LEABHARLANN CHOLÁISTE NA TRÍONÓIDE, BAILE ÁTHA CLIATH | TRINITY COLLEGE LIBRARY DUBLIN
Ollscoil Átha Cliath | The University of Dublin

Terms and Conditions of Use of Digitised Theses from Trinity College Library Dublin

Copyright statement

All material supplied by Trinity College Library is protected by copyright (under the Copyright and Related Rights Act, 2000 as amended) and other relevant Intellectual Property Rights. By accessing and using a Digitised Thesis from Trinity College Library you acknowledge that all Intellectual Property Rights in any Works supplied are the sole and exclusive property of the copyright and/or other IPR holder. Specific copyright holders may not be explicitly identified. Use of materials from other sources within a thesis should not be construed as a claim over them.

A non-exclusive, non-transferable licence is hereby granted to those using or reproducing, in whole or in part, the material for valid purposes, providing the copyright owners are acknowledged using the normal conventions. Where specific permission to use material is required, this is identified and such permission must be sought from the copyright holder or agency cited.

Liability statement

By using a Digitised Thesis, I accept that Trinity College Dublin bears no legal responsibility for the accuracy, legality or comprehensiveness of materials contained within the thesis, and that Trinity College Dublin accepts no liability for indirect, consequential, or incidental, damages or losses arising from use of the thesis for whatever reason. Information located in a thesis may be subject to specific use constraints, details of which may not be explicitly described. It is the responsibility of potential and actual users to be aware of such constraints and to abide by them. By making use of material from a digitised thesis, you accept these copyright and disclaimer provisions. Where it is brought to the attention of Trinity College Library that there may be a breach of copyright or other restraint, it is the policy to withdraw or take down access to a thesis while the issue is being resolved.

Access Agreement

By using a Digitised Thesis from Trinity College Library you are bound by the following Terms & Conditions. Please read them carefully.

I have read and I understand the following statement: All material supplied via a Digitised Thesis from Trinity College Library is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of a thesis is not permitted, except that material may be duplicated by you for your research use or for educational purposes in electronic or print form providing the copyright owners are acknowledged using the normal conventions. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone. This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

A Finite-State Approach to the Event Calculus and Type Coercion

Derek Kelleher

TRINITY COLLEGE DUBLIN
26 OCT 2016
ACADEMIC REGISTRY

PhD
COMPUTER SCIENCE
+ STATISTICS

A thesis submitted to the University of Dublin,

Trinity College, for the degree of

Doctor of Philosophy

2016

50691052
50691052

THESIS
11233

Declaration

I declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work.

I agree to deposit this thesis in the University's open access institutional repository or allow the library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.



Dei

DUBLIN

Thesis 11233

Abstract

The Event Calculus (EC) and Finite-State Temporality are two very different formalisms that both address Temporal Semantics. The link between these formalisms is Monadic Second-Order Logic (MSO).

Real-line models for the EC are shown, under certain circumstances, to be transformable into finite, discrete string models. These circumstances are those suitable for applying the EC to temporal semantics.

The strings are shown to satisfy Monadic Second-Order formulas translated from EC-formulas, indicating the entire real-line is not necessary for temporal semantics, EC-predicates can be replaced with MSO-formulas, and discrete models used.

Strings forming regular languages are shown to have their limits in the description of natural language semantics. Introducing context-free languages impacts on certain relations used in FST, leading to an increase in complexity.

The string models are interpreted under truthmaker semantics, separating them into presuppositions and exact verifiers. Presuppositions relating to temporal order, and negation are found to be important.

The EC's approach to type coercion is implemented in FST, the importance of forces in aspectual classification is investigated, and various coercions are shown to be linked to changes in vocabulary.

Acknowledgements

I would like to express my sincere thanks and gratitude to my supervisor Tim Fernando for his guidance and support on difficult topics, and in difficult times. I would also like to thank Carl Vogel who always provides an interesting and fresh perspective. Both have provided much more time to helping me than could be expected of anyone.

I am also very grateful to the people who have shared office space with me over the years: Martin, Alfredo, Berenger, and Shane for their interesting opinions and discussions on all things Computer Science.

I would like to thank the School of Computer Science and Statistics for funding my research.

Finally, I would like to thank my parents and family for providing support and additional funding.

Contents

1 Introduction **6**

1.1 Thesis Outline and Contributions 10

2 Background **13**

2.1 The Event Calculus 13

2.2 Monadic Second-Order Logic 20

2.3 Finite State Temporality 22

2.4 Events and Aspect 25

3 Between the Event Calculus and Finite State Temporality **40**

3.1 Event Calculus Models 40

3.2 Between Continuous and Discrete Models 43

3.3 Between the Event Calculus and Monadic Second-Order Logic . . 54

3.3.1 HoldsAt 56

3.3.2 Initially 61

3.3.3 Happens 63

3.3.4 Initiates 65

3.3.5 Terminates 70

3.3.6 Releases 73

3.3.7 Trajectory 73

3.4 Truthmaker Semantics 77
3.4.1 Initially 79
3.4.2 HoldsAt and Happens 80
3.4.3 Initiates, Terminates, and Releases 83
3.4.4 Trajectory 85
3.5 Finite-State Temporality and Context-Free Languages 86

4 Type Coercion **92**
4.1 Introduction 92
4.2 Type Coercion 104
4.3 Activities $\rightsquigarrow$ Accomplishments 106
4.4 Achievements $\rightsquigarrow$ Accomplishments 108
4.5 Accomplishments $\rightsquigarrow$ Activities 110

5 Conclusion **113**

Chapter 1

Introduction

In recent years, the Event Calculus (EC), originally developed by Kowalski and Sergot (1989) with AI applications in mind, has been profitably applied to natural language semantics.

The EC is a first-order logical language with states, instantaneous events, and continuous forces in its ontology. What distinguishes the EC from other formalisms is its inclusion of an independent timeline in its ontology. Events happen, and states hold at specific times, represented by real-numbers. Events are instantaneous and cause immediate change, giving the EC an almost physics-based character.

The extension of this formalism to natural language semantics (specifically temporal semantics) is given a comprehensive treatment in Van Lambalgen and Hamm's (VLH) book "The Proper Treatment of Events" (2005), with a very recent treatment in the Stanford Encyclopedia of Philosophy (Hamm 2014). The link between this physics-based language and complex event structures is made by associating verb meaning with "scenarios", sets of formulas that describe the event or action in question.

The temporal model underlying this is the positive real-line (including 0). This is most notably seen in VLH's treatment of continuous change, where a description is given of the state of the property undergoing change at each real-number time-point.

An alternative approach has been developed by Fernando (2013*a*), known as Finite-State Temporality (FST). Temporal phenomena are analyzed using finite-state methods, such as strings, regular languages, and operations on these objects.

The basic entity in FST is the fluent, essentially a temporal proposition. This can be any entity that can be given a temporal interpretation, such as a state, event, force, or time-point. This differs somewhat from the EC, where the term "fluent" describes states and continuous forces only, and not instantaneous events or time-points. Note that this is mostly a difference in the terminology used within each formalism and does not reflect a deeper difference.

These fluents are collected into sets, with each set forming an alphabet symbol. Strings and languages are formed from these symbols, with string position corresponding to temporal order, i.e. if the symbol representing set A precedes the symbol representing set B in a string, the fluents in set A temporally precede the fluents in set B. Complex events can be represented as strings/languages capturing the temporal order of temporal propositions. For example, the untensed event "reach the top" (of a mountain, say) can be represented as a simple state change from the proposition "¬at-the-top" to the proposition "at-the-top":

¬at-the-top	at-the-top

These strings have *segmentations* as models, a sequence of intervals that cover the entire timeline, with no two intervals overlapping. A segmentation is a model for a string if and only if it is the same length as the string, and all

the fluents in an alphabet symbol at a string position hold of the corresponding interval in the segmentation. Clearly, FST is a part of the Interval Semantics tradition (Bennett and Partee 1978).

Although a logical formalism may seem quite different to a finite-state formalism, they can be linked through Monadic Second-Order Logic (MSO). In MSO, monadic predicates represent alphabet symbols, and sets of string positions can be quantified over. A formula such as:

$$\exists x(P_a(x))$$

is interpreted as: "there is a string position with "a" as the symbol at that string position", or:

$$\forall x(P_a(x)) \tag{1.1}$$

meaning: "every string position has the symbol "a" at that position".

Strings serve as models of the formulas if the formula describes the string. For example the string:

aaa

having the symbol "a" at every string position is a model of the formula (1.1), while:

bbb

is not (not having the symbol "a" at every string position).

An important theorem due to Büchi-Elgot-Trackhtenbrot solidifies the connection, namely that the strings serving as models of MSO-formulas form regular

languages. Keeping the complexity level of the languages used in FST at regular languages enables a variety of finite-state operations to be used without getting into "undecidable" territory.

The strings serving as models of MSO-formulas can themselves be given an interpretation in the manner described above: by segmentations of the timeline.

With strings serving as models of MSO-formulas representing entire timelines, it is interesting to consider what it is about the string that satisfies the formula. This is the key topic of interest in Truthmaker Semantics, where exact verifiers, states that are wholly relevant to the sentences they verify, are investigated.

As mentioned above, the EC has recently been applied to temporal semantics. One area in which both the EC and FST have been applied is the phenomenon of Type Coercion. Categorizing events according to their internal temporal structure (or aspectual type) has long been of interest, and has settled on five categories:

- States
- Accomplishments
- Achievements
- Activities
- Semelfactives

Type coercion occurs when an event, usually viewed as one of the above types, behaves as a different type in certain contexts. For example, the event described by:

John sneezed (semelfactive)

behaves as an activity in the presence of the progressive:

John was sneezing (activity)

While the details of the above will be explained later, note that the former is a punctual occurrence with no consequent state, while the latter is an ongoing process with no specified endpoint.

Various approaches have been taken to aspectual classification and type coercion such as (Moens and Steedman 1988). The EC defines a structure called an "eventuality" that contains certain elements such as a "culminating event" and "consequent state". The particular elements that make up the eventuality determine its aspectual type. Coercion is therefore viewed as changes in these elements. FST uses the traditional notions of telicity (having a natural endpoint), and durativity (having duration or "internal structure") as being diagnostic of aspectual type. By giving a finite-state account of these two features, events represented by strings can be assigned to aspectual types. An FST account of type coercion is a core part of this thesis.

1.1 Thesis Outline and Contributions

The thesis is structured as follows:

- Chapter 1: Introduction.
- Chapter 2: This chapter gives the background on the broad topics touched on in the thesis. It is especially concerned with describing the three key formalisms encountered: the Event Calculus, Finite-State Temporality, and Monadic Second-Order Logic. It also describes various approaches to the topic of the aspectual classification of events. Other topics which could have been included, such as Truthmaker semantics and Type Coercion, are

discussed in the main part of the thesis, as the discussion of the background on these topics often ties in to key issues raised in those sections.

Section 2.1 gives an overview of the EC, as given in "The Proper Treatment of Events". It describes the formulas, axioms, models, etc. of the EC.

- Chapter 3: This chapter gives an account of the EC in finite-state terms. Section 3.1 expands on the treatment given in VLH of models for the EC (EC-models). VLH specify in detail a model for the EC axioms that deal with instantaneous change. However, a model for the axioms that deal with continuous change is not specified, and no outline is given of what this model may look like. When continuous change is accounted for, many aspects of the model given by VLH need to be adjusted. Section 3.2 shows that the EC scenarios (finite sets of formulas) given in VLH obey the Finite Approximability Hypothesis, that they use a finite set of fluents and each fluent defines an alternation bounded subset of the real-line. This means the EC-models can be partitioned into finite models, which can in turn be represented as strings. Section 3.3 translates EC-predicates into formulas of MSO. These formulas are arrived at intuitively, with their correctness investigated by first showing that they are satisfied by real-line models, and then showing that they are satisfied by the string models from the previous section. Being MSO-formulas, the strings satisfying them form regular languages. Taking the strings as models for the MSO translations of EC-formulas, Section 3.4 gives an account of this relation in terms of truthmaker semantics. A link is made between the language 2^{Σ} and the state space, with temporal and negation constraints forming a modalized state space. The strings that serve as models are separated into presuppositions and exact verifiers. Section 3.5 investigates the limits of expressing natural language semantics in this form. Certain constructions

will be represented as context-free languages, and it is shown that when a regular language is superposed with a context-free language, the result is not regular.

- Chapter 4: introduces the EC approach to aspectual classification, the eventuality, composed of fluents and events, with EC-formulas relating the components of the eventuality. Parallels are drawn with the FST approach and it is shown how the alphabet of the EC corresponds to the alphabet of FST. Given the translations of Chapter 3, the EC-formulas can be translated into FST languages. Viewing the constraints as giving the structure of the eventuality, type coercion is linked to changes in alphabet (or focus). Various type coercions are shown to be accounted for in this approach.

- Chapter 5: Conclusion.

Much of the material in this thesis has been peer-reviewed. Section 3.5 appears in (Kelleher and Vogel 2013). Sections 3.1, 3.2, and 3.3 have similar themes to (Kelleher et al. 2016), and Chapter 4 shares much material with (Kelleher et al. 2015).

Chapter 2

Background

2.1 The Event Calculus

The Event Calculus (hereafter referred to as the EC), originally developed by Kowalski and Sergot (1989), is a formalism that "axiomatizes the idea that all change must be due to a cause"(Van Lambalgen and Hamm 2005, p.36)(Note that the book "The Proper Treatment of Events" will be referred to as VLH from now on). It can therefore be seen as embodying the Commonsense Principle of Inertia, that "a property persists unless it is caused to change by an event".

The EC is a many-sorted first order logic with sorts for:

1. individual objects
2. real numbers
3. time-dependent properties
4. variable quantities
5. event types

The time-dependent properties, known as fluents, are taken to represent linguistic states and activities.

All predicates in the EC have an explicit temporal parameter, which quantifies over real numbers. Thus the real line can be viewed as an independent timeline against which events happen, and fluents hold.

In the EC, events and forces differ in two respects:

- Their temporal profile: Events happen at temporal instants, they have no duration. Forces hold for a temporal period, i.e. longer than an instant. A puntual "event" causes a fluent to hold, or cease holding, after the event.
- Type of change caused: Events cause instantaneous change, causing a fluent to hold, or cease holding, immediately after the event. Forces over an interval of time causes a fluent to hold at the end of that interval. To capture the continuous nature of change, this fluent is usually parameterized with a real-valued function of time.

For example, reaching the top of a mountain can be considered a punctual occurrence that causes a change in state from not being at the top of the mountain to being at the top of the mountain. Climbing a mountain could be conceived of as a force that takes place over a period of time that causes a continuous change in the climbers height on the mountain.

While it may seem that an event is just a very short force, or a force is a protracted event, they are represented by different entities. In addition, the $<$ sign (as opposed to a $\leq$ sign) in Axiom 4 (see below) prevents a force being used in a punctual manner. Therefore, it should be kept in mind that the terms "force" and "event" in the EC differ from their colloquial usage, and have separate meanings in the EC, and cause different kinds of change.

As a result of the above, a fluent, which previously did not hold, can come to hold in two ways: as a result of an event, which is usually the case when the

fluent is considered to go from not holding to holding instantaneously, or as a result of the application of a force over a period of time, for example a climbing force holding for a certain amount of time resulting in the climber reaching a certain height on the mountain, i.e. the fluent representing the climber being at that height coming to hold.

The predicates of the EC are as follows:

- Initially(f): The fluent f holds at the beginning of time.
- Happens(e,t): Event e happens at time t.
- Initiates(e,f,t): If event e happens at time t, then f will hold after t.
- Terminates(e,f,t): If event e happens at time t, then f will cease to hold after t.
- Trajectory(f_1,t,f_2,d): If the fluent f_1 holds between t and t+d, then the fluent f_2 will hold at t+d. Note the difference between fluents f_1 and f_2. Fluent f_1 represents a force, while fluent f_2 represents a state. As both states and forces are represented by the same kind of entity, they can only be distinguished by the argument positions they take within predicates. For example, the first argument of a Trajectory predicate will always be a force fluent.
- Releases(e,f,t): The Releases predicate prevents a clash between instantaneous and continuous change. As will be seen, the axioms of the EC ensure that if no event takes place, the value of a fluent (whether it holds or does not hold) remains the same. If it undergoes continuous change, this will be due to a force, not an event (as the EC conceives of it). This can lead to a contradiction where the fluent has undergone some change due to a force but the original value still holding because no event has

taken place. For example, suppose the height of a climber on a mountain is represented by a parameterized fluent h(x), and a climbing force causes a change in the fluent from h(0) to h(m), where m is the maximum height. Since no event has taken place, the axioms entail that both h(0) and h(m) to hold, an obvious contradiction. So before continuous change takes place, the initial fluent (here h(0)) is exempted from inertia by the Releases predicate.

- Clipped(t_1,f,t_2): A terminating or releasing event happens between t_1 and t_2.
- Declipped(t_1,f,t_2): An initiating or releasing event happens between t_1 and t_2.
- HoldsAt(f,t): As fluents are part of the domain and not predicates themselves, the HoldsAt predicate is needed to express that the fluent f is true at the time t.

The EC axioms are as follows:

Axiom 1 Initially(f)$\rightarrow$ HoldsAt(f,0).

Axiom 2 HoldsAt(f,r) $\wedge$ (r $<$ t) $\wedge$ ($\neg\exists$s $<$ r HoldsAt(f,s)) $\wedge$ $\neg$Clipped(r,f,t) $\rightarrow$ HoldsAt(f,t)

Axiom 3 Happens(e,t) $\wedge$ Initiates(e,f,t) $\wedge$ (t $<$ t$'$) $\wedge$ $\neg$Clipped(t,f,t$'$) $\rightarrow$ HoldsAt(f,t$'$)

Axiom 4 Happens(e,t) $\wedge$ Initiates(e,f_1,t) $\wedge$ (t $<$ t$'$) $\wedge$ (t$'$ = t + d) $\wedge$ Trajectory(f_1,t,f_2,d) $\wedge$ $\neg$Clipped(t,f_1,t$'$) $\rightarrow$ HoldsAt(f_2,t$'$)

Axiom 5 Happens(e,s) $\wedge$ (t $<$ s $<$ t$'$) $\wedge$ (Terminates(e,f,s) $\vee$ Releases(e,f,s)) $\rightarrow$ Clipped(t,f,t$'$)

Axiom 1 definies the Initially predicate in terms of the HoldsAt predicate, stating that if Initially(f) is true, then the fluent f holds at the time point 0.

