Situations from events to proofs

Tim Fernando

Abstract. String representations of events are applied to Robin Cooper's proposal that propositions in natural language semantics are types of situations. Links with the higher types of proof-theoretic semantics are forged, deepening type-theoretic interpretations of Discourse Representation Structures to encompass event structures.

1 Introduction

Barwise and Perry's *situation semantics* sprang in no small measure from unhappiness with the possible worlds analysis (PW) of a proposition.

(PW) A proposition is the set of possible worlds in which it is true.

Proof-theoretic semantics (e.g. *intuitionistic type theory*, [ML84]) offers an alternative (PT) to (PW), replacing truth in a possible world by proof.

(PT) A proposition is the type of objects that prove it.[1]

Situation semantics is linked to intuitionistic type theory in [Coo05] on the basis of the following reading of [Aus61] in [BP83]:

> a statement is true when the actual situation to which it refers is of the type described by the statement. (p. 160)

With this in mind, Cooper points to [Ran94] for applications of (PT) to linguistic semantics, including events in [Dav80].

> For non-mathematical propositions Ranta draws a parallel with Davidson's (1980) event-based approach. In type theory a proof of the proposition that *Amundsen flew over the North Pole* would be a flight by Amundsen over the North Pole. Mixing type theoretical and situation theoretical terminology we might say that a proof of this proposition is a situation in which Amundsen flew over the North Pole and that type theoretical propositions are types of situations. ([Coo05], page 335)

Cooper goes on to encode situations as type-theoretic records, noting along the way similarities with *Discourse Representation Theory* (DRT, [KR93]). The present paper aims to deepen these connections, focusing on temporal matters that are analyzed in DRT using *event structures*. It strays from intuitionistic type theory in adopting a computational model for events altogether different from the typed functional programming assumed in proof-theoretic semantics.

[1] For present purposes, let us put aside any difference in meaning between "set" and "type" (except that the former is used chiefly in connection with truth conditions and the latter with proof conditions).

Nonetheless, the resulting account is, I will argue, compatible with the spirit, if not with the letter, of [Ran94, Coo05] and [KR93].

Compatibility with possible worlds semantics is a more tortuous affair. It will prove instructive to highlight two familiar principles associated with the view (PW) that a proposition is a set of possible worlds. The first, (CM), concerns the use Carnap and Montague make of possible worlds to bridge notions of intension (or sense) with extension (or reference).

(CM) Intension is a function from possible worlds to extensions.

The second, (TV), specifies the possible extensions of a sentence.

(TV) Sentences refer to one of two truth values.

If propositions are intensions of sentences, as we henceforth assume, then (PW) becomes an immediate consequence of (CM), (TV) and the equivalence between the space 2^W of functions from a set W to the two-element set $2 = \{0, 1\}$ and the family $\mathsf{Pow}(W)$ of subsets of W. Barwise and Perry reject (TV) and an argument for (TV) they attribute to Davidson (among others) which they call "the slingshot, for reasons buried in the early history of situation semantics" ([BP83], page 24)] Instead, in situation semantics, a statement refers to a situation (as Cooper reminds us above). The switch here from sentences to statements hints at the context dependence of reference famously captured by Kaplan's notion of *character* ([Kap79]) as a function from contexts to *contents*, the latter being the intensions characterized in (CM). In place of (CM), Barwise and Perry propose a "relational theory of meaning," an early form of which is put as follows.

> The leading idea of situation semantics is that the meaning of a simple declarative sentence is a relation between utterances and described situations. The interpretation of a statement made with such a sentence on a specific occasion is the described situation. ([BP83], page 19)

As is clear from [Per97], this passage leaves out later developments in situation semantics, including elaborations of utterances into pairs of situations (called discourse situations and connective situations). Compared to possible worlds, situations are much smaller; situations are partial in that, unlike possible worlds, they may fail to make an arbitrary sentence true or false.

Without aspiring to be fully faithful to any particular version of situation semantics, the present paper follows situation semantics in employing situations for two purposes:

(i) as alternatives to truth values in (TV), and

(ii) as alternatives to possible worlds in (CM).

This is not to say that we need only replace possible worlds and extensions in (CM) by situations for an account of what the intension of a sentence is. That account would be in direct conflict with propositions-as-types (PT), if intensions of sentences are, as we have agreed, propositions. To clarify matters, notice that (CM) implies (UR), assuming possible worlds in (CM) are relabeled "circumstances of evaluation."

(UR) The reference of a sentence is uniquely determined by the proposition the sentence expresses and the circumstances of evaluation.

Reducing circumstances of evaluation from possible worlds (in (CM)) to partial situations raises the question: could there not be propositions and circumstances of evaluation that together underdetermine reference, contradicting (UR)? We may, of course, hardwire existence and uniqueness of reference by throwing out circumstances of evaluation that lead to non-deterministic reference. But might such circumstances not be interesting in their own right? A plausible approach to ambiguity and presupposition failure is to explore the non-determinism possible in relations, as opposed to functions (from say, (CM) or, for that matter, Kaplan). That said, if we put ambiguity and presupposition failure aside, can we speak of "*the* described situation"?

No, or so Davidson argues in page 91 of [Dav67], where he considers sentence (S1).

(S1) Amundsen flew to the North Pole in May 1926.

Davidson cautions against thinking (S1) describes an event[2] because of potential complications with uniqueness; the sentence may describe several flights or "a kind of event." In any case, the crucial point for Davidson is that the sentence is true iff "there is an event that makes it true" ([Dav67], page 91). Here Davidson calls events proofs (inasmuch as what makes a sentence true is a proof). More generally, he associates certain action sentences S with sets $E(S)$ of events such that

$$S \text{ is true} \quad \text{iff} \quad \text{there is an event } e \text{ such that } e \in E(S)\ . \tag{1}$$

Possible worlds do *not* appear in (1), the idea being that (1) is purely extensional, with events as particulars. Observe, however, that (1) may hold for quite unrelated pairs S and $E(S)$. For instance, given any event $\hat{e}$, we can satisfy (1) by setting

$$E(S) \overset{\text{def}}{=} \begin{cases} \{\hat{e}\} & \text{if } S \text{ is true} \\ \emptyset & \text{otherwise} \end{cases}$$

independently of whether or not S is remotely about $\hat{e}$. If the "actual" world were an event, we could, of course, choose that for $\hat{e}$, and claim that S is in some sense about $\hat{e}$. Even then, however, the question remains: why speak of events if one, $\hat{e}$, will do?

In fact, we need more than one. An event may appear in many expressions as arguments to various predicates from which $E(S)$ is, in practice, formed. *Logical form* is central to [Dav67]; it is unfortunate that (1) hides the logical form behind $E(S)$ and the possibility that e occurs many places within that

[2]Throughout this paper, we follow Cooper in understanding events as situations.

form. But what if that logical form has no structure, or rather, its structure is "atomic"? Proof-theoretic semantics provides interpretations of logical connectives but otherwise says nothing about the interpretations of atomic expressions (which are free of connectives). Whatever one may think of logical atomism, there is considerable linguistic interest in developing what Parsons calls *subatomic semantics*, by which he means

> the study of those "formulas of English" that are treated as atomic formulas in most investigations of English. The main hypothesis to be investigated is that simple sentences of English contain subatomic quantification over events. ([Par90], page ix)

Significant aspects of subatomic semantics concern temporality. Indeed, it is difficult to see anything beyond the temporal traces of events in the event structures of [KR93].

