Representing and Reasoning with Events from Natural Language

Miguel Leith and Jim Cunningham

Department of Computing
Imperial College of Science, Technology and Medicine
180 Queen's Gate, London, England

E-mails: *mfl@doc.ic.ac.uk* and *rjc@doc.ic.ac.uk*

Abstract. Linguistic categories such as progressives, perfectives, tense and temporal adverbials are at the heart of our ability to describe events in natural language. Following on from the work of Moens and Steedman and the later work of Kent, we have identified a fragment of an interval tense logic of Halpern and Shoham that is expressive enough to represent the temporal readings of many simple sentences involving the linguistic categories listed above, and computable enough for entailment checking to be manageable in a reasonable time scale. We show how one can model the semantics of formulae from the fragment using simple timelines and how one can support entailment checking by comparing timelines using a simple algorithm.

Keywords: temporal reasoning, natural language understanding.

1 Introduction

Linguistic categories such as progressives, perfectives, tense and temporal adverbials are at the heart of our ability to describe events in natural language. Through these categories, which we will loosely call *temporal*, we express the relative positions of events in time, characterized by temporal relations such as *later than, adjoins, overlaps, begins, ends* or *includes*. Even from the simplest sentences involving temporal categories people are able to understand something of the relative positioning of events in time. For example the sentence *Max stopped running* might give rise to the informal temporal reading in Fig. 1.

The analysis of events in natural language is complicated by the existence of sentences which embody a sense of culmination, for example *accomplishments* like *Jill ate the apple*, and sentences which involve various types of temporal adverbial phrase such as *in five minutes* or *at three o'clock*. Even for quite simple sentences it is hard to find a straightforward formalism that is suitable for representing their temporal readings and checking their entailments while being applicable in wider problems such as analysing narrative texts. Our work here focuses on providing such a formalism.

The logical representation of tense and aspect in natural language has considerable tradition and many open issues which will not be discussed here. The

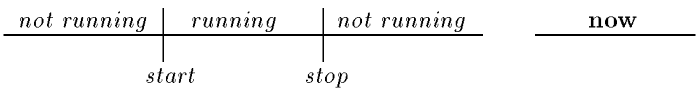

Fig. 1. Time periods for the sentence *Max stopped running*

reader is referred to the recent book by Verkuyl [9]. We take as our point of departure a thesis by Kent [4]. Kent draws on work of Moens [6], Lascarides [5], and Moens and Steedman [7] in order to develop a formal account of linguistic behaviour in a complex interval logic.[1] In doing so he provides substantial evidence for the adequacy of interval models in explicating the entailments between simple related sentences.[2] However despite the use of representational conventions which extend modal style tense logics, from a computational perspective the absence of a proof theory for Kent's logic is a hindrance to exploitation.

We build on certain aspects of Kent's work but adopt a different choice of formalism and characterization of linguistic behaviour. We show how we may use formulae in the temporal modal logic known as HS, described by Halpern and Shoham in [3], to represent the temporal readings of various sentences involving temporal linguistic categories while retaining the implicit temporality and economy of expression of tense language. We then show that by using these temporal readings and the properties of HS logic we can demonstrate some of the entailments that exist between sentences. Our aims are made achievable by defining a fragment of HS called HSF1. The fragment is sufficiently expressive to represent the temporal readings of the kinds of sentences we are interested in, while the semantic models that we build for its formulae (which we refer to as *timelines*) are quite simple, allowing entailments to be checked by carrying out comparisons with a straightforward algorithm. Examples of the types of sentences we will be looking at are given in Fig. 2. They include certain uses of the present progressive, past tense, present perfective and past progressive, with both activity and accomplishment verb phrases. Some sentences also include simple temporal adverbial expressions. Examples of the types of entailments we can account for are given in Fig. 3. Within the later sections of this paper we will give examples of how we model the listed sentences and entailments.

The structure of this paper, then, is as follows. In Sect.2 we give a summary of the syntax and semantics of the logic HS and discuss the modal operators which are of immediate use to us. In Sect.3 we describe the logic fragment HSF1, show how we can model the semantics of its formulae as timelines and

---

[1] Kent's logical notation is complex partly because time is allowed to branch in both directions and intervals can be either open or closed. Logical operators are defined as mappings between interval sets, his proofs are then external mathematical arguments.

