J Philos Logic (2011) 40:371–396
DOI 10.1007/s10992-010-9155-1

Constructing Situations and Time

Tim Fernando

Received: 17 February 2009 / Accepted: 30 August 2010 / Published online: 16 September 2010

Abstract Situations serving as partial worlds as well as events in natural language semantics are constructed from a type-theoretic interpretation of first-order formulae and (after a type reduction) temporal formulae. Limitations of the Russell-Wiener-Kamp derivation of time from events are discussed and overcome to give a more widely applicable account of temporal granularity. Finite situations are formulated as strings of observations, conceptualized to persist inertially (in the absence of forces).

Keywords Situations · Events · Time · Strings · Natural language semantics · Types

1 Introduction

Situations have been part of the toolkit of natural language semantics since at least [29], coming to the fore in *situation semantics*, "the leading idea" of which 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. [4, p. 19]

The aforementioned relation can be construed as an instance of what is called the *character of* a sentence A in [20], a function

$$\mathrm{char}_A : X \to 2^Y$$

T. Fernando (✉)
Trinity College Dublin, Dublin, Ireland
e-mail: tfernand@tcd.ie

from a set X of "contexts of use" to the family 2^Y of subsets of a set Y of "circumstances of evaluation." Put utterances u into X and situations s into Y, and let

$$u \ A\text{-describes } s \quad \text{iff} \quad s \in \text{char}_A(u)$$

so that

$$\text{char}_A(u) = \{s \in Y \mid u \ A\text{-describes } s\} .$$

It is customary, at a fixed context u, to read

$$\text{'}s \in \text{char}_A(u)\text{' as } s \text{ supports the truth of } A$$

and to interpret conjunction $\wedge$ as intersection $\cap$

$$\text{char}_{A \wedge B}(u) \quad = \quad \text{char}_A(u) \cap \text{char}_B(u) \tag{1}$$

inasmuch as at u,

$$\begin{aligned} s \text{ supports the truth of } A \wedge B \quad \text{iff} \quad & s \text{ supports the truth of } A \ \text{ and} \\ & s \text{ supports the truth of } B. \end{aligned}$$

In his influential critique [32] of [4], Soames argues that because of equations such as (1), the "semantic value of a sentence at a context" cannot be a set of "truth-supporting circumstances" whether these circumstances are possible worlds or parts thereof, called situations. One way around Soames' argument is to sharpen the sets $\text{char}_A(u)$ of truth-supporting circumstances to types[1] $[\![A]\!]_u$ that do not reduce conjunction to intersection

$$[\![A \wedge B]\!]_u \ \neq \ [\![A]\!]_u \cap [\![B]\!]_u .$$

Under the *formulae-as-types* paradigm from proof theory (e.g. [37]), $[\![A]\!]_u$ is the type of proofs of A at context u

$$[\![A]\!]_u \ = \ \{a \mid a \text{ is a proof of } A \text{ at context } u\}$$

and conjunctions are interpreted by pairing proofs of conjuncts

$$[\![A \wedge B]\!]_u \ = \ \{\langle a, b\rangle \mid a \in [\![A]\!]_u \text{ and } b \in [\![B]\!]_u\} .$$

The pairs $\langle a, b\rangle$ are ordered

$$\langle a, b\rangle = \langle a', b'\rangle \quad \text{iff} \quad a = a' \ \text{ and } \ b = b'$$

so that in general,

$$[\![A \wedge B]\!]_u \neq [\![B \wedge A]\!]_u \quad \text{whereas} \quad \text{char}_{A\wedge B}(u) = \text{char}_{B\wedge A}(u) .$$

Proof-theoretic interpretations of logical connectives, especially as formulated in Intuitionistic Type Theory [22], have in recent years been applied to natural

[1]For our purposes, the words set, type and collection are interchangeable, and are to be understood within ordinary set theory in classical logic.

Table 1 Ingredients for evaluating the meaning of A

Kaplan scheme	Classic instance	Proposed instance
Context of use	Speaker, time, ...	u
Evaluation circumstances	Possible world	Intensional index i
Extension	Truth value $\in \{0, 1\}$	Type $[\![A]\!]_{u,i}$

language semantics (e.g. [7, 16, 28]). A basic worry expressed early on by Sundholm is that

> ... it is not at all clear that one can export the 'canonical proof-objects' conception of meaning outside the confined area of constructive mathematics. In particular, the treatment of atomic sentences such as OWN[x, y] is left intolerably vague ... and it is an open problem how to remove that vagueness. [34, p. 503]

To address such concerns about atomic formulae A, a subscript i serving, in Kaplan's terms, as circumstances of evaluation is used in [12] to form the type $[\![A]\!]_{u,i}$ subject to the constraint

$$s \in [\![A]\!]_{u,i} \text{ implies } s \sqsubseteq i \qquad \text{(for atomic } A\text{)}$$

where $\sqsubseteq$ is a suitable "part-of" relation. More on $\sqsubseteq$ below. For now, suffice it to say i is an "intensional index" that together with u determines the type $[\![A]\!]_{u,i}$ in place of a truth value $\in \{0, 1\}$ (Table 1). The intuition is that

$$i \text{ supports the truth of } A \text{ at } u \quad \text{iff} \quad [\![A]\!]_{u,i} \neq \emptyset$$

leading to a reconstruction of the character of A relative to u as the set of intensional indices supporting the truth of A at u.

$$\text{char}_A(u) \;=\; \{i \mid [\![A]\!]_{u,i} \neq \emptyset\} \qquad \text{(ch)}$$

Under (ch), $[\![A]\!]_{u,i}$ can be regarded as the set of truthmakers for the truth-bearer determined by the triple A, u, i,[2] with the understanding that

the proposition A, u, i express is true iff a truthmaker for it exists.

Character and Soames' problem aside, what use is the set $[\![A]\!]_{u,i}$ of truthmakers?

The distinction between $s \in \text{char}_A(u)$ and $s \in [\![A]\!]_{u,i}$ aligns with distinctions independently made in [30] and [21] to analyze event prediction, causal statements and definite descriptions, as well as truth (see Table 2).[3] The

[2]The term "truth-maker" was introduced in [24] as a neutral label for an entity that makes a proposition or "truthbearer" true. A truthful statement need *not* pick out a unique truthmaker. To extract a truthbearer from a sentence A, an utterance u and an intensional index i are required.

[3]To keep the context for the relations in Table 2 alike (i.e., u), one can redefine a truthmaker of A at u to be a pair $\langle i, s\rangle$ such that $s \in [\![A]\!]_{u,i}$. For the comparison with Kaplan's character in Table 1, however, it is convenient to separate i from s.

Table 2 Two relations between s and A (for some fixed, implicit context)

Schubert	Kratzer	Situation s
$A * s$	s supports A	index $[\![A]\!]_{u,s} \neq \emptyset$
$A ** s$	s exemplifies A	truthmaker $s \in [\![A]\!]_{u,i}$

relation $s \in \text{char}_A(u)$ allows for some slack between s and A insofar as it can be expected to be persistent relative to the part-of relation $\sqsubseteq$ on situations

$$s \in \text{char}_A(u) \text{ and } s \sqsubseteq s' \quad \text{implies} \quad s' \in \text{char}_A(u)$$

(which, as noted in Section 3 below, often but not always holds). By contrast, the fit between s and A must be tightened if we are to require

the event time of a sentence is the temporal
projection of a truthmaker of the sentence (tp)

where event time is conceived as in the analysis of tense in, for instance [29]. Fleshing out the link between the notion of time underlying (tp) and the types $[\![A]\!]_{u,i}$ behind (ch) is the main aim of the present paper, which is organized as follows.

In Section 2, we supply details of the types $[\![A]\!]_{u,i}$ for first-order formulae A, where i is a model M (appropriate for the vocabulary of A) and u is a partial function mapping variables to objects in the universe $|M|$ of M. In Section 3, we focus on the special case where the domain of u is a singleton set consisting of a variable ranging over temporal points on which to base a Kripke semantics for *Linear Temporal Logic* (LTL, e.g. [10]). The pairs and functions serving as truthmakers in Section 2 are reduced systematically to schedules in Section 3, where they play the role of both intensional indices and truthmakers, linked by a part-of relation $\sqsubseteq$. As truthmakers, schedules can be construed as events [8], orderings on which are well-known to induce notions of time [18, 35, 36]. We take up these temporal notions in Section 4, introducing pre- and post-states that lead to temporal propositions subject to inertia in Section 5, where schedules are finitized to strings. Tables 3 and 4 summarize what lies ahead.