Axiom 2 formalizes the commonsense principle of inertia for a fluent that has come to hold through a force (as opposed to an event), i.e. continuous change. As there is no time point s earlier than r at which the fluent holds, r is the earliest time point at which f holds. If f is not clipped (through a terminating or releasing event) between time points r and t, then f will continue to hold at timepoint t. This captures the idea that a property (f) that holds will continue to hold, unless an event takes place that forces it not to hold.

Axiom 3 formalizes the commonsense principle of inertia for a fluent that has come to hold due to an event (as opposed to a force), i.e. instantaneous change. An event at timepoint t in conjunction with the Initiates predicate has caused a fluent f to hold after t. The negation of the Clipped predicate states that a terminating or releasing event, which would cause the fluent t no longer hold, has not happened between t and t'. Therefore, the fluent f will hold at timepoint t'.

Axiom 4 gives the meaning of the Trajectory predicate. If an event initiates a force fluent f_1, and that force holds from timepoint t to timepoint t+d, i.e. is not clipped, then the fluent f_2 will hold at timepoint t+d.

Axiom 5 defines the Clipped predicate used in axioms 2, 3, and 4. A fluent is clipped between timepoints t and t' if an event e happens at a timepoint s between these times, and that event is a terminating or releasing event for the fluent f.

As pointed out in VLH, a simple model for the EC is given by taking the extensions of Initially and Happens to be empty (basically no fluent holds at the beginnng of time, and no events happen that could cause them to hold). This is a model for the EC as long as the extension of HoldsAt is empty, i.e. no

fluent holds at any time).

This model (involving no change) obviously does not give a full idea of the consequences of the EC, and VLH sketch a class of models which give a fuller treatment. It should be noted that the model is given in a slightly informal intuitive presentation,and does not deal with continuous change.

In this model, fluents are interpreted as sets of intervals over $\mathbb{R}_+$, open at the start point, and closed at the end point, unless they start at 0, in which case they are closed at both start and end point. While the domain of the EC is the set of points on the real line, the principle of inertia allows for them to be collected into intervals. If an initiating event occurs, and is followed some time later by a terminating event, the fluent will hold at all times after the initiating event, until the terminating event causes it not to hold, thus allowing the points it holds for to be collected into an interval that is open at the beginning (where the initiating event took place), but closed at the end (where the terminating event took place).

Instead of being given an independent extension, events are seen as derivative of fluents. Events are divided into initiating (e_f^+) and terminating (e_f^-) events for a particular fluent f, with:

$$e_f^+ := \{(f, r) | \exists s((r, s] \in f)\}$$
$$e_f^- := \{(f, s) | \exists r((r, s] \in f\}$$

The interpretations of the predicates are as follows (note that VLH only give a model for instantaneous change, with no interpretation for predicates such as

Trajectory and Releases):

$$HoldsAt := \{(f,t)|\exists I \in f(t \in I)\}$$

$$Initially := \{f|\exists s > 0[0.s] \in f\}$$

$$Happens := \{(e,t)|\exists f((e = e_f^+ \vee e = e_f^-) \wedge (f,t) \in e\}$$

$$Initiates := \{(e,f,t)|e = e_f^+ \wedge (f,t) \in e\}$$

$$Terminates := \{(e,f,t)|e = e_f^- \wedge (f,t) \in e\}$$

VLH (p.42) show how this model satisfies the EC axioms.

The EC axioms specify the general causal relationships that hold. Causal relationships which are specific to particular events and forces are given by *scenarios*. For example, some "situations" will have a natural end point (reaching the top of a mountain), and the scenario representing this situation will specify that a reaching-the-top event takes place when the climber has reached the maximum height. Open-ended situations (such as running), will not specify an end point that must be reached.

A state S(t) at time t is a first order formula built from:

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

A scenario is a conjunction of statements of the form:

1. Initially(f)
2. S(t) → Initiates(e,f,t)
3. S(t) → Terminates(e,f,t)

4. S(t) $\rightarrow$ Happens(e,t)

5. S(t) $\rightarrow$ Releases(e,f,t)

6. S(f_1,f_2,t,d) $\rightarrow$ Trajectory(f_1,t,f_2,d)

As noted earlier, the EC arose in an AI setting (Mueller 2014). VLH (p.44) bridge the gap between this physics-based AI formalism and linguistic applications by viewing scenarios as the lexical entries for various "eventualities". For example, the accomplishment "cross the street" is given by the following scenario:

1. Initially(one-side)

2. Initially(distance(0))

3. HoldsAt(distance(m),t) $\wedge$ HoldsAt(crossing,t) $\rightarrow$ Happens(reach,t)

4. Initiates(start,crossing,t)

5. Releases(start,distance(0),t)

6. Initiates(reach,other-side,t)

7. Terminates(reach,crossing,t)

8. HoldsAt(distance(x),t) $\rightarrow$ Trajectory(crossing,t,distance(x+d),d)

9. HoldsAt(distance(x_1),t) $\wedge$ HoldsAt(distance(x_2),t) $\rightarrow$ $x_1 = x_2$

2.2 Monadic Second-Order Logic

Monadic Second-Order Logic (MSO) is a second-order formalism with unary predicates. It has a particular importance in finite-state methods as its formulas precisely describe the regular languages. The syntax of MSO should be familiar

to anyone with a background in logic. A distinct feature is that predicate symbols are always of the form:

$$P_a$$

for various symbols "a". This feature exemplifies the formalism's relation to finite-state methods, where the symbol "a" is usually an alphabet symbol in the language, with P_a describing the set of string positions of which a holds.

Though MSO can be interpreted over infinite models (and this is one approach taken in Chapter 3), it is usually interpreted over finite models, particularly strings. Given a set of n string positions, let:

$$[n] = \{1, 2, \ldots, n\}$$

MSO has a "successor" function S(x), which describes the usual successor relation, for all $i \in [n-1]$:

$$S(i) = i + 1$$

An MSO-model for a finite set of symbols Σ is a tuple:

$$\langle [n], S, \{[\![P_a]\!]\}_{a \in \Sigma} \rangle$$

An MSO-model can be given as the string $\alpha_1 \ldots \alpha_n$ over the alphabet 2^{Σ} where:

$$\alpha_i := \{a \in \Sigma | i \in [\![P_a]\!]\}$$

The Büchi-Elgot-Trakhtenbrot Theorem (see (Thomas 1997) or (Libkin 2013))

makes the connection between MSO-formulas and string models. It states that a language is definable in MSO if and only if it is regular, i.e. the strings that serve as models for an MSO-formula form a regular language.

2.3 Finite State Temporality

An interesting approach to event classification posits states as the primitive entities, with the other classes arising out of the application of operators to states (Dowty 1979). Achievements such as "John spotted the man", are viewed as changes from the state of "John not seeing the man" to "John seeing the man". There are clear associations to finite-state automata which move from state to state depending on input symbols. Fernando (2004) makes these associations explicit by representing events as "runs of machines". Strings accepted by the automata are likened to motion pictures or comic strips, capturing the event.

In (Fernando 2004), the focus is shifted to strings and languages as opposed to the automata which accept them. Given a finite set of fluents Φ, form the power set 2^{Φ}, and let this be the alphabet Σ. For example, take $\Phi = \{a, b\}$. the alphabet $\Sigma = \{\{\}, \{a\}, \{b\}, \{a, b\}\}$. For increased legibility, the curly braces surrounding fluents are replaced with boxes, giving:

$$\Sigma = \{\boxed{\ }, \boxed{\text{a}}, \boxed{\text{b}}, \boxed{\text{a,b}}\}$$

Strings are formed from this alphabet in the usual way, with a string being a finite sequence of symbols from the alphabet, and a language being a set of strings.

Fernando shows how strings of fluents representing temporal propositions can be interpreted as temporal objects, with a fluents position within the string corresponding to its temporal order, i.e. if fluent A appears in a string position

to the left of the string position of fluent B, the fluent A temporally precedes fluent B. For example:

$$L = \boxed{\neg swim(john, x)} \quad \boxed{swim(john, x)}^{+} \quad \boxed{swim(john, m)}$$

$$L' = \qquad\qquad \boxed{mile(m)}^{*}$$

The above information can be combined through "superposition". The superposition of two langauges is the componentwise union of their strings:

$$L\&L' = \bigcup_{k \geq 1} \{(\alpha_1 \cup \alpha'_1) \ldots (\alpha_k \cup \alpha'_k) \mid \alpha_1 \ldots \alpha_k \in L \text{ and } \alpha'_1 \ldots \alpha'_k \in L'\}$$

Intuitively, snapshots taken at the same temporal instant are merged, forming a larger picture of the world at that time:

$$\boxed{\neg swim(john, x) mile(m)} \boxed{swim(john, x) mile(m)}^{+} \boxed{swim(john, m) mile(m)}$$

Note the presence of negated fluents in the above languages. In this respect, FST differs from EC, which would negate the HoldsAt predicate instead of the fluent, i.e. ¬HoldsAt(f,t) vs. HoldsAt(¬f,t).

The set of regular languages is closed under superposition ensuring that the superposition operation does not take us to a higher level of complexity Fernando (2004).

Superposition allows us to define a reflexive, transitive relation (a pre-order) associated with the concept of subsumption. To preserve reflexivity subsumption $\trianglerighteq$ is defined by:

$$L \trianglerighteq L' \quad \text{iff} \quad L \subseteq L\&L'$$

Subsumption can be thought of as relating to "information content". A language that subsumes another is more specific than that language. It contains all the information of the other language, and more. This is easier to see if we move from the level of languages to the level of strings. If s and s' are strings, where $s = \alpha_1 \ldots \alpha_n$ and $s' = \beta_1 \ldots \beta_k$, we say $s \trianglerighteq s'$ iff $n = k$ and for every i, $\alpha_i \supseteq \beta_i$. So every symbol of s contains all the fluents (information) of the corresponding symbol of s', and possibly more. Some properties of the subsumption operator are:

- $L_1 \trianglerighteq L_2$ iff $(\forall s \in L_1)(\exists s' \in L_2) s \trianglerighteq s'$
- $\trianglerighteq$ is reflexive and transitive
- $L_1 \trianglerighteq L_2 \& L_3$ iff $L_1 \trianglerighteq L_2$ and $L_1 \trianglerighteq L_3$

A useful tool in defining languages is the string constraint. Fernando (2006) defines a constraint:

$$L \implies L' \tag{2.1}$$

as the set of strings such that whenever $s \trianglerighteq \Box^n L \Box^m$, then $s \trianglerighteq \Box^n L' \Box^m$. Note that constraints define regular languages (Fernando 2006).

An example may make this concept clearer. The constraint $\boxed{\phi}\boxed{} \implies \boxed{}\boxed{\psi}$ says that if a string contains two consecutive boxes that can be subsumed by the language on the left hand side of the above implication, then in that string, those two boxes must be subsumed by the sequence on the right-hand side of the implication, i.e. any string that has ϕ in a box must have ψ in the next box. So the string $\boxed{\chi}\boxed{\phi}\boxed{\psi}$ meets this constraint, but the string $\boxed{\phi}\boxed{\chi}\boxed{\psi}$ does not.

2.4 Events and Aspect

The classification of verbs/events goes as far back as Aristotle (Bostock 1994) who, in the Metaphysics 1048, noted that some verbs have a natural end or result that others do not: "at the same time we see and have seen, understand and have understood, think and have thought; but we cannot at the same time learn and have learnt, or become healthy and be healthy". Twentieth Century philosphers such as Ryle and Kenny took up this challenge, and proposed classifications of their own. Ryle (2009) distinguished between achievements (resultative, happen at a particular moment) from activities (irresultative, happen over a period of time). Kenny (1963) looked at certain entailments and noted the imperfective paradox (more below). However, it is Vendler's classification which has endured and which most modern investigations use as a starting point.

Vendler (1957) distinguishes between four classes of verb (though his distinctions are now more widely applied to phrases and sentences). The four classes along with example are given in the table below:

Activities	**Accomplishments**	**Achievements**	**States**
Run	Run a mile	Arrive	Love
Drive	Draw a circle	Reach	Know
Carry	Paint a picture	Leave	Be tall
Push	Build a house	Sneeze	Recognize

The classification into four different categories is based on whether an event is positive or negative for two binary features. The first feature is whether it can occur in the progressive. The second is whether it is telic (having a natural end-point) or not. Activities occur in the progressive and are non-telic, Accomplishments occur in the progressive but are telic, Achievements do not occur in the progressive and are telic, and States do not occur in the progressive and are non-telic.

Non-telicity does not mean that activities/states do not end, just that there is no inherent end-point associated with them. While running or being in love could go on indefinitely (though obviously not in practice), running a mile or sneezing both have inherent end-points (the instant at which a mile has been run, or the sneeze finishes). We can check whether a verb is telic or non-telic by modifiying it with "in/for X time". Only activities and statives can be modified with "for", while only accomplishments and achievements can be modified with "in":

- Mike ran for ten minutes.
- Mike ran a mile in ten minutes.
- Mike knew her for ten years.
- Mike found it in ten minutes.

Typically, only activities and accomplishments appear in the progressive:

- Mike is running.
- Mike is running a mile.
- *Mike is knowing her.
- *Mike is finding it.

However, the progressive test is not foolproof, as some achievements can occur in the progressive. This is not generally seen as an achievement occurring in the progressive, but the achievement undergoing a type shift to a different class that can occur with the progressive (see below). The progressive suggests an ongoing process. Landman (1992) associates the occurrence of a predicate in the progressive with the concept of an event having "stages". A stage is a part of an event that is not identical to the event as a whole. Achievements

cannot occur in the progressive as they are too short to have stages, and states (which are long enough to have stages) are non-dynamic, so stages cannot be distinguished.

Lakoff (1965) posited the existence of semantic features such as +CAUSATIVE and +INCHOATIVE based on distinctions between sentences such as the following:

- The soup was cool
- The soup cooled
- John cooled the soup

with +INCHOATIVE being present in the deep structure for a sentence when some change or "becoming" occurs, and +CAUSATIVE being present when it is specified that something caused this change. Independently, McCawley (1968) proposed that words like "kill" were made up of the combination of components such as CAUSE, BECOME, NOT, and ALIVE.

Inspired by this, Dowty (1979) theorized that "the different aspectual properties of the various kinds of verbs can be explained by postulating a single homogeneous class of predicates, stative predicates, plus three or four sentential operators and connectives" (such as BECOME and CAUSE). Dowty considered stative predicates to be those the can be judged true or false of an individual by reference to the state of the world at a single moment of time.

Von Wright (1963) (also 1968) noted that certain events involve a change of state, specifically from the negation of a state to its unnegated state. For example, the event of opening a door can be viewed as a change of state from the door not being open to the state of the door being open. Von Wright used an operator T ("and next") to formalize this notion with:

- $\neg p T p$ The state p comes about

- pT¬p The state p comes to an end
- pTp The state p remains
- ¬pT¬p The state ¬p remains

Dowty defined the operator BECOME in terms of von Wright's logic:

$$\text{BECOME}(\phi) = \neg\text{pTp}$$

and also gave a teporal interpretation:

$$\text{BECOME}(\phi) \text{ is true at t iff } \phi \text{ is true at t and false at t}-1$$

Dowty suggests that all achievement verbs have a logical structure involving BECOME.

This analysis of BECOME explains why achievements cannot have an iterative interpretation:

- *Mark recognized the man for a week

Assuming a discrete time model, Dowty would interpret the above sentence as stating that at all times t, BECOME(ϕ) was true, with ϕ being the state of having recognized the man. Take any t_i for which this is the case. Since BECOME(ϕ) is true at t_i, $\neg\phi$ must be true at t_{i-1}. However, since BECOME(ϕ) is true at t_{i-1}, then ϕ must be true at t_{i-1}, leading to a contradiction with both ϕ and $\neg\phi$ being true at t_{i-1}.

The exception to this is when achievements contain indefinite plurals or mass nouns:

- Mark recognized people for an hour

Dowty viewed this as a difference in quantifier scope with an existential quantification lying within the scope of a universal temporal quantification noting that these sentences can be paraphrased as

- For an hour, Mark recognized some x such that x was a person

but not

- There was some x such that x was a person and, for an hour, Mark recognized x

The former of the above two paraphrases leads to different ϕ_i in BECOME(ϕ_i) which avoids the contradictions that iterated achievements can otherwise lead to.

Dowty associates accomplishments with the CAUSE operator, proposing that they have the structure (ϕ CAUSE ψ), where ψ is a BECOME sentence or an activity predicate, and ψ is a BECOME sentence. The sentence:

- John killed Bill

would have the form:

- ((John does something) CAUSE (BECOME¬(Bill is alive)))

while the sentence:

- John painted a picture

would have the form:

- ((John paints) CAUSE (BECOME(a picture exists)))

This is based on an observation from Geis (1970) that accomplishment sentences are elliptical:

- John dissolved the Alka Seltzer

- John dissolved the Alka Seltzer by doing something
- John's doing something dissolved the Alka Seltzer

More evidence for this analysis comes from factitive constructions such as:

- Jesse shot him dead
- she hammered the metal flat
- He swept the floor clean

where an activity verb and a resultant state are given. For example, the last of these sentences can be analyzed as:

- ((He sweeps the floor) CAUSE (BECOME(the floor is clean)))

Activities are identified with the DO operator combining with stative predicates. Ross (1972) proposed a DO element in the deep structure of activity sentences, noting that the morpheme "do" (or "did") can occur in these sentences:

- What I did then was call the grocer

Note that these morphemes do not occur in stative sentences:

- What I did then was be in Boston

It is then possible to give an account of the Vendler classes in terms of statives and the operators BECOME, CAUSE, and DO:

- Statives: $\pi(\alpha_1 \ldots \alpha_n)$ where π is a stative predicate (John knows the answer)
- Activities: DO(α_1, ($\pi(\alpha_1 \ldots \alpha_n)$)) where α_1 is the agent (John is walking)
- Achievements: BECOME($\pi(\alpha_1 \ldots \alpha_n)$) (John discovered the solution)

- Accomplishments: ((DO(α_1, ($\pi_1(\alpha_1 \ldots \alpha_n)$)) CAUSE (BECOME($\pi_2(\beta_1 \ldots \beta_n)$)) (John climbed the mountain)

with some additional categories definable:

- Stative causatives: ($\pi_1(\alpha_1 \ldots \alpha_n)$ CAUSE $\pi_1(\alpha_1 \ldots \alpha_n)$) (John's living nearby caused Mary to prefer this neighbourhood)
- Inchoation of accomplishments: BECOME(ϕ) where ϕ has the form of an accomplishment above (John began to build a house)
- Non-agentive accomplishments: ((BECOME(ϕ)) CAUSE (BECOME(ψ))) (The door's opening caused the lamp to fall down)

Krifka (1989) (see also (Krifka 1992), and (Krifka 1998)) gives a mereological account of the distinction between telic and atelic event predicates.

