Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

ELSEVIER

Artificial Intelligence 153 (2004) 13–48

Artificial Intelligence

www.elsevier.com/locate/artint

A unifying semantics for time and events

Brandon Bennett [a,*,1], Antony P. Galton [b]

[a] *School of Computing, University of Leeds, Leeds, LS2 9JT, UK*
[b] *Department of Computer Science, University of Exeter, Exeter, EX4 4QF, UK*

Received 14 December 2001

Abstract

We give a formal semantics for a highly expressive language for representing temporal relationships and events. This language, which we call *Versatile Event Logic* (VEL), provides a general temporal ontology and semantics encompassing many other representations. The system incorporates a number of features that have not been widely employed in AI formalisms. It has the ability to describe alternative histories using a modal operator. It provides a semantics for individuals that explicitly models their identity through time and across alternative possible histories; and enables one to distinguish between necessary and extensional identity of individuals. In virtue of its treatment of individuals and count nouns, the formalism offers a solution to certain puzzles of identity, which arise when individuals are described in different ways. We propose that VEL can be used as a foundational interlingua for comparing and interfacing different AI languages and illustrate this by considering how Situation Calculus and Event Calculus can be represented within VEL.

Keywords: Temporal logic; Semantics

1. Introduction

Many researchers in the field of AI have recognised the need for formal representation languages capable of expressing high-level information in a naturalistic form—e.g., [11, 29]. Of central importance to such a representation is a precise analysis of the semantics of actions, events and temporal relations. Among the most influential of the large number

* Corresponding author.
E-mail addresses: brandon@comp.leeds.ac.uk (B. Bennett), A.P.Galton@exeter.ac.uk (A.P. Galton).
URL: http://www.comp.leeds.ac.uk/brandon(B. Bennett).
1 Supported by the EPSRC under grants GR/M56807 and GR/N13159.

0004-3702/$ – see front matter © 2003 Published by Elsevier B.V.
doi:10.1016/j.artint.2003.02.001

of formalisms proposed to deal with these are: the *Situation Calculus* [35], Allen's theory of action and time [2] and the *Event Calculus* [32,45]. Philosophers have also examined various logical aspects of events and we aim to take into account in particular the analyses given by Davidson [9] and Galton [16]. We present a unifying language that we call *Versatile Event Logic* (VEL), embodying a range of insights into the logic of time and events.

Representations of time within formal languages can be readily divided into three main approaches:

- Explicit reference to time points and a temporal ordering relation.
- Explicit reference to temporal intervals and the temporal relationships between them (which implicitly constrain the ordering of the end-points of the intervals).
- Use of propositional tenses to convey temporal relationships among facts (without explicit reference to any temporal entities).

Whereas most temporal logics incorporate only one of these ways of referring to temporal structure, in our formalism all three of them happily coexist. A single VEL formula can contain time-point and interval variables as well as tenses (though it turns out, unsurprisingly, that all of these constructs are in fact reducible to logically equivalent expressions involving just time points and the temporal ordering relation).

Our formalism incorporates each of the following analyses:

Events as transitions. A view of events that can be found in some of the earliest AI systems is that an event is a transition between states. This idea is the basis of STRIPS [15] and also the *Situation Calculus* [35]. The transition account of events can be handled very nicely within the general framework of modal logics, where a class of state transitions can be regarded as an accessibility relation. This idea has been used to provide the semantics for *dynamic logic* [26].

Events as occurrences over intervals. An approach widely adopted in AI is to correlate events with the time intervals over which they occur [2,18,20]. This is done by means of a quasi-logical predicate $\mathsf{Occurs}(\mathbf{e}, \delta)$, which says that an event of type **e** occurs during the interval δ. Temporal relationships between events can then be described in terms of relations between the intervals over which they occur; and in particular by the 13 interval relations of Allen [1].

Davidson's existential analysis. Davidson [9] gave an analysis of events according to which every action verb is associated with an implicit existentially quantified event-token variable. For example the sentence 'John saw Mary in London on Tuesday' would have the logical form

$$\exists\varepsilon[\mathsf{Saw}(\varepsilon, \mathsf{j}, \mathsf{m}) \wedge \mathsf{In}(\varepsilon, \mathsf{l}) \wedge \mathsf{On}(\varepsilon, \mathsf{t})].$$

This explains how a large number of modifying phrases may be attached to verbs and also accounts for a wide range of inferences concerning verbs and their modifiers. For instance, from the above example we can infer $\exists\varepsilon[\mathsf{Saw}(\varepsilon, \mathsf{j}, \mathsf{m})]$ by straightforward first-order reasoning.

Although Davidson's work is not very well known to AI researchers, the logical analysis it provides is very close to that employed by a wide variety of AI representations based on *reified* events. In particular, semantic networks [33] typically treat an event as a special kind of entity. Other entities are linked to the event by arcs describing their relationship to the event. The main difference between this approach and Davidson's analysis is that the fundamental rôles of agent and patient (and perhaps also indirect object) are separated out from the verb rather than treated as integral. Thus, many AI representations would exhibit a logical form such as

$$\exists\varepsilon[\mathsf{Saw}(\varepsilon) \wedge \mathsf{Agent}(\varepsilon, \mathsf{j}) \wedge \mathsf{Patient}(\varepsilon, \mathsf{m}) \wedge \mathsf{In}(\varepsilon, \mathsf{l}) \wedge \mathsf{On}(\varepsilon, \mathsf{t})].$$

The difference between this and Davidson's formulation does not seem to have any concrete semantic significance and in developing our semantics we have found it more intuitive to treat verbs as relation-like with a small fixed arity, as does Davidson. However, as will be explained below, we do not treat the event variable as a hidden argument of the verb but as bound to the verb by means of a kind of quantifier.

The ability to refer directly to individual event instances is undoubtedly useful for representing certain kinds of knowledge. For instance, it enables one to represent adverbs as predicates of events and allows the possibility of describing causality in terms of a binary relation between events.

Event radicals. Galton [16,17] analyses events in terms of *event radicals*, syntactic units which refer to event-types and combine with temporal *aspect* operators to form propositions. For example, if the event-type John-tie-his-shoelace were operated on by the *progressive* aspect, we would obtain the proposition *Prog*(John-tie-his-shoelace), meaning that an event of type 'John-tie-his-shoelace' is in progress. We would read this as 'John is tying his shoelace'.

We want to encompass the essential insights of all these different approaches within a single semantic framework. Our paper may be regarded as an attempt to produce a detailed *ontology* (as advocated by, e.g., Guarino [23]) of time and events accounting for their various linguistic manifestations. Whereas much recent work on ontology for AI systems has been carried out at the level of axioms (e.g., [34]), ours is founded at the semantic level. We give a rich intensional semantics, in terms of which all our temporal operators are interpreted. This analysis is in some respects similar to *Montague Semantics* [39,40], although we are concerned only with the various logical forms of temporal expression, not with their syntactic realisation in a natural language. We do give axioms governing the behaviour of our operators; but formulating a complete proof system is a subject for further research.

It must be emphasised that the VEL formalism is *not* intended to contribute directly to solving the problem of how to *reason* about actions and change. This problem involves many factors that are completely beyond the scope of this work. In particular we do not address the much studied *frame problem* or any of its sub-problems, which have been studied in considerable detail within the AI literature (see, e.g., [3,8,25,27,42,44]). VEL gives us a very general language for talking about time and events but is completely agnostic about the workings of phenomena such as causality and inertia that determine what sequences of events are possible or likely.[2] The question of whether practical reasoning about events requires non-monotonic inference (or some other non-classical form of entailment) is orthogonal to the issues addressed by VEL; indeed there is no reason why such inference mechanisms could not be built on top of VEL.

It must be further acknowledged that even in terms of expressive power there is little that can be said within VEL that could not be said by means of some other existing formalism. However, the contribution of VEL is to put together a variety of expressive devices within a single language, thus clarifying their logical interdependencies.

2. Temporal ontology

The semantics of our event logic must include a structure that determines relationships between different times. The simplest model treats time as a set of linearly ordered points. This set may be further constrained to be either discrete, dense or continuous and either bounded or unbounded in each direction (or to satisfy more esoteric conditions). If we want to be able to employ modal operators to distinguish between analytic and contingent propositions, we shall need a richer structure in which there are possible states of the world which do not occur at any point in the actual history of the world. We can think of the set of all possible histories of the world as forming a branching tree structure. A number of possible semantics and axiom sets based on such structures are discussed in [47]. Branching structures have also been proposed as a basis for AI reasoning systems [36,46] but this approach does not seem to have been widely adopted. However, in the theory of computation, branching time logics such as CTL* [13] have been extensively studied [14,37,38].

2.1. History structures

To model the temporal order of states of an evolving universe and the relationship between possible alternatives to the actual history of the world we employ a structure that we call a *History Structure*. This is a tuple $\mathcal{H} = \langle S, T, \prec, H \rangle$, where:

- S is a set $\{\mathfrak{s}_i\}$ of world states (called simply *states* for short),
- T is a set $\{\mathfrak{t}_i\}$ of time points,
- $\prec$ is an irreflexive linear order on T (which may be further constrained to be dense, continuous etc.),
- H is a set $\{\mathfrak{h}_i\}$ of histories, each of which is a function from T to S.

A pair $\langle \mathfrak{h}_i, \mathfrak{t}_i \rangle$ (where $\mathfrak{h}_i \in H$ and $\mathfrak{t}_i \in T$) is the *index* of a possible world in the tree, and $h_i(t_i)$ denotes the world state that holds at that index. Each state is associated with a set of *entities* N_s. An *interval* $[\mathfrak{t}_1 \ldots \mathfrak{t}_2]$ is the set $\{\mathfrak{t} \mid \mathfrak{t} \in T \wedge \mathfrak{t}_1 \preceq \mathfrak{t} \preceq \mathfrak{t}_2\}$ of time points.

2 It can be argued that any language which is sufficiently expressive to describe a variety of different possible models of causation must itself be to some degree neutral about the nature of causation.

Thus defined, history structures can take a wide variety of different forms but our commonsense view of time builds in an asymmetry between the past and the future: from the perspective of any point in time there is a single determinate past but many possible futures. To capture this intuition we specify a restricted class of history structures called *History Trees*, characterised by the following properties:

SconfP $(\forall \mathfrak{h}_1, \mathfrak{h}_2 \in H)[(\mathfrak{h}_1 = \mathfrak{h}_2) \vee (\exists \mathfrak{t} \in T)(\forall \mathfrak{t}' \in T)[(\mathfrak{h}_1(\mathfrak{t}') = \mathfrak{h}_2(\mathfrak{t}')) \leftrightarrow (\mathfrak{t}' \preceq \mathfrak{t})]]$.
Gclock $(\forall \mathfrak{h}_1, \mathfrak{h}_2 \in H)(\forall \mathfrak{t}_1, \mathfrak{t}_2 \in T)[\mathfrak{h}_1(\mathfrak{t}_1) = \mathfrak{h}_2(\mathfrak{t}_2) \rightarrow (\mathfrak{t}_1 = \mathfrak{t}_2)]$.
SconfP gives strict confluence in the past: every two distinct histories agree up to and including some time point and after that point they never agree.
Gclock tells us that we cannot have the same state at two distinct time points (there is a 'global clock').

The confluence property induces an equivalence relation on histories with respect to any given time point:

$$\mathfrak{h}_1 \overset{\mathfrak{t}}{\approx} \mathfrak{h}_2 \equiv_{def} (\forall \mathfrak{t}' \in T)\big[(\mathfrak{t}' \preceq \mathfrak{t}) \rightarrow \mathfrak{h}_1(\mathfrak{t}') = \mathfrak{h}_2(\mathfrak{t}')\big].$$

This means that the two histories agree on the states holding at all times up to and including $\mathfrak{t}$ (after which they may or may not diverge); it can be read as '$\mathfrak{h}_1$ and $\mathfrak{h}_2$ coincide up to $\mathfrak{t}$'. The **SconfP** property implies that any two histories coincide up to some time point.

This model of branching time has the following useful properties: (a) on each history the time structure is linear and can be described by a relatively simple logic; (b) reasoning about alternative histories can be handled by an $S5$ modality.

One might for certain purposes want to drop or weaken either or both of the conditions **SconfP** and **Gclock**. **SconfP** means that all possible states are accessible by travelling back in time and then forward again along some alternative possible history. If we omit **SconfP** we would allow other states that are not accessible in this way; and this would enable us to introduce a language with a more general modalities of logical possibility and necessity.[3] By omitting **Gclock** we allow the possibility that the same state may occur at different times. This is useful if we want to think of states as configurations which are independent of their place in the history structure. For instance, in [5] states are modelled in terms of distributions of physical matter in space.

2.2. States

A world state $\mathfrak{h}(\mathfrak{t}) \in S$ determines all those properties of the world at time $\mathfrak{t}$ in history $\mathfrak{h}$ which are invariant across all the histories coincident with $\mathfrak{h}$ up to $\mathfrak{t}$. We call these the properties which are *settled* at $\mathfrak{t}$ in $\mathfrak{h}$. If $\mathfrak{h}_1(\mathfrak{t}_1) = \mathfrak{h}_2(\mathfrak{t}_2)$, this does not necessarily mean that the same propositions are true at $\langle \mathfrak{h}_1, \mathfrak{t}_1 \rangle$ and $\langle \mathfrak{h}_2, \mathfrak{t}_2 \rangle$. This is because the truth of certain propositions depends upon the future, which depends on the actual history.

In the present paper, the nature of states is left unanalysed. They are just elements in the semantics. However, one might well want to associate a state with some further structure.

[3] In this setting we could not, as we do in VEL, define $\Box$ in terms of $\circledast$ and Holds-at; we would have to define it in terms of quantification over states.

For instance one could identify states with distributions of matter (as in [5]) or with sets of certain fundamental propositions.

It is worth emphasising that our notion of state corresponds to a total specification of all properties, whereas there is another usage of 'state' which refers to some limited part or aspect of the world. Thus one might want to talk, for instance, of the 'state' of John's being asleep. In the semantics of VEL this circumstance would hold in a set of world states, i.e., a subset of S. We call such a set a *partial state*. Moreover, the proposition 'John is asleep' would hold at a set $\{\ldots\langle\mathfrak{h}_i,\mathfrak{t}_i\rangle\ldots\}$ of index points corresponding to the set $\{\ldots\mathfrak{h}_i(\mathfrak{t}_i)\ldots\}$ of states.

2.3. Episodes

In specifying the semantics for events we shall need to refer to the interval in which an event occurs; however, an interval $[\mathfrak{t}_1\ldots\mathfrak{t}_2]$ alone is not enough to locate the event within the branches of the structure: we must also take into account the history on which it occurs. This leads us to consider a pair $\langle\mathfrak{h},[\mathfrak{t}_1\ldots\mathfrak{t}_2]\rangle$. We now observe that for any history $\mathfrak{h}'$ coincident with $\mathfrak{h}$ up to $\mathfrak{t}_2$, the same sequence of states, and hence the same event, occurs. Therefore we define the *episode* $[\mathfrak{t}_1\ldots\mathfrak{t}_2]^{\approx}_{\mathfrak{h}}$ (where $\mathfrak{t}_1\preceq\mathfrak{t}_2$) by:

$$[\mathfrak{t}_1\ldots\mathfrak{t}_2]^{\approx}_{\mathfrak{h}}=\left\{\langle\mathfrak{h}',[\mathfrak{t}_1\ldots\mathfrak{t}_2]\rangle\mid\mathfrak{h}'\in H\wedge\mathfrak{h}'\overset{\mathfrak{t}_2}{\approx}\mathfrak{h}\right\}.$$

$\mathbf{Ep}_{\mathcal{T}}$ denotes the set of all episodes on tree $\mathcal{T}$.