A bit of temporal logic (aka Priorean tense logic) provides a glimpse of what the notions of intension, extension and circumstances of evaluation become below. Let $\hat{S}$ be the formula

$$\mathsf{previous}(\varphi) \quad \wedge \quad (\psi \ \mathsf{until} \ \chi)$$

over the atomic propositions φ, ψ and χ. Let us interpret $\hat{S}$ relative to a discrete timeline such as the integers (with well-defined previous and next moments). Under the standard Kripke semantics for temporal logic (e.g. [Eme92]), $\hat{S}$ is true at a moment now iff

previous to now, φ is true and ψ is true from now until χ is true.

Accordingly, an event proving $\hat{S}$ is a string of the form

$$\boxed{\varphi}\boxed{\mathsf{now},\ \psi}\boxed{\psi}^{n}\boxed{\chi} \qquad \text{(for some } n \geq 0\text{)}$$

with length $(n+3)$, specifying φ one moment before now and ψ from now until χ. We enclose snapshots in boxes and string these together into film strips. Using the notation in (1), we associate with $\hat{S}$ the event-type

$$\begin{aligned} E(\hat{S}) &= \boxed{\varphi}\boxed{\mathsf{now},\ \psi}\boxed{\psi}^{*}\boxed{\chi} \\ &= \{\boxed{\varphi}\boxed{\mathsf{now},\ \psi}\boxed{\chi},\ \boxed{\varphi}\boxed{\mathsf{now},\ \psi}\boxed{\psi}\boxed{\chi},\ \boxed{\varphi}\boxed{\mathsf{now},\ \psi}\boxed{\psi}\boxed{\psi}\boxed{\chi},\ \ldots\}\ . \end{aligned}$$

Next, we say a string s *occurs in* a string c if we can form s from c by selectively deleting boxes at the beginning and end of c, as well as various occurrences of formulas in c. For example,

$$\boxed{\varphi}\boxed{\mathsf{now},\ \psi}\boxed{\psi}\boxed{\chi} \quad \text{occurs in} \quad \boxed{\varphi, \psi}\boxed{\varphi, \chi}\boxed{\mathsf{now},\ \varphi, \psi}\boxed{\varphi, \psi}\boxed{\varphi, \chi} \qquad (2)$$

as can be seen by deleting everything underlined in

$$\underline{\underline{\boxed{\varphi, \psi}}}\boxed{\varphi, \underline{\chi}}\boxed{\mathsf{now},\ \underline{\varphi}, \psi}\boxed{\underline{\varphi}, \psi}\boxed{\underline{\varphi}, \chi}\ .$$

We can then relativize (1) to a circumstance of evaluation c, formulated as a string, by agreeing that

$$S \text{ occurs in } c \quad \stackrel{\text{def}}{\Longleftrightarrow} \quad (\exists e \in E(S))\ e \text{ occurs in } c\ . \tag{3}$$

It follows from (2) and (3) that

$$\hat{S} \quad \text{occurs in} \quad \boxed{\varphi,\psi}\boxed{\varphi,\chi}\boxed{\mathsf{now},\ \varphi,\psi}\boxed{\varphi,\psi}\boxed{\varphi,\chi}\ .$$

$\hat{S}$ also occurs in

$$c_1 \quad = \quad \boxed{\varphi,\psi}\boxed{\mathsf{now},\ \varphi,\psi}\boxed{\psi,\chi}\boxed{\varphi,\chi}\boxed{\varphi,\psi}\boxed{\varphi,\psi}$$

this time because

$$\boxed{\varphi}\boxed{\mathsf{now},\ \psi}\boxed{\chi} \text{ and } \boxed{\varphi}\boxed{\mathsf{now},\ \psi}\boxed{\psi}\boxed{\chi} \quad \text{both occur in} \quad c_1\ .$$

We define the *c-extension of* S, $\text{ext}(S, c)$, to be the set of strings in $E(S)$ that occur in c

$$\text{ext}(S, c) \quad \stackrel{\text{def}}{=} \quad \{s \in E(S) \mid s \text{ occurs in } c\}$$

so that

$$S \text{ occurs in } c \quad \text{iff} \quad \text{ext}(S, c) \neq \emptyset\ .$$

As $\text{ext}(S, c)$ need not be a singleton $\{s\}$, we ought perhaps not speak of reference, leaving that to what Austin calls *demonstrative conventions*. If reference is not fixed in (3), then (3) must, of course, fall short of Barwise and Perry's reading of Austinian truth (quoted by Cooper). On the other hand, the intension/type $E(\hat{S})$ described by $\hat{S}$ above is in harmony with Kripke semantics for $\hat{S}$, formulated in terms of truth at worlds, insofar as every world w is fully determined by a set E_w of events, and

$$\hat{S} \text{ is true at } w \quad \text{iff} \quad E(\hat{S}) \cap E_w \neq \emptyset\ .$$

In place of the Carnap-Montague picture (CM) of intension as a function from possible worlds to extensions, the intension of S is identified with the set $E(S)$ of possible values (extensions) of S, some subset of which get realized in a circumstance of evaluation.

The strings we build below are, in general, designed to ground the higher types of proof-theoretic semantics in finite observations over time ([Fer04]). But are we not, by working with strings, smuggling an assumption of discrete time that for all we know, is unfounded? In section 2, we link discreteness to the assumption that our observations are finite, and construct arbitrary event structures such as the real line from inverse limits of strings. Whereas events for Davidson are particulars, our strings are representations that can be embedded in incompatible circumstances of evaluation. In section 3, we study explicit and contextual entailments, treating strings as both extensions and circumstances of evaluation. These two roles of strings are related in section 4 to proof-theoretic semantics.

2 Event tokens and time in languages

Let us henceforth call a set of strings a *language* (following the custom in formal language theory) and temporal formulas *fluents* (following the custom in AI since [MH69]). The strings we are interested in are sequences of sets of fluents, enclosed in boxes to reinforce the intuition of snapshots arranged into a movie. For example, we can portray the phrase *rain from dawn to dusk* by the set

$$\begin{aligned} & \fbox{$rain,\ dawn$}\fbox{$rain$}^{*}\fbox{$rain,\ dusk$} \\ = \ & \{\fbox{$rain,\ dawn$}\fbox{$rain$}^{n}\fbox{$rain,\ dusk$} \mid n \geq 0\} \end{aligned}$$

of strings $\fbox{$rain,\ dawn$}\fbox{$rain$}^{n}\fbox{$rain,\ dusk$}$ of length $n+2$ that describe $n+2$ successive moments of time (for $n \geq 0$), each with rain, the first at dawn and the last at dusk. The different values of n support different levels of temporal granularity: the larger the n, the finer the grain. It is not obvious, however, that we should think of each of these infinitely many strings as distinct events. Surely we can reduce the infinite language to some finite core that captures its essence: three snapshots, $\fbox{$rain,\ dawn$}$, $\fbox{$rain$}$ and $\fbox{$rain,\ dusk$}$, arranged in a particular order. Indeed, it is tempting to reduce every string $\fbox{$rain,\ dawn$}\fbox{$rain$}^{n}\fbox{$rain,\ dusk$}$ with $n \geq 1$ to the string $\fbox{$rain,\ dawn$}\fbox{$rain$}\fbox{$rain,\ dusk$}$ of length 3. Let us define, in general, the *interval reduction* $\mathsf{ir}(s)$ of a string s inductively

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

(for all sets a, a' of fluents), reducing a block aa of two a's to one, in line with the dictum "no time without change" ([KR93], page 674).

