3-1-2015

Back To Events: More on the Logic of Verbal Modification

Lucas Champollion
champollion@nyu.edu

Back To Events: More on the Logic of Verbal Modification

Back to Events: More on the Logic of Verbal Modification

Lucas Champollion*

1 Introduction

Beaver and Condoravdi (2007) propose to move away from event semantics (Davidson 1967) as a theory of verbal modification. The system they present, *linking semantics*, provides a clean and compositional account of the interaction of events and quantifiers. Beaver and Condoravdi present it as surpassing both (Neo-)Montagovian semantics and Davidsonian event semantics. But as we will see, their system embodies a nonstandard view of the semantics of action sentences that does not use events, and this means that some of the benefits of event semantics are lost.

This paper shows that we can have our cake and eat it too: we can reconcile linking semantics with Davidsonian event semantics and keep the strengths of both systems. This is part of a larger project about integrating Montague semantics with event semantics (Champollion 2010, 2014).

The central idea in linking semantics is that verbs and verbal projections denote sets of "role assignments", that is, partial functions that map labels like ARG1, ARG2 and T (intuitively, agent, theme, and time) to appropriate values. For example, in a model where John kicked Bill at 1pm, the sets denoted by *kick*, by *kick Bill* and by *John kick Bill* each contain at least the role assignment $g_1 = [\text{ARG1}, j; \text{ARG2}, b; \text{T}, 1pm]$.

Linking semantics provides accounts of the interaction of events and quantifiers, of verbal modifiication, tense, temporal anaphora, and stacked temporal modifiers. Beaver and Condoravdi (2007) give the derivation in (1) (slightly simplified here) as an illustration of how the system deals with quantification. Here, g, g' ranges over role assignments, and L over sets of role assignments. The notation $g + [\text{ARG1}, x]$ can be read as an instruction to extend g by a new entry that maps ARG1 to x. It is only defined when g does not already contain an entry for ARG1. Following Beaver and Condoravdi (2007), I will write $(g + [\text{ARG1}, y]) + [\text{ARG2}, x]$ as $g + [\text{ARG1}, y; \text{ARG2}, x]$ and I will occasionally rely on obvious equivalences, for example by writing $g + [\text{ARG2}, x](\text{T})$ as $g(\text{T})$.

(1)　　A diplomat visited every country.

　　a.　$[\![every\ country{:}\text{ARG2}]\!] = \lambda L \lambda g \forall x.\textbf{country}(x) \to L(g + [\text{ARG2}, x])$
　　b.　$[\![a\ diplomat{:}\text{ARG1}]\!] = \lambda L \lambda g \exists y.\textbf{diplomat}(y) \land L(g + [\text{ARG1}, y])$
　　c.　$[\![visit]\!] = \lambda g.\textbf{visit}(g)$
　　d.　$[\![\text{-}ed]\!] = \lambda L \lambda g.L(g) \land g(\text{T}) < \textbf{now}$
　　e.　$[\![visited\ every\ country{:}\text{ARG2}]\!]$
　　　　$= \lambda g \forall x.\textbf{country}(x) \to [\textbf{visit}(g + [\text{ARG2}, x]) \land g(\text{T}) < \textbf{now}]$
　　f.　$[\![a\ diplomat{:}\text{ARG1}\ visited\ every\ country{:}\text{ARG2}]\!]$
　　　　$= \lambda g \exists y.\textbf{diplomat}(y) \land \forall x.\textbf{country}(x) \to [\textbf{visit}(g + [\text{ARG1}, y; \text{ARG2}, x]) \land g(\text{T}) < \textbf{now}]$
　　　　$= \lambda g.g < \textbf{now} \land \exists y.\textbf{diplomat}(y) \land \forall x.\textbf{country}(x) \to \textbf{visit}(g + [\text{ARG1}, y; \text{ARG2}, x])$

