Aspect and Interval Tense Logic *

Miguel Leith† (mfl@doc.ic.ac.uk) and Jim Cunningham (rjc@doc.ic.ac.uk)
Department of Computing, Imperial College of Science, Technology and Medicine

Abstract. Linguistic phenomena of tense and aspect have been investigated in a great deal of theoretical work in linguistics, philosophy and computer science. Modern tense logics, established by Prior, are part of this effort. Point tense logics offer an intuitive representation of tense but lack the expressiveness to represent many aspectual structures. Interval tense logics offer more expressiveness but in the general case can be computationally intractable. From a linguistic perspective there is the problem of precisely how to formalise the aspectual structures, such as a culmination and a culminated process. In this paper we define a computationally tractable augmented fragment of Halpern and Shoham's interval tense logic HS and apply it to represent a core set of aspectual structures, which are incorporated into a temporal semantics of a simple fragment of English. We model the logic fragment using timelines and define two procedures, one for constructing the minimal timelines that satisfy a formula and one for checking semantic entailments between one formula and another by comparing their timelines. The former is applied to compute models of temporal readings and the latter to check entailments between them. Possible extensions to the logic fragment and timeline models are discussed as ways of accounting for a wider range of linguistic behaviour.

1. Introduction

In natural language we convey temporal information through a variety of linguistic categories, such as the simple tenses (past, present and future), the progressive, the perfect, adverbial phrases, conjunctions and determiners in noun phrases. By formalising theories of tense and aspect in terms of logics one may gain insight into how humans may understand, organise and generate the temporal information present in sentences and narratives. Modern tense logics, established by Prior in the 1960s (see [8]), are motivated by the study of these phenomena. Prior held that logic should be related as closely as possible to intuitions embodied in everyday discourse. Point tense logics have succeeded in this respect with regard to tense, but lack the expressiveness to represent the range of aspectual structures required in a wider theory. Interval logics are more expressive but in the general case can be com-

* We wish to thank Stephen Sommerville, Robin Cooper and the anonymous reviewer of an earlier version of this paper for their comments.
† This paper includes Ph.D work which was supported by the Esprit 6283 GOAL project and Imperial College access funds.

putationally intractable. As well as the question of tractability there are more linguistic questions regarding how to represent a core set of aspectual structures that includes the culmination and culminated process.

In this paper we develop representations and models of aspectual structures starting from related work in tense and aspect (Section 2) and Halpern and Shoham's interval tense logic HS (see [1], also summarised here in Section 3). We first define a fragment of HS called HSF1, describe how to model it using timelines and show how to automatically construct the timelines the satisfy a formula (Section 3). We then augment the fragment with a *focus* operator to create a new language HSF1A, and apply HSF1A to represent a number of aspectual structures, the perfect and simple tenses (Section 4). Following this we demonstrate a straightforward link between our representations and natural language via a temporal semantics for a fragment of English, and in Section 5 we demonstrate how linguistic entailments can be checked automatically by comparing the timeline models of the temporal readings of sentences. Section 6 concludes the paper.

2. Related Work on Tense and Aspect

In this section we give brief summaries of the work of seven researchers that is of particular significance to ours: Reichenbach (1947, [11]), Prior (1967, [8]), Vendler (1967, [14]), Kamp (1968, [2]), Moens (1987, [5]), Lascarides (1988, [4]), Kent (1993, [3]) and Verkuyl (1993, [16]).

2.1. Reichenbach

Reichenbach's characterisation of tense in *Elements of Symbolic Logic*, [11], has had lasting influence on later theories. In it he argues that there are three times underlying tense, rather than two, namely the *speech point* (S), *event point* (E) and *reference point* (R). The reference point, denoting the time being referred to in the utterance, is Reichenbach's main innovation. The other points (or equivalent concepts) had been around for some time. In Reichenbach's theory, the event point and reference point coincide in the case of the simple tenses (past, present and future) whereas in uses of the perfect such as the past perfect and the present perfect, the event point and reference point are distinct (see Figure 1). Later work has shown how Reichenbach's theory can be extended to account for a wider range of linguistic behaviour than he originally investigated (see [13] for a summary).

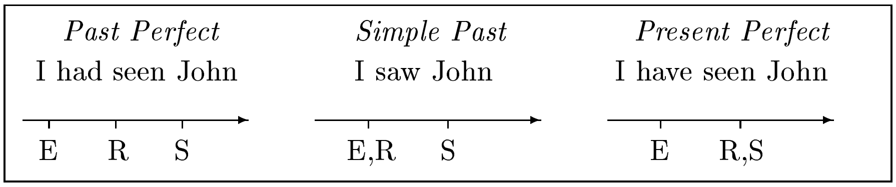

Figure 1. Past vs perfect according to Reichenbach

2.2. Prior and Kamp

A.N. Prior is widely regarded as the founder of modern tense logic. Prior's work on tense logic revives and continues on from ideas of ancient and medieval logicians concerning time, in which two of the fundamental assumptions were that tense distinctions are a proper subject of logical reflection, and what is true at one time is in many cases false at another time, and vice versa. In his major work *Past, Present and Future*, [8], Prior discusses four tense operators F (at some time in the future), G (is always going to be), P (at some time in the past) and H (has always been) and gives various axiomatizations. Prior also published work on the semantical function of the word *now* in [9]. Prior held that logic should be related as closely as possible to intuitions embodied in everyday discourse, and his tense logic can indeed account for a large number of linguistic inferences. For a discussion of Prior's work see [7]. Later work by Kamp added more expressiveness to tense logic with *since* and *until* operators (see [2]).

2.3. Vendler

Vendler proposed a categorisation of verbs in terms of the four aspectual classes that they can denote: States (e.g. *love*, *know*, *possess*, *rule*), Activities (e.g. *run*, *swim*, *walk*), Accomplishments (e.g. *run a mile*, *draw a circle*, *eat a sandwich*) and Achievements (e.g. *win a race*, *recognise him*, *reach the top*). He distinguished between the verb categories by contrasting their linguistic behaviour under various circumstances. For example, States and Achievements were contrasted with Activities and Accomplishments by citing sentences involving the progressive, with which only Activities and Accomplishments were claimed to be compatible, and Activities were contrasted with Accomplishments by illustrating the differences in the ways they may be combined with durational *in* and *for* adverbial phrases.

Vendler's work has been criticised because a number of counter-examples exist to the generalisations of his classes, however it is still very influential. For example Moens in [5] refines Vendler's theory in various ways, one of which is to allow aspectual class mobility, in the sense that an event may be *coerced* from one aspectual class to another under certain restrictions.

2.4. Moens

Like Vendler, Moens addresses issues of aspectual classification, but whereas Vendler classifies verbs, Moens classifies the basic forms of whole sentences. For example the sentence *Max was running* has a basic form *Max run*, where the meaning of *Max run* is classified as a Process $MaxRun$. The meaning of a sentence is arrived at by applying linguistic temporal operators to the meaning of the basic form. For example the temporal semantics of a sentence *Max was running* would be analysed as a semantic expression:

$$\text{past}(\text{progressive}(\text{process}(MaxRun)))$$

Moens defines five aspectual classes: State, Process, Culminated Process, Point and Culmination. Verkuyl in [16] illustrates the correspondence of four of these classes to Vendler's classes, as shown in Figure 2. Events may be *coerced* from one class to another by rules described

Moens	*Vendler*
State	State
Process	Activity
Culminated Process	Accomplishment
Point	–
Culmination	Achievement

Figure 2. The correspondence between Moens' and Vendler's aspectual classes

in Moens' *aspectual network*. For example, by iterating the meaning of a Point such as *Sandra hiccup*, one may coerce it into a Process, for example in the sentence *Sandra was hiccupping.*

Many of the permissible transitions in Moens' aspectual network are related to an event structure called a *nucleus*, which comprises a culmination, an associated preparatory process and a consequent state (see Figure 3). The formal representation of the nucleus structure and

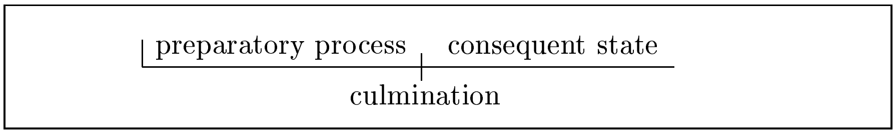

Figure 3. The *nucleus* event structure

its aspectual classes the Process, Culminated Process and Culmination is the main link between Moens' work and ours.

2.5. Lascarides

Lascarides in [4] gives a formal semantic treatment of the progressive and temporal adverbials in English, which uses the homogeneous[1] first order interval logic IQ (see [12]). Parts of the theory are based on the ideas of Moens. As well as improving on previous semantic analyses, Lascarides work demonstrates the adequacy of an interval logic in the representation of a number of aspectual classes. However her use of one (instantaneous) aspectual class for both accomplishment and achievement expressions, in contrast to two separate classes (the culminated process and culmination proposed by Moens) has led to criticism that important distinctions in linguistic behaviour are lost (see [3]).

2.6. Kent

Kent in [3] formalises a theory of tense and aspect that is related to the work of Moens and the later work of Lascarides. He characterises linguistic operations of tense and aspect as operators in an interval language derived from the logic IQ (see [12]). Kent differs from Lascarides by treating separately the aspectual classes *culminated process* and *culmination*. He also gives a new definition of processes, as non-homogeneous, and formalises a further aspectual class, the repeated culminated process. Kent argues that branching time is necessary to account for the behaviour of culminated processes, which introduces extra complexity into his semantics. Later we provide a representation which does not require branching time.

2.7. Verkuyl

In [16] Verkuyl presents a theory of aspectuality containing an account of the way terminative aspect (the expression of boundedness) can be compositionally formed on the basis of semantic information expressed by different syntactic elements, in particular the verb and its arguments. He uses an ontological classification of eventualities, *State*, *Process* and *Event* which are distinguished by the features [$\pm$ADD TO] (a temporal feature whose value is determined in verbs) and [$\pm$SQA] (an atemporal feature whose value is determined in noun phrases).

The value of the feature [$\pm$ADD TO] for a verb is determined by whether the verb denotes a sense of addition of some quantity from

[1] A proposition is homogeneous if it is true on all sub-intervals of any interval at which it is true.

one moment to the next. The feature therefore has the positive value [+ADD TO] for verbs such as *eat* (addition of consumed matter), *knit* (addition of constructed material), *walk* (addition of distance), but the negative value [-ADD TO] for verbs like *want*, *hate* and *seem*.

The acronym SQA in the feature [±SQA] stands for Specified Quantity. The feature has a positive value for noun phrases that specify a clear delimitation of quantity and a negative value for those which do not. For instance, the noun phrases *sandwiches* and *some porridge* do not have a clear delimitation of quantity, and will therefore have the feature value [-SQA], whereas the noun phrases *three sandwiches* or *an apple* do, and will therefore have the feature value [+SQA].

Verkuyl argues that *terminative* aspect, the aspect of Events, is determined by the combination of [+ADD TO] and [+SQA] in a sentence (e.g. *Judith ate a sandwich*), whereas otherwise the event is *durative* (e.g. in *Judith ate sandwiches* or *Nobody wanted a sandwich*). His theory includes a compositional account for the formation of terminative aspect.

3. A Fragment of HS Logic

Lascarides and Kent have both formalised aspectual classes in interval logics, building on the work of Moens. They differ over the formal properties of the class *process* and the classes involving culmination, where Lascarides uses one class *moment* and Kent distinguishes between *culminated process* and *culmination*. In this paper we keep a distinction between culminated processes (extended) and culminations (punctual) but unlike Kent we model their aspectual structure in linear time. Our definitions of these and other aspectual structures in Section 4 make use of Reichenbach's *reference time*, which also allows a distinction between the perfect and the simple tenses. The interval tense logic we apply as the basis for our representation is HS, defined by Halpern and Shoham in [1] and further investigated by Venema in [15]. HS expresses interval relations as a set of six basic modal operators from which other interval relations may be derived.

The validity problem for HS is of high complexity, being undecidable in all but the simplest classes of frames. This does not prove to be a problem however, because we find that our expressive requirements are met by a small fragment of HS.

We begin the section with a summary of HS and then define our fragment, called HSF1, and show how to model it using *timelines*, which consist of a total ordering of time point variables and a set of assignments between intervals and atomic propositions. A procedure for constructing the timelines that satisfy a formula is defined, and a series of examples is given. Lastly we define a type of timeline called a Closest Position Timeline, which we apply in Section 4, and outline how operators from point tense logic may be represented in HS.

3.1. HS Logic

HS is the interval tense logic of Halpern and Shoham, which was defined in [1] and further investigated by Venema in [15]. In HS the set of possible worlds is the set of closed intervals in a temporal frame which may be dense or discrete, and linear or branching. Its six basic modal operators ($\langle B\rangle$, $\langle \underline{B}\rangle$, $\langle E\rangle$, $\langle \underline{E}\rangle$, $\langle A\rangle$, $\langle \underline{A}\rangle$), and eight derived modal operators ($\langle O\rangle$, $\langle \underline{O}\rangle$, $\langle L\rangle$, $\langle \underline{L}\rangle$, $\langle D\rangle$ and $\langle \underline{D}\rangle$), $[[EP]]$and $[[BP]]$) express a set of relative positional interval relationships, illustrated in Figure 4.

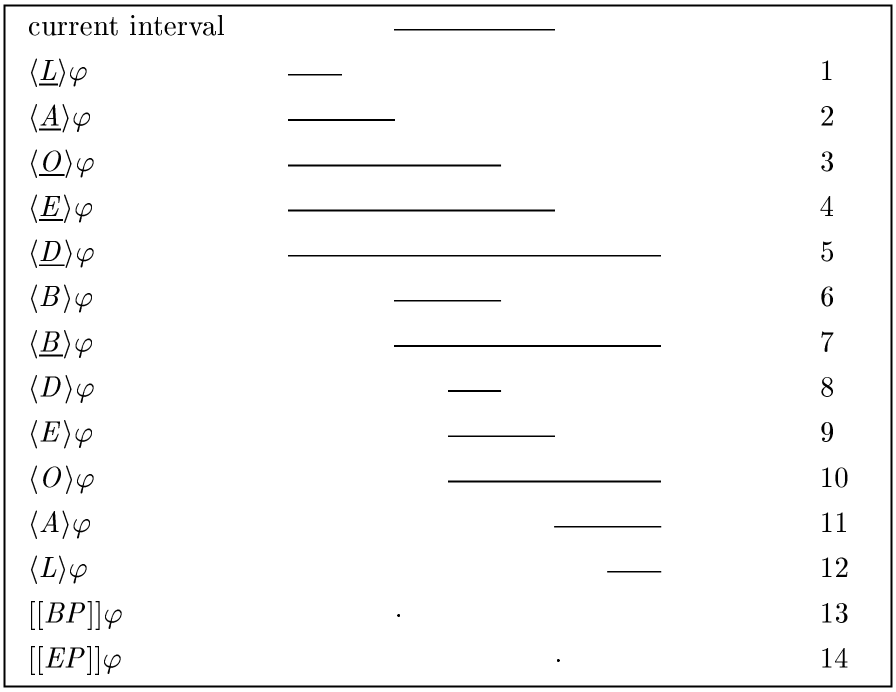

Figure 4. Pictorial illustration of relative interval positions and their HS formulae

3.1.1. *Syntax of HS*

Given Φ_0 as the set of atomic propositions, made up of the propositional constants $p_0, p_1, \ldots$, 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.

