Representing Temporal Operators with Dependent Event Types

Felix Bradley and Zhaohui Luo

Department of Computer Science
Royal Holloway, University of London

12 June 2024

Presentation Overview

ROYAL HOLLOWAY UNIVERSITY OF LONDON

1. Event Semantics
2. Representing Temporal Operators
3. Extending Dependent Event Types

Introduction

ROYAL HOLLOWAY UNIVERSITY OF LONDON

Motivation: temporal operators?

It's unclear how to implement temporal operators for modern type theories

Perhaps event semantics offer a way forward

Event Semantics

Event types build on Davidson's event semantics, which were proposed for describing both actions and adverbial modifications

Prior work on formalising event semantics in type theory uses Montague grammar

Dependent event types as introduced by Luo and Soloviev are relatively recent

Montague semantics and Davidson's work is based on simple type theory

These formal semantics have been further studied and developed in modern type theories such as MLTT

Event Semantics

"Peter slowly, carefully slices the bread."

In the traditional Montague semantics:

$$\mathit{slowly}(\mathit{carefully}(\mathit{slice}(\text{Peter})))(\text{Bread})$$

With event types:

$$\exists e : \text{Event}\,.\mathit{agent}(e) = \text{Peter} \wedge \mathit{patient}(e) = \text{Bread} \wedge \mathit{slice}(e) \wedge \mathit{slowly}(e) \wedge \mathit{carefully}(e)$$

With dependent event types:

$$\exists e : \text{Event}_{AP}(\text{Peter}, \text{Bread}).\mathit{slice}(e) \wedge \mathit{slowly}(e) \wedge \mathit{carefully}(e)$$

Event Semantics

Dependent event types prevent incorrect semantics that are possible with simple event types[1]

"Nobody talked."

Correct: $\neg\exists h : \mathsf{Agent}.(\mathit{human}(x) \wedge \exists e : \mathsf{Event}_A(x).\mathit{talks}(e))$

Incorrect: $\neg\exists e : \mathsf{Event}_A(x).(\exists h : \mathsf{Agent}.\mathit{human}(x) \wedge \mathit{talks}(e))$

[1] Example due to [CL20]

DETs for Simple Type Theory

We extend Church's simple type theory with the following types:

$$\frac{\Gamma \text{ valid}}{\Gamma \vdash \text{Event type}} \qquad \frac{\Gamma \text{ valid}}{\Gamma \vdash \text{Agent type}} \qquad \frac{\Gamma \text{ valid}}{\Gamma \vdash \text{Patient type}} \qquad \frac{\Gamma \text{ valid}}{\Gamma \vdash \text{Time type}}$$

And these subtypes:

$$\frac{\Gamma \vdash a : \text{Agent}}{\Gamma \vdash \text{Evt}_A(a) \text{ type}} \qquad \frac{\Gamma \vdash p : \text{Patient}}{\Gamma \vdash \text{Evt}_P(p) \text{ type}} \qquad \frac{\Gamma \vdash t : \text{Time}}{\Gamma \vdash \text{Evt}_T(t) \text{ type}}$$

And these *other* subtypes:

$$\frac{\Gamma \vdash a : \text{Agent} \quad \Gamma \vdash p : \text{Patient}}{\Gamma \vdash \text{Evt}_{AP}(a, p) \text{ type}} \qquad \frac{\Gamma \vdash a : \text{Agent} \quad \Gamma \vdash t : \text{Time}}{\Gamma \vdash \text{Evt}_{AT}(a, t) \text{ type}} \ \ldots$$

$$\frac{\Gamma \vdash p : \text{Patient} \quad \Gamma \vdash t : \text{Time}}{\Gamma \vdash \text{Evt}_{PT}(p, t) \text{ type}} \qquad \frac{\Gamma \vdash a : \text{Agent} \quad \Gamma \vdash p : \text{Patient} \quad \Gamma \vdash t : \text{Time}}{\Gamma \vdash \text{Evt}_{APT}(a, p, t) \text{ type}} \ \ldots$$

