Between the Event Calculus and Finite State Temporality

Derek Kelleher, Tim Fernando(✉), and Carl Vogel

Trinity College Dublin, Dublin 2, Ireland
kellehdt@tcd.ie, {Tim.Fernando,vogel}@cs.tcd.ie

Abstract. Event Calculus formulas dealing with instantaneous and continuous change are translated into regular languages interpreted relative to finite models. It is shown that a model over the real line for a restricted class of these Event Calculus formulas (relevant for natural language semantics) can be transformed into a finite partition of the real line, satisfying the regular languages. Van Lambalgen and Hamm's treatment of type coercion is reduced to changes in the alphabet from which the strings are formed.

Keywords: Event Calculus · Finite State Temporality · Type coercion

1 Introduction

An important application of the Event Calculus (EC), originally developed by Kowalski and Sergot [14], has been Lambalgen and Hamm's treatment of Event Semantics [15] (hereafter referred to as VLH). The EC concentrates on change in time-dependent properties, known as fluents, formalizing both instantaneous change (due to punctual events), and continuous change (under some force). The underlying model is taken to be a continuum, the real line, with a predicate (HoldsAt) interpreting fluents relative to points (real numbers). This is in contrast to the common practice since Bennett and Partee [3] of evaluating temporal propositions over intervals.

Another approach to event semantics, which uses finite-state methods, is Finite State Temporality (FST). In recent years FST has been shown to be applicable to a diverse range of linguistic phenomena, such as Dowty's Aktionsart [9]. Strings are taken from an alphabet made up of subsets of a finite set of fluents (Σ), shown visually as boxes containing the fluents. Intuitively, fluents in the same box hold at the same time, and fluents in subsequent boxes hold of subsequent times. For example, the event of John going from his home to the library could be represented as the string:

at(john,home)		at(john,library)

(1)

the empty box above symbolising that there is an interval of time between John being at home, and being at the library, with no commitment made to what

A. Foret et al. (Eds.): FG 2015/2016, LNCS 9804, pp. 112–129, 2016.
DOI: 10.1007/978-3-662-53042-9_7

occurred during this interval. A fuller representation could add to that box fluents representing John's journey to the library, and split it into various boxes representing various stages of this journey.

It should be noted that while both the EC and FST talk of "fluents", there are some crucial differences between the two formalisms in their interpretation of this notion. In the EC, a fluent is a "time-dependent property" in that a fluent holding of an instant indicates that property holds of that instant. For example, if the fluent "building" holds of time t, then there is some building going on at time t. Events have a different status, being the entities that initiate or terminate the fluents (cause them to become true or false), and temporal instants form another ontological category. In FST, "fluents" correspond to EC-fluents, events, temporal instants, and complex entities built from these. These entities are differentiated by their various properties. For example, FST-fluents corresponding to events are not homogeneous, they hold only of particular intervals, and not any subintervals of that interval, whereas FST-fluents corresponding to EC-fluents are homogeneous.

Arising from the differences between these two formalisms is the question of whether the whole continuum is needed, or whether some finite representation will suffice for natural language semantics. Specifically, is anything lost by translating models for EC predicates that deal with instantaneous and continuous change, with the real line as the domain, into models for FST representations of these predicates, where the domain is a finite partition of the real line. The FST representations of EC predicates will be languages whose various strings represent different temporal granularities consistent with the EC predicates. A major advantage of this approach is that entailment can be expressed in terms of set-theoretic inclusions between languages [7], which is a decidable question for regular languages. Restricting our representations to regular languages ensures the computability of these entailments.

One application of the EC to event semantics is representing type coercion. Since Vendler [19], it is common to assign verb phrases or eventualities to certain categories such as "achievement" (punctual with consequent state), or "activity" (protracted with no associated end point). Under certain circumstances, however, eventualities typically associated with one category can behave as if in another category. For example, "sneeze" is usually considered to be a "point" (punctual with no consequent state). However, when it occurs with the progressive ("John was sneezing"), it can no longer be viewed as a punctual occurrence and is coerced into an activity (becoming a process of multiple sneezes by iteration).

Taking the concepts of instantaneous and continuous change from the EC, the possibility of representing type coercion in FST will be explored. The main idea is to associate coercion with a "change in perspective", implemented through a change in the alphabet from which strings are drawn.

Section 2 gives a brief introduction to Finite State Temporality. Section 3 will show how models for the EC-predicates can be translated into models for the FST-languages, and vice-versa. It also will show that if an EC-model satisfies an EC-predicate, the translated FST-model will satisfy the translated

FST-language, and vice-versa. Section 4 will address type coercion, and will show how this can be implemented in FST as changes in Σ.

2 Finite State Temporality

Fix a finite set Φ of fluents. The alphabet from which strings are drawn is made up of subsets of Φ:

$$\Sigma = 2^{\Phi} \tag{2}$$

For example, if $\Phi = \{f, g\}$, then $\Sigma = \{\{\}, \{f\}, \{g\}, \{f, g\}\}$. To emphasise that the elements of this alphabet are being used as symbols, and to improve readability, the traditional curly brackets are replaced with boxes enclosing the fluents, with the empty set becoming $\boxed{}$.

A typical string over this alphabet might be $\boxed{f, g}\boxed{}\boxed{f}$ (instead of $\{f, g\}\{\}\{f\}$), while a typical language might look like $\boxed{f, g}^*$, consisting of the strings ϵ, $\boxed{f, g}$, $\boxed{f, g}\boxed{f, g}$, $\boxed{f, g}\boxed{f, g}\boxed{f, g}$, etc.

Fernando [9] shows how a set of temporal points can be partitioned into a finite set of intervals, with a satisfaction relation holding between intervals and fluents. Since the EC takes the positive real line ($\mathbb{R}_+$) as its domain, for the purposes of this paper it is natural to take this as the set of temporal points to be partitioned. A model for FST consists of a segmentation of $\mathbb{R}_+$, and an interpretation function, $[\![]\!]_{FST}$, mapping a fluent to the set of intervals which satisfy it.

A segmentation is a sequence $\mathbb{I} = I_1 \ldots I_n$ of intervals such that:

$$I_i \text{ m } I_{i+1} \text{ for } 1 \leq i < n \tag{3}$$

Here, "m" is the relation "meets", defined by Allen [1] as: A meets B if and only if A precedes B, there is no point between A and B, and A and B do not overlap. $\mathbb{I}$ is a segmentation of I (in symbols: $\mathbb{I} \nearrow I$) if and only if $I = \bigcup_{i=1}^{n} I_i$.