3.1.2. *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 $[s,t] = \{x \in T \mid s \leq x \leq t\}$ 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 $[s,t] \in V(P)$
$\not\models^M_{[s,t]} false$
$\models^M_{[s,t]} (\varphi_1 \rightarrow \varphi_2)$ 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 $\models^M_{[u,s]} \varphi$

A wff φ is said to be $satisfiable$ relative to a set of temporal structures A if, in some interpretation $\langle\langle T,<\rangle, V\rangle$ such that $\langle T,<\rangle \in A$, we have $(s,t) \models \varphi$ for some $s \in T$ and $t \in T$, where $s \leq t$. A wff is satisfiable relative to a temporal structure if it is satisfiable relative to the singleton consisting of that structure. φ is valid relative to A if $\neg\varphi$ is not satisfiable relative to A.

3.1.3. *A Subset of HS Operators*

It has been shown by Halpern and Shoham in [1] 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 [15], 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 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 [3] as a property of State propositions and the progressive of Processes. 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 $[D](MaxRun)$ hold at an interval we are saying that Max does a bit of running at all intervals/points during that interval. The application of the $[D]$ operator in defining aspectual structure is discussed in Section 4.

3.2. The HSF1 Fragment

We begin this subsection with a definition of the fragment HSF1, followed by an informal analysis of the fragment and a definition of how to model its formulae as *timelines*. We then describe a procedure that can construct the timelines of an HSF1 formula and give a full example, followed by a definition of an interesting type of timeline called a Closest Position Timeline. In Section 4 we extend HSF1 and use the new language to formalise a number of aspectual structures.

3.2.1. *Syntax of HSF1*

HSF1 ::= Exp2 | PossOp "(" HSF1 ")"

PossOp ::= $\langle D\rangle \mid \langle \underline{D}\rangle \mid \langle E\rangle \mid \langle \underline{E}\rangle \mid \langle B\rangle \mid \langle \underline{B}\rangle \mid \langle L\rangle \mid \langle \underline{L}\rangle \mid$
$\langle A\rangle \mid \langle \underline{A}\rangle \mid \langle O\rangle \mid \langle \underline{O}\rangle \mid [[EP]] \mid [[BP]]$

Exp2 ::= ExtendedAtomic | HSF1 "$\wedge$" HSF1

ExtendedAtomic ::= Atomic | $[D]$ "(" Atomic ")"

Atomic ::= PropConstant | "$\neg$" PropConstant

PropConstant ::= letter {letter | digit | "(" | ")"}

Figure 5. HSF1 Syntax

Figure 5 defines the syntax of HSF1 using grammar rules. The fragment contains all of the HS possibility operators, shown in Figure 4, plus one of the necessitation operators, $[D]$. The operators are restricted to appear only in their positive form i.e. they may never be preceded by the operator $\neg$. For example, no HSF1 formula may contain a sub-formula such as $\neg\langle\underline{A}\rangle(p)$ or $\neg[[EP]](p)$. The $\neg$ operator is permitted only in the case of atomic propositions e.g. $\neg p$. The homogeneity operator, $[D]$, is restricted to appear only before atomic formulae, resulting in a class of formulae we term *extended atomic* e.g. $[D](p)$, $[D](\neg p)$. The class of extended atomic formulae includes the class of atomic formulae.

DEFINITION 1. *An* **outermost extended atomic formula** *in an HSF1 formula F is any occurrence of an extended atomic formula in F except for any atomic formula p within $[D](p)$ or any atomic formula $\neg p$ within $[D](\neg p)$.*

By the recursion in the syntax definition of HSF1, possibility operators may be nested to any depth.

3.2.2. *An Informal Analysis of HSF1*

The motivation for HSF1 is primarily linguistic in the spirit of Priorian point tense logics. The HSF1 syntax was constructed with the intention of specifying arrangements of atomic propositions in time, with homogeneity playing an important part. At a given interval an atomic proposition has a truth value *true* or *false*. Additionally, the presence of $[D]$ operator allows us to add the concepts of *on* and *off*. An atomic proposition p is said to be *on* at any interval during an interval at which $[D](p)$ holds and *off* at any interval during an interval at which $[D](\neg p)$ holds. A discussion of the propositional arrangements that we apply in a linguistic context is provided in Section 4.

HSF1 does not allow $\rightarrow$ to occur either explicitly or by reduction to its equivalent form $\neg F_1 \vee F_2$. To see why, consider two of the restrictions in the HSF1 fragment: there is no disjunction operator $\vee$, and the $\neg$ operator is constrained to apply only to atomic propositions. Since there is no explicit disjunction, one cannot directly write $\neg F_1 \vee F_2$, and since the $\neg$ operator is only permitted at the atomic level, there can be no formula of the form $\neg(F_1 \wedge \neg F_2)$, which would re-write to $\neg F_1 \vee F_2$. Therefore implication may not arise in an HSF1 formula. This raises the question of how expressive HSF1 is, or, more specifically: what range of temporal situations can an HSF1 formula describe? To answer this, we categorise the types of component that comprise an HSF1 formula and analyse how they relate to each other. We split the components of HSF1 into three categories: possibility operators, extended atomic formulae and conjunctions. An explanation of how we arrive at this categorisation follows.

HSF1 restricts necessitation operators considerably. Firstly the only necessitation operator that can arise is $[D]$, which is explicitly incorporated. Necessitation cannot arise by re-writing a formula $\neg\langle X\rangle(F)$ where $\langle X\rangle$ is some possibility operator because the restriction on the $\neg$

operator prevents such formulae occurring. Secondly, $[D]$ is restricted to precede only atomic formulae, where the resulting form (e.g. $[D](p)$) means that the atomic formula p holds at all sub-intervals of an interval at which $[D](p)$ holds. Since the only way for necessitation to occur in HSF1 is as a homogeneous period of some atomic proposition, we can make a clear distinction between possibility operators (which express relative positions between intervals) and extended atomic formulae (which express that an atomic proposition holds at or throughout an interval). The other type of component in HSF1 is a conjunction, which allows more than one formula to hold at an interval.

Having classified the components of an HSF1 formula, we are in a position to informally identify the range of temporal situations that may be specified by HSF1. Since a possibility operator specifies an interval's position relative to the current interval and an extended atomic formula specifies the atomic value (or values) that hold at (or throughout) the current interval, we infer that an HSF1 formula specifies temporal arrangements of extended atomic formulae within some time frame. In the next subsection we define how HSF1 formulae are modelled as timelines. Figure 6 provides an illustrative example of an HSF1 formula and a timeline model that satisfies it, where $*$ is the current interval.

HSF1 formula

$$\langle \underline{L} \rangle (p_1) \wedge \langle L \rangle ([D](p_2) \wedge \langle O \rangle (\neg p_3))$$

Timeline model

$$| \quad p_1 \quad | \qquad\qquad | \quad * \quad | \qquad\qquad | \; [D](p_2) |$$
$$| \quad \neg p_3 \quad |$$

Figure 6. An illustration of an HSF1 formula and a timeline model which satisfies it.

3.2.3. *Modelling of HSF1 Formulae*

3.2.3.1. *Timelines* We use time frames for which the set of time points is dense and unbounded, and ordered linearly. We refer to an arrangement of extended atomic formulae along a dense unbounded line of time as a *timeline*, and if it is consistent (see below) we refer to it as a *timeline model.* Since a timeline is a semantic structure, it contains explicit representations of intervals and their relative positions, in contrast to the implicit relations in HSF1. We represent an interval as a pair of point variables (s, t). A pair of point variables is enough to uniquely specify any closed interval for the type of time frame we are working with.

DEFINITION 2. *A* **timeline** *is a triple* $\langle CI, ORD, IM\rangle$ *where CI is its current interval (i.e. now), ORD is an ordering and IM is an interval mapping.*

DEFINITION 3. *A timeline's* **ordering** *is a total ordering from earliest to latest of a finite set of point variables over dense, unbounded time.*

DEFINITION 4. *A timeline's* **interval mapping** *is a finite set of bindings.*

DEFINITION 5. *A* **binding** *within a timeline's interval mapping is a pairing of an interval with a finite set of extended atomic formulae. More than one extended atomic formula may hold at an interval. A binding takes the form* $((t_x,\ t_y) : \{\ F_1,\ F_2\ \dots\ F_n\ \})$ *where each each* F_x *is an extended atomic formula, the set of formulae need not be ordered, and the point variable* t_x *is before or equal to* t_y *in the timeline's ordering ORD.*

DEFINITION 6. *An* **association** *in a timeline is one element of a binding from its interval mapping i.e. a pairing of an interval with a single extended atomic formula. An association therefore takes the form* $((t_x,\ t_y) : F)$ *where F is an extended atomic formula and the point variable* t_x *is before or equal to* t_y *in the timeline's ordering ORD.*

DEFINITION 7. *Two associations are* **inconsistent** *with each other if they take any of the forms depicted in Figure 7 (we omit the converses of the last four cases, in which the occurrences of p and* $\neg p$ *are exchanged). A timeline is* **consistent** *(i.e. a timeline model) if and only if none of its associations are inconsistent with each other.*

DEFINITION 8. *A* **sub-association** *of an association is one which follows from the properties of the* $[D]$ *operator as depicted in Figure 8, where for each pair of associations the sub-association is depicted below the other.*

3.2.3.2. *Relating Timelines to HS Models* For any timeline model $(CI,\ ORD,\ IM)$, the components ORD and IM are together a *timeline representation* of an HS model $((T, <), V)$ where T is dense and unbounded and $<$ is a total ordering, if the following conditions hold: (a)

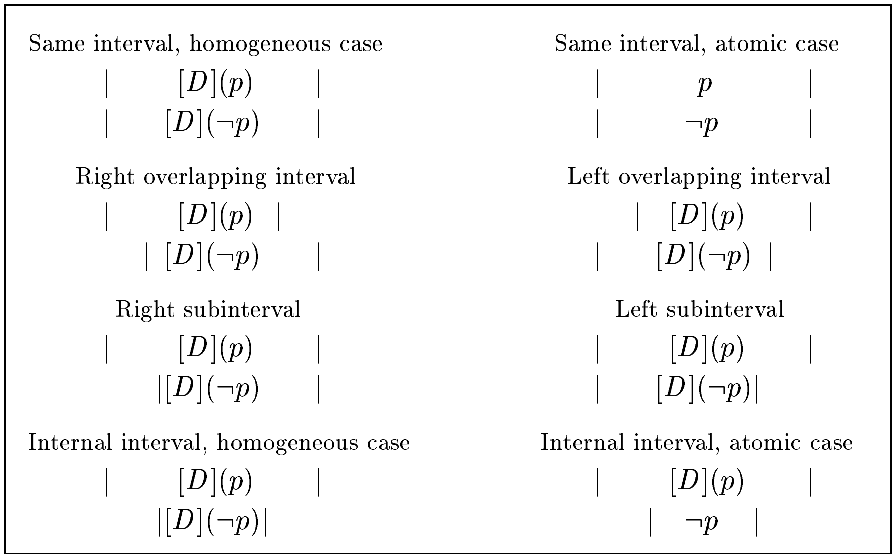

Figure 7. Association inconsistencies

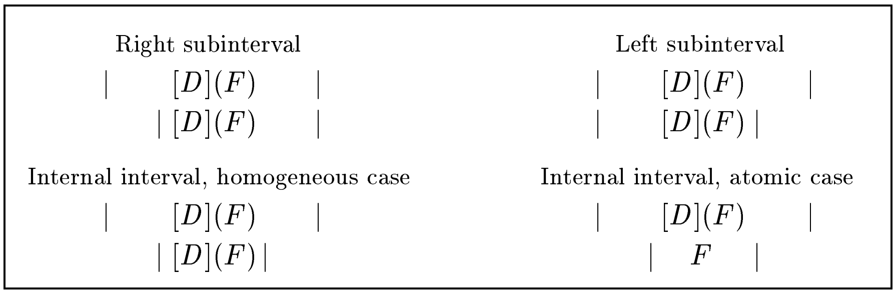

Figure 8. The four types of sub-association.

the domain of every point variable in the timeline's ordering ORD is T, (b) for any two point variables t_x and t_y, if t_x is before t_y in ORD, $t_x < t_y$, (c) every interval in the timeline's interval mapping IM is a closed interval from $INT((T,<))$, (d) every atomic proposition in an association in IM is a member of Φ_0, the set of atomic propositions, and (e) an association $((t_x, t_y) : F)$ occurs in IM if and only if $\models^{((T,<),V)}_{[t_x,t_y]} F$.

An HSF1 formula F is true in a timeline $((t_x, t_y), ORD, IM)$ if and only if there is an HS model M such that $\models^{M}_{[t_x,t_y]} F$ and ORD and IM are together a timeline representation of M.

Following the relationships just defined, we propose that the truth relation for an HSF1 formula with respect to timeline models may be represented by the rules below. The lists indexed by Roman numerals are conjunctions of conditions. We require that a formula $\neg\neg F$ is always re-written as F. A phrase of the form "there is a s" refers to the existence of a point variable in a timeline model's ordering. The symbols $<$ and $\leq$ constrain the ordering of the timeline model, for example $t_x < t_y$ constrains ORD such that the point variable t_x is before t_y. Consistency of a timeline is maintained in rules 1 (II-III) and 2 (III-V). Sub-associations are considered in the rules 1 (I) and 2 (II).

1 An atomic formula F is true in a timeline $\langle (s,t), ORD, IM \rangle$ iff

  I the association $((s,\, t) : F)$ occurs in IM **or** there is a u, v such that $u < s$ and $t < v$ and the association $((u,\, v) : [D](F))$ occurs in IM

  II the association $((s,\, t) : \neg F)$ does not occur in IM

  III there is no u, v such that $u < s$ and $t < v$ and the association $((u,\, v) : [D](\neg F))$ occurs in IM

2 $[D](F)$ is true in a timeline $\langle (s,t), ORD, IM \rangle$ iff

  I F is atomic

  II the association $((s,\, t) : [D](F))$ occurs in IM **or** there is a u such that $u < s$ and the association $((u,\, t) : [D](F))$ occurs in IM **or** there is a u, v such that $u < s$ and $t < v$ and the association $((u,\, v) : [D](F))$ occurs in IM **or** there is a v such that $s < v$ and the association $((s,\, v) : [D](F))$ occurs in IM.