Episodes are not themselves sets of time points so we do not distinguish whether they are open or closed. However, the open/closed distinction becomes important when we consider the time points over which a proposition is true. A proposition might be true/false at all points within an episode, or at either of the end points of the episode, or any combination of these.[4]

2.4. Event-types

An *event-type* is a set of episodes. This may be specified in various ways; one important way of specifying event-types is in terms of partial states. For, example, given the partial state $S'\subseteq S$, the event-type

$$\begin{aligned}\Big\{[\mathfrak{t}\ldots\mathfrak{t}]^{\approx}_{\mathfrak{h}}\mid(\exists\mathfrak{t}_1,\mathfrak{t}_2)\Big[&(\mathfrak{t}_1\prec\mathfrak{t}\prec\mathfrak{t}_2)\wedge\\&(\forall\mathfrak{t}')[((\mathfrak{t}_1\prec\mathfrak{t}'\prec\mathfrak{t})\rightarrow\mathfrak{h}(\mathfrak{t}')\notin S')\wedge((\mathfrak{t}\prec\mathfrak{t}'\prec\mathfrak{t}_2)\rightarrow\mathfrak{h}(\mathfrak{t}')\in S')]\Big]\Big\}\end{aligned}$$

picks out *inceptions* of partial state S', i.e., transitions between an interval over which S' does not hold and an interval over which it does.

---

[4] Questions of whether propositions are true over open or closed intervals (and the associated *dividing instant* problem) are beyond the scope of the present paper. We believe that this depends on the particular proposition in question and also (as in cases such as the light bulb switching from off to on) on the particular and somewhat arbitrary stipulations that one must make to give precise truth conditions to natural language statements that are normally used rather imprecisely. A discussion of these issues can be found in [20], especially pp. 343–345.

2.5. Event-tokens

Event-tokens are occurrences of event-types. This being so it is tempting to regard event-types as predicates over a domain of event tokens. Indeed this is the essence of Davidson's account. However, a naïve development of this idea can easily lead to contradictions. These are most apparent when we try to formulate the conditions under which an event-token can be an occurrence of more than one event-type.

Consider the event-types:

- $\mathbf{l} =$ *John-stand-on-one-leg*,
- $\mathbf{a} =$ *John-eat-an-apple*,
- $\mathbf{la} =$ *John-stand-on-one-leg-while-eat-apple*.

On the one hand it seems that **l** and **a** are disjoint kinds of event, since a token of **l** might cause John's leg to become stiff, whereas a token of **a** could not,[5] so there can be no token that is a token of both these types. If the two event-types apply to the same sequence of world states then this just means that two different tokens occur simultaneously. However, on the other hand, one might wish to say that a token of type **la** is also a token of both type **l** and type **a**, since these are just less specific descriptions of a particular occurrence described by **la**.

We believe that these intuitions about the token/type relationship cannot be reconciled as long as we divorce event-tokens from their types. In our semantics an event-token is always associated with the event-type of which it is a token. To model this we identify an event token ε with a pair $\langle \eta, \mathbf{e} \rangle$, where $\mathbf{e}$ is an event-type and η a specific episode over which an event of that type occurs—i.e., we must have $\eta \in \mathbf{e}$. Thus an event token can regarded an episode seen from a certain perspective determined by an event-type.

Identity criteria for event-tokens are a very thorny philosophical issue and we do not pretend that our semantics gives a fully adequate account. However, we do believe that our theory is a reasonable simplification, which also allows one to make some useful comparisons between event-tokens.

If two tokens occur over the same episode they are merely contemporaneous and not necessarily connected in any other way. However, if $\mathbf{e}_1 \subset \mathbf{e}_2$, then this means that for every episode (in every possible history) over which an event of type $\mathbf{e}_1$ occurs, an event of type $\mathbf{e}_2$ also occurs. In other words, type $\mathbf{e}_2$ is a generalisation of type $\mathbf{e}_1$. Correspondingly, if we have $\varepsilon_1 = \langle \eta, \mathbf{e}_1 \rangle$ and $\varepsilon_2 = \langle \eta, \mathbf{e}_2 \rangle$, then, although these are distinct tokens, ε_2 can be seen as representing the same occurrence as ε_1 but from a more general perspective.

2.6. Entities and individuals

In standard first-order logic, names are taken as basic expressions denoting *individuals*, which are taken as satisfying a fixed set of predicates. However, in a temporal language in

[5] Clearly, identical event-tokens must participate in the same causal relationships. Davidson [10] proposed that, conversely, causal equivalence is a sufficient condition for the identity of event tokens. But our argument does not depend on this stronger claim.

which one can describe events and changes one must be able to refer to individuals which persist through time although their properties may change. In the context of our history tree model of time, we stipulate that an individual is *realised* in each world state (in which it exists) by an *entity*. These entities correspond to 'time slices' of individuals located within the branching history structure. Thus an individual is modelled by a function from states to entities. At those worlds where an individual does not exist it denotes the value ø. If at some state $\mathfrak{s}$ two individuals correspond to the same entity we say they are *co-realised* at $\mathfrak{s}$.

Formally, the time structure together with the individuals which inhabit it is modelled by a VEL *frame*. This is a structure: $\mathcal{F} = \langle \mathcal{T}, N, I \rangle$ where

- $\mathcal{T} = \langle S, T, \prec, H \rangle$ is a history tree.
- N is a set of *entities*,
- I is a set of individuals, each of which is a function from S to $N \cup \{ø\}$.

In this general semantics, no constraints are imposed on the set of entities or on the functions corresponding to individuals. However, one might well want to add further constraints which tie the theory to some particular model of reality. For instance, in [5] the 'entities' are taken as regular open subsets of $\mathbb{R}^3$, which correspond to possible extensions of individuals.

Note that we have chosen to identify individuals as functions from states to entities rather than using a slightly more general model in which individuals are identified with functions from $H \times T$ to S. Thus, at any given time $\mathfrak{t}$, each individual function will have the same value in all histories that coincide up to $\mathfrak{t}$. This seems to be a natural condition which corresponds to the way that we normally use proper names: the attribution of a name does not depend on any assumptions or predictions about the future of the individual in question.[6]

2.7. Verbs

In the current paper we take a rather coarse-grained view of the combination of verbs with their arguments. We shall model a verb as a function from tuples of individuals to event-types. To allow for possible indexicality within a verb expression we treat verbs as intensional, so that this function will also depend on the history and time at which the verb is evaluated.

[6] However, if we were to weaken the conditions **SconfP** and **Gclock**, so that the same state could occur at more than one point in a history structure, we might also want to modify the semantics of individuals to allow individual functions to have different values for different occurrences of the same state. Suppose, in some state $\mathfrak{s}$ we have two rigid objects a and b that are physically identical (but in different locations and thus having different properties relative to other objects). Then after a sequence of movements of the objects we may reach a state that is physically identical to $\mathfrak{s}$ except that the positions of a and b are interchanged. So if we take the final state as a recurrence of the initial state, we see that the properties of a and b (and hence the entities that their individual functions denote) are not determined only by the state.

2.8. Propositions

The propositional expressions of the VEL language have a truth value (**t** or **f**) that will in general vary according to the history and time at which the proposition is evaluated (and also according to the interpretation of its non-logical symbols). We could allow truth values to vary arbitrarily over the indices $\langle \mathfrak{h}, \mathfrak{t} \rangle$ of a history tree; however, in so far as propositions are used to describe properties of the real world, we believe that it is desirable to constrain their temporal variability to rule out some pathological models. Specifically, we impose the condition that it is not possible for a proposition to change its truth value infinitely often during a finite period of time.

Infinitely frequent oscillation of a proposition's truth value is sometimes called *intermingling* [12,19,21]. Prohibiting intermingling is equivalent to requiring that for every proposition φ and any time point $\mathfrak{t}$, there are time points $\mathfrak{t}_1$ and $\mathfrak{t}_2$, where $\mathfrak{t}_1 < \mathfrak{t} < \mathfrak{t}_2$, such that the truth value of φ is constant for all time points in the open interval $(\mathfrak{t}_1, \mathfrak{t})$ and is also constant (though not necessarily with the same truth value) over the open interval $(\mathfrak{t}, \mathfrak{t}_2)$. This latter formulation of the condition corresponds directly to the VEL axioms that we shall employ in Section 7 to enforce non-intermingling.

3. Informal overview of the VEL language

In Sections 4 and 5 we shall give a fully formal syntax and semantics for VEL but first we consider the language from an informal point of view.

It should be noted that in the formal semantics each expression will be given a denotation relative to an index point $\langle \mathfrak{h}, \mathfrak{t} \rangle$ within a history tree structure. At this point $\mathfrak{h}$ is the *actual history*, $\mathfrak{t}$ is the *actual time* and the value of $\mathfrak{h}(\mathfrak{t})$ is the *actual state*.

3.1. Variables and nominal terms

In VEL there are six types of object that can be named and quantified over. These types and the variables that are used to denote them are as follows:

- individuals—$a, b, c, \ldots,$
- times—t_i,
- intervals—δ_i,
- event-tokens—ε_i,
- states—s_i,
- observable values—o_i.

3.2. Logical functions and predicates

A small number of logical functions are used to transfer information between different semantic types:

- $\mathsf{b}(\delta)$ and $\mathsf{e}(\delta)$ return times which are respectively the beginning and end points of the interval δ.
- $\mathsf{state}(\tau)$ gives the world state that obtains at the time denoted by τ in the actual history.
- $\mathsf{dur}(\varepsilon)$ gives the interval of occurrence of the event token ε.

The following logical predicates are also employed:

- $\alpha = \beta$ is true just in case the terms α and β have the same extension.
- $\tau_1 \leqslant \tau_2$ is true just in case the time term τ_1 denotes a time equal to or earlier than τ_2.
- $\mathsf{AT}(\tau)$ is true just in case τ denotes the actual time.

The proposition $\mathsf{AT}(\tau)$ is true only at index points $\langle \mathfrak{h}, \mathfrak{t} \rangle$, where the index $\mathfrak{t}$ is the time point denoted by the object language time term τ. The AT predicate can be used to construct propositions whose truth at a given time point depends on other time points. For example, $\mathsf{AT}(t) \wedge (\exists t')[(t' \leqslant t) \wedge \varphi(t')]$ is true iff $\varphi(t')$ holds for some t' that is before or equal to the actual time.

3.3. Count nouns and quantifiers over individuals

A component of VEL that is probably unfamiliar to most AI researchers is the use of Gupta's [24] theory of count nouns. At each index $\langle \mathfrak{h}, \mathfrak{t} \rangle$, a count noun picks out a set of individuals. This means that the entities picked out by a count noun are doubly relative to the possible world (as identified by the index) at which a count noun expression is evaluated: the count noun is evaluated at some possible world to give a set of individuals which are instances of the count noun. Each individual must in turn be evaluated relative to a possible world to yield an entity existing at that world (see Section 2.6 above).

The reason for this double relativity is that we very often employ count nouns to pick out individuals at some particular world and then consider properties of these individuals at another world. For example, in the sentence "Some girl will become president of the USA", the phrase "some girl" quantifies over individuals who are girls at the present time. The individual concepts corresponding to these girls will denote different entities in different possible worlds but in many of these worlds those entities will not be girls. Thus the claim (at least in one reading) does not imply that the predicted female president will still be young when she takes office.

Quantification over individuals takes the form

$$(\forall \textsc{c}, x)[\varphi(x)],$$

where $\textsc{c}$ is a count noun. The count noun operates like a *sort* but it also fixes the trans-world identity conditions of quantified variables by determining the entities denoted by an individual variable at each index in the history tree.

We also define:

(D1) $(\exists \textsc{c}, x)\varphi(x) \equiv_{def} \neg(\forall \textsc{c}, x)\neg\varphi(x)$,
(D2) $\textsc{c}(\alpha) \equiv_{def} (\exists \textsc{c}, x)[x = \alpha]$,
(D3) $\textsc{c}[\alpha] \equiv_{def} (\exists \textsc{c}, x)[\Box(x = \alpha)]$.

C$[\alpha]$ means that α denotes one of the individuals that is (at the actual index) a C; whereas, C(α) means that α denotes an individual that is merely co-realised by some individual that is a C. Thus we might write:

$$(\exists \text{PRESIDENT}, a)[\text{WOMAN}[a]],$$

since presidents and women are both individuals of the same kind and share the same trans-world identity criteria; but

$$(\exists \text{QUANTITY-OF-WOOD}, a)[\text{SHIP}(a)],$$

since, although ships and quantities of wood may be materially identical (co-realised) in some world state, they do not have the same trans-world identity criteria (e.g., consider the ship of Theseus).

3.4. Propositional operators

VEL includes a variety of well-known temporal operators and modalities. **P** and **F** are past and future tense operators defined in the normal way. $\Box$ is just an $S5$ modality operator—$\Box\varphi$ is true if φ is true at every index point. The $⊛$ operator is a bit more subtle: $⊛\varphi$ is true at $\langle \mathfrak{h}, \mathfrak{t} \rangle$ if φ is true at $\mathfrak{t}$ in all histories which share the same initial segment as $\mathfrak{h}$ up to time $\mathfrak{t}$. Thus, $⊛\mathbf{F}\varphi$ means that φ is 'inevitable'. Dual operators $\Diamond$ and $⟐$ are defined in terms of $\Box$ and $⊛$ in the usual way. ($⟐$ can be used to model the E operator of CTL* [13], which has a similar interpretation.)

We also have operators $\lhd\varphi$ and $\rhd\varphi$, meaning respectively that φ 'has just been true' or 'is just about to become true'. $\rhd\varphi$ is rather like the standard 'next time' operator, except that it makes sense within a dense of continuous time series. It means that φ is true over some open set of time points, whose greatest lower bound is the actual time. $\lhd$ is defined analogously. Such operators were proposed by Barringer et al. [4].

3.5. Relations and verbs

Relations represent arbitrary intensional constraints over the universe of possible denotations of nominal terms. At each index point an n-ary relation is associated with the set of n-tuples drawn from this universe for which the relation holds. These tuples must respect the type of the relation (see Section 5.1 below). Verbs combine with one or more expressions denoting individuals to produce an expression that denotes an event-type. At each index a verb is associated with a mapping from n-tuples of individuals to event-types.

3.6. Syntactic roles of event-types

Expressions of the category of event-type can combine with other expressions in various different ways.