2 Pairs and Functions as Truthmakers

Recall that first-order formulae A are related to models M and functions u mapping variables to objects in M by satisfaction $\models$, with clauses such as

$$M, u \models A \wedge B \quad \text{iff} \quad M, u \models A \quad \text{and} \quad M, u \models B$$

Table 3 Situation in Sections 2 and 3

	Section 2	Section 3
Formula	First-order	LTL
Intensional index	Model	Schedule
Truthmaker	Pair, function	Schedule

Table 4 Time in Sections 4 and 5

	Section 4	Section 5
Situation	Schedule	String
Projection	Russell-Wiener-Kamp	Block compression
Bound	Pre- and post-states	Inertia (force)

leading to (1).

$$\mathrm{char}_{A\wedge B}(u) \;=\; \mathrm{char}_A(u) \cap \mathrm{char}_B(u) \tag{1}$$

Let us assume that for atomic formulae A, we have types $[\![A]\!]_{u,M}$ that are non-empty precisely when A is $\models$-satisfied by M, u

$$M, u \models A \quad \text{iff} \quad [\![A]\!]_{u,M} \neq \emptyset. \tag{$\models_A$}$$

There are any number of ways to establish the biconditional ($\models_A$), including

$$[\![A]\!]_{u,M} \;=\; \begin{cases} \{A\} & \text{if } M, u \models A \\ \emptyset & \text{otherwise} \end{cases} \tag{2}$$

(arguably the worst example of "armchair ontology" decried in [31]). For arbitrary formulae A, we will not take the biconditional ($\models_A$) for granted by asserting say, (2), but will instead adopt proof-theoretic interpretations of logical connectives such as

$$\text{a proof of } A \wedge B \text{ is a pair } \langle a, b\rangle \text{ of a proof } a \text{ of } A \text{ and a proof } b \text{ of } B. \tag{$\wedge$p}$$

As we shall see, such interpretations nevertheless extend ($\models_A$) from atomic formulae A to arbitrary first-order formulae A.

Given proofs of atomic formulae, pairs and functions are built in the simplest case through the Cartesian product

$$I \times J \;=\; \{\langle i, j\rangle \mid i \in I \text{ and } j \in J\}$$

and function space

$$I \to J \;=\; \text{set of functions with domain } I \text{ and range } J$$

for various sets I and J. In particular, ($\wedge$p) amounts to

$$[\![A \wedge B]\!]_{u,M} \;=\; [\![A]\!]_{u,M} \times [\![B]\!]_{u,M} \tag{3}$$

and the clause

$$\text{a proof of } A \supset B \text{ is a function mapping proofs of } A \text{ to proofs of } B \tag{$\supset$p}$$

can be restated concisely as

$$[\![A \supset B]\!]_{u,M} \;=\; [\![A]\!]_{u,M} \to [\![B]\!]_{u,M}\,. \tag{4}$$

For quantification $(\forall x)A$ and $(\exists x)A$, it is convenient to work with dependent forms $\prod_{i\in I} \hat{J}(i)$ and $\sum_{i\in I} \hat{J}(i)$ of $I \to J$ and $I \times J$ (respectively), where $\hat{J}$ is a set-valued function with domain I. We put functions that map $i \in I$ to objects in $\hat{J}(i)$ into

$$\prod_{i \in I} \hat{J}(i) \;=\; \left\{ f{:}I \to \bigcup_{i \in I} \hat{J}(i) \mid f(i) \in \hat{J}(i) \text{ for each } i \in I \right\}$$

and input/output pairs for such functions into

$$\sum_{i \in I} \hat{J}(i) \;=\; \left\{ \langle i, j \rangle \mid i \in I \text{ and } j \in \hat{J}(i) \right\} .$$

Notice that if $\hat{J}$ is the constant function mapping each $i \in I$ to $\hat{J}(i) = J$, then

$$\prod_{i \in I} \hat{J}(i) = I \to J \quad \text{and} \quad \sum_{i \in I} \hat{J}(i) = I \times J.$$

Next, to interpret $(\exists x)A$ and $(\forall x)A$ at u, M, we define, given any object a, a function u^x_a identical to u except possibly at x which u^x_a maps to a

$$u^x_a(y) \;=\; \begin{cases} a & \text{if } y = x \\ u(y) & \text{if } y \in \mathsf{domain}(u) - \{x\} . \end{cases}$$

Functions mapping $a \in |M|$ to proofs in $[\![A]\!]_{u^x_a, M}$ make up

$$[\![(\forall x)A]\!]_{u,M} \;=\; \prod_{a \in |M|} [\![A]\!]_{u^x_a, M} \tag{5}$$

while pairs $\langle a, b \rangle$ of objects $a \in |M|$ and $b \in [\![A]\!]_{u^x_a, M}$ go into

$$[\![(\exists x)A]\!]_{u,M} \;=\; \sum_{a \in |M|} [\![A]\!]_{u^x_a, M} . \tag{6}$$

For negation, it is customary to identify $\neg A$ with $A \supset \bot$ for some "absurd" formula $\bot$

$$[\![\bot]\!]_{u,M} \;=\; \emptyset \tag{7}$$

with $M, u \not\models \bot$.

Theorem 1 *Clauses* (3)–(7) *preserve the biconditional*

$$M, u \models A \quad \textit{iff} \quad [\![A]\!]_{u,M} \neq \emptyset. \tag{$\models_A$}$$

That is, if ($\models_A$) *holds for atomic formulae* A *then it holds for all formulae* A *built from them with* $\wedge, \supset, \forall, \exists, \bot$ *and interpreted according to* (3)–(7).

Theorem 1 says that satisfaction $\models$ falls out of clauses (3)–(7) applied to interpretations of atomic formulae respecting $\models$.

Beyond the first-order connectives above, dependent forms $\wedge_x$ and $\supset_x$ of conjunction $\wedge$ and implication $\supset$ are interpretable through the dependent type constructs $\sum$ and $\prod$

$$
\begin{aligned}
[\![A \wedge_x B]\!]_{u,M} &= \sum_{a \in [\![A]\!]_{u,M}} [\![B]\!]_{u^x_a, M} \\
[\![A \supset_x B]\!]_{u,M} &= \prod_{a \in [\![A]\!]_{u,M}} [\![B]\!]_{u^x_a, M}
\end{aligned}
$$

extending the mechanisms of *Discourse Representation Theory* (DRT, [19]) for anaphora and presupposition (e.g. [28]). Whereas first-order formulae can dispense with truthmakers (Theorem 1), the connectives $\wedge_x$ and $\supset_x$ explicitly manipulate them through the variable x. Interpretations can be extracted from that variable of, for instance, the problematic pronouns *he* and *it* in the donkey sentence (a).

(a) If a farmer owns a donkey, he beats it.
(b) Whenever a man rides a donkey, he gives it a treat.

The variant (b), from [21], requires, in addition, that instances of riding be related to instances of treat-giving (lest a man ride a donkey repeatedly but give it a treat only once; [12]). These instances are events e that occur in [19] within predications such as e : ride(m, d). In [3], these events get beefed up into *main eventualities* that semantically underpin rhetorical relations between discourse segments. In both cases, they are ontological commitments recorded as *discourse referents*, distinct from truthbearers (called *conditions*). As truthmakers of atomic formulae, the precise nature of events is left open-ended under the proof-conditional clauses (3)–(7) above.

3 Partial Indices as Type-Reduced Pairs and Functions

Clauses (3)–(7) from the previous section yield truthmakers for non-atomic formulae A different from the armchair example (2)

$$
[\![A]\!]_{u,M} = \begin{cases} \{A\} & \text{if } M, u \models A \\ \emptyset & \text{otherwise.} \end{cases} \tag{2}
$$

Instead, functions and pairs are formed according to the logical connectives in the truthbearer, adhering closely to the principle (lft)

the logical complexity of a sentence shapes the form of its truthmakers. (lft)

For example, a truthmaker of $(A \wedge B) \supset (\forall x)C$ is a function that maps certain pairs (making $A \wedge B$ true) to certain functions (making $(\forall x)C$ true). In [24], Mulligan, Simons and Smith attack "the dogma of logical form," upholding instead

> the independence of ontological from logical complexity: ontologically complex objects (those having proper parts) are not for that reason also

> in some way logically complex, any more than there is reason to suppose that to every logically complex (true) sentence there corresponds an ontologically complex entity which makes it true. [24, Section 3]

Smith presses on in [31], arguing that

> there is no superficial feature (for example, the logical form of the corresponding sentence) which will allow us to determine in some quasi-automatic fashion the totality of all of that to which reference is made in a given judgment. [31, Section 12]

In response, one might deny that logical form is a superficial feature of a sentence. Indeed, if work in DRT [3, 19] has taught us anything, it is that the construction of logical forms for English texts is a challenging task that must draw on all kinds of information to uncover what the texts leave implicit. Beyond making allowances for complications in the step from English to logic, however, there is also the narrower issue of how logical forms are interpreted. Under clauses (4) and (7) from the previous section, negations $\neg A$, equated with $A \supset \bot$, can only be made true by the empty function $\emptyset$

$$\begin{aligned}
[\![\neg A]\!]_{u,M} &= [\![A]\!]_{u,M} \to \emptyset \\
&= \begin{cases} \{\emptyset\} & \text{if } [\![A]\!]_{u,M} = \emptyset \\ \emptyset & \text{otherwise.} \end{cases}
\end{aligned}$$

Hence, negating A twice leads to truthmakers worse than (2)

$$[\![\neg\neg A]\!]_{u,M} = \begin{cases} \{\emptyset\} & \text{if } M, u \models A \\ \emptyset & \text{otherwise} \end{cases}$$

assuming ($\models_A$) is satisfied

$$M, u \models A \quad \text{iff} \quad [\![A]\!]_{u,M} \neq \emptyset. \tag{$\models_A$}$$

The empty set is a terrible truthmaker, giving no indication whatsoever of what a truthbearer is about.