[2] There is an additional point to be made regarding the expressiveness of interval logics when compared to point logics. Certain properties may hold at intervals in time but not at points within those intervals, as was commented by Allen and Ferguson in [2].

1 - *Max is running.*
2 - *Max has run.*
3 - *Max stopped running.*
4 - *Jill ate the apple in five minutes.*
5 - *Jill is eating an apple.*
6 - *Fred was playing the piano for an hour.*
7 - *Mark started racing at two o'clock.*

Fig. 2. Example sentences for which we wish to model semantic readings

1 - *Max is running* $\Rightarrow$ *Max has run*
2 - *Jill is eating an apple* $\not\Rightarrow$ *Jill has eaten the apple*
3 - *Jill stopped eating an apple* $\not\Rightarrow$ *Jill ate the apple*
4 - *Fred walked to the station* $\Rightarrow$ *Fred was walking to the station*

Fig. 3. Example entailments which we wish to model

illustrate some of our temporal readings. In Sect.4 we describe an entailment checker which can carry out comparisons between timelines in a reasonable time scale, and demonstrate its use in modelling linguistic entailments. Finally in Sect.5 we make our conclusions.

2 The Logic HS

HS is the interval tense logic of Halpern and Shoham, which was first described in [3] and further investigated by Venema in [8]. In HS the set of possible worlds is the set of closed intervals in a temporal frame which may be dense or discrete. Its six basic modal operators express relative positional interval relationships, and are described below.

$\langle B\rangle\varphi$: *φ holds at a beginning interval of the current interval.*
$\langle \underline{B}\rangle\varphi$: *$\varphi$ holds at an interval begun by the current interval*
$\langle E\rangle\varphi$: *φ holds at an end interval of the current interval.*
$\langle \underline{E}\rangle\varphi$: *$\varphi$ holds at an interval ended by the current interval.*
$\langle A\rangle\varphi$: *φ holds at an interval adjoining the current interval at its end.*
$\langle \underline{A}\rangle\varphi$: *$\varphi$ holds at an interval adjoining the current interval at its beginning.*

Another six may be expressed by derived modal operators[3], namely $\langle L\rangle\varphi \equiv \langle A\rangle\langle A\rangle\varphi$ (later), $\langle \underline{L}\rangle\varphi \equiv \langle \underline{A}\rangle\langle \underline{A}\rangle\varphi$ (earlier), $\langle D\rangle\varphi \equiv \langle B\rangle\langle E\rangle\varphi$ (during), $\langle \underline{D}\rangle\varphi \equiv \langle \underline{B}\rangle\langle \underline{E}\rangle\varphi$ (included by), $\langle O\rangle\varphi \equiv \langle E\rangle\langle \underline{B}\rangle\varphi$ (overlapping) and $\langle \underline{O}\rangle\varphi \equiv \langle B\rangle\langle \underline{E}\rangle\varphi$ (overlapped by).

[3] Those familiar with Allen's interval logic, described in [1], may notice a similarity between the HS modal operators and Allen's interval relations. Specifically, DURING(i1, i2) is similar to $\langle D\rangle$ and $\langle \underline{D}\rangle$, STARTS(i1, i2) to $\langle B\rangle$ and $\langle \underline{B}\rangle$, FINISHES(i1, i2) to $\langle E\rangle$ and $\langle \underline{E}\rangle$, BEFORE(i1, i2) to $\langle L\rangle$ and $\langle \underline{L}\rangle$, OVERLAP(i1, i2) to $\langle O\rangle$ and $\langle \underline{O}\rangle$, MEETS(i1, i2) to $\langle A\rangle$ and $\langle \underline{A}\rangle$

There are also derived modal operators for the beginning point and end point of an interval. Since the proposition $[B]\mathit{false}$ only holds at point intervals in HS logic, Halpern and Shoham define the beginning point modal operator $[[BP]]$ and the end point modal operator $[[EP]]$[4] as:

$$[[BP]]\varphi \equiv ((\varphi \wedge [B]\mathit{false}) \vee (\langle B\rangle(\varphi \wedge [B]\mathit{false}))$$
$$[[EP]]\varphi \equiv ((\varphi \wedge [B]\mathit{false}) \vee (\langle E\rangle(\varphi \wedge [B]\mathit{false}))$$

The correspondence of the fourteen modal operators to fourteen possible relative positions between intervals is illustrated in the diagram Fig. 4.