Event modalities. In the style of dynamic logics we employ event-types within a propositional operation $[\mathbf{e}]\varphi$, which says that φ holds in all states accessible from the current state *via* an occurrence of an event of type **e**. The dual operator $\langle\mathbf{e}\rangle\varphi$ is defined in the usual way.

The Occurs *relation.* The Occurs relation provides a flexible way of talking about types of events in terms of conditions which hold before and after (and perhaps also during) the time period over which an event of that type takes place. For example, following Galton [20], we could define the event-type where a body moves between two positions by:

$$\Box(\forall \text{BODY}, b)(\forall r_1, r_2)[\mathsf{Occurs}(\mathbf{move}(b, r_1, r_2), \delta) \leftrightarrow \\ \mathsf{Holds\text{-}at}((\mathsf{ext}(b) = r_1), \mathsf{begin}(\delta)) \wedge \mathsf{Holds\text{-}at}((\mathsf{ext}(b) = r_2), \mathsf{end}(\delta)) \wedge \\ \forall t[(\mathsf{b}(\delta) < t < \mathsf{e}(\delta)) \rightarrow \mathsf{Holds\text{-}at}((\mathsf{ext}(b) \neq r_1 \wedge \mathsf{ext}(b) \neq r_2), t)].$$

Here r_i are observable variables. ext is a function returning an observable value giving the region occupied by an individual.[7] The moving object b is constrained to be at r_1 at the beginning of the movement and r_2 at the end, and it cannot be at either of these locations at any other time within the movement. We do not suggest that this corresponds exactly to what we would in natural language call a 'movement', which is ambiguous in various ways.

Quantifying over singular events. A quantification over event-tokens is a formula of the form $\langle\forall \mathbf{e}, \varepsilon\rangle[\varphi(\varepsilon)]$. Allowing event-types to act as sortals in quantification over event tokens provides a mechanism for associating event-types with event token variables. Because the event-token variable carries with it an event-type as well as an episode the truth conditions of properties of event-tokens (and also the values of functions involving them) are not determined solely by the time period over which they occur. If we allowed unrestricted quantification over singular events, the language would become second-order. This is because we would be implicitly quantifying over the associated event-types, and event-types are interpreted as sets of episodes.[8]

Aspect operators. The operators *Perf*, *Pros* and *Prog* operate on event-type expressions to produce propositions. *Perf*($\mathbf{e}$), *Pros*($\mathbf{e}$) and *Prog*($\mathbf{e}$) mean respectively that event $\mathbf{e}$ has happened, will happen, and is happening.[9] If we want to say that an event is in progress without asserting that it will definitely be completed we can use the form $\diamondsuit$*Prog*($\mathbf{e}$). We can say that an event is just about to happen using the form $\neg$*Prog*($\mathbf{e}$) $\wedge \triangleright$*Prog*($\mathbf{e}$) (that an event has just been completed can be represented similarly but with $\triangleleft$).

Subsumption and identity of event-types. We will sometimes want to relate one event-type expression to another, either by saying that they are equivalent: $\mathbf{e}_1 \equiv \mathbf{e}_2$; or that one is a 'more specific' type of event than another: $\mathbf{e}_1 \sqsubseteq \mathbf{e}_2$. In terms of occurrence, these mean

[7] The semantics of spatial extensions and relationships between them will not be considered in this paper; however, Bennett [5] presents a language called $\mathcal{O}$, which is compatible with VEL and does have a semantics that explicitly models spatial extensions of moving material bodies.

[8] By defining very general types of event we can often express properties that might appear to require quantification over all event-tokens. We do this in our formulation of Event Calculus in Section 9.2, where we quantify over the type **any-event** defined in (D41).

[9] In the case of instantaneous event-types the construct *Prog* $\mathbf{e}$ doesn't really make sense (see [16] for a discussion of this). But according to the formal semantics given later in this paper it will always be false.

that for every interval over which an event of type $\mathbf{e}_1$ occurs, an event of type $\mathbf{e}_2$ also occurs, and in the case of $\equiv$ the converse also holds.

3.7. Event-type abstraction

The construct $\langle\delta:\varphi\rangle$ serves as a powerful means of defining event-types in terms of conditions on the time intervals over which they occur. Typically δ will occur as a free variable in φ, in which case we may write $\langle\delta:\varphi(\delta)\rangle$. When evaluated at an index $\langle\mathfrak{h},\mathfrak{t}\rangle$, the expression $\langle\delta:\varphi(\delta)\rangle$ refers to the event-type corresponding to the set of all episodes $[\mathfrak{t}_1\ldots\mathfrak{t}_2]^{\approx}_{\mathfrak{h}}$ such that $\varphi([\mathfrak{t}_1\ldots\mathfrak{t}_2])$ is true at index $\langle\mathfrak{h}',\mathfrak{t}\rangle$.[10] To illustrate the use of event abstraction, let us define[11] a verb $\mathbf{collide}(a,b)$ describing (punctual) collision events between two individuals:

$$\Box(\forall\ \text{BODY}, a, b)[\mathbf{collide}(a,b)\ \equiv$$
$$\langle\delta : ((\mathsf{b}(\delta) = \mathsf{e}(\delta) = t) \wedge \mathsf{Holds\text{-}at}(\mathsf{C}(a,b) \wedge \lhd\neg\mathsf{C}(a,b), t))\rangle]$$

Here C is the connection relation of the RCC spatial theory [41].

3.8. Holds-at and Holds-in

VEL includes the construct $\mathsf{Holds\text{-}at}(\varphi,\tau)$, which has been employed in many AI representations. It is true just in case φ is true (in the actual history) at the time point denoted by τ.

$\mathsf{Holds\text{-}in}(\varphi,\varsigma)$ asserts that φ is 'settled' as true at the state $\mathfrak{s}$ denoted by ς. This means that φ is true at every index $\langle\mathfrak{h},\mathfrak{t}\rangle$ such that $\mathfrak{h}(\mathfrak{t}) = \mathfrak{s}$.

4. Syntax

We now formally specify the syntax of VEL.

4.1. Variables and non-logical constants

The language has variables of six *nominal types*, $\sigma_i, \sigma_t, \sigma_\delta$, σ_ε, σ_s, σ_o, which can be quantified over. Any variable that is not bound by a quantifier behaves as a constant. The vocabulary sets of these types of variables are respectively:

- Individuals, $\mathcal{V}_i = \{a, b, c, \ldots\}$.
- Times, $\mathcal{V}_t = \{t_i\}$.
- Intervals, $\mathcal{V}_\delta = \{\delta_i\}$.

[10] Thus, in general, the abstracted event-type depends on the time $\mathfrak{t}$ of the point of evaluation; but in most useful cases $\varphi(\delta)$ will actually be invariant over time, unless φ contains some embedded indexical.

[11] In fact we give a universal identity axiom; but, if we take C as being primitive, we can regard the r.h.s. of the identity as a definition of $\mathbf{collide}(a,b)$.

- Event-Tokens, $\mathcal{V}_\varepsilon = \{\varepsilon_i\}$.
- States, $\mathcal{V}_s = \{s_i\}$.
- Observable Values, $\mathcal{V}_o = \{o_i\}$.

In many cases quantification over times, intervals, events and world states can be avoided by the use of tense and aspect operators, modal operators and various other constructs. For practical applications one might employ sublanguages of VEL that allow little or no quantification.

Four other types of non-logical constant are not quantified over:[12]

- Count nouns, $\mathcal{V}_c = \{\mathrm{C}_i\}$.
- Relations, $\mathcal{V}_R = \{A, B, C, \ldots\}$.
- Functions, $\mathcal{V}_f = \{f_i\}$.
- Verbs, $\mathcal{V}_v = \{\mathbf{v}_i\}$.

A VEL vocabulary is thus a 10-tuple $\mathcal{V} = \langle \mathcal{V}_i, \mathcal{V}_t, \mathcal{V}_\delta, \mathcal{V}_\varepsilon, \mathcal{V}_s, \mathcal{V}_o, \mathcal{V}_c, \mathcal{V}_R, \mathcal{V}_f, \mathcal{V}_v \rangle$. The sets $\mathcal{V}_R$, $\mathcal{V}_f$ and $\mathcal{V}_v$ are partitioned into sub-classes according to the numbers and types of arguments that these expressions can combine with.

It may seem that the diversity of syntactic types in VEL is excessive; but the idea of VEL is to be inclusive and to incorporate as many as possible of the kinds of expression that have been employed in temporal representations. VEL is not intended to provide a practical logic but rather a comprehensive umbrella language into which many other formalisms can be translated.[13]

4.2. Terms and functions

There are three types of term:

- variables (of the six nominal types),
- the symbol null (of type σ_i),
- functional terms of the form $f(\beta_1, \ldots, \beta_n)$, where $\beta_1, \ldots, \beta_n$ are terms and $f \in \mathcal{V}_f$.

Functions come in a variety of different types according to the nominal types they accept as arguments and the nominal type of the resulting functional term. A function type is described by a pair $\langle \sigma_0, \langle \sigma_1, \ldots, \sigma_n \rangle\rangle$ specifying the result and argument types respectively.

[12] In principle we could also allow quantification over these; but we would then immediately get a second order language which would be highly intractable and most probably unaxiomatisable.

[13] It might be argued that to obtain a still more comprehensive logic we should also have variables referring to histories. We have chosen not to do so here. While this does to some extent limit the expressive power of the language relative to its semantics, we do not believe that this is a serious shortcoming. Unrestricted history quantification does not seem to correspond to any structure found in normal temporal discourse. Moreover, a restricted (and arguably more natural) version of quantification over histories can be expressed in VEL by the $\circledast$ operator, which will be introduced below.

The unary logical functions b and e are both of type $\langle\sigma_t, \langle\sigma_\delta\rangle\rangle$, state is of type $\langle\sigma_s, \langle\sigma_t\rangle\rangle$ and dur of type $\langle\sigma_\delta, \langle\sigma_\varepsilon\rangle\rangle$).

4.3. Event-type expressions

An event-type, or $\sigma_{\mathbf{e}}$, expression is either:

- a 0-ary verb $\in \mathcal{V}_v$,
- an expression $\mathbf{v}(\alpha_1, \ldots, \alpha_n)$, where $\mathbf{v}$ is an n-ary verb $\in \mathcal{V}_v$ and the αs are of type σ_i,
- an *event abstraction* $\langle\delta : \varphi\rangle$, where δ is any variable of type σ_δ and φ is a proposition (as defined below).

4.4. Propositions

An atomic proposition has one of the following forms:

- $\alpha_1 = \alpha_2$, where the α's are terms of the same type,
- $\tau_1 \leqslant \tau_2$, where the τ's are terms of type σ_t,
- $\mathsf{AT}(\tau)$, where τ is a term of type σ_t,
- $R(\beta_1, \ldots, \beta_n)$, where $R \in \mathcal{V}_R$ and $\beta_1, \ldots, \beta_n$ are terms.

Each argument of a given relation R must be of a specific nominal type. The type of an n-ary relation can be described by a tuple $\langle\sigma_1, \ldots, \sigma_n\rangle$, where the σs are the types of its arguments (the result type always being a proposition).

More generally, a proposition of VEL is either an atomic proposition or has one of the forms:

- one of: $\neg\varphi$, $\varphi_1 \wedge \varphi_2$, $\mathbf{F}\varphi$, $\mathbf{P}\varphi$, $\Box\varphi$, $\boxast\varphi$, $\rhd\varphi$, $\lhd\varphi$,
- $\mathsf{Holds\text{-}at}(\varphi, \tau)$, where τ is of type σ_t,
- $\mathsf{Holds\text{-}in}(\varphi, \varsigma)$, where ς is of type σ_s,
- $\mathsf{Occurs}(\mathbf{e}, \delta)$, where $\mathbf{e}$ is of type $\sigma_{\mathbf{e}}$ and δ is of type σ_δ,
- $[\mathbf{e}]\varphi$, where $\mathbf{e}$ is of type $\sigma_{\mathbf{e}}$,
- $(\forall\chi)\varphi$, where χ is a variable of one of the types $\sigma_t, \sigma_\delta, \sigma_s, \sigma_o$,
- $(\forall\kappa, \upsilon)\varphi$, where κ is of type σ_c and υ is a variable of type σ_i,
- $\langle\forall\theta, \varepsilon\rangle\varphi$, where θ is of type $\sigma_{\mathbf{e}}$ and ε is a variable of type σ_ε,
- one of: $\mathit{Perf}\,\theta$, $\mathit{Prog}\,\theta$, $\mathit{Pros}\,\theta$, where θ is type $\sigma_{\mathbf{e}}$,

where in each case φ is any VEL proposition.

5. Formal semantics

We now give a fully formal semantics wherein each well formed VEL expression is evaluated relative to an index $\langle\mathfrak{h}, \mathfrak{t}\rangle$ on a VEL frame and an assignment $\mathcal{A}$, which determines the values of non-logical constants. We write $[\![\alpha]\!]^{\mathcal{A}}_{\mathfrak{h},\mathfrak{t}}$ to refer to the denotation of α at index $\langle \mathfrak{h}, \mathfrak{t} \rangle$ according to the assignment $\mathcal{A}$.

For a propositional expression φ the value of $[\![\varphi]\!]^{\mathcal{A}}_{\mathfrak{h},\mathfrak{t}}$ is either **t** or **f**. In this case it is often convenient to use the notation $[\![\varphi]\!]_{TS}$ to denote the set of assignments and indices for which φ is true—i.e., the 'truth set' of φ. Formally we have $[\![\varphi]\!]_{TS} = \{\langle \mathcal{A}, \mathfrak{h}, \mathfrak{t} \rangle \mid [\![\varphi]\!]^{\mathcal{A}}_{\mathfrak{h},\mathfrak{t}} = \mathbf{t}\}$.

To describe the semantics of expressions such as quantifiers, we will need to talk about certain classes of similar assignments. We use the notation $\mathcal{A}^{(x \to v)}$ to refer to that assignment $\mathcal{A}'$ which assigns the value v to the variable x and for all other variables assigns exactly the same values as $\mathcal{A}$.

5.1. VEL model structures

A VEL *model* is a structure $\mathcal{M} = \langle \mathcal{F}, O, \mathcal{V}, \mathcal{A} \rangle$, where:

- $\mathcal{F} = \langle \mathcal{T}, N, I \rangle$ is a VEL frame, with $\mathcal{T} = \langle S, T, \prec, H \rangle$,
- O is the set of possible values of observables,
- $\mathcal{V}$ is a vocabulary,
- $\mathcal{A}$ is an assignment structure.