We can represent intervals by fluents and the thirteen basic interval relations of [All83], listed in Table 1, by strings derived from ir and a natural form of conjunction between languages L and L'. The *superposition $L\&L'$ of L and L'* consists of the componentwise union of strings in L and L' of the same length

$$L\&L' \stackrel{\text{def}}{=} \bigcup_{n\geq 0}\{(a_1 \cup b_1)\cdots(a_n \cup b_n) \mid a_1\cdots a_n \in L \text{ and } b_1\cdots b_n \in L'\}$$

([Fer04]). For example,

$$\begin{aligned} \fbox{$rain,\ dawn$}\fbox{$rain$}^{*}\fbox{$rain,\ dusk$} &= \fbox{rain}^{+} \ \& \ (\fbox{$dawn$}\Box^{*}\fbox{$dusk$}) \\ &= \fbox{rain}^{+} \ \& \ \fbox{$dawn$}\Box^{+} \ \& \ \Box^{+}\fbox{$dusk$} \end{aligned}$$

where $L^{+} \stackrel{\text{def}}{=} LL^{*}$ (and parentheses are suppressed, as & is associative). To loosen the tight temporal overlap between L and L' in $L\&L'$, we pad L and L' to the left and right by empty boxes $\Box$ (giving $\Box^{*}L\Box^{*}$ and $\Box^{*}L'\Box^{*}$) and then

φ before ψ	$\boxed{\varphi}\Box\boxed{\psi}$	φ after ψ	$\boxed{\psi}\Box\boxed{\varphi}$
φ meets ψ	$\boxed{\varphi}\boxed{\psi}$	φ met-by ψ	$\boxed{\psi}\boxed{\varphi}$
φ contains ψ	$\boxed{\varphi}\boxed{\varphi,\psi}\boxed{\varphi}$	φ finished-by ψ	$\boxed{\varphi}\boxed{\varphi,\psi}$
φ started-by ψ	$\boxed{\varphi,\psi}\boxed{\varphi}$	φ equals ψ	$\boxed{\varphi,\psi}$
φ overlapped-by ψ	$\boxed{\psi}\boxed{\varphi,\psi}\boxed{\varphi}$	φ finishes ψ	$\boxed{\psi}\boxed{\varphi,\psi}$
φ overlaps ψ	$\boxed{\varphi}\boxed{\varphi,\psi}\boxed{\psi}$	φ during ψ	$\boxed{\psi}\boxed{\varphi,\psi}\boxed{\psi}$
φ starts ψ	$\boxed{\varphi,\psi}\boxed{\psi}$		

Table 1: Allen relations ([All83]) as strings

unpad their superposition. That is, we define the *loose superposition* $L\&^{\Box}L'$ *of* L *and* L' by

$$L\&^{\Box}L' \overset{\text{def}}{=} \mathsf{unpad}(\Box^* L \Box^* \ \& \ \Box^* L' \Box^*)$$

where $\mathsf{unpad}(s)$ is s with all initial and final $\Box$'s stripped off unless $s = \Box$

$$\mathsf{unpad}(s) \overset{\text{def}}{=} \begin{cases} s & \text{if } s = \Box \text{ or} \\ & \quad s \text{ neither begins nor ends with } \Box \\ \mathsf{unpad}(s') & \text{if } s = \Box s' \text{ or else if } s = s'\Box\ . \end{cases}$$

and $\mathsf{unpad}(L) \overset{\text{def}}{=} \{\mathsf{unpad}(s) \mid s \in L\}$. Now, we can generate the thirteen strings representing the Allen relations in Table 1 by

$$\begin{aligned} \mathsf{ir}(\mathsf{ir}^{-1}(\boxed{\varphi}) \ \&^{\Box} \ \mathsf{ir}^{-1}(\boxed{\psi})) &= \mathsf{ir}(\boxed{\varphi}^+ \ \&^{\Box} \ \boxed{\psi}^+) \\ &= \mathsf{ir}(\mathsf{unpad}(\Box^*\boxed{\varphi}^+\Box^* \ \& \ \Box^*\boxed{\psi}^+\Box^*)) \\ &= \boxed{\varphi}(\Box+\epsilon)\boxed{\psi} \ + \ \boxed{\psi}(\Box+\epsilon)\boxed{\varphi} \ + \\ &\quad (\boxed{\varphi}+\epsilon+\boxed{\psi})\boxed{\varphi,\psi}(\boxed{\varphi}+\epsilon+\boxed{\psi}) \end{aligned}$$

(where $+$ is choice/union and ϵ is the null string of length 0).

What about relations between more than two intervals? Take the real line $(\Re, <)$, used in [LH05] with the admission that

> One cannot simply assume that we have a dense set of events in memory to derive from this that cognitive (and not just physical) time may be assumed to be continuous. It is much more reasonable to assume that density arises in the limit of adding more and more events, and that, at each stage, memory contains only finitely many events. (p. 12, extended version of Chapter 2 from `http://staff.science.uva.nl/ ~michiell/docs/ProofsTime.pdf`)

Let us suppose each real number is a fluent so that we can turn any finite sequence of real numbers

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

into the string

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

with empty boxes as place-holders for the uncountably many points sandwiched between r_i and r_{i+1}. We can construct $(\Re, <)$ from such strings through inverse limits that we can define for any set Φ of fluents ($\Re$ or not). Given a finite subset X of Φ, let $\rho_X : \mathsf{Pow}(\Phi)^* \to \mathsf{Pow}(X)^*$ be the function that maps a string $a_1 \cdots a_n$ of sets a_i of fluents to its X*-restriction* $\rho_X(a_1 \cdots a_n)$ obtained by intersecting each a_i with X

$$\rho_X(a_1 \cdots a_n) \quad \stackrel{\text{def}}{=} \quad (a_1 \cap X) \cdots (a_n \cap X) \ .$$

For example,

$$\rho_{\{r_2,r_4\}}(\boxed{r_1}\Box\boxed{r_2}\Box\boxed{r_3}\Box\boxed{r_4}) \quad = \quad \Box\Box\boxed{r_2}\Box\Box\Box\boxed{r_4} \ .$$

Next, we compose ir and unpad with ρ_X to define

$$\pi_X(s) \quad \stackrel{\text{def}}{=} \quad \mathsf{ir}(\mathsf{unpad}(\rho_X(s))) \quad = \quad \mathsf{unpad}(\mathsf{ir}(\rho_X(s)))$$

so that, for instance,

$$\pi_{\{r_2,r_4\}}(\boxed{r_1}\Box\boxed{r_2}\Box\boxed{r_3}\Box\boxed{r_4}) \quad = \quad \boxed{r_2}\Box\boxed{r_4} \ .$$

A Φ*-point* is a function p from the set $Fin(\Phi)$ of finite subsets of Φ to

$$\bigcup_{X \in Fin(\Phi)} \mathsf{Pow}(X)^*$$

such that for every $X \in Fin(\Phi)$ and every subset Y of X,

$$p(Y) \quad = \quad \pi_Y(p(X)) \in \mathsf{Pow}(Y)^* \ .$$

For example, each string $s \in \mathsf{Pow}(\Phi)^*$ is represented by the Φ-point p_s mapping X to $\pi_X(s)$. More interestingly, we can represent the real line $(\Re, <)$ by the $\Re$-point $p^{\Re}$ where

$$p^{\Re}(\{r_1, r_2, \ldots, r_n\}) \quad \stackrel{\text{def}}{=} \quad \boxed{r_1}\Box\boxed{r_2}\Box\cdots\Box\boxed{r_n}$$

for any finite sequence $r_1 < r_2 < \cdots < r_n$ of real numbers. The moral here is that whether or not time is dense (like the reals or rationals), finite observations lead to discrete *observation* times. We can refine discrete orders arbitrarily, by working with languages and not simply strings in isolation. But the refinements will each be discrete, mirroring the step-by-step operation of computer programs (and instructions, in general), each step of which is decomposable into finer steps.