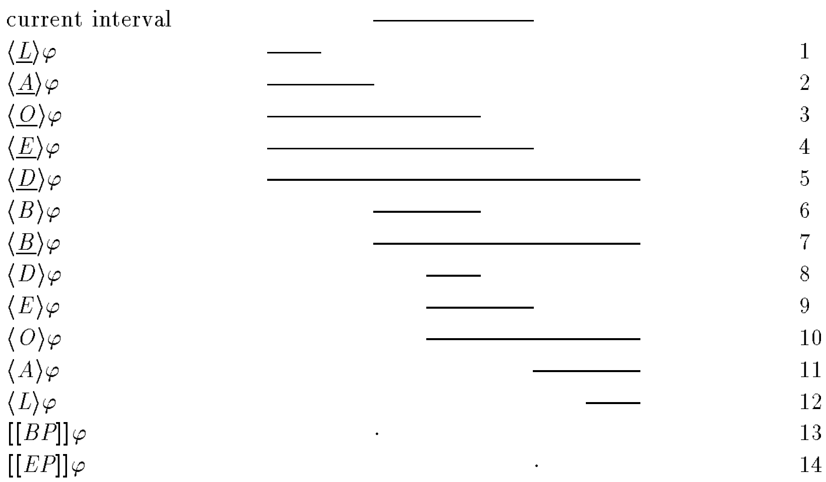

Fig. 4. Pictorial illustration of relative interval positions and their HS formulae

2.1 Syntax of HS

Given Φ_0 as the set of atomic propositions, made up of the propositional constants p_0, p_1, ..., the syntax of HS is defined inductively as follows.

All members of Φ_0 are well formed formulae (wffs).
false is a wff.
$\varphi_1 \rightarrow \varphi_2$ is a wff iff φ_1 is a wff and φ_2 is a wff.
$\langle B\rangle\varphi$, $\langle E\rangle\varphi$, $\langle A\rangle\varphi$, $\langle \underline{B}\rangle\varphi$, $\langle \underline{E}\rangle\varphi$ and $\langle \underline{A}\rangle\varphi$ are all wffs iff φ is a wff.

The connectives $\neg$, $\wedge$, $\vee$ and $\leftrightarrow$ and the truth constant *truth* are defined in terms of $\rightarrow$ and *false* in the usual manner. The necessitation operators $[B]$, $[E]$ etc. are defined as usual as $\neg\langle B\rangle\neg$, $\neg\langle E\rangle\neg$ etc.

[4] We will tend to look at points simply as special cases of intervals.

2.2 Semantics of HS

A *(temporal) frame* is a pair $F = (T,<)$, where T is a set of time points and $<$ is a strict partial order on T. The *interval set* of a frame F is called $INT(F)$ and is defined as the set of all closed intervals $[s,t] = \{x \in T \mid \mathrm{s} \leq \mathrm{x} \leq \mathrm{t}\}$ in T. A model M is a pair (F, V), where F is a frame and V is a *valuation*, i.e. a map $\Phi_0 \mapsto 2^{INT(F)}$. Next we inductively define the truth relation $\models$ as follows:

$\models^M_{[s,t]} P$ iff $[s,t] \in V(P)$

$\not\models^M_{[s,t]} \mathit{false}$

$\models^M_{[s,t]} (\varphi_1 \rightarrow \varphi_2)$ iff $\models^M_{[s,t]} \varphi_1$implies $\models^M_{[s,t]} \varphi_2$

$\models^M_{[s,t]} \langle B \rangle \varphi$ iff there is a point u such that $s \leq u < t$ and $\models^M_{[s,u]} \varphi$

$\models^M_{[s,t]} \langle \underline{B} \rangle \varphi$ iff there is a point u such that $t < u$ and $\models^M_{[s,u]} \varphi$

$\models^M_{[s,t]} \langle E \rangle \varphi$ iff there is a point u such that $s < u \leq t$ and $\models^M_{[u,t]} \varphi$

$\models^M_{[s,t]} \langle \underline{E} \rangle \varphi$ iff there is a point u such that $u < s$ and $\models^M_{[u,t]} \varphi$

$\models^M_{[s,t]} \langle A \rangle \varphi$ iff there is a point u such that $t < u$ and $\models^M_{[t,u]} \varphi$

$\models^M_{[s,t]} \langle \underline{A} \rangle \varphi$ iff there is a point u such that $u < s$ and $\models^M_{[u,s]} \varphi$

The concepts of *validity* and *satisfiability* of formulae with respect to models, frames and classes of frames are defined in the usual manner.