  III there is no u, v such that $s < u < v < t$ and either the association $((u,\, v) : [D](\neg F))$ or the association $((u,\, v) : \neg F)$ occurs in IM

  IV there is no u, v such that $u < t$ and $t \leq v$ and the association $((u,\, v) : [D](\neg F))$ occurs in IM

  V there is no u, v such that $u \leq s$ and $s < v$ and the association $((u,\, v) : [D](\neg F))$ occurs in IM

3 $F_1 \wedge F_2$ is true in a timeline $\langle (s,t), ORD, IM \rangle$ iff F_1 is true in a timeline $\langle (s,t), ORD, IM \rangle$ and F_2 is true in a timeline $\langle (s,t), ORD, IM \rangle$

4 $\langle B \rangle (F)$ is true in a timeline $\langle (s,t), ORD, IM \rangle$ if there is a u such that $s \leq u$ and $u < t$ and F is true in a timeline $\langle (s,u), ORD, IM \rangle$

5 $\langle \underline{B} \rangle (F)$ is true in a timeline $\langle (s,t), ORD, IM \rangle$ iff there is a u such that $t < u$ and F is true in a timeline $\langle (s,u), ORD, IM \rangle$

6 $\langle E \rangle (F)$ is true in a timeline $\langle (s,t), ORD, IM \rangle$ iff there is a u such that $s < u$ and $u \leq t$ and F is true in a timeline $\langle (u,t), ORD, IM \rangle$

7 $\langle \underline{E} \rangle(F)$ is true in a timeline $\langle (s,t), ORD, IM \rangle$ iff there is a u such that $u < s$ and F is true in a timeline $\langle (u,t), ORD, IM \rangle$

8 $\langle A \rangle(F)$ is true in a timeline $\langle (s,t), ORD, IM \rangle$ iff there is a u such that $t < u$ and F is true in a timeline $\langle (t,u), ORD, IM \rangle$

9 $\langle \underline{A} \rangle(F)$ is true in a timeline $\langle (s,t), ORD, IM \rangle$ iff there is a u such that $u < s$ and F is true in a timeline $\langle (u,s), ORD, IM \rangle$

10 $\langle O \rangle(F)$ is true in a timeline $\langle (s,t), ORD, IM \rangle$ iff $\langle E \rangle \langle \underline{B} \rangle(F)$ is true in a timeline $\langle (s,t), ORD, IM \rangle$

11 $\langle \underline{O} \rangle(F)$ is true in a timeline $\langle (s,t), ORD, IM \rangle$ iff $\langle B \rangle \langle \underline{E} \rangle(F)$ is true in a timeline $\langle (s,t), ORD, IM \rangle$

12 $\langle L \rangle(F)$ is true in a timeline $\langle (s,t), ORD, IM \rangle$ iff $\langle A \rangle \langle A \rangle(F)$ is true in a timeline $\langle (s,t), ORD, IM \rangle$

13 $\langle \underline{L} \rangle(F)$ is true in a timeline $\langle (s,t), ORD, IM \rangle$ iff $\langle \underline{A} \rangle \langle \underline{A} \rangle(F)$ is true in a timeline $\langle (s,t), ORD, IM \rangle$

14 $\langle D \rangle(F)$ is true in a timeline $\langle (s,t), ORD, IM \rangle$ iff $\langle B \rangle \langle E \rangle(F)$ is true in a timeline $\langle (s,t), ORD, IM \rangle$

15 $\langle \underline{D} \rangle(F)$ is true in a timeline $\langle (s,t), ORD, IM \rangle$ iff $\langle \underline{B} \rangle \langle \underline{E} \rangle(F)$ is true in a timeline $\langle (s,t), ORD, IM \rangle$

16 $[[EP]](F)$ is true in a timeline $\langle (s,t), ORD, IM \rangle$ iff F is true in a timeline $\langle (t,t), ORD, IM \rangle$

17 $[[BP]](F)$ is true in a timeline $\langle (s,t), ORD, IM \rangle$ iff F is true in a timeline $\langle (s,s), ORD, IM \rangle$

3.2.4. *Timeline Construction*

DEFINITION 9. *A* **minimal timeline model** *that satisfies an HSF1 formula F is a timeline model which satisfies F such that no timeline resulting from the removal of any of the associations from its interval mapping or the replacement any of the associations in its interval mapping by a sub-association satisfies F.*

We argue below that there is a procedure for determining all the minimal timelines that satisfy an HSF1 formula.

3.2.4.1. *A Procedure for Minimal Timeline Construction* A procedure *construct_timeline* takes the following arguments: an HSF1 formula F, a current interval CI of the form $(t_x,\, t_y)$, an ordering ORD

containing a finite number of point variables and at least t_x and t_y such that $t_x \leq t_y$ in the ordering, and an interval mapping IM containing a finite number of associations (possibly none). A successful call of the procedure *construct_timeline*(F, CI, ORD, IM) returns a solution pair (ORD', IM') such that every association in IM is either present in IM' or is a sub-association of an association in IM', IM' contains a finite number (possibly zero) of associations that are not present in IM, ORD' contains the point variables of ORD in the same order with the possible inclusion of a finite number of other point variables, and the timeline (CI, ORD', IM') satisfies F. The procedure is guaranteed to construct only the *minimal* timelines that satisfy F if and only if IM is empty, otherwise it is possible that IM' in any solution pair (ORD', IM') may contain associations that prevent the timeline (CI, ORD', IM') from minimally satisfying F. It is guaranteed to terminate if and only if there is a finite number of associations in IM and a finite number of point variables in ORD.

The procedure uses the HSF1 truth relation rules for each syntactic case of F, recursively reducing to a sub-formula (or sub-formulae in the case of a conjunction). Alternative solution pairs are possible due to the possibility of alternative positions of a new point variable in an ordering in the cases for rules 4 to 9.

We examine the procedure on a case by case basis for each rule, looking first at rule 1, where F is an atomic formula, i.e. F is either p or $\neg p$. If the association $((t_x, t_y)\colon F)$ is a sub-association of an association in IM, we need not do anything, and simply return (ORD, IM). Otherwise, checks are made as to whether the association is consistent with all of the associations in IM, by applying the consistency conditions from 1 (II) and 1 (III). If it is consistent, the association is added to IM, to yield IM', and (ORD, IM') is returned, otherwise this call to the procedure has failed.

The case for rule 2, where the formula F has the form $[D](F')$, is similar, except that there are different sub-association and consistency conditions. Additionally, if there are any sub-associations of $((t_x, t_y)\colon [D](F'))$ in IM', they are removed because they are redundant, to yield IM'', and (ORD, IM'') is returned.

In the case for rule 3 we have a formula $F_1 \wedge F_2$. First we call *construct_timeline*(F_1, CI, ORD, IM) to obtain a result (ORD', IM'), then we call *construct_timeline*(F_2, CI, ORD', IM') to obtain a result (ORD'', IM''), which is returned.

In the cases for rules 4 to 9, there is a declaration of the existence of a new point variable and an ordering constraint is introduced. By way of example we consider rule 4, for the operator $\langle B \rangle$, which means *at a beginning interval of.* We have a formula F which is $\langle B \rangle(F')$, a current interval CI which is a pair of point variables (t_x, t_y), and an ordering ORD in which there may be zero or more point variables between t_x and t_y. The operator $\langle B \rangle$ declares the existence of a point variable, call it t_z, such that $t_x \leq t_z < t_y$. If t_x is t_y, there is no successful result to this call, because there can be no points in between those of the current interval. Otherwise, there are alternative positions where t_z may be exist in the ordering. Firstly it can be t_x. Secondly it can be any existing point variable in between t_x and t_y. Thirdly it can be a new point variable inserted into a position in the ordering between t_x and t_y. In general, given an ordering containing m point variables, there are at most $2m + 1$ alternative positions where a point variable t_z may occur, and $m + 1$ of these result in the insertion a new point variable to create a new ordering. An interpretation of the operator $\langle B \rangle$ adds constraints that t_z may not be before t_x and must be before t_y, which means that for an ordering containing m point variables, there would be at most $2m - 2$ alternative positions for t_z, hence there would be at most $2m - 2$ alternative new current intervals. A total of $m - 1$ of these alternatives would result in a changed ordering due to the insertion of a new point variable. Given the maximum number of alternative current intervals, there would be at most $2m - 2$ alternative calls to $construct\_timeline(F', CI', ORD', IM)$ when interpreting $\langle B \rangle$, where CI' and ORD' are together the result of one of the possible interpretations of $\langle B \rangle$ relative to CI and ORD.

For rules 5 to 9 the principle is much the same as for rule 4, the only variation being in the ordering constraint that determines the legal positions where the new point variable may occur.

Rules 10 to 15 simply rewrite the top level HS operator of the formula as a nested pair of HS operators from rules 4 to 9. Therefore *construct_timeline* in these cases can simply rewrite the formula F as F' and return the result of $construct\_timeline(F', CI, ORD, IM)$.

Rules 16 and 17, for $[[BP]](F')$ and $[[EP]](F')$, are similar to rules 4 to 9 in that they determine a new current interval, but neither operator can give rise to a new point variable. The procedure can in these cases simply create a new current interval CI' and return the result of $construct\_timeline(F', CI', ORD, IM)$.

3.2.4.2. *Remarks on Complexity* In order to construct a single timeline model that satisfies an HSF1 formula F containing n outermost extended atomic formulae and m HS operators, starting with a timeline that contains k associations in its interval mapping, the construction procedure outlined above will: (a) add at most $2m$ point variables to ORD (not counting those present initially) (b) add at most n associations to IM (not counting those present initially), (c) carry out at most

$\sum_{i=k}^{n+k-1} i$ checks to determine whether a possible new association is consistent with an existing one, (d) carry out at most $\sum_{i=k}^{n+k-1} i$ checks to determine whether a possible new association is a sub-association of existing one, and (e) carry out at most $\sum_{i=k}^{n+k-1} i$ checks to determine whether an existing association is a sub-association of a newly added association.

To show (a), note the construction procedure cases for rules 10 to 15, where there are at most two new point variables added to ORD for any encountered HS operator, for rules 4 to 9, where there is at most one new point variable added, and for rules 16 to 17, where there are no new point variables added.

To show (b), note the construction procedure cases for rules 1 and 2, where there is at most one new association added to IM for any outermost extended atomic formula.

To show (c), consider the construction procedure cases for rules 1 and 2. Starting with the first outermost extended atomic formula interpreted in F, when there are k associations in the timeline's interval mapping, the maximum number of association consistency checks to be made is k, one per existing association. For the second, assuming that a new association was added, there are at most $(k+1)$ checks to be made, for the third at most $(k+2)$, continuing until the interpretation of the last of a total of n outermost extended atomic formulae contained in F, where the maximum number of checks to be made is $(n+k-1)$. The sum of these all is $\sum_{i=k}^{n+k-1} i$.

The arguments for (d) and (e) are similar to that for (c), because whenever an outermost extended atomic formula in F is interpreted, the construction procedure cases for rules 1 and 2 indicate that there are checks to determine whether the possible new association would be a sub-association of each existing one, and if not, the construction procedure case for rule 2 indicates that there are checks to determine whether each existing association would be a sub-association of the new one.

3.2.4.3. *Example of Timeline Construction* There follows an example of timeline construction for the formula:

$$\langle \underline{L} \rangle ([[EP]](\langle \underline{A} \rangle ([D](p)) \wedge \langle A \rangle ([D](\neg p))))$$

proceeding as described over Figures 9 to 16. Figure 9 shows the initial timeline of the procedure. An ordering is represented as a sequence of point variables marked by a vertical line | with a variable name immediately below. An asterisk *, positioned between two point variables, marks the initial current interval (i.e. the current interval of a timeline of the whole formula F). The current interval of a timeline of a formula that is currently under analysis (a sub-formula F' of F) is marked by a curved line joining the interval's left bound with its right bound. The current interval of a timeline of the immediate super-formula of F' (the previous interval) is marked with a dotted curved line in a similar way (initially there is no previous interval). In Figure 10 the $\langle \underline{L} \rangle$ operator,

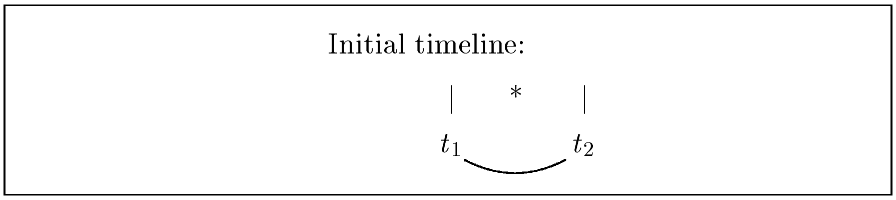

Figure 9. Timeline construction example, part 1

which means *earlier than*, is analysed. The previous interval (t_1, t_2), marked by a dotted curve, is the current interval of Figure 9. Two new point variables are inserted into the ordering in positions earlier than (t_1, t_2), and become the left and right bounds of the new current interval, (t_4, t_3). In Figure 11 the $[[EP]]$ operator, which means *end*

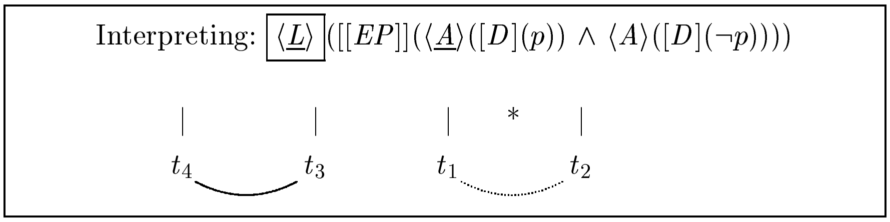

Figure 10. Timeline construction example, part 2

point of, is analysed. The end point variable of the previous interval, t_3, becomes both the left and right bound of the new current interval, (t_3, t_3). The ordering does not have to be changed as no new point variables have to be added. In Figure 12 the $\langle \underline{A} \rangle$ operator, which means

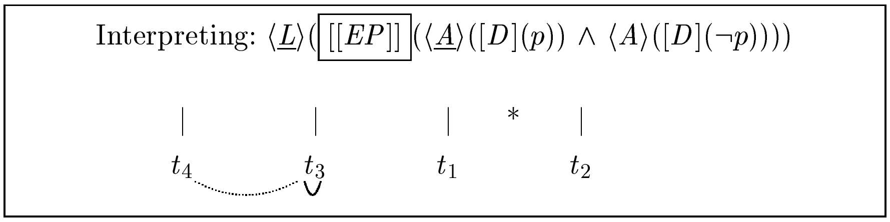

Figure 11. Timeline construction example, part 3

adjoins at the beginning of, is analysed. The beginning point variable of the previous interval, t_3, becomes the right bound of the new current interval, but there are three alternative positions where the left bound of the new current interval may go, as there is an existing point variable before t_3. We pick the first alternative from Figure 12 as the timeline