Krifka defines a part-structure P = $\langle U_P, \oplus_P, \leq_P, <_P, \otimes_P \rangle$ with:

- U_P is a set of entities
- $\oplus_P$ is the sum operation, a function from $U_P \times U_P$ to U_P that is idempotent, commutative, and associative
- $\leq_P$ is the part relation, defined as $\forall x, y \in U_P(x \leq_P y \leftrightarrow x \oplus_P y = y)$
- $<_P$ is the proper part relation, defined as $\forall x, y \in U_P(x <_P y \leftrightarrow x \leq_P y \wedge x \neq y)$
- $\otimes_P$ is the overlap relation, defined as $\forall x, y \in U_P(x \otimes_P y \leftrightarrow \exists z \in U_P(z \leq_P x \wedge z \leq_P y))$
- Remainder Principle: $\forall x, y \in U_P(x <_P y \rightarrow \exists\, !z(\neg(z \otimes_P x) \wedge x \oplus_P z = y))$

and then defines two properties of predicates, *cumulative* and *quantized*:

- $\forall X \subseteq U_P(CUM_P(X) \leftrightarrow \exists x, y(X(x) \wedge X(y) \wedge \neg x = y) \wedge \forall x, y(X(x) \wedge X(y) \rightarrow X(x \oplus_P y)))$
- $\forall X \subseteq U_P(QUA_P(X) \leftrightarrow \forall x, y(X(x) \wedge X(y) \rightarrow \neg y <_P x))$

An example of a cumulative predicate would be "apples", and an example of a quantized predicate would be "three apples".

Events also form a part-structure and can be related to the entities that "participate" in the event through thematic relations (usually the theme relation). For example, when dealing with an event of "eating an apple", Θ(y,e) is the theme relation where y is the apple and e is the eating event.

Krifka identifies four important properties of thematic relations:

- Mapping to subevents: $\forall x, y \in U_P\ \forall e \in U_E(\Theta(x, e) \wedge y \leq_P x \rightarrow \exists e'(e' \leq_E e \wedge \Theta(y, e')))$ i.e. whenever Θ holds for an object x and an event e, and y is part of x, then y stands in the relation Θ to a part e′ of e.
- Mapping to subobjects: $\forall x \in U_P\ \forall e, e' \in U_E(\Theta(x, e) \wedge e' \leq_E e \rightarrow \exists y(y \leq_P x \wedge \Theta(y, e')))$ i.e. whenever Θ holds for an object x and an event e, and e′ is part of e, then e′ stands in the relation Θ to a part y of x.
- Uniqueness of events: $\forall x, y \in U_P\ \forall e \in U_E(\Theta(x, e) \wedge y \leq_P x \rightarrow \exists\ !e'(e' \leq_E e \wedge \Theta(y, e')))$ i.e. if there is an event of eating an apple, for example, then for each part of that apple there is a unique part of the event of eating which corresponds to the eating of that part of the apple.
- Uniqueness of objects: $\forall x \in U_P\ \forall e, e' \in U_E(\Theta(x, e) \wedge e' \leq_E e \rightarrow \exists\ !y(y \leq_P x \wedge \Theta(y, e')))$ i.e. If there is an event of eating an apple, then for each subpart of that event there is a unique part of the apple which is being eaten over that subevent.

These properties together establish that there is a one-to-one mapping between object and event, i.e. each part of the event corresponds to a unique

part of the object and vice-versa. The cumulativity/quantization of the object predicate then transfers to the event predicate when the thematic relation has these properties.

Rothstein (2008) notes that it is a restricted form of cumulativity of event predicates that defines atelicity. The sum operation must form a singlular entity from the events it is summing, generally requiring the events be temporally adjacent, for example, summing a run from 3pm to 4pm and a run from 4pm to 5pm gives a run from 3pm to 5pm. If e, e′, and (e⊕e′) are all in the denotation of a predicate X and e is not part of e′, then either e or e′ end before (e⊕e′). Therefore there is an event that is part of (e⊕e′) which has an earlier ending point. This means that the end point of (e⊕e′) is not determined by the predicate X and X is not telic. However if a predicate X is quantized, with an event e in its denotation, no proper part of e can be in the denotation of X. Therefore, any part of e which is also in X is identical to e and X is telic.

States are cumulative. If Mike lived in Amsterdam from 1992 to 1994, and from 1994 to 1995, then (assuming these two time periods can be summed to form a singular entity i.e. no gaps) Mike lived in Amsterdam from 1992 to 1995. Their non-dynamicity comes from lack of change, which can be associated with their homogeneity down to instants. If Mike knew Jane for a year, then he knew her at every moment of that year.

There are a number of linguistic tests for whether a verb describes a state:

- States do not occur in the progressive:
  - *Mike is knowing the answer
- States in the present tense have a non-habitual reading:
  - Mike knows the answer
- States do not occur with "force" or "persuade":

- *Mike was forced to know the answer

- States do not occur as imperatives:
  - *Know the answer

- States do not occur with adverbs that indicate agency:
  - *Mike deliberately knew the answer

- States do not appear in pseudo-cleft constructions:
  - *What Mike did was know the answer

- Could + state entails would + state. For example:
  - Mike could know the answer if he read the book

  entails:
  - Mike would know the answer if he read the book

  whereas:
  - Mike could run if it was not raining

  does not entail:
  - Mike would run if it was not raining

- States are atelic (being cumulative), and therefore can occur with "for α time":
  - Mike lived in Amsterdam for ten years

- States lack stages and therefore cannot occur in the progressve (with some exceptions):
  - *Mike was knowing the answer

Activities are cumulative and therefore atelic. If Mike ran from 2pm to 3pm, and from 3pm to 4pm, then he ran from 2pm to 4pm. This property distinguishes activities from accomplishments: both of Mike's runnings are summed into one larger running, but if Mike read a book from 2pm to 3pm, and read a book from 3pm to 4pm, these two readings of a book cannot be summed into one larger reading of a book, as the book was read twice, or two different books were read.

As well as differing from states on the above linguistic tests, activities pattern differently than accomplishments as follows:

- The imperfective paradox: "Mike is $\phi-$ing" entails "Mike has $\phi-$ed". If Mike is currently running, then it can be concluded that he has ran, however if Mike is currently running a mile, then it cannot be concluded that he has ran a mile. Similarly, if an activity occurs with "stop", it has different entailments to an accomplishment. "Mike stopped running" entails that "Mike ran", while "Mike stopped running a mile" does not entail that "Mike ran a mile".

- As noted, activities are cumulative (and therefore atelic) and so can occur with "for α time":

  - Mike ran for an hour

- Activities and accomplishments have differing entailments when they occur with "almost":

  - Mike almost ran a mile
  - Mike almost ran

  The former can be interpreted in two way: that Mike did not run at all, or that he did run, but less than a mile. The latter can only be read in one way: That Mike did not run at all.

Accomplishments are non-cumulative and therefore telic. For example, if two temporally adjacent events of reading a book occur, they cannot be summed into one larger event of reading a book. This may be seen to be because accomplishments have an associated end point i.e. "finishing the book". Two temporally adjacent events that this predicate applies to, cannot be summed into one event that the predicate still applies to as their end-points do not coincide. However, accomplishments can be quantized. Take any subevent of the event of "read a book". At least one of the start point or end points of the reading of the book cannt be a part of any proper subevent. Therefore the only subevent of this to which the predicate "read a book" still applies is the event itself.

Achievements are non-cumulative, usually involving a change in state from a predicate not holding to that predicate holding, and one cannot die twice or recognize someone twice (at least in quick succession).

Achievements differ from the other classes in the following way:

- Achievements occur with "in α time":
  - Mike noticed the painting in a few minutes

  Note that while achievements and accomplishments can both occur with "in α time", they have slightly differing interpretations in this construction:
  - Mike read the book in an hour

  or
  - It took Mike an hour to read the book

  indicate that the period of reading was one hour, while:
  - Mike noticed the painting in a few minutes

or

- It took Mike a few minutes to notice the painting

indicate that there was some preparatory period (of being in the gallery for example) that lasted a few minutes, while the noticing event itself was instantaneous.

- When achievements and accomplishments occur with "in α time", they have differing entailmants. For example, if an accomplishment appears with "in α time":

  - Mike painted a picture in an hour

  then Mike was painting a picture during that hour. But if an achievement appears with "in α time":

  - Mike noticed the painting in a few minutes

  then it is not the case that Mike was noticing the painting during those few minutes.

- Achievements do not occur with "finish":

  - *Mike finished finding the treasure

- Certain adverbs such as "carefully", and "obediently" (those that impute intention) cannot occur with achievments:

  - *Mike carefully found the treasure

The notion of a "preparatory period" associated with an instantaneous event is realized more fully in Moens and Steedman (1988). The authors note that a theory of event structure must have more structure than just temporal relations such as "before" or "after". In the following example (Ritchie 1979), the when clause bears a different temporal relation to each of the main clauses:

- When they built the 39th St bridge, a local architect drew up the plans
- When they built the 39th St bridge, they used the best materials
- When they built the 39th St bridge, they solved most of their traffic problems

They also show that approximate temporal coincidence does not accurately capture the semantics of "when", with the following unnatural example:

- *When my car broke down, the sun set

nor does "when" denote a scientifically causal relation due to its intransitivity:

- When John left, Sue cried
- When Sue cried, her mother got upset
- When John left, Sue's mother got upset

The authors point out that the last sentence does not follow from the former two sentences.

Moens and Steedman claim that the relevant relation between these events is one of "contingency", which is meant to describe the expected relations between differing event types. For example, a "culmination" (achievement) is generally followed by a consequent state:

- Mike reached the top of the mountain

is a cumination, where immediately afterwards:

- Mike is at the top of the mountain

or:

- Mike has climbed the mountain

is true.

They propose that, underlying the Vendlerian categories, is a structure which they call the "event nucleus", with a culmination, and associated preparatory process and consequent state.

Focussing on different elements of the event nucleus leads to the different Vendler categories. Take an event nucleus which describes an event of climbing a mountain. The preparatory process is the climbing, the culmination is reaching the top, and the consequent state is being at the top of the mountain. The preparatory process is associated with activities, i.e.:

- Mike was climbing

describes an activity. The culmination is associated with achievements, i.e.:

- Mike reached the top

describes an achievement. The preparatory process together with a description of the culmination being reached forms an accomplishment, i.e.:

- Mike climbed the mountain

describes an accomplishment. Finally, the consequent state is clearly a state:

- Mike was at the top of the mountain

Chapter 3

Between the Event Calculus and Finite State Temporality

3.1 Event Calculus Models

In presenting a model for the EC (as seen in the above literature review), the goal of Van Lambalgen and Hamm (VLH) was "to sketch an intuitively appealing class of models". These models are intuitive in that initiating and terminating events do not alternate randomly (as in an example below), and appealing in that events actually happen. As VLH note, a simple, though uninteresting, model can be given by having the extension of Initially be empty, the extension of Happens be empty, and the extension of HoldsAt be empty, i.e. nothing holds at time 0, no events happen, and no fluent ever holds.

While it may be intuitive and appealing, the class of models presented in

VLH (p.41) deals only with instantaneous change. If the model is expanded to deal with continuous change, some issues arise. Recall that for VLH, the interpretation of the fluents is a set of intervals of the form (a,b] or [0,b] where a,b $\in \mathbb{R}_+$, with the extension of HoldsAt (for that particular fluent) being the set of points that make up these intervals. Instead of being given independently, the interpretation of events is given in terms of the interpretations of the fluents. Simply put, if a fluent does not hold for all points from t′ to t (inclusive), and holds for all points after t up until t″, then an initiating event e happened at time t and (f,t) is in the extension of e (similarly for terminating events).

However, once continuous change is allowed, a fluent may go from not holding to holding under the influence of a force, as well as due to an initiating event. For example if there are t, t′ $\in \mathbb{R}_+$ such that:

$$\forall d(t + d \leq t') \rightarrow (f_1, t, f_2, d) \notin [\![Trajectory]\!]$$

and:

$$\exists d \forall d'(t' < t + d' \leq t + d) \rightarrow (f_1, t, f_2, d') \in [\![Trajectory]\!]$$

then, by Axiom 4, if f_1 holds between t and t+d, f_2 will hold for every point after t′ up until t+d (inclusive), i.e. the extension of f_2 will contain an interval of the form (t′,t+d], and according to VLH's class of models, an initiating event for the fluent f_2 must have happened at t′. This is clearly not the case in the above model, as f_2 came to hold due to the force f_1, not an initiating event.

In addition, the extension of the fluents may no longer only be intervals of the form (a,b] or [0,b]. Suppose that (f_1,t,f_2,d) is in the extension of Trajectory for exactly one d, and there are no initiating or terminating events for f_2. Then if f_1 holds for all times greater than t and less than or equal to t+d, f_2 will only

hold for the point t′ = t+d, which is not an interval of the form (a,b].

It is clear from the above that giving the interpretation of the fluents as "basic" and deriving the interpretation of events and the EC-predicates from them is no longer sufficient. The extensions of the predicates must be given independently of each other, though to serve as models of the EC-axioms, some dependencies must be adhered to. For example, due to axiom 3, if for some event e, fluent f, and t, t′ ∈ $\mathbb{R}_+$:

$$(e,t) \in [\![Happens]\!]$$

and:

$$(e,f,t) \in [\![Initiates]\!]$$

and:

$$\forall t''(t < t'' < t') \rightarrow (e,f,t'') \notin [\![Terminates]\!] \wedge (e,f,t'') \notin [\![Releases]\!]$$

then:

$$(f,t') \in [\![HoldsAt]\!]$$

from which it can be concluded that an EC-model that satisfies Initiates(e,f,t) will satisfy the following condition:

$$t \in [\![Happens(e)]\!] \text{ implies } (\exists t')(\forall t'')(t < t'' \leq t' \rightarrow t'' \in [\![HoldsAt(f)]\!])$$

These conditions on models of EC-formulas will prove useful below.

3.2 Between Continuous and Discrete Models

In this section it will be shown that, under certain assumptions, a continuous, infinite model for the Event Calculus (EC), in particular a theory made up of the EC axioms and a scenario, can be transformed into a discrete, finite model for certain MSO formulas and FST languages (essentially, translations of the EC predicates). The basic idea is to partition the infinite, dense real line into a finite sequence of intervals that are "homogeneous", i.e. each fluent has a constant truth-value over this interval.

Before attempting this, a natural question to ask is whether a finite partition that meets the above homogeneity condition is guaranteed to exist for an arbitrary EC-model. A distinction must be made between models for the EC in general, and models for the version of the EC given in Van Lambalgen and Hamm (VLH), and useful for natural language semantics. Clearly, it is possible for a model which gives subsets of the real line as fluent extensions to have a fluent alternating between holding and not holding.

In the EC, the full real line is not used, as it is taken to start at 0, and is therefore left-bounded. It may seem that as it is not right-bounded (stretching to infinity) that a finite partition would be impossible. This can be true, even for non-dense linear orders. Take a model over the positive integers (and 0) for the following formulas:

$$HoldsAt(f, 0) \tag{3.1}$$

$$HoldsAt(f, t) \Longrightarrow Happens(e_T, t) \tag{3.2}$$

$$\neg HoldsAt(f, t) \Longrightarrow Happens(e_I, t) \tag{3.3}$$

$$Initiates(e_I, f, t) \tag{3.4}$$

$$Terminates(e_T, f, t) \tag{3.5}$$

The fluent f holds at 0, which by (3.2) and (3.5) will cause a terminating event to happen at 0. Therefore f will not hold at 1, which by (3.3) and (3.4) will cause an initiating event to happen. Therefore, f will hold at 2 and so on. The truth-value of f will constantly change, which while not a problem for a finite order, will not allow a finite partition over an order stretching infinitely to the right.

As noted above, the intervals in a partition are required to be homogeneous. A subset I of $\mathbb{R}_+$ is *U-homogeneous* if

$$(\exists t \in I)\ t \in U \quad \Longleftrightarrow \quad (\forall t \in I)\ t \in U$$

— i.e.,

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

Either every point in I is in U, or no point in I is in U. Of course, U may have more points in it than those contained in I.

A *U-segmentation* is a segmentation $I_1 \cdots I_n$ of $\mathbb{R}_+$ such that for all i between

1 and n, I_i is U-homogeneous and when $i < n$,

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

To see when a U-segmentation exists, the following notions are useful. Given a subset U of $\mathbb{R}_+$ and a positive integer $n > 0$, an (n, U)-*alternation* is a string $t_1 t_2 \cdots t_n$ of length n such that for $1 \leq i < n$, $t_i \in \mathbb{R}_+$, $t_i < t_{i+1}$ and

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

An alternation is essentially a sequence of points which alternate between being in U, and not being in 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

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

i.e. t and t′ are part of the same homogeneous interval.

Lemma For any subset U of $\mathbb{R}_+$,

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

For $\Rightarrow$, a U-segmentation of length n implies there can be an $(n+1, U)$-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. □

Given a U-segmentation $\mathbb{I} = I_1 \cdots I_n$, let us put the intervals I_i that are subsets of U into the set

$$\mathbb{I}^U \;:= \{I_i \mid 1 \leq i \leq n \text{ and } I_i \subseteq U\}$$

pronounced: $\mathbb{I}$ restricted to U

— which is either the set of odd intervals $I_1, I_3, \ldots$ or the set of even intervals $I_2, I_4, \ldots$ in $\mathbb{I}$.

Lemma U-segmentations are unique. Moreover, if $\mathbb{I} = I_1 \cdots I_n$ is a U-segmentation, then

(i) $(\mathbb{I}^U)_\bullet = U$ and

(ii) for every $X \subseteq \mathit{Ivl}(<)$ such that $X_\bullet = U$, and for every $I \in X$, there is a unique $I' \in \mathbb{I}^U$ such that $I \subseteq I'$ (from which it follows that the cardinality of X is no smaller than $\mathbb{I}^U$).

Finite approximability hypothesis EC Scenarios use only finitely many fluents, and each fluent defines an a.b. subset of $\mathbb{R}$

The above is now generalized to a finite set of fluents, each defining an alternation bounded subset of $\mathbb{R}_+$. Let these fluents be $\mathrm{f}_1, \ldots, \mathrm{f}_n$, and the defined subsets be $\mathrm{U}_1, \ldots, \mathrm{U}_n$.