The preceding discussion assumes temporal moments/instants are given. An alternative approach described in [KR93] is to start with an *event structure* $\langle \mathsf{E}, <, \bigcirc \rangle$ consisting of a set E of events, and binary relations of temporal precedence $<$ and temporal overlap $\bigcirc$ satisfying postulates (P_1) – (P_7), listed

(P_1)	$e < e'$ implies not $e' < e$
(P_2)	$e < e' < e''$ implies $e < e''$
(P_3)	$e \bigcirc e$
(P_4)	$e \bigcirc e'$ implies $e' \bigcirc e$
(P_5)	$e < e'$ implies not $e \bigcirc e'$
(P_6)	$e_1 < e_2 \bigcirc e_3 < e_4$ implies $e_1 < e_4$
(P_7)	$e < e'$ or $e \bigcirc e'$ or $e' < e$

Table 2: Postulates for event structures ([KR93], page 667)

in Table 2.[3] By (P_7), (P_5) and (P_4), we can define $\bigcirc$ from $<$ through the equivalence

$$e \bigcirc e' \quad \text{iff} \quad \text{neither } e < e' \text{ nor } e' < e \ .$$

In cases where $\bigcirc$ is $=$ on E, $<$ is a linear order. But in general, we

(i) form sets of pairwise overlapping events, collecting them in

$$O(\bigcirc) \stackrel{\text{def}}{=} \{i \subseteq \mathsf{E} \mid (\forall e, e' \in i)\ e \bigcirc e'\}$$

(ii) lift $<$ existentially to $O(\bigcirc)$, defining for $i, i' \in O(\bigcirc)$,

$$i <^O i' \stackrel{\text{def}}{\Leftrightarrow} (\exists e \in i)(\exists e' \in i')\ e < e'$$

(iii) equate temporal instants with $\subseteq$-maximal elements of $O(\bigcirc)$

$$I(\bigcirc) \stackrel{\text{def}}{=} \{i \in O(\bigcirc) \mid (\forall i' \in O(\bigcirc))\ i \subseteq i' \text{ implies } i = i'\}$$

and say e *occurs at* i if $e \in i$.

Theorem (Kamp). *$<^O$ linearly orders $I(\bigcirc)$ and for each $e \in \mathsf{E}$,*

$$\{i \in I(\bigcirc) \mid e \in i\}$$

is a $<^O$-interval (i.e. for all $i, j, k \in I(\bigcirc)$, if $e \in j$, $e \in k$ and $j <^O i <^O k$ then $e \in i$).

How do event structures and Kamp's theorem above relate to Φ-points? Given a Φ-point p, let us form the triple $\langle \mathsf{E}^p, <^p, \bigcirc^p \rangle$ by collecting fluents with interval temporal profiles in

$$\mathsf{E}^p \stackrel{\text{def}}{=} \{\varphi \in \Phi \mid p(\{\varphi\}) = \boxed{\varphi}\}$$

[3] The first two postulates are superfluous. (P_1) is derivable from (P_2), (P_3) and (P_5); and (P_2) from (P_3) and (P_6).

and (with Table 1 in mind) setting

$$\varphi <^p \psi \quad \overset{\text{def}}{\Leftrightarrow} \quad p(\{\varphi,\psi\}) \in \fbox{φ}(\Box + \epsilon)\fbox{ψ} \tag{4}$$

$$\varphi \bigcirc^p \psi \quad \overset{\text{def}}{\Leftrightarrow} \quad p(\{\varphi,\psi\}) \in (\fbox{φ} + \epsilon + \fbox{ψ})\fbox{φ,ψ}(\fbox{φ} + \epsilon + \fbox{ψ}). \tag{5}$$

for all $\varphi, \psi \in \Phi$. For instance, if p is the point $p^{\Re}$ for the real line, then E^p is $\Re$, $<^p$ is $<$ and $\bigcirc^p$ is $=$ on $\Re$. It is easy to verify that for every Φ-point p, $\langle \mathsf{E}^p, <^p, \bigcirc^p \rangle$ is an event structure.

How do we resolve the choices in (4) and (5)? Given an event structure $\langle \mathsf{E}, <, \bigcirc \rangle$, let us define a *compaction of* $\langle \mathsf{E}, <, \bigcirc \rangle$ to be an E-point p such that E is E^p, $<$ is $<^p$, $\bigcirc$ is $\bigcirc^p$ and moreover,

$$p(\{e,e'\}) = \fbox{e}\Box\fbox{e'} \quad \text{iff} \quad (\exists e'' < e')(\exists e''' \bigcirc e'')\ e < e''' \tag{6}$$

$$p(\{e,e'\}) \in \fbox{e}\fbox{e,e'}(\fbox{e} + \epsilon + \fbox{e'}) \quad \text{iff} \quad e \bigcirc e' \text{ and } (\exists e'' < e')\ e \bigcirc e'' \tag{7}$$

$$p(\{e,e'\}) \in (\fbox{e} + \epsilon + \fbox{e'})\fbox{e,e'}\fbox{e} \quad \text{iff} \quad e \bigcirc e' \text{ and } (\exists e'' \bigcirc e)\ e' < e'' \tag{8}$$

for all $e, e' \in \mathsf{E}$. The strings $\fbox{$e$}\Box\fbox{$e'$}$ and $\fbox{$e$}\fbox{$e,e'$}$ pose a problem for the notion of temporal instants as maximal sets of pairwise overlapping events, as neither $\Box$ nor $\fbox{$e$}$ is maximal

$$\Box \subset \fbox{e} \subset \fbox{e,e'}.$$

To overcome this problem, let us introduce whenever $e < e'$,

(i) a fluent $\mathsf{past}(e')$ that stretches indefinitely back over the past of e', and

(ii) a fluent $\mathsf{future}(e)$ that stretches indefinitely forward over the future of e.[4]

We can then flesh out $\fbox{$e$}\Box\fbox{$e'$}$ to

$$\fbox{$e, \mathsf{past}(e')$}\fbox{$\mathsf{future}(e), \mathsf{past}(e')$}\fbox{$\mathsf{future}(e), e'$}$$

and $\fbox{$e$}\fbox{$e,e'$}$ to

$$\fbox{$e, \mathsf{past}(e')$}\fbox{e, e'}$$

satisfying (6) and (7) with $e'' = \mathsf{past}(e')$ and $e''' = \mathsf{future}(e)$. (For (8), $e'' = \mathsf{future}(e')$ will do.) To build past and future into an event structure $\langle \mathsf{E}, <, \bigcirc \rangle$, we extend

[4] This is similar to the Walker instants described in chapter 2 of [LH05] but without the assumption that the past and future sets are non-empty.