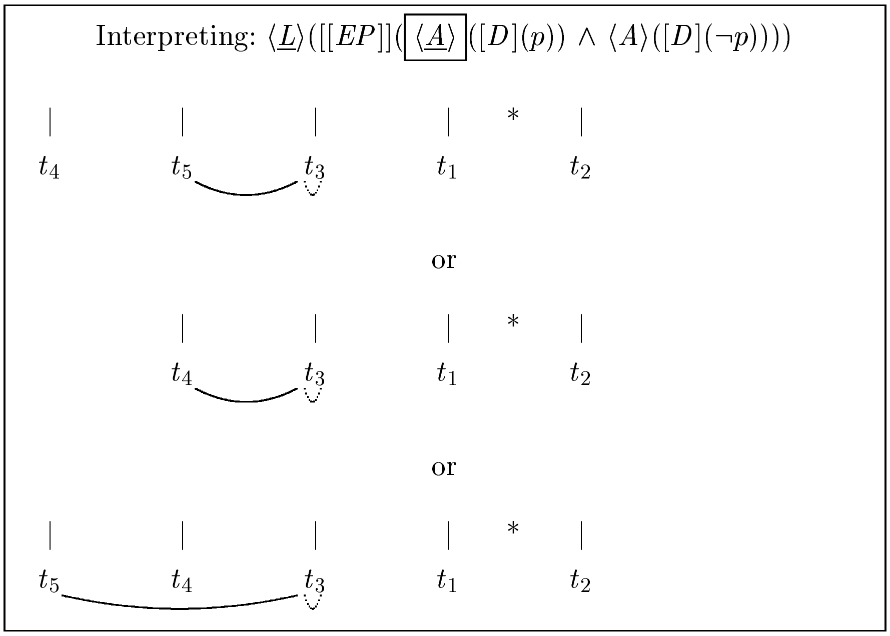

Figure 12. Timeline construction example, part 4

from which to continue the construction example from, noting that the full example would fan out into alternative timelines at this juncture. In Figure 13 the extended atomic formula $[D](p)$ is analysed. The interval mapping is updated, represented in the diagram by the formula $[D](p)$ being positioned at the current interval, (t_5, t_3). In Figure 14 the $\langle A \rangle$

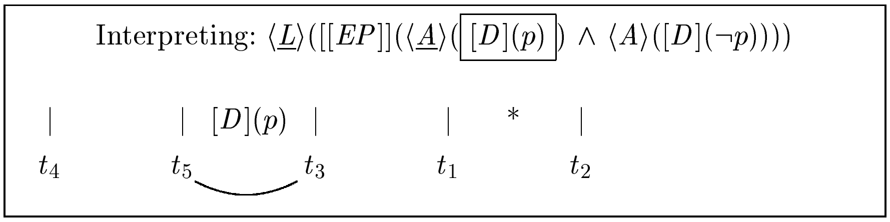

Figure 13. Timeline construction example, part 5

operator, which means *adjoins at the end point of*, is analysed. The end point variable of the previous interval, t_3, becomes the left bound of the new current interval, but there are five alternative positions where the right bound of the new current interval may go, as there are two existing point variables after t_3. Again we pick the first alternative from Figure 14 as the timeline from which to continue the construction ex-

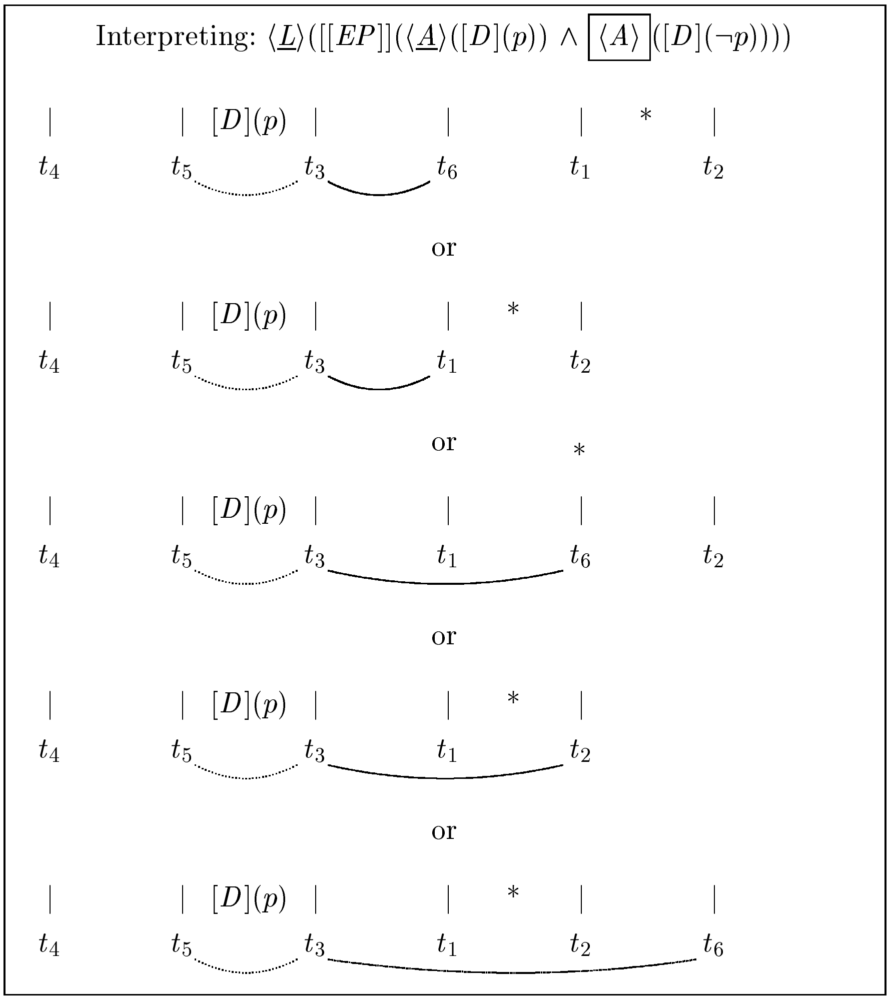

Figure 14. Timeline construction example, part 6

ample from, noting that the full example would fan out into alternative timelines at this juncture. In Figure 15 the extended atomic formula $[D](\neg p)$ is analysed. The interval mapping is updated, represented in the diagram by the formula $[D](\neg p)$ being positioned at the current interval, (t_3, t_6). Finally, in Figure 16 we remove any superfluous point variables from the timeline. By superfluous point variables we mean any point variables which are not part of the Start Interval or Interval Mapping. The only point variable that is superfluous here is t_4.

Interpreting: $\langle \underline{L} \rangle([[EP]](\langle \underline{A} \rangle([D](p)) \wedge \langle A \rangle(\boxed{[D](\neg p)})))$

$|$	$|\ [D](p)\ |\ [D](\neg p)\,|$	$|\quad * \quad|$
t_4	$t_5 \qquad t_3 \qquad t_6$	$t_1 \qquad t_2$

Figure 15. Timeline construction example, part 7

Removing superfluous point variable(s): t_4

$|\ [D](p)\ |\ [D](\neg p)\,|$	$|\quad * \quad|$
$t_5 \qquad t_3 \qquad t_6$	$t_1 \qquad t_2$

Figure 16. Timeline construction example, part 8

3.2.5. *Single-timeline HSF1 Formulae*

Generally an HSF1 formula may have more than one timeline that satisfies it, as we saw in the construction example. This poses the question: in what situations do alternative timelines arise? The answer is that certain pairs of HSF1 possibility operators result in alternative timelines whenever one operator appears at a level beneath the other in a formula.

We call pairs of HSF1 operators that in behave this way *cross-directional*. The two basic cross-directional pairs are: $(\langle B \rangle, \langle \underline{B} \rangle)$ and $(\langle E \rangle, \langle \underline{E} \rangle)$. For example whenever an HSF1 formula contains $\langle B \rangle$ at a level beneath $\langle \underline{B} \rangle$ or vice-versa, interpreting the formula will give rise to alternative timelines. Venema used the term *mirror images* in [15] in reference to these pairs, because they are reflections of each other in a northwest plane correspondence.

It is possible to derive all other HS possibility operators using $\langle B \rangle$, $\langle \underline{B} \rangle$, $\langle E \rangle$, $\langle \underline{E} \rangle$, the operators $\neg$ and $\wedge$, and the propositions $false$ and $true$ (Venema has shown this in [15]). Further cross-directional pairs become evident by re-writing the derived operators in their original form. For example since $[[EP]](\varphi)$ can be re-written as $((\varphi \wedge [B]false) \vee (\langle E \rangle(\varphi \wedge [B]false))$, the operator pair $([[EP]], \langle \underline{E} \rangle)$ is cross-directional, deriving from the pair $(\langle E \rangle, \langle \underline{E} \rangle)$. In the construction example, the cross-directional pairs are: $([[EP]], \langle \underline{A} \rangle)$ and $(\langle \underline{L} \rangle, \langle A \rangle)$.

3.2.6. *Closest Position Timelines*

As we saw in the construction example, each time a possibility operator is interpreted the current interval is changed by inserting or selecting point variables for the new current interval in appropriate positions in the timeline ordering.

Whenever a point variable is to be selected or inserted before/after some point variable t in an ordering, there may be one or more positions that it can go with respect to other point variables. We refer to the position immediately before/after t as the *closest position*, i.e. the position at which there are no intervening point variables (there are always intervening *points* in dense time, but here we are referring to point variables in the timeline ordering). If there are no point variables before/after t, the only available position is the closest position. We define a Closest Position Timeline (CPT) as a timeline constructed using only the closest position throughout.

In the construction example, in both cases where alternatives arose in Figures 12 and 14, we chose the closest position. In Figure 17 we illustrate all of the timelines that can model the formula from the construction example in Section 3.2.4.3, of which there are five, where the CPT is listed first. In the general case there are HSF1 formulae

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

1 (CPT) $\quad \mid [D](p) \mid [D](\neg p) \mid \qquad \mid * \mid$

2 $\quad \mid [D](p) \mid [D](\neg p) \mid * \mid$

3 $\quad \mid [D](p) \mid \qquad \mid * \mid$
$\quad \mid [D](\neg p) \mid$

4 $\quad \mid [D](p) \mid \qquad \mid * \mid$
$\quad \mid [D](\neg p) \mid$

5 $\quad \mid [D](p) \mid \qquad \mid * \mid$
$\quad \mid [D](\neg p) \mid$

Figure 17. Example of a CPT among alternative timelines of an HSF1 formula

for which the CPT would be inconsistent, e.g. $[D](\neg p) \wedge \langle B \rangle(\langle L \rangle(p))$. However, any HSF1 formula which has no inconsistent timelines must have a CPT that satisfies it, because a CPT always exists for a formula (there is always a closest position before and after a point variable). Since the temporal readings that arise from the sentence semantics in Section 4 do not give rise to any timelines which are inconsistent, each one has a CPT which satisfies it (no inconsistencies are possible because whenever $[D](p)$, $[D](\neg p)$ and p appear they hold at different, non-overlapping intervals).

CPTs separate future, past, and present in preference to alternatives with overlapping intervals. They are used in the semantics given in Section 4. In Figure 17 the CPT represents a transition of a proposition from positive homogeneity to negative homogeneity (referred to later as a *stop point*). Generally entailments can be demonstrated without restricting the class of timelines, for example it follows from all of the timelines in Figure 17 that $[D](p)$ holds in the past, that $[D](\neg p)$ holds later in the past, and that the two periods adjoin each other. However a CPT is useful because it provides a standard way of determining a single timeline that can act as a default and allow an entailment check between two formulae to be carried out without having to consider all possible timelines of the left formula. Entailment checks are discussed in Section 5.

3.3. Representation of Since and Until

We note at this point that HSF1 can express the point tense logic operators S (since) and U (until) (see [2]) with the limitation that they may be applied only to atomic formulae or conjunctions of atomic formulae.

The formula aSb, which means that the proposition b holds at a point in the past and the proposition a holds at all intervening points, can be expressed in HSF1 as the formula $\langle \underline{A} \rangle([D](a) \wedge [[BP]](b))$.

Similarly, the formula aUb, which means that the proposition b holds at a point in the future and the proposition a holds at all intervening points, can be expressed in HSF1 as the formula $\langle A \rangle([D](a) \wedge [[EP]](b))$.

3.4. Representation of Henceforth and Has Always Been

In order to represent notions of *henceforth* (at all intervals after the current interval) and *has always been* (at all intervals before the current interval) for atomic formulae there would have to be additions to the fragment. Using HS operators, these concepts could be expressed as $[A]([D](F))$ and $[\underline{A}]([D](F))$ respectively, where F is p or $\neg p$ for some atomic proposition p. These could be extensions of the definition of *extended atomic formulae*. In order to represent such a formula in an association of a timeline's interval mappping, timelines would have to be extended with two new point variable representations, one for future infinity and one for past infinity.

4. Temporal Sentence Semantics

Having defined and investigated the HSF1 fragment, we next apply it in the representation of tense and aspect in natural language. We begin with two passages of groundwork. Firstly we define *on* and *off* for atomic propositions and the transitions *start*, *stop* and *culminate*, which we compare to the *nucleus* structure used by Moens. Secondly we examine how to represent Reichenbach's *reference time* by augmenting the HSF1 fragment with a new operator *focus*, resulting in a new language HSF1A. These definitions and additions are then used in the analysis of a fragment of English. We define a grammar for the English fragment, followed by a semantics for the fragment that enables the derivation of temporal readings of sentences as HSF1A formulae. We then derive some example temporal readings and illustrate the timelines that model them. Finally we outline how an account for the linguistic behaviour of an extended English fragment may be provided via extensions to the logic fragment and timeline models.

4.1. The Treatment of Start, Stop and Culmination

In our temporal readings of natural language sentences importance is given to three types of time point: a start point, stop point and culmination point of an atomic proposition. In this subsection we define each of the three point types in terms of the HSF1 formulae that hold at them and illustrate the principles with diagrams. To make this possible we first require a formal definition of *on* and *off*.

DEFINITION 10. *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. It is neither on nor off at other intervals. It is inconsistent for an atomic proposition to be both on and off at an interval.*

$\vert$	$[D](p)$	$\vert$
(	$\langle \underline{D} \rangle([D](p))$	)

Figure 18. An atomic proposition when it is *on*

Kent provides definitions of *start* and *stop* in [3]. The HSF1 definitions of *start* and *stop* below agree with Kent's convention that they are points of change from being *off* to *on* and vice-versa respectively.

A start point of an atomic proposition is a point of transition from it being *off* to being *on*. This means that $[D](\neg p)$ must hold at an

|	$[D](\neg p)$	|
(	$\langle \underline{D} \rangle([D](\neg p))$	)

Figure 19. An atomic proposition when it is *off*

interval adjoining the point on the left, and $[D](p)$ must hold at an interval adjoining the point on the right.

DEFINITION 11. *A* **start point** *of an atomic proposition p is a point at which the formula* $\langle \underline{A} \rangle([D](\neg p)) \wedge \langle A \rangle([D](p))$ *holds.*

|	$[D](\neg p)$	|	$[D](p)$	|

Figure 20. Start point of an atomic proposition, the transition from *off* to *on*

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

DEFINITION 12. *A* **stop point** *of an atomic proposition p is a point at which the formula* $\langle \underline{A} \rangle([D](p)) \wedge \langle A \rangle([D](\neg p))$ *holds.*

|	$[D](p)$	|	$[D](\neg p)$	|

Figure 21. Stop point of an atomic proposition, the transition from *on* to *off*

A culmination point of an atomic proposition is a point at which it is said to be *finished* or *completed*. Culmination points are critical to our understanding of linguistic expressions such as *Jack finished eating dinner* in contrast to *Jack stopped eating dinner*. An issue for some time in event representation has been the question of what a proposition's truth value is at a stop or start point. We define a culmination point of an atomic proposition as a stop point at which the atomic proposition is true. Formally, this is defined below.