While there are various ways to partition the real line, one particular form of segmentation is especially useful where the EC is concerned:

$$[0, b_1](a_2, b_2] \ldots (a_{n-1}, b_{n-1}](a_n, \infty), \text{ where } \forall i \;\; a_{i+1} = b_i \tag{4}$$

So $[0, b_1]$ includes every point between 0 and b_1, including both 0 and b_1 (closed at both ends), whereas $(a_i, b_i]$ includes every point between a_i and b_i, including b_i, but not a_i (open at the beginning, closed at the end).

The reason for choosing this form of segmentation (where I_1 is closed at both ends, but $I_2, \ldots, I_n$ are open at the beginning and closed at the end) is as follows: In FST, one use of fluents is to represent states (though they can also represent events or "times", see Sect. 2.1). The interpretation of these stative fluents will be a set of intervals from the segmentation. Intuitively, if an interval I is in the interpretation of a fluent f, then f holds across the interval I.

Many temporal reasoning problems specify initial conditions: states or properties which hold at the beginning of time (EC's Initially predicate). These fluents hold from time-point 0 onwards, which requires an interval which includes 0 in their interpretation. This interval will have the form $[0, s]$ for some $s \in \mathbb{R}_+$.

Other states are "brought into being" by events. The assumption in EC (and common in the literature [2,11]) is that these states hold after the event, not during it. So, if an event happens at time point a_i and brings a state f into being, then an interval in the interpretation of f will not include a_i, but will include the points after it, giving an interval that is open at the beginning.

If an event at b_i causes state f to cease holding, the effect will be seen after b_i, meaning that f still holds at b_i but not at any point after. So b_i will be in an interval that is in the interpretation of f, and this interval will be closed at the end. Any fluent caused to hold by an event, and subsequently caused to stop holding by another event will therefore have an interval that is open at the beginning and closed at the end in its interpretation (with inertia causing it to hold at all points within the interval).

A satisfaction relation $\models_{FST}$ is defined between fluents and intervals:

$$\mathrm{I} \models_{FST} \mathrm{f} \Longleftrightarrow \mathrm{I} \in [\![\mathrm{f}]\!]_{FST} \tag{5}$$

This can be extended to a relation holding between segmentations and strings:

$$\mathrm{I}_1 \ldots \mathrm{I}_n \models_{FST} \alpha_1 \ldots \alpha_n \Longleftrightarrow (\forall \mathrm{f} \in \alpha_\mathrm{i})\ \mathrm{I_i} \models_{\mathrm{FST}} \mathrm{f}, \text{ for } 1 \leq \mathrm{i} \leq \mathrm{n} \tag{6}$$

and further extended to a relation holding between segmentations and languages:

$$\mathrm{I}_1 \ldots \mathrm{I}_n \models_{FST} \mathrm{L} \Longleftrightarrow (\exists \mathrm{s} \in \mathrm{L})\ \mathrm{I}_1 \ldots \mathrm{I_n} \models_{\mathrm{FST}} \mathrm{s} \tag{7}$$

2.1 States, Events, and Times

As noted in the introduction, the EC formalises two notions of change, instantaneous change due to punctual events, and continuous change due to some force. These distinctions may seem more at home in the field of physics than linguistics but, as will be seen later, linguistic "eventualities" can be represented as complex sequences of change, both instantaneous and continuous.

Before continuing, some comments must be made on differences between the EC's and FST's ontologies, and how they are related. In the EC, a distinction is drawn between "fluents", which are time-dependent properties that "hold" of time-points, and events, which are punctual entities that "happen" at time-points, causing an instantaneous change by "initiating" (causing to hold), or "terminating" (causing to cease to hold) fluents. Temporal points make up a third category, represented by the positive real numbers (and 0).

In FST, the word "fluent" encompasses these three categories, though their fluents have different properties. Stative fluents represent states/properties. They are homogeneous. A fluent is homogeneous if for all intervals I and I′ such that $\mathrm{I} \cup \mathrm{I}'$ is an interval:

$$\mathrm{I} \models_{FST} \mathrm{f} \text{ and } \mathrm{I}' \models_{FST} \mathrm{f} \Longleftrightarrow \mathrm{I} \cup \mathrm{I}' \models_{FST} \mathrm{f} \tag{8}$$

Essentially, if a stative fluent holds over an interval, and that interval is split into two intervals, the fluent will hold over both intervals. If John slept from 2pm to 4pm, then John slept from 2pm to 3pm, and John slept from 3pm to 4pm (and vice-versa).

Event fluents represent events. Depending on the event they may be homogeneous or "whole". Fernando [9] defines a fluent as being whole if for all intervals I and I′ such that $\mathrm{I} \cup \mathrm{I}'$ is an interval:

$$\mathrm{I} \models_{FST} \mathrm{f} \text{ and } \mathrm{I}' \models_{FST} \mathrm{f} \text{ implies } \mathrm{I} = \mathrm{I}' \tag{9}$$

The EC makes use of an explicit timeline, necessitating the use of "temporal fluents" in FST to represent this timeline. For every possible interval (including points), there is a "temporal fluent" which marks this interval, i.e. every interval has a fluent whose interpretation is that interval. For simplicity of presentation, we use the same symbol for the fluent as we use for the interval or the point that it marks. It should be noted that the interval $(\mathrm{a}, \mathrm{b}]$ is a complex symbol built from brackets, a comma, and numbers/variables, while the fluent "$(\mathrm{a}, \mathrm{b}]$" is an atomic symbol. This allows statements of the form $\{t\} \models t$ to be used.

3 Translating Language and Models

VLH (p.41) sketch an "intuitively appealing class of models of EC" with domain $\mathbb{R}_+$. The interpretation of fluents is given by the model, and is taken to be a set of intervals of the form $(\mathrm{a}, \mathrm{b}]$ $(\mathrm{a}, \mathrm{b} \in \mathbb{R}_+)$, or $[0, \mathrm{a}]$ $(\mathrm{a} \in \mathbb{R}_+)$. The intervals are, in a sense, "maximal". They are the longest contiguous stretches for which f holds. If an interval is in the interpretation of a fluent f, f will hold for time-point in that interval, but the time-point itself will not be in the interpretation of f.