An alternative to $\neg A$ providing proofs other than $\emptyset$ is *strong negation* $\overline{A}$ [25], which, as will become clear presently, points to the importance of atomic formulae (taken for granted in the previous section). The idea is to pair every n-ary relation symbol R with another $\overline{R}$ of the same arity (doubling the relation symbols, if necessary), with the proviso that $\overline{\overline{R}} = R$ and the understanding that a model M interprets $\overline{R}$ as R's falsity set

$$\overline{R}^M = |M|^n - R^M. \tag{8}$$

The map $\overline{R(t_1, \ldots, t_n)} = \overline{R}(t_1, \ldots, t_n)$ is then extended to non-atomic formulae by pushing the negations inside (into atomic formulae closed under negation) following De Morgan-like laws

$$\begin{aligned}
\overline{A \supset B} &= A \wedge \overline{B} & \qquad \overline{A \wedge B} &= A \supset \overline{B} \\
\overline{(\forall x)A} &= (\exists x)\overline{A} & \qquad \overline{(\exists x)A} &= (\forall x)\overline{A}
\end{aligned}$$

so that $\overline{\overline{A}} = A$. For $\bot$, we put $\overline{\bot} = \top$ and $\overline{\top} = \bot$, where $M, u \models \top$ and

$$[\![\top]\!]_{u,M} = \{\emptyset\} .$$

It is easy to see that although $\overline{A}$ and $\neg A$ may have different truthmakers

$$[\![\overline{A}]\!]_{u,M} \neq [\![\neg A]\!]_{u,M}$$

they agree on satisfaction $\models$ inasmuch as

$$\begin{aligned} [\![\overline{A}]\!]_{u,M} \neq \emptyset \quad &\text{iff} \quad [\![\neg A]\!]_{u,M} \neq \emptyset \\ &\text{iff} \quad M, u \not\models A \end{aligned}$$

assuming ($\models_A$) and (8).

Next, generalizing from first-order models M to intensional indices i that may fall short of worlds (in failing to settle the truth/falsity of sentences), it is useful to weaken Boolean complementation (8) to

$$R^i \cap \overline{R}^i = \emptyset . \tag{9}$$

Assuming a pairing R, $\overline{R}$ of relation symbols over indices i respecting (9), the notion of a diagram in model theory [6, p. 68] generalizes to the *diagram* $\Delta(i)$ of i given by the set

$$\Delta(i) = \{\langle u, A\rangle \mid [\![A]\!]_{u,i} \neq \emptyset \text{ and } A \text{ is atomic}\}$$

of pairs $\langle u, A\rangle$ of contexts u and atomic formulae A for which $[\![A]\!]_{u,i}$ is non-empty. Diagrams induce a natural partial order $\sqsubseteq$ on intensional indices

$$i \sqsubseteq i' \quad \text{iff} \quad \Delta(i) \subseteq \Delta(i'). \tag{10}$$

For models M and M' of the same vocabulary, $\sqsubseteq$ is equality

$$M \sqsubseteq M' \quad \text{iff} \quad M = M'$$

but situations more partial than possible worlds give rise to intensional indices i and i' with $i \sqsubseteq i'$ but $i \neq i$. Clearly, for atomic formulae A, we have persistence

$$i \sqsubseteq i' \text{ and } [\![A]\!]_{u,i} \neq \emptyset \quad \text{implies} \quad [\![A]\!]_{u,i'} \neq \emptyset .$$

What about non-atomic formulae A? Well-known counter-examples to persistence are universal formulae such as (c).

(c) Every mistake was corrected.

$$(\forall x)(\text{mistake}(x) \supset \text{corrected}(x))$$

The culprit is the function construct $\prod$ for interpreting $\forall$ and $\supset$. (The pair construct $\sum$, by contrast, poses no obstacle to persistence.) If, however, we apply $\prod$ judiciously, it turns out we can secure persistence for many non-atomic formulae A. We illustrate this point in the remainder of this paper, exploring what T. 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. [26, p. ix]

The truthmakers for many such simple sentences are events that can be conceived as strings of snapshots—a conception arguably implicit in Linear Temporal Logic (LTL, [10, 27]).

LTL provides a language $\mathcal{L}(P)$ of temporal formulae φ built from some set P of atomic temporal formulae. Following [13], we shall define sets $[\![\varphi]\!]_{u,i}$ of truthmakers where

(i) contexts u of use are given by integers $n \in \mathbb{Z} = \{0, 1, -1, 2, -2, \ldots\}$ representing temporal points in a discrete linear order without endpoints, and
(ii) intensional indices i are given by relations $s \subseteq \mathbb{Z} \times P$ between $\mathbb{Z}$ and P, called *schedules* [13] such that

$$s \;=\; \Delta(s) \;=\; \{\langle n, p\rangle \in \mathbb{Z} \times P \mid s, n \models p\}$$

and the temporal span of s is its domain $\{n \in \mathbb{Z} \mid (\exists p \in P)\ \langle n, p\rangle \in s\}$.

It will be convenient to assume the set P of atomic temporal formulae is closed under a strong negation map $\bar{\cdot}$

$$\text{for all } p \in P, \quad \overline{p} \in P$$

and that the schedules s are coherent in that for all $n \in \mathbb{Z}$ and $p \in P$,

$$\langle n, p\rangle \in s \quad \text{implies} \quad \langle n, \overline{p}\rangle \notin s$$

(the analogue of (9) above). In general, it will not be true of a schedule s that for all $n \in \mathbb{Z}$ and $p \in P$,

$$\langle n, p\rangle \in s \quad \text{or} \quad \langle n, \overline{p}\rangle \notin s$$

unless s is a world — i.e., s is $\subseteq$-maximal among the schedules. (As $s = \Delta(s)$, $\subseteq$ is the same as the partial order $\sqsubseteq$ defined in (10).) As explained in [13], the sets $[\![\varphi]\!]_{u,i} = [\![\varphi]\!]_{n,s}$ can be derived from certain sets $[\varphi]_n \subseteq 2^{\mathbb{Z}\times P}$ via

$$[\![\varphi]\!]_{n,s} \;=\; \left\{s' \in [\varphi]_n \mid s' \subseteq s\right\} \tag{11}$$

preserving the usual Kripke semantics $\models$ for LTL in that for all worlds s,

$$s, n \models \varphi \quad \text{iff} \quad [\![\varphi]\!]_{n,s} \neq \emptyset\,. \tag{12}$$

An immediate consequence of (11) is that all LTL formulae φ are persistent over schedules s and s'

$$s, n \models \varphi \text{ and } s \subseteq s' \quad \text{implies} \quad s', n \models \varphi\,.$$

What we establish next (beyond the results reported in [13]) is that the sets $[\varphi]_n$ can be obtained from systematically reducing pairs and functions that interpret φ at n through the constructs $\sum$ and $\prod$ from the previous section. Useful truthmakers can be derived from the interpretation of logical connectives given by clauses (3)–(7). More precisely, sets $\{\!|\varphi|\!\}_n$ of pairs and functions are defined such that for some reduction π of these pairs and functions to schedules,

$$[\varphi]_n \;=\; \{\pi(a) \mid a \in \{\!|\varphi|\!\}_n\}$$

whence as a corollary of (12), we have

Theorem 2 *For every LTL-formula φ, schedule s and integer n,*

$$s, n \models \varphi \quad iff \quad (\exists a \in \{\!|\varphi|\!\}_n)\, \pi(a) \subseteq s\ .$$

We owe a definition of the sets $\{\!|\varphi|\!\}_n$ and the type reduction π. Assuming that the elements of $\{\!|\varphi|\!\}_n$ are in the simplest case pairs $\langle n, p\rangle \in \mathbb{Z} \times P$ and constructed inductively as pairs $\langle a, a'\rangle$ and functions f, it is easy to define π

$$\begin{aligned} \pi(\langle n, p\rangle) &= \{\langle n, p\rangle\} \\ \pi(\langle a, a'\rangle) &= \pi(a) \cup \pi(a') \\ \pi(f) &= \bigcup\{\pi(f(n)) \mid n \in \mathsf{domain}(f)\}\ . \end{aligned}$$

As for the sets $\{\!|\varphi|\!\}_n$, we need to be clearer about the formulae at stake and how they get evaluated through $\models$.

In the case of an atomic temporal formula $p \in P$,

$$\{\!|p|\!\}_n \;=\; \{\langle n, p\rangle\}$$

corresponding to

$$s, n \models p \quad \text{iff} \quad \langle n, p\rangle \in s\ .$$

As already mentioned, we use strong negation, which involves pairing not only atomic temporal formulae with their duals, but also logical/temporal connectives. To sidestep persistence complications with $\supset$, we take disjunction $\vee$ as the dual of $\wedge$ (instead of $\supset$), setting

$$\{\!|\varphi \wedge \psi|\!\}_n \;=\; \{\!|\varphi|\!\}_n \times \{\!|\psi|\!\}_n$$

and

$$\{\!|\varphi \vee \psi|\!\}_n \;=\; \{\!|\varphi|\!\}_n \cup \{\!|\psi|\!\}_n$$

(straying from the intuitionistic requirement specifying which disjunct holds). The self-dual temporal connective Next increments time

$$\begin{aligned} \{\!|\mathrm{N}{\scriptstyle\mathrm{EXT}}\,\varphi|\!\}_n &= \{\!|\varphi|\!\}_{n+1} \\ s, n \models \mathrm{N}{\scriptstyle\mathrm{EXT}}\,\varphi \quad &\text{iff} \quad s, n+1 \models \varphi \end{aligned}$$

while PREVIOUS decrements it

$$\llbracket \text{PREVIOUS } \varphi \rrbracket_n \quad = \quad \llbracket \varphi \rrbracket_{n-1}$$

$$s, n \models \text{PREVIOUS } \varphi \quad \text{iff} \quad s, n-1 \models \varphi \ .$$

As for UNTIL, we have

$$s, n \models \varphi \text{ UNTIL } \psi \ \text{ iff } \ (\exists m \geq n)\ (s, m \models \psi \text{ and for } n \leq k < m,\ s, k \models \varphi)\ .$$