DEFINITION 13. *A* **culmination point** *of an atomic proposition p is any stop point of p at which p holds.*

By comparing the above definition of a culmination point with the concept of a *nucleus* which was pictured in Figure 3, one may see a close similarity in temporal structure. The notable difference is that we

$$| \quad [D](p) \quad |_p \; [D](\neg p)|$$

Figure 22. Culmination point of an atomic proposition

have concentrated on the notion of generic *finishing* whereas a nucleus contains an additional notion of a *consequent state.*

4.2. Focus

Reichenbach's *reference time*, summarised in Section 2, is a point (or interval) of reference in a temporal reading of a sentence, independent of event time and speech time. It is a specification of the point (or interval) in time which is being focused on in an utterance.

Recalling the earlier definitions of timeline models of HSF1 formulae in Section 3, we note that a timeline can be presented as a time diagram in a way that is similar to Reichenbach's time diagrams e.g. in Figure 1. Given that reference time is in a sense a record of an interval and HSF1 is a fragment of interval logic, it is feasible to embed a concept similar to an interval reference within a timeline. We add a new operator to the HSF1 syntax, called *focus* in recognition of the focal nature of reference time, and call the new language HSF1A. There may be zero or more

HSF1A ::= Exp2 | PossOp "(" HSF1A ")" |
focus "(" HSF1A ")"

PossOp ::= $\langle D\rangle \mid \langle \underline{D}\rangle \mid \langle E\rangle \mid \langle \underline{E}\rangle \mid \langle B\rangle \mid \langle \underline{B}\rangle \mid \langle L\rangle \mid \langle \underline{L}\rangle \mid$
$\langle A\rangle \mid \langle \underline{A}\rangle \mid \langle O\rangle \mid \langle \underline{O}\rangle \mid [[EP]] \mid [[BP]]$

Exp2 ::= ExtendedAtomic | HSF1A "$\wedge$" HSF1A

ExtendedAtomic ::= Atomic | $[D]$ "(" Atomic ")"

Atomic ::= PropConstant | "$\neg$" PropConstant

PropConstant ::= letter {letter | digit | "(" | ")"}

Figure 23. HSF1A Syntax

occurrences of the *focus* operator in an HSF1A formula. We define a focused HSF1A formula as one which contains one or more occurrences of the *focus* operator, and define an unfocused HSF1A formula as one which contains no occurrences of the *focus* operator (an unfocused HSF1A formula is an HSF1 formula).

Next we define a new type of timeline that can contain a *focus interval.*

DEFINITION 14. *A* **focused timeline** *is a quadruple* $\langle CI, FI, ORD, IM\rangle$ *where* CI *is its current interval,* FI *is its focus interval,* ORD *is an ordering and* IM *is an interval mapping.*

We refer to the timelines defined in Section 3 as *unfocused* timelines. To give an informal overview, a focused HSF1A formula may only be true in focused timelines, and an unfocused HSF1A formula (i.e. an HSF1 formula) may only be true in unfocused timelines.

The truth relation for HSF1A formulae with respect to focused timelines is defined in the rules below. Some of the rules refer to the truth relation rules of sub-formulae with respect to *unfocused* timelines, for which we refer the reader to Section 3. An interesting case in the truth relation occurs when there is a conjunction of two focused formulae. If a formula F has the form $F_1 \wedge F_2$ and both F_1 and F_2 are true in focused timelines with different foci, generally both foci would be equally valid as the focus in a timeline satisfying F. Therefore in such a case F is considered true in two alternative focused timelines. Extended atomic formulae contain no focus operator so they are only true in unfocused timelines, hence they do not appear in the rules below.

1. focus(F) is true in a focused timeline $\langle (s,t),(s,t), ORD, IM\rangle$ iff either F is true in a focused timeline $\langle (s,t), FI, ORD, IM\rangle$ where FI is some interval or F is true in an unfocused timeline $\langle (s,t), ORD, IM\rangle$

2. $F_1 \wedge F_2$ is true in a focused timeline $\langle (s,t), FI, ORD, IM\rangle$ iff any of the following five statements hold:

   a) F_1 is true in a focused timeline $\langle (s,t), FI, ORD, IM\rangle$ and F_2 is true in an unfocused timeline $\langle (s,t), ORD, IM\rangle$
   b) F_1 is true in an unfocused timeline $\langle (s,t), ORD, IM\rangle$ and F_2 is true in an focused timeline $\langle (s,t), FI, ORD, IM\rangle$
   c) F_1 is true in a focused timeline $\langle (s,t), FI, ORD, IM\rangle$ and F_2 is true in a focused timeline $\langle (s,t), FI, ORD, IM\rangle$
   d) F_1 is true in a focused timeline $\langle (s,t), FI', ORD, IM\rangle$ and F_2 is true in a focused timeline $\langle (s,t), FI, ORD, IM\rangle$ such that FI is not the same interval as FI'
   e) F_1 is true in a focused timeline $\langle (s,t), FI, ORD, IM\rangle$ and F_2 is true in a focused timeline $\langle (s,t), FI', ORD, IM\rangle$ such that FI is not the same interval as FI'

3. $\langle B\rangle(F)$ is true in a focused timeline $\langle (s,t), FI, ORD, IM\rangle$ iff there is a u such that $s \leq u$ and $u < t$ and F is true in a focused timeline $\langle (s,u), FI, ORD, IM\rangle$.

The focused timeline rules for $\langle \underline{B} \rangle$, $\langle E \rangle$, $\langle \underline{E} \rangle$, $\langle A \rangle$, $\langle \underline{A} \rangle$, $\langle O \rangle$, $\langle \underline{O} \rangle$, $\langle L \rangle$, $\langle \underline{L} \rangle$, $\langle D \rangle$, $\langle \underline{D} \rangle$, $[[EP]]$ and $[[BP]]$ correspond to those for unfocused timelines except for the presence of FI as shown in the rule for $\langle B \rangle$ above.

Figure 24 shows an example of a focus operator being interpreted during timeline construction. It is what would have been part 4 of the timeline construction example in Section 3 if the formula under analysis had been:

$$\langle \underline{L} \rangle([[EP]](focus(\langle \underline{A} \rangle([D](p)) \wedge \langle A \rangle([D](\neg p)))))$$

The current interval is marked with the focus sign, $+$, signifying that the Focus Interval element of the timeline has become the current interval, (t_3, t_3).

Interpreting: $\langle \underline{L} \rangle([[EP]](\boxed{focus}(\langle \underline{A} \rangle([D](p)) \wedge \langle A \rangle([D](\neg p)))))$

$$\begin{array}{ccccc} | & |_{+} & | & * & | \\ t_4 & \underset{\vee}{t_3} & t_1 & & t_2 \end{array}$$

Figure 24. Timeline construction example, revised part 4

We have demonstrated that the *focus* operator, corresponding to reference time, is a straightforward addition to the syntax of HSF1 and the HSF1 truth relation rules. Later the *focus* operator is incorporated in the temporal readings of sentences from a fragment of English.

4.3. Obtaining Temporal Readings

4.3.1. *Grammar for an English Fragment*

The fragment of English under analysis here includes simple declarative clauses exhibiting the past and present simple tenses, the modal future using *will*, the perfect, the progressive and the verbs *start*, *stop* and *finish*. A context free grammar for the fragment is shown in Figure 25. The grammatical category MOD stands for modal verb (which includes *will* in the fragment under analysis), AUX for auxiliary verb (includes *have* and *be*), TV for transition verb (includes *start*, *stop* and *finish*), V for main verb (includes *run*, *run to*, *play*, *draw*, *eat* and *win*), DET for determiner (includes *the*, *a* and *an*). The category $NOUN$ includes peoples' names and common nouns such as *apple*, *dinner*, *piano*, *picture*, *race* and *station*. Some example sentences from the fragment are listed in Figure 26.

$$
\begin{array}{l}
S \leftarrow NP\ VP\ NP \\
S \leftarrow NP\ VP \\
NP \leftarrow NOUN \\
NP \leftarrow DET\ NOUN \\
VP \leftarrow MOD\ AUX\ AUX\ V \\
VP \leftarrow MOD\ AUX\ TV\ V \\
VP \leftarrow MOD\ AUX\ V \\
VP \leftarrow MOD\ TV\ V \\
VP \leftarrow AUX\ AUX\ V \\
VP \leftarrow AUX\ TV\ V \\
VP \leftarrow MOD\ V \\
VP \leftarrow AUX\ V \\
VP \leftarrow V
\end{array}
$$

Figure 25. Context free grammar for the English fragment under analysis

Max is running
Max stopped running
Jack will be eating dinner
Jill ate an apple
Jane had finished drawing the picture
Mark will have won the race
Fred will play the piano
Mike played tennis

Figure 26. Example sentences in the English fragment

4.3.2. *Temporal Semantics for an English Fragment*

This section defines semantic rules to work out the temporal readings of sentences from the grammar defined above as HSF1A formulae. Linguistic features are shown as subscripts of grammatical categories. The feature *vf* stands for *verb form*, and has the possible values *inf* (infinitive, e.g. "eat"), *prp* (present participle e.g. "eating"), *pap* (past participle e.g. "eaten"), *pas* (past simple e.g. "ate"), *prs* (present simple e.g. "eats"). Numerical subscripts are only used to distinguish two instances of a category, for example two auxiliary verbs in a verb phrase. The temporal semantics of syntactic categories and individual words is denoted by enclosing them within the double brackets "[[" and "]]". Note that double brackets are also used in a different way for the HS operators $[[BP]]$ and $[[EP]]$. Some lambda calculus terms are written in shorthand as labels (e.g. $PAST$, $PRESENT$, CUL). These are defined after the semantic rules.

1. $[[NP\ \ VP]] = ([[VP]])\ [[NP]]$

2. $[[NP_1\ VP\ NP_2]] = ((([[VP]])\ [[NP_1]])\ [[NP_2]]$
3. $[[MOD\ AUX_1\ AUX_2\ V]] = ([[V]])\ ((([[AUX_2]])\ (([[AUX_1]])\ [[MOD]]))$
4. $[[MOD\ AUX\ TV\ V]] = ([[V]])\ ((([[TV]])\ (([[AUX]])\ [[MOD]]))$
5. $[[MOD\ AUX\ V]] = ([[V]])\ (([[AUX]])\ [[MOD]])$
6. $[[MOD\ TV\ V]] = ([[V]])\ (([[TV]])\ [[MOD]])$
7. $[[AUX_1\ AUX_2\ V]] = ([[V]])\ (([[AUX_2]])\ [[AUX_1]])$
8. $[[AUX\ TV\ V]] = ([[V]])\ (([[TV]])\ [[AUX]])$
9. $[[TV\ V]] = ([[V]])\ [[TV]]$
10. $[[AUX\ V]] = ([[V]])\ [[AUX]]$
11. $[[MOD\ V]] = ([[V]])\ [[MOD]]$
12. $[[TV_{vf:pas}]] = ([[TV_{vf:inf}]])PAST$
13. $[[TV_{vf:pap}]] = [[TV_{vf:inf}]]$
14. $[[AUX_{vf:pas}]] = ([[AUX_{vf:inf}]])PAST$
15. $[[AUX_{vf:prs}]] = ([[AUX_{vf:inf}]])PRESENT$
16. $[[AUX_{vf:pap}]] = [[AUX_{vf:inf}]]$
17. $[[V_{vf:pas}]] = ([[V_{vf:inf}]])PAST$
18. $[[V_{vf:pap}]] = [[V_{vf:inf}]]$
19. $[[V_{vf:prp}]] = [[V_{vf:inf}]]$
20. $[[\text{a/an}\ NOUN]] = [[NOUN]]$
21. $[[\text{the}\ NOUN]] = [[NOUN]]$

The lambda terms used to represent the semantics of main verbs, which are defined below, vary according to whether the verb is transitive or intransitive (e.g. compare the term for the intransitive *run* with those for the transitive verbs below) and what the verb's associated aspectual structure is (e.g. *draw*, *run to* and the transitive case of *eat* are all associated by default with the aspectual structure of a culminated process, whereas *run* and *play* are by default processes and *win* is by default a culmination).

$$[[\text{draw}_{vf:inf}]] = \lambda xyz.(y)\lambda v.(z)\lambda w.((x)CP)draw(v,w)$$

$[[\text{eat}_{vf:inf}]] = \lambda xyz.(y)\lambda v.(z)\lambda w.((x)\,CP)\,eat(v,w)$
$[[\text{run}_{vf:inf}]] = \lambda xy.(y)\lambda v.((x)\,PROC)\,run(v)$
$[[\text{run}_{vf:inf}\text{to}]] = \lambda xyz.(y)\lambda v.(z)\lambda w.((x)\,CP)\,runto(v,w)$
$[[\text{play}_{vf:inf}]] = \lambda xyz.(y)\lambda v.(z)\lambda w.((x)\,PROC)\,play(v,w)$
$[[\text{win}_{vf:inf}]] = \lambda xyz.(y)\lambda v.(z)\lambda w.((x)\,CUL)\,win(v,w)$

We note here that *type coercion* of events is currently a topic of considerable interest in the field of tense and aspect. It aims to explain the phenomenon whereby the aspectuality of a phrase can shift when it is combined with certain arguments or modifiers (for a recent investigation see [10]). Rather than attempt a formalisation of event type coercion here, the semantic rules defined below allow one aspectual structure to replace another that is by default associated with the main verb of a sentence, regardless of what that default structure is. The lambda terms that can carry out such a replacement are those representing the semantics of the auxiliary *be* and the transition verbs *start*, *stop* and *finish*. Some of the example temporal readings later in this section illustrate this.

$[[\text{be}_{vf:inf}]] = \lambda xyz.((x)PROG)z$
$[[\text{start}_{vf:inf}]] = \lambda xyz.((x)START)z$
$[[\text{stop}_{vf:inf}]] = \lambda xyz.((x)STOP)z$
$[[\text{finish}_{vf:inf}]] = \lambda xyz.((x)FINISH)z$

Lastly the semantic rules for the auxiliary *have*, the modal *will* and nouns are defined below.

$[[\text{have}_{vf:inf}]] = PERFECT$
$[[\text{will}]] = FUTURE$
$[[NOUN]] = \lambda x.(x)NOUN$ e.g. $[[\text{Max}]] = \lambda x.(x)Max$

Many of the lambda terms above refer to labels (e.g. $PAST$, CUL, $PROG$). The lambda terms for these labels are defined next.

CUL is a term containing the structure for the aspectual class called a Culmination by Moens, corresponding to the Vendler aspectual class *Achievement*. It is defined as:

$$\lambda xy.(x)[[EP]](focus(y \wedge \langle \underline{A} \rangle([D](y)) \wedge \langle A \rangle([D](\neg y))))$$

The HSF1A formula contained in CUL reflects the definition of a culmination point earlier in this section. The *focus* operator occurs at the point of transition from *on* to *off*. $FINISH$ is defined to be the same lambda term as CUL.