(i) every (finite or infinite) subset X of E to

$$X_+ \stackrel{\text{def}}{=} X \cup \{\mathsf{past}(e) \mid e \in X \text{ and } (\exists e')\ e' < e\} \cup \{\mathsf{future}(e) \mid e \in X \text{ and } (\exists e')\ e < e'\}$$

(ii) $\bigcirc$ to overlap $\bigcirc_+$ on E_+, setting for all $e, e' \in \mathsf{E}$,

$$e \bigcirc_+ e' \stackrel{\text{def}}{\Leftrightarrow} e \bigcirc e'$$

with $\mathsf{past}(e)$ and $\mathsf{future}(e)$ given by existential quantification

$$\begin{aligned} \mathsf{past}(e) \bigcirc_+ e' &\stackrel{\text{def}}{\Leftrightarrow} (\exists e'' < e)\ e'' \bigcirc e' \\ \mathsf{future}(e) \bigcirc_+ e' &\stackrel{\text{def}}{\Leftrightarrow} (\exists e'' > e)\ e'' \bigcirc e' \\ \mathsf{past}(e) \bigcirc_+ \mathsf{future}(e') &\stackrel{\text{def}}{\Leftrightarrow} (\exists e'' < e)(\exists e''' > e')\ e'' \bigcirc e''' \end{aligned}$$

and so on, and

(iii) $<$ to precedence $<_+$ on E_+, defining $<_+$ to be the union

$$\begin{aligned} < \ &\cup\ \{(\mathsf{past}(e), e') \in \mathsf{E}_+ \times \mathsf{E}_+ \mid \text{not } \mathsf{past}(e) \bigcirc_+ e'\} \\ &\cup\ \{(e, \mathsf{future}(e')) \in \mathsf{E}_+ \times \mathsf{E}_+ \mid \text{not } e \bigcirc_+ \mathsf{future}(e')\} \\ &\cup\ \{(\mathsf{past}(e), \mathsf{future}(e')) \in \mathsf{E}_+ \times \mathsf{E}_+ \mid \text{not } \mathsf{past}(e) \bigcirc_+ \mathsf{future}(e')\} \end{aligned}$$

so that for all $x, y \in \mathsf{E}_+$,

$$x <_+ y \quad \text{iff} \quad \text{neither } x \bigcirc_+ y \text{ nor } y <_+ x \ .$$

One can show that for every subset X of E, $\langle X_+, <_+^X, \bigcirc_+^X \rangle$ is an event structure, where $<_+^X$ and $\bigcirc_+^X$ are the restrictions of $<_+$ and $\bigcirc_+$ to X_+, respectively

$$<_+^X \stackrel{\text{def}}{=} <_+ \cap\ (X_+ \times X_+) \quad \text{and} \quad \bigcirc_+^X \stackrel{\text{def}}{=} \bigcirc_+ \cap\ (X_+ \times X_+) \ .$$

We can then form a compaction p of $\langle \mathsf{E}, <, \bigcirc \rangle$ as follows. Given a finite subset X of E, we linearly order the temporal instants in $\langle X_+, <_+^X, \bigcirc_+^X \rangle$ via Kamp's theorem, arranging the instants in increasing order $a_1 \cdots a_n$, and setting

$$p(X) \stackrel{\text{def}}{=} \pi_X(a_1 \cdots a_n) \ .$$

For the record,

Theorem. *Every event structure has a compaction.*

A projection π_X filters out information beyond the bounded set X of observations. The asymmetry (P_1) of temporal precedence rules out event recurrence, in line with Davdison's conception of events as particulars. The event structure constructed from a Φ-point p leaves out fluents φ such that $p(\{\varphi\}) \neq \boxed{\varphi}$. Take, for example, φ to be *rain*. Or more abstractly, suppose

$$p(\{\varphi\}_+) = \boxed{\mathsf{past}(\varphi)}\boxed{\varphi}\boxed{\mathsf{future}(\varphi)} \ .$$

Then we would expect that for the negation $\overline{\varphi}$ of φ,

$$p(\{\overline{\varphi}\}) \quad = \quad \fbox{$\overline{\varphi}$}\Box\fbox{$\overline{\varphi}$}$$

which implies $\overline{\varphi} \notin \mathsf{E}^p$. For partial snapshots, it is convenient to explicitly represent negations, stepping beyond event-occurrences within an event structure to event-types as languages.

3 Situation-types and constraints as languages

If we &-superpose $\fbox{$rain,\ dawn$}\fbox{$rain$}^*\fbox{$rain,\ dusk$}$ with $\Box^+\fbox{$noon$}\Box^+$, we get

$$\fbox{$rain,\ dawn$}\fbox{$rain$}^*\fbox{$rain, noon$}\fbox{$rain$}^*\fbox{$rain,\ dusk$}$$

which filters out the string $\fbox{$rain,\ dawn$}\fbox{$rain,\ dusk$}$ of length 2, and fleshes out the remaining strings in $\fbox{$rain,\ dawn$}\fbox{$rain$}^+\fbox{$rain,\ dusk$}$ by including *noon* in the middle. To capture the growth of information here, let us say that L *subsumes* L' and write $L \trianglerighteq L'$ if the superposition of L and L' includes L

$$L \trianglerighteq L' \quad \stackrel{\text{def}}{\iff} \quad L \subseteq L\&L'$$

(roughly: L is at least as informative as L' over the same temporal stretch). Conflating a string s with the singleton language $\{s\}$, it follows that L subsumes L' exactly if each string in L subsumes some string in L'

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

where $\trianglerighteq$ holds between strings of the same length related componentwise by inclusion

$$a_1 a_2 \cdots a_n \trianglerighteq b_1 b_2 \cdots b_m \quad \text{iff} \quad n = m \text{ and } a_i \supseteq b_i \text{ for } 1 \leq i \leq n\ .$$

For example, $\fbox{$\varphi, \psi$} \ \trianglerighteq\ \fbox{φ} \ \trianglerighteq\ \fbox{φ} + \fbox{ψ}$. As a type with instances $s \in L$, a language L is essentially a disjunction $\bigvee_{s \in L} s$ of conjunctions s (as is clear from the model-theoretic interpretations spelled out in [Fer04]).