Note that

$$\prod_{n \leq k < m} \llbracket \varphi \rrbracket_k = \{ f \mid f \text{ is a function with domain } \{k \mid n \leq k < m\} \text{ s.t. } (\forall k \in \mathsf{domain}(f))\ f(k) \in \llbracket \varphi \rrbracket_k \}$$

which we pair with $\llbracket \psi \rrbracket_m$ to set

$$\llbracket \varphi \text{ UNTIL } \psi \rrbracket_n \ = \ \left\{ \langle d,\ f \rangle \mid (\exists m \geq n)\ d \in \llbracket \psi \rrbracket_m \text{ and } f \in \prod_{n \leq k < m} \llbracket \varphi \rrbracket_k \right\}$$

$$= \bigcup_{m \geq n} \left(\llbracket \psi \rrbracket_m \times \prod_{n \leq k < m} \llbracket \varphi \rrbracket_k \right) \ .$$

To step from UNTIL to its de Morgan dual RELEASE,

$$s, n \models \varphi \text{ RELEASE } \psi \ \text{ iff } \ (\forall m \geq n)\ (s, m \models \psi \text{ or } (\exists k \geq n)(k < m \text{ and } s, k \models \varphi))$$

we replace $\bigcup$ by $\prod$, $\times$ by $\cup$ and $\prod$ by $\bigcup$.

$$\llbracket \varphi \text{ RELEASE } \psi \rrbracket_n \ = \prod_{m \geq n} \left(\llbracket \psi \rrbracket_m \cup \bigcup_{n \leq k < m} \llbracket \varphi \rrbracket_k \right)$$

$$= \left\{ f \mid f \text{ is a function with domain } \{m \mid m \geq n\} \text{ s.t. } (\forall m \geq n)\ f(m) \in \llbracket \psi \rrbracket_m \text{ or } f(m) \in \bigcup_{n \leq k < m} \llbracket \varphi \rrbracket_k \right\}$$

Similarly for SINCE and its dual INITIATE, turning $<$ around

$$\llbracket \varphi \text{ SINCE } \psi \rrbracket_n \ = \bigcup_{m \leq n} \left(\llbracket \psi \rrbracket_m \times \prod_{m < k \leq n} \llbracket \varphi \rrbracket_k \right)$$

$$\llbracket \varphi \text{ INITIATE } \psi \rrbracket_n \ = \prod_{m \leq n} \left(\llbracket \psi \rrbracket_m \cup \bigcup_{m < k \leq n} \llbracket \varphi \rrbracket_k \right)$$

where

$$s, n \models \varphi \text{ SINCE } \psi \quad \text{iff} \quad (\exists m \leq n)\ (s, m \models \psi \text{ and for } m < k \leq n,\ s, k \models \varphi)$$

$$s, n \models \varphi \text{ INITIATE } \psi \quad \text{iff} \quad (\forall m \leq n)\ (s, m \models \psi \text{ or } (\exists k \leq n)(m < k \text{ and } s, k \models \varphi))$$

Finally, there is no way $\bot$ can be true

$$\langle\!\langle \bot \rangle\!\rangle_n \ = \ \emptyset$$

but always one for $\top$

$$\langle\!\langle \top \rangle\!\rangle_n \ = \ \{\emptyset\}$$

($\emptyset$ being an element of $(\prod x \in \emptyset) I$, for any set I). We now have the ingredients for an argument by induction on φ to establish Theorem 2.

4 Time Ordering Events

The previous section equated time with the set $\mathbb{Z}$ of integers for the purpose of interpreting temporal propositions from LTL. It is a trivial matter, however, to interpret LTL against the real line $\mathbb{R}$, decorating the unary operators NEXT and PREVIOUS with real numbers $\epsilon > 0$ and defining

$$\langle\!\langle \text{NEXT}_\epsilon \varphi \rangle\!\rangle_n = \langle\!\langle \varphi \rangle\!\rangle_{n+\epsilon}$$

for

$$s, n \models \text{NEXT}_\epsilon \varphi \quad \text{iff} \quad s, n + \epsilon \models \varphi$$

and

$$\langle\!\langle \text{PREVIOUS}_\epsilon \varphi \rangle\!\rangle_n = \langle\!\langle \varphi \rangle\!\rangle_{n-\epsilon}$$

for

$$s, n \models \text{PREVIOUS}_\epsilon \varphi \quad \text{iff} \quad s, n - \epsilon \models \varphi \ .$$

Theorem 2 carries over, with NEXT becoming NEXT_1 and PREVIOUS PREVIOUS_1. But the question for natural language applications is then how to choose ϵ? There is an embarrassing wealth of choices in $\mathbb{R}$ that far outstrips the bounded precision of natural language descriptions. The problem is that arbitrarily small increments in $\mathbb{R}$ lead to complications with vagueness that plague any attempt to pin down the exact moment of change in $\mathbb{R}$ [14]. Consider (d) and (e).

(d) Pat left the room at 3:05 pm.
(e) Pat left the room not a picosecond before 3:05 pm.

If, given (d), (e) is not patently ridiculous, then certainly some variant of (e) is (with a picosecond replaced by a sufficiently small fraction thereof). Departures and other events have bounded temporal granularity, making it

Table 5 Some temporal relations on E from $s \subseteq T \times E$ and $<$ on T

Given	Let
$s \subseteq T \times E$	$e\ ov(s)\ e'$ iff $(\exists t)\ s(t,e)$ and $s(t,e')$
	$e\ in(s)\ e'$ iff $(\forall t$ s.t. $s(t,e))\ s(t,e')$
and $<$ on T	$e <_s e'$ iff $(\forall t, t'$ s.t. $s(t,e)$ and $s(t',e'))\ t < t'$

absurd to draw against $\mathbb{R}$ the sharp yes-no distinctions demanded by a bivalent semantics.[4]

Rather than plotting the events once and for all against the real line, we will tailor our notion of time to suit a set E of relevant events so that temporal granularity can be refined by expanding E, or coarsened by restricting E. Towards this end, we modify the components of a schedule $s \subseteq \mathbb{Z} \times P$ from Section 3, replacing not only P by E, but also $\mathbb{Z}$ by an arbitrary set T of times, subject to the construal

$$s(t,e) \text{ says `} e\ s\text{-occurs at } t\text{'}.$$

But what set T of times do we choose? The precise choice does not matter, as the idea is that, given any T whatsoever, a time $t \in T$ reduces to the set

$$\{e \in E \mid s(t,e)\}$$

of events that s-occur at t. There are problems in fleshing this simple picture out, but as we shall see, these problems can be overcome largely by fixing s up with a linear order on T.

For orientation, let us recall the Russell-Wiener construction of time [18] which makes do, at the outset, with no explicit notion of time. Instead, binary relations $\bigcirc$ and $\prec$ on a set E are assumed to satisfy conditions

$$\begin{array}{ll} (A_1) & e \bigcirc e \quad \text{(i.e. } \bigcirc \text{ is reflexive)} \\ (A_2) & e \bigcirc e' \text{ implies } e' \bigcirc e \\ (A_3) & e \prec e' \text{ implies not } e \bigcirc e' \\ (A_4) & e \prec e' \bigcirc e'' \prec e''' \text{ implies } e \prec e''' \\ (A_5) & e \prec e' \text{ or } e \bigcirc e' \text{ or } e' \prec e \end{array}$$

that allow us to regard $\bigcirc$ as temporal overlap, and $\prec$ as complete temporal precedence. More precisely, for any schedule $s \subseteq T \times E$ and linear order $<$ on T, let $ov(s)$ and $<_s$ be the binary relations given in Table 5

$$e\ ov(s)\ e' \text{ iff } e \text{ and } e'\ s\text{-occur at some time in common}$$

$$e <_s e' \text{ iff } e\ s\text{-occurs only before } e' \text{ (relative to } <).$$

Then conditions (A_2), (A_3) and (A_4) are met by interpreting $\bigcirc$ as $ov(s)$ and $\prec$ as $<_s$. Furthermore, under this interpretation, (A_1) holds provided every E-event s-occurs at some time

$$(\forall e \in E)(\exists t \in T)\ s(t,e) \qquad (\dagger)$$

[4]The vagueness of temporal propositions at issue here is the subject of [14], part of which we now recount (up to Theorem 3 below).

Table 6 The Russell-Wiener-Kamp construction of time from $\bigcirc$ and $\prec$

Given	Let
$\bigcirc \subseteq E \times E$	$E_{\bigcirc} = \{t \subseteq E \mid (\forall e, e' \in t)\ e \bigcirc e'\}$
	$T_{\bigcirc} = \subseteq\text{-max}(E_{\bigcirc})$
	$s_{\bigcirc} = \{(t, e) \in T_{\bigcirc} \times E \mid e \in t\}$
and $\prec$ on E	$t \prec_{\bigcirc} t'$ iff $(\exists e \in t)(\exists e' \in t')\ e \prec e'$

and for (A_5), it suffices that the times at which an event s-occurs form a $<$-interval

$$\text{for every event } e \in E \text{ and time } t \in T,$$
$$s(t, e) \text{ whenever } (\exists t_1 < t)(\exists t_2 > t)\ s(t_1, e) \text{ and } s(t_2, e)\ . \qquad (\ddagger)$$

Let us call a quadruple $\langle s, T, E, <\rangle$ made of a relation $s \subseteq T \times E$ and a linear order $<$ on T an *interval schedule* if conditions (†) and (‡) hold. Having extracted relations $ov(s)$ and $<_s$ meeting (A_1)–(A_5) from an interval schedule $\langle s, T, E, <\rangle$, we may ask can we go the opposite direction: given relations $\bigcirc$ and $\prec$ satisfying (A_1)–(A_5), can we construct an interval schedule $\langle s, T, E, <\rangle$ such that $\bigcirc$ and $\prec$ are $ov(s)$ and $<_s$ respectively?