The assignment structure $\mathcal{A}$ is used to interpret all the non-logical atomic symbols of VEL. Specifically, $\mathcal{A} = \langle a_i, a_t, a_\delta, a_\varepsilon, a_s, a_o, a_R, a_f, a_c, a_v \rangle$, where:

- $a_i : \mathcal{V}_i \to I$ assigns an individual to each individual variable.
- $a_t : \mathcal{V}_t \to T$ assigns a time point to each time variable.
- $a_\delta : \mathcal{V}_\delta \to (T \times T)$ assigns to each interval variable an interval $[\mathfrak{t}_1 \ldots \mathfrak{t}_2]$, such that $\mathfrak{t}_1 \preceq \mathfrak{t}_2$.
- $a_\varepsilon : \mathcal{V}_\varepsilon \to (\mathbf{Ep}_\mathcal{T} \times \wp(\mathbf{Ep}_\mathcal{T}))$, assigns to each event-token variable a pair $\langle \eta, E \rangle$, where η is episode and E is a set of episodes and $\eta \in E$.
- $a_s : \mathcal{V}_s \to S$ assigns a state to each state variable.
- $a_o : \mathcal{V}_o \to O$ assigns an observable value to each observable variable.
- $a_c : \mathcal{V}_c \to ((H \times T) \to \wp(I))$ assigns to each count noun a function from indices to sets of individuals.
- $a_v : \mathcal{V}_v \to ((H \times T \times I^n) \to \wp(\mathbf{Ep}_\mathcal{T}))$ assigns a denotation to each n-ary verb, this is a mapping from history and time indices and tuples of individuals to sets of episodes.

The assignment functions a_R and a_f are slightly harder to specify because of the variety of argument types. For each nominal type σ let $\mathbf{U}_\sigma$ be the range of a_σ. Then, for relations of type $\sigma = \langle \sigma_1, \ldots, \sigma_n \rangle$ we have:

- $a_{R_\sigma} : \mathcal{V}_R \to ((H \times T) \to \wp(\mathbf{U}_{\sigma_1} \times \cdots \times \mathbf{U}_{\sigma_n}))$ assigns to each relation symbol a mapping from indices to the set of tuples of objects (from domains appropriate to the type σ) that satisfy the relation;

and for functions of type $\sigma = \langle \sigma_0, \langle \sigma_1, \ldots, \sigma_n \rangle \rangle$

- $a_{f_\sigma} : \mathcal{V}_o \to ((H \times T) \to ((\mathbf{U}_{\sigma_1} \times \cdots \times \mathbf{U}_{\sigma_n}) \to \mathbf{U}_{\sigma_0}))$ assigns a suitable intensional function to each function symbol of type σ.

We can then define generic functions a_R and a_f which can be applied respectively to any relational or any functional symbol:

- $a_R = \bigcup_\sigma a_{f_\sigma}$,
- $a_f = \bigcup_\sigma a_{R_\sigma}$.

To ensure that the propositions constructed from the basic vocabulary are not affected by intermingling (see Section 2.8) we need to impose some further condition on VEL models. Specifically, we require that all those denotations which are functions of time (i.e., the denotations of individuals, count nouns, verbs, relations and functions) do not oscillate infinitely often within any finite interval. Also, the denotations of any pair of individuals cannot be such that they change to and from being co-realised infinitely often in a finite period; and the set of episodes in any event-type (or in the episode set associated with an event-token) cannot contain an infinite number of episodes that share a common history and either begin or end infinitely often within some finite interval. Because these constraints are somewhat awkward to specify formally, we instead express them in terms of the general condition that

- for any proposition φ, the value $[\![\varphi]\!]^{\mathcal{A}}_{\mathfrak{h},\mathfrak{t}}$ cannot change its value infinitely often within a finite time interval.

This implicitly imposes the required restrictions on all denotations of propositional constituents.

5.2. The semantic denotation function

We now specify the semantic denotation for all meaningful expressions of VEL using standard set theory and quantification as a meta-language. Variables $\mathfrak{t}_i$ range over elements of the set of times (T) and $\mathfrak{h}_i$ range over histories (elements of H). For any non-logical symbol κ, $\mathcal{A}(\kappa)$ is the result of applying to κ the appropriate function in $\mathcal{A}$, according to the type of κ.

5.2.1. Atomic symbols and nominal terms

(S1) $[\![\gamma]\!]^{\mathcal{A}}_{\mathfrak{h},\mathfrak{t}} = \mathcal{A}(\gamma)$, where γ is any atomic non-logical symbol.
(S2) $[\![\mathsf{null}]\!]^{\mathcal{A}}_{\mathfrak{h},\mathfrak{t}} = \emptyset$.
(S3) $[\![\mathsf{b}(\delta)]\!]^{\mathcal{A}}_{\mathfrak{h},\mathfrak{t}} = \mathfrak{t}_1$, where $[\![\delta]\!]^{\mathcal{A}}_{\mathfrak{h},\mathfrak{t}} = [\mathfrak{t}_1 \ldots \mathfrak{t}_2]$.
(S4) $[\![\mathsf{e}(\delta)]\!]^{\mathcal{A}}_{\mathfrak{h},\mathfrak{t}} = \mathfrak{t}_2$, where $[\![\delta]\!]^{\mathcal{A}}_{\mathfrak{h},\mathfrak{t}} = [\mathfrak{t}_1 \ldots \mathfrak{t}_2]$.
(S5) $[\![\mathsf{state}(\tau)]\!]^{\mathcal{A}}_{\mathfrak{h},\mathfrak{t}} = \mathfrak{h}([\![\tau]\!]^{\mathcal{A}}_{\mathfrak{h},\mathfrak{t}})$.
(S6) $[\![\mathsf{dur}(\varepsilon)]\!]^{\mathcal{A}}_{\mathfrak{h},\mathfrak{t}} = [\mathfrak{t}_1 \ldots \mathfrak{t}_2]$, where $[\![\varepsilon]\!]^{\mathcal{A}}_{\mathfrak{h},\mathfrak{t}} = \langle [\mathfrak{t}_1 \ldots \mathfrak{t}_2]^{\approx}_{\mathfrak{h}}, E\rangle$.
(S7) $[\![g(\beta_1, \ldots, \beta_n)]\!]^{\mathcal{A}}_{\mathfrak{h},\mathfrak{t}} = a_f(g)(\mathfrak{h}, \mathfrak{t}, \langle [\![\beta_1]\!]^{\mathcal{A}}_{\mathfrak{h},\mathfrak{t}}, \ldots, [\![\beta_n]\!]^{\mathcal{A}}_{\mathfrak{h},\mathfrak{t}}\rangle)$.

5.2.2. Event-type expressions

Complex event-types are constructed either by combining a verb with individual constants or terms or by the event abstraction operator:

(S8) $[\![\mathbf{v}(\alpha_1, \ldots, \alpha_n)]\!]^{\mathcal{A}}_{\mathfrak{h},\mathfrak{t}} = a_v(\mathbf{v})(\mathfrak{h}, \mathfrak{t}, \langle a_i(\alpha_1), \ldots, a_i(\alpha_n)\rangle)$.
(S9) $[\![\langle\delta:\varphi\rangle]\!]^{\mathcal{A}}_{\mathfrak{h},\mathfrak{t}} = \{[\mathfrak{t}_1 \ldots \mathfrak{t}_2]^{\approx}_{\mathfrak{h}} \mid \langle\mathcal{A}^{\delta\to[\mathfrak{t}_1\cdots\mathfrak{t}_2])}, \mathfrak{h}', \mathfrak{t}\rangle \in [\![\varphi]\!]_{TS})\}$.

5.2.3. Atomic propositional formulae

(S10) $[\![R(\beta_1, \ldots, \beta_n)]\!]_{TS} = \{\langle\mathcal{A}, \mathfrak{h}, \mathfrak{t}\rangle \mid \langle[\![\beta_1]\!]^{\mathcal{A}}_{\mathfrak{h},\mathfrak{t}}, \ldots, [\![\beta_n]\!]^{\mathcal{A}}_{\mathfrak{h},\mathfrak{t}}\rangle \in a_R(R)(\mathfrak{h}, \mathfrak{t})\}$.
(S11) $[\![\alpha_1 = \alpha_2]\!]_{TS} = \{\langle\mathcal{A}, \mathfrak{h}, \mathfrak{t}\rangle \mid [\![\alpha_1]\!]^{\mathcal{A}}_{\mathfrak{h},\mathfrak{t}}(\mathfrak{h}(\mathfrak{t})) = [\![\alpha_2]\!]^{\mathcal{A}}_{\mathfrak{h},\mathfrak{t}}(\mathfrak{h}(\mathfrak{t}))\}$.
(S12) $[\![\tau_1 \leqslant \tau_2]\!]_{TS} = \{\langle\mathcal{A}, \mathfrak{h}, \mathfrak{t}\rangle \mid [\![\tau_1]\!]^{\mathcal{A}}_{\mathfrak{h},\mathfrak{t}} \preceq [\![\tau_2]\!]^{\mathcal{A}}_{\mathfrak{h},\mathfrak{t}}\}$.
(S13) $[\![\mathsf{AT}(\tau)]\!]_{TS} = \{\langle\mathcal{A}, \mathfrak{h}, \mathfrak{t}\rangle \mid [\![\tau]\!]^{\mathcal{A}}_{\mathfrak{h},\mathfrak{t}} = \mathfrak{t}\}$.

5.2.4. Complex propositions

The Boolean connectives have standard interpretations:

(S14) $[\![\neg\varphi]\!]_{TS} = \{\langle\mathcal{A}, \mathfrak{h}, \mathfrak{t}\rangle \mid \langle\mathcal{A}, \mathfrak{h}, \mathfrak{t}\rangle \notin [\![\varphi]\!]_{TS}\}$.
(S15) $[\![\varphi_1 \wedge \varphi_2]\!]_{TS} = [\![\varphi_1]\!]_{TS} \cap [\![\varphi_2]\!]_{TS}$.

The constructs $\mathsf{Holds\text{-}at}(\varphi, \tau)$ and $\circledast\varphi$ are interpreted as follows:

(S16) $[\![\mathsf{Holds\text{-}at}(\varphi, \tau)]\!]_{TS} = \{\langle\mathcal{A}, \mathfrak{h}, \mathfrak{t}\rangle \mid \langle\mathcal{A}, \mathfrak{h}, [\![\tau]\!]^{\mathcal{A}}_{\mathfrak{h},\mathfrak{t}}\rangle \in [\![\varphi]\!]_{TS}\}$.
(S17) $[\![\circledast\varphi]\!]_{TS} = \{\langle\mathcal{A}, \mathfrak{h}, \mathfrak{t}\rangle \mid (\forall\mathfrak{h}')[(\mathfrak{h}' \overset{\mathfrak{t}}{\approx} \mathfrak{h}) \to \langle\mathcal{A}, \mathfrak{h}', \mathfrak{t}\rangle \in [\![\varphi]\!]_{TS}]\}$.

The following clauses specify the various kinds of quantification that can be expressed in VEL:

(S18) $[\![(\forall\mathrm{C}, x)[\varphi]]\!]_{TS} = \{\langle\mathcal{A}, \mathfrak{h}, \mathfrak{t}\rangle \mid (\forall\mathfrak{i})(\mathfrak{i} \in [\![\mathrm{C}]\!]^{\mathcal{A}}_{\mathfrak{h},\mathfrak{t}} \to [\![\varphi]\!]^{\mathcal{A}^{(x\to i)}}_{\mathfrak{h},\mathfrak{t}} = \mathbf{t})\}$.
(S19) $[\![\langle\forall\,\mathbf{e}, \varepsilon\rangle[\varphi]]\!]_{TS} = \{\langle\mathcal{A}, \mathfrak{h}, \mathfrak{t}\rangle \mid (\forall\mathfrak{t}_1, \mathfrak{t}_2, \eta)[([\mathfrak{t}_1 \ldots \mathfrak{t}_2]^{\approx}_{\mathfrak{h}} \in [\![\mathbf{e}]\!]^{\mathcal{A}}_{\mathfrak{h},\mathfrak{t}} \wedge \eta = \langle[\mathfrak{t}_1 \ldots \mathfrak{t}_2]^{\approx}_{\mathfrak{h}}, [\![\mathbf{e}]\!]^{\mathcal{A}}_{\mathfrak{h},\mathfrak{t}}\rangle) \to [\![\varphi]\!]^{\mathcal{A}^{(\varepsilon\to\eta)}}_{\mathfrak{h},\mathfrak{t}} = \mathbf{t}]\}$.
(S20) $[\![\forall\,\delta[\varphi]]\!]_{TS} = \{\langle\mathcal{A}, \mathfrak{h}, \mathfrak{t}\rangle \mid (\forall\mathfrak{t}_1, \mathfrak{t}_2)[(\mathfrak{t}_1 \preceq \mathfrak{t}_2) \to [\![\varphi]\!]^{\mathcal{A}^{(\delta\to[\mathfrak{t}_1\ldots\mathfrak{t}_2])}}_{\mathfrak{h},\mathfrak{t}} = \mathbf{t}]\}$.
(S21) $[\![\forall\,\chi_\sigma[\varphi]]\!]_{TS} = \{\langle\mathcal{A}, \mathfrak{h}, \mathfrak{t}\rangle \mid (\forall v \in \mathbf{U}_\chi)[[\![\varphi]\!]^{\mathcal{A}^{(\chi\to v)}}_{h,t} = \mathbf{t}]\}$, where σ is one of the types σ_t, σ_s or σ_o and χ_σ is a variable of type σ.

Quantification over individuals is always restricted to those satisfying a given count noun. Similarly event-token quantification is always over tokens of a given type. Quantification over the other types is unrestricted.

6. Definable constructs

The operators and constructs we defined semantically in the previous section constitute the logical primitives of VEL. We shall now use these to define a much richer vocabulary which provides a very expressive language for describing time and events, and in particular allows one to use all the modes of temporal representation that were considered in Section 1.

6.1. Temporal relations

The temporal relations $<$, $\geqslant$ and $>$ between time points have standard definitions in terms of the primitives $\leqslant$ and $=$. By using the functions $\mathsf{b}(\delta)$ and $\mathsf{e}(\delta)$, it is also trivial to define each of Allen's 13 relations that can hold between temporal intervals [1].

6.2. Propositional operators

Standard past and future tense operators are definable in terms of Holds-at and the AT predicate.[14]

(D4) $\mathbf{P}\varphi \equiv_{def} (\exists t_1, t_2)[\mathsf{AT}(t_1) \wedge (t_1 > t_2) \wedge \mathsf{Holds\text{-}at}(\varphi, t_2)]$.
(D5) $\mathbf{F}\varphi \equiv_{def} (\exists t_1, t_2)[\mathsf{AT}(t_1) \wedge (t_1 < t_2) \wedge \mathsf{Holds\text{-}at}(\varphi, t_2)]$.

We also define the following tense-like operators that allow us to say that a proposition is true throughout some open interval either immediately before or immediately after the current time point. These enable one to describe the ways in which truth values of propositions can change in dense or continuous time.

(D6) $\lhd\varphi \equiv_{def} (\exists t_1, t_2)[\mathsf{AT}(t_2) \wedge (t_1 < t_2) \wedge (\forall t_3)[(t_1 \leqslant t_3 < t_2) \rightarrow \mathsf{Holds\text{-}at}(\varphi, t_3)]]$.
(D7) $\rhd\varphi \equiv_{def} (\exists t_1, t_2)[\mathsf{AT}(t_1) \wedge (t_1 < t_2) \wedge (\forall t_3)[(t_1 < t_3 \leqslant t_2) \rightarrow \mathsf{Holds\text{-}at}(\varphi, t_3)]]$.

We now define the necessity operator $\Box\varphi$, which says that φ is true at all index points in the history tree model structure. The following definition is justified by the confluence property of history trees enforced by the **SconfP** condition.

