Telicity is launching and atelicity is entrainment

Toronto Dog Days 3, 7 August 2014

Bridget Copley
bridget.copley@sfl.cnrs.fr
SFL (CNRS/Paris 8)
This talk reflects joint work with Heidi Harley, University of Arizona (Copley & Harley 2014, in prep).
The idea
Launching and entrainment

Causes can have either of two temporal relationships to their effects.

(1) a. launching

\[ \text{Diagram of launching relationship} \]

b. entrainment

\[ \text{Diagram of entrainment relationship} \]

Launching and entrainment

However, event semantics at the interface with syntax has not paid much explicit attention to entraining causation. There seems to be rather a focus on launching causation (and stative results).

(2) Pustejovsky 1995
Mary ran to the store: \([\text{cause}(\text{act}(m), \text{become}(\text{at}(m, \text{the-store}) \text{ BY run})]\)

(3) Higginbotham 2000
John saddled the horse: \(\exists e \exists e' \ [\text{agent}(John,e) \ \text{saddled}(\text{the horse},e') \ (e,e') \text{ is a telic pair}]\)

(4) Ramchand 2008
Mary opened the door: \(\text{Subject}(\text{Mary}, e_1) \&\)

a. \([v] = \lambda P \lambda x \lambda e \exists e_1, e_2 \ [P(e_2) \& v(e_1) \& \text{State}(e_1) \& e = e_1 \rightarrow e_2 \& \text{Subject} \ (x, e_1)]\)

b. \([V] = \lambda P \lambda x \lambda e \exists e_1, e_2 \ [P(e_2) \& V(e_1) \& \text{Process}(e_1) \& e = (e_1 \rightarrow e_2) \& \text{Subject} \ (x, e_1)]\)
Telicity and atelicity

▶ Krifka 1998 e.g.: Certain predicates involve a mapping from parts of the event to parts of object; quantized object ⇒ telicity.

▶ Kennedy & Levin 2008, e.g.: Certain predicates involve a mapping from parts of the event to degrees on a scale; endpoint ⇒ telicity.

▶ Filip & Rothstein 2005, Filip 2008, Beavers 2012: Both these mechanisms for telicity are needed.

Proposal: The key intuitions above are retained, but modeling entrainment as well as launching will allow us to simplify the logical forms. Telicity involves launching causation, while atelicity involves entraining causation.
Proposal (Copley & Harley to appear, in prep)

- Both telic and atelic predicates specify a result situation as well as a causing situation.
- The result situation need not be a state, though it can be.
- Given a causing and a result situation, world/lexical knowledge tells us whether the result situation begins as the causing situation ends (launching), or begins as the causing situation begins, as a “cotemporal result” (entrainment).

(5)

a. Mary ate apples for/*in five minutes.
b. Mary ate an apple *for/in five minutes.

(6)

a. Mary pushed the cart for/(*)in five minutes.
b. Mary pushed the cart to the fence (*)for/in five minutes.

(7) Mary heated the soup for/in five minutes.
Proposal (the new bit)

The goal now: to use both kinds of causation, along with quantization and scales, to represent telicity and atelicity, in a formal semantics.

We’ll do this by using the notion of force instead of event (Copley & Harley to appear, in prep), so first we’ll look at two reasons why we need forces to represent causation in verbal predicates.
A force-theoretic framework
Why forces?

Firstly, event arguments work well for launching and for some cases of entrainment . . .

(8)  
   a. Mary ate an apple.  
   b. Mary heated the soup.

. . . but not for all cases of entrainment, because event individuation fails.

(9)  
   a. Mary ate apples.  
   b. Mary danced.

What we really need is the notion of an input of energy or force.
Why forces?

Secondly, without forces it’s hard to distinguish between, e.g. staying eventualities and being eventualities (Copley & Harley, to appear):

(10) a. John was in the room.
    b. John stayed in the room.

(11) a. #John is being in the room.
    b. John is staying in the room.

(12) a. stay: \(e_1 \text{ Cause } e_2 \text{ & } \text{Be}(e_2, p)\) ?
    b. stay: \(e_1 \text{ Cause } e_2 \text{ & } \text{Become}(e_2, p)\) ?

Instead: \textit{stay} refers to a force with a cotemporal stative result.
The framework

Copley & Harley to appear, in prep

- Essential type difference is of dynamicity: forces and situations
- A force is represented in language by a function from an initial situation to a *ceteris paribus* final situation
- Dual ontology: conceptual ontology \(\neq\) linguistic ontology

<table>
<thead>
<tr>
<th>conceptual</th>
<th>linguistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>inputs of energy (\varphi)</td>
<td>force functions (f)</td>
</tr>
<tr>
<td>situations (\sigma)</td>
<td>situations (s)</td>
</tr>
</tbody>
</table>

Force functions are type \(\langle s, s \rangle\) (or type \(f\) for short).

The evaluation function is used, as usual, to mediate between the linguistic and conceptual level. So one may write \([f] = \varphi\), or \([s] = \sigma\), e.g.
The idea

A force-theoretic framework

The implementation

Conclusion

Why forces?