It cannot be assumed that all models of the EC give rise to finite segmentations of the real line. VLH give the example of an initiating event for a fluent f happening a time t iff $\mathrm{t} \in \mathbb{Q}$, with terminating events happening at a time t' iff $\mathrm{t}' \in \mathbb{R} - \mathbb{Q}$. This will lead to a fluent varying infinitely between holding and not holding over any interval, a situation which rules out a finite segmentation.

One necessary condition for there to be a finite segmentation is that each fluent is "alternation bounded" [8]. Intuitively this rules out the above case of a fluent varying infinitely between holding and not holding over some interval. We say a fluent is alternation bounded if the set of points or intervals in its extension (U) is alternation bounded. To define when a set of points U taken from a linear order $(T, <)$ is alternation bounded, the following notions are useful.

Given a subset U of T, a *segmentation of* U is a sequence $\mathbb{I} = I_1 \cdots I_n$ of intervals such that $U = \bigcup_{i=1}^{n} I_i$ and $I_i < I_{i+1}$ for $1 \leq i < n$.

A subset I of T is *U-homogeneous* if

$$(\exists t \in I)\ t \in U \iff (\forall t \in I)\ t \in U$$

— i.e.,

$$I \subseteq U \text{ or } I \cap U = \emptyset.$$

A *U-segmentation* is a segmentation $I_1 \cdots I_n$ of T such that for all i between 1 and n, I_i is U-homogeneous and when $i < n$,

$$I_i \subseteq U \iff I_{i+1} \cap U = \emptyset.$$

Given a subset U of T and a positive integer $n > 0$, an (n,U)*-alternation* is a string $t_1 t_2 \cdots t_n \in T$ of length n such that for $1 \leq i < n$, $t_i < t_{i+1}$ and

$$t_i \in U \iff t_{i+1} \notin U.$$

U is *alternation bounded* (a.b.) if for some positive integer n, there is no (n,U)-alternation. It will also be useful to define the equivalence relation

$$\begin{aligned} t \sim_U t' &\iff \text{there is an } (n,U)\text{-alternation with both } t \text{ and } t' \text{ only if } t = t' \\ &\iff [t,t'] \cup [t',t] \text{ is } U\text{-homogeneous} \end{aligned}$$

Lemma For any subset U of T,

$$\text{there is a } U\text{-segmentation} \iff U \text{ is a.b.}$$

For $\Rightarrow$, a U-segmentation of length n implies there can be no $(n+1,U)$-alternation. Conversely, let n be the largest positive integer for which there is an (n,U)-alternation. Let $t_1 \cdots t_n$ be an (n,U)-alternation, and define $I_1 \cdots I_n$ by

$$I_i := \{t \in T \mid t_i \sim_U t\}$$

Then $I_1 \cdots I_n$ is a U-segmentation.

As well as the traditional axioms of the EC, VLH include formulas in the EC, together known as a scenario, which further constrain the possible models. A formula in the scenario with the following form:

$$S(t) \Longrightarrow Happens(e,t) \tag{10}$$

where S(t) is a first-order formula built from:

1. literals of the form $(\neg)\mathrm{HoldsAt(f,t)}$
2. equalities between fluent terms, and between event terms
3. formulas in the language of the structure $(\mathbb{R}, <; +, \times, 0, 1)$

can cause a fluent to vary infinitely, as can be seen in the below example:

$$\begin{aligned} &HoldsAt(f,t) \Longrightarrow Happens(e_1, t+1) \\ &\neg HoldsAt(f,t) \Longrightarrow Happens(e_2, t+1) \\ &Initiates(e_2, f, t) \\ &Terminates(e_1, f, t) \\ &HoldsAt(f, 0) \end{aligned} \tag{11}$$

Of course, in this particular example, the fluent can only vary infinitely over an interval stretching to infinity, and while arguments can be made that this is an unrealistic condition for natural language semantics, the formulation of the EC does allow it.

Due to the inertial axioms, when a fluent is initiated, it holds until it is terminated. An initiating event has no effect on a fluent that already holds, and similarly, a terminating event has no effect on a fluent that does not hold. It follows that an infinite variation of the holding of a fluent over an interval can only arise if there is an infinite sequence of alternating initiating and terminating events.

A further necessary condition for there to exist a finite segmentation of an EC-model is that we are dealing with only a finite number of these alternation bounded fluents. Each of these fluents defines a finite segmentation, and taking a finite number of these fluents together and intersecting the intervals in their segmentations gives a finite segmentation. It will be seen in Sect. 4 on type coercion that eventualities are defined using a finite number of fluents, and type coercion involves adding or removing a finite number of fluents, allowing us to only deal with finite segmentations. Hereafter, when we refer to an "EC-model" it is understood we are discussing models consistent with these conditions:

1. The inferences of interest involve only a finite number of fluents.
2. Each fluent is alternation bounded.

An FST-model can be formed from an EC-model as follows:

1. For each fluent f in EC, form $[\![f]\!]_{EC \to FST} : I \in [\![f]\!]_{EC}$ implies $(\forall I' \subseteq I) I' \in [\![f]\!]_{EC \to FST}$
2. Choose a segmentation $\mathbb{I}$ of $\mathbb{R}_+$
3. Form $[\![f]\!]_{FST}$ by relativizing $[\![f]\!]_{EC \to FST}$ to $\mathbb{I} = I_1 \ldots I_n : I_i \in [\![f]\!]_{FST}$ iff $I_i \in [\![f]\!]_{EC \to FST}$. Only those intervals that are part of the segmentation, $\mathbb{I}$, can be in $[\![f]\!]_{FST}$.

An EC-model can be formed from an FST-model as follows:

1. Suppose $\mathbb{I} \nearrow \mathbb{R}_+$. Find a sequence $I_i \ldots I_j$ in $\mathbb{I}$ such that for all $(i \leq q \leq j)$ $I_q \in [\![f]\!]_{FST}, I_{i-1} \notin [\![f]\!]_{FST}$ and $I_{j+1} \notin [\![f]\!]_{FST}$. This sequence will be the longest unbroken stretch for which f holds.
2. Put $I = \bigcup_{r=i}^{j} I_r$ in $[\![f]\!]_{EC}$.

3.1 Initially

In the EC, Initially(f) signifies that the fluent f "was true at the beginning of time" (VLH p.38). It can be translated as follows:

$$\mathrm{L}_{\mathrm{Initially(f)}} = \langle \trianglerighteq \rangle \boxed{\mathrm{f}}\boxed{}^{*} \tag{12}$$

Every string in this language includes the fluent f in its first box/symbol. Understanding the above equation requires two definitions:

$$s \in \langle R \rangle L \Longleftrightarrow (\exists s' \in L)\ s\ R\ s' \tag{13}$$

where R is some relation between strings. Every string in ⟨R⟩L bears the relation R to some string in L.