A subset I of $\mathbb{R}$ is *X-homogeneous* iff

$$(\forall U \in X)\ \text{I is U-homogeneous}$$

An *X-segmentation* is a segmentation $I_1 \cdots I_n$ of $\mathbb{R}$ such that for all i between 1 and n, I_i is X-homogeneous and when $i < n$,

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

Let us say that a set U has a *right transition point* at t iff

$$(\exists t' > t)\ (\forall t'')\ ((t < t'' \leq t') \rightarrow (t \in U \rightarrow t'' \notin U) \vee (t \notin U \rightarrow t'' \in U))$$

and a *left transition point* at t iff

$$(\exists t' < t)\ (\forall t'')\ ((t' \leq t'' < t) \rightarrow (t \in U \rightarrow t'' \notin U) \vee (t \notin U \rightarrow t'' \in U))$$

Transition points are related to the topological notion of *boundary* given in (Fernando 2015*b*), defined as the closure of a set minus the interior of the set. While boundary is the more general notion, transition points are useful to describe the form that certain EC-predicates impose on models.

An alternation bounded fluent has a finite number of transition points, as it defines a segmentation (say of length n), and a segmentation of length n has n-1 transition points. As every point in $\mathbb{R}_+$ is in a segmentation, and there is no overlap between intervals, for any two contiguous intervals, I_i and I_{i+1}, either I_i is right-closed and I_{i+1} is left-open, or I_i is right-open and I_{i+1} is left-closed. Also, either I_i is part of U and I_{i+1} is not, or I_i is not a part of U and I_{i+1} is. According to the above definitions, there must be a transition point between the two intervals, and since there are a finite number of contiguous intervals,

there are a finite number of transition points.

Also, if a fluent has a finite number of transition points, it is alternation bounded. First note that if there is a right transition point at t and t is in U, then every point up to and including the next right transition point is also in U (not including the next transition point if it is a left transition point). Similarly, if there is a right transition point at t and t is not in U, then every point up to and including the next right transition point is also not in U (not including the next transition point if it is a left transition point). A similar argument can be made for left transition points. A finite number of transition points $t_1 \ldots t_n$ give n+1 U-homogeneous intervals $I_1 \ldots I_{n+1}$ where when $i \leq n$:

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

Therefore $t_1 \ldots t_n$ define a U-segmentation, and U is alternation bounded.

The number of transition points that the extension of an EC-fluent has must be given by those EC-formulas in which the fluent appears. Clearly, Initially(f) does not define a transition point. A model for Initially(f) has in the extension of f the set of points [0,s] for some $s \in \mathbb{R}_+$, and s can be any real number up to infinity. If s is less than infinity, a Terminating or Releasing event for f must have taken place. It is this predicate which defines the transition point, not Initially.

The presence of Trajectory(f_1,t,f_2,d) in an EC-theory has no effect on the extension of f_1, as this predicate is essentially a conditional, spelling out the consequences of f_1 holding, but not committing to whether or not f_1 actually holds. The fluent f_2 is usually a fluent-valued function of d to represent continuous change. For example, to represent "climbing the mountain", f_2 could be of the form h(x+g(d)), where g is a real-valued function of d, giving the distance climbed in time d, h(x) being the height at time t, and h(x+g(d)) the height

at time t+d. Given that for each $x \in \mathbb{R}_+$, h(x) is a distinct fluent, there is a potentially infinite set of fluents describing continuous change. Whether continuous change needs to e described by infinite means is open to debate, but as stated above, the set of fluents must be finite to achieve a finite segmentation. If increased granularity is required, more fluents can be added to the vocabulary, as long as the set remains finite.

Of course, even with a finite number of fluents as the range of $f_2(x)$, a periodic function could cause an individual fluent to vary between holding and not holding infinitely. Therefore, to have a finite number of transition points, periodic functions (such as sin) must be ruled out. Note that VLH (p.51) make similar restrictions. One option would be to recast periodic change as an increasing process. For instance, a circular walk would mean that at various times the walker would be at the start position, while at all other times he is not. Instead of treating the position as the changing value, the percentage (or some other measure) of the walk completed could be used, an increasing value with each fluent only holding once.

Given axiom 3:

$$Happens(e,t) \wedge Initiates(e,f,t) \wedge (t < t') \wedge \neg Clipped(t,f,t') \rightarrow HoldsAt(f,t')$$

when Initiates(e,f,t) is present in conjunction with Happens(e,t), it is interpreted as follows: a punctual event e happens at time t, and at every point after (but not including) t, f holds up until a time t' where a terminating or releasing event happens.

In a model for this axiom and EC-formula, t clearly meets the criteria for being a transition point, i.e. there will exist some t' (the point at which the next terminating event occurs for example) for which all points between t and t' are in the extension of f. Terminates(e,f,t) and Releases(e,f,t) are interpreted

in a similar manner. If an event e happens at time t, then f (which holds at t) will cease to hold for all points after t, until the next initiating event occurs.

In a model for these formulas, there will exist a t' (the time of the next initiating event for example), for which all points between t and t' are not in the extension of f. Therefore, it is occurrences of the predicate Happens(e,t) where e is an argument of a predicate Initiates(e,f,t), Terminates(e,f,t), or Releases(e,f,t) that defines transition points.

Given the fixed axioms, whether the interpretation of a fluent in an EC-model has a finite or infinite number of transition points will depend on the formulas in the scenario. Recall from section 2.1 that a scenario contains formulas of the following form (ignoring Initially and Trajectory given the comments above):

1. S(t) $\rightarrow$ Initiates(e,f,t)
2. S(t) $\rightarrow$ Terminates(e,f,t)
3. S(t) $\rightarrow$ Happens(e,t)
4. S(t) $\rightarrow$ Releases(e,f,t)

with S(t) being:

1. literals of the form ($\neg$)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).

Note that Initiates, Terminates, and Releases usually appear as formulas in themselves (without S(t)$\rightarrow$), specifying causal relationships between events and fluents. For example, take this excerpt from the scenario given in section 2.1:

1. Initiates(start,crossing,t)

2. Releases(start,distance(0),t)

basically stating that for all times t (the t being implicitly universally quantified), if the event start happens at t, then the fluent crossing holds after t, and the fluent distance(0) is released from the inertial axioms.

Formulas of this kind do not affect the **number** of transition points, though they will specify whether a particular event defines a transition from the fluent holding to not holding (in the case of Terminates or Releases), or from the fluent not holding to holding (in the case of Initiates).

What will determine the number of transition points are the formulas containing Happens. Atomic formulas such as Happens(e,t_0), where t_0 is a particular time can only define a single transition point, so if the Happens predicate appears only in this form in a finite scenario, there will be only a finite number of transition points.

But it may be noted in the scenario given in section 2.1 that Happens often appears as the consequence of a HoldsAt antecedent:

$$HoldsAt(distance(m), t) \wedge HoldsAt(crossing, t) \rightarrow Happens(reach, t)$$

This can lead to fluents with infinite transition points as follows. Consider the following scenario:

1. Initiates(e_I,f,t)
2. Terminates(e_T,f,t)
3. $\neg$HoldsAt(f,t)$\rightarrow$ Happens(e_I, t+1)
4. HoldsAt(f,t)$\rightarrow$ Happens(e_T, t+1)

Since f does not hold initially, an initiating event will occur and f will come to hold. Since it no longer holds a terminating event will occur and it will cease

to hold, then another initiating event will occur causing it to hold again and so on. In a model such as the real line which is unbounded to the right this will lead to infinite transition points and a fluent which is not alternation bounded.

As a result, scenarios of the above form (where a fluent holding causes itself to not hold, and vice-versa) must be ruled out. Of course, it is not hard to conceive of situations where the above scenario holds, such as a pendulum swinging, or breathing in and out. When a pendulum reaches its leftmost point, gravity causes it to swing back to its rightmost point, and vice-versa. This indicates that there may be some linguistic phenomena that the restriction too finite partitions may not be able to handle.

It is possible that cases such as the above can be handled by a judicious choice of the fluents used to describe it. For example, it seems difficult to describe the above scenario without either bundling the left and right swing into one event, i.e. "The pendulum swung left and right forever", or treating the left swing and the right swing as the same thing, a swing, i.e. "The pendulum swung infinitely". Using one fluent to describe not just a state, but an alternation between a state holding and not holding may address this issue. This could be explored in future work.

While it cannot be assumed that finite partitions can deal with all temporal linguistic constructions, the linguistic phenomena addressed in this thesis fall under the aspectual types discussed in the literature review. They involve either finite length such as climbing a mountain, or open-ended processes such as running (where no endpoint is specified).

In addition, in VLH, formulas with a HoldsAt antecedent and Happens consequent are solely used to relate the culminating event that occurs when a maximum/minimum is reached (see Chapter 4).

The above arguments give the conditions under which an EC scenario defines

an alternation bounded fluent. It is clear that a finite scenario uses only a finite number of fluents, and therefore the finite approximability hypothesis holds.

Fernando (2013*b*) shows that a finite set of alternation bounded fluents defines a segmentation. Note that only the truth-value of one fluent need change between adjacent intervals. For instance, take fluents f and g, with f holding of the intervals [0,4] and (7,∞), and not holding for (4,7], and g holding for the interval [0,2], and not holding for (2,∞). A segmentation of the real-line which is homogeneous for each fluent would be I_1 = [0,2], I_2 = (2,4], I_3 = (4,7], and I_4 = (7,∞), with g not holding for the adjacent intervals I_2, I_3, and I_4. Therefore, the EC-models used in VLH, can be partitioned into finite models.

A segmented model $I_1 \ldots I_n$ can be transformed into a string model $\Delta(I_1) \ldots \Delta(I_n)$ with:

$$f \in \Delta(I_i) \text{ iff } I_i \models f$$

The purpose of this section was to show how certain models for the EC could be represented by finite partitions of the real-line, and consequently, also as strings. It may also be noted that EC models which cannot be given finite partitions, may be finitely approximated by sets of strings, such as the language:

$$\boxed{\mathrm{f}}\boxed{\neg\,\mathrm{f}}^{+}$$

The potential for approximating models using languages such as the above can be investigated in future work.

3.3 Between the Event Calculus and Monadic Second-Order Logic

The Event Calculus can be related to Finite State Temporality through Monadic Second-Order Logic. Below, it will be seen that there are natural translations from the EC into MSO by treating the non-temporal parameters of the EC-predicate as a part of the MSO-predicate. This will be seen to have appealing consequences when the MSO-formulas are interpreted over string models. This direct style of translation will also be seen to have less appealing consequences in the case of certain EC-predicates. In these cases a more intuitive translation will be preferred.

As will be seen in section 3.5, it is important in FST that the languages involved are regular languages. Therefore, for the string models that satisfy MSO translation of EC formulas, it is necessary to demonstrate that the strings can be collected into languages that are regular. Given the relationship between the models of MSO formulas and regular languages discussed in the literature review, i.e. the strings satisfying an MSO formula together form a regular language, it is enough to show that the string models formed from the partitions of EC models satisfy the MSO formulas which are translations of the EC formulas.

More formally, given an EC formula ϕ, and a language L(ϕ), where L(ϕ) is the set of strings formed from partitioned ECmodels of ϕ, the following hypothesis will be proved:

Finite state event calculus hypothesis For all EC formulas ϕ, L(ϕ) is a regular language.

To investigate the "correctness" of the translations, two approaches are taken:

1. The MSO translations are shown to be true relative to the same domain

($\mathbb{R}_+$) as the EC-formulas. Consequently, interpretations must be given for the MSO-predicates in a way that is coherent with the interpretations of EC-predicates. The purpose of this approach is an initial evaluation of the translations over the same domain as the EC, before examining their correctness relative to finite domains.

2. The MSO translations are shown to be true relative to string models, specifically segmentations of an EC-model. These string models, forming regular languages, can then be interpreted in FST, allowing applications of the EC to natural language semantics to be adopted in FST.

As discussed in the previous section, the existence of a string model relies on finitely many fluents being used. This assumption is acceptable in that the scenarios in VLH are finite and contain finitely many fluents. However, some fluents are parameterized with real-number arguments, which could be considered to give rise to an infinite number of fluents. These parameterized fluents are used to represent the change undergone due to a force, such as the height of a climber on a mountain. It is open to debate whether such precision is a necessary part of the meaning of phrases such as "climb a mountain". Clearly, a listener can understand that a climber climbed a mountain without being aware of her exact height at each instant of time. For the purposes of this chapter, it should be kept in mind that a parameterized fluent from the EC will be approximated by finitely many fluents in FST. If more precision is required, more fluents can be added,though will always be limited to a finite number.

3.3.1 HoldsAt

In an EC-model HoldsAt is interpreted as a set of fluent/time pairs. Recall from the specification of an EC-model in 2.1:

$$[\![HoldsAt]\!] = \{(f,t)|\exists I \in f(t \in I)\}$$

Given the interpretation of HoldsAt for a finite set of fluents, for each fluent the following interpretation can be defined:

$$[\![HoldsAt(f)]\!] = \{t|(f,t) \in [\![HoldsAt]\!]\}$$

Keeping the temporal argument, but making the fluent argument part of the predicate, the HoldsAt predicate is translated into a set of MSO-predicates, rather than a single predicate. There is a separate MSO-predicate for each fluent, e.g. $\mathrm{P}_f(\mathrm{x})$. Since there are finitely many fluents f, there are a finite number of MSO-predicates P_f. As noted above, under the first approach where MSO translations are interpreted over the same domain as the EC-formulas, a suitable interpretation must be given for the MSO-predicates. The obvious candidate for the interpretation of P_f is that it has the same interpretations as HoldsAt(f):

$$[\![P_f]\!] := [\![HoldsAt(f)]\!]$$

The translation of the formula:

$$HoldsAt(f,t_i)$$

where t_i is a temporal constant is:

$$\exists x(P_f(x) \wedge P_{t_i}(x))$$

the idea being that there is a timepoint named t_i and that the predicate P_f holds of this point.

The predicate P_{t_i} in the above formula is a nominal temporal predicate (Blackburn et al. 1993), which takes the place of the temporal constants used in the EC. A nominal predicate has unary extension, i.e. a temporal nominal predicate holds of one time point only. For example, the MSO-predicate P_0 only holds of timepoint 0, and essentially names that point. Nominal temporal predicates can be incorporated into MSO-formulas that represent EC-predicates which contain temporal constants. For example, HoldsAt(f,0) becomes $\exists x(P_f(x) \wedge P_0(x))$. Here, the usual interpretation of the existential as "one or more" is constrained by the nominal predicate to "only one". Note that not every timepoint needs to be represented by a nominal temporal predicate, only those timepoints that are "of interest", timepoints where events happen, or fluents are claimed to hold for example. An EC scenario that can be finitely partitioned will have a finite number of timepoints of interest, meaning a finite number of nominal temporal predicates are necessary.

Real Model

A model for the EC-formula HoldsAt(f,t_i) where t_i is a temporal constant has:

$$EC \models HoldsAt(f, t_i)$$

iff

$$(f, t_i) \in [\![HoldsAt]\!]$$

$$t_i \in [\![HoldsAt(f)]\!]$$

iff

$$t_i \in [\![P_f]\!]$$

iff

$$MSO \models \exists x(P_f(x) \wedge P_{t_i}(x))$$

String Model

Over a string model the formula:

$$\exists x(P_f(x) \wedge P_{t_i}(x))$$

is satisfied if for some string position i:

$$f \in \Delta(I_i) \text{ and } t_i \in \Delta(I_i)$$

iff:

$$I_i \models f \text{ and } I_i = \{t_i\}$$

iff:

$$t_i \in [\![HoldsAt(f)]\!]$$

iff:

$$EC \models HoldsAt(f, t_i)$$

Negation of HoldsAt

HoldsAt sometimes appears in negated form (¬HoldsAt(f,t)). This is to be interpreted as the fluent f does not hold at time t. This can be translated into an MSO-predicate in two ways: $\neg P_f(x)$ or $P_{\neg f}(x)$. Clearly negated propositions make sense where negated events would not.

The latter option is chosen because, as mentioned in the literature review, FST makes use of negated fluents. In addition, a string can be viewed as a visual representation of a timeline. The presence of a negated fluent seems clear as a statement of that fluent not holding at a string position rather than the absence of that fluent at a string position (from a visual perspective). The absence of a fluent would require a viewer to keep the underlying alphabet in mind and conclude that any fluents absent at a string position did not hold at that position.

$P_{\neg f}$ is given the following interpretation:

$$t \in [\![P_{\neg f}]\!] \iff t \notin [\![P_f]\!]$$

Real Model

A model for the EC-formula ¬HoldsAt(f,t_i) where t_i is a temporal constant has:

$$EC \models \neg HoldsAt(f, t_i)$$

iff

$$(f, t_i) \notin [\![HoldsAt]\!]$$

iff

$$t_i \notin [\![HoldsAt(f)]\!]$$

iff

$$t_i \notin [\![P_f]\!]$$

iff

$$t_i \in [\![P_{\neg f}]\!]$$

iff

$$MSO \models \exists x(P_{\neg f}(x) \wedge P_{t_i}(x))$$

String Model

Over a string model the formula:

$$\exists x(P_{\neg f}(x) \wedge P_{t_i}(x))$$

is satisfied if for some string position i:

$$\neg f \in \Delta(I_i) \text{ and } t_i \in \Delta(I_i)$$

iff:

$$I_i \nvDash f \text{ and } I_i = \{t_i\}$$

iff:

$$t_i \notin [\![HoldsAt(f)]\!]$$

iff:

$$EC \models \neg HoldsAt(f, t_i)$$

3.3.2 Initially

While Initially has no temporal parameter, it is given a temporal interpretation through Axiom 1:

$$Initially(f) \rightarrow HoldsAt(f, 0)$$

No other axiom makes reference to Initially, therefore it can be translated into a monadic predicate that asserts that the fluent f holds at timepoint 0:

$$\exists x(P_f(x) \wedge P_0(x))$$

Real Model

A model for the EC-formula Initially(f) has:

$$EC \models Initially(f)$$

iff

$$(f, 0) \in [\![HoldsAt]\!]$$

iff

$$0 \in [\![HoldsAt(f)]\!]$$

iff

$$0 \in [\![P_f]\!]$$

iff

$$MSO \models \exists x(P_f(x) \wedge P_0(x))$$

String Model

Over a string model:

$$s \models \exists x(P_f(x) \wedge P_0(x))$$

is satisfied if for some string position i:

$$f \in \Delta(I_i) \text{ and } 0 \in \Delta(I_i)$$

iff:

$$I_i \models f \text{ and } I_i = \{0\}$$

iff:

$$0 \in [\![HoldsAt(f)]\!]$$

iff:

$$EC \models HoldsAt(f, 0)$$

3.3.3 Happens

In an EC-model, Happens(e,t) is interpreted as a set of event-type/time pairs. Given this interpretation of Happens, the following interpretation can be defined for a finite set of event-type fluents:

$$[\![Happens(e)]\!] := \{t | (e, t) \in [\![Happens]\!]\}$$

Similarly to HoldsAt, Happens is translated into a set of MSO-predicates, a separate predicate for each event-type e:

$$P_e$$

There is a key difference between EC-formulas containing Happens and EC-formulas containing HoldsAt. As noted earlier, negated event-types (¬e) do not make sense. Also, negated Happens predicates do not appear in the EC so negation does not have to be dealt with here.