The framework

\[ \phi \Leftrightarrow f \]

arises from (all) the individuals and property attributions in

\[ \sigma \Leftrightarrow S \]
We define two linked sequences, one of situations and one of force functions:

(13) \[ \ldots, s_{-2}, s_{-1}, s_0, s_1, s_2, \ldots \]
     \[ \ldots, f_{-2}, f_{-1}, f_0, f_1, f_2, \ldots \]

(14) a. Let \( f = \text{net}(s) \) iff \([f]\) is the net force of \([s]\)
     b. \( f_n = \text{net}(s_n) \)
     c. \( s_{n+1} = f(s_n) \)

(15) a. \( \text{init}(f_n) = s_n \)
     b. \( \text{fin}(f_n) = s_{n+1} \)
Stative predicates:

- predicates of situations, type: \(\langle s,t \rangle\)
- example: [be in the room](s), [know French](s)

Dynamic predicates:

- predicates of forces, type \(\langle f,t \rangle\), aka \(\langle \langle s,s \rangle,t \rangle\)
- examples: [eat](f), [stay](f)

Voice, when present, introduces the agent/causer: source(f).
The implementation
Adding scales and quantization

(16) a. Dowty’s (1979) BECOME: \( \neg \text{p}(t_1) \& \text{p}(t_2) \)
b. Copley & Harley’s (to appear) \( v_{become} \):
\( \neg \text{p}(\text{init}(f)) \& \text{p}(\text{fin}(f)) \)
c. New \( v_{change} \): \( \text{p}(x)(\text{init}(f)) \text{ R } \text{p}(x)(\text{fin}(f)) \) 
  (cf. Koenig & Chief 2008)

R can either be specified lexically, by context, or (perhaps) in flavors of little \( v \) that select verbs. When we define R, we’ll call \( \text{p}(x)(\text{init}(f)) \) “\( d_0 \)” and \( \text{p}(x)(\text{fin}(f)) \) “\( d_1 \)”.

(17) a. scalar adjectives behave as usual
b. New predicate “EXT” = extent (and see also Kardos 2012)
  quantized: existence is categorical: object either exists (\( d=1 \)) or it doesn’t (\( d=0 \))
  non-quantized: no endpoints but possible to compare degrees, i.e. amounts

c. When current degree equals degree specified in predicate for \( s_1 (= \text{fin}(f)) \), \( s_1 \) begins
(18) Mary ate an apple.
   a. eat: predicate is EXT, $d_0 > d_1$
   b. quantized object: $d=0$ or $d=1$
   c. $\lambda f. \text{EXT([an apple]) (init}(f)) \ R \ \text{EXT([an apple])(fin}(f))$
      & [eat](f) & source(Mary, f)
   d. $\text{fin}(f)$ begins when current $d = 1$
      $\Rightarrow$ launching causation $\Rightarrow$ telic

(19) Mary heated the soup (sufficiently).
   a. there is a contextual endpoint of scale $d_{max}$
   b. heat: predicate is hot, $d_0 < d_1$
   c. $\lambda f. \text{hot}(x) (init}(f)) \ R \ \text{hot}(x)(fin}(f)) & source(Mary, f)$
   d. $\text{fin}(f)$ begins when current $d = d_{max}$
      $\Rightarrow$ launching causation $\Rightarrow$ telic
Atelicity

(20) Mary ate apples.
   a. eat: predicate is EXT, \( d_0 > d_1 \)
   b. non-quantized object: \( d \) can have any value
   c. \( \lambda f . \ ext([\text{apples}]) \ (init(f)) \ R \ ext([\text{apples}]) \ (fin(f)) \ & \ ext([\text{eat}]) \ (fin(f)) \ & \ source(Mary, f) \)
   d. \( fin(f) \) begins when current \( d < d_0 \)
      \( \Rightarrow \) entrainment \( \Rightarrow \) atelic

(21) Mary heated the soup (some).
   a. there is no endpoint of scale \( d_{max} \)
   b. heat: predicate is hot, \( d_0 < d_1 \)
   c. \( \lambda f . \ hot(x) \ (init(f)) \ R \ hot(x) \ (fin(f)) \)
      \& source(Mary, f)
   d. \( fin(f) \) begins when current \( d < d_0 \)
      \( \Rightarrow \) entrainment \( \Rightarrow \) atelic
Atelicity

(22) Mary danced.
   a. dance = create dance, predicate is EXT
   b. non-quantized object: d can have any value
   c. \( \lambda f . \text{EXT}([\text{dance}]) \ (\text{init}(f)) \ R \ \text{EXT} \ ([\text{dance}]) \ (\text{fin}(f)) \ & \ \text{source}(\text{Mary}, f) \)
   d. \text{fin}(f) \ begins \ when \ current \ d < d_0 \\
      \Rightarrow \ \text{entrainment} \ \Rightarrow \ \text{atelic}

(23) Mary stayed.
   a. contextually-provided LOC predicate
   b. stay: \( d_1 = d_0 \)
   c. \( \lambda f . \text{LOC}(\text{Mary}) \ (\text{init}(f)) \ R \ \text{LOC}(\text{Mary}) \ (\text{fin}(f)) \ & \ \text{source}(\text{Mary}, f) \)
   d. \text{fin}(f) \ begins \ when \ current \ d = d_0 \\
      \Rightarrow \ \text{entrainment} \ \Rightarrow \ \text{atelic}
Conclusion

- There are two kinds of causation: launching, which corresponds to telicity, and entrainment, which corresponds to atelicity.
- By incorporating scale degrees and quantization (the latter in part by using an extent predicate EXT) into Copley & Harley’s force-theoretic framework, we can effectively model telicity and atelicity.