Russell-Wiener-Kamp says yes, forming a set $T_{\bigcirc}$ of temporal instants and a schedule $s_{\bigcirc}$ simply from temporal overlap $\bigcirc$.[5] Essentially, $T_{\bigcirc}$ is defined so that $s_{\bigcirc}$ becomes the converse of membership $\in$

$$s_{\bigcirc} = \left\{\langle t, e\rangle \in T_{\bigcirc} \times E \mid e \in t\right\}$$

and a moment $t \in T_{\bigcirc}$ becomes the set of events in E that $s_{\bigcirc}$-occur at t

$$t = \left\{e \in E \mid s_{\bigcirc}(t, e)\right\}\ . \qquad (13)$$

To eliminate the mention of t on the right side of (13), we require that all events in t $\bigcirc$-overlap

$$(\forall e, e \in t)\ e \bigcirc e'\ .$$

In other words, an instant $t \in T_{\bigcirc}$ must be an element of the set $E_{\bigcirc}$ defined in Table 6. We pick out the $\subseteq$-maximal elements of $E_{\bigcirc}$ for $T_{\bigcirc}$

$$T_{\bigcirc} = \left\{t \in E_{\bigcirc} \mid \left(\forall t' \in E_{\bigcirc}\right)\ t \subseteq t' \text{ implies } t = t'\right\}$$

[5] The importance of $\bigcirc$ in $T_{\bigcirc}$ and $s_{\bigcirc}$ is the reason I have opted above for (A_1)–(A_5), instead of the more economical presentation (B_1)–(B_2) from [35], with $\bigcirc$ relegated to an abbreviation for the complement of the union $\prec \cup \succ$ of $\prec$ and its converse $\succ$.

(B_1) not $e \prec e$

(B_2) if ($e_1 \prec e_2$ and $e'_1 \prec e'_2$) then ($e_1 \prec e'_2$ or $e'_1 \prec e_2$)

(B_1) says $\prec$ is irreflexive, and (B_2) $\prec$ is an *interval ordering*.

in order that an element t of $T_{\bigcirc}$ be the set of *all* events that $s_{\bigcirc}$-occur at t. An example is helpful. Let $\hat{E}$ be the set $\{e_1, e_2\}$ and $e_1 \hat{\bigcirc} e_2$, so that

$$\hat{E}_{\hat{\bigcirc}} = 2^{\hat{E}}$$

$$\hat{T}_{\hat{\bigcirc}} = \{\{e_1, e_2\}\} .$$

For a non-empty linear order on times, let us add an event e_0 to $\hat{E}$ to form

$$\tilde{E} = \hat{E} \cup \{e_0\}$$

and stipulate that e_0 overlaps with e_1 but completely precedes e_2

$$\tilde{\bigcirc} = \hat{\bigcirc} \cup \{(e_0, e_0),\ (e_0, e_1),\ (e_1, e_0)\}$$

$$\tilde{\prec} = \{(e_0, e_2)\} .$$

Then

$$\tilde{E}_{\tilde{\bigcirc}} = \left\{ t \subseteq \tilde{E} \mid \text{not } \{e_0, e_2\} \subseteq t \right\}$$

with $\subseteq$-maximal elements $\{e_0, e_1\}$ and $\{e_1, e_2\}$, which we temporally order by existential quantification to get

$$\{e_0, e_1\} \tilde{\prec}_{\tilde{\bigcirc}} \{e_1, e_2\} \ \text{ because } e_0 \tilde{\prec} e_2 .$$

In general, given $\prec$ on E, we define

$$t \prec_{\bigcirc} t' \ \text{ iff } \ (\exists e \in t)(\exists e' \in t')\ e \prec e'$$

for all $t, t' \in T_{\bigcirc}$. So much for the definitions in Table 6. The punch line is that if $\bigcirc$ and $\prec$ satisfy (A_1) to (A_5), then $\langle s_{\bigcirc}, T_{\bigcirc}, E, \prec_{\bigcirc} \rangle$ is an interval schedule, and the relations $\bigcirc$ and $\prec$ are the same as the relations $ov(s)$ and $<_s$ (respectively) induced by $s = s_{\bigcirc}$ and $< = \prec_{\bigcirc}$.

But now, note that apart from $ov(s)$ and $<_s$, there are many more temporal relations between events that can be defined from a schedule s. An example that will be useful below is the relation $in(s)$

$$e\ in(s)\ e' \ \text{ iff } \ (\forall t)\ s(t, e) \text{ implies } s(t, e')$$

$$\text{(i.e., } e\ s\text{-occurs only during } e') .$$

If $\hat{E} = \{e_1, e_2\}$ and $e_1 \hat{\bigcirc} e_2$, then $T_{\hat{\bigcirc}}$ consists of just one instant, $\hat{E}$, so that for $s = s_{\hat{\bigcirc}}$, $e_1\ in(s)\ e_2$ even though we can devise a schedule s' on $\hat{E}$ such that $e_1\ ov(s')\ e_2$ but not $e_1\ in(s')\ e_2$. Evidently, the construction of $T_{\bigcirc}$ is too coarse to capture temporal relations such as $in(s)$.

Table 7 From s and $<$ to $s^<$ incorporating *pre* and *post*

$s^< = s \cup \{\langle t, pre(e)\rangle \mid (\exists t' > t)\ s(t', e) \text{ and } (\forall t' \leq t) \text{ not } s(t, e)\}$
$\quad \cup \{\langle t, post(e)\rangle \mid (\exists t' < t)\ s(t', e) \text{ and } (\forall t' \geq t) \text{ not } s(t, e)\}$

Is this a defect, however, of the Russell-Wiener-Kamp construction itself or of the inputs we feed to the construction? Rather than inputting any triple $\langle E, \bigcirc, \prec\rangle$ satisfying (A_1)–(A_5), we might check that the input has enough structure to encode interesting temporal relations definable from a schedule and linear order. Precisely what we mean by "interesting" will become clear shortly, but for a start, we might beef up the input to an interval schedule $\langle s, T, E, <\rangle$. If we are to construct temporal instants from $\subseteq$-max$(E_\bigcirc)$, we should make sure that an instant $t \in T$ is not discarded simply because an event e does not s-occur at t. That is, we might turn *not e* into an event.[6] In fact, if we are to satisfy the assumption (A_5) above, two negations might be required, one $<$-before e, and one $<$-after e. Let us call these $pre(e)$ and $post(e)$ respectively, and extend s to the schedule $s^<$ defined in Table 7 so that

$$s^<(t, pre(e)) \text{ iff } (\exists t' > t)\ s(t', e) \text{ and } (\forall t' \leq t) \text{ not } s(t, e)$$

$$s^<(t, post(e)) \text{ iff } (\exists t' < t)\ s(t', e) \text{ and } (\forall t' \geq t) \text{ not } s(t, e)\ .$$

The overlap $ov(s^<)$ induced by $s^<$ is a binary relation on the *pre*, *post*-extension

$$E_+ = E \cup \{pre(e) \mid e \in E\} \cup \{post(e) \mid e \in E\}$$

of E. We can capture the the precedence relation $<_s$ on E induced by s and $<$ through $ov(s^<)$ and *pre*, *post*

$$e <_s e' \text{ iff } e\ ov(s^<)\ pre(e') \text{ and not } e\ ov(s^<)\ e'$$

$$\text{iff } post(e)\ ov(s^<)\ e' \text{ and not } e\ ov(s^<)\ e'$$

for all $e, e' \in E$. The same goes for $in(s)$

$$e\ in(s)\ e' \text{ iff } e\ ov(s^<)\ e' \text{ and neither } e\ ov(s^<)\ pre(e') \text{ nor } e\ ov(s^<)\ post(e')$$

[6] In doing so, we subvert the dictum "no events no time" behind the identification of $T_\bigcirc$ with the $\subseteq$-maximal elements of $E_\bigcirc$. (Note that $\emptyset$ is not $\subseteq$-maximal in $E_\bigcirc$ unless $E = \emptyset$.)

and all 13 relations between intervals in [1].[7] A measure of the structure of s to preserve is given by the equivalence $\sim_s$ on T holding between times at which exactly the same events s-occur

$$t \sim_s t' \;\text{ iff }\; (\forall e \in E)\; s(t,e) \text{ iff } s(t',e)\ .$$

Injecting the linear order $<$ into s leads to

$$t \sim_{s^<} t' \;\text{ iff }\; (\forall y \in E_+)\; s^<(t,y) \text{ iff } s^<(t',y)\ .$$

Clearly, $\sim_{s^<}$ refines $\sim_s$, differentiating, for instance, the subsets

$$\{t \in T \mid \text{not } (\exists t' \leq t)\; t' \in \mathsf{domain}(s)\}$$

and

$$\{t \in T \mid \text{not } (\exists t' \geq t)\; t' \in \mathsf{domain}(s)\}$$

of $T - \mathsf{domain}(s)$, assuming these are non-empty.

