Dependent Event Types

Zhaohui Luo[1] and Sergei Soloviev[2]

[1] Univ of London, U.K.
zhaohui.luo@hotmail.co.uk
[2] IRIT, Toulouse, France
Sergei.Soloviev@irit.fr

Employing dependent types for a refined treatment of event types provides a nice improvement to Davidson's event semantics [3, 7]. We consider dependent event types indexed by thematic roles and show that subtyping between them plays an essential role in semantic interpretations. It is also shown that dependent event types give a natural solution to the event quantification problem in combining event semantics with the Montague semantics [1, 8, 4].

For instance, $Evt_A(a)$ is the dependent type of events whose agents are $a : Agent$. The dependent event types abide by subtyping relationships:

$$Evt_{AP}(a,p) \leq Evt_A(a) \leq Event \;\text{ and }\; Evt_{AP}(a,p) \leq Evt_P(p) \leq Event,$$

where $a : Agent$, $p : Patient$ and $Event$ is the type of all events. With such dependent event types, subtyping is crucial. Consider `John talked loudly`: its Davidsonian event semantics would be $\exists e : Event.\ talk(e)\ \&\ loud(e)\ \&\ agent(e,j)$, where $talk,\ loud : Event \rightarrow \mathtt{t}$. With dependent event types, the semantics would be $\exists e : Evt_A(j).\ talk(e)\ \&\ loud(e)$, in which the terms such as $talk(e)$ are only well-typed because $Evt_A(j) \leq Event$.

It has been argued that there is some incompatibility between (neo-)Davidsonian event semantics and the traditional compositional semantics, as the event quantification problem shows: the following two possible interpretations of `No dog barks` are both well-formed formulas, although (2) is incorrect:

(1) $\neg\exists x : \mathbf{e}.\ dog(x)\ \&\ \exists e{:}Event.\ bark(e)\ \&\ agent(e,x)$

(2) (#) $\exists e : Event.\ \neg\exists x : \mathbf{e}.\ dog(x)\ \&\ bark(e)\ \&\ agent(e,x)$

To exclude such incorrect interpretations, various informal solutions have been proposed [1, 8]. With dependent event types, this problem is solved naturally and formally – the incorrect semantic interpretations such as (4) below are excluded because they are ill-typed, while the correct one (3) is well-typed.

(3) $\neg\exists x : \mathbf{e}.\ (dog(x)\ \&\ \exists e : Evt_A(x).\ bark(e))$

(4) (#) $\exists e : Evt_A(x).\ \neg\exists x : \mathbf{e}.\ dog(x)\ \&\ bark(e)$

The underlying formal system C_e is the extension of Church's simple type theory [2], as used in the Montague semantics, with dependent event types and the subtyping relations. C_e can be faithfully embedded into UTT[C], i.e., the type theory UTT [5] extended with coercive subtyping in C [6], where C contains the subtyping judgements that correspond to the above subtyping relations between dependent event types. Since UTT[C] has nice meta-theoretic properties such as normalisation and logical consistency, so does C_e.

The paper is available online at http://www.cs.rhul.ac.uk/home/zhaohui/DET.pdf.

References

1. Champollion, L.: The interaction of compositional semantics and event semantics. Linguistics and Philosophy 38, 31–66 (2015)
2. Church, A.: A formulation of the simple theory of types. J. Symbolic Logic 5(1) (1940)
3. Davidson, D.: The logical form of action sentences. In: S. Rothstein (ed.). The Logic of Decision and Action. University of Pittsburgh Press (1967)
4. de Groote, P., Winter, Y.: A type-logical account of quantification in event semantics. Logic and Engineering of Natural Language Semantics 11 (2014)
5. Luo, Z.: Computation and Reasoning: A Type Theory for Computer Science. Oxford University Press (1994)
6. Luo, Z., Soloviev, S., Xue, T.: Coercive subtyping: theory and implementation. Information and Computation 223, 18–42 (2012)
7. Parsons, T.: Events in the Semantics of English. MIT Press (1990)
8. Winter, Y., Zwarts, J.: Event semantics and abstract categorial grammar. Proc. of Mathematics of Language 12, LNCS 6878 (2011)