2.3 Operators of HS That Are of Immediate Use to Us

It has been shown by Halpern and Shoham in [3] that the validity problem in HS logic is undecidable in all but the simplest classes of frames such as finite sets of integers. However, as is pointed out by Venema in [8], there may be natural and useful fragments of the logic that behave better. Such a fragment that interests us is one where we restrict the set of modal operators to all of the possibility operators plus one of the necessitation operators, $[D]$. The $[D]$ operator is of immediate use to us, as it captures a highly important principle in temporal logics known as *homogeneity.* A proposition is said to be homogeneous if it is true at all sub-intervals of any intervals at which it holds. Homogeneity is used by Kent in [4] as a property on states. If someone was running for five minutes, for instance, then they were running at all periods of time within that five minutes, unless an exception is specified, in which case the proposition may be split into two or more homogeneous periods instead. By having a formula $[D]$*(Max_run)* hold at an interval we are saying that Max is running at all intervals/points during that interval. Using the $[D]$ operator to characterize homogeneity appears to be more elegant than in classical temporal logics, where one defines an axiom such as $HOLDS(p,\ T) \Leftrightarrow (\forall t.IN(t,\ T) \Rightarrow HOLDS(p,\ t))$.

3 HSF1: A Fragment of HS logic

The HSF1 fragment contains all of the possibility operators from HS and one of the necessitation operators, $[D]$. We begin this section by defining the syntax and semantics of the fragment. We then describe how we represent the semantics of formulae from the fragment using specialised models which we refer to as *timelines*. The algorithm by which timelines are constructed is described. This is followed by a description of our treatment of states and culminations, and lastly by some examples of how we represent temporal readings of simple sentences as HSF1 formulae.

3.1 Syntax of HSF1

The syntax of HS is defined using grammar rules as follows.

HSF1 : Exp2 | PossOp HSF1
PossOp : $\langle D\rangle$ | $\langle \underline{D}\rangle$ | $\langle E\rangle$ | $\langle \underline{E}\rangle$ | $\langle B\rangle$ | $\langle \underline{B}\rangle$ | $\langle L\rangle$ | $\langle \underline{L}\rangle$ | $\langle A\rangle$ | $\langle \underline{A}\rangle$ | $\langle O\rangle$ | $\langle \underline{O}\rangle$
Exp2 : Exp3 | HSF1 $\wedge$ HSF1 | HSF1 $\vee$ HSF1
Exp3 : ExtendedAtomic | $[[BP]]$ Atomic | $[[EP]]$ Atomic
ExtendedAtomic : Atomic | $[D]$ Atomic
Atomic : PropConstant | $\neg$ PropConstant
PropConstant : p_1 | p_2 | p_3 ...

3.2 Semantics of HSF1

Semantics of the modal operators in HSF1 are the same as those given in the semantics of HS. However, we impose a restriction on frames by requiring that the set of points T in a temporal frame is dense, linear and unbounded.

3.3 Timelines

Proposition 1. *Any HSF1 formula may be modelled semantically by a finite number of alternative data structures which we will refer to as timelines.*

Definition 2. A timeline is a pair $\langle$*Ordering, Bindings*$\rangle$ where *Ordering* is a sequence of point labels in linear dense unbounded time and *Bindings* is a set of associations $\langle (Pt1,\ Pt2)\!: P\rangle$ where *Pt1* and *Pt2* are point labels (from the ordering) signifying the bounds of an interval and P is an extended atomic formula.

Definition 3. A timeline is said to be inconsistent iff any two of its bindings take the following forms or their converses:
$\langle (Pt1,\ Pt2) : p\rangle$ and $\langle (Pt1,\ Pt2) : \neg p\rangle$
$\langle (Pt1,\ Pt2) : [D](p)\rangle$ and $\langle (Pt1,\ Pt2) : [D](\neg p)\rangle$
$\langle (Pt1,\ Pt2) : [D](p)\rangle$ and $\langle (Pt3,\ Pt4) : \neg p\rangle$ where $Pt1 < Pt3$ and $Pt4 < Pt2$
$\langle (Pt1,\ Pt2) : [D](p)\rangle$ and $\langle (Pt3,\ Pt4) : [D](\neg p)\rangle$
where ($Pt3 < Pt1$ and $Pt1 < Pt4$) or ($Pt3 < Pt2$ and $Pt2 < Pt4$)

Proposition 1 is justified because in the case of the HSF1 fragment, the semantics of an HSF1 formula can be interpreted by a procedure which builds specialised semantic models. The models, which we refer to as *timelines*, comprise an ordering relation represented by a series of point labels and a truth valuation represented by a set of associations between pairs of point labels (intervals) and extended atomic formulae. In general there may be more than one possible timeline for a given HSF1 formula but each temporal reading of a sentence in our examples is represented by a formula which has only one timeline.