$\trianglerighteq$ is the relation "subsumes". If s and s′ are strings, where $s = \alpha_1 \dots \alpha_n$ and $s' = \beta_1 \dots \beta_k$ then:

$$s \trianglerighteq s' \text{iff } n = k, \text{ and for every } i,\ \alpha_i \supseteq \beta_i \tag{14}$$

So every symbol of s contains all the fluents (information) of the corresponding symbol of s′, and possibly more. Fernando [6] shows that this relation can be computed using finite-state methods.

According to VLH (p.42), a model for Initially(f) will have, in its interpretation of f, an interval that begins at 0:

$$Initially := \{f | (\exists s > 0)[0, s] \in [\![f]\!]_{EC}\} \tag{15}$$

In FST, a model for $L_{\text{Initially(f)}}$ is a segmentation of $\mathbb{R}_+$, where the first interval is in the interpretation of f:

$$I_1 \dots I_n \nearrow \mathbb{R}_+ \text{ and } I_1 \in [\![f]\!]_{FST} \tag{16}$$

EC → FST: If there is an EC-model for Initially(f) then, by (15), there is some $s \in \mathbb{R}$ where $[0, s] \in [\![f]\!]_{EC}$. By construction of an FST-model, $[0, s]$ and all its subintervals are in $[\![f]\!]_{EC \to FST}$. Any segmentation where $I_1 = [0, r]$, with $r \leq s$, will have $I_1 \in [\![f]\!]_{FST}$, satisfying (16).

FST → EC: For an FST-model of $L_{\text{Initially(f)}}$ where $I_1 \dots I_n \nearrow \mathbb{R}_+$ and $I_1 \in [\![f]\!]_{FST}$ there exists some j with 1<j≤n such that $I_j \notin [\![f]\!]_{FST}$. By construction of an EC-model, $I = \bigcup_{r=1}^{j-1} I_r \in [\![f]\!]_{EC}$. Since I_1 is included in I, I begins at 0 and therefore is of the form $[0, s]$ ($s \in \mathbb{R}$), which fulfills condition (15).

3.2 Instantaneous Change: Happens and Initiates

When dealing with events and their effects on fluents, some new fluents will prove helpful:

$$I \models_{FST}\ )_e \Longleftrightarrow (\forall J \text{ such that } I \text{ m } J) I \not\models_{FST}\ f \text{ and } J \models_{FST} f \tag{17}$$

$$(a, b] \models_{FST}\ \langle end \rangle t \Longleftrightarrow \{b\} \models_{FST} t \tag{18}$$

$$(a, b] \models_{FST}\ \langle start \rangle t \Longleftrightarrow \{a\} \models_{FST} t \tag{19}$$

The fluent $)_e$ is used to mark the end of events. In the EC, events are punctual: they occur at points (interpreted as real numbers). In FST, events are allowed to occur over intervals. The reason for this will become clear when type coercion is dealt with. If $\langle end\rangle t$ holds of an interval, then that interval ends at the time point t. If $\langle start\rangle t$ holds of an interval, then that interval starts after the time point t.

The EC treats change due to an event as instantaneous. A fluent initiated by a punctual event holds AFTER, but not AT, the time at which the event occurs. When moving from punctual events to events that happen over intervals, a choice must be made as to when the change occurs. In keeping with the EC, FST formalizes the change as happening after the event ends. Only initiating events are dealt with below, the definitions for terminating events are broadly similar. Note that the fluents f and $)_e$ are linked in that this particular event initiates the fluent f, other events initiate other fluents.

Happens(e, t), which signifies that event e occurs at time t, can be translated as follows:

$$\mathrm{L}_{\mathrm{Happens(e,t)}} = \langle \sqsupseteq \rangle \boxed{(_e} \boxed{}^{*} \boxed{\langle end\rangle t,)_e} \tag{20}$$

If $\mathrm{s} = \alpha_1 \ldots \alpha_n$, $\mathrm{s} \sqsupseteq \mathrm{s}'$ iff there is some substring r of s, $\alpha_i \ldots \alpha_j$, and $\mathrm{r} \trianglerighteq \mathrm{s}'$. $(_e$ represents the start of the event. It cannot be translated from the EC as the EC treats events as punctual. Its purpose in FST is to allow punctual events to be "stretched" and given an internal structure. It causes no problem in translations as $)_e$ is the necessary fluent to represent instantaneous change.

VLH (p.42) define the extension of Happens as follows:

$$Happens := \{(e,t) | (\exists f)(f,t) \in e\} \tag{21}$$

In an EC-model, the event e will either initiate or terminate some fluent f. If it is an initiating event then there must be some $\mathrm{s} \in \mathbb{R}$ for which $(\mathrm{t}, \mathrm{s}] \in [\![f]\!]_{EC}$.

In FST, a model for $\mathrm{L}_{\mathrm{Happens(e,t)}}$ will be a segmentation $I_1 \ldots I_n \nearrow \mathbb{R}_+$ with the following conditions:

$$(\exists I_i, I_j) I_i < I_j \text{ and } I_i \models_{FST} (_e \text{ and } I_j \models_{FST})_e \text{ and } I_j \models_{FST} \langle end\rangle t \tag{22}$$

Note that $<$ is the relation "precedes". For intervals I and J, I precedes J if its end point is before J's start point.

Using the convention $\{t\} \models_{FST} t$, and the definition of $\langle end\rangle t$, then $I_j = (a_j, \mathrm{t}]$ for some $a_j \in \mathbb{R}$. From the definition of $)_e$, $(a_j, \mathrm{t}] \not\models_{FST}$ f and $(\mathrm{t}, b_{j+1}] \models_{FST}$ f for some $b_{j+1} \in \mathbb{R}$.