(D8) $\Box\varphi \equiv_{def} (\forall t)[\ \mathsf{Holds}-\mathsf{at}(\ \circledast\forall t'[\mathsf{Holds\text{-}at}(\varphi, t')],\ t)]$.

The following definitions characterise Galton's [20,21] *perturbation* and *dominance* relations which can be used to describe possible state changes in physical systems. One proposition is a perturbation of another just in case it is possible for there to be a direct transition between a state in which φ holds and a state in which ψ holds. φ is said to dominate ψ iff: whenever an open interval over which φ holds immediately precedes or

[14] Here we take **P** and **F** as referring strictly to the past and future, whereas some logics use weaker tense operators such that whenever φ is true $\mathbf{P}\varphi$ and $\mathbf{F}\varphi$ are also true.

follows an open interval over which ψ holds, then φ must also hold at the intermediate point between these two intervals.

(D9) $\mathsf{Perturbation}(\varphi, \psi) \equiv_{def} \Diamond((\varphi \wedge (\lhd\psi \vee \rhd\psi)) \vee (\psi \wedge (\lhd\varphi \vee \rhd\varphi)))$.
(D10) $\mathsf{Dominates}(\varphi, \psi) \equiv_{def} \Box(((\lhd\varphi \wedge \rhd\psi) \vee (\lhd\psi \wedge \rhd\varphi)) \rightarrow \varphi)$.

6.3. Operators forming propositions from event-types

The Occurs predicate, which relates event-types to the intervals over which they occur, has the following simple definition:

(D11) $\mathsf{Occurs}(\mathbf{e}, \delta) \equiv_{def} (\exists \mathbf{e}, \varepsilon)[\mathsf{dur}(\varepsilon) = \delta]$.

The aspect operators, which convert event radicals into propositions, are defined as follows:

(D12) $\mathit{Perf}(\mathbf{e}) \equiv_{def} (\exists\delta)(\exists t)[\mathsf{AT}(t) \wedge (\mathsf{e}(\delta) \leqslant t) \wedge \mathsf{Occurs}(\mathbf{e}, \delta)]$.
(D13) $\mathit{Pros}(\mathbf{e}) \equiv_{def} (\exists\delta)(\exists t)[\mathsf{AT}(t) \wedge (t \leqslant \mathsf{b}(\delta)) \wedge \mathsf{Occurs}(\mathbf{e}, \delta)]$.
(D14) $\mathit{Prog}(\mathbf{e}) \equiv_{def} (\exists\delta)(\exists t)[\mathsf{AT}(t) \wedge (\mathsf{b}(\delta) < t < \mathsf{e}(\delta)) \wedge \mathsf{Occurs}(\mathbf{e}, \delta)]$.

We use the notation $\mathbf{e}(\varepsilon)$ to say that the event token ε is of type $\mathbf{e}$. This is defined by

(D15) $\mathbf{e}(\varepsilon) \equiv_{def} (\exists \mathbf{e}, \varepsilon')[\varepsilon = \varepsilon']$.

We may also want to compare event-types. The relation $\mathbf{e}_1 \sqsubseteq \mathbf{e}_2$ means that $\mathbf{e}_1$ is a sub-type of $\mathbf{e}_2$, and $\mathbf{e}_1 \equiv \mathbf{e}_2$ means that $\mathbf{e}_1$ and $\mathbf{e}_2$ are equivalent:

(D16) $\mathbf{e}_1 \sqsubseteq \mathbf{e}_2 \equiv_{def} \Box(\forall\delta)[\mathsf{Occurs}(\mathbf{e}_1, \delta) \rightarrow \mathsf{Occurs}(\mathbf{e}_2, \delta)]$.
(D17) $\mathbf{e}_1 \equiv \mathbf{e}_2 \equiv_{def} (\mathbf{e}_1 \sqsubseteq \mathbf{e}_2) \wedge (\mathbf{e}_2 \sqsubseteq \mathbf{e}_1)$.

We can also define event modalities analogous to those used in dynamic logic:

(D18) $[\mathbf{e}]\varphi \equiv_{def} \boxast(\forall\delta)[(\mathsf{AT}(\mathsf{b}(\delta)) \wedge \mathsf{Occurs}(\mathbf{e}, \delta)) \rightarrow \mathsf{Holds\text{-}at}(\varphi, \mathsf{e}(\delta))]]$.
(D19) $\langle\mathbf{e}\rangle\varphi \equiv_{def} \Diamond\!\!\ast(\exists\delta)[\mathsf{AT}(\mathsf{b}(\delta) \wedge \wedge \mathsf{Occurs}(\mathbf{e}, \delta) \wedge \mathsf{Holds\text{-}at}(\varphi, \mathsf{e}(\delta))]]$.

6.4. Complex event-types

Using event abstraction we can define more complex event types. Conjunctive and disjunctive event-types can be defined as follows:

(D20) $\langle \mathbf{e}_1 \sqcap \mathbf{e}_2 \rangle =_{def} \langle \delta : (\mathsf{Occurs}(\mathbf{e}_1, \delta) \wedge \mathsf{Occurs}(\mathbf{e}_2, \delta)) \rangle$.
(D21) $\langle \mathbf{e}_1 \sqcup \mathbf{e}_2 \rangle =_{def} \langle \delta : (\mathsf{Occurs}(\mathbf{e}_1, \delta) \vee \mathsf{Occurs}(\mathbf{e}_2, \delta)) \rangle$.

We can also define an event-type corresponding to a concatenated sequence of two component event-types:

(D22) $\langle \mathbf{e}_1; \mathbf{e}_2 \rangle =_{def} \langle \delta : (\exists \delta_1, \delta_2)[\mathsf{Occurs}(\mathbf{e}_1, \delta_1) \wedge \mathsf{Occurs}(\mathbf{e}_2, \delta_2) \wedge \mathsf{b}(\delta) = \mathsf{b}(\delta_1) \wedge \mathsf{e}(\delta_1) = \mathsf{b}(\delta_2) \wedge \mathsf{e}(\delta_2) = \mathsf{e}(\delta)] \rangle$.

We can introduce operators for converting an event-type, **e**, into punctual event-types corresponding to beginnings and endings of events of type **e**:

(D23) $\langle \mathsf{begin}\, \mathbf{e} \rangle =_{def} \langle \delta_1 : (\exists \delta_2)[\mathsf{Occurs}(\mathbf{e}, \delta_2) \wedge \mathsf{b}(\delta_1) = \mathsf{e}(\delta_1) \wedge \mathsf{b}(\delta_1) = \mathsf{b}(\delta_2)] \rangle$.
(D24) $\langle \mathsf{end}\, \mathbf{e} \rangle =_{def} \langle \delta_1 : (\exists \delta_2)[\mathsf{Occurs}(\mathbf{e}, \delta_2) \wedge \mathsf{b}(\delta_1) = \mathsf{e}(\delta_1) \wedge \mathsf{e}(\delta_1) = \mathsf{e}(\delta_2)] \rangle$.

VEL allows the definition of many more useful operators for forming complex event-types. For instance we can define various types of 'process' consisting of sequences of repeating and possibly overlapping sub-events. These will not be covered in the present paper.

6.5. *Constructing event-types from propositions*

It is often natural to define event-types by reference to the changing truth values of propositions. Within VEL we can define punctual event types which occur at the beginning and end points at which a proposition is true, or at points where a proposition is instantaneously true:

(D25) $\langle \mathsf{begin}\, \varphi \rangle \equiv_{def} \langle \delta : \mathsf{Holds\text{-}at}((\varphi \wedge \lhd\neg\varphi), \mathsf{b}(\delta)) \wedge (\mathsf{b}(\delta) = \mathsf{e}(\delta)) \rangle$.
(D26) $\langle \mathsf{end}\, \varphi \rangle \equiv_{def} \langle \delta : \mathsf{Holds\text{-}at}((\varphi \wedge \rhd\neg\varphi), \mathsf{b}(\delta)) \wedge (\mathsf{b}(\delta) = \mathsf{e}(\delta)) \rangle$.
(D27) $\langle \mathsf{inst}\, \varphi \rangle \equiv_{def} \langle \langle \mathsf{begin}\, \varphi \rangle \sqcap \langle \mathsf{end}\, \varphi \rangle \rangle$.

We can also regard the period of time over which a proposition is true as characterising an event. It is convenient to first define $\mathsf{Holds\text{-}on}(\varphi, \delta)$ to mean that φ holds at all time points interior to the interval δ.[15]

(D28) $\mathsf{Holds\text{-}on}(\varphi, \delta) \equiv_{def} (\mathsf{b}(\delta) < \mathsf{e}(\delta)) \wedge (\forall t)[(\mathsf{b}(\delta) < t < \mathsf{e}(\delta)) \rightarrow \mathsf{Holds\text{-}at}(\varphi, t)]$.

We now define the construct $\langle \varphi \rangle$ to refer to the event-type which occurs during every interval such that φ is true for all times between its end points:[16]

(D29) $\langle \varphi \rangle \equiv_{def} \langle \delta : \mathsf{Holds\text{-}at}(\lhd\neg\varphi, \mathsf{b}(\delta)) \wedge \mathsf{Holds\text{-}on}(\varphi, \delta) \wedge \mathsf{Holds\text{-}at}(\rhd\neg\varphi, \mathsf{e}(\delta)) \rangle$.

Note that $\langle \varphi \rangle$ occurs over an interval whether or not φ holds at its end points. Depending on the meaning of the proposition φ, intervals over which it holds may be either open or closed at each end. However we could easily define more restricted operators that give event-types corresponding to only open or only closed periods over which φ holds (or indeed periods that are open at one end and closed at the other).

---

[15] Note, that according to this definition, if δ is punctual no proposition Holds-on δ.
[16] The expression $\langle \varphi \rangle$ has the same meaning as the 'po φ' construct introduced in [16].

6.6. Predicates and functions of states

Using Holds-at and the state function together with the alternative history operator, we can define the condition that a proposition holds in a particular state as follows:[17]

(D30) $\mathsf{Holds\text{-}in}(\varphi, s) \equiv_{def} \Box(\forall t)[\mathsf{state}(t) = s \rightarrow \mathsf{Holds\text{-}at}(\circledast\varphi, t)]$.

A proposition φ can only hold in a state s if the truth of φ at s does not depend on the future, otherwise its truth value is not settled by the state and $\mathsf{Holds\text{-}in}(\varphi, s)$ is taken to be false.

When describing scenarios in terms of states it is useful to have a predicate to pick out the actual state and a function to give the time at which a state occurs:

(D31) $\mathsf{AS}(s) \equiv_{def} (\exists t)[\mathsf{AT}(t) \wedge \mathsf{state}(t) = s]$.
(D32) $\mathsf{time}(s) = \tau \equiv_{def} \Diamond(\mathsf{state}(\tau) = s)$.

AS enables us to define useful relationships of accessibility between states. $(\mathbf{e}: s_1 \Rightarrow s_2)$ says that one can get from state s_1 to state s_2 *via* an occurrence of an event of type **e** (this relation is closely related to the accessibility relation of the $[\mathbf{e}]$ modal operator). $s_1 \Rightarrow s_2$ says that (either $s_1 = s_2$ or) on some history s_2 lies in the future of s_1.

(D33) $(\mathbf{e}: s_1 \Rightarrow s_2) \equiv_{def} \Diamond(\mathsf{AS}(s_1) \wedge (\exists\delta)[\mathsf{AT}(\mathsf{b}(\delta) \wedge \mathsf{Occurs}(\mathbf{e}, \delta) \wedge \mathsf{state}(\mathsf{e}(\delta)) = s_2])$.
(D34) $(s_1 \Rightarrow s_2) \equiv_{def} (s_1 = s_2) \vee \Diamond(\mathsf{AS}(s_1) \wedge \mathbf{F}\,\mathsf{AS}(s_2))$.

7. Proof theory

The logic VEL is not designed as a language for automated reasoning. It is a general semantic theory of events, within which a variety of less expressive but more practical representations can be embedded. Nevertheless, a proof system for VEL could be used to test inferences within sublanguages of VEL and also for proving properties of the relationships between these sublanguages. Given the complexity of VEL the task of specifying an axiom system for the language is extremely difficult. Nevertheless we present the following set of axioms which we believe is close to being complete relative to our semantics.

7.1. An axiomatic proof system

We need to axiomatise the following undefined logical symbols: $\neg\varphi$, $\varphi \wedge \psi$, $\beta_1 = \beta_2$, $\tau_1 \leqslant \tau_2$, $\mathsf{Holds\text{-}at}(\varphi, t)$, $\mathsf{AT}(t)$, $\circledast\varphi$, null, $\mathsf{b}(\delta)$, $\mathsf{e}(\delta)$, $\mathsf{dur}(\varepsilon)$, $\mathsf{state}(t)$, $\langle\delta : \varphi(\delta)\rangle$ and the six kinds of quantifier.

[17] If we were to weaken the **SconfP** and **Gclock** constraints on history trees, other definitions of Holds-in might be more natural.

7.1.1. Substitution notation

In stating the proof theory we shall use the following notation to specify the result of substituting one variable for another in a formula:

- $\varphi^{t \Rightarrow t'}$ is the result of substituting t' for one or more occurrences of t in φ, which do not occur within the scope of a quantification w.r.t. either t or t'.
- $\varphi^{t \Rrightarrow t'}$ is the result of substituting t' for *all* occurrences of t in φ which do not occur within the scope of a quantification w.r.t. t'. None of these occurrences may occur within the scope of a quantification w.r.t. t.

The first of these describes the type of substitution justified by an equality and the second describes that used in instantiating a universal quantifier. In both cases the substituent must be free and the substituent must not become bound.

7.1.2. Classical propositional logic

At the core of the system is the classical propositional logic:

(R1) $\vdash \varphi$ if φ is a classical propositional theorem.
(R2) If $\vdash \varphi$ and $\vdash (\varphi \to \psi)$ then $\vdash \psi$.

7.1.3. Identity and quantification theory

All types of nominal variables and terms obey the axioms of reflexive identity and substitution of necessary identicals:

(A1) $\alpha = \alpha$, where α is any term.
(A2) $(\Box(\alpha = \alpha') \wedge \varphi) \to \varphi^{\alpha \Rightarrow \alpha'}$, where α and α' are nominal terms of the same type.

Since VEL is an intensional logic, substitution generally requires necessary identity rather then mere extensional identity. However, all variables except individuals are purely extensional. Thus we have the following axiom:

(A3) $(\chi = \chi') \to \Box(\chi = \chi')$, where χ and χ' are variables both of the same type, which is one of σ_t, σ_δ, σ_s, σ_o or σ_ε.

Although individuals are intensional, their reference at an index $\langle h, t\rangle$ is determined by the state $h(t)$. Thus, where $h(t) = h'(t)$ (i.e., h and h' are confluent at t) an individual constant will have the same denotation at $\langle h, t\rangle$ and $\langle h', t\rangle$. This semantic constraint is captured by the axiom:

(A4) $(\alpha_1 = \alpha_2) \to \circledast(\alpha_1 = \alpha_2)$, where α_1 and α_2 are symbols of type σ_i.