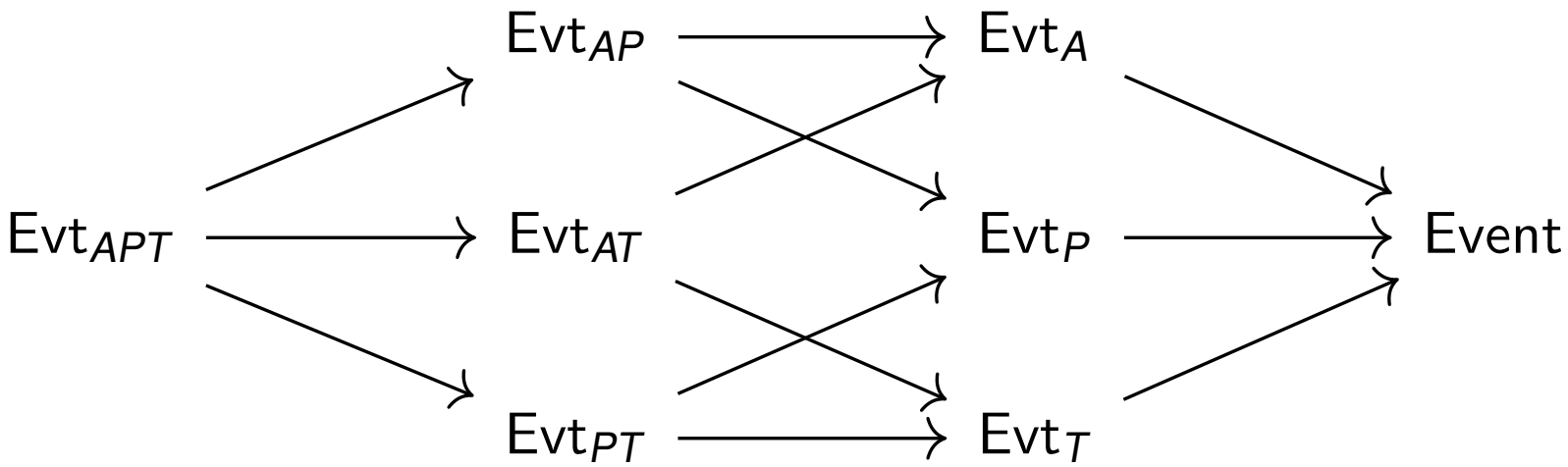
And the following subsumptive subtyping rules:

$$\frac{\Gamma \vdash a : \mathsf{Agent}}{\Gamma \vdash \mathsf{Evt}_A(a) \leq \mathsf{Event}} \qquad \frac{\Gamma \vdash p : \mathsf{Patient}}{\Gamma \vdash \mathsf{Evt}_P(p) \leq \mathsf{Event}} \qquad \frac{\Gamma \vdash t : \mathsf{Time}}{\Gamma \vdash \mathsf{Evt}_T(t) \leq \mathsf{Event}}$$

$$\frac{\Gamma \vdash a : \mathsf{Agent} \quad \Gamma \vdash p : \mathsf{Patient}}{\Gamma \vdash \mathsf{Evt}_{AP}(a,p) \leq \mathsf{Evt}_A(a)} \qquad \frac{\Gamma \vdash a : \mathsf{Agent} \quad \Gamma \vdash p : \mathsf{Patient}}{\Gamma \vdash \mathsf{Evt}_{AP}(a,p) \leq \mathsf{Evt}_P(p)}$$

DETs for Simple Type Theory

Subtyping rules become more complicated as number of thematic relations increases



Representing Temporal Operators

We can define the typical $\Box$ and $\Diamond$ operators as follows:

$$\Diamond A(ref) \stackrel{\text{def}}{=} \exists(t : \text{Time}).(ref \preceq t \wedge A(t))$$

$$\Box A(ref) \stackrel{\text{def}}{=} \forall(t : \text{Time}).(ref \preceq t \rightarrow A(t))$$

We can encode and represent simple sentences such as “John will talk” as

$$(\Diamond A)(now) \text{ where } A(t) \stackrel{\text{def}}{=} \exists(e : \text{Event}_{AT}(\text{John}, t).\text{talk}(e)$$

Representing Temporal Operators

ref is a reference time stamp used to represent the *speaking time* of a given sentence

e.g. “There will be flying cars in 30 years’ time.”

This sentence has different meanings when spoken in / read and understood in e.g. 1985 versus 2015

$\Box$ and $\Diamond$ have type $(\mathsf{Time} \to \mathbf{t}) \to (\mathsf{Time} \to \mathbf{t})$, meaning we can nest them

e.g. “There will always be another solar eclipse.”

$\Box\Diamond(A')(now)$ where $A'(t) \stackrel{\text{def}}{=} \exists(e : \mathsf{Event}_{PT}(\mathsf{Earth}, t).\mathsf{eclipse}(e)$

Generalising Dependent Event Types

ROYAL HOLLOWAY UNIVERSITY OF LONDON

Luo and Soloviev studied the extension of Church's simple type theory and UTT[2] with dependent event types [LS17, LS20, CL20].

We can generalise the results of Luo and Soloviev to any set of thematic relations, and to other modern type theories such as Martin-Löf type theory.

Church's simple type theory used subsumptive subtyping. For MTTs such as MLTT and UTT, we need coercive subtyping.

Coercive subtyping is well-studied and proven to be a conservative extension of theories written in LF[3] [Xue13, LSX13].

[2] Luo's Unified Theory of dependent Types

[3] The typed version of Martin-Löf's logical framework [Luo94, NPMS90].