EC → FST: From above ($\exists$ s) $(\mathrm{t}, \mathrm{s}] \in [\![f]\!]_{EC}$. Because the intervals in the interpretation of fluents are maximal, there must be an interval (at least a point) that meets (t, s] which is not in the interpretation of f. Therefore, $(\mathrm{r}, \mathrm{t}] \notin [\![f]\!]_{EC}$ for some $\mathrm{r} \in \mathbb{R}$. Constructing an FST-model from this, (t, s] and all its subintervals are in $[\![f]\!]_{EC\rightarrow FST}$, and there is some $\mathrm{q} \in \mathbb{R}$ for which (q, t] and all its subintervals are not in $[\![f]\!]_{EC\rightarrow FST}$. An example segmentation would be $[0, \mathrm{b}_1] \ldots (\mathrm{q}, \mathrm{t}](\mathrm{t}, \mathrm{s}] \ldots (\mathrm{a}_n, \infty)$. $[\![f]\!]_{EC\rightarrow FST}$ relativized to this will have (q, t]

$\notin [\![f]\!]_{FST}$, $(t,\, s] \in [\![f]\!]_{FST}$ satisfying the conditions above with $a_j = q$, and $b_{j+1} = s$. (Again there will be many segmentations of $\mathbb{R}_+$ that satisfy this language).

FST → EC: For an FST-model of $L_{\text{Happens(e,t)}}$ where $I_1 \ldots I_n \nearrow \mathbb{R}_+$ and $I_j \notin [\![f]\!]_{FST}$ and $I_{j+1} \in [\![f]\!]_{FST}$, there exists some r with j+1<r≤n such that $I_r \notin [\![f]\!]_{FST}$. By construction of an EC-model $I = \bigcup_{d=j+1}^{r-1} I_d \in [\![f]\!]_{EC}$. Since I_{j+1} is the first interval in I, I is of the form (t, s], satisfying the condition above for an EC-model.

Initiates(e, f, t) can be translated as follows:

$$L_{\text{Initiates(e,f,t)}} = \boxed{\langle end\rangle t,)_e}\boxed{} \Longrightarrow \boxed{}\boxed{\text{f}} \tag{23}$$

This is encoded as a constraint on strings. The constraint $L \Longrightarrow L'$ is the set of strings s, such that every stretch of s that subsumes L also subsumes L′. The notion of "stretched" is formalized as follows: s′ is a factor of s if s = us′v for some (possibly empty) strings u and v.

$$s \in L \Longrightarrow L' \Longleftrightarrow \text{ for every factor } s' \text{ of } s, s' \trianglerighteq L \text{ implies } s' \trianglerighteq L' \tag{24}$$

The constraint for $L_{\text{Initiates(e,f,t)}}$ says that every string in this language that has a box containing $\langle end\rangle t$ and $)_e$, must contain f in the following box.

An EC-model for Initiates(e, f, t) must have the following condition: $(\exists s \in \mathbb{R})(t,\, s] \in [\![f]\!]_{EC}$.

In FST, a model for $L_{\text{Initiates(e,f,t)}}$ will be a segmentation $I_1 \ldots I_n \nearrow \mathbb{R}_+$ with the following conditions:

$$(\forall I_i, I_j)[I_i \text{ m } I_j \text{ and } I_i \models_{FST})_e \text{ and } I_i \models_{FST} \langle end\rangle t] \longrightarrow I_j \models_{FST} f \tag{25}$$

EC → FST: From above (∃ s) (t, s]$\in [\![f]\!]_{EC}$. For the same reason as above for the Happens predicate, (t, s] and all its subintervals are in $[\![f]\!]_{EC\rightarrow FST}$, and there is some q ∈ ℝ for which (q, t] and all its subintervals are not in $[\![f]\!]_{EC\rightarrow FST}$. Taking as an example segmentation, [0, b_1]...(q, t](t, s]...(a_n,∞), the interval (q, t] meets (t, s], it satisfies $\langle end\rangle t$, it also satisfies $)_e$ (by definition of $)_e$). Therefore it meets the conditions of the antecedent of (25), and since it is given that (t, s] $\in [\![f]\!]_{EC}$, and therefore in $[\![f]\!]_{FST}$, the consequent is also true, fulfilling the conditions for an FST-model.

FST → EC: Take an FST-model of $L_{\text{Initiates(e,f,t)}}$ where $I_1 \ldots I_n \nearrow \mathbb{R}_+$ and (25) holds. Given that there is only one interval I_i in this segmentation which satisfies $\langle end\rangle t$, and only one interval I_j that it meets, we can reduce (25) to:

$$(a_i, t] \models_{FST})_e \longrightarrow (t, b_j] \models_{FST} f \tag{26}$$

By definition of $)_e$, (a_i,t]$\not\models_{FST}$ f. (26) then becomes:

$$(a_i, t] \not\models_{FST} \ f \longrightarrow (t, b_j] \models_{FST} \ f \tag{27}$$

there exists some r with j+1<r≤n such that $I_r \notin [\![f]\!]_{FST}$. By construction of an EC-model, $I = \bigcup_{d=j}^{r-1} I_d \in [\![f]\!]_{EC}$. Since I_{j+1} is the first interval in I, I is of the form $(\mathrm{t}, \mathrm{s}]$, satisfying the condition above for an EC-model.

3.3 Continuous Change: Trajectory

Trajectory$(f_1, t, f_2, t+d)$ signifies that if fluent f_1 holds between t and t + d, then fluent f_2 will hold at t + d. In VLH, f_2 is generally a real-valued function describing continuous change, so the Trajectory predicate describes continuous change as long as fluent f_1 holds. It can be translated as follows:

$$\mathrm{L}_{\mathrm{Trajectory(f_1,t,f_2,t+d)}} = \boxed{\langle start\rangle t, f_1}\boxed{f_1}^{*}\boxed{\langle end\rangle t+d, f_1} \Longrightarrow \boxed{}^{*}\boxed{\langle end\rangle f_2} \tag{28}$$

While VLH does not give a model for Trajectory, it is not hard to construct an intuitive model consistent with the other predicates:

$$\begin{aligned} Trajectory := \{(f_1, t, f_2, t+d) | (\exists a, b \in \mathbb{R}_+) \ a \le t \le t+d \le b \text{ and} \\ (a, b] \in [\![f_1]\!]_{EC} \longrightarrow (\exists I \in [\![f_2]\!]_{EC}) t+d \in I\} \end{aligned} \tag{29}$$

A FST-model for $\mathrm{L}_{\mathrm{Trajectory(f_1,t,f_2,t+d)}}$ will be a segmentation of $\mathbb{R}_+$ with the following conditions:

$$(\exists I_i = (t, b_i], I_j = (a_j, t{+}d])(\forall i \le p \le j) I_p \models_{FST} \ f_1 \longrightarrow \{t{+}d\} \models_{FST} \ f_2 \tag{30}$$

EC → FST: If the antecedent of (30) is true, then the sequence $I_1 \dots I_j$ is part of an unbroken stretch over which f holds. Let the start and end points of this stretch be a and b from (29). Therefore, (a,b] $\in [\![f]\!]_{EC}$, fulfilling the antecedent of the conditional contained in (29). Now from the consequent of (29), there is an interval I $\in [\![f_2]\!]_{EC}$, so all its subintervals, notably {t + d}, are in $[\![f_2]\!]_{EC\to FST}$. By definition of $\langle end\rangle t + d$, $(\mathrm{a}_j, t+d] \in [\![\langle end\rangle f_2]\!]_{EC\to FST}$. Relativized to the segmentation given, $(\mathrm{a}_j, t+d] \in [\![\langle end\rangle f_2]\!]_{FST}$. Therefore {t + d}$\models_{FST} \ f_2$.