Given a suitable interpretation of the EC-predicate Happens(e,t) as a set of

event-type/time pairs, the interpretation of P_e is given by:

$$[\![P_e]\!] := [\![Happens(e)]\!]$$

The translation of Happens(e,t_i) (where t_i is a constant) into MSO is:

$$\exists x(P_e(x) \wedge P_{t_i}(x)) \tag{3.6}$$

the idea being that there is a timepoint named t_i and that the predicate P_e holds of this point.

Real Model

A model for the EC-formula Happens(e,t_i) where t_i is a temporal constant has:

$$EC \models Happens(e, t_i)$$

iff

$$(e, t_i) \in [\![Happens]\!]$$

iff

$$t_i \in [\![Happens(e)]\!]$$

iff

$$t_i \in [\![P_e]\!]$$

iff

$$MSO \models \exists x(P_e(x) \wedge P_{t_i}(x))$$

String Model

Over a string model:

$$s \models \exists x(P_e(x) \wedge P_{t_i}(x))$$

iff for some string position i:

$$e \in \Delta(I_i) \text{ and } t_i \in \Delta(I_i)$$

iff:

$$I_i \models e \text{ and } I_i = \{t_i\}$$

iff:

$$t_i \in [\![Happens(e)]\!]$$

iff:

$$EC \models Happens(e, t_i)$$

3.3.4 Initiates

If Initiates(e,f,t) was translated into a set of monadic predicates in a similar manner to the above predicates, the arguments e and f would be made part of

the predicate, giving a translation of Initiates(e,f,t) into the monadic predicates:

$$P_{Initiates(e,f)}$$

with differing e and f giving different monadic predicates.

There is a clear disadvantage to this approach for this predicate. If the above translation was interpreted over a string model, Initiates(e,f) would be an alphabet symbol in the string. However, the basic elements of FST are states and events. Clearly the predicates P_f and P_e, giving alphabet symbols f and e, matches up nicely with the basic elements of FST. It is difficult to see what meaning could be assigned to an alphabet symbol Initiates(e,f).

An alternative option is to not attempt a "direct" translation in the style above, but to capture the intended interpretation of Initiates(e,f,t), as a relationship between e and f, constrained by Axiom 3. Recall that Axiom 3 states that if an event happens at a timepoint, and that event is an initiating event for a fluent, and that fluent is not terminated or released between that timepoint and a later timepoint, then the fluent still holds at that later timepoint. Therefore, if an initiating event occurs then the fluent holds at all subsequent times until a terminating or releasing event takes place. If no such event occurs then the fluent will hold forever. Note that in VLH no treatment of terminating/releasing events occurring at the same time as initiating events is given, so if an initiating event occurs, the fluent is guaranteed to hold for some set of immediately subsequent points. Therefore, an EC-model for Initiates(e,f,t) must comply with the following constraint:

$$t \in [\![Happens(e)]\!] \text{ implies } (\exists t' > t)(\forall t'')(t < t'' < t \rightarrow t'' \in [\![HoldsAt(f)]\!]) \tag{3.7}$$

The translation of Initiates(e,f,t) into an MSO-predicate is therefore:

$$\forall x(P_e(x) \rightarrow P_f(S(x)))$$

where S(x) is the successor function (see section 2.2).

However, this creates an issue over dense linear orders such as the real-line, where no point has a successor (for a point x and any later point x′, there are infinite points between x and x′, so x′ cannot be the successor of x).

The above formula can still be given an interpretation. Under this interpretation, S(x) is not a successor function of x but an abbreviation of the following MSO-formula:

$$P_f(S(x)) \equiv (\exists x' > x)(\forall x'')(x < x'' < x \rightarrow P_f(x''))$$

Essentially, P_f holds of every point after x up to some later point. While no successor of x is identified, the intended interpretation of Initiates(e,f,t) is captured,i.e. that after the point x (at which the initiating event occurs), P_f holds for some set of immediately subsequent points. Of course, over a discrete linear order, such as a string model, there is no issue with identifying a successor of any point.

Real Model

A model over the real-line for this MSO-formula will comply with the following condition, for every t:

$$t \in [\![P_e]\!] \text{ implies } (\exists t' > t)(\forall t'')(t < t'' < t \rightarrow t'' \in [\![P_f]\!])$$

$$EC \models Initiates(e, f, t)$$

iff (by 3.7):

$t \in [\![Happens(e)]\!]$ implies $(\exists t' > t)(\forall t'')(t < t'' < t \rightarrow t'' \in [\![HoldsAt(f)]\!])$

Suppose:

$$t \in [\![P_e]\!]$$

iff:

$$t \in [\![Happens(e)]\!]$$

implies:

$$(\exists t' > t)(\forall t'')(t < t'' < t \rightarrow t'' \in [\![HoldsAt(f)]\!])$$

iff:

$$(\exists t' > t)(\forall t'')(t < t'' < t \rightarrow t'' \in [\![P_f]\!])$$

Since:

$$t \in [\![P_e]\!]$$

implies:

$$(\exists t' > t)(\forall t'')(t < t'' < t \rightarrow t'' \in [\![P_f]\!])$$

then:

$$MSO \models \forall x(P_e(x) \rightarrow P_f(S(x)))$$

String Model

Recall that in an EC-model for Initiates(e,f,t), an initiating event for a fluent f occurring at time t defines a right-transition point, where:

$$(\exists t' > t)\ (\forall t'')\ ((t < t'' < t') \rightarrow t'' \in [\![HoldsAt(f)]\!])$$

Therefore, for an alternation-bounded fluent f, collecting the points t'' where t<t'' $\leq$t' into an interval I_i, then there is a segmentation of the EC-model where $I_i \subseteq$ [[HoldsAt(f)]] and $I_{i+1} \cap$ [[HoldsAt(f)]] = $\emptyset$.

Therefore, a segmentation of an EC-model for Initiates(e,f,t) must satisfy the following conditions:

$$e \in \Delta(I_{i-1}) \text{ implies } f \in \Delta(I_i)$$

iff:

$$s \models \forall x(P_e(x) \rightarrow P_f(S(x)))$$

3.3.5 Terminates

In the EC, Terminates has a similar interpretation to Initiates. If an event happens and that event is a terminating event for a fluent f, then that fluent will cease to hold after the event, up until the next initiating event happens (or some continuous force causes it to hold). In other terms, the negated fluent will hold until the next initiating event takes place (As discussed in section 3.3.1, a fluent not holding will be treated as the negation of the fluent holding ($\neg \mathrm{P}_f$ vs. $\mathrm{P}_{\neg f}$)).

This is translated into MSO as:

$$\forall x(P_e(x) \rightarrow P_{\neg f}(S(x)))$$

An EC-model that satisfies Terminates(e,f,t) will satisfy the following condition:

$$t \in [\![Happens(e)]\!] \text{ implies } (\exists t')(\forall t'')(t < t'' \leq t' \rightarrow t'' \notin [\![HoldsAt(f)]\!]) \quad (3.8)$$

A model over the real-line for the MSO translation of Terminates(e,f,t) will satisfy the following condition, for every t:

$$t \in [\![P_e]\!] \text{ implies } (\exists t' > t)(\forall t'')(t < t'' < t \rightarrow t'' \in [\![P_{\neg f}]\!]) \quad (3.9)$$

which is equivalent to:

$$t \in [\![P_e]\!] \text{ implies } (\exists t' > t)(\forall t'')(t < t'' < t \rightarrow t'' \notin [\![P_f]\!]) \quad (3.10)$$

Real Model

A model over the real-line for this MSO-formula will comply with the following condition, for every t:

$$t \in [\![P_e]\!] \text{ implies } (\exists t' > t)(\forall t'')(t < t'' < t \rightarrow t'' \notin [\![P_f]\!])$$

$$EC \models Terminates(e, f, t)$$

iff (by 3.8):

$$t \in [\![Happens(e)]\!] \text{ implies } (\exists t' > t)(\forall t'')(t < t'' < t \rightarrow t'' \notin [\![HoldsAt(f)]\!])$$

Suppose:

$$t \in [\![P_e]\!]$$

iff:

$$t \in [\![Happens(e)]\!]$$

implies:

$$(\exists t' > t)(\forall t'')(t < t'' < t \rightarrow t'' \notin [\![HoldsAt(f)]\!])$$

iff:

$$(\exists t' > t)(\forall t'')(t < t'' < t \rightarrow t'' \notin [\![P_f]\!])$$

Since:

$$t \in [\![P_e]\!]$$

implies:

$$(\exists t' > t)(\forall t'')(t < t'' < t \rightarrow t'' \notin [\![P_f]\!])$$

then:

$$MSO \models \forall x(P_e(x) \rightarrow P_{\neg f}(S(x)))$$

String Model

Similarly to an initiating event, a terminating event for a fluent f occurring at time t defines a right-transition point, where:

$$(\exists t' > t)\ (\forall t'')\ ((t < t'' < t') \rightarrow t \in [\![HoldsAt(f)]\!] \rightarrow t'' \notin [\![HoldsAt(f)]\!])$$

Therefore, for an alternation-bounded fluent f, collecting the points t'' where t<t'' $\leq$t' into an interval I_i, then there is a segmentation of the EC-model where $I_i \cap$ [[HoldsAt(f)]] = $\emptyset$ and $I_{i+1} \subseteq$ [[HoldsAt(f)]].

Therefore, a segmentation of an EC-model for Terminates(e,f,t) must satisfy the following conditions:

$$e \in \Delta(I_{i-1}) \text{ implies } \neg f \in \Delta(I_i)$$

iff:

$$s \models \forall x(P_e(x) \rightarrow P_{\neg f}(S(x)))$$

3.3.6 Releases

Releases(e,f,t) indicates that if an event e happens at time t, then f is exempt from inertia after time t. In the EC, this is achieved by treating e as a terminating event so that when f is released from inertia it no longer holds.

Therefore, the translation of Releases(e,f,t) into an MSO-formula is:

$$\forall x(P_e(x) \rightarrow P_{\neg f}(S(x)))$$

and since Releases(e,f,t) imposes the same conditions on a model as Terminates(e,f,t), the proof that they have equivalent models over the real-line and string models is as above.

3.3.7 Trajectory

The meaning of Trajectory(f_1,t,f_2,d) is constrained by Axiom 4, which states that if a fluent f_1 holds at time t, and is not terminated or released between t and a later time t′ =t+d, then the fluent f_2 will hold at t′. The fluent f_2 is usually a fluent-valued function of d, with each distinct d giving a distinct fluent.

An EC-model for Trajectory(f_1,t,f_2,d) will satisfy the following constraint:

$$[(\forall t'')(t < t'' \leq t' \rightarrow t'' \in [\![HoldsAt(f_1)]\!]] \rightarrow t' \in [\![HoldsAt(f_2)]\!] \qquad (3.11)$$

A further issue in translating this predicate into MSO is that MSO does not have an addition function needed for t′ = t+d. However, it is possible to represent Trajectory as a set of MSO-formulas given by a schema, with d acting as a parameter for this schema. Suppose for d$\in \mathbb{R}$, $f_2(d)=f_d$, and x′ =x+d, then

the set of MSO-formulas serving as the translation of Trajectory(f_1,t,f_2,d) is:

$$\lambda_d := (\forall x'')(x < x'' \leq x' \rightarrow P_{f_1}(x'')] \rightarrow P_{f_d}(x')$$

The proofs below will deal with an individual formula given by a particular choice of d. Since this d is arbitrary, the proofs can be generalized to all values for d, and therefore the entire set of formulas given by the schema.

Real Model

A real-line model satisfying this MSO-formula will comply with the constraint:

$$[(\forall t'')(t < t'' \leq t' \rightarrow t'' \in [\![P_{f_1}]\!]] \rightarrow t' \in [\![P_{f_d}]\!]$$

$$EC \models Trajectory(f_1, t, f_d, d)$$

iff (by 3.11):

$$[(\forall t'')(t < t'' \leq t' \rightarrow t'' \in [\![HoldsAt(f_1)]\!]] \rightarrow t' \in [\![HoldsAt(f_d)]\!]$$

Suppose:

$$(\forall t'')(t < t'' \leq t' \rightarrow t'' \in [\![P_{f_1}]\!])$$

iff:

$$(\forall t'')(t < t'' \leq t' \rightarrow t'' \in [\![HoldsAt(f_1)]\!])$$

implies:

$$t' \in [\![HoldsAt(f_d)]\!]$$

iff:

$$t' \in [\![P_{f_d}]\!]$$

Since:

$$(\forall t'')(t < t'' \leq t' \rightarrow t'' \in [\![P_{f_1}]\!])$$

implies:

$$t' \in [\![P_{f_d}]\!]$$

then:

$$MSO \models (\forall x'')(x < x'' \leq x' \rightarrow P_{f_1}(x'')] \rightarrow P_{f_d}(x')$$

String Model

While the above translation is appropriate for models over the real-line, it is not appropriate for string models. For instance, if d=3 so x′ =x+3, then x′ is referring to the third string position after x. But given that each string position can correspond to a temporal interval of arbitrary length, it is unlikely that the string position x′ will correspond to 3 time units after x.

This can be achieved using nominal temporal predicates. Suppose there is a given d∈ $\mathbb{R}$ and times t, t′ such that t′ =t+d. Two temporal predicates P_t and $P_{t'}$ are introduced to the vocabulary with only the point interval {t} satisfying

P_t and the point interval $\{t'\}$ satisfying $P_{t'}$. It can be ensured that x and x′ refer to t and t′ (=t+d) respectively, with the formulas $P_t(x)$ and $P_{t'}(x')$, leading to the translation:

$$\lambda_d := \forall x, x'[[P_t(x) \wedge P_{t'}(x') \wedge \forall x''(x < x'' \leq x' \rightarrow P_{f_1}(x))] \rightarrow P_{f_d}(x')]$$

An EC-model for Trajectory(f_1,t,f_2,d) will have:

$$\forall t''(t < t'' \leq t' \rightarrow t'' \in [\![HoldsAt(f_1)]\!] \text{ implies } t' \in [\![HoldsAt(f_2)]\!]$$

Therefore, a segmentation of this EC-model will contain a sequence of intervals $I_{i-1}I_i \ldots I_jI_{j+1}$, each of which (apart from $I_{i-1} = \{t\}$ at which the initiating event for f_1 takes place) satisfy f_1 where I_i is of the form (t,a] or (t,a) for some a$\in \mathbb{R}_+$ (it must be left-open as an initiating event for f_1 happened at t), and I_j is of the form (b,t′) or [b,t′) (must be right-open as $I_{j+1} = \{t'\}$ and so cannot be included in I_j) for some b$\in \mathbb{R}_+$, with I_{j+1} satisfying both f_1 and f_d.

Therefore, a segmentation of an EC-model which satisfies Trajectory(f_1,t,f_2,d) will comply with the following conditions:

$$I_{i-1} = \{t\} \text{ and } I_{j+1} = \{t'\}$$

and:

$$\forall k(i \leq k \leq j+1 \rightarrow f_1 \in \Delta(I_k)) \text{ implies } f_d \in \Delta(I_{j+1})$$

iff:

$$s \models \forall x, x'[[P_t(x) \wedge P_{t'}(x') \wedge \forall x''(x < x'' \leq x' \rightarrow P_{f_1}(x))] \rightarrow P_{f_d}(x')]$$

3.4 Truthmaker Semantics

Given the MSO translation of an EC predicate ϕ (call it MSO(ϕ)), this formula is satisfied by a string, with the set of strings satisfying it forming a regular language (by the BET theorem).

To formulate the relation between these strings-as-models and MSO formulas in terms of truthmaker semantics, the distinction between a string "supporting" the (truth of) the formula, and exemplifying the (truth of) the formula (Kratzer 1989) must be captured.

A string serving as a model for an MSO formula can be thought of as an index, with the "part" of that string (give a suitable "part of" relation) which actually makes the formula true being the denotation.

According to Fine (2015):

> The idea of truthmaking is the idea of something on the side of the world - a fact, perhaps, or a state of affairs - verifying, or making true, something on the side of language or thought - a statement, perhaps, or a proposition.

Here, the "side of the world" will be linked to string models, whereas the "side of language" will be linked to the MSO-formulas from section 3.3 that they satisfy. Fine notes that a cornerstone of a satisfactory semantics is to ascertain *how* what is in the world makes sentences of the language true.

In the previous sections it was shown how an EC model could be partitioned into a string, and the string would serve as a model for the MSO-translations of the EC formulas. These string models have the potential to contain extraneous information (in the form of fluents), especially if the alphabet is expanded. The goal of this section is to focus on what it is about these various string models that makes the MSO formula true, in line with Fine's observation above.

Truthmaker semantics can be considered a part of the objectual approach (in contrast to the clausal approach (Davidson 1967)), where "objects" of some sort are in a satisfaction relation with sentences. Traditionally, under the objectual approach, these objects have been possible worlds (Montague 1970), however, truthmaker semantics is concerned with situations (Barwise and Perry 1983), or states, entities which make up a possible world.

States/situations can verify a sentence in three ways:

- Loose: The state necessitates the statement
- Inexact: The state is partially relevant to the statement. For example, the fact that John is tall and slim is an inexact verifier of the statement "'John is tall"
- Exact: The state is wholly relevant to the statement. For example, the fact that John is tall is an exact verifier of the statement "John is tall"

Truthmaker semantics is concerned with exact verification.

The states in truthmaker semantics together make up a state space. This state space has a mereological structure, i.e. states are ordered by a part-of relation. The state space may feature "impossible" states such as it being hot and cold at the same time. In contrast to this, a *modalized* state space is also given, which only features possible states.