CP is a term containing the structure for the aspectual class called a

Culminated Process by Moens, corresponding to the Vendler aspectual class *Accomplishment*. It is defined as:

$$\lambda xy.(x)[[EP]](y \wedge \langle \underline{A} \rangle(focus([D](y))) \wedge \langle A \rangle([D](\neg y)))$$

The HSF1A formula contained in CP reflects the definition of a culmination point earlier in this section. The focus operator occurs at the maximal interval at which the $[D]$ operator holds, indicating that focus is durative in this case and spans the process leading up to the culmination point.

$FUTURE$ is a term for the future simple tense. It is defined as:

$$\lambda x.(x)\lambda y.\langle L \rangle(y)$$

$PAST$ is a term for the past simple tense. It is defined as:

$$\lambda x.(x)\lambda y.\langle \underline{L} \rangle(y)$$

$PERFECT$ is a term for perfective aspect. It is defined as:

$$\lambda w.(w)\lambda xy.(y)\lambda z.(x)focus(\langle \underline{L} \rangle(z))$$

$PRESENT$ is a term for the present simple tense. It is defined as:

$$\lambda x.(x)\lambda y.y$$

$PROC$ is a term containing the structure for the aspectual class called a Process by Moens, corresponding to the Vendler aspectual class *Activity*. It is defined as:

$$\lambda xy.(x)focus([D](y))$$

The focus operator occurs at the maximal interval at which the $[D]$ operator holds, indicating that focus is durative in this case and spans the whole process.

$PROG$ is a term for the progressive. It is defined as:

$$\lambda xy.(x)focus(\langle \underline{D} \rangle([D](y)))$$

The HSF1A formula contained in $PROG$ reflects the definition of an atomic proposition being *on* earlier in this section. It is only intended as an account of one sense of the progressive, as it is not an accurate analysis of a repetitive sense such as *I am studying every day*. The focus operator occurs at an internal interval of the maximal interval at which the $[D]$ operator holds, which means that the focus is within the overall structure.

$START$ is a term representing a start of some activity. It is defined as:

$$\lambda xy.(x)[[EP]](focus(\langle \underline{A} \rangle([D](\neg y)) \wedge \langle A \rangle([D](y))))$$

The HSF1A formula contained in $START$ reflects the definition of a start point earlier in this section. The focus operator occurs at the point of transition from *off* to *on*.

$STOP$ is a term representing a stop in some activity. It is defined as:

$$\lambda xy.(x)[[EP]](focus(\langle \underline{A} \rangle([D](y)) \wedge \langle A \rangle([D](\neg y))))$$

The HSF1A formula contained in $STOP$ reflects the definition of a stop point earlier in this section. The focus operator occurs at the point of transition from *on* to *off*.

4.3.3. *Example Temporal Readings*
We give here the HSF1A temporal readings of some example sentences. Each example shows the application of the semantic rules defined above to a sentence analysed into syntactic categories according to the grammar in Figure 25, to derive an HSF1A formula, from which a timeline model (its closest position timeline) is constructed. In the derivation of temporal readings, $=_\beta$ is shorthand for *by beta reduction*, $=_{rule\ x}$ is shorthand for *by semantic rule x* where x is the index of one of the semantic rules defined earlier, and $=_{lex\ rule}$ is shorthand for *by a lexical semantic rule* i.e. by one of the semantic rules that assigns a lambda term to a word in its base form.

4.3.3.1. *Max is running* The derivation of the HSF1A temporal reading is described below, starting with the lexical syntactic categories:

$\underline{NP \leftarrow NOUN}$:

$[[\text{Max}]] =_{lex\ rule} \lambda x.(x)Max$

$\underline{V}$:

$[[\text{running}]] = [[\text{run}_{vf:prp}]] =_{rule\ 19} [[\text{run}_{vf:inf}]] =_{lex\ rule}$
$\lambda xy.(y)\lambda v.((x)PROC)run(v)$

$\underline{AUX}$:

$[[\text{ is }]] = [[\text{be}_{vf:prs}]] =_{rule\ 15} ([[\text{be}_{vf:inf}]])PRESENT =_{lex\ rule}$
$(\lambda xyz.((x)PROG)z)PRESENT =_\beta$
$\lambda yz.((PRESENT)PROG)z$

$\underline{VP \leftarrow AUX\ V}$:

[[is running]] $=_{rule\ 10}$ ([[running]]) [[is]]
which, replacing [[running]] with its λ-term derived earlier, $=_\beta$
$\lambda y.(y)\lambda v.(([[\text{is}]])PROC)run(v)$

$\underline{S \leftarrow NP\ VP}$:

[[Max is running]] $=_{rule\ 1}$ ([[is running]]) [[Max]]
which, replacing [[is running]] with its λ-term derived earlier, $=_\beta$
$([[\text{Max}]])\lambda v.(([[\text{is}]])PROC)run(v)$
which, replacing [[Max]] with its λ-term derived earlier, $=_\beta$
$(\lambda v.(([[\text{is}]])PROC)run(v))Max =_\beta (([[\text{is}]])PROC)run(Max)$
which, replacing [[is]] with its λ-term derived earlier, $=_{\beta\ (twice)}$
$((PRESENT)PROG)run(Max) =_\beta$
$((PROG)\lambda y.y)run(Max) =_{\beta\ (twice)}$
$(\lambda y.y)focus(\langle\underline{D}\rangle([D](run(Max)))) =_\beta focus(\langle\underline{D}\rangle([D](run(Max))))$

Temporal reading of *Max is running*
$focus(\langle\underline{D}\rangle([D](run(Max))))$

Closest Position Timeline

|　　　a　　　|

|　*+　|

Timeline Index
a: $[D](run(Max))$　　*: *Now*　　+: *Focus*

Figure 27. A temporal reading of *Max is running*

4.3.3.2. *Jill ate an apple* The derivation of the HSF1A temporal reading is described below:

$\underline{NP \leftarrow NOUN}$:

[[Jill]] $=_{lex\ rule}$ $\lambda x.(x)Jill$

$\underline{NP \leftarrow DET\ NOUN}$:

[[an apple]] $=_{rule\ 20}$ [[apple]] $=_{lex\ rule}$ $\lambda x.(x)apple$

$\underline{VP{\leftarrow}V}$:

$[[\text{ate}]] = [[\text{eat}_{vf:pas}]] =_{rule\ 17} ([[\text{eat}_{vf:inf}]])PAST =_{lex\ rule}$
$(\lambda xyz.(y)\lambda v.(z)\lambda w.((x)\,CP)\,eat(v,w))\,PAST =_\beta$
$\lambda yz.(y)\lambda v.(z)\lambda w.((PAST)\,CP)\,eat(v,w)$

$\underline{S{\leftarrow}NP\ \ VP\ \ NP}$:

$[[\text{Jill ate an apple}]] =_{rule\ 2} (([[\text{ate}]])[[\text{Jill}]])[[\text{an apple}]]$
which, replacing [[ate]] with its λ-term derived earlier, $=_{\beta\,(twice)}$
$([[\text{Jill}]])\lambda v.([[\text{an apple}]])\lambda w.((PAST)\,CP)\,eat(v,w)$
which, replacing [[Jill]] with its λ-term derived earlier, $=_\beta$
$(\lambda v.([[\text{an apple}]])\lambda w.((PAST)\,CP)\,eat(v,w))\,Jill =_\beta$
$([[\text{an apple}]])\lambda w.((PAST)\,CP)\,eat(Jill,w)$
which, replacing [[an apple]] with its λ-term derived earlier, $=_\beta$
$(\lambda w.((PAST)\,CP)\,eat(Jill,w))\,apple =_\beta$
$((PAST)\,CP)\,eat(Jill,apple) =_\beta$
$((CP)\lambda y.\langle\underline{L}\rangle(y))\,eat(Jill,apple)$
which, abbreviating $eat(Jill, apple)$ as p, $=_{\beta\,(twice)}$
$(\lambda y.\langle\underline{L}\rangle(y))[[EP]](p \wedge \langle\underline{A}\rangle(focus([D](p))) \wedge \langle A\rangle([D](\neg p)))) =_\beta$
$\langle\underline{L}\rangle([[EP]](p \wedge \langle\underline{A}\rangle(focus([D](p)))) \wedge \langle A\rangle([D](\neg p))))$

<u>Temporal reading of *Jill ate an apple*</u>
(The proposition $eat(Jill, apple)$ is abbreviated below as p)

$\langle\underline{L}\rangle([[EP]](p \wedge \langle\underline{A}\rangle(focus([D](p))) \wedge \langle A\rangle([D](\neg p))))$

<u>Closest Position Timeline</u>

| + $|_a$ b | | * |

<u>Timeline Index</u>

+: *Focus*	a: $eat(Jill, apple)$	*: *Now*
+: $[D](eat(Jill, apple))$	b: $[D](\neg eat(Jill, apple))$	

Figure 28. A temporal reading of *Jill ate an apple*

4.3.3.3. *Jane had finished drawing the picture* The derivation of the HSF1A temporal reading is described below:

$\underline{NP{\leftarrow}NOUN}$:

$[\![\text{Jane}]\!] =_{lex\ rule} \lambda x.(x)Jane$

$\underline{NP{\leftarrow}DET\ NOUN}$:

$[\![\text{the picture}]\!] =_{rule\ 21} [\![\text{picture}]\!] =_{lex\ rule} \lambda x.(x)picture$

$\underline{AUX}$:

$[\![\text{had}]\!] = [\![\ \text{have}_{vf:pas}\ ]\!] =_{rule\ 14} ([\![\text{have}_{vf:inf}]\!])PAST =_{lex\ rule}$
$(PERFECT)PAST =_{\beta}$
$(PAST)\lambda xy.(y)\lambda z.(x)focus(\langle\underline{L}\rangle(z)) =_{\beta}$
$(\lambda xy.(y)\lambda z.(x)focus(\langle\underline{L}\rangle(z)))\lambda x.\langle\underline{L}\rangle(x) =_{\beta}$
$\lambda y.(y)\lambda z.(\lambda x.\langle\underline{L}\rangle(x))focus(\langle\underline{L}\rangle(z)) =_{\beta}$
$\lambda y.(y)\lambda z.\langle\underline{L}\rangle(focus(\langle\underline{L}\rangle(z)))$

$\underline{TV}$:

$[\![\text{finished}]\!] = [\![\text{finish}_{vf:pap}]\!] =_{rule\ 13} [\![\text{finish}_{vf:inf}]\!] =_{lex\ rule}$
$\lambda xyz.((x)FINISH)z$

$\underline{V}$:

$[\![\text{drawing}]\!] = [\![\text{draw}_{vf:prp}]\!] =_{rule\ 19} [\![\text{draw}_{vf:inf}]\!] =_{lex\ rule}$
$\lambda xyz.(y)\lambda v.(z)\lambda w.((x)CP)draw(v,w)$

$\underline{VP{\leftarrow}AUX\ TV\ V}$:

$[\![$ had finished drawing $]\!] =_{rule\ 8} ([\![\text{drawing}]\!])(([\![\text{finished}]\!])[\![\text{had}]\!])$
which, replacing $[\![\text{drawing}]\!]$ with its λ-term derived earlier, $=_{\beta}$
$\lambda yz.(y)\lambda v.(z)\lambda w.((([\![\text{finished}]\!])[\![\text{had}]\!])CP)draw(v,w)$

Derivation for $([\![\text{finished})[\![\text{had}]\!]$:

Replacing $[\![\text{finished}]\!]$ and $[\![\text{had}]\!]$ with their λ-terms derived earlier,
$([\![\text{finished}]\!])[\![\text{had}]\!] =_{\beta}$
$\lambda yz.((\lambda y.(y)\lambda z.\langle\underline{L}\rangle(focus(\langle\underline{L}\rangle(z))))FINISH)z =_{\beta}$
$\lambda yz.((FINISH)\lambda z.\langle\underline{L}\rangle(focus(\langle\underline{L}\rangle(z))))z$

$\underline{S \leftarrow NP\ VP\ NP}$:

$[[$ Jane had finished drawing the picture $]] =_{rule\ 2}$
$((([[\text{had finished drawing}]])[[\text{Jane}]])[[\text{the picture}]]$
which, replacing [[had finished drawing]] with its λ-term derived earlier, $=_{\beta\ (twice)}$
$([[\text{Jane}]])\lambda v.([[\text{the picture}]])\lambda w.(((([[\text{finished}]])[[\text{had}]])CP)draw(v,w)$
which, replacing [[Jane]] with its λ-term derived earlier, $=_{\beta}$
$(\lambda v.([[\text{the picture}]])\lambda w.(((([[\text{finished}]])[[\text{had}]])CP)draw(v,w))Jane =_{\beta}$
$([[\text{the picture}]])\lambda w.(((([[\text{finished}]])[[\text{had}]])CP)draw(Jane,w)$
which, replacing [[the picture]] with its λ-term derived earlier, $=_{\beta}$
$(\lambda w.(((([[\text{finished}]])[[\text{had}]])CP)draw(Jane,w))picture =_{\beta}$
$(((([[\text{finished}]])[[\text{had}]])CP)draw(Jane,picture)$
which, abbreviating $draw(Jane,picture)$ as p and replacing (([[finished]]) [[had]]) with its λ-term derived earlier, $=$
$((\lambda yz.((FINISH)\lambda z.\langle \underline{L}\rangle(focus(\langle \underline{L}\rangle(z))))z)CP)p =_{\beta\ (twice)}$
$((FINISH)\lambda z.\langle \underline{L}\rangle(focus(\langle \underline{L}\rangle(z))))p =_{\beta\ (twice)}$
$(\lambda z.\langle \underline{L}\rangle(focus(\langle \underline{L}\rangle(z))))[[EP]](focus(p \wedge \langle \underline{A}\rangle([D](p)) \wedge \langle A\rangle([D](\neg p)))) =_{\beta}$
$\langle \underline{L}\rangle(focus(\langle \underline{L}\rangle([[EP]](focus(p \wedge \langle \underline{A}\rangle([D](p)) \wedge \langle A\rangle([D](\neg p)))))))$

Temporal reading of *Jane had finished drawing the picture*
(The proposition $draw(Jane,picture)$ is abbreviated as p)

$\langle \underline{L}\rangle(focus(\langle \underline{L}\rangle([[EP]](focus(p \wedge \langle \underline{A}\rangle([D](p)) \wedge \langle A\rangle([D](\neg p)))))))$

Closest Position Timeline

$|\quad a \quad |_b \quad c \quad | \qquad | \quad + \quad | \qquad | \quad * \quad |$

Timeline Index
a: $[D](draw(Jane,picture))$ $*$: *Now*
b: $draw(Jane,picture)$ $+$: *Focus*
c: $[D](\neg draw(Jane,picture))$

Figure 29. A temporal reading of *Jane had finished drawing the picture*

4.3.4. *Extending the English Fragment*

Outlined below are some remarks on how phenomena in an extended English fragment could be accounted for using extensions to HSF1, HSF1A and timelines.