To express requirements such as occurrences of *dawn* and *dusk* sandwich *noon*

$$\fbox{$dawn$}\Box^*\fbox{$dusk$} \quad \Rightarrow \quad \Box^+\fbox{$noon$}\Box^+\ ,$$

let us define

(i) for any string s, a *factor of* s to be a string x such that $s = uxv$ for some (possibly null) strings u and v, and

(ii) for any languages L and L', the *constraint* $L \Rightarrow L'$ to be the set of strings s such that every factor of s that subsumes L also subsumes L'

$$\begin{aligned} L \Rightarrow L' \quad \stackrel{\text{def}}{=} \quad & \{s \in \mathsf{Pow}(\Phi)^* \mid \text{(for every factor } x \text{ of } s) \\ & \qquad x \trianglerighteq L \text{ implies } x \trianglerighteq L'\} \end{aligned}$$

([BK03, FN05]). With $\Rightarrow$, we can define the φ-*bivalent* language

$$\square \quad \Rightarrow \quad \boxed{\varphi} + \boxed{\overline{\varphi}}$$

consisting of strings $a_1 a_2 \cdots a_n \in \mathsf{Pow}(\Phi)^*$ such that for each i from 1 to n (inclusive), $\varphi \in a_i$ or $\overline{\varphi} \in a_i$. While we may want to work with strings that do not belong to this language (allowing a string to be silent on φ), it makes sense to restrict our events to strings in the φ-*consistent* language

$$\boxed{\varphi, \overline{\varphi}} \quad \Rightarrow \quad \emptyset$$

requiring that no symbol in a string contain both φ and its negation $\overline{\varphi}$. Similarly, to require that a fluent r occur at most once in a string, the constraint

$$\boxed{\mathrm{r}} \square^* \boxed{\mathrm{r}} \quad \Rightarrow \quad \emptyset$$

precludes the occurrence of r in two different positions in a string. We may impose this on r equal to some moment now or other.

A Reichenbachian approach to tense and aspect locates the event time E relative to not only a speech time S but also a *reference time* R ([Rei47]). To see the point behind R, consider the pair (S2), (S3). (The oddness of (S3) is marked by the superscript ?.)

(S2) Pat left Dublin but is back (in Dublin).

(S3) ?Pat has left Dublin but is back (in Dublin).

In both (S2) and (S3), Pat's departure L_0 from Dublin

$$L_0 \stackrel{\text{def}}{=} \mathcal{L}(\text{Pat leave Dublin}) \trianglerighteq \boxed{\text{in}(p,d)} \square^* \boxed{\overline{\text{in}(p,d)}}$$

(where the fluent in(p, d) is read Pat-in-Dublin) precedes the speech time S

$$\begin{aligned} \mathcal{L}(\text{Pat left Dublin}) &\trianglerighteq L_0 \square^* \boxed{S} \\ \mathcal{L}(\text{Pat has left Dublin}) &\trianglerighteq L_0 \square^* \boxed{S} . \end{aligned}$$

R allows us to distinguish $\mathcal{L}$(Pat left Dublin) from $\mathcal{L}$(Pat has left Dublin), coinciding with the end of L_0 in the former case

$$\mathcal{L}(\text{Pat left Dublin}) \trianglerighteq \boxed{\text{in}(p,d)} \square^* \boxed{\overline{\text{in}(p,d)}, R} \square^* \boxed{S} \tag{9}$$

and with S in the latter

$$\mathcal{L}(\text{Pat has left Dublin}) \trianglerighteq \boxed{\text{in}(p,d)} \square^* \boxed{\overline{\text{in}(p,d)}} \square^* \boxed{R, S} . \tag{10}$$

In general, we apply aspectual and tense operators in sequence, forming, for instance,

$$\begin{aligned} \mathcal{L}(\text{Pat left Dublin}) &= \textsc{Past}(\textsc{Simp}(L_0)) \\ \mathcal{L}(\text{Pat has left Dublin}) &= \textsc{Pres}(\textsc{Perf}(L_0)) \end{aligned}$$

with tense operators for the past and present applied after operators for simple and perfect aspect. If an event with temporal projection E is represented by a language L, we get three aspectual operators on L.

$$
\begin{array}{rcll}
\textsc{Simp}(L) & \stackrel{\text{def}}{=} & L \;\&\; \Box^* \boxed{R} & \text{(i.e. } E = R) \\
\textsc{Prog}(L) & \stackrel{\text{def}}{=} & L \;\&\; \Box^+ \boxed{R} \Box^+ & \text{(i.e. } R \text{ contained in } E) \\
\textsc{Perf}(L) & \stackrel{\text{def}}{=} & L \Box^* \boxed{R} & \text{(i.e. } E < R)
\end{array}
$$

As a position from which to view the event represented by L, R says in the case of $\textsc{Simp}(L)$, that the event has reached completion; in the case of $\textsc{Prog}(L)$, that it has *not* quite gotten there yet but is on its way; and in $\textsc{Perf}(L)$, that it is history. Tense then locates S relative to R, coinciding with R for the present

$$\textsc{Pres}(L) \stackrel{\text{def}}{=} (L \;\&^{\Box} \boxed{S}) \cap \text{`}R = S\text{'}$$

and coming after R for the past

$$\textsc{Past}(L) \stackrel{\text{def}}{=} (L \;\&^{\Box} \boxed{S}) \cap \text{`}R < S\text{'}$$

where '$R = S$' is the set of strings

$$(\boxed{R} + \boxed{S}) \Rightarrow \boxed{R, S}$$

in which R and S co-occur, and '$R < S$' is the set of strings

$$(\boxed{R, S} + \boxed{S} \Box^* \boxed{R}) \Rightarrow \emptyset$$

in which R precedes S.

To account for the contrast between (S2) and (S3), it suffices that $\overline{\text{in}(p, d)}$ persist forward in $\mathcal{L}$(Pat has left Dublin)

$$\mathcal{L}(\text{Pat has left Dublin}) \trianglerighteq \Box^+ \boxed{\overline{\text{in}(p, d)}, S} \tag{11}$$

but not in $\mathcal{L}$(Pat left Dublin)

$$\mathcal{L}(\text{Pat left Dublin}) \not\trianglerighteq \Box^+ \boxed{\overline{\text{in}(p, d)}, S} \,. \tag{12}$$

If the postcondition $\overline{\text{in}(p, d)}$ of L_0 were to flow up to R, then we can derive (11) from (10) while respecting (12), given (9). Independent confirmation that R is a barrier to persistence (forward) is provided by the non-entailment

$$\text{Pat was in Dublin} \not\vdash \text{Pat is in Dublin}$$

where

$$\mathcal{L}(\text{Pat was in Dublin}) \trianglerighteq \boxed{\text{in}(p, d)}^* \boxed{\text{in}(p, d), R} \Box^* \boxed{S} \,.$$

If in(p, d) were to spill past R and over to S, $\boxed{\text{in}(p, d), S}$ would mean Pat is in Dublin.

We can express the persistence of an *inertial* fluent φ by introducing a non-inertial fluent $\mathsf{f}\overline{\varphi}$ that marks the application of a force against φ. The constraint

$$\boxed{\varphi}\,\Box \quad\Rightarrow\quad (\Box\boxed{\varphi} + \boxed{\mathsf{f}\overline{\varphi}}\,\Box + \boxed{R}\,\Box)$$

then says that φ persists to the next moment unless some force is applied against φ or R coincides with φ.[5] The persistence of the postcondition $\overline{\mathrm{in}(p,d)}$ of L_0 in (11) arises from the assumption that *no* force is applied against the fluent at the end of L_0. Instances where a force is applied against the postcondition surface as cases such as (S4) of non-persistence.