Variables denoting time points, intervals, states and observable values obey completely standard quantification theory. Let ν be a variable of any of the types σ_t, σ_δ, σ_s or σ_o and ν' be any variable of the same type as ν. Then the proof theory for quantification over these types of variable can be specified by the following rule and axiom:

(R3) If $\vdash (\varphi \rightarrow \psi)$ then $\vdash (\varphi \rightarrow \forall \nu[\psi])$, provided ν is not free in φ.
(A5) $\forall \nu[\varphi] \rightarrow \varphi^{\nu \Rrightarrow \nu'}$.

7.1.4. Temporal logic

The temporal logic obeys the following inference rule and axioms:

(R4) If $\vdash \varphi$ then $\vdash \mathsf{Holds\text{-}at}(\varphi, t)$.
(A6) $(t \leqslant t' \wedge t' \leqslant t'') \rightarrow t \leqslant t''$.
(A7) $t \leqslant t' \vee t' \leqslant t$.
(A8) $(t \leqslant t' \wedge t' \leqslant t) \leftrightarrow t = t'$.
(A9) $(\mathsf{Holds\text{-}at}(\varphi, t) \wedge \mathsf{Holds\text{-}at}(\varphi \rightarrow \psi, t)) \rightarrow \mathsf{Holds\text{-}at}(\psi, t)$.
(A10) $\neg\mathsf{Holds\text{-}at}(\varphi \wedge \neg\varphi, t)$.
(A11) $\mathsf{Holds\text{-}at}(\varphi, t) \vee \mathsf{Holds\text{-}at}(\neg\varphi, t)$.
(A12) $\mathsf{Holds\text{-}at}(\varphi, t) \leftrightarrow \mathsf{Holds\text{-}at}(\mathsf{Holds\text{-}at}(\varphi, t), t')$.
(A13) $t \leqslant t' \leftrightarrow \mathsf{Holds\text{-}at}((t \leqslant t'), t'')$.
(A14) $\forall t[\mathsf{Holds\text{-}at}(\varphi, t')] \rightarrow \mathsf{Holds\text{-}at}(\forall t[\varphi], t')$.
(A15) $(\mathsf{AT}(t) \wedge \mathsf{AT}(t')) \rightarrow t = t'$.
(A16) $\mathsf{Holds\text{-}at}(\mathsf{AT}(t), t)$.
(A17) $\varphi \rightarrow \exists t[\mathsf{Holds\text{-}at}(\varphi, t)]$.

Axioms (A6)–(A8) constrain the temporal comparison $\leqslant$ to satisfy the usual axioms for a total linear order. (A9) and (A10) ensure that the set of formulae holding at each time point is closed under implication and consistent. (A11) ensures that, for every proposition φ, at each time point either φ or $\neg\varphi$ holds. (A14) means that time quantification commutes with the $\mathsf{Holds\text{-}at}(\ldots, t)$ construct. Finally, axioms (A15)–(A17) give essential properties of the $\mathsf{AT}(t)$ predicate.

The temporal axioms given so far constrain time to be an arbitrary linear order. For applications in ontological modelling of physical processes there are strong arguments for requiring that time be dense (see, e.g., [20,22]). This can easily be ensured by adding the axiom:[18]

(A18) $\forall t_1 t_2[t_1 < t_2 \rightarrow \exists t[t_1 < t < t_2]]$.

To ensure that the truth values of propositions do not fluctuate infinitely often in a finite period of time we also impose the following axioms, which can be concisely expressed using the $\triangleright$ and $\triangleleft$ operators:

(A19) $\triangleright\varphi \vee \triangleright\neg\varphi$.
(A20) $\triangleleft\varphi \vee \triangleleft\neg\varphi$.

[18] One might also want time to be continuous; but this requires a second-order axiom and doesn't seem to be necessary for most applications.

These say that at each time point any proposition has a constant truth value during some open interval immediately following that point and also during some open interval immediately preceding the point.[19]

7.1.5. The $\boxast$ operator

The $\boxast$ operator satisfies the following proof rule and axioms:[20]

(R5) If $\vdash \varphi$ then $\vdash \boxast\varphi$.
(A21) $(\boxast\varphi \wedge \boxast(\varphi \to \psi)) \to \boxast\psi$.
(A22) $\boxast\varphi \to \varphi$.
(A23) $\diamondast\varphi \to \boxast\diamondast\varphi$.
(A24) $\mathsf{AT}(t) \to \boxast\mathsf{AT}(t)$.
(A25) $t \leqslant t' \to \boxast(t \leqslant t')$.
(A26) $\forall tt'[\mathsf{Holds\text{-}at}(\boxast\varphi, t) \wedge t \leqslant t' \to \mathsf{Holds\text{-}at}(\boxast\mathsf{Holds\text{-}at}(\varphi, t), t')]$.

(R5) is the rule of necessitation and axioms (A21)–(A23) are the standard modal schemata **K**, **T** and **5**. Thus $\boxast$ is an $S5$ modal operator. Axioms (A24) and (A25) ensure that the actual time and all temporal inequalities are the same for alternative histories. Finally, (A26) captures the interaction between $\boxast$ and the $\mathsf{Holds\text{-}at}(\varphi, t)$ construct, arising from the fact that all historical alternatives at a given time share a common history up to that time.[21]

7.1.6. Intervals

The b and e functions obey the following axioms, which ensure that the beginning of an interval cannot come after its end and that there is a unique interval determined by every two time points:

(A27) $\mathsf{b}(\delta) \leqslant \mathsf{e}(\delta)$.
(A28) $(t_1 \leqslant t_2) \to (\exists\delta)[t_1 = \mathsf{b}(\delta) \wedge t_2 = \mathsf{e}(\delta)]$.
(A29) $(\mathsf{b}(\delta_1) = \mathsf{b}(\delta_2) \wedge \mathsf{e}(\delta_1) = \mathsf{e}(\delta_2)) \to (\delta_1 = \delta_2)$.

---

[19] These axioms are not equivalent: for example, in real-number time, if φ is true on the intervals $(2^{-2n}, 2^{-2n+1)})$ (for $n = 1, 2, 3, \ldots$) and at no other times, then (A20) holds throughout but (A19) is false at $t = 0$. If the temporal theory were combined with a theory of physical reality, it is possible that these axioms might be derivable from basic continuity properties of physical change.

[20] These axioms were inspired by a set given by Thomason [47] and attributed to Kamp [31] for a logic whose models are similar to but somewhat less constrained than our history trees. Kamp found that his axioms were incomplete relative to his intended models, because they do not support inferences dependent on temporal interpolations involving more than one history; however, we believe that our explicit axiomatisation of a time structure shared by all histories circumvents this problem.

[21] Since $\boxast$ is an $S5$ modality one can derive both the Barcan formula and the converse Barcan formula for $\boxast$ with respect to time quantification—i.e., we have the theorem $\forall t[\boxast\varphi] \leftrightarrow \boxast\forall t[\varphi]$.

7.1.7. Count nouns

To specify the logic of count nouns we use a proof rule and axioms which are a simplification of those given by Gupta [24].[22] Count noun quantification obeys the rule:[23]

(R6) If $\vdash \varphi \rightarrow \psi$ then $\vdash \varphi \rightarrow (\forall \text{C}, x)[\psi]$, provided x is not free in φ

as well as the axioms:

(A30) $((\forall \text{C}, x)\varphi \wedge \text{C}[a]) \rightarrow \varphi^{x \Rrightarrow a}$.
(A31) $(\forall \text{C}, x)\text{C}[x]$, provided x is not free in C.
(A32) $\neg \text{C}(\mathsf{null})$.

Rule (R6) and axiom (A30) provide a standard basis for quantification theory, with the slight modification that the substitutions allowed by (A30) are restricted to individuals falling under the appropriate count noun. Axiom (A31) specifies a connection between count noun quantification and necessary identity.[24] (A32) ensures that the null entity does not belong to the domain of any count noun.

Gupta also motivates the adoption of the following count noun *separation* axiom,[25] which we may want to impose, although it is not actually necessitated by the model theory given for VEL. The axiom enforces the condition that if the same count noun applies (or possibly applies) to each of two co-realised, non-null individuals, then those individuals must be identical (i.e., co-realised at all worlds). Equivalently, if two individuals are co-realised but not identical then they must be individuals of different categories, having different identity and persistence criteria (e.g., a pot and a lump of clay):

(A33) $(a = b \wedge \neg(a = \mathsf{null}) \wedge \Diamond\text{C}[a] \wedge \Diamond\text{C}[b]) \rightarrow \Box(a = b)$.

We also add the axiom (A34), which enforces the condition that the individuals falling under a count noun (at the index point where it is evaluated) must have a non-empty extension (at that index).

(A34) $(\forall \text{C}, x)[\neg(x = \mathsf{null})]$.

[22] The only substantial difference is that we (in (A2)) always require necessary identity to permit substitutions, whereas Gupta allows extensionally equivalent individual variables to be substituted into contexts that are not within the scope of a modal operator. This is because our semantics allows atomic relations and verbs to be intensional, so no context can be guaranteed transparent by considering its syntax alone.

[23] Actually Gupta uses the weaker proof rule: if $\vdash \varphi$ then $\vdash (\forall \text{C}, x)[\psi]$; and adds the further axioms $((\forall \text{C}, x)(\varphi \rightarrow \psi) \wedge (\forall \text{C}, x)\varphi) \rightarrow (\forall \text{C}, x)\psi$ and $\varphi \rightarrow (\forall \text{C}, x)\varphi$, provided x is not free in φ. Together these are equivalent to our stronger proof rule.

[24] Recall that $\text{C}[x] \equiv_{def} (\exists \text{C}, x')[\Box(x = x')]$. Gupta says that this axiom may not be independent of the others but we have not been able to derive it.

[25] We actually use Gupta's AS13*, the weaker of two separation axioms that he considers.

7.1.8. Event quantification

The event quantification employed in VEL takes the form of quantifying over the event-tokens of a particular event-type. This is much the same as ordinary sorted quantification and inferences involving event quantifiers can be reduced to the following simple rules:

(R7) If $\vdash (\varphi \rightarrow \psi)$ then $\vdash (\varphi \rightarrow \langle \forall \mathbf{e}, \varepsilon \rangle [\psi])$, provided ε does not occur free in φ.

(A35) $(\langle \forall \mathbf{e}, \varepsilon \rangle \varphi \wedge \mathbf{e}(\varepsilon')) \rightarrow \varphi^{\varepsilon \Rightarrow \varepsilon'}$.

7.1.9. Event abstraction and the dur function

The following axiom characterises the event abstraction operator by ensuring that the intervals over which the instances of an abstracted event-type occur satisfy the interval predicate that was used to make the abstraction:[26]

(A36) $\varphi(\delta) \leftrightarrow \langle \exists \langle \delta' : \varphi(\delta') \rangle, \varepsilon \rangle [\mathsf{dur}(\varepsilon) = \delta]$.

7.1.10. States and the state function

Finally we ensure that the state function respects the structure of the history frames. We need to ensure that the history tree constraint **Gclock** holds, that every two histories have a confluence point, and that this point is unique (**SconfP**):

(A37) $(\mathsf{state}(t_1) = s \wedge \Diamond \mathsf{state}(t_2) = s) \rightarrow t_1 = t_2$.

(A38) $(\forall s_1, s_2)(\exists s_3)[(s_3 \Rightarrow s_1) \wedge (s_3 \Rightarrow s_2)]$.

(A39) $(\forall t_1, t_2)[(t_1 < t_2 \wedge \mathsf{state}(t_1) = s_1 \wedge \mathsf{state}(t_2) = s_2) \rightarrow$
$\Box(\mathsf{state}(t_2) = s_2 \rightarrow \mathsf{state}(t_1) = s_1)]$.

7.2. Completeness

The task of establishing a complete axiom system for VEL is ongoing work. However, Bennett [7] does give a completeness proof for a large fragment of the logic including the constructs: $\neg\varphi$, $\varphi \wedge \psi$, $\beta_1 = \beta_2$, $\tau_1 \leqslant \tau_2$, $\mathsf{Holds\text{-}at}(\varphi, t)$, $\mathsf{AT}(t)$, $\boxtimes\varphi$, together with quantification over time variables. That paper shows that the proof system consisting of rules (R1)–(R17) and axioms (A1)–(A4) is complete (for that sub-language) with respect to the branching history semantics.

Our axioms for interval variables and the b and e functions are very straightforward and we believe that it will be a routine matter to extend the completeness proof to include this part of the language. Gupta [24] proves completeness of his axioms with respect to a very general modal semantics which seems to be compatible with the more specialised semantics of VEL. However, giving a rigorous completeness proof for the count-noun axioms within the setting of VEL would be a substantial task. As regards our axioms for event quantification, abstraction and the state function, these are based on our intuitive

[26] Expressed in terms of the defined Occurs relation, (A36) becomes $\varphi(\delta) \leftrightarrow \mathsf{Occurs}(\langle \delta' : \varphi(\delta') \rangle, \delta)$. And from this it is easy to prove that $(\mathbf{e} \equiv \langle \delta : \varphi(\delta) \rangle) \leftrightarrow \Box \forall \delta [\varphi(\delta) \leftrightarrow \mathsf{Occurs}(\mathbf{e}, \delta)]$.

understanding of the VEL models. Here it is possible that we have omitted some essential property.

8. Additional logical constraints

Because of the general nature of our semantics, possible interpretations of certain types of expression are more fluid than one would normally want. Specifically, since expressions such as relations, event-types and count nouns are intensional they could have quite different 'meanings' at different indices in a history tree. This limits the class of valid inferences involving multiple index points. Clearly, most ordinary concepts have a much more rigid meaning than this. Hence we present a number of constraints which one will often wish to apply to the basic vocabulary of theories formulated within VEL.[27]

8.1. Static relations

In general, relation symbols may correspond to intentional relationships holding among individuals. However, many fundamental relations are extensional in that, at every index point, whether the relation holds of some tuple of individuals depends only on the entities denoted by the individuals at the particular state corresponding to that index point. In other words the truth of these relations depends on the actual state of the world. We thus refer to them as static relations. Amongst the static relations are all spatial relations and most other physical properties. An example of a non-static relation is '$\mathsf{is\text{-}approaching}(a, b)$'.

A static predicate $\varphi(x)$ satisfies the following axiom schema, which means that only extensional equivalence (rather than necessary identity) is required for substitution into $\varphi(x)$.

(C1) $(\varphi(\alpha) \wedge (\alpha = \beta)) \rightarrow \varphi(\beta)$.

8.2. Subjective event-types

Normally the meaning of an event-type expression is objective in that whatever possible world it is evaluated at, it denotes the same set of episodes. However, within the intensional language of VEL it is possible to define subjective event-types whose denotation is not constant. For instance we might define a type consisting of all accidents that take place during 'the next snow storm'. Since 'the next snow storm' varies according where we are in a VEL frame, any event-type defined in terms of a phrase such as this would be subjective.