4.3.4.1. *Quantifiers and During* A possible temporal semantics for temporal adverbial phrases such as *every morning* may be enabled by extending the HSF1 and HSF1A fragments to include logical implication between HSF1/HSF1A formulae and allowing $[D]$ to take as its argument any formula of the extended fragment, rather than limiting the argument to be an atomic formula. A temporal semantics of an adverbial phrase such as *every DURATION* where *DURATION* is a noun denoting some period of time such as *morning*, *day*, *year* etc. could then be built around the formula:

$$[D](DURATION \rightarrow \langle D \rangle(F))$$

where F is some formula of the extended fragment. A possible reading for the sentence *Max is running every morning* would then take the form:

$$\langle \underline{D} \rangle([D](morning \rightarrow \langle D \rangle([D](run(Max)))))$$

One way to model such a reading using a timeline would be to extend the definition of a timeline to include two new types of association: *implying associations* and *implied associations*, where every implying association would be required to be related to an implied association by a semantic implication. For the reading above this would mean that in the modelling timeline there would be a semantic implication between an implying association $((s,\, t)$: $[D](morning \rightarrow))$ and an implied association $((v,\, w)$: $\rightarrow [D](run(Max))$ where $s < v < w < t$ in the timeline's ordering.

A possible semantics for prepositional phrases involving the preposition *during* e.g. *during the summer* could be enabled by making further use of the extension to HSF1/HSF1A that was outlined above, using a formula of the form:

$$DURATION \rightarrow \langle D \rangle(F)$$

where F is some formula of the extended fragment. A possible reading for the sentence *Max is running every morning during the summer* would then take the form:

$$\langle \underline{D} \rangle(\langle D \rangle(summer \rightarrow [D](morning \rightarrow \langle D \rangle([D](run(Max))))))$$

The above reading assumes that *the summer* is treated as equivalent to *some summer*, hence the use of the form $\langle \underline{D} \rangle(\langle D \rangle(F))$ which means that the interval at which F holds could be anywhere in time in relation to *now*. If *the summer* was a reference to a particular summer, the reading may well position the summer differently in time. The positioning of the proposition *summer* would depend on resolving any references in the head of the prepositional phrase (e.g. *last summer*) and the context of the utterance.

Quantifiers in noun phrases in subject and object positions of a sentence could, following a well established approach, be represented using first order predicate calculus quantifiers. In this case the following additions could be made to the semantic rules.

$[\![$ every $NOUN$ $]\!] = \lambda y.\forall Y(([\![NOUN]\!])\lambda z.z(Y)) \rightarrow (y)Y$
$[\![$ a/an $NOUN$ $]\!] = \lambda y.\exists Y(([\![NOUN]\!])\lambda z.z(Y)) \wedge (y)Y$

This would result in the following temporal reading for the sentence *Every boy was running*:

$$\forall Y(boy(Y) \rightarrow (\langle \underline{L} \rangle (focus(\langle \underline{D} \rangle ([D](run(Y)))))))$$

In finite domains such a reading could be modelled by constructing a timeline for each entity denoted by the predicate representing the universally quantified noun, for example one timeline could be constructed for each boy denoted by the predicate *boy* in the context of the utterance.

4.3.4.2. *When* We have used timelines to model clauses, however focused timelines are expected to be applicable in modelling complex sentences. The subordinating conjunction *when* is of particular interest due to its various temporal meanings *at which time*, *during which time* and *immediately after*. Given a sentence S_1 *when* S_2 where S_1 and S_2 are clauses, if an HSF1A temporal reading were constructed for each clause and focused timelines were constructed for the readings, the position of the *focus* interval in each timeline relative to the aspectual structure being described may be of relevance in determining which possible meaning of *when* is being used, because *focus* can distinguish points from intervals and the progressive from other forms. One possible way of building a single timeline that models a sentence S_1 *when* S_2 is by *merging* one timeline into the other under the constraints that the current intervals of the two timelines coincide and the positions of the foci of the timelines are used to determine where the aspectual structures of the timelines are placed relative to each other. To *merge* a timeline T_2 (with interval mapping IM' and an ordering ORD') into a timeline T_1 (with interval mapping IM and ordering ORD) means, informally, that (a) every point variable in ORD' is either renamed to be one of the existing point variables in ORD or included as a new point variable to ORD, such that the renamed or included point variables retain the relative positions of the original point variables in ORD', and (b) a copy of every association in IM', with its point variables renamed accordingly, is included in IM provided it is not a sub-association of an existing association in IM.

4.3.4.3. *In, For and At Adverbials* Moens and Steedman argue in [6] that durational *in* adverbial phrases such as *in ten minutes* or *in less than an hour* can combine with Culminated Processes but not with Processes. For example, *Max ran to the station in ten minutes* is felicitous, but *Max ran in ten minutes* is infelicitous in the durational sense unless there is some special context provided. In contrast, *for* adverbial phrases are felicitous with Processes, e.g. *Max ran for five minutes* but are only felicitous with Culminated Processes if the context allows a repetition of the event. For instance the sentence *Max ran a mile for a day* would require an interpretation where the Culminated Process *Max run a mile* is repeated throughout the day.

The HSF1A definition of a Culminated Process differs from that of a Process in that the former is terminative, consisting of a stop point at which the proposition holds, whereas the latter is not terminative, consisting of an interval throughout which a proposition is *on*. This contrast between a *point* of transition occurring within a Culminated Process and an *interval* of homogeneity which comprises a Process offers a route to formalising the difference in linguistic behaviour. Extending the lambda terms CP and $PROC$ with a third parameter w as defined below identifies the places within the HSF1A formulae where the contrast between terminative and non-terminative aspectual structures may be realised.

$$CP' = \lambda wxy.(x)[[EP]](w \wedge y \wedge \langle \underline{A} \rangle (focus([D](y))) \wedge \langle A \rangle ([D](\neg y)))$$
$$PROC' = \lambda wxy.(x) focus(w \wedge [D](y))$$

If we formalise the apparent terminative requirement in the semantics of *in* using a formula $[[POINT]]$ which is true only at points, and the apparent non-terminative requirement in the semantics of *for* using a formula $[[INTVL]]$ which is only true at durative intervals, one may note that if $[[INTVL]]$ is passed as w to CP' the resulting formula is inconsistent as the operator $[[EP]]$ makes the current interval a point, but there is no such inconsistency if $[[POINT]]$ is passed. In the case of $PROC'$ the reverse holds, i.e. if $[[POINT]]$ is passed the formula is inconsistent because the conjoined sub-formula $[D](y)$ for some atomic proposition y may only hold at a durative interval, but there is no such inconsistency if $[[INTVL]]$ is passed. The fragments HSF1 and HSF1A would have to be extended with $[[INTVL]]$ and $[[POINT]]$ and the modelling extended with their semantics in order to be used in the outlined account. HS formulae exist for $[[POINT]]$ and $[[INTVL]]$, which are defined as $[B](false)$ and $\langle B \rangle (true)$ respectively, where the proposition *false* is always false and *true* is always true. The unique proposition *true* may also be a suitable addition to HSF1 and HSF1A as a value for w in CP' or $PROC'$ when there are no terminative or non-terminative requirements indicated by temporal adverbial phrases in a sentence.

An indication of the time duration in an *in* or *for* prepositional phrase may be represented as an atomic proposition e.g. $hours(two)$ and abstracted by a further parameter v in the corresponding lambda terms, which would be revised as follows:

$$CP'' = \lambda vwxy.(x)[[EP]](w \wedge y \wedge \langle \underline{A} \rangle (focus(v \wedge [D](y))) \wedge \langle A \rangle ([D](\neg y)))$$
$$PROC'' = \lambda vwxy.(x) focus(v \wedge w \wedge [D](y))$$

Note that in both cases the proposition representing duration holds at the focused durative interval.

The discussion above outlines possible logical representations for some of the differences between the linguistic behaviour of *in* and *for*. A basis for an account for alternative interpretations such as those involving repetition of a Culminated Process throughout some interval or of a possible coercion from a Proces to a Culminated Process is the subject of future research.

A temporal *at* adverbial phrase can combine felicitously with a phrase denoting a focus on a point of transition e.g. *Max started running at ten o'clock* or *Jack finished eating lunch at noon*, but it is infelicitous with a phrase denoting a focus on a durative interval e.g. *Max ran to the station at four o'clock* or *Max worked at five o'clock*. Using lambda terms extended in a similar way to CP'' and $PROC''$, the terms for the transitional aspectual structures $START$ and $FINISH$ would be as follows.

$$START'' = \lambda vwxy.(x)[[EP]](focus(v \wedge w \wedge \langle \underline{A} \rangle ([D](\neg y)) \wedge \langle A \rangle ([D](y))))$$
$$FINISH'' = \lambda vwxy.(x)[[EP]](focus(v \wedge w \wedge y \wedge \langle \underline{A} \rangle ([D](y)) \wedge \langle A \rangle ([D](\neg y))))$$

It is possible to formalise the apparent requirement that the focus is on a point of transition in the semantics of *at* by ensuring that the formula passed to v takes the form $[[POINT]] \wedge p$ where p is a proposition denoting some instant in time e.g. $clockhour(4)$. This would result in inconsistency in CP'' and $PROC''$, which agrees with the linguistic behaviour, but remain consistent with $FINISH''$ and $START''$, as well as similarly revised terms for CUL and $STOP$.

4.3.4.4. *Noun Phrases and Aspect* The sentence semantics defined earlier does not consider the contribution of subject and object noun phrases to establishing the aspectual structure denoted in the temporal reading of a sentence, and therefore cannot account for the differences between sentences such as *Jack ate porridge* (a Process) and *Jill ate the apple* (a Culminated Process). Verkuyl in [16] describes how a feature Specified Quantity in noun phrases can influence how terminative aspect of eventualities is determined. A possible extension to the the sentence semantics provided here could be made based on whether a noun phrase is compatible with a terminative or non-terminative aspectual structure, by having a noun phrase provide a temporal component in its semantics as well as a propositional one, in a similar way to the outlined account for *in*, *for* and *at adverbials*, i.e. use of a $[[POINT]]$ formula may indicate compatibility with terminative aspect, $[[INTVL]]$ with non-terminative aspect. Verbs that allow both terminative and non-terminative semantics depending on their noun phrase arguments would then have to be associated with more than one alternative semantic term, each with a different aspectual structure.

5. Reasoning with Timelines

Having defined how to determine HSF1A temporal readings of simple English sentences and and how model them as focused timelines, we consider next how the comparison of timeline models can be used as a basis to check entailments between temporal readings.

5.1. Entailment Checker Principles

In the general case, a semantic entailment between two HSF1A formulae $F_l \models F_r$ is valid if *each* timeline model Tl_I that satisfies F_l is a supertimeline of a timeline Tr_i that satisfies F_r.

DEFINITION 15. *A timeline T_1 with current interval CI, ordering ORD and interval mapping IM is a* **supertimeline** *of a timeline T_2 with current interval CI', ordering ORD' and interval mapping IM', iff (a) CI and CI' are the same interval, (b) ORD' contains all the point variables of ORD such that for any two point variables t_x, t_y in ORD', if $t_x < t_y$ in ORD, $t_x < t_y$ in ORD', and (c) every association in IM' is either an association in IM or a sub-association of an association in IM.*

DEFINITION 16. *If a timeline T_1 is a supertimeline of a timeline T_2, T_2 is referred to as a* **subtimeline** *of T_1.*

In order to determine whether one timeline is a subtimeline of another it is necessary to check that restrictions (a), (b) and (c) in Definition 15 above hold. To carry out these checks in a procedure, the definitions below are important.

DEFINITION 17. *An* **association copy** *of an association occurring in an interval mapping IM of a timeline T_1 to an interval mapping*

IM' of a timeline T_2 is the result of ensuring that the association is also present in IM'.

DEFINITION 18. *An* **association derivation** *of an association occurring in an interval mapping IM of a timeline T_1 to an interval mapping IM' of a timeline T_2 is the result of ensuring that a sub-association of the association is present in IM'.*

DEFINITION 19. *An* **association propagation** *is an association copy or an association derivation.*

Association copying corresponds to the first of the alternatives in rules 1 (I) and 2 (II) of the truth relation definition in Section 3. Association derivation corresponds to the remaining alternatives in rules 1 (I) and 2 (II) which are depicted in Figure 8.

We refer to a check as to whether a timeline that satisfies a formula F_r is a subtimeline of a timeline that satisfies a formula F_l as a *subtimeline query*, in which a query is made to find a subtimeline Tr_i of Tl_I that satisfies F_r. For a full entailment check, there is one subtimeline query for each timeline that satisties F_l. A restricted entailment check may be carried out by considering only the Closest Position Timeline of F_l, so that only one subtimeline query has to be made. We use the notation $F_l \models_{cpt} F_r$ for a semantic entailment restricted such that only the CPT of F_l is considered. In this section we will restrict our investigation of semantic entailment checking to $\models_{cpt}$. Note that this does not restrict the timelines of F_r that may be considered.

5.2. Subtimeline Querying

There are two inputs to a subtimeline query: a supertimeline Tl_I and a formula F_r. The target output of a query is a subtimeline of Tl_I, call it Tr_i, that is a timeline of F_r.

A subtimeline query is a formula traversal, similar to timeline construction structurally. The difference is that it carries out a bottom-up interpretation on the formula F_r rather than a top-down interpretation on the formula F_l, and obtains associations by propagation from the supertimeline Tl_I rather than simply asserting them. Consider the case of the formula $\langle \underline{A} \rangle ([D](\neg p)$. Timeline construction begins with an initial current interval and an ordering containing just the point variables of that interval. The operator $\langle \underline{A} \rangle$ is interpreted at the initial current interval, determining a new current interval, at which the extended atomic formula $[D](\neg p)$ is then asserted to hold. A subtimeline query, however, begins with the ordering and current interval of the supertimeline Tl_I, and first interprets $[D](\neg p)$, by attempting an association propagation from Tl_I. If successful this determines an interval at which the operator $\langle \underline{A} \rangle$ is then interpreted bottom-up. Finally, since $\langle \underline{A} \rangle$ is the topmost operator of the formula, the new current interval is compared to the current interval of the supertimeline to check that the positioning of the subtimeline is correct.