Theorem 3 *For any interval schedule $\langle s, T, E, <\rangle$, the Russell-Wiener-Kamp set of times induced by the temporal overlap relation $ov(s^<)$ of the schedule $s^<$ defined from s and $<$ is the set*

$$T_{ov(s^<)} = \{\{y \in E_+ \mid s^<(t,y)\} \mid t \in T\}$$

reducing each time $t \in T$ to the set $\{y \in E_+ \mid s^<(t,y)\}$ of E_+-events that $s^<$-occur at t.

According to Theorem 3, a time in T is never discarded in $T_{ov(s^<)}$, as it might be were $s^<$ simply s, but merely lumped together with other $\sim_{s^<}$-equivalent times in T. The reduction $t \mapsto \{e \in E \mid s(t,e)\}$ is brought in line with Russell-Wiener-Kamp by stepping up from E to E_+ and s to $s^<$.

[7] An interval or more generally, a period in [2] can be reduced to a pair $\langle a, b\rangle$ of points. For $e = \langle a, b\rangle$ and $e' = \langle a', b'\rangle$,

$$\begin{aligned}
a < a' &\;\text{ iff }\; e\ ov(s^<)\ pre(e') \\
b < b' &\;\text{ iff }\; post(e)\ ov(s^<)\ e' \\
a = a' &\;\text{ iff }\; e\ ov(s^<)\ e' \text{ and neither } e\ ov(s^<)\ pre(e') \text{ nor } e'\ ov(s^<)\ pre(e) \\
b = b' &\;\text{ iff }\; e\ ov(s^<)\ e' \text{ and neither } e\ ov(s^<)\ post(e') \text{ nor } e'\ ov(s^<)\ post(e)
\end{aligned}$$

so that for instance,

$$\begin{aligned}
e \text{ finishes } e' &\;\text{ iff }\; a' < a \text{ and } b = b' \\
&\;\text{ iff }\; pre(e)\ ov(s^<)\ e' \text{ and } e\ ov(s^<)\ e' \text{ and} \\
&\qquad \text{neither } e\ ov(s^<)\ post(e') \text{ nor } e'\ ov(s^<)\ post(e)\ .
\end{aligned}$$

Whatever improvements Theorem 3 makes using $s^{<}$ in place of s, it does nothing about a second limitation of Russell-Wiener-Kamp: namely, the requirement (A_5) on $\bigcirc, \prec$, amounting on interval schedules $\langle s, T, E, <\rangle$ to the interval condition

$$\text{for every event } e \in E \text{ and time } t \in T,$$
$$s(t,e) \text{ whenever } (\exists t_1 < t)(\exists t_2 > t)\ s(t_1,e) \text{ and } s(t_2,e)\ . \qquad (\ddagger)$$

Applying Russell-Wiener-Kamp to events that violate $(A_5)/(\ddagger)$ can produce unwanted results. Suppose, for example, that events e_0, e_1 and e_2 had temporal projections under $\hat{s}$ of

$$\{t \mid \hat{s}(t,e_0)\} = \{t \in \mathbb{R} \mid 0 < t < 2\}$$
$$\{t \mid \hat{s}(t,e_1)\} = \{t \in \mathbb{R} \mid 1 < t < 3\}$$
$$\{t \mid \hat{s}(t,e_2)\} = \{t \in \mathbb{R} \mid 0 < t < 1 \text{ or } 2 < t < 3\}\ .$$

The intersection of all three sets is empty, although any two of e_0, e_1 and e_2 $\hat{s}$-overlap. The obvious problem is that the temporal projection of e_2 is split between $\{t \in \mathbb{R} \mid 0 < t < 1\}$ and $\{t \in \mathbb{R} \mid 2 < t < 3\}$. But if b_1 were an event with temporal projection $\{t \in \mathbb{R} \mid 0 < t < 1\}$ and b_2 were an event with temporal projection $\{t \in \mathbb{R} \mid 2 < t < 3\}$, we can remove e_2 from the schedule $\hat{s}$ before applying Russell-Wiener-Kamp, as we can count on b_1 and b_2 to cover for e_2. To generalize the idea, recall the relation $in(s)$ induced by a schedule $s \subseteq T \times E$

$$e\ in(s)\ e' \ \text{ iff } \ (\forall t)\ s(t,e) \text{ implies } s(t,e')$$

and observe that

$$e\ ov(s)\ e' \ \text{ iff } \ (\exists b\ in(s)\ e)(\exists b'\ in(s)\ e')\ b\ ov(s)\ b' \qquad (14)$$
$$e <_s e' \ \text{ iff } \ (\forall b\ in(s)\ e)(\forall b'\ in(s)\ e')\ b <_s b'\ . \qquad (15)$$

Now, the point is to apply Russell-Wiener-Kamp to the restriction of s to a subset B of E that can serve as a temporal basis for E, as spelled out in (I), and is well-behaved, in the sense of (II).

(I) Equivalences (14) and (15) hold with quantification in the right hand side relativized to B

$$e\ ov(s)\ e' \ \text{ iff } \ (\exists b, b' \in B)\ b\ in(s)\ e \text{ and } b'\ in(b)\ e' \text{ and } b\ ov(s)\ b'$$
$$e <_s e' \ \text{ iff } \ (\forall b, b' \in B \text{ s.t. } b\ in(s)\ e \text{ and } b'\ in(s)\ e')\ b <_s b'\ .$$

(II) The interval condition $(\ddagger)$ holds with B in place of E: for every event $e \in B$ and time $t \in T$,

$$s(t,e) \text{ whenever } (\exists t_1 < t)(\exists t_2 > t)\ s(t_1,e) \text{ and } s(t_2,e)\ .$$

Just as one may need to adjoin $pre(e)$ and $post(e)$ to E (extending s to $s^<$), one may need to break apart events in E into well-behaved pieces that we can put into B.[8] We must be careful that these pieces do not become too small, lest we defeat the purpose of making time just fine-grained enough to order events in E. In the next section, we get around the interval condition (‡) through strings and projections on strings that bring us back to Russell-Wiener-Kamp.

5 Strings and Inertia

Binary relations $s \subseteq T \times X$ between a set T of times and a set X are central to Sections 3 and 4, where they are called schedules. Section 4 explores possibilities for T other than the set $\mathbb{Z}$ of integers (assumed in Section 3) by replacing the atomic LTL propositions (constituting X in Section 3) with events. What do these second components (X) of schedules in Sections 3 and 4 have in common? They can both be described as temporal propositions, or following the custom in Artificial Intelligence since [23], *fluents*, provided we re-construct the schedules $s \subseteq T \times E$ in Section 4 from relations $s_\circ \subseteq T \times Occur(E)$ between T and the image $Occur(E)$ of some 1-1 function $e \mapsto occur(e)$ from events $e \in E$ to fluents $occur(e)$, putting

$$s(t,e) \ \text{ iff } \ s_\circ(t, occur(e))$$

which is to say,

$$e \ s\text{-occurs at } t \ \text{ iff } \ s_\circ, t \models occur(e) \ .$$

Section 4 proceeds directly to the relations $s \subseteq T \times E$, leaving out the map $e \mapsto occur(e)$. Is there a pay-off in recognizing $s_\circ \subseteq T \times Occur(E)$ beyond conceptualizing schedules in Sections 3 and 4 alike as notions of satisfaction between times and fluents?

Yes, or so the present section claims. Events are analyzed as ways fluents can be true at some time or other. To begin with, the expansion of $s \subseteq T \times E$ to $s^< \subseteq T \times E_+$ becomes a special case of applying connectives on fluents — in particular, PAST (for “sometime in the present or past”) and FUTURE (for “sometime in the present or future”), definable from the LTL-connectives $\top$, SINCE and UNTIL in Section 3

$$\text{PAST}(\varphi) = \top \text{ SINCE } \varphi$$

$$\text{FUTURE}(\varphi) = \top \text{ UNTIL } \varphi \ .$$

[8] Indeed, $pre(e)$ and $post(e)$ can be viewed as interval parts of the complement of e that are suited for inclusion in B.

Conjoining the negation $\overline{\varphi}$ of φ with $\textsc{Past}(\varphi)$ and $\textsc{Future}(\varphi)$, negated and unnegated, let

$$\begin{aligned}
pre_\circ(\varphi) &= \overline{\varphi} \wedge \overline{\textsc{Past}(\varphi)} \wedge \textsc{Future}(\varphi) \\
post_\circ(\varphi) &= \overline{\varphi} \wedge \textsc{Past}(\varphi) \wedge \overline{\textsc{Future}(\varphi)} \\
never(\varphi) &= \overline{\varphi} \wedge \overline{\textsc{Past}(\varphi)} \wedge \overline{\textsc{Future}(\varphi)} \\
hole(\varphi) &= \overline{\varphi} \wedge \textsc{Past}(\varphi) \wedge \textsc{Future}(\varphi) \ .
\end{aligned}$$

Given an interval schedule $\langle s, T, E, < \rangle$, an event $e \in E$ and a time $t \in T$, we have

$$s_\circ, t \models occur(e) \vee pre_\circ(occur(e)) \vee post_\circ(occur(e))$$

as $never(occur(e))$ and $hole(occur(e))$ are ruled out by the conditions

$$(\forall e \in E)(\exists t \in T)\ s(t, e) \tag{†}$$

for every event $e \in E$ and time $t \in T$,

$$s(t, e) \text{ whenever } (\exists t_1 < t)(\exists t_2 > t)\ s(t_1, e) \text{ and } s(t_2, e) \tag{‡}$$

on an interval schedule.[9] Fluents of the form $occur(e)$ aside, however, there is no reason to require of an arbitrary fluent that the set of times at which it is true be non-empty or an interval.