We will give here an informal explanation of the procedure by which timelines are built, which is known as the Timeline Construction Algorithm. The algorithm is a recursive traversal of an HSF1 formula that builds an ordering relation and truth valuation along the way. At each stage of the recursive traversal we are at a particular *current interval* denoted by a pair of point labels e.g. (*Pt1*, *Pt2*). Each modal operator (except for $[D]$) is a possibility operator in its positive form (e.g. $\langle E\rangle(\varphi)$ etc.) and effectively asserts the existence of an interval in given temporal relation to the current interval. Thus the effect of encountering a possibility operator in the algorithm is to change the current interval and update the ordering relation to reflect the points of the new current interval in their correct positions. If the algorithm encounters an extended atomic formula it will create a *binding* between the current interval and the formula. For instance if it encounters a formula $[D](p)$ at current interval (*Pt3*, *Pt4*) it will create a binding $\langle(Pt3, Pt4) : [D](p)\rangle$ and add it to the list of bindings that the algorithm is building.

By the preceding argument a timeline modelling an HSF1 formula of depth n (where we count the starting interval and each possibility modality as adding 1 to the depth) has at most $2n$ time points in its ordering. Also a timeline modelling an HSF1 formula that contains m extended atomic formulae has at most m bindings.

3.4 Single-timeline HSF1 Formulae

Generally an HSF1 formula may have more than one timeline that may model its semantics. However those formulae which may be modelled by only one timeline are of special interest. Our temporal reading formulae all share this property, which is advantageous since entailments between them may be checked very quickly. It is only certain pairs of HSF1 operators which, when appearing as sub-formulae, result in alternative timelines. A simple example is the formula $\langle B\rangle\langle\underline{B}\rangle(p)$. At the stage where we analyse $\langle\underline{B}\rangle$ the current interval is, say, (*Pt1*, *Pt3*) and the ordering is [... *Pt1* ... *Pt3* ... *Pt2* ...]. The new current interval will be (*Pt1*, *Pt4*) where *Pt4* is a new point that may be inserted anywhere after *Pt3* in the ordering. This means it can go in any of three places: before, at or after *Pt2*. We call pairs of HSF1 operators that behave this way *cross-directional*. The three basic cross-directional pairs are: $\langle B\rangle$ and $\langle\underline{B}\rangle$, $\langle E\rangle$ and $\langle\underline{E}\rangle$, $\langle A\rangle$ and $\langle\underline{A}\rangle$[5]. Expansions of derived operators result in further cross-directional pairs

[5] Venema used the term *mirror images* in [8] in reference to these pairs, because they are reflections of each other in a northwest plane correspondence.

e.g. since $\langle L \rangle \equiv \langle A \rangle \langle A \rangle$, the pair $\langle \underline{A} \rangle$ and $\langle L \rangle$ is cross-directional.

3.5 The Treatment of Start, Stop and Culmination

In our temporal readings of natural language sentences great importance is given to three special types of time point: a state's start point, stop point and culmination point. Here we define what we mean by a state and illustrate each of the three point types in terms of the HSF1 formulae that hold at them.

Definition 4. An HSF1 state is a formula of the form $[D](P)$ where P may be any atomic formula. The state derived from an atomic proposition p is on at any interval/point at which $\langle \underline{D} \rangle([D](p))$ holds and off at any interval/point at which $\langle \underline{D} \rangle([D](\neg p))$ holds.

A start point of a state is a point of transition from off to on. This means that $[D](\neg p)$ must hold at an interval adjoining the point on the left, and $[D](p)$ must hold at an interval adjoining the point on the right.