5.3. Entailment Check Example

We give here an example of a semantic entailment check under the restriction that we consider only the CPT of the supertimeline, so there is only one subtimeline query. The entailment to check is:

$$\langle \underline{L} \rangle([[EP]](\langle \underline{A} \rangle([D](\neg p)) \wedge \langle A \rangle([D](p)))) \models_{cpt} \langle \underline{L} \rangle(\neg p \wedge \langle L \rangle([D](p)))$$

Prior to the query we construct the CPT for the entailing formula, i.e. the supertimeline. We then begin the query, providing as input the supertimeline and the entailed formula, with the aim of obtaining a subtimeline that satisfies the entailed formula. The initial state of the query is shown in Figure 30. Interpreting the formula bottom-up, we try to

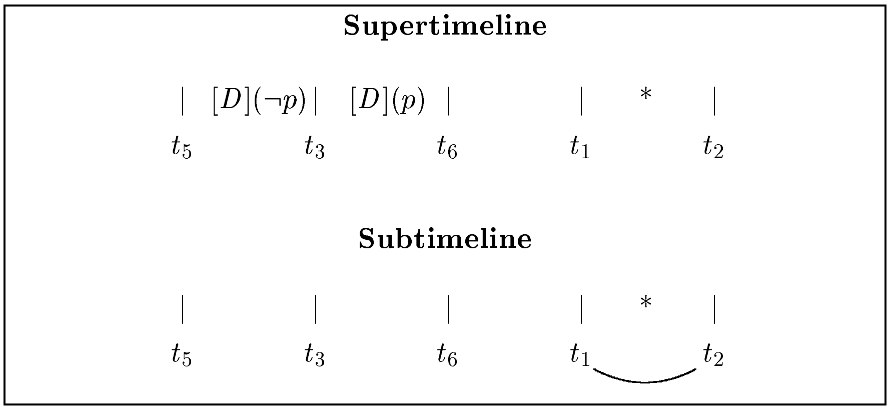

Figure 30. Subtimeline query example, part 1

propagate the formula $\neg p$ from an association in the supertimeline. This may be achieved with an association derivation, as shown in Figure 31. Then we try to obtain an association supporting the formula $[D](p)$. We can propagate an appropriate association by copying, as shown in Figure 32. Three association derivations are also applicable (left, right and internal homogeneous), but we do not show these cases. Next, the bottom-up traversal leads to the $\langle L \rangle$ operator, which is shown being interpreted in Figure 33. Notice that the interpretation of the operator is in reverse compared to timeline construction as the formula traversal is bottom-up rather than top-down. The operator interpretation in this case may lead to many alternative positions for the new current interval,

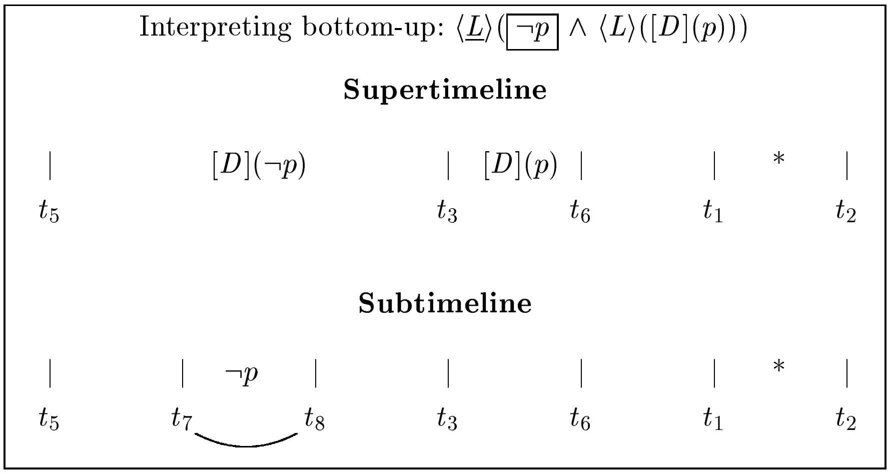

Figure 31. Subtimeline query example, part 2

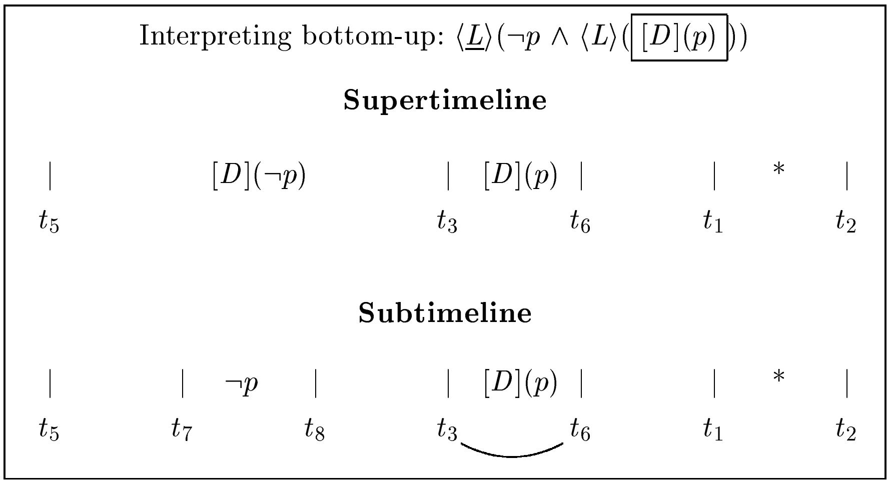

Figure 32. Subtimeline query example, part 3

just as in timeline construction, but for brevity we show only the relevant one. Then the $\wedge$ operator is interpreted, which results in a check that the current interval is the same for both branches of the formula which $\wedge$ partitions. The check is successful (the current intervals in Figure 31 and Figure 33 are the same). Finally, the $\langle \underline{L} \rangle$ operator is interpreted bottom-up. Again there are many alternative positions for the new current interval, but the relevant one meets initial current interval shown in Figure 30. The illustration in Figure 34 confirms the successful end of the query, in which a subtimeline has been found that satisfies the entailed formula.

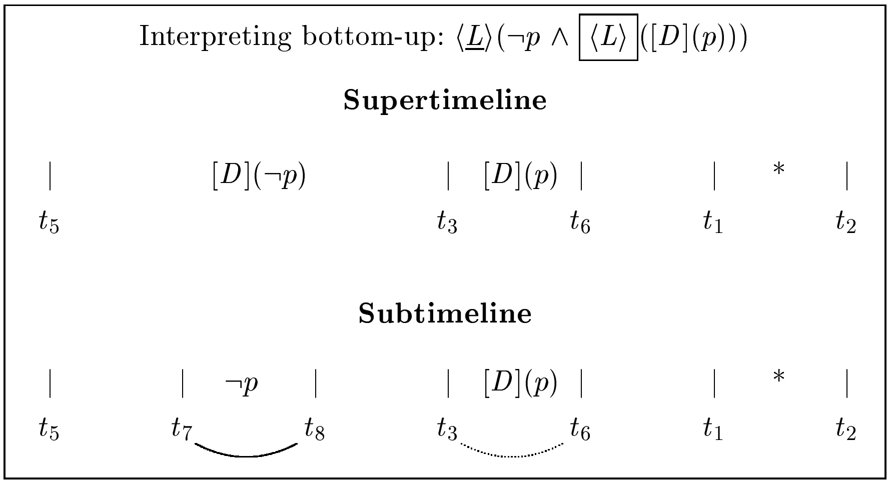

Figure 33. Subtimeline query example, part 4

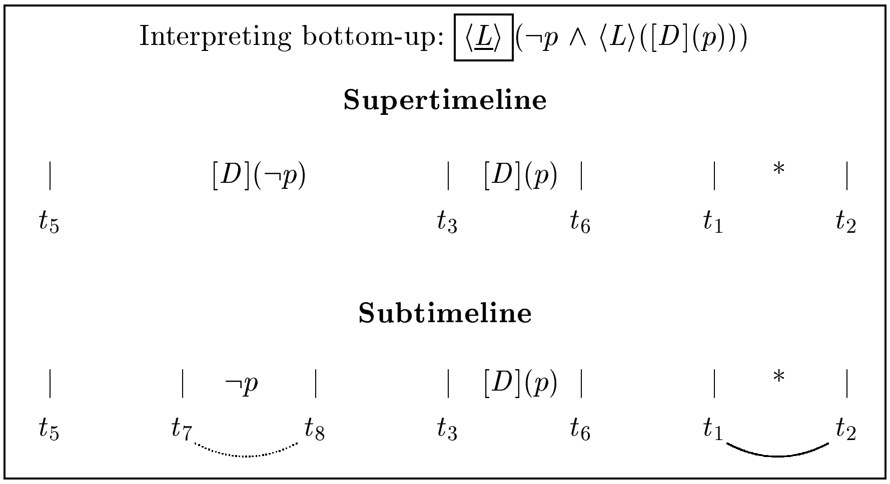

Figure 34. Subtimeline query example, part 5

The example above did not contain an occurrence of the *focus* operator. The the linguistic entailment checks below do, because the operator is used in the temporal sentence semantics described in Section 4. However, in the present work we do not enforce any constraints between the focus intervals of the supertimeline and subtimeline.

5.4. Explanation of Failed Subtimeline Queries

Subtimeline queries may fail for two kinds of reason. An *association failure* occurs when not all extended atomic formulae of the entailed formula could be propagated from associations in the supertimeline.

A *pattern failure* occurs when all extended atomic formulae could be propagated but not in a pattern that formed a timeline satisfying the entailed formula.

In the entailment checks below, if an association failure occurs we provide details of the associations which could not be propagated. If a pattern failure occurs we provide details of a timeline in which all associations were propagated but not in an appropriate pattern.

5.5. Linguistic Entailment Results

Having summarised the principles of the entailment checker, we next give some examples of its results. Figures 35 to 38 show checks for various semantic entailments between HSF1A temporal readings of English sentences.

Entailment Check:
(The proposition $run(Max)$ is abbreviated below as p).

$focus(\langle \underline{D} \rangle([D](p))) \models_{cpt} focus(\langle \underline{L} \rangle(focus([D](p))))$

Supertimeline (CPT)

```
          |          A          |
                     |  *+  |
```

Timeline Index
A: $[D](run(Max))$ *: *Now* +: *Focus*

Subtimeline

```
          |  a  |    |  *+  |
```

Timeline Index
a: $[D](run(Max))$ *: *Now* +: *Focus*

Figure 35. Entailment check for *Max is running* $\Rightarrow$ *Max has run*

Entailment Check:
(The proposition $runto(Max, station)$ is abbreviated below as p).

$focus(\langle\underline{D}\rangle([D](p))) \models_{cpt}$
$focus(\langle\underline{L}\rangle([[EP]](p \wedge \langle\underline{A}\rangle(focus([D](p))) \wedge \langle A\rangle([D](\neg p)))))$

Supertimeline (CPT)

| A |

| *+ |

Timeline Index
A: $[D](runto(Max, station))$ *: *Now* +: *Focus*

Failed association propagation: $[D](\neg runto(Max, station))$

Figure 36. Entailment check for *Max is running to the station* $\not\Rightarrow$ *Max has run to the station*

6. In Conclusion

Two problem areas in the formal treatment of aspect concern the computational tractability of interval logics and the formalisation of a core set of aspectual structures that includes culminations and culminated processes. In this paper we have defined a computationally tractable augmented fragment of interval tense logic and defined logical representations of the aspectual structures under investigation. We have demonstrated computational tractability using two procedures, one for constructing the timelines of a formula and one for checking logical semantic entailments by comparing timelines. We have also shown how our aspectual structure definitions can be incorporated into a temporal semantics for a fragment of English and demonstrated checks for linguistic entailments as logical entailments between their temporal readings. Possible extensions to the logic fragment and timeline models were discussed as ways of providing an account for a wider range of linguistic behaviour.

References

1. Halpern, J. and Y. Shoham: 1986, 'A Propositional Modal Logic of Time Intervals'. In: *Proceedings of Symposium on Logic in Computer Science*. Cambridge, Massachusetts, pp. 279–292.

Entailment Check:
(The proposition $eat(Jack, dinner)$ is abbreviated below as p).

$\langle \underline{L} \rangle([[EP]](focus(p \wedge \langle \underline{A} \rangle([D](p)) \wedge \langle A \rangle([D](\neg p))))) \models_{cpt}$
$\langle \underline{L} \rangle([[EP]](focus(\langle \underline{A} \rangle([D](p)) \wedge \langle A \rangle([D](\neg p)))))$

Supertimeline (CPT)

$| \quad A \quad |_+ \; B \quad | \qquad | \quad * \quad |$

Timeline Index
A: $[D](eat(Jack, dinner))$ $\quad$ +: *Focus* $\quad$ +: $eat(Jack, dinner)$
B: $[D](\neg eat(Jack, dinner))$ $\quad$ *: *Now*

Subtimeline

$| \quad a \quad |_+ \; b \quad | \qquad | \quad * \quad |$

Timeline Index
a: $[D](eat(Jack, dinner))$ $\quad$ +: *Focus*
b: $[D](\neg eat(Jack, dinner))$ $\quad$ *: *Now*

Figure 37. Entailment check for *Jack finished eating dinner* ⇒ *Jack stopped eating dinner*

2. Kamp, J. A. W.: 1968, 'Tense Logic and the Theory of Linear Order'. Ph.D. thesis, University of California at Los Angeles.
3. Kent, S.: 1993, 'Modelling Events from Natural Language'. Ph.D. thesis, Imperial College, London.
4. Lascarides, A.: 1988, 'A Formal Semantic Theory of the Progressive'. Ph.D. thesis, University of Edinburgh.
5. Moens, M.: 1987, 'Tense, Aspect and Temporal Reference'. Ph.D. thesis, University of Edinburgh.
6. Moens, M. and M. Steedman: 1988, 'Temporal Ontology and Temporal Reference'. *Computational Linguistics 14* pp. 15–28.
7. Øhstrøm, P. and P. Hasle: 1995, *Temporal Logic*. Kluwer Academic Publishers.
8. Prior, A.: 1967, *Past, Present and Future*. Oxford University Press.
9. Prior, A.: 1968, 'Now'. *Noûs* **2**, 101–119.
10. Pulman, S.: 1997, 'Aspectual Shift as Type Coercion'. *Transactions of the Philological Society, 95:2.*
11. Reichenbach, H.: 1947, *Elements of Symbolic Logic*. Macmillan, London.
12. Richards, B., I. Bethke, J. van der Does, and J. Oberlander: 1989, *Temporal Representation and Inference*. Academic Press.
13. Steedman, M.: 1997, *Temporality in Natural Language*. Tutorial notes, ICTL 1997, Manchester, England.

Entailment Check:
(The proposition $eat(Jack, dinner)$ is abbreviated below as p).

$\langle \underline{L} \rangle([[EP]](focus(\langle \underline{A} \rangle([D](p)) \wedge \langle A \rangle([D](\neg p))))) \models_{cpt}$
$\langle \underline{L} \rangle([[EP]](focus(p \wedge \langle \underline{A} \rangle([D](p)) \wedge \langle A \rangle([D](\neg p)))))$

Supertimeline (CPT)

$|\quad \mathrm{A} \quad |_{+} \ \mathrm{B} \quad | \qquad | \quad * \quad |$

Timeline Index
A: $[D](eat(Jack, dinner))$ +: *Focus*
B: $[D](\neg eat(Jack, dinner))$ *: *Now*

Subtimeline Pattern Failure 1

$|\quad \mathrm{a} \quad |_{b} \qquad | \quad \mathrm{c} \quad | \qquad | \quad * \quad |$

Timeline Index
a: $[D](eat(Jack, dinner))$ b: $eat(Jack, dinner)$
c: $[D](\neg eat(Jack, dinner))$ *: *Now*

Other pattern failures are omitted.

Figure 38. Entailment check for *Jack stopped eating dinner* $\not\Rightarrow$ *Jack finished eating dinner*

14. Vendler, Z.: 1967, 'Verbs and Times'. *Linguistics in Philosophy.*
15. Venema, Y.: 1990, 'Expressiveness and Completeness of an Interval Tense Logic'. *Notre Dame Journal of Formal Logic* **31**(4), 529–547.
16. Verkuyl, H.: 1993, *A theory of aspectuality.* Cambridge University Press.