \quad \text{iff} \quad M, f \text{ supports a } \delta\text{-instantiation } e \text{ of } L \text{ such} \\ \text{that } \text{first}(e) = {}^\delta r \text{ and } \text{last}(e) = {}^\delta r' .$$

Note that (r, r') describes, in effect, the open interval $(r - \frac{\delta}{2}, r' + \frac{\delta}{2})$. For every open interval $o = (a, b)$ over \mathfrak{R} , let us write o_δ for $(a - \frac{\delta}{2}, b + \frac{\delta}{2})$. We can then interpret a form of iteration $\text{ITER}(L)$ of L against an open set o by checking if there are “enough” δ -points $o' \subseteq o_\delta$ such that $L @_\delta o'$. That is,

$$M, f \models \text{ITER}(L) @_\delta o \quad \text{iff} \quad \{o' \subseteq o_\delta \mid M, f \models L @_\delta o'\} \in Q_{\delta, o}$$

for some family $Q_{\delta, o}$ of sets of δ -points contained in o_δ . The family $Q_{\delta, o}$ may vary not only with the choice of the model M (just as the interpretation of rain does in first-order logic) but also with the particular language L . Hanging the superscript L on $Q_{\delta, o}$, it is natural to ask: how does $Q_{\delta, o}^L$ change with L ? It may well turn out that this quantificational interpretation has very little inferential bite, spurring us on to re-analyze $\text{ITER}(L)$ in finite-state terms along the lines of §2.3. Much remains to be said.

References

- N. Chang, D. Gildea, and S. Narayanan. A dynamic model of aspectual composition. In *Proc. CogSci 98*, 1998.
- Donald Davidson. The logical form of action sentences. In N. Rescher, ed., *The Logic of Decision and Action*, pages 81–95. University of Pittsburgh Press, 1967.
- David R. Dowty. *Word Meaning and Montague Grammar*. Reidel, 1979.
- Tim Fernando. A finite-state approach to event semantics. In *Proc. of the 9th International Symposium on Temporal Representation and Reasoning (TIME-2002)*, pages 124–131. IEEE CS Press, 2002. Section 2 of this paper is elaborated in <http://www.cs.tcd.ie/Tim.Fernando/jlc.pdf>.
- Ray Jackendoff. Parts and boundaries. In B. Levin and S. Pinker, eds., *Lexical and Conceptual Semantics*, pages 9–45. Blackwell, 1991.
- Manfred Krifka. Nominal reference, temporal constitution and quantification in event semantics. In R. Bartsch, J. van Benthem, and P. van Emde Boas, eds., *Semantics and Contextual Expressions*, pages 75–115. Foris, 1989.
- M. Moens and M. Steedman. Temporal ontology and temporal reference. *Computational Linguistics*, 14(2):15–28, 1988.
- Reinhard Muskens. Talking about trees and truth-conditions. *Journal of Logic, Language and Information*, 10(4):417–455, 2001.
- Ralf Naumann. Aspects of changes: a dynamic event semantics. *J. Semantics*, 18:27–81, 2001.
- Terence Parsons. *Events in the Semantics of English: A Study in Subatomic Semantics*. MIT Press, 1990.
- Stephen Pinker. *Learnability and Cognition: The Acquisition of Argument Structure*. MIT Press, 1989.
- James Pustejovsky. The generative lexicon. *Computational Linguistics*, 17(4): 409–441, 1991.
- Allan Ramsay. The co-operative lexicon. In H. Bunt, R. Muskens, and G. Rentier, eds., *Proc. International Workshop on Computational Semantics*, pages 171–180. ITK, Tilburg, 1994.
- Allan Ramsay. Dynamic and underspecified interpretation without dynamic or underspecified logic. In H. Bunt and R. Muskens, eds., *Computing Meaning*, pages 57–72. Kluwer, 1999.
- Mark Steedman. *The Productions of Time*. Draft, <ftp://ftp.cogsci.ed.ac.uk/pub/steedman/temporality/temporality.ps.gz>, July 2000.
- C. Tenny and J. Pustejovsky. A history of events in linguistic theory. In C. Tenny and J. Pustejovsky, eds., *Events as Grammatical Objects*, pages 3–37. CSLI, Stanford, 2000.
- Alice G.B. ter Meulen. English aspectual verbs as generalized quantifiers. In J. Carter et al., eds., *Proc. NELS 20*. GLSA, Department of Linguistics, University of Massachusetts, 378–390.
- Satoshi Tojo. Event, state and process in arrow logic. *Minds and Machines*, 9:81–103, 1999.
- Zeno Vendler. *Linguistics in Philosophy*. Cornell University Press, 1967.
- Henk Verkuyl. Events as individuals. In J. Higginbotham, F. Pianesi, and A.C. Varzi, eds., *Speaking of Events*, pages 169–205. Oxford University Press, 2000.
- Dag Westerståhl. Quantifiers in formal and natural languages. In D. Gabbay and F. Guenther, eds., *Handbook of Philosophical Logic*, volume IV, pages 173–209. Reidel, 1989.