(S4) Pat has been to Dublin ($\not\models$ Pat is in Dublin)

The formation of the Reichenbachian perfect extends the event's temporal stretch, yielding a situation greater than the event. To extract the information contained in a string, it is useful to relax the strict temporal match-up in

$$L \;\trianglerighteq\; L',$$

weakening its second argument L' to the set $L'^{\Box}$ of strings in L' with any number of leading and trailing $\Box$'s deleted or added

$$\begin{aligned} L'^{\Box} &\stackrel{\text{def}}{=} \Box^{*}\mathsf{unpad}(L')\Box^{*} \\ &= \{s \mid \mathsf{unpad}(s) \in \mathsf{unpad}(L')\}\ . \end{aligned}$$

We define *weak subsumption* $\underline{\blacktriangleright}$ by

$$\begin{aligned} L\underline{\blacktriangleright}L' &\stackrel{\text{def}}{\iff} L \trianglerighteq L'^{\Box} \\ &\quad\text{iff}\quad \mathsf{unpad}(L) \subseteq L\&^{\Box}\, L'\ . \end{aligned}$$

Weak subsumption compares information content in the same way as $\trianglerighteq$

$$L\underline{\blacktriangleright}L' \quad\text{iff}\quad (\forall s \in L)(\exists s' \in L')\ s\underline{\blacktriangleright}s'\ .$$

but without insisting that strings have the same length

$$s\underline{\blacktriangleright}s' \quad\text{iff}\quad (\exists s'')\ \mathsf{unpad}(s'') = \mathsf{unpad}(s') \text{ and } s \trianglerighteq s''\ .$$

[5] Forces that make φ true appear in the backward persistence constraint

$$\Box\boxed{\varphi} \quad\Rightarrow\quad (\boxed{\varphi}\,\Box + \boxed{\mathsf{f}\varphi}\,\Box)$$

stating that if φ is true, it must have been so previously or else was forced to be true. The fluents $\mathsf{f}\varphi$ and $\mathsf{f}\overline{\varphi}$ were assumed identical in [FN05]. Differentiating them allows us to formulate the constraint "succeed unless opposed"

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

saying an unopposed force on φ brings φ about at the next moment. Were $\mathsf{f}\varphi$ and $\mathsf{f}\overline{\varphi}$ the same, (suo) would have no bite (denoting, as it would, the universal language $\mathsf{Pow}(\Phi)^{*}$).

An early paper on the relevance of inertia to linguistics is [Dow86]; later works include [Ste00] and [LH05], where a different formal approach to inertia is pursued.

Insofar as $L \underline{\blacktriangleright} L'$ says every instance of L contains some instance of L', it is natural to say L' *occurs in* L when $L \underline{\blacktriangleright} L'$. This generalizes the notion of occurrence discussed in the introduction.

Having weakened the second argument of $\trianglerighteq$, we now strengthen its first argument with background contextual information encoded by a set $C \subseteq \mathsf{Pow}(\Phi)^*$ of strings. To illustrate, an event $L = \fbox{rain,now}$ of *raining now* updates a background

$$C \quad = \quad (\Box + \fbox{rain} + \fbox{$\overline{\text{rain}}$})^+ \;\&\; (\Box^+ \fbox{now} \Box^+)$$

to give

$$C[L] \quad = \quad (\Box + \fbox{rain} + \fbox{$\overline{\text{rain}}$})^* \fbox{rain,now} (\Box + \fbox{rain} + \fbox{$\overline{\text{rain}}$})^* \ .$$

Evidently, neither the intersection $C \cap L$ nor the superposition $C \& L$ will do for the result $C[L]$ of updating C by L. Combining $\cap$ and $\&$, let

$$\begin{aligned} C[L] &\overset{\text{def}}{=} \{s \in C \mid s \underline{\blacktriangleright} L\} \\ &= (L^{\Box} \& C) \cap C \ . \end{aligned}$$

Not only does context grow eliminatively, $C[L] \subseteq C$, but $C[L] \underline{\blacktriangleright} L$, so that we can strengthen $\underline{\blacktriangleright}$ to a relation $\vdash^C$ sensitive to C, defining

$$L \vdash^C L' \quad \overset{\text{def}}{\Longleftrightarrow} \quad C[L] \underline{\blacktriangleright} L'$$

([Fer06]).

4 On to proof-theoretic semantics

How do the relations $\trianglerighteq$ and $\vdash^C$ square with proof-theoretic semantics? More notation is convenient. Given a string c and a language L, let

(i) $E_L(c)$ be the set of strings in L that c subsumes

$$E_L(c) \overset{\text{def}}{=} \{s \in L \mid c \trianglerighteq s\}$$

(ii) $\text{ext}(L, c)$ be the set of strings in L that occur in c

$$\text{ext}(L, c) \overset{\text{def}}{=} \{s \in L \mid c \underline{\blacktriangleright} s\}$$

(iii) $e_L(c)$ be some element of $E_L(c)$ if $E_L(c) \neq \emptyset$ else $e_L(c) = c$, and

(iv) $e(L, c)$ be some element of $\text{ext}(L, c)$ if $\text{ext}(L, c) \neq \emptyset$ else $e(L, c) = c$.

It follows that

$$\begin{aligned} L \trianglerighteq L' \quad &\text{iff} \quad (\forall x \in L)\ E_{L'}(x) \neq \emptyset \\ &\text{iff} \quad (\forall x \in L)\ e_{L'}(x) \in L' \end{aligned}$$

and

$$L \vdash^C L' \quad \text{iff} \quad (\forall x \in C)(\forall y \in \mathrm{ext}(L,x))\ \mathrm{ext}(L',x) \neq \emptyset$$
$$\quad\quad\quad\quad \text{iff} \quad (\forall x \in C)(\forall y \in \mathrm{ext}(L,x))\ e(L',x) \in L' \ .$$

Adopting the familiar sequent format $\Gamma \longmapsto \Theta$ for an assertion (or judgment) Θ based on assumptions (or context) Γ, we might then express $L \trianglerighteq L'$ as

$$x \in L \quad \longmapsto \quad e_{L'}(x) \in L' \tag{13}$$

and $L \vdash^C L'$ as

$$x \in C,\ y \in \mathrm{ext}(L,x) \quad \longmapsto \quad e(L',x) \in L' \ . \tag{14}$$

If we are to replace $e_{L'}(x)$ and $e(L',x)$ by terms $t(x)$ and $t'(x,y)$ that can be formed through a system of rules, we must beef L' up to $E_{L'}(x)$ and $\mathrm{ext}(L',x)$ respectively,[6] resulting in

$$x \in L \quad \longmapsto \quad t(x) \in E_{L'}(x) \tag{15}$$

for (13) and

$$x \in C,\ y \in \mathrm{ext}(L,x) \quad \longmapsto \quad t'(x,y) \in \mathrm{ext}(L',x) \tag{16}$$

for (14). In both (15) and (16), the first variable typed, x, relativizes the languages of all other terms appearing in the sequent (L' in (15), and L, L' in (16)). To isolate the variable x to the left of $\longmapsto$, we can put (16) as

$$x \in C \quad \longmapsto \quad (\lambda y \in \mathrm{ext}(L,x))\, t'(x,y) \ \in \ \mathrm{ext}(L,x) \to \mathrm{ext}(L',x)$$

with a function $(\lambda y \in \mathrm{ext}(L,x))\, t'(x,y)$ to the right of $\longmapsto$. The example of $e(L',x)$ from (14) for $t'(x,y)$ suggests the function need not consult y during its computation; the typing $y \in \mathrm{ext}(L,x)$ serves simply to ensure that the search for a string in $\mathrm{ext}(L',x)$ succeeds, provided $L \vdash^C L'$. At any rate, we need not worry about termination of the computation, so long as we bound the search to strings occurring in (or, in the case of $t(x)$, subsumed by) x.