FST → EC: For an FST-model of $\mathrm{L}_{\mathrm{Trajectory(f_1,t,f_2,t+d)}}$ where $I_1 \dots I_n \nearrow \mathbb{R}_+$ and, supposing the antecedent contained in (29) is true, $(t, b_i], (a_j, t+d]$ and all the intervals between them are in $[\![f_1]\!]_{FST}$ (by construction of an FST-model. Therefore, the consequent of (30) is true, so {t + d}$\models$ f_2, so {t + d} $\in [\![f_2]\!]_{FST}$. {t + d} is either a maximal interval for which f_2 holds or is a subinterval of that maximal interval. Either way, there is some I $\in [\![f_2]\!]_{EC}$ with t+d ∈ I.

While VLH (p.45) require a real-valued function of time in place of f_2, and while this can be implemented in FST as long as only a finite number of values from this function are used as fluents, it is not clear that natural language semantics needs the precision of this real-valued function to address continuous change. As will be seen in the next section, when dealing with type coercions that involve adding elements to an event structure, it is not always clear in advance what form these elements will have. An alternative in FST to the above representation of continuous change is as follows:

$$\boxed{\mathrm{f_1}} \Longrightarrow \boxed{\mathrm{f_{2\uparrow}}} \tag{31}$$

The above essentially says that if $\mathrm{f_1}$ is in a box, then $\mathrm{f_{2\uparrow}}$ is in the same box, or as long as $\mathrm{f_1}$ holds, $\mathrm{f_2}$ is increasing, or moving along some trajectory. For type coercion, it will be useful to have a fluent that holds when the end of the trajectory is reached, $\mathrm{f}_{2,MAX}$.

These fluents can be interpreted model-theoretically, relative to an underlying function, representing trajectory or path taken, in the model. This is in line with the approach of Kennedy and Levin [13], who propose that the semantics of degree achievements (such as “the soup cooled”) rely on a “difference function”, a function that measures the amount an object changes along a scalar dimension as a result of participating in an event. Certain functions describing change will have an end-point or maximum. This may be contextually given, as Fernando [10] assumes, or may be a natural feature of the scale against which change is measured. Closed scales (such as “smooth”) have a natural maximum, complete smoothness in this case [18].

3.4 Scenarios

The EC relates instantaneous and continuous change to natural language semantics through scenarios, which “state the specific causal relationships holding in a given situation” (VLH, p.43). Only the elements of the a scenario dealing with change will be dealt with here.

A scenario is a conjunct of statements of the form:

1. $\mathrm{Initially(f)}$
2. $\mathrm{S(t) \rightarrow Initiates(e, f, t)}$
3. $\mathrm{S(t) \rightarrow Terminates(e, f, t)}$
4. $\mathrm{S(t) \rightarrow Happens(e, t)}$
5. $\mathrm{S}(f_1, f_2, t, d) \rightarrow \mathrm{Trajectory(f_1, t, f_2, d)}$

where $\mathrm{S(t)}$ is a conjunction of statements of the form $\mathrm{HoldsAt(f, t)}$. Element 5 is referred to as the “dynamics”, relating fluent $\mathrm{f_1}$ (viewed as a force), to the change in $\mathrm{f_2}$ (viewed as a changing, partial object).

As with EC-predicates, each element of an EC-scenario can be translated into FST. If $\mathrm{S(t)}$ is not a part of an element (for instance if one element of the scenario is $\mathrm{Initiates(e, f, t)}$, as opposed to $\mathrm{S(t) \rightarrow Initiates(e, f, t)}$), then the equivalent element of the FST-scenario is the language given above as equivalent. If $\mathrm{S(t)}$ is part of an element then the equivalent will be a constraint relating fluents holding at a certain time to the language given above as equivalent. For example, $\mathrm{HoldsAt(f, t) \rightarrow Happens(e, t)}$ can be translated as follows:

$$\boxed{}^{*}\boxed{\langle end\rangle\ \text{t},\langle end\rangle\ \text{f}} \Longrightarrow \boxed{}^{*}\boxed{(_e}\boxed{}^{*}\boxed{)_e} \tag{32}$$

For the temporal fluents occurring in a string to have the correct ordering, one further set of constraints is needed. In an FST-model, every interval is ordered relative to every other interval by the "precedes" relation ($<$). A scenario will contain the set of constraints:

$$\forall I, I' \text{ in an FST-model such that } I < I' : \boxed{\text{I'}}\boxed{}^{*}\boxed{\text{I}} \Longrightarrow \emptyset \tag{33}$$

The above constraint says that if, in an FST-model, an interval I precedes an interval I′, then any string with the fluent I′ preceding I will not be in the language.

4 Type Coercion

Having developed the tools necessary to represent instantaneous and continuous change in FST, these are now applied to the natural language semantics problem of type coercion.

4.1 Eventualities vs. Events

The EC treats "events" as punctual occurrences that cause an instantaneous change (a fluent becomes initiated or terminated). However, most occurrences described as events have a more complex structure. Moens and Steedman [16] (hereafter referred to as M&S) have proposed a three-part event structure, called a "nucleus", which consists of a preparatory process, a culmination, and a consequent state. They note that all of these elements can be compound. The culmination "reaching the top of Mt. Everest" may have a number of processes such as climbing, eating lunch, etc. as part of its preparatory process, and there may be many consequent states.

Due to this, it is important to note that the discussion of coercion must have a quite general character. If adding a consequent state to an event structure, the question arises: what or which consequent state? For this reason, non-specific fluents such as f_3, g_1 etc. will often be used to describe "sailent' states or activities that have been added to an event structure.