States can be linked to strings as follows:

- Given an alphabet Σ, the language 2^{Σ} is the state space. For example, if both f and $\neg$f are in the alphabet, some strings will feature both fluents in the same string position, clearly an "impossible state"
- A modalized state space can be given by placing certain conditions or

constraints on the language such as:

$$\boxed{\text{f},\neg\text{f}} \Longrightarrow \emptyset \tag{3.12}$$

which forbids any string in the language from containing a box with both the fluents f and ¬f

- There are at least two candidates for the part-of relation on strings: subsumption ($\trianglerighteq$) and containment ($\sqsupseteq$)
- Exact verifiers will be strings or languages which are wholly relevant to the formula under consideration

Reducing truthmaking to a string being a part of an index requires some attention be paid to contexts or presuppositions. While (3.12) is a general constraint on the modalized state space, certain formulas will need additional specific constraints to rule out impossible states, as will be seen below. These constraints can be thought of as presuppositions (Wilson 1975).

The above can be captured by the following relation, decomposing an index into presuppositions and a denotation (exact verifier):

$$s \models MSO(\phi) \text{ iff } s \in C \cap \langle \text{part-of} \rangle L(\phi)$$

where s is a model for MSO(ϕ), C is the context or presuppositions, $\langle part-of \rangle$ is a suitable part of relation, and L(ϕ) is the denotation or "truthmaker".

3.4.1 Initially

Recall the translation of the EC predicate Initially(f) into an MSO formula:

$$\exists x(P_f(x) \wedge P_0(x))$$

satisfied by strings from the language:

$$\langle \sqsupseteq \rangle \boxed{\text{f,0}}$$

$\langle \sqsupseteq \rangle$ s is the set of strings s′ such that s′ $\sqsupseteq$s.

Strings containing the above length-1 string, without any other constraint may contain the fluents 0 and f in one or more string positions that are not the initial position, for example:

$$\boxed{}\boxed{\text{0,f}}\boxed{\text{f}}$$

This string meets the part-of condition but is clearly not a model for MSO(Initially(f)). That "0" is a fluent that can appear only once, and must appear in the initial string position can be added to the language C, as a presupposition. Therefore, C must comply with the constraint:

$$\langle \trianglerighteq \rangle \boxed{0}\boxed{}^{*}$$

$\langle \trianglerighteq \rangle$ s is the set of strings s′ such that s′$\trianglerighteq$s.

Therefore, every string s has the fluent "0" appearing in the first box. As each string must have a box containing both "0" and f, the initial position must also contain the fluent f, and therefore satisfies MSO(Initially(f)).

3.4.2 HoldsAt and Happens

Temporal Constraints

The constraint (or presupposition) needed in the previous section (that the temporal fluent "0" appears in the first string position) can be thought of as a temporal constraint, fulfilled in the EC by the theory of the real closed field.

This suggests that other constraints about the positions of temporal fluents may be needed for the other predicates.

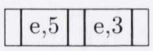
For example suppose the EC-formula:

$$Happens(e, 3) \wedge Happens(e, 5)$$

is given. this can be translated into the MSO-formula:

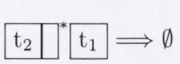
$$\exists x(P_e(x) \wedge P_3(x)) \wedge \exists x(P_e(x) \wedge P_5(x))$$

which would be satisfied by strings such as:



Clearly, the fluent "3" should precede the fluent "5" and the above string is deviant in that respect. Strings which contain temporal fluents are only models of MSO-translations of EC-formulas if the temporal fluents obey order restrictions.

Therefore, given a set of temporal fluents T, for all t_1, $t_2 \in$ T, where $t_1 < t_2$, the language C must obey the following constraint:



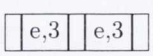
Furthermore, as each temporal fluent marks an indivisible point, the same temporal fluent should not appear in distinct string positions, or the EC-formula:

$$Happens(e, 3)$$

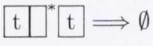
translated into the MSO-formula:

$$\exists x(P_e(x) \wedge P_3(x))$$

could have the following string model:



Therefore, for all t∈T, the language C must obey the following constraint:



$$\Longrightarrow \emptyset$$

Finally, distinct temporal fluents should not appear in the same string positions, or the EC-formula:

$$Happens(e, 3) \wedge Happens(e, 5)$$

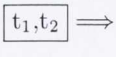
translated into the MSO-formula:

$$\exists x(P_e(x) \wedge P_3(x)) \wedge \exists x(P_e(x) \wedge P_5(x))$$

would be satisfied by string models such as:



Therefore, for all t_1, $t_2 \in$ T, where $t_1 \neq t_2$, the language C must obey the following constraint:



$$\Longrightarrow \emptyset$$

Negation

When discussing the MSO-translation of the EC-formula ¬HoldsAt(f,t) above, negated fluents (¬f) were introduced. Take the following EC-formula:

$$HoldsAt(f, 3)$$

translated into the MSO-formula:

$$\exists x(P_f(x) \wedge P_3(x))$$

which can have the following string model:

$$\boxed{\text{f,}\neg\text{f,3}}$$

which is deviant in the sense that a fluent and its negation appear in the same string position.

Therefore, for all fluents f, the language C must obey the following constraint:

$$\boxed{\text{f,}\neg\text{f}} \Longrightarrow \emptyset$$

3.4.3 Initiates, Terminates, and Releases

The translation of Initiates into an MSO-formula differed from HoldsAt and Happens in that they were existentially quantified conjunctions, whereas the translation of Initiates was a universally quantified conditional.

Therefore, for a string to serve as a model for this formula, the entire string must be taken into account, not just a part of it. Of course, the use of the term "part" here is in relation to length, but note that "part" may also be used in relation to the alphabet. Take an alphabet $\Sigma = \{e_1, e_2, f_1, f_2\}$. Clearly, for the

formula Initiates(e,f,t), the occurrence of symbols e_2 and f_2 are irrelevant.

As a starting point take the following biconditional:

$$s \models \forall x(P_e(x) \longrightarrow P_f(S(x))) \Longleftrightarrow s \in \boxed{\text{e}}\boxed{\ } \Longrightarrow \boxed{\ }\boxed{\text{f}}$$

The language given by:

$$\boxed{\text{e}}\boxed{\ } \Longrightarrow \boxed{\ }\boxed{\text{f}}$$

captures the intuition that whenever an event e occurs, it will be followed by for some interval by the fluent f. As this language completely captures that intuition, there are no additional presuppositions needed.

While strings in this language serve as models for the MSO-formula, the idea is to use a part-of relation to focus on the crucial elements of the string that make the MSO-formula true. The universal nature of the formula makes this more tricky than for the existentially quantified formulas.

Languages in the form of constraints already have a part-of relation built into them, both in terms of length (if e is at any position in the string, not just a particular position, then f will be at the next position), and in terms of alphabet (there can be other fluents in a string that complies with this constraint other than e and f). In this case, an additional part-of relation such as $\sqsupseteq$ or $\trianglerighteq$ is redundant.

In fact, introducing either of the above relations leads to issues with persistence (a formula is *persistent* if when it is satisfied by a string, it is satisfied by all strings of which that string is a part). Choosing $\trianglerighteq$ as the part-of relation allows the string:

$$\boxed{\text{e}}\boxed{\text{f,e}}\boxed{\ }$$

to be a model for MSO(Initiates) as it bears the $\trianglerighteq$ relation to:

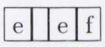


which is a string in L(Initiates). However:



is not a model for MSO(Initiates).

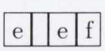
Choosing $\sqsupseteq$ creates the same issue with the alphabet, but also an issue with length, for example, the string:



bears the relation $\sqsupseteq$ to:



which is in L(Initiates). However:

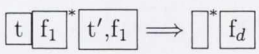


is not a model for MSO(Initiates).

Clearly, if a part-of relation is necessary, it must be a trivial one, i.e. equality.

3.4.4 Trajectory

MSO(Trajectory) is satisfied by strings from the following language:



(3.13)

While it may seem that similar persistence issues would affect this formula as with MSO(Initiates), there is a crucial difference. Since the fluent f_d is related to specific times t and t′, a string which contains a string which satisfies MSO(Trajectory) will also satisfy this formula. This larger string cannot contain (either by $\sqsupseteq$ or $\trianglerighteq$) t and t′ in other positions as they are nominals, only appearing once. Therefore, it is impossible that this larger string will subsume the left-hand side of (3.13) and not subsume the right-hand side. Though these part-of relations can be used with MSO(Trajectory), they still have a trivial character, given that the constraint (3.13) has an inherent part-of relation.

Finally, the appearance of temporal fluents in the string models indicates that the temporal constraints given in previous sections must apply.

3.5 Finite-State Temporality and Context-Free Languages

Some natural language constructions concerning events are difficult to represent in this regular framework. These constructions suggest a relationship between the frequency of events, similar to the dependency of symbol frequencies on other symbol frequencies found in many context-free languages ($L = a^n b^n$). Together with their increase in complexity over regular langauges, this makes context-free languages a natural area of interest in this field.

There are consequences for entailments based on these languages. Suppose given two languages A and B, it is desired to show that A entails B. This can be achieved by intersecting A and the complement of B giving a set of "counterexamples" to the entailment, and checking if the resulting language is empty (i.e. there are no counterexamples), a decidable problem for context-free languages.

The superposition operation can be used to add in background constraints to the entailment, essentially does A entail B when certain fluents are added to the strings in the language. Given that superposing a context-free language can result in a language that is not context-free, this would prevent certain entailments being possible.

As an example, take the expression "A as often as B", where A and B can be thought of as events. This construction implies a frequency relationship between the occurrence of two events, where A occurs at least as many times as B, possibly more. These events do not have to occur in an ordered sequence, where a sequence of As are followed by a sequence of Bs, As and Bs can occur in any order as long as the overall frequncy relationship is maintained. To be accurate, we must also allow for "instants" of time that separate any occurrence of the events we are interested in. Representing event A as the symbol a, event B as the symbol b, and an instant of time in which events may be occurring, but which are not relevant to our analysis by $\square$, we get strings of the form $a\square^*b$, $a\square^*b\square^*b\square^*a$, $b\square^*b\square^*a\square^*a\square^*a$ etc. These strings form a context-free language.

Moens and Steedman (1988) highlight the complicated nature of the phrase "when". They suggest that "When A, B" implies not a strictly temporal relationship, but a causal one, making it a prime candidate for representation by a context-free language. Note that "When I swear, I put money into the swear jar" implies that if I swear twice, I put money into the jar twice, but not necessarily in a particular temporal order. I may swear twice during the day, and have to wait until I get home to put money in the jar.

More formal (and seemingly less natural) constructions such as "an equal number of times as", while rare, do have a place in more formal literature such as legal documents. This particular construction appears in locations as varied as "The Federal Code of the United States of America" and the bye-laws of the

town of New Canaan,CT: "he shall choose alternates in rotation so that the alternates chosen by the Chairman shall be seated as nearly an equal number of times as is possible".[1]

Our analysis is restriced to the case of there being a frequency relationship between two types of events. It should be noted that the addition of a third event would lead to strings characterised by languages that are not context-free, similar to the difference between $a^n b^n$ and $a^n b^n c^n$. While these constructions tend to seem less natural ("I cried as often as I laughed, and I laughed as often as I sang"), they cannot be discounted.

The above linguistic data, while by no means exhaustive, provides a steady base from which to explore context-free languages in a finite-state temporality framework. The ubiquity of the phrases "when" or "whenever" highlights the need for this extension, while their causal nature, as opposed to temporal nature, suggests further ontological applications.

Superposition, as the central operation in the finite-state temporality framework, must be re-examined in light of our inclusion of context-free languages. The key question is whether the result of superposing a context-free language with either a regular language or another context-free language, is itself regular, context-free, or otherwise.

Proposition 1 *The set of context-free languages is not closed under superposition.*

Proof(by counter-example): Let the set $\{\phi\}$ be represented by the symbol $\boxed{\phi}$, and the set $\{\psi\}$ be represented by the symbol $\boxed{\psi}$. The language $L_1 = \boxed{\phi}^n \boxed{\psi}^n$ is context-free, as is the language $L_2 = \boxed{\phi}^m \boxed{\psi}^{2m}$. L_1 is given by the grammar:

[1] http://ecode360.com/9045062

$S \rightarrow \boxed{\phi} S \boxed{\psi}$

$S \rightarrow e$

and L_2 by the grammar:

$S \rightarrow \boxed{\phi} S \boxed{\psi}\boxed{\psi}$

$S \rightarrow e$

The superposition of these two languages will contain strings consisting of three possible symbols: $\{\phi\} \cup \{\phi\} = \{\phi\}$ represented as $\boxed{\phi}$, $\{\phi\} \cup \{\psi\} = \{\phi, \psi\}$ represented as $\boxed{\phi\psi}$, and $\{\psi\} \cup \{\psi\} = \{\psi\}$ represented as $\boxed{\psi}$.

Strings in the language L_1 have length 2n, and strings in the language L_2 have length 3m. Strings can only be superposed if they have equal length, therefore only strings of length 6r, for some positive integer r, from both languages can be superposed, resulting in strings of the same length. Strings in L_1 will consist of 3r $\boxed{\phi}$s followed by 3r $\boxed{\psi}$s, and strings in L_2 will consist of 2r $\boxed{\phi}$s followed by 4r $\boxed{\psi}$s. The superposition of these two strings will consist of 2r $\boxed{\phi}$s superposed with $\boxed{\phi}$s, r $\boxed{\phi}$s superposed with $\boxed{\psi}$s, and 3r $\boxed{\psi}$s superposed with $\boxed{\psi}$s, resulting in strings of the form $\boxed{\phi}^{2r}\boxed{\phi\psi}^{r}\boxed{\psi}^{3r}$. Introducing a homomorphism from $\boxed{\phi}$*com* to 'a', from $\boxed{\phi\psi}$ to 'b', and from $\boxed{\psi}$ to 'c', we have an equivalent language $a^{2r}b^{r}c^{3r}$.

If this language were context-free, given that it is infinite, there would be some constant K such that any string longer than K would be representable as a string uvxyz such that v and y are not empty and are pumpable. If we choose the string $a^{2K}b^{K}c^{3K}$ as a string longer than K, we should be able to factorize it in this manner. If we chose v to have both as and bs or both bs and cs, then upon pumping it, we would have bs before as or cs before bs, which would result in a string not in our language. The same considerations apply to choosing y. Therefore v and y must each contain only as, or only bs, or only cs. Pumping v and y would therefore increase the number of one or two of the symbols but not all three, thereby losing the frequency relationship between the three symbols.

The language cannot be context-free.□

Proposition 2 *The superposition of a context-free language with a regular language is context-free.*

Proof: Given L_1, a context-free language, and L_2, a regular language, let $P = \langle Q_P, \Sigma, \Gamma, \Delta_P, q_{P0}, F_P \rangle$ be a pushdown-automaton accepting L_1 and $A = \langle Q_A, \Sigma, \delta_A, q_{A0}, F_A \rangle$ be a finite-state-automaton accepting L_2. Δ_P is the set of transitions of the form $(q_i, a, A) \rightarrow (q_j, \gamma)$ interpreted as: when in state q_i , with input symbol a, and symbol A at the top of the stack, go to state q_j and replace A by the string γ, and δ_A is the set of transitions of the form $(q_i, a) \rightarrow (q_j)$ interpreted as: when in state q_i with input symbol a, go to state q_j. A pushdown automaton is formed by taking $R = \langle Q_P \times Q_A, \Sigma, \Gamma, \Delta_{P\times A}, (q_{P0}, q_{A0}), F_P \times F_A \rangle$, with transitions $\Delta_{P\times A}$ constructed as follows:

1. If Δ_P contains a rule of the form $(q_0, e, e) \rightarrow (q_1, S)$, then $\Delta_{P\times A}$ contains a rule of the form $((q_0, q_0), e, e) \rightarrow ((q_1, q_0), S)$.

2. If Δ_P contains a rule of the form $(q_1, e, A) \rightarrow (q_1, \gamma)$, then $\Delta_{P\times A}$ contains rules of the form $((q_1, q_x), e, A) \rightarrow ((q_1, q_x), \gamma)$ for every $q_x \in Q_A$.

3. If Δ_P contains a rule of the form $(q_1, a, a) \rightarrow (q_1, e)$, then $\Delta_{P\times A}$ contains rules of the form $((q_1, q_x), a \cup b, a) \rightarrow ((q_1, q_y), e)$ if and only if there is a transition $(q_x, b) \rightarrow (q_y)$ in δ_A.

The new transitions are akin to running the PDA and FSA in tandem, where a state (q_x, q_y), while strictly a state of R, can be thought to represent the simultaneous states of P and A. A rule of type 1 and rules of type 2 perform the same stack operations as the PDA they were derived from. Therefore, R can produce on its stack the same set of strings that P produces on its stack. No input symbol is being read while these stack operations are performed, therefore

R should remain in state (q_x, q_y). Rules of type 3 ensure that if R is in a state (q_1, q_y) with an input symbol $a \cup b$, and encounters the terminal symbol a on its stack, along with there being a transition in A from q_y to q_z on input b, then R will transition to state (q_1, q_z), and delete a from its stack. These are exactly the states that P and A would separately be in upon reading input a and b respectively. Thus, if P reads a string $a_1 \dots a_n$ and is in a final state with an empty stack (i.e. P accepts this string), and A reads a string $b_1 \dots b_n$ and is in a final state (i.e. A accepts this string), then R will be in a final state upon reading the superposition of these two strings. If P accepts a language L_1 and A accepts a language L_2, then R will accept $L = L_1 \& L_2$.□

If we superpose the context-free language that represents "I laughed as often as I cried" with the regular langauge that represents "an hour" to get a language representing "In an hour, I cried as often as I laughed", this language, as the superposition of a context-free langauge and a regular language, will be context-free.

Chapter 4

Type Coercion

4.1 Introduction

It may have been noted that the EC's notion of "event" as a punctual occurrence that causes stative change is somewhat limited when it comes to describing the broad range of events encountered in aspectual classification. An obvious shortcoming is that many events have duration and are not punctual.

VLH (p.84) compose punctual events, force fluents, and stative fluents into a structure called an "eventuality" (Bach 1986), intended to represent this broader range of events. An *eventuality* is composed of four fluents (f_1,f_2,e,f_3), where:

- f_1 is a fluent which represents an activity, something which exerts a force
- f_2 is a parameterized fluent, representing a parameterized object or state, which is driven by the force f_1
- e is the culminating event, representing a canonical goal
- f_3 is a fluent which represents the state of having achieved the goal

Note that it is possible for one or more of the argument slots to be empty.