With these functions in mind, let us call a function f on strings over the alphabet $\mathsf{Pow}(\Phi)$ a $\trianglerighteq$*-map* if for all $s \in \mathsf{Pow}(\Phi)^*$, $s \trianglerighteq f(s)$. If we collect in $L \overset{\trianglerighteq}{\to} L'$ all $\trianglerighteq$-maps that send strings in L to strings in L'

$$f \in L \overset{\trianglerighteq}{\to} L' \quad \overset{\text{def}}{\iff} \quad f \text{ is a } \trianglerighteq\text{-map and } (\forall s \in L)\ f(s) \in L'$$

then we can reduce $L \trianglerighteq L'$ to the non-emptiness of $L \overset{\trianglerighteq}{\to} L'$

$$L \trianglerighteq L' \quad \text{iff} \quad L \overset{\trianglerighteq}{\to} L' \text{ is inhabited (i.e. non-empty)} \ .$$

[6]This is because we can no longer rely on the fact that for all languages L and strings x, $e_L(x) \trianglerighteq x$ and $e(L,x) \underline{\blacktriangleright} x$.

Weakening subsumption $\trianglerighteq$, we define a function f on strings over the alphabet $\mathsf{Pow}(\Phi)$ to be a $\underline{\blacktriangleright}$-*map* if for all $s \in \mathsf{Pow}(\Phi)^*$, $s \underline{\blacktriangleright} f(s)$. Then $L \vdash^C L'$ reduces to the existence of a $\underline{\blacktriangleright}$-map from $C[L]$ to L'

$$L \vdash^C L' \quad \text{iff} \quad \text{there is a } \underline{\blacktriangleright}\text{-map } f \text{ such that } (\forall s \in C[L])\ f(s) \in L' \ .$$

We can go beyond any single (finite) string c to a set C of strings that approximates a world, assuming C is $\underline{\blacktriangleright}$-*directed* in that for all $c, c' \in C$, there exists $c'' \in C$ such that $c'' \underline{\blacktriangleright} c$ and $c'' \underline{\blacktriangleright} c'$. Forming the union of their extensions

$$\text{ext}(L, C) \ \stackrel{\text{def}}{=}\ \bigcup_{c \in C} \text{ext}(L, c) \ ,$$

it is natural to modify (16) to

$$x \in C,\ y \in \text{ext}(L, x) \quad \longmapsto \quad \hat{t}(x, y) \in (\exists z \in C)\, \text{ext}(L', z) \tag{17}$$

for which the problem is how to bound the variable z. Our approach of building an inference relation $\vdash^C$ by incorporating background assumptions C into the first argument of $\underline{\blacktriangleright}$ shifts the complexity from proof terms $t, t', \ldots$ to C. Constraints $L \Rightarrow L'$ are useful in forming C, and involve $\trianglerighteq$-maps insofar as

$$s \in L \Rightarrow L' \quad \text{iff} \quad \{x \in \text{Fac}(s) \mid x \trianglerighteq L\} \ \trianglerighteq L'$$

where Fac(s) is the set of factors of s

$$\text{Fac}(s) \ \stackrel{\text{def}}{=}\ \{x \in \mathsf{Pow}(\Phi)^* \mid (\exists u, v \in \mathsf{Pow}(\Phi)^*)\ s = uxv\} \ .$$

There are variants of $\Rightarrow$ that apply to various cases of (17), described in [FN05], drawing on methods in [BK03].

Acknowledgments

This paper is an update on my invited talk at SPR 2001. I thank the organizers of that workshop and editors of this volume, especially Kepa Korta for his patience. I am grateful also to Yukinori Takubo, Takao Gunji, Ikumi Imani, Hiroshi Mito and Norihiko Ogata for much appreciated interest.

References

[All83] James F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

[Aus61] John L. Austin. Truth. In J.O. Urmson and G.J. Warncock, editors, *Philosophical Papers*. Oxford University Press, 1961.

[BK03] Kenneth R. Beesley and Lauri Karttunen. *Finite State Morphology*. CSLI Publications, Stanford, 2003.

[BP83] Jon Barwise and John Perry. *Situations and Attitudes*. MIT Press, Cambridge, Mass., 1983.

[Coo05] Robin Cooper. Austinian truth, attitudes and type theory. *Research on Language and Computation*, 3(4):333–362, 2005.

[Dav67] Donald Davidson. The logical form of action sentences. In Nicholas Rescher, editor, *The Logic of Decision and Action*, pages 81–95. University of Pittsburgh Press, 1967. Reprinted in [Dav80].

[Dav80] Donald Davidson. *Essays on Actions and Events.* Clarendon Press, Oxford, 1980.

[Dow86] David R. Dowty. The effects of aspectual class on the temporal structure of discourse: semantics or pragmatics? *Linguistics and Philosophy*, 9:37–61, 1986.

[Eme92] E. Allen Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B: Formal Methods and Semantics, pages 995–1072. MIT Press, 1992.

[Fer04] Tim Fernando. A finite-state approach to events in natural language semantics. *Journal of Logic and Computation*, 14(1):79–92, 2004.

[Fer06] Tim Fernando. Finite-state temporal projection. In *Proc. 11th International Conference on Implementation and Application of Automata*, Lecture Notes in Computer Science 4094, pages 230–241. Springer-Verlag, 2006.

[FN05] T. Fernando and R. Nairn. Entailments in finite-state temporality. In *Proc. 6th International Workshop on Computational Semantics*, pages 128–138. Tilburg University, 2005.

[Kap79] David Kaplan. On the logic of demonstratives. *Journal of Philosophical Logic*, 8(1), 1979.

[KR93] H. Kamp and U. Reyle. *From Discourse to Logic.* Kluwer Academic Publishers, Dordrecht, 1993.

[LH05] M. van Lambalgen and F. Hamm. *The Proper Treatment of Events.* Blackwell, 2005.

[MH69] J. McCarthy and P. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In M. Meltzer and D. Michie, editors, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, 1969.

[ML84] Per Martin-Löf. *Intuitionistic Type Theory.* Bibliopolis, Napoli, 1984. Notes by Giovanni Sambin of a series of lectures given in Padua, June 1980.

[Par90] Terence Parsons. *Events in the Semantics of English: A Study in Subatomic Semantics.* MIT Press, Cambridge, MA, 1990.

[Per97] John Perry. Situation semantics. In *Routledge Encyclopedia of Philosophy*. London, Routledge, 1997.

[Ran94] Aarne Ranta. *Type-Theoretical Grammar*. Oxford University Press, Oxford, 1994.

[Rei47] Hans Reichenbach. *Elements of Symbolic Logic*. London, Macmillan, 1947.

[Ste00] Mark Steedman. *The Productions of Time*. Draft, ftp://ftp.cogsci.ed.ac.uk/pub/steedman/temporality/temporality.ps.gz, July 2000. Subsumes 'Temporality,' in J. van Benthem and A. ter Meulen, editors, *Handbook of Logic and Language*, pages 895–935, Elsevier North Holland, 1997.