The EC implements a similar event structure to that of M&S which they call "eventualities", having the following form: (f_1, f_2, e, f_3) where f_1 is an activity, f_2 is a fluent representing a partial object which changes under the influence of the activity, e is a culminating event, and f_3 is a consequent state.

Different approaches have used different terminologies for what are, essentially, the same categories. What Vendler [19] called activities, achievements, and accomplishments, M&S call processes, culminations, and culminated processes respectively.

4.2 Activity $\rightsquigarrow$ Accomplishments

VLH (p.171) discuss the case of "additive coercion", where a scenario is elaborated or added to. Alternative views of this type of coercion are given by Pulman [17] and M&S. Pulman sees this as supplementing a process with a consequent state (Pulman does not include culminating events in his ontology). The activity of "swimming" would have a salient consequent state added, perhaps as general as "has swum". M&S propose that the process is bundled into a point, with a new preparatory process and consequent state added. They cite the example "has John worked in the garden?" as only making sense if John working in the garden was part of some plan, with the preparatory process being whatever preparation was involved to work in the garden, the culmination being the working in the garden viewed as a point, and the consequent state perhaps being "has worked in the garden".

Both these views can be accounted for in FST. In the first, the activity f_1 is augmented with a fluent representing the change in the partial object (swimming augmented with the trajectory of the swim), a culminating event (finishing the swim), and a consequent state (has swum). In alphabet terms:

$$\Sigma \rightsquigarrow \Sigma' = \Sigma \cup \{f_{2\uparrow}, f_{2,MAX}, (_e,)_e, f_3\} \tag{34}$$

The scenario will also have to be augmented with general constraints of the form:

$$\boxed{\mathrm{f}_1} \Longrightarrow \boxed{f_{2\uparrow}} \tag{35}$$

$$\boxed{\mathrm{f}_{2,MAX}} \Longrightarrow \boxed{)_e} \tag{36}$$

$$\boxed{)_e}\boxed{} \Longrightarrow \boxed{}\boxed{f_3} \tag{37}$$

For M&S's coercion to be implemented, f_1, the original activity and internal structure of some event, is deleted, and fluents representing its start and end points are added, turning it into a point with no internal structure. A new preparatory process, g_1 is added, and a new consequent state, g_3. In alphabet terms:

$$\Sigma \rightsquigarrow \Sigma' = \Sigma \cup \{g_1, g_{2\uparrow}, g_{2,MAX}, (_e,)_e, g_3\} - \{f_1\} \tag{38}$$

The scenario will be elaborated as follows:

$$\boxed{\mathrm{g}_1} \Longrightarrow \boxed{g_{2\uparrow}} \tag{39}$$

$$\boxed{\mathrm{g}_{2,MAX}} \Longrightarrow \boxed{)_e} \tag{40}$$

$$\boxed{)_e}\boxed{} \Longrightarrow \boxed{}\boxed{g_3} \tag{41}$$

4.3 Achievements $\rightsquigarrow$ Accomplishments

As pointed out in M&S, the progressive needs a process (activity), or culminated process (accomplishment) as "input", so what to make of the following:

$$\text{John was reaching the top} \tag{42}$$

"Reach the top" is usually seen as a culmination (achievement), punctual with an associated consequent state. The progressive coerces this culmination into a culminated process by adding a preparatory process, and focussing on this.

For VLH (p.172), the achievement $(\text{-}, \text{-}, \mathrm{e}_1, \mathrm{f}_3)$, where e_1 would be the culmination (reaching the top), and f_3 would be the consequent state of this (perhaps being at the top), would be coerced into an accomplishment $(\mathrm{g}_1, \mathrm{g}_2, \mathrm{e}_2, \mathrm{g}_3)$, where g_1 and g_2 are "unknown parameters", depending on what preparatory process is being associated with the culmination. Presumably the culminations and consequent states of "reach the top" viewed as an achievement and as an accomplishment are the same. The addition of the fluents representing the preparatory process and changing partial object means a dynamics must be added to the scenario to relate these fluents.

To account for the addition of a preparatory process, and changing object in FST, the fluents marking the beginning and ending of events $(_e$ and $)_e$ are associated with a fluent prog_e which signifies that the event in question is ongoing:

$$I \models prog_e \Longleftrightarrow I \models (_e \text{ or } I \models)_e \text{ or } [(\exists I_i, I_j) I_i < I < I_j \text{ and} I_i \models (_e \text{ and } I_j \models)_e]$$

This fluent corresponds to g_1 from the EC. Its addition to the alphabet of FST will lead to a finer-grained segmentation, effectively giving the event, previously viewed as punctual, an internal structure. In alphabetic terms:

$$\Sigma \rightsquigarrow \Sigma' = \Sigma \cup \{prog_e, g_{2\uparrow}, g_{2,MAX}, (_d,)_d, g_3\} - \{f_3\} \tag{43}$$

4.4 Accomplishments $\rightsquigarrow$ Activities

M&S discuss the case of a culminated process being coerced into a process in the presence of the progressive:

$$\text{Roger was running a mile} \tag{44}$$

For this to happen, the culmination and consequent state must be "stripped off", leaving the preparatory process.

In VLH (p.172), this process is described as subtractive coercion, essentially removing the last two elements from the event-nucleus, and removing statements relating to the culmination and consequent state from the scenario. The same can be achieved in FST by removing those constraints from the scenario that "cause" the culminating event to happen at the maximum point in the trajectory of the run, and relate the consequent state to the occurrence of the culminating event. In alphabetic terms:

$$\Sigma \rightsquigarrow \Sigma' = \Sigma - \{(_e,)_e, f_3\} \tag{45}$$

Another way for this coercion to occur is if the direct object is a mass noun. This would lead to a lack of end point in the trajectory, perhaps even a lack of trajectory at all.

4.5 Points $\rightsquigarrow$ Activities

As noted in M&S, the progressive requires a process as "input", yet interpretations can be given for:

Harry was sneezing (46)

M&S see this as being coerced by iteration, referring to a number of sneezes, rather than one sneeze.

Both Comrie [4] and Pulman [17] provide another interpretation for this coercion, where what was viewed as a point is stretched into a process, having internal structure.

VLH (p.175) deal with the first type, viewing it as a change from (-, -, e, -) to (f_1, -, -, -) or (f_1, f_2, -, -). While the EC needs to coerce an event into an activity fluent (a method for this is provided in [12]), FST does not face this ontological problem.