Definition 5. A start point of a state $[D](p)$ is any point at which the formula $\langle \underline{A} \rangle([D](\neg p)) \wedge \langle A \rangle([D](p))$ holds.

| a | b |

Index
a - $[D](\neg p)$ b - $[D](p)$

Fig. 5. Start point of a state

A stop point of a state is a point of transition from on to off. This means that $[D](p)$ must hold at an interval adjoining the stop point on the left, and $[D](\neg p)$ must hold at an interval adjoining the stop point on the right.

Definition 6. A stop point of a state $[D](p)$ is any point at which the formula $\langle \underline{A} \rangle([D](p)) \wedge \langle A \rangle([D](\neg p))$ holds.

| a | b |

Index
a - $[D](p)$ b - $[D](\neg p)$

Fig. 6. Stop point of a state

A culmination point of a state is a point at which the state is said to be *finished* or *completed*. Culmination points are a critical factor in our understanding of accomplishment expressions such as *Max walked to the station* or *Jill ate the apple* where the progression of an event is *directed* towards a particular goal whose status is relevant to our perception of proceedings. In the case of activity expressions such as *Max worked* or *Fred played the piano* a directed progression is not apparent. We define a culmination of a state as a stop point at which the state's atomic proposition is true.

Definition 7. A culmination point of a state $[D](p)$ is any stop point of $[D](p)$ at which p holds.

$| \quad a \quad |_b \quad c \quad |$

Index
a - $[D](p)$ b - p c - $[D](\neg p)$

Fig. 7. Culmination point of a state

If we know that a state definitely did not culminate (i.e. it was broken off) we may represent this as a *break point.*

Definition 8. A break point of a state $[D](p)$ is any stop point of $[D](p)$ at which $\neg p$ holds.

$| \quad a \quad |_b \quad c \quad |$

Index
a - $[D](p)$ b - $\neg p$ c - $[D](\neg p)$

Fig. 8. Break point of a state

3.6 Temporal Readings as HSF1 Formulae

In this subsection we show how formulae from the HSF1 fragment may be used to represent the temporal readings of the sentences given in Fig. 2.

HSF1 formula:
$\langle \underline{L} \rangle ([D](\neg run(max)) \wedge \langle \underline{A} \rangle ([D](run(max)) \wedge \langle \underline{A} \rangle ([D](\neg run(max)))))$

Timeline

```
| a | b | c |   | * |
```

Timeline Index
a - $[D](\neg run(max))$
b - $[D](run(max))$
c - $[D](\neg run(max))$
* - **now**

Fig. 9. A temporal reading of the sentence *Max stopped running*

In the above reading of *Max stopped running* one sees a timeline similar to the informal example given in the introduction of this paper, but now expressed by an HSF1 formula. Looking back into the past from **now** we see an interval c where the state of Max running is off at all sub-intervals following the point at which Max stops running. Adjoining c on the left is the interval b for which the state is on at all sub-intervals, preceded by another interval a for which the state is off prior to his starting to run.

HSF1 formula:
$\langle \underline{L} \rangle ([D](\neg eat(jill,\ apple)) \wedge \langle \underline{A} \rangle ([[EP]](eat(jill,\ apple)) \wedge [D](eat(jill,\ apple)) \wedge duration(minutes(5)) \wedge \langle \underline{A} \rangle ([D](\neg eat(jill,\ apple)))))$

Timeline

```
| a | b |c d |   | * |
```

Timeline Index
a - $[D](\neg eat(jill,\ apple))$
b - $[D](eat(jill,\ apple))$
    $duration(minutes(5))$
c - $eat(jill,\ apple)$
d - $[D](\neg eat(jill,\ apple))$
* - **now**

Fig. 10. A temporal reading of the sentence *Jill ate the apple in five minutes*

In the above reading of *Jill ate the apple in five minutes*, we can see two things of fresh interest. The basic timeline structure is similar to that of *Max stopped running* in Fig. 9 but there is a culmination point at the end of the state for which Jill is eating an apple (point c) which indicates that Jill has finished eating the apple, and a duration proposition holding at interval b, which is our interpretation of the adverbial phrase *in five minutes* in its durational sense. Hence the proposition *duration(minutes(5))* holds only at the outermost interval of the eating state.