Subjective event-types do not correspond to events in the ordinary sense but they arise inevitably within a sufficiently rich intensional language. Most events which one will want to talk about are not subjective and satisfy:

[27] Originally we attempted to build such constraints into the actual semantics of VEL but we found that this was incompatible with the definitional capabilities of the language.

(C2) $\mathsf{Holds\text{-}at}((\forall \mathbf{e}, \varepsilon)\varphi, t) \leftrightarrow (\forall \mathbf{e}, \varepsilon)\mathsf{Holds\text{-}at}(\varphi, t)$.
(C3) $(\forall s_1, s_2)[\Diamond(\mathbf{e} : s_1 \Rightarrow s_2) \rightarrow \Box(\mathbf{e} : s_1 \Rightarrow s_2)]$.

8.3. Verbs of direct participation

Participation in many verbs is determined completely by the static properties exemplified by individuals during the period over which the events corresponding to that verb occur. In the place of an individual argument α of such a verb we can substitute any other individual which is co-realised by α at all indices during the episode over which the event occurs. For instance if a ship burns then the quantity of wood co-realised with that ship also burns. More generally, this concept of *direct participation* can be attributed to a particular argument place within an expression describing a dynamic property. This is enforced by constraints of the form:

(C4) $(\mathsf{Occurs}(\mathbf{v}(\alpha), \delta) \wedge (\forall t)[(\mathsf{b}(\delta) \leqslant t \leqslant \mathsf{e}(\delta))$
$\rightarrow \mathsf{Holds\text{-}at}(\alpha = \beta, t)]) \rightarrow \mathsf{Occurs}(\mathbf{v}(\beta), \delta)$.

9. Relation to other AI theories

We now look at how two established AI theories of events can be described within VEL. It is worth noting that, despite the expressiveness of VEL, we encountered considerable difficulty in arriving at reasonable formulations. Because AI theories are oriented towards achieving certain reasoning tasks, they employ abstractions of the structure of time and events that are geared to achieving a compromise between expressivity and computational tractability. But how to spell out the nature of these abstractions relative to the very general model structures provided by VEL is not always obvious.

9.1. Situation Calculus

The Situation Calculus [35] is a very well-known and influential representation for actions. The basic idea of this system is to use a language which contains, in addition to propositional expressions (*fluents*), variables over a domain of *situations*. Fluents are related to situations by the special relation $\mathsf{Holds}(\varphi, s)$. Action types are then modelled as functions from situations to situations. More specifically, if a is an action type, then $\mathsf{result}(a, s)$ is a term denoting the situation which occurs immediately after action a is performed in situation s.

Let us look at how the Situation Calculus could be interpreted within VEL. Our formalism already has situation variables and McCarthy's Holds predicate corresponds very closely to our Holds-in predicate. The only problem is the result function. The Situation Calculus is based on an assumption that when an action is performed in a given state then a unique state will result; but within our model of states and events there may be many different situations which could be the result of a given action because any given event-type can occur in various different ways.

We have the option of simply introducing *sit-calc-actions* into VEL as functions from states to states (possibly with additional parameters). We could then reconstruct Situation Calculus faithfully within VEL. However, this leaves completely unconstrained the relationship between sit-calc-actions and the verbs, event-types and event-tokens already provided by VEL. But we ought to have some strong connection between sit-calc-actions and event-types.

A drastic solution is to force all VEL events to have a unique result state by adding a further axiom:

(A40) $\langle \mathbf{e} \rangle \mathsf{AS}(s) \to [\mathbf{e}]\mathsf{AS}(s)$.

We could then identify sit-calc-actions directly with event-types. However, this would severely restrict the generality of our framework. In particular it would mean that we could no longer define event-types in terms of arbitrary conditions on the time intervals over which they occur.

A similar but more subtle approach is to treat Situation Calculus actions as a special sub-type of VEL events for which (A40) holds. Thus rather than adopting (A40) as a general axiom, we would have a specific constraining formula $\langle \mathbf{sca}_i \rangle \mathsf{AS}(s) \to [\mathbf{sca}_i]\mathsf{AS}(s)$ for each sit-calc-action $\mathbf{sca}_i$. So, only the sit-calc-actions would be constrained to correspond to deterministic events.

At first consideration it may seem that, within the branching time structure of VEL, it would make little sense to simply deem certain event-types to be deterministic. However, since event-types are semantically identified with classes of episodes, any event-type must have at least one deterministic sub-type. Moreover, many modes of reasoning in terms of arbitrary event-types can be shown to be reducible to equivalent forms of reasoning with deterministic event-types. For instance, whenever there is a consecutive series of intervals $\delta_1, \ldots, \delta_n$, satisfying a sequence of event types $\mathbf{e}_1, \ldots, \mathbf{e}_n$, the same series of intervals must satisfy a sequence of deterministic event-types $\mathbf{e}'_1, \ldots, \mathbf{e}'_n$, such that each $\mathbf{e}'_i \subseteq \mathbf{e}_i$. So, to find whether a given goal can be achieved by a series of events, we need only consider whether it can be achieved by a series of deterministic events. Also, effect axioms for non-deterministic event-types, which constrain the states of all their possible outcomes, will clearly apply equally to any deterministic sub-types of those events. Thus, confining attention to deterministic events can often be a very useful simplifying abstraction of the space of possible event sequences. This is perhaps a reason for the power of the Situation Calculus ontology. However, we believe that allowing non-deterministic events provides for much more flexible modelling of parallel event sequences, especially where we do not wish to impose any fixed synchronisation between parallel events.

Another approach to representing the Situation Calculus, which may also be more appropriate in some cases, is to recast the function result as a kind of quantifier over possible resulting states. Thus rather than writing $\Phi(\mathsf{result}(\mathbf{e}))$ (where $\Phi(\ldots)$ is any VEL predicate that can take a state variable as argument) we would use the form $(\mathsf{result}\ \mathbf{e}, s)[\Phi(s)]$, where $(\mathsf{result}\ \mathbf{e}, s)[\Phi(s)] \equiv_{def} (\forall s')[(\mathbf{e} : s \Rightarrow s') \to \Phi(s')]$. By using this transform, axioms which make sense under a deterministic model of action, can be transformed into analogous axioms which hold in a branching model that allows an occurrence of a given event-type occurring in a given state to lead to multiple possible outcome states.

9.2. Event calculus

Another formalism that has been used for describing and reasoning about actions and events is the Event Calculus [32]. An account of this calculus and its variants can be found in [44]. At the heart of the Event Calculus is a principle of inertia: if an event causes some 'fluent' (i.e., proposition) to hold, then it will continue to hold until some later event causes it to become false. This idea can be formalised by the following basic axioms of Event Calculus:[28]

(EC1) $\mathsf{Holds\text{-}at}(\varphi, t_2) \leftarrow$
$\mathsf{Happens}(\varepsilon, t_1) \wedge \mathsf{Initiates}(\varepsilon, \varphi, t_1) \wedge (t_1 < t_2) \wedge \neg\mathsf{Clipped}(t_1, \varphi, t_2).$

(EC2) $\mathsf{Clipped}(t_1, \varphi, t_2) \leftarrow \mathsf{Happens}(\varepsilon, t) \wedge (t_1 < t < t_2) \wedge \mathsf{Terminates}(\varepsilon, \varphi, t).$

The predicate Clipped is introduced primarily to achieve a Horn clause formulation. Since it is not normally employed except within these two axioms (whereas the other predicates will also occur in scenario descriptions) it can be eliminated to give an equivalent single axiom:

(EC) $\mathsf{Holds\text{-}at}(\varphi, t_2) \leftarrow \mathsf{Happens}(\varepsilon, t_1) \wedge \mathsf{Initiates}(\varepsilon, \varphi, t_1) \wedge (t_1 < t_2) \wedge$
$\neg\exists\varepsilon'[\mathsf{Happens}(\varepsilon', t) \wedge (t_1 < t < t_2) \wedge \mathsf{Terminates}(\varepsilon', \varphi, t)].$

Here we formulate the axiom as applying to event-tokens, although later versions of the Event Calculus often work in terms of event types. Direct representation of the analogous event-type axiom would not be possible in VEL because we do not allow quantification over event-types.[29] Nevertheless, although inertia is specified in terms of tokens, one can still express conditions on the Initiates and Terminates predicates in terms of event-types, and these conditions would be inherited by the tokens of these types. Thus we can equally well emulate Event Calculus style reasoning, in which a causal theory is formulated either in terms of event-types or tokens (or both if we wish). Moreover, at the end of this section we shall give an encoding of Event Calculus in terms of event-types, where we use schematic event-type variables instead of quantification.

Let us now define VEL analogues of the Event Calculus predicates. Holds-at can be represented directly by the VEL predicate of the same name. With the other predicates we meet a small problem, because the events of Event Calculus are normally taken as being punctual (happening at a single point in time), whereas VEL events can occur either over punctual or extended intervals. To account for this we could restrict the axiom to apply only to punctual events. However, a more general approach seems to be available. The Event Calculus reasons about cause and effect in terms of which fluents are made to hold

[28] These are the axioms given for the 'Simplified Event Calculus' of Shanahan [44, p. 252].

[29] Of course we could add event-type quantification, but this would have a second-order character, which we would like to avoid.

or cease to hold immediately after an event. Hence, it is the end point of the event that is relevant to this kind of reasoning. So in our VEL encoding we shall define Happens, Initiates and Terminates by reference to the end points of the events involved:

(D35) $\mathsf{Happens}(\varepsilon, \tau) \equiv_{def} (\tau = \mathsf{e}(\mathsf{dur}(\varepsilon)))$.
(D36) $\mathsf{Initiates}(\varepsilon, \varphi, \tau) \equiv_{def} (\tau = \mathsf{e}(\mathsf{dur}(\varepsilon)) \wedge \mathsf{Holds\text{-}at}(\rhd\varphi, \tau))$.
(D37) $\mathsf{Terminates}(\varepsilon, \varphi, \tau) \equiv_{def} \mathsf{Initiates}(\varepsilon, \neg\varphi, \tau)$.

In order to specify what happens and the effects of initiation and termination in terms of event-types we could use the following defined predicates:

(D38) $\mathsf{Happens}^*(\mathbf{e}, \tau) \equiv_{def} \exists\delta[\mathsf{Occurs}(\mathbf{e}, \delta) \wedge \mathsf{e}(\delta) = \tau]$.
(D39) $\mathsf{Initiates}^*(\mathbf{e}, \varphi) \equiv_{def} \Box(\langle\forall\mathbf{e}, \varepsilon\rangle[\forall t[(\mathsf{e}(\mathsf{dur}(\varepsilon)) = t) \rightarrow \mathsf{Initiates}(\varepsilon, \varphi, t)]])$.
(D40) $\mathsf{Terminates}^*(\mathbf{e}, \varphi) \equiv_{def} \mathsf{Initiates}^*(\mathbf{e}, \neg\varphi)$.

Since event-token quantification in VEL is always relative to an event-type we will also need a further definition to enable us to quantify over a type of generic event that occurs over every arbitrary interval. Thus we define:[30]

(D41) **any-event** $\equiv_{def} \langle\delta : \delta = \delta\rangle$.

Using our definitions EC can now be represented in VEL by the formula:

(VEL-EC) $\langle\forall\mathbf{any\text{-}event}, \varepsilon\rangle[(\mathsf{Happens}(\varepsilon, t_1) \wedge \mathsf{Initiates}(\varepsilon, \varphi, t_1) \wedge \neg\langle\exists\mathbf{any\text{-}event}, \varepsilon'\rangle$
$(\exists t)[\mathsf{Happens}(\varepsilon', t) \wedge (t_1 < t < t_2) \wedge \mathsf{Terminates}(\varepsilon', \varphi, t)]) \rightarrow \mathsf{Holds\text{-}at}(\varphi, t_2)]$.

What is the status of this formula? Since it represents the basis of Event Calculus reasoning one might hope that it is valid within VEL. But as it stands it is not. The problem is that VEL allows the following counter-example to VEL-EC: φ might be initiated by some event at t_1 and remain true for all times t such that $(t_1 < t < t_2)$ but then φ could still become false at t_2.

The reason why $(t_1 < t < t_2)$ works for the Event Calculus is that it treats fluents as holding over intervals that are open at their beginning and closed at their end.[31] We shall call such an interval a BOEC interval; and a proposition that only holds over such intervals

[30] Note that tokens of **any-event** will be distinct form tokens of any more specific type, so in quantifying over **any-event** tokens we are *not* quantifying over arbitrary event-tokens. But, since our definitions of Happens, Initiates and Terminates only refer to the end times of event-tokens and the changes in proposition that occur after these times, whenever an event of a more specific type happens and initiates or terminates a proposition there will also be a token of **any-event** that happens at the same time and initiates/terminates the same proposition. Hence, cause and effect rules specified for tokens of more specific event-types will be mirrored by the occurrences of tokens of the generic **any-event** event-type.

[31] To be more precise, this requirement only applies to those fluents that are initiated or terminated. It is these *inertial* fluents that are forced by the axioms to maintain a persistent truth value between initiating and terminating events. A dual interpretation, where the duration of inertial fluents is closed at the beginning and open at the end, is also possible.

will be called a BOEC proposition. When an event ε changes a proposition φ from true to false, φ is taken as remaining true up to the time at which the event 'Happen's and then becoming false for some BOEC interval immediately following the time of the event.

From an ontological viewpoint, the condition that propositions should be BOEC seems to us unnatural. In our view, whether a proposition holds over an open or a closed interval is not an arbitrary modelling decision, but is determined by the meaning of the proposition. For instance a state of 'connectedness' between two objects that can move continuously in space can hold only over closed intervals, because the distance $d_t(a,b)$ between objects a and b at time t must be a continuous function of time. Thus any interval over which $d_t(a,b) = 0$ must be closed. Similarly, 'disconnectedness' between a and b occurs over intervals where $d_t(a,b) > 0$, and these must be open.

Nevertheless, if one is interested in implementing practical reasoning algorithms, we accept that there may be good reasons to describe the world using BOEC propositions. In many cases one can get round the limitations of the BOEC requirement by adjusting the meanings of non-BOEC propositions to get a closely related BOEC version. For example, one can use a modified version of 'connectedness', $\mathsf{C}_{\lhd}(x,y)$, which is true at a time point t just in case x and y were connected throughout some open interval immediately preceding t (i.e., $\mathsf{C}_{\lhd}(x,y) \equiv_{def} \lhd\mathsf{C}(x,y)$). There would still be a problem representing propositions that hold for an isolated instant (e.g., a thrown ball being at the apex of its trajectory). However, in richer versions of the Event Calculus, which have been developed to take account of continuously varying fluents, additional apparatus is used which can represent propositions that are true at instants or over closed intervals (e.g., by the use of a 'Trajectory' predicate as proposed in [43] and further developed in [44,45]).

In VEL we can easily define a propositional operator 'BOEC', such that $\mathsf{BOEC}(\varphi)$ is true just in case φ is BOEC:[32]

(D42) $\mathsf{BOEC}(\varphi) \equiv_{def} \Box((\varphi \rightarrow \lhd\varphi) \wedge (\neg\varphi \rightarrow \lhd\neg\varphi))$.