Events in FST can be represented by a multitude of fluents, providing different "perspectives" on events. Up to now $(_e)_e$ have been used to mark the start and end points of events, where we were not concerned with their internal or overall structure. Not representing the internal structure leads to the events being viewed as points. A fluent representing internal structure can be introduced, and as a fluent with a model-theoretic definition, there are no ontological obstacles to doing this.

To represent a point-like event being "stretched' the fluent prog_e, defined above, is added. This represents the event in progress as a homogeneous, stative fluent.

To represent iteration, a fluent iter_e is defined, which holds of any interval if two or more events happen within that interval. It can be thought of as a homogeneous process. While a particular subinterval may not contain a sneeze, it is considered part of the process of iteratively sneezing.

$$I \models_{FST} iter_e \iff (\exists I_i < I_j < I_k < I_l) \text{ s.t. } \begin{cases} I_i \models_{FST} (_e \\ \text{and } I_j \models_{FST})_e \\ \text{and } I_k \models_{FST} (_e \\ \text{and } I_l \models_{FST})_e \end{cases} \tag{47}$$

This coercion can be achieved by adding either prog_e or iter_e to the alphabet:

$$\Sigma \rightsquigarrow \Sigma' = \Sigma \cup \{prog_e\} \tag{48}$$

$$\Sigma \rightsquigarrow \Sigma' = \Sigma \cup \{iter_e\} \tag{49}$$

4.6 States ⇝ Activities

Croft [5] gives the following example of a state being coerced into a process:

She is resembling her mother more and more every day (50)

Though the acceptability of this type of coercion varies, VLH (p.173) give an account where a fluent representing the state "resembles her mother" is coerced into an activity fluent. They argue against coercing the state into the changing partial object, as might be expected given that increasing resemblance over time could be viewed as a trajectory.

Since a state is homogeneous, there are no problems treating it as an activity in FST. Activities and states are both represented by homogeneous fluents.

The activity fluent, which previously represented a state, must be related to a changing, partial object (represented by f_2) by some constraint set, leading to the crucial difference between "resembles" and "is resembling more and more".

5 Conclusion

The Event Calculus differs from other temporal representations by directly formalizing change, both instantaneous change as a result of events, and continuous change as a result of some constant force. Section 3 described how EC-predicates can be implemented as regular languages in FST. Furthermore, it was shown how a model, with a finite segmentation of the real number line as domain, for these FST-languages, could be formed from a model (with certain conditions excluding those models for which no finite segmentation exists), with the real number line as domain, for the equivalent EC-predicates. This shows that if a natural language semantics problem can be described in terms of change (or at least the kind of change that the EC formalizes), then the full real number line is not needed to model this problem.

Section 4 discusses type coercion, which the EC has been applied to. It is shown how various types of coercion, implemented in the EC as changes in scenario, or transformations of type (from fluents to events), can be implemented in FST. Type coercions in FST can be viewed as changes in "focus" or "perspective", formalized as changes in the alphabet from which strings are drawn.

Acknowledgement. This research is enhanced by support from Science Foundation Ireland through the CNGL Programme (Grant 12/CE/I2267) in the ADAPT Centre (www.adaptcentre.ie) at Trinity College Dublin. The ADAPT Centre for Digital Content Technology is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund.

References

1. Allen, J.F.: An interval-based representation of temporal knowledge. IJCAI **1**, 221–226 (1981)
2. Allen, J.F., Ferguson, G.: Actions and events in interval temporal logic. In: Stock, O. (ed.) Spatial and Temporal Reasoning, pp. 205–245. Springer, Dordrecht (1997)
3. Bennett, M., Partee, B.H.: Toward the logic of tense and aspect in English. Wiley Online Library (1978)
4. Comrie, B.: Aspect: An Introduction to the Study of Verbal Aspect and Related Problems. Cambridge University Press, Cambridge (1976)
5. Croft, W.: The structure of events. In: Tomasello, M. (ed.) The New Psychology of Language, Chap. 3. Lawrence Erlbaum Associates, Mahwah (1998)
6. Fernando, T.: Finite-state temporal projection. In: Ibarra, O.H., Yen, H.-C. (eds.) CIAA 2006. LNCS, vol. 4094, pp. 230–241. Springer, Heidelberg (2006)
7. Fernando, T.: Temporal propositions as regular languages. In: 6th International Workshop on Finite-State Methods and Natural Language Processing, pp. 132–48 (2008)
8. Fernando, T.: Partitions representing change homogeneously. In: Aloni, M., Franke, M., Roelofsen, F. (eds.) A festschrift for Jeroen Groenendijk, Martin Stokhof, and Frank Veltman, pp. 91–95. Onbekend (2013)
9. Fernando, T.: Segmenting temporal intervals for tense and aspect. In: The 13th Meeting on the Mathematics of Language, p. 30 (2013)
10. Fernando, T.: Incremental semantic scales by strings. In: EACL 2014, p. 63 (2014)
11. Halpern, J.Y., Shoham, Y.: A propositional modal logic of time intervals. J. ACM (JACM) **38**(4), 935–962 (1991)
12. Hamm, F., van Lambalgen, M.: Nominalization, the progressive and event calculus. Linguist. Philos. **26**, 381–458 (2003)
13. Kennedy, C., Levin, B.: Measure of change: the adjectival core of degree achievements. In: McNally, L., Kennedy, C. (eds.) Adjectives and Adverbs: Syntax, Semantics and Discourse, pp. 156–182. Oxford University Press, Oxford (2008)
14. Kowalski, R., Sergot, M.: A logic-based calculus of events. In: Schmidt, J.W., Thanos, C. (eds.) Foundations of Knowledge Base Management, pp. 23–55. Springer, Heidelberg (1989)
15. van Lambalgen, M., Hamm, F.: The Proper Treatment of Events. Wiley, New York (2005)
16. Moens, M., Steedman, M.: Temporal ontology and temporal reference. Comput. Linguist. **14**(2), 15–28 (1988)
17. Pulman, S.G.: Aspectual shift as type coercion. Trans. Philol. Soc. **95**(2), 279–317 (1997)
18. Solt, S.: Measurement scales in natural language. Lang. Linguist. Compass **9**(1), 14–32 (2015)
19. Vendler, Z.: Verbs and times. The Philos. Rev. **66**, 143–160 (1957)