HSF1 formula:
$\langle \underline{D} \rangle([D](eat(jill,\ apple)) \wedge \langle A \rangle([D](\neg eat(jill,\ apple))) \wedge \langle \underline{A} \rangle([D](\neg eat(jill,\ apple))))$

Timeline

```
                    |   a   |         b          |   c   |
                                  |   *   |
```

Timeline Index
a - $[D](\neg eat(jill,\ apple))$ * - **now**
b - $[D](eat(jill,\ apple))$ c - $[D](\neg eat(jill,\ apple))$

Fig. 11. A temporal reading of the sentence *Jill is eating an apple*

The temporal reading of the present progressive sentence *Jill is eating an apple* takes a form where the eating state includes the current interval. Present progressive readings are the same for activity verb phrases such as *run*. The distinction in behaviours occurs is that where one can entail *Max has run* from *Max is running* or *Max stopped running*, one may not entail *Jill has eaten the apple* from *Jill is eating an apple* or *Jill stopped eating an apple*. In the past tense and perfective usage of accomplishment verb phrases such as *eat an apple* there is a definite sense of having culminated the eating event. In contrast activity verb phrases such as *run* which involve generic undirected states do not necessarily culminate in perfective and past tense readings.

HSF1 formula:
$\langle \underline{L} \rangle([D](race(mark)) \wedge [[BP]](clock(hour(2))) \wedge \langle \underline{A} \rangle([D](\neg race(mark))))$

Timeline

```
                    |   a   |b  c   |      |   *   |
```

Timeline Index
a - $[D](\neg race(mark))$ c - $[D](race(mark))$
b - $clock(hour(2))$ * - **now**

Fig. 12. A temporal reading of the sentence *Mark started racing at 2 o'clock*

In our last example we show a reading of a sentence involving an at-adverbial phrase. We read the use of *at* with a chronological complement as signifying that a proposition representing the clock time, *2 o'clock* in this case, holds at a particular point, in this case the point at which Mark started racing. The index b appears as a subscript because the proposition it indexes holds at a point rather than a durative interval.

4 An Entailment Checker for HSF1 formulae

An entailment *F1* $\models$ *F2* where *F1* and *F2* are HSF1 formulae holds if and only if for each timeline that may be *constructed* for *F1* a timeline may be *constructed* for *F2* such that the latter is a *substructure* of the former.

4.1 The Substructure Fitting Algorithm

The Substructure Fitting Algorithm establishes whether a timeline is a substructure of another timeline by attempting to *fit* the former into the latter. A successful fit means that a new timeline which represents the match of the two timelines can be built. We will use the terms *substructure timeline*, *superstructure timeline* and *matched timeline* to distinguish between the operative timelines in this algorithm. Note that for an entailment *F1* $\models$ *F2* a superstructure timeline is a model of *F1* and a substructure timeline is a model of *F2*.

The principle of the algorithm is to attempt to build a new timeline (the matched timeline) which has a set of bindings that pairs each binding of the substructure timeline with a binding from the superstructure timeline such that the resultant ordering is consistent with the orderings of the substructure timeline and superstructure timeline. At each stage of the recursive algorithm, a binding (SubBd) is removed from the set of substructure bindings (SubBds) and an attempt is made to match its formula (SubBdF) with a formula (SupBdF) of some binding (SupBd) from the set of superstructure bindings (SupBds) such that the formulae match in such a way that the new ordering (MatchOrd) can be updated to remain consistent with the superstructure ordering (SupOrd) and substructure ordering (SubOrd). This is repeated until either SubBds has been emptied (success) or it has been established that SubBds cannot possibly be emptied by the algorithm (failure).

The formulae in a pair of bindings may be matched in any of the ways depicted in Fig. 13.

Exact match, atomic case

$[\ \ prop\ \ ]$
$[\ \ prop\ \ ]$

Exact match, homogeneous case

$[\ [D](prop)\ \ ]$
$[\ [D](prop)\ \ ]$

Right homogeneous match

$[\qquad [D](prop)\ \ ]$
$[\ [D](prop)\ \ ]$

Left homogeneous match

$[\ [D](prop)\qquad]$
$[\ [D](prop)\ \ ]$

Internal match, homogeneous case

$[\qquad [D](prop)\qquad]$
$[\ [D](prop)\ \ ]$

Internal match, atomic case

$[\ [D](prop)\ \ ]$
$[\ \ prop\ \ ]$

Fig. 13. How formulae in a pair of bindings may be matched

4.2 Entailment Examples

Entailment check:

$$\langle \underline{D} \rangle ([D](run(max)) \wedge$$
$$\quad \langle A \rangle ([D](\neg run(max))) \wedge$$
$$\quad \langle \underline{A} \rangle ([D](\neg run(max))) \models$$
$$\langle \underline{L} \rangle ([D](run(max)) \wedge$$
$$\quad \langle \underline{A} \rangle ([D](\neg run(max)))$$