VLH's eventualities describe aspectual types by filling certain argument slots in the 4-tuple with the appropriate fluents/event and leaving other slots blank. An eventuality can be a:

- State: $(-,-,-,+)$
- Activity (strict): $(+,-,-,-)$
- Activity (wide): $(+,+,-,-)$
- Accomplishment: $(+,+,+,+)$
- Achievement: $(-,-,+,+)$
- Semelfactive: $(-,-,+,-)$. Note that this is called a point in VLH.

Of course, there is more to aspectual types than just the presence or absence of fluents. These fluents must play the appropriate "role" in the eventuality. VLH achieve this by constraining where the fluents can appear in EC-predicates:

- f_1 can only appear as the first argument in Trajectory, and not at all in Releases, i.e. it is the force which drives the change.
- f_2 and f_3 can appear only in the third argument of Trajectory, and in Releases, i.e. they can be affected by a force and can be exempt from inertia.

Eventualities are clearly similar to Moens and Steedman's event nucleus, with f_1 corresponding to the preparatory process, e corresponding to the culmination, and f_3 corresponding to the consequent state. The fluent f_2, however, is not a part of Moens and Steedman's event nucleus. This fluent will allow a relationship between the preparatory process and the culmination to be expressed, with the preparatory process causing changes in f_2, and certain values of f_2 being associated with the culmination (for example, being at the maximum

height of the mountain being associated with the culmination of "reaching the top of the mountain").

The relationships between the elements of the eventuality (essentially the relationships between the elements of the event nucleus) are given by EC-formulas:

- f_1 and f_2: The relationship between the preparatory process and the changing partial object is given by a Trajectory formula with f_1 as the first argument, and f_2 as the third.

- f_2 and e: When the changing, partial object reaches some maximum (height for "climbing a mountain") or minimum (water level for "emptying a bucket") the culminating event occurs. This is given by a formula such as:

$$HoldsAt(f_2(max), t) \longrightarrow Happens(e, t)$$

  i.e. when f_2 reaches its maximum value the culminating event e happens.

- e and f_3: The relationship between the culminating event and the consequent state is given by an Initiates formula, with the event e initiating the state f_3.

- e and f_1: While e is a culminating event resulting in a consequent state f_3, it is also a terminating event for the force f_1, with this relationship given by a Terminates formula, with the event e terminating the force f_1.

While it may be proposed that these formulas contribute to aspectual type, as well as the elements of the 4-tuple, the function of the formulas is to relate the elements of the eventuality to each other. Moens and Steedman propose that an event nucleus underlies all aspectual types:

> ...each primitive entity of a given type, such as the culmination

event of *Harry's reaching the top*, carries intimations of other associated events and states, such as the process by which the culmination was achieved and the consequent state that followed.

i.e. an eventuality perceived as a certain aspectual type still has the other associated elements of the eventuality, though these are not focussed on. Aspectual type is a result of the focus or perspective of the speaker. In terms of the EC, the formulas given above, relating the elements of the eventuality to each other, therefore contribute to the "contingency" relationships of the event nucleus and not aspectual type. It is the choice of vocabulary (which elements of the 4-tuple are expressed) that determines aspectual type.

Armed with the translations of EC-formulas into strings, some interesting comparisons can be made between aspectual classification in the EC and aspectual classification using strings.

Fernando (2015*a*) defines the traditional notions of telicity and durativity in terms of string properties, with an event, represented by a string (str(e)), being:

- Σ-telic if str(e)$\trianglerighteq \boxed{\neg\phi}^{+}\boxed{\phi}$ for some fluent $\phi \in \Sigma$, capturing the intuition that a telic event ends with a state holding, which previously did not hold. For example, climbing a mountain ends with the state of being at the top of the mountain.

- Σ-durative if bc(str(e)) has length ≥ 3. Strings of length 2 capture simple transitions such as achievements and semelfactives. Having length greater than 3 is meant to capture the intuition that durativity corresponds to having internal structure.

A force f is decomposed into two fluents:

- ap(f), representing the application of force f. Note that forces are defeasible, i.e. an opposing force may cancel it out, leading to the second

fluent,

- ef(f), representing that the force f was effective. For some cases this may just be the ϕ from the above definition of telicity. However, in other cases (semelfactives for instance), there are no "salient" consequent states. In this case, ef(f) just means that the force has been effectively applied, thought it could also be mapped to the short-term consequences that semelfactives can have ("has coughed").

As seen in the literature review, events (non-statives) have been placed into four categories, depending on whether they are positive or negative for two binary features: telicity and durativity, as in the table below:

	−durative	+durative
−telic	semelfactive	activity
+telic	achievement	accomplishment

Table 1. *Aspectual categories of events*

suggesting the following table with the four categories represented by strings:

	−durative	+durative
−telic	semelfactive $ap(f) \mid ef(f)$	activity $ap(f) \mid ap(f), ef(f) \mid ef(f)$
+telic	achievement $\neg\varphi \mid \varphi$	accomplishment $\neg\varphi, ap(f) \mid \neg\varphi, ap(f), ef(f) \mid \varphi, ef(f)$

Table 2. *A minimal picture of ±durativity and φ-telicity*

Iterating applications of a force f gives the language:

$$L(f) := \boxed{\text{ap(f)}} \; \boxed{\text{ap(f),ef(f)}}^{*} \; \boxed{\text{ef(f)}}$$

somewhat like Moens & Steedman's event nucleus, suggesting the basic unit of an event is the application of a force.

The following works show that forces, not just states, to describe continuous change. Beavers (2013) uses the idea of a "force recipient" from Hovav and Levin (2001) to make aspectual classifications. Various linguistic tests are diagnostic of whether a force causes change or not:

- What happened to the lamp is that John rubbed it.
- #What happened to the lamp is that John saw it.
- John rubbed the lamp but nothing changed about it.

Beavers classifies "rub" as having the potential for change, positing the existence of a "latent scale". That the lamp in the above example could undergo change can be seen in the following example:

- John rubbed the lamp clean.

The combination of "happening" and "seeing" is not always problematic. For instance, "What happened to the password is that John saw it". Here, perhaps, the seeing of the password causes a change of state from being secret to not being secret.

As well as the possibility of undergoing change, Beavers makes aspectual classifications based on the complexity of the underlying scale, with a simplex scale having two subparts, and a complex scale having more than two subparts. Punctuality of an event is seen as resulting from an object undergoing change along a simplex scale, while durativity is seen as resulting from an object undergoing change along a complex scale.

Beaver's analysis fails when applied to verbs such as "sit", typically thought of as states, but unlike most states, able to appear in the progressive. Firstly, note that sitting can be seen as application of a force:

- What happened to the chair is that John sat on it (Here the durative act of sitting is meant, rather than the punctual initial act of sitting down).
- ?John sat on the chair and nothing changed about it

The latter may be acceptable for a hard, wooden chair, but unacceptable for a soft armchair that sags under the weight of a person sitting in it.

The potential change a chair can undergo while being sat on is therefore along a simplex scale, going from its initial state to a sagging state during the initial event of being sat upon. The following durative event of sitting causes no change whatsoever. The question arises as to how the progressive applies to this durative sitting, when the progressive is thought to require multiple "stages" (Landman 1992). Given this, it seems the notion of a continuous force is necessary to explain when the progressive can occur.

It is interesting to note that these "stative" verbs that can appear in the progressive coincide with what Maienborn (2007) calls "Davidsonian" states, in contrast to "Kimian" states (predicates such as "tall"). Key properties of these two different types of states are given, Davisonian states being perceptible, and locatable in space and time, while Kimian states are neither perceptible nor locatable in space (though they are locatable in time). It is possible these properties can be related to forces, and their spatiotemporal nature.

The four strings representing the aspectual categories can be obtained from L(f) by the following operators:

- dur$(L) := L\&\square\square\square^{+}$
- non-dur$(L) := L - \text{dur}(L)$
- cul$(L, \phi) := L\&\boxed{\neg\phi}^{+}\boxed{\phi}$

Kennedy and Levin (2008) propose that the behaviour of incremental theme verbs ("eat"), degree achievements ("cool"), and directed motion verbs ("as-

cend") arises from one shared element of their meanings: a measure of the degree to which an object changes relative to some scalar dimension over the course of an event. Under this analysis, verbs based on gradable adjectives like "cool" directly lexicalize measure functions which (given an object and a time) return a value of a degree on a scale. In the case of "cool", this is likely to be a degree on a temperature scale.

Fernando (2015*a*) assumes a set D[ψ] of $\psi-$degrees (where ψ might represent the temperature of the soup say), the fluent $\downarrow$deg[ψ] can appear in a string position if there is some degree that deg[ψ] is less than or equal to, where in the previous string position it is greater than that same degree, so deg[ψ] has fallen between string positions. This can be expressed as the biconditional constraint:

$$\boxed{}\boxed{\downarrow\text{deg}[\psi]} \Leftrightarrow \sum_{d \in D[\psi]} \boxed{deg[\psi] > d}\boxed{d \geq deg[\psi]}$$

with $L \Leftrightarrow L'$ abbreviating the intersection of $L \Rightarrow L'$ and $L' \Rightarrow L$.

There is a key difference between the fluent $\downarrow$deg[ψ] and other stative fluents. A string containing multiple occurrences of a stative fluent ϕ, such as:

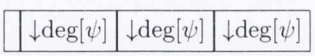
$$\boxed{\phi}\boxed{\phi}\boxed{\phi}$$

can be compressed into the string:

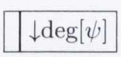
$$\boxed{\phi}$$

under the "no time without change" dictum, where transitions between boxes mark change. This can only be thought of as three separate observations of $\boxed{\phi}$ if there are other background fluents in the string to distinguish the three occurrences. It is possible to expand one observation to three as long as there

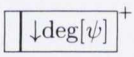
are background fluents to distinguish the three observations. The string with three boxes is no more informative than the string with one box. However, while the string:



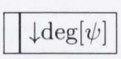
can be compressed into the string:



the two strings are not strictly equivalent. The first string has the degree falling between each of the boxes. The second string could have the degree fall between the two boxes but then stay constant. For this reason, the language:



will not automatically be compressed to the string:



↓deg[ψ], denoting that the degree has fallen between boxes, is less informative than a fluent representing the precise degree, which is the position taken by VLH. The parameterized fluent $f_2(x)$ is a function from degrees to fluents. Whether knowing the exact degree at each time point is necessary is an open question. For example, does Mary need to know John's exact height on a mountain at each temporal point to understand that John climbed the mountain? It seems enough for Mary to understand that John's height on the mountain continuously increased until he reached the peak. Hoever, there does need to be at least one degree to account for accomplishments: the maximum/minimum.

A representation of the peak of the mountain is useful (and maybe even necessary). For this reason, a fluent deg_m is introduced (m as in max/min). This can be associated with the EC's use of a constant to represent the completion of the "partial" object. For example, VLH use the constant c to represent the completed house (house(x) becomes house(c)).

Of course, it is possible that the maximum/minimum degree is not a fixed value and may change based on context or circumstance. For example, there are many different stages at which a house could be considered to be complete. Consider a house consisting of a main building with a conservatory extension. Is the house complete when the main building is finished, or when the conservatory extension is? A position will not be taken on this, apart from noting that no matter which fluents are used to represent this "partial object", be they individual degrees or $\downarrow\text{deg}[\psi]$, deg_m must be the to the right of all other fluents representing degrees. In the case of individual degrees, no degree can exceed the maximum, and in the case of $\downarrow\text{deg}[\psi]$, the degree cannot increase once the maximum has been reached. This can be achieved through the following constraints:

$$\boxed{deg_m}\boxed{}^{*}\boxed{\downarrow\text{deg}[\psi]} \Longrightarrow \emptyset$$

While individual degrees may or may not be necessary, they can be implemented in FST, though only a finite number can be used at any time (to keep the alphabet finite). Suppose there are a set of degrees, where each degree is a real-number. For example, the completion of a house given as a percentage may be a real-number between 0 and 100. For each degree $d \in D[\psi]$ the following fluent can be defined:

$$\psi_{=d} := deg[\psi] = d$$

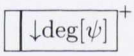
giving an infinite set of fluents.

Given that these degrees are being used to describe continuous change, and the precision of the real-number line, the above fluent will only hold of point intervals (unless there is some sort of pause stage (Landman and Rothstein 2010). When the level of precision given by individual degrees is not necessary, fluents specifying the exceeding of a threshold value can be used, such as the following (repeated from above):

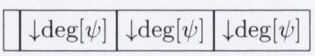
$$deg[\psi] \leq d$$

In face, these may be preferred, as they hold over intervals and provide a coarser granularity.

Given a string from the language:



such as:



the question arises as to what is causing the degree to fall between each of these boxes.

For a particular degree $d_i \in D[\psi]$, the fluent ψ_{d_i} can be defined as:

$$\psi_{d_i} := deg[\psi] \leq d_i$$

giving the following:

$$\neg\psi_{d_i} = deg[\psi] > d_i$$

allowing the following biconditional constraint to be defined:

$$\boxed{\boxed{\downarrow \deg[\psi]} \Leftrightarrow \sum_i \boxed{\neg \psi_{d_i} \mid \psi_{d_i}}}$$

The summation over i ensures the fluent $\downarrow \deg[\psi]$ will appear whenever the degree falls, and not just for one degree fall.

Given the inertial law (Fernando 2008):

$$\boxed{\boxed{\phi} \Longrightarrow \boxed{\phi}\boxed{} + \boxed{force(\phi)}}$$

then:

$$\boxed{\boxed{\downarrow \deg[\psi]} \Leftrightarrow \sum_i \boxed{force(\psi_{d_i})}}$$

i.e. for every box that contains $\downarrow \deg[\psi]$, the previous box contains a force for ψ_{d_i}.

With ap(f)=force(ϕ), and ef(f)=$\downarrow \deg[\psi]$, the string:

$$\boxed{\boxed{} \mid \downarrow \deg[\psi] \mid \downarrow \deg[\psi] \mid \downarrow \deg[\psi]}$$

becomes:

$$\boxed{\text{ap(f)} \mid \text{ap(f)},\downarrow \deg[\psi] \mid \text{ap(f)},\downarrow \deg[\psi] \mid \downarrow \deg[\psi]}$$

The correspondence between the EC's aspectual classification and FST's is relatively clear. While an EC eventuality is a 4-tuple, an FST eventuality is a string. Interestingly, both formalisms can be seen as highlighting the importance of a finite vocabulary/alphabet with:

- f_1 in the EC corresponding to ap(f) in FST

- $f_2(x)$ in the EC corresponding to either:

  1. $\downarrow \deg[\psi]$ and $\deg_m$, or
  2. $\{\psi_{=d} | d \in D[\psi]\}$ and $\deg_m$

  depending on the required level of precision.

- e in the EC corresponding to force(ϕ), which may be the same as ap(f)

- f_3 in the EC corresponding to ϕ

The mapping is not completely neat, with the EC featuring "inchoative" events which initiate and terminate forces (forces being quasi-stative in the EC), and the event which terminates the force being the same event which initiates the consequent state.

4.2 Type Coercion

It is often noted in the literature on aspect and the Vendler categories that verbs, verb phrases, or sentences are **typically** members of a certain category, allowing that in certain contexts they can be interpreted as members of a different category.

Pulman (1997) notes that although aspectual status is not "an objective semantic property of the situations sentences describe", it must be determined for the entailments of a sentence to be understood. He gives a number of instances of when aspectual status needs to be determined:

- A sentence which describes a state does not advance the temporal reference point. Sentences describing events do:

  - The printer jammed. The paper was too thick.
  - The printer jammed. The warning light came on.

The printer jamming and the paper being too thick can be evaluated at the same temporal point, while the warning light coming on is implied to be happening at a later time than the printer jamming.

- Aspectual status can determine whether a sentence describes one event or a series of events:
  - The princess protested as the photographer interrupted. Her protest was ignored.
  - The princess protested throughout her trip. Her protests were ignored.

  Given that the structure of "protest" is that of a relatively instantaneous happening, it must be given an "iterative" reading in the second example.

- Determining the aspectual differences between "build" and "pitch" on one hand, and "drive" and "hire" on the other leads to the conclusion that the first and third sentences describe events that happened over the entire time period, while the second and fourth sentences do not, despite their superficial similarities:
  - Joe was building his house there for a year
  - Joe pitched his tent there for a year
  - Joe drove a car for a day
  - Joe hired a car for a day

According to Kamp and Reyle (1993), "the central problem for the theory of aspect is to determine how the aspectual characteristics of complex phrases are determined by those of their parts". This may seem to suggest that aspectual status is a compositional phenomenon, especially given data such as

the following where occurrence with different modifiers gives different aspectual status:

- The soldiers arrived in an hour
- The soldiers arrived for an hour

though Pulman notes that determination of aspectual status "seems to fall in the middle ground between compositional, syntax-driven semantic assembly, and full-blown contextual reasoning".

4.3 Activities $\rightsquigarrow$ Accomplishments

Moens and Steedman (1988) treat the perfect construction as a function requiring a culmination as input:

- Harry has reached the top

and a consequent state as output. The use if the perfect usually requires that the consequent state still holds to some degree. For example:

- I have spilled my coffee

is no longer appropriate after the spillage has been cleaned.

If the culmination occurs with an activity, it will be coerced into a culmination, subject to the limitations imposed by contextual knowledge. An activity can be coerced into a culmination which itself may have a preparatory process and consequent state.

Pulman argues that the perfect is also appropriate with a culminated process (accomplishment), citing the example:

- Joe has grown old

where the consequent state is that Joe is old, and not a new consequent state from "bundling" the accomplishment into a culmination and adding new preparatory processes and consequent states. This indicates that an activity can be coerced into an accomplishment simply by adding a culmination and consequent state.