We can now take the basis of Event Calculus reasoning as being represented by the VEL axiom ($\mathsf{BOEC}(\varphi) \rightarrow$ VEL-EC), which can be shown to be valid according to the VEL semantics.

An alternative approach to representing Event Calculus within VEL would be to weaken VEL-EC by replacing $(t_1 < t < t_2)$ by $(t_1 < t \leqslant t_2)$. This would mean that some modification would be required in the reasoning mechanisms that are usually applied to compute inferences in Event Calculus, but we see no reason why this should not be feasible.

The two alternatives correspond to the following slight modifications of VEL-EC, each of which is valid in VEL:

(VEL-EC1) $\mathsf{BOEC}(\varphi) \rightarrow \langle\forall \textbf{any-event}, \varepsilon\rangle[(\mathsf{Happens}(\varepsilon, t_1) \wedge \mathsf{Initiates}(\varepsilon, \varphi, t_1) \wedge$
$\neg\langle\exists \textbf{any-event}, \varepsilon'\rangle(\exists t)[\mathsf{Happens}(\varepsilon', t) \wedge (t_1 < t < t_2) \wedge$
$\mathsf{Terminates}(\varepsilon', \varphi, t)]) \rightarrow \mathsf{Holds\text{-}at}(\varphi, t_2)]$.

[32] This definition also implies that φ cannot undergo the pathological intermingling behaviour that was considered in Section 2.8. Thus, where we restrict to BOEC propositions, we do not need to enforce axioms (A19) and (A20).

(VEL-EC2) $\langle\forall\textbf{any-event}, \varepsilon\rangle[(\mathsf{Happens}(\varepsilon, t_1) \wedge \mathsf{Initiates}(\varepsilon, \varphi, t_1) \wedge \neg\langle\exists\textbf{any-event}, \varepsilon'\rangle(\exists t)[\mathsf{Happens}(\varepsilon', t) \wedge (t_1 < t \leqslant t_2) \wedge \mathsf{Terminates}(\varepsilon', \varphi, t)]) \rightarrow \mathsf{Holds\text{-}at}(\varphi, t_2)]$.

We conclude this section by showing how each of VEL-EC1 and VEL-EC2 can be transformed back into a Horn clause formulation which is very close to published formats of the Event Calculus (e.g., [45]).

First we shall define $\mathsf{Clipped}_<$ and $\mathsf{Clipped}_\leqslant$ as follows:

(D43) $\mathsf{Clipped}_<(t_1, \varphi, t_2) \equiv_{def} \langle\exists\textbf{any-event}, \varepsilon\rangle[\mathsf{Happens}(\varepsilon, t) \wedge (t_1 < t < t_2) \wedge \mathsf{Terminates}(\varepsilon, \varphi, t)]$.

(D44) $\mathsf{Clipped}_\leqslant(t_1, \varphi, t_2) \equiv_{def} \langle\exists\textbf{any-event}, \varepsilon\rangle[\mathsf{Happens}(\varepsilon, t) \wedge (t_1 < t \leqslant t_2) \wedge \mathsf{Terminates}(\varepsilon, \varphi, t)]$.

From this it is easy to show that the following schematic Horn formulae are valid, where **e** stands for any event-type:

(Cl1) $\mathsf{Clipped}_<(t_1, \varphi, t_2) \leftarrow \mathsf{Happens}^*(\mathbf{e}, t) \wedge (t_1 < t < t_2) \wedge \mathsf{Terminates}{*}(\mathbf{e}, \varphi, t)$.

(Cl2) $\mathsf{Clipped}_\leqslant(t_1, \varphi, t_2) \leftarrow \mathsf{Happens}^*(\mathbf{e}, t) \wedge (t_1 < t \leqslant t_2) \wedge \mathsf{Terminates}^*(\mathbf{e}, \varphi, t)$.

If we substitute $\mathsf{Clipped}_<$ and $\mathsf{Clipped}_\leqslant$ respectively into VEL-EC1 and VEL-EC2, from the resulting formulae we can prove the following Horn schemas:

(VEL-EC-HS1) $\mathsf{Holds\text{-}at}(\varphi, t_2) \leftarrow \mathsf{BOEC}(\varphi) \wedge \mathsf{Happens}^*(\mathbf{e}, t_1) \wedge \mathsf{Initiates}^*(\mathbf{e}, \varphi, t_1) \wedge \neg\mathsf{Clipped}_<(t_1, \varphi, t_2)$.

(VEL-EC-HS2) $\mathsf{Holds\text{-}at}(\varphi, t_2) \leftarrow \mathsf{Happens}^*(\mathbf{e}, t_1) \wedge \mathsf{Initiates}^*(\mathbf{e}, \varphi, t_1) \wedge \neg\mathsf{Clipped}_\leqslant(t_1, \varphi, t_2)$.

We could use either VEL-EC-HS1 with Cl1 or VEL-EC-HS2 with Cl2 as the basis of an inference procedure, which would work in much the same way as existing Event Calculus implementations (e.g., SLD resolution with negation as failure). This means that an Event Calculus inference engine could be used as a reasoning module for VEL. Of course it would be limited to computing inferences from sets of VEL formulae that are expressed in form analogous to an Event Calculus theory. Also, if negation as failure is used to resolve the negated Clipped literals, one would be importing an additional non-monotonic inference rule into the VEL logic.

10. Conclusion

We have given a precise formal semantics for a very expressive temporal language (VEL), which is capable of describing events in a variety of different ways. In particular it incorporates within a single system several of the most influential approaches to representing time and events. The framework provides a general temporal ontology within which more practically oriented representations can be interpreted. It also supports the design of modular reasoning systems combining decision procedures for tractable sublanguages of VEL.

In further work we hope to establish a complete axiom system for VEL and to relate our system to formalisms such as CTL* [13], whose computational properties are better understood [14]. We would also like to explore how VEL might take account of issues such as the *frame problem* and how it relates to non-monotonic formalisms that deal with this.

We plan to extend VEL to incorporate the representation of space and matter. Preliminary work on describing physical processes [5,6] suggests that VEL is well suited to this purpose. It would also be useful to establish the relationship of the VEL semantics to ontologies based on four-dimensional spatio-temporal regions [28,30].

References

[1] J.F. Allen, An interval-based representation of temporal knowledge, in: Proceedings of IJCAI-81, Vancouver, BC, 1981, pp. 221–226.

[2] J.F. Allen, Towards a general theory of action and time, Artificial Intelligence 23 (2) (1984) 123–154.

[3] A.B. Baker, Nonmonotonic reasoning in the framework of the situation calculus, Artificial Intelligence (Special Issue on Knowledge Representation) 49 (1991) 5–23.

[4] H. Barringer, R. Kuiper, A. Pnueli, A really abstract concurrent model and its fully abstract semantics, in: Proceedings of the 13th ACM Symposium on Principles of Programming Languages (POPL), St. Petersburg Beach, FL, 1986, pp. 173–183.

[5] B. Bennett, Space, time, matter and things, in: C. Welty, B. Smith (Eds.), Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS'01), Ogunquit, 2001, pp. 105–116.

[6] B. Bennett, Physical objects, identity and vagueness, in: D. Fensel, D. McGuinness, M.-A. Williams (Eds.), Proceedings of the Eighth International Conference on Principles of Knowledge Representation and Reasoning (KR2002), Toulouse, France, Morgan Kaufmann, San Francisco, CA, 2002.

[7] B. Bennett, A logic of branching histories with a shared linear time series, in: Foundations and Applications of Spatio-Temporal Reasoning (FASTR), AAAI Spring Symposium, Palo Alto, CA, 2003.

[8] F.M. Brown (Ed.), Proceedings of the 1987 Workshop, The Frame Problem in Artificial Intelligence, Morgan Kaufmann, Los Altos, CA, 1987.

[9] D. Davidson, The logical form of action sentences, in: N. Rescher (Ed.), The Logic of Decision and Action, University of Pittsburgh Press, Pittsburgh, PA, 1967, reprinted in: Essays on Actions and Events, Oxford University Press, Pittsburgh, 1980.

[10] D. Davidson, The individuation of events, in: N. Rescher (Ed.), Essays in Honor of Carl G. Hempel, D. Reidel, 1969, reprinted in: Essays on Actions and Events, Oxford University Press, Oxford, 1980.

[11] E. Davis, Representations of Commonsense Knowledge, Morgan Kaufmann, San Mateo, CA, 1990.

[12] E. Davis, Infinite loops in finite time: Some observations, in: B. Nebel, C. Rich, W. Swartout (Eds.), Proceedings of the Third International Conference, Principles of Knowledge Representation and Reasoning (KR'92), Cambridge, MA, Morgan Kaufmann, San Mateo, CA, 1992, pp. 47–58.

[13] E.A. Emerson, J.Y. Halpern, 'Sometimes' and 'not never' revisited: On branching time versus linear time temporal logics, J. ACM 33 (1) (1986) 151–178.

[14] E.A. Emerson, C.S. Jutla, The complexity of tree automata and logics of programs, SIAM J. Comput. 29 (1) (1999) 132–158.

[15] R.E. Fikes, N.J. Nilsson, STRIPS: A new approach to the application of theorem proving to problem solving, Artificial Intelligence 2 (1971) 189–208.

[16] A.P. Galton, The Logic of Aspect: An Axiomatic Approach, Clarendon Press, Oxford, 1984.

[17] A.P. Galton, The logic of occurrence, in: A.P. Galton (Ed.), Temporal Logics and their Applications, Academic Press, London, 1987, pp. 169–196, Chapter 5.

[18] A.P. Galton, Towards an integrated logic of space, time and motion, in: Proceedings of IJCAI-93, Chambéry, France, 1993.

[19] A.P. Galton, An investigation of 'non-intermingling' principles in temporal logic, J. Logic Comput. 6 (2) (1996) 271–294.

[20] A.P. Galton, Space, time and movement, in: O. Stock (Ed.), Spatial and Temporal Reasoning, Kluwer, Dordrecht, 1997, pp. 321–352, Chapter 10.
[21] A.P. Galton, Qualitative Spatial Change, Oxford University Press, Oxford, 2000.
[22] A.P. Galton, Transitions in continuous time, with an application to qualitative changes in spatial relations, in: H. Barringer, et al. (Eds.), Proceedings of the Second International Conference on Temporal Logic, Advances in Temporal Logic, ICTL'97, Kluwer, Dordrecht, 2000, pp. 279–297.
[23] N. Guarino (Ed.), Proceedings of the 1st International Conference, Formal Ontology in Information Systems (FOIS-98), Trento, Italy, Frontiers in Artificial Intelligence and Applications, vol. 46, IOS Press, Amsterdam, 1998.
[24] A. Gupta, The Logic of Common Nouns: An Investigation in Quantified Modal Logic, Yale University Press, New Haven, CT, 1980.
[25] S. Hanks, D. McDermott, Default reasoning, nonmonotonic logics, and the frame problem, in: Proceedings of AAAI-86, Philadelphia, PA, 1986, pp. 328–333.
[26] D. Harel, Dynamic Logic, in: D. Gabbay, F. Guenthner (Eds.), Handbook of Philosophical Logic, vol. II: Extensions of Classical Logic, Reidel, Dordrecht, 1984, pp. 497–604.
[27] P.J. Hayes, The frame problem and related problems in artificial intelligence, in: A. Elithorn, D. Jones (Eds.), Artificial and Human Thinking, Jossey-Bass, San Fransisco, CA, 1973, pp. 495–520.
[28] S.M. Hazarika, A.G. Cohn, Qualitative spatio-temporal continuity, in: D.R. Montello (Ed.), Spatial Information Theory: Foundations of Geographic Information Science; Proceedings of COSIT'01, Morro Bay, CA, in: Lecture Notes in Comput. Sci., vol. 2205, Springer, Berlin, 2001, pp. 92–107.
[29] P.J. Hayes, The naïve physics manifesto, in: D. Michie (Ed.), Expert Systems in the Micro-Electronic Age, Edinburgh University Press, Edinburgh, 1979.
[30] M. Heller, The Ontology of Physical Objects: Four Dimensional Hunks of Matter, Cambridge University Press, Cambridge, 1991.
[31] J. Kamp, The logic of historical necessity, part I, Unpublished typescript, 1979.
[32] R. Kowalski, M. Sergot, A logic-based calculus of events, New Gen. Comput. 4 (1986) 67–95.
[33] F. Lehmann (Ed.), Semantic Networks in Artificial Intelligence, Pergamon Press, Oxford, 1992.
[34] D. Lenat, CYC: A large-scale investment in knowledge infrastructure, Comm. ACM 38 (11) (1995).
[35] J. McCarthy, P. Hayes, Some philosophical problems from the standpoint of artificial intelligence, in: B. Meltzer, D. Michie (Eds.), Machine Intelligence, vol. 4, Edinburgh University Press, Edinburgh, 1969, pp. 463–502.
[36] D. McDermott, A temporal logic for reasoning about processes and plans, Cognitive Sci. 6 (1982) 101–155.
[37] F. Moller, G. Birtwistle (Eds.), Logics for Concurrency: Structure versus Automata, Lecture Notes in Compute. Sci., vol. 1043, Springer, Berlin, 1996.
[38] F. Moller, A. Rabinovich, On the expressive power of CTL*, in: Proceedings of LICS'99: Logic in Computer Science, Trento, Italy, 1999.
[39] R. Montague, English as a formal language. Linguaggi nella Società e nella Tecnica, 1970. Reprinted in: H. Thomason (Ed.), Formal Philosophy: Selected Papers of Richard Montague, Yale University Press, New Haven, CT, 1970.
[40] R. Montague, Universal grammar. Theoria, 1970, pp. 153–167. Reprinted in: H. Thomason (Ed.), Formal Philosophy: Selected Papers of Richard Montague, Yale University Press, New Haven, CT, 1970.
[41] D. Randell, Z. Cui, A. Cohn, A spatial logic based on regions and connection, in: Proc. 3rd Internat. Conf. on Knowledge Representation and Reasoning, Morgan Kaufmann, San Mateo, CA, 1992, pp. 165–176.
[42] E. Sandewall, Features and Fluents, Oxford University Press, Oxford, 1994.
[43] M. Shanahan, Representing continuous change in the event calculus, in: L.C. Aiello (Ed.), Proceedings of the 9th European Conference on Artificial Intelligence (ECAI-90), Stockholm, Sweden, 1990, pp. 598–603.
[44] M. Shanahan, Solving the Frame Problem: A Mathematical Investigation of the Common Sense Law of Intertia, MIT Press, Cambridge, MA, 1997.
[45] M. Shanahan, The event calculus explained, in: M.J. Wooldridge, M. Veloso (Eds.), Artificial Intelligence Today, in: Lecture Notes in Comput. Sci., vol. 1600, Springer, Berlin, 1999, pp. 409–430.
[46] Y. Shoham, Time for action: On the relation between time, knowledge and action, in: Proceedings of the IJCAI-89, Detroit, MI, 1989, pp. 954–959.
[47] R. Thomason, Combinations of tense and modality, in: D. Gabbay, F. Guenthner (Eds.), Handbook of Philosophical Logic, vol. II, Reidel, Sweden, 1984, pp. 135–165, Chapter 3.