The question is do we not need (†) and (‡) to capture the notion of time underlying the Russell-Wiener-Kamp construction? To see that we do not, it is useful to form strings $\alpha_1\alpha_2\cdots\alpha_k \in (2^E)^*$ made up of symbols $\alpha_i \subseteq E$ to describe k successive times $t_1, t_2, \ldots, t_k$ such that for $1 \le i \le k$, each $e \in \alpha_i$ occurs at t_i. For an exact representation, we say the interval schedule $\langle s, E, \{t_1, \ldots, t_k\}, < \rangle$ with $t_1 < t_2 < \ldots < t_k$ is *representable as* $\alpha_1\alpha_2\cdots\alpha_n$ if $n = k$ and for $1 \le i \le k$,

$$\alpha_i = \{e \in E \mid s(t_i, e)\} \ .$$

For instance, the interval schedule $\langle \tilde{s}, \{e, e'\}, \{1, 2, \ldots, 7\}, < \rangle$ where e $\tilde{s}$-occurs precisely at 1–3, while e' $\tilde{s}$-occurs precisely at 2–6 (and $<$ is the usual order on $\{1, 2, \ldots, 7\}$) is representable as the string

$$\boxed{e}\boxed{e, e'}\boxed{e, e'}\boxed{e'}\boxed{e'}\boxed{e'}\square \tag{16}$$

[9]That is, an event $e \in E$ partitions T into at most three components,

$$\begin{aligned}
T_{pre(e)} &= \{t \in T \mid s^<(t, pre(e))\} \\
T_e &= \{t \in T \mid s(t, e)\} \\
T_{post(e)} &= \{t \in T \mid s^<(t, post(e))\}
\end{aligned}$$

which we will presently draw as a string

$$\boxed{pre(e)}\boxed{e}\boxed{post(e)} \in (2^{E_+})^*$$

of length 3, consisting of the symbols $\boxed{pre(e)}$ and $\boxed{e}$ and $\boxed{post(e)}$.

of length 7 (where boxes are used in place of curly braces to suggest a cartoon strip). Note that the Russell-Wiener-Kamp construction on the overlap relation $ov(\tilde{s}^<)$ described by Theorem 3 reduces (16) to the string

$$\boxed{e}\boxed{e, e'}\boxed{e'}\Box\ .$$

In general, the *block compression* $\mathsf{bc}(\mathbf{s})$ of a string $\mathbf{s}$ compresses a block α^n of $n \geq 2$ copies of α into one

$$\mathsf{bc}(\mathbf{s}) = \begin{cases} \mathbf{s} & \text{if length}(\mathbf{s}) \leq 1 \\ \mathsf{bc}(\alpha\mathbf{s}') & \text{if } \mathbf{s} = \alpha\alpha\mathbf{s}' \\ \alpha\,\mathsf{bc}(\alpha'\mathbf{s}') & \text{if } \mathbf{s} = \alpha\alpha'\mathbf{s}' \text{ where } \alpha \neq \alpha' \end{cases}$$

for all symbols α and α', in accordance with the slogan "no time without change" [19, p. 674]. Now, for any interval schedule $\langle s, T, E, < \rangle$, observe that if E is finite, then so is the the Russell-Wiener-Kamp set $T_{ov(s^<)}$ of times induced by $s^<$, as $T_{ov(s^<)} \subseteq 2^{E_+}$, for any schedule s, no matter how large T is. Consequently, it is not difficult to see that

Theorem 4 *Given an interval schedule $\langle s, T, E, < \rangle$, if E is finite, then the Russell-Wiener-Kamp construction on the overlap relation $ov(s^<)$ of $s^<$ is representable as a string $\mathbf{s} \in (2^E)^*$ such that $\mathsf{bc}(\mathbf{s}) = \mathbf{s}$.*

What if E were infinite? The idea is to replicate the Russell-Wiener-Kamp construction on $ov(s^<)$ by gluing together arbitrary finite approximations of s.[10] More precisely, for any finite subset X of E, we form the restriction

$$s \upharpoonright X = \{(t, e) \in s \mid e \in X\}$$

of s to X, or, in terms of strings, we project a string in $(2^E)^*$ to one in $(2^X)^*$ by componentwise intersection r_X with X

$$r_X(\alpha_1 \cdots \alpha_n) \;=\; (\alpha_1 \cap X) \cdots (\alpha_n \cap X)$$

[10] Resorting here to finite approximations is in line with the assertion by van Lambalgen and Hamm 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. [39, p. 12]

Also, at the conclusion of [35], S.K. Thomason anticipates [36] with the suggestion to devise

> a plausible explanation of how finite information-processors like us would come to *think of* time as a continuum ... For that problem, I think, the appropriate mathematical structure is not an event ordering (which models the world's events) but a category ... of finite event ordering (which models obervers' developing apprehensions of the worlds's events). [35, pp. 95–96]

I hope to provide a detailed comparison of the present work with that of [36] elsewhere, and explore the possibility that the various constructions of time in [36] (i.e., Russell-Wiener-Kamp's, Walker's and Thomason's own) come to the same thing in a suitable subcategory of event orderings (closed under "past" and "future" events).

so that for example, if $X = \{e'\}$,

$$r_X(\boxed{e}\boxed{e, e'}\boxed{e, e'}\boxed{e'}\boxed{e'}\boxed{e'}\Box) = \Box\boxed{e'}\boxed{e'}\boxed{e'}\boxed{e'}\boxed{e'}\Box\,.$$

Next, if the function $\mathsf{bc}_X : (2^E)^* \to (2^X)^*$ applies r_X before bc

$$\mathsf{bc}_X(\mathbf{s}) \;=\; \mathsf{bc}(r_X(\mathbf{s}))$$

then we need only construct inverse limits with respect to projections bc_X, for finite subsets X of E. An instructive example is provided by the real line $\langle \mathbb{R}, < \rangle$; for $X = \{-1, 2, 7\}$, the bc_X-approximation of the schedule $\{(r, r) \mid r \in \mathbb{R}\}$ is

$$\Box\boxed{-1}\Box\boxed{2}\Box\boxed{7}\Box.$$

In general, the bc_X-approximation of Russell-Wiener-Kamp on $s^<$ is the string obtained from Theorem 4 applied to the restriction $s \restriction X$ of s to X.

One of the advantages of block compression bc on strings over Russell-Wiener-Kamp on interval schedules is that we can dispense with the requirements (†) and (‡). Nor does the string $\mathbf{s}$ in Theorem 4 need to explicitly mention *pre*- or *post*-events, as the sequentiality of strings already builds in the order $<$ extending s to $s^<$. This point is clear from the fluents $pre_\circ(\varphi)$, $post_\circ(\varphi)$ above, and suggests reconstruing strings over the alphabet 2^E as strings over the alphabet 2^Φ, for some set Φ of fluents such as *occur*(e), generalizing from occurrences/tokens to types φ [11, Section 3]. But is block compression compatible with the LTL semantics of fluents (reviewed in Section 3 above)?

Consider, for instance, the string

$$\boxed{p}\boxed{q}\boxed{q} \tag{17}$$

which bc reduces to $\boxed{p}\boxed{q}$. Does not LTL differentiate (17) from $\boxed{p}\boxed{q}$ in that it expands out the former but not the latter to

$$\boxed{p}\boxed{q,\ \textsc{Previous}\ p}\boxed{q,\ \textsc{Previous}\ q,\ \textsc{Previous}(\textsc{Previous}\ p)} \tag{18}$$

(establishing the consistency of $q \wedge \textsc{Previous}(\textsc{Previous}\ p)$)? Not necessarily. To claim (18) follows from (17) is to presuppose a clock that interprets the fluents

$$\textsc{Previous}\ p,\ \textsc{Previous}\ q,\ \textsc{Previous}(\textsc{Previous}\ p),\ \ldots$$

However, what if, as is commonly assumed, any clock can be refined by another, leading to any number of operators $\textsc{Previous}_\epsilon$ and $\textsc{Previous}_{\epsilon'}$ with which to expand (17) out as

$$\boxed{p}\boxed{q,\ \textsc{Previous}_\epsilon\, p}\boxed{q,\ \textsc{Previous}_{\epsilon'} q,\ \textsc{Previous}_{\epsilon'}(\textsc{Previous}_\epsilon\, p)}\ ?$$

How then do we choose ϵ and ϵ'? If the problem of choosing ϵ and ϵ' does not arise and we can take the step from (17) to (18) for granted, then it is because we have already fixed a notion of time, and there is no need to appeal

Table 8 Some boundary conditions on strings $\alpha_1\alpha_2\cdots\alpha_k$ for $1 < i \leq k$

(a)	$e \in \alpha_i$	implies	$e \in \alpha_{i-1}$ or $pre(e) \in \alpha_{i-1}$
(b)	$e \in \alpha_{i-1}$	implies	$e \in \alpha_i$ or $post(e) \in \alpha_i$
(c)	$pre(e) \in \alpha_i$	implies	$pre(e) \in \alpha_{i-1}$
(d)	$post(e) \in \alpha_{i-1}$	implies	$post(e) \in \alpha_i$
(a′)	$\varphi \in \alpha_i$	implies	$\varphi \in \alpha_{i-1}$ or $\mathrm{f}\varphi \in \alpha_{i-1}$
(b′)	$\varphi \in \alpha_{i-1}$	implies	$\varphi \in \alpha_i$ or $\mathrm{f}\overline{\varphi} \in \alpha_{i-1}$
(c′)	$\mathrm{f}\varphi \in \alpha_{i-1}$	implies	$\varphi \in \alpha_i$ or $\mathrm{f}\overline{\varphi} \in \alpha_{i-1}$

to block compression (which, indeed, does nothing to (18)) or to Russell-Wiener-Kamp. We had better think twice about expanding (17) out as (18). In any case, we should be careful to spell out information implicit in a string before applying block compression (which, as is clear from the definition of bc_X, assumes some set X of events "of interest" has been picked out).