In VLH, an activity can be represented in two ways:

- Strict: in which the eventuality structure is of the form $(f_1, -, -, -)$ and the only element of the scenario containing the fluent f_1 is of the form:

  - Initiates(start,f_1,t)

  where start is the event that initiates the activity f_1

- Wide: in which the eventuality structure is of the form $(f_1, f_2, -, -)$ and as well as the Initiates formula from the strict activity, the scenario must relate f_1 to f_2 through the Trajectory predicate:

  - HoldsAt(f_2(x),t)$\rightarrow$ Trajectory(f_1,t,f_2(x+g(d)),d)

For these activities to be coerced into accomplishments, a culmination e, and a consequent state f_3 must be added, and the scenario augmented with formulas relating f_2 to e, and e to f_1 and f_3:

- HoldsAt(f_1,t)$\wedge$ HoldsAt(f_2(max),t)$\rightarrow$ Happens(e,t)
- Terminates(e,f_1,t)
- Initiates(e,f_3,t)

This can be achieved in FST by augmenting the alphabet with the fluents:

- force(f_3) (equivalent to the event e in the EC)
- f_3

The addition of these fluents to the alphabet means the following constraints describing the event nucleus can be applied:

- $\boxed{\text{ap(f), deg}_m} \Longrightarrow \boxed{\text{force(f}_3\text{)}}$
- $\boxed{\text{force(f}_3\text{)}\,|\,} \Longrightarrow \boxed{\,|\,\neg\text{ap(f}_1\text{)}}$
- $\boxed{\text{force(f}_3\text{)}\,|\,} \Longrightarrow \boxed{\,|\,\text{f}_3}$

To see how the change in alphabet, which causes previously vacuous constraints to now be applicable, leads to a change in aspectual type, take a string which contains an activity as a subpart, i.e. contains a string from the language:

$$\boxed{\text{ap(f}_1\text{)}}\boxed{\text{ap(f}_1\text{)},\downarrow\text{deg}[\psi]}^*\boxed{\downarrow\text{deg}[\psi]}$$

where $\text{ef(f}_1\text{)} = \downarrow\text{deg}[\psi]$. If a contextual minimum deg_m is given, and the force is applied long enough for it to be reached, then the string will contain as a subpart a string from the following language:

$$\boxed{\text{ap(f}_1\text{)}}\boxed{\text{ap(f}_1\text{)},\downarrow\text{deg}[\psi]}^*\boxed{\text{ap(f}_1\text{), deg}_m}$$

By the constraints given above, the final box must contain $\text{force(f}_3\text{)}$:

$$\boxed{\text{ap(f}_1\text{)}}\boxed{\text{ap(f}_1\text{)},\downarrow\text{deg}[\psi]}^*\boxed{\text{ap(f}_1\text{), deg}_m\text{,force(f}_3\text{)}}$$

and by the constraints given above must be followed by a box containing f_3, meaning the string contains a string from the following language as a subpart:

$$\boxed{\text{ap(f}_1\text{)}}\boxed{\text{ap(f}_1\text{)},\downarrow\text{deg}[\psi]}^*\boxed{\text{ap(f}_1\text{), deg}_m\text{,force(f}_3\text{)}}\boxed{\text{f}_3}$$

The block compression of this language has length greater than 3 and is therefore durative.

4.4 Achievements $\rightsquigarrow$ Accomplishments

Moens and Steedman view the progressive as a function which takes a preparatory process (or activity) as input:

- The president is speaking

and results in a (progressive) state that describes the process as in progress at the reference time.

Achievements are viewed as being instantaneous, so it is difficult to see how they could be described as being in process at a time. Yet occasionally achievements do appear in the progressive:

- John was reaching the top

Moens and Steedman suggest that a preparatory process is added to the culmination and it is this which is asserted to be ongoing at the reference time. Note that as a preparatory process can end before the contingent culmination is reached, the progressive of these culminations does not necessarily assert that the culmination actualy occurred:

- Mike was winning the race when he was overtaken by another competitor

Instead of adding a preparatory process, Pulman allows that the culmination be "stretched" into a preparatory process with a new culmination.

VLH deal with these kinds of coercion by adding the fluents f_1 and f_2 to the eventuality structure, and augmenting the scenario with the following formulas:

- Initiates(start,f_1,t)
- Releases(start,f_2,t)
- HoldsAt(f_2(x),t)$\rightarrow$ Trajectory(f_1,t,f_2(x+g(d)),d)

and for a particular value c:

- HoldsAt(f_1,t)$\wedge$ HoldsAt(f_2(c),t)$\rightarrow$ Happens(e,t)

This can be achieved in FST by augmenting the alphabet with the fluents:

- ap(f_1), and either
- $\uparrow$deg[ψ] and deg_m, or
- $\{\psi_{=d} | d \in D[\psi]\}$ and deg_m

The addition of these fluents to the alphabet means the following constraints describing the event nucleus can be applied:

- $\boxed{\text{ap(f)}, \text{deg}_m} \Longrightarrow \boxed{\text{force}(f_3)}$
- $\boxed{\text{force}(f_3)}\boxed{\ } \Longrightarrow \boxed{\ }\boxed{\neg\text{ap}(f_1)}$
- $\boxed{\text{ap}(f_1)}\boxed{\ } \Longrightarrow \boxed{\ }\boxed{\downarrow\text{deg}[\psi]}$

giving strings from the language:

$$\boxed{\text{ap}(f_1)}\boxed{\text{ap}(f_1),\downarrow\text{deg}[\psi]}^*\boxed{\text{ap}(f_1),\ \text{deg}_m,\text{force}(f_3)}\boxed{f_3}$$

which, as in the previous section, is durative and telic, i.e. an accomplishment.

4.5 Accomplishments $\rightsquigarrow$ Activities

As well as achievements, accomplishments (culminated processes) can occur in the progressive:

- John was climbing the mountain

As noted above, progressives require their "input" to be an activity (preparatory process). An accomplishment can be coerced into an activity by "stripping off" the culmination. It is the process which is stated to be ongoing at the

reference time. As with achievements, the culmination is not asserted to have taken place, and can be explicitly denied, as in:

- John was climbing the mountain, but didn't reach the peak

VLH treat this as a case of "subtractive coercion", where formulas are removed from the scenario. They discuss the case of:

- John drank a glass of wine

which is an accomplishment, versus:

- John drank wine

which is an activity.

Both eventualities have a Trajectory predicate in their scenario, viewing "drinking" as a force which changes the amount of wine consumed. Therefore, both the accomplishment and the activity have a formula of the following form in their scenarios:

- HoldsAt(f_2(x),t)$\rightarrow$ Trajectory(f_1,t,f_2(x+g(d)),d)

where f_1 is the activity, and f_2 the changing partial object.

However, the accomplishment also has the following formulas in its scenario:

- Terminates(e(c),f_1,t)
- HoldsAt(f_2(c),t)$\wedge$ HoldsAt(f_1,t)$\rightarrow$ Happens(e(c),t)

Here, f_2(x) is the quantity of wine consumed. When this amount has a bound on it, such as having consumed the entire glass, this can be represented by the fluent f_2(c). In the above scenario, when this quantity is consumed a terminating event e(c) occurs which stops f_1 (in this case the "drinking").

By deleting the last two formulas from the scenario, the contextual max/min given by the constant c is no longer related to a culminating event. Furthermore,

the absence of this culminating event ensures no consequent state is reached (though by deleting the formula relating the culminating event to the consequent state, it could not be reached anyway).

This can be achieved in FST by removing the following fluents from the alphabet:

- force(f_3)
- f_3

which causes the following constraints to no longer apply:

- $\boxed{\text{ap(f)}, \deg_m} \Longrightarrow \boxed{\text{force}(f_3)}$
- $\boxed{\text{force}(f_3)}\boxed{\ } \Longrightarrow \boxed{\ }\boxed{\neg\text{ap}(f_1)}$
- $\boxed{\text{ap}(f_1)}\boxed{\ } \Longrightarrow \boxed{\ }\boxed{\downarrow\deg[\psi]}$

essentially the reverse of the additive coercion example from above.

With no contextual max/min supplied, and no constraints giving culminating events or consequent states, the application of a force (representing the preparatory process) will cause change that has an open-ended character, characteristic of activities, such as "walking" with no inherent endpoint, such as the language:

$$\boxed{\text{ap}(f_1)}\boxed{\text{ap}(f_1),\downarrow\deg[\psi]}^*\boxed{\downarrow\deg[\psi]}$$

Chapter 5

Conclusion

In the preceding chapters, the relationship between the Event Calculus and Finite State Temporality has been examined. The two very different formalisms exemplify different positions on much-debated questions such as whether points or intervals are the basic entities in temporal semantics, and whether the precision allowed for in using the real-line is necessary for linguistic phenomena.

While Van Lambalgen and Hamm have shown how the Event Calculus can deal with these phenomena,by showing that an account of the Event Calculus can be given in finite-state terms under the circumstances found in temporal semantics, this thesis has demonstrated that an interval based discrete semantics is adequate to deal with aspectual phenomena.

Expanding EC-Models

In section 3.1 it was shown that the models given for the EC in VLH did not deal with continuous change. To give a full account of the relationship between EC-models and MSO-models, a fuller account was needed. When continuous change was added it was found that:

- The interpretation of events could no longer be given as derivative of the interpretation of fluents, and must be given independently.
- The intervals acting as interpretations of fluents could no longer be assumed to have one particular form, (a,b], and could possibly be of the form (a,b),{a}, or [a,b].

Partitioning Real-line Models

To be able to represent EC-models as finite models, and thus as string models, the fluents involved must be finite in number and each fluent must define and alternation bounded subset of the real line. It was found in section 3.2 that most EC-predicates do not affect whether or not a fluent is alternation bounded. Those that do are:

- Trajectory: Periodic functions such as sin or cos were ruled out as candidates for the 3rd argument f_2. As the EC itself rules out these functions, it can be concluded that this condition for being alternation bounded is met.
- Happens: Only a finite number of events can take place. As no linguistic scenario has an infinite number of events, this condition is also met.

From this, and the fact that scenarios are finite, it can be concluded that the EC-models in VLH obey the Finite Approximability Hypothesis, and can be represented as strings. Therefore, the full real-line is not necessary for temporal semantics (at least those areas to which the EC is applied).

Translating EC-predicates into MSO-formulas

In section 3.3 the EC-predicates were translated into MSO-formulas, and the suitability of the translation was investigated by showing that both real-line

models and string models derived from the EC-models satisfy the formulas. Nominal temporal predicates were found to assist the translation by uniquely marking timepoints of interest. Negation was found to best handled by introducing negated fluents into the alphabet. The translations were found to be:

- HoldsAt(f,t): $\exists x(P_f(x) \wedge P_{t_i}(x))$
- $\neg$HoldsAt(f,t): $\exists x(P_{\neg f}(x) \wedge P_{t_i}(x))$
- Happens(e,t): $\exists x(P_e(x) \wedge P_{t_i}(x))$
- Initiates(e,f,t): $\forall x(P_e(x) \rightarrow P_f(S(x)))$
- Terminates(e,f,t) and Releases(e,f,t): $\forall x(P_e(x) \rightarrow P_{\neg f}(S(x)))$
- Trajectory(f_1,t,f_2,d): $\lambda_d := \forall x, x'[[P_t(x) \wedge P_{t'}(x') \wedge \forall x''(x < x'' \leq x' \rightarrow P_{f_1}(x))] \rightarrow P_{f_d}(x')]$

Truthmaker Semantics

Investigation into the exact verifiers of the MSO-translations of EC-formulas in section 3.4 revealed recurring patterns. The presuppositions necessary to give a modalized state space dealt with either temporal fluents or negation. Conflicting fluents appearing in the same string position, or temporal fluents not obeying order restrictions were disallowed. The following constraints were found to be necessary:

- $\langle \trianglerighteq \rangle \boxed{0}\boxed{\ }^*$
- For all t_1, $t_2 \in$ T, where $t_1 <$ t_2 $\boxed{t_2}\boxed{\ }^*\boxed{t_1} \Longrightarrow \emptyset$
- For all t$\in$T $\boxed{t}\boxed{\ }^*\boxed{t} \Longrightarrow \emptyset$
- For all t_1, $t_2 \in$ T, where $t_1 \neq t_2$ $\boxed{t_1,t_2} \Longrightarrow \emptyset$
- For all f$\in \Sigma$ $\boxed{f,\neg f} \Longrightarrow \emptyset$

Context-freeness of Certain Temporal Constructions

Certain natural language constructions expressing relationships between events were translated as context-free languages. The superposition operation allows languages to be componentwise combined into a new language, similar to the fusion of states. It was proved in section 3.5 that the superposition of a context-free language with a regular language is context-free, not regular, and that the superposition of two context-free languages was beyond context-free, with consequences for entailments between these languages.

Aspectual Classification and Type Coercion

In chapter 4 it was shown that the eventuality structure of VLH, a structure consisting of states and events, related through EC-formulas could be mapped to FST, with the components of the eventuality corresponding to the alphabet of the string language as follows:

- f_1 in the EC corresponding to ap(f) in FST
- $f_2(x)$ in the EC corresponding to either:
  1. $\downarrow \deg[\psi]$ and $\deg_m$, or
  2. $\{\psi_{=d} | d \in D[\psi]\}$ and $\deg_m$

  depending on the required level of precision.
- e in the EC corresponding to force(ϕ), which may be the same as ap(f)
- f_3 in the EC corresponding to ϕ

Forces were shown to be essential in the description of continuous change.

With the constraints translated from the EC-formulas temporally relating the elements of the alphabet to each other, type coercion was shown to be accounted for by changes in the alphabet, causing certain constraints to apply

or no longer apply, changing the type of the string. The notions of telicity and durativity were used to assess the type of the string before and after the coercion, and the coercions were shown to be valid.

Bibliography

Bach, E. (1986), 'The algebra of events', *Linguistics and philosophy* **9**(1), 5–16.

Barwise, J. & Perry, J. (1983), 'Situations and attitudes'.

Beavers, J. (2013), 'Aspectual classes and scales of change', *Linguistics* **51**(4), 681–706.

Bennett, M. & Partee, B. H. (1978), *Toward the logic of tense and aspect in English*, Wiley Online Library.

Blackburn, P. et al. (1993), 'Nominal tense logic', *Notre Dame Journal of formal logic* **34**(1), 56–83.

Bostock, D. (1994), *Metaphysics: books Z and H*, Oxford University Press.

Davidson, D. (1967), 'The logical form of action sentences'.

Dowty, D. (1979), 'Word meaning and montague grammar: The semantics of verbs and times in generative semantics and in montague's ptq, d', *Reidel, Dordrecht* .

Fernando, T. (2004), 'A finite-state approach to events in natural language semantics', *Journal of Logic and Computation* **14**(1), 79–92.

Fernando, T. (2006), Finite-state temporal projection, *in* 'Implementation and Application of Automata', Springer, pp. 230–241.

Fernando, T. (2008), 'Branching from inertia worlds', *Journal of Semantics* **25**(3), 321–344.

Fernando, T. (2013*a*), Dowtys aspect hypothesis segmented, *in* 'Proceedings of the 19th Amsterdam Colloquium', pp. 107–114.

Fernando, T. (2013*b*), 'Partitions representing change homogeneously', *A festschrift for Jeroen Groenendijk, Martin Stokhof, and Frank Veltman* .

Fernando, T. (2015*a*), Negation and events as truthmakers. Presented at the 20th Amsterdam Colloquium.

Fernando, T. (2015*b*), 'Prior and temporal sequences for natural language', *Synthese* pp. 1–13.

Fine, K. (2015), Truthmaker semantics. Forthcoming in Blackwell Philosophy of Language Handbook.

Geis, M. L. (1970), Adverbial subordinate clauses in English., PhD thesis, Massachusetts Institute of Technology.

Hamm, F. (2014), Tense and aspect. Stanford Encyclopedia of Philosophy http://plato.stanford.edu/entries/tense-aspect/.

Hovav, M. R. & Levin, B. (2001), 'An event structure account of english resultatives', *Language* **77**(4), 766–797.

Kamp, H. & Reyle, U. (1993), *From discourse to logic: Introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory*, number 42, Springer Science & Business Media.

Kelleher, D., Fernando, T. & Vogel, C. (2015), 'Temporal forces and type coercion in strings'.

Kelleher, D., Fernando, T. & Vogel, C. (2016), Between the event calculus and finite state temporality, *in* 'International Conference on Formal Grammar', Springer, pp. 112–129.

Kelleher, D. & Vogel, C. (2013), Finite state temporality and context-free languages, *in* 'Tenth International Conference on Computational Semantics', pp. 335–339.

Kennedy, C. & Levin, B. (2008), 'Measure of change: The adjectival core of degree achievements', *Adjectives and adverbs: Syntax, semantics and discourse* pp. 156–182.

Kenny, A. (1963), 'Actions, emotions and will'.

Kowalski, R. & Sergot, M. (1989), A logic-based calculus of events, *in* 'Foundations of knowledge base management', Springer, pp. 23–55.

Kratzer, A. (1989), 'An investigation of the lumps of thought', *Linguistics and philosophy* **12**(5), 607–653.

Krifka, M. (1989), 'Nominal reference, temporal constitution and quantification in event semantics', *Semantics and contextual expression* **75**, 115.

Krifka, M. (1992), 'Thematic relations as links between nominal reference and temporal constitution', *Lexical matters* **2953**.

Krifka, M. (1998), *The origins of telicity*, Springer.

Lakoff, G. (1965), 'On the nature of syntactic irregularity.'.

Landman, F. (1992), 'The progressive', *Natural language semantics* **1**(1), 1–32.

Landman, F. & Rothstein, S. (2010), 'Incremental homogeneity in the semantics of aspectual for-phrases', *Syntax, Lexical Semantics and Event structure. Oxford. UP* .

Libkin, L. (2013), *Elements of finite model theory*, Springer Science & Business Media.

Maienborn, C. (2007), On davidsonian and kimian states, *in* I. Comorovski & K. von Heusinger, eds, 'Existence: Syntax and Semantics', Kluwer, Dordrecht, pp. 107–130.

McCawley, J. D. (1968), 'Lexical insertion in a transformational grammar without deep structure'.

Moens, M. & Steedman, M. (1988), 'Temporal ontology and temporal reference', *Computational linguistics* **14**(2), 15–28.

Montague, R. (1970), 'English as a formal language'.

Mueller, E. T. (2014), *Commonsense Reasoning: An Event Calculus Based Approach*, Morgan Kaufmann.

Pulman, S. G. (1997), 'Aspectual shift as type coercion', *Transactions of the Philological Society* **95**(2), 279–317.

Ritchie, G. D. (1979), 'Temporal clauses in english', *Theoretical Linguistics* **6**(1-3), 87–116.

Ross, J. R. (1972), Act, *in* 'Semantics of natural language', Springer, pp. 70–126.

Rothstein, S. (2008), *Structuring events: A study in the semantics of aspect*, Vol. 5, John Wiley & Sons.

Ryle, G. (2009), *The concept of mind*, Routledge.

Thomas, W. (1997), *Languages, automata, and logic*, Springer.

Van Lambalgen, M. & Hamm, F. (2005), *The proper treatment of events*, Vol. 4, Blackwell Publishing.

Vendler, Z. (1957), 'Verbs and times', *The philosophical review* pp. 143–160.

Von Wright, G. H. (1963), 'Norm and action'.

Von Wright, G. H. (1968), 'An essay in deontic logic and the general theory of action'.

Wilson, D. (1975), 'Presupposition, assertion, and lexical items', *Linguistic Inquiry* pp. 95–114.