——————–

Superstructure (case 1 of 1):

```
|   a   |         b         |   c   |
               |   *   |
```

Timeline Index

a - $[D](\neg run(max))$
b - $[D](run(max))$
* - **now**
c - $[D](\neg run(max))$

——————–

Substructure (case 1 of 1):

```
|   d   |   e   |       |   *   |
```

Timeline Index

d - $[D](\neg run(max))$
e - $[D](run(max))$
* - **now**

——————–

Match:

```
|  a/d  |            b             |   c   |
        |   e   |       |   *   |
```

Fig. 14. Entailment check for: *Max is running* $\Rightarrow$ *Max has run*

Entailment check:

$\langle \underline{L} \rangle([D](\neg eat(jill, apple)) \wedge$
$\quad \langle \underline{A} \rangle([D](eat(jill, apple)) \wedge$
$\quad\quad \langle \underline{A} \rangle([D](\neg eat(jill, apple))))) \models$
$\langle \underline{L} \rangle([D](\neg eat(jill, apple)) \wedge$
$\quad \langle \underline{A} \rangle([[EP]](eat(jill, apple)) \wedge$
$\quad\quad [D](eat(jill, apple)) \wedge$
$\quad\quad \langle \underline{A} \rangle([D](\neg eat(jill, apple)))))$

——————————-

Superstructure (case 1 of 1):

| a | b | c | | * |

Timeline Index

a - $[D](\neg eat(jill, apple))$
b - $[D](eat(jill, apple))$
c - $[D](\neg eat(jill, apple))$
* - **now**

——————————-

Substructure (case 1 of 1):

| d | e $|_f$ g | | * |

Timeline Index

d - $[D](\neg eat(jill, apple))$
e - $[D](eat(jill, apple))$
f - $eat(jill, apple)$
g - $[D](\neg eat(jill, apple))$
* - **now**

——————————-

Match:
None.

Fig. 15. Entailment check for: *Jill stopped eating an apple* $\not\Rightarrow$ *Jill ate the apple*

5 Conclusion

We believe that the contribution of this paper is in demonstrating a route to simple and precise temporal representations of natural language events. It is pleasing to explicate prior formal analyses using interval semantics by utilising an expressive interval logic, interesting to find that only a tractable fragment seems to be needed, and satisfying to discover that timelines are sufficient to model the semantics of the fragment's formulae.

It is not evident to us that natural language should require excessively complex temporal models. AI approaches to understanding natural language, in our view, properly place emphasis on representing complex propositions and coping with the dynamic effect of actions. Nevertheless, the problem of temporal representation arises both in the local understanding and generation of elegant sentences and in the wider problems of scenario understanding and planning. A forthcoming thesis will discuss work on the understanding of narrative texts by analysing them into timelines using a process of refinement and consistency checking. Rather than a monolithic integrated solution, the need to mixing different logics and representations may be important for human and machine understanding of natural language.

References

1. James Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154, 1984.
2. James Allen and George Ferguson. Actions and events in interval temporal logic. Technical Report 521, University of Rochester, 1994.
3. Joseph Halpern and Yoav Shoham. A propositional modal logic of time intervals. In *Proceedings of Symposium on Logic in Computer Science*, pages 279–292, Cambridge, Massachusetts, June 16-18, 1986.
4. Stuart Kent. *Modelling Events from Natural Language.* PhD thesis, Imperial College, London, 1993.
5. Alex Lascarides. *A Formal Semantic Theory of the Progressive.* PhD thesis, University of Edinburgh, 1988.
6. Marc Moens. *Tense, Aspect and Temporal Reference.* PhD thesis, University of Edinburgh, 1987.
7. Marc Moens and Mark Steedman. Temporal ontology and temporal reference. *Computational Linguistics 14*, pages 15–28, 1988.
8. Yde Venema. Expressiveness and completeness of an interval tense logic. *Notre Dame Journal of Formal Logic*, 31(4):529–547, 1990.
9. Henk Verkuyl. *A theory of aspectuality.* Cambridge University Press, 1993.