If a string $\alpha_1\alpha_2\cdots\alpha_k$ is to fully describe the relation $s^<$ induced by an interval schedule $\langle s, T, E, <\rangle$, then the interval condition (‡) on an event $e \in E$ leads to conditions (a), (b), (c) and (d) in Table 8 for $1 < i \leq k$. What happens when we step from events e to fluents φ and relax (‡)? An important idea from artificial intelligence (e.g. [23]) that has made inroads into linguistic semantics (e.g. [9, 33, 38]) is that many fluents are "inertial" in that they remain true (or false) over time unless a force is applied against (or for) them. To formalize this idea, we introduce non-inertial fluents $\mathrm{f}\varphi$ and $\mathrm{f}\overline{\varphi}$ indicating that a force is exerted to make φ true and (respectively) false. For inertial φ, we transform (a) and (b) into (a′) and (b′); (b′) says φ persists forward (to the next box) unless a force is exerted to falsify φ, while (a′) says φ persists backward (to the previous box) unless a force was previously exerted to make φ true. That is, φ persists forward and backward, in the absence of any force for or against φ. This is a simple formulation of the commonsense law of inertia. For (a′) and (b′) to differ from (a) and (b), $\mathrm{f}\varphi$ and $\mathrm{f}\overline{\varphi}$ must diverge from $pre(e)$ and $post(e)$. Instead of (c) and (d), we have the "succeed unless opposed" constraint (c′) asserting that $\mathrm{f}\varphi$ brings φ about at the next box unless a countervailing force $\mathrm{f}\overline{\varphi}$ is applied against φ.[11] Whence do such fluents $\mathrm{f}\varphi$ spring? A possible source is an action A that is reduced in STRIPS [15] to precondition, add and delete lists, inducing the constraint on strings $\alpha_1\alpha_2\cdots\alpha_k$ that for $1 \leq i \leq k$, if apply(A) belongs to α_i then so does every fluent in the set

$$\text{precondition-list}(A) \cup \{\mathrm{f}\varphi \mid \varphi \in \text{add-list}(A)\} \cup \{\mathrm{f}\overline{\varphi} \mid \varphi \in \text{delete-list}(A)\} .$$

An obvious approach to inference is to analyze both the denotations and indices for fluents as regular languages. Inclusion between regular languages is decidable, and regular languages enjoy extensive closure properties (e.g. [5, 13]).

[11] That processes may run in competition with each other (rather than in splendid isolation) distinguishes the present framework from Dynamic Logic [17]—not to mention the intermediate states that strings with length > 2 add to input/output pairs (interpreting Dynamic Logic).

6 Conclusion

Situations are constructed in Section 2 as pairs and functions making first-order formulae true in accordance with *formulae-as-types*, a 'dogma of logical form' associating pairs with conjunctions and existential quantification, and functions with implications and universal quantification. The logical forms are unpacked in Section 3 by reducing pairs and functions alike to schedules, on which a partial order $\sqsubseteq$ can be defined, linking situations-as-truthmakers (e.g. events) to situations-as-indices (suitably partialized, as advocated by situation-theorists). Moreover, as relations between time and fluents, schedules have temporal projections connected in Sections 4 and 5 to the Russell-Wiener-Kamp construction of time from events. Structuring those events within strings built from fluents allows us not only to sidestep two problematic assumptions ((†) and (‡)) in Russell-Wiener-Kamp, but also to formulate a notion of inertia regulating change over time (amenable to finite-state methods).

Acknowledgements I thank two referees for useful feedback, and Hans van Ditmarsch for his role in seeing this paper through.

References

1. Allen, J. F. (1983). Maintaining knowledge about temporal intervals. *Communications of the ACM, 26*(11), 832–843.
2. Allen, J. F., & Ferguson, G. (1994). Actions and events in interval temporal logic. *Journal of Logic and Computation, 4*(5), 531–579.
3. Asher, N., & Lascarides, A. (2003). *Logics of conversation*. Cambridge University Press.
4. Barwise, J., & Perry, J. (1983). *Situations and attitudes*. Cambridge: MIT Press.
5. Beesley, K. R., & Karttunen, L. (2003). *Finite state morphology*. Stanford: CSLI Publications.
6. Chang, C. C., & Keisler, H. J. (1973). *Model theory*. Amsterdam: North-Holland.
7. Cooper, R. (2005). Austinian truth, attitudes and type theory. *Research on Language and Computation, 3*(4), 333–362.
8. Davidson, D. (1967). The logical form of action sentences. In N. Rescher (Ed.), *The logic of decision and action* (pp. 81–95). University of Pittsburgh Press.
9. Dowty, D. R. (1986). The effects of aspectual class on the temporal structure of discourse: Semantics or pragmatics? *Linguistics and Philosophy, 9*, 37–61.
10. Emerson, E. A. (1992). Temporal and modal logic. In J. van Leeuwen (Ed.), *Handbook of theoretical computer science. Formal methods and semantics* (Vol. B, pp. 995–1072). MIT Press.
11. Fernando, T. (2007). Observing events and situations in time. *Linguistics and Philosophy, 30*(5), 527–550.
12. Fernando, T. (2009). Situations as indices and as denotations. *Linguistics and Philosophy, 32*(2), 185–206.
13. Fernando, T. (2009). Situations in LTL as strings. *Information and Computation, 207*(10), 980–999.
14. Fernando, T. (2010). Temporal propositions as vague predicates. In M. Aloni, & K. Schulz (Eds.), *Amsterdam colloquium 2009. LNAI* (Vol. 6042, pp. 143–152). Springer.
15. Fikes, R. E., & Nilsson, N. J. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence, 2*, 189–208.
16. Ginzburg, J. (2005). Abstraction and ontology: Questions as propositional abstracts in type theory with records. *Journal of Logic and Computation, 15*(2), 113–130.
17. Harel, D., Kozen, D., & Tiuryn, J. (2000). *Dynamic logic*. MIT Press.

18. Kamp, H. (1979). Events, instants and temporal reference. In R. Bäuerle, U. Egli, & A. von Stechow (Eds.), *Semantics from different points of view* (pp. 27–54). Berlin: Springer.
19. Kamp, H., & Reyle, U. (1993). *From discourse to logic*. Dordrecht: Kluwer.
20. Kaplan, D. (1978). On the logic of demonstratives. *Journal of Philosophical Logic, 8*(1), 81–98.
21. Kratzer, A. (2008). Situations in natural language semantics. In *Stanford encyclopedia of philosophy*. http://plato.stanford.edu/entries/situations-semantics/.
22. Martin-Löf, P. (1984). *Intuitionistic type theory*. Napoli: Bibliopolis. Notes by Giovanni Sambin of a series of lectures given in Padua, June 1980.
23. McCarthy, J., & Hayes, P. (1969). Some philosophical problems from the standpoint of artificial intelligence. In M. Meltzer, & D. Michie (Eds.), *Machine intelligence* (Vol. 4, pp. 463–502). Edinburgh University Press.
24. Mulligan, K., Simons, P., & Smith, B. (1984). Truth-makers. *Philosophy and Phenomenological Research, 44*, 287–321.
25. Nelson, D. (1949). Constructible falsity. *Journal of Symbolic Logic, 14*(1), 16–26.
26. Parsons, T. (1990). *Events in the semantics of English: A study in subatomic semantics*. Cambridge: MIT Press.
27. Prior, A. (1967). *Past, present and future*. Oxford: Clarendon Press.
28. Ranta, A. (1994). *Type-theoretical grammar*. Oxford: Oxford University Press.
29. Reichenbach, H. (1947). *Elements of symbolic logic*. London: Macmillan.
30. Schubert, L. (2000). The situations we talk about. In J. Minker (Ed.), *Logic-based artificial intelligence* (pp. 407–439). Dordrecht: Kluwer.
31. Smith, B. (1999). Truthmaker realism. *Australasian Journal of Philosophy, 77*(3), 274–291.
32. Soames, S. (1985). Lost innocence. *Linguistics and Philosophy, 8*(1), 59–71.
33. Steedman, M. (1997). *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, & A. ter Meulen (Eds.), *Handbook of logic and language* (pp. 895–935). Elsevier.
34. Sundholm, G. (1986). Proof theory and meaning. In D. Gabbay, & F. Guenthner (Eds.), *Handbook of philosophical logic* (Vol. 3, pp. 471–506). Dordrecht: Reidel.
35. Thomason, S. K. (1984). On constructing instants from events. *Journal of Philosophical Logic, 13*, 85–96.
36. Thomason, S. K. (1989). Free constructions of time from events. *Journal of Philosophical Logic, 18*, 43–67.
37. Troelstra, A. S. (1999). From constructivism to computer science. *Theoretical Computer Science, 211*, 233–252.
38. van Lambalgen, M., & Hamm, F. (2005). *The proper treatment of events*. Blackwell.
39. van Lambalgen, M., & Hamm, F. (2006). Additional material for *The proper treatment of events*. http://staff.science.uva.nl/~michiell/docs/ProofsTime.pdf. Accessed June 2006.