Generalising Dependent Event Types

Fix a type theory T written in LF. Let $\mathcal{R}$ be an arbitrary set of thematic relations. Then consider the extension $T_E[\mathcal{R}]$ of T by the following:

First, the following constant types and constant type families:

- $\mathsf{Entity} : \mathsf{Type}$
- $\mathsf{label}(r) : \mathsf{Type}$ for each $r \in \mathcal{R}$, where label is a meta-constant
- $\mathsf{Evt}_{r_1,\ldots,r_n} : (\mathsf{label}(r_1))(\ldots)(\mathsf{label}(r_n))\mathsf{Type}$ for every $n \in 1, \ldots, |\mathcal{R}|$ and for each subset $\{r_1, \ldots, r_n\} \subseteq \mathcal{R}$ up to reordering.

Generalising Dependent Event Types

Next, the constant functions

$$c[r_1, \ldots, r_n][r_1, \ldots, r_{i-1}, r_{i+1}, \ldots, r_n] : (\mathsf{Evt}_{r_1,\ldots,r_n})\mathsf{Evt}_{r_1,\ldots,r_{i-1},r_{i+1},\ldots,r_n},$$

the coercive subtyping relations

$$\mathsf{Evt}_{r_1,\ldots,r_n} \leq_{c[r_1,\ldots,r_n][r_1,\ldots,r_{i-1},r_{i+1},\ldots,r_n]} \mathsf{Evt}_{r_1,\ldots,r_{i-1},r_{i+1},\ldots,r_n}$$

and the coherence conditions

$$\begin{aligned} c[r_1, \ldots, r_{j-1}, r_{j+1}, \ldots, r_n][r_1, \ldots, r_{i-1}, r_{i+1}, \ldots, r_{j-1}, r_{j+1}, \ldots, r_n] & \\ \circ c[r_1, \ldots, r_n][r_1, \ldots, r_{j-1}, r_{j+1}, \ldots, r_n] & \\ = c[r_1, \ldots, r_{i-1}, r_{i+1}, \ldots, r_n][r_1, \ldots, r_{i-1}, r_{i+1}, \ldots, r_{j-1}, r_{j+1}, \ldots, r_n] & \\ \circ c[r_1, \ldots, r_n][r_1, \ldots, r_{i-1}, r_{i+1}, \ldots, r_n] & \end{aligned}$$

for every $n \in 1, \ldots, |\mathcal{R}|$, for each $i, j \in 1, \ldots, n$ such that $i < j$, and for each subset $\{r_1, \ldots, r_n\} \subseteq \mathcal{R}$ up to reordering.

Generalising Dependent Event Types

ROYAL HOLLOWAY UNIVERSITY OF LONDON

Fix a type theory T written in LF. Let $\mathcal{R}$ be an arbitrary set of thematic relations. Consider the extension $T_E[\mathcal{R}]$ of T.

Theorem

$T_E[\mathcal{R}]$ is a conservative extension of T.

Corollary

If T is logically-consistent/strongly-normalising/etc. then $T_E[\mathcal{R}]$ is likewise.

Final Remarks

MLTT's types-as-propositions logic causes problems for natural language semantics

Proof irrelevance would be great, but proof irrelevance in MLTT would collapse all types to singletons

MLTT with h-logic is suitable for semantics due to its implementation of mere propositions and propositional truncation [Luo19]

Coercive subtyping is already well-studied for theories expressed in LF

Future work is to examine the expressibility of $\mathrm{MLTT}_{\mathrm{h}}$ in LF / coercive subtyping for $\mathrm{MLTT}_{\mathrm{h}}$

References I

Stergios Chatzikyriakidis and Zhaohui Luo.
Formal Semantics in Modern Type Theories.
ISTE, 2020.

Donald Davidson.
The Logical Form of Action Sentences.
In *Essays on Actions and Events*. Oxford University Press, 09 2001.

Zhaohui Luo and Sergei Soloviev.
Dependent event types.
In *Proc of the 24th Workshop on Logic, Language, Information and Computation (WoLLIC'17)*, LNCS, pages 216–228, London, 2017.

Zhaohui Luo and Sergei Soloviev.
Dependent event types.
Manuscript, 2020.

Zhaohui Luo, Sergei Soloviev, and Tao Xue.
Coercive subtyping: Theory and implementation.
Information and Computation, 223:18–42, 2013.

Zhaohui Luo.
Computation and Reasoning: A Type Theory for Computer Science.
Oxford University Press, London, 1994.

Zhaohui Luo.
Proof-irrelevance in type theoretic semantics.
In *Logic and Algorithms in Computational Linguistics 2018*. Springer, 2019.

References II

Bengt Nordström, Kent Petersson, and Jan M. Smith.

Programming in Martin-Löf's Type Theory: An Introduction.
International Series of Monographs in Computer Science. Oxford University Press, London, 1990.

Tao Xue.

Theory and Implementation of Coercive Subtyping.
PhD thesis, Royal Holloway, University of London, 2013.