Linking semantics defines truth with respect to a model M by existentially quantifying over times. (The interpretation function is of course also defined with respect to a model, as well as temporal and world parameters, which I have left out here for convenience.) The resulting rule, again slightly simplified compared with the original formulation because I left out worlds, is as follows:

(2)　　$M \models S$ iff $\exists t [\![S]\!]([\text{T}, t])$

When applied to the last line of the derivation in (1), this yields:

(3)　　$M \models [\![A\ diplomat\ visited\ every\ country]\!]$ iff
　　　　$\exists t.t < \textbf{now}. \exists y.\textbf{diplomat}(y) \land \forall x.\textbf{country}(x) \to \textbf{visit}([\text{ARG1}, y; \text{ARG2}, x; \text{T}, t])$

The relative scope of the two quantifiers supplied by the two noun phrases is correct. As this derivation shows, linking semantics does not use quantifying-in or quantifier raising, which Beaver and

Condoravdi see as problematic. I will discuss this reading in more detail below.

I will start by pointing out that linking semantics is not really a "logic" of verbal modification because it treats the relevant phenomenon by a special-purpose axiom (Section 2). Another axiom concerns the treatment of time and brings its own problems (Section 3). To fix these problems, I will formulate a version of event semantics that works in a very similar way to linking semantics (Section 4), though the treatment of temporal anaphora makes it necessary to add a dynamic layer (Section 5). I use a simple setup but there are ways to extend it to more complex phenomena like intersentential temporal anaphora and quantificational subordination (Section 6). Some interesting differences between temporal and ordinary anaphora emerge (Section 7).

2 The Treatment of Adverbial Modification

In event semantics, verbal modifiers like *at noon* and *in the bathroom* are interpreted conjunctively, so that entailments like (4) are modeled as logical entailments (5) (Carlson 1984, Parsons 1990).

(4)   Jones buttered the toast at noon. $\Rightarrow$ Jones buttered the toast.

(5) $[\exists e.\mathbf{butter}(e) \wedge \mathbf{ag}(e) = j \wedge \mathbf{th}(e) = t \wedge \tau(e) = \mathbf{noon}] \Rightarrow [\exists e.\mathbf{butter}(e) \wedge \mathbf{ag}(e) = j \wedge \mathbf{th}(e) = t]$

This treatment of verbal modification is considered a very powerful argument in favor of event semantics (Landman 2000:ch. 1). One would expect the natural-language entailment in (4) to correspond to a logical entailment, and this is the case in event semantics. But in linking semantics, the entailment in (6) is not underwritten by predicate logic, so it no longer comes for free.

(6) $\mathbf{butter}([\text{ARG}1, j; \ \text{ARG}2, toast; \ \text{T}, noon]) \Rightarrow \mathbf{butter}([\text{ARG}1, j; \ \text{ARG}2, toast])$

The reason that this entailment is not accounted for by predicate logic alone is that there is no guarantee that a predicate like **butter** should hold of a certain role assignment just because it holds of another one. In particular, the predicate might hold of the role assignment (7a) without holding of the assignment (7b), contrary to intuition.

(7)   a. $[\text{ARG}1, j; \ \text{ARG}2, toast; \ \text{T}, noon]$
         b. $[\text{ARG}1, j; \ \text{ARG}2, toast]$

Now, in such a situation it is always possible to add axioms that impose constraints on possible models in order to generate entailments when the underlying logic by itself does not generate them. Beaver and Condoravdi apply this method as follows. The goal is for the entailment in (4) to be reflected in the properties of the predicate **butter**. When understood as a description of events, the role assignment (7b) is more general than the role assignment (7a), since both assignments agree on their first and second arguments but only the former specifies a time. Generalizing from this case, one expects that whenever **butter** holds of a given role assignment, it also holds of any more general role assignment. And of course this should not only be the case for **butter** but for any verb. Linking semantics guarantees this via the following principle:

(8)   **Argument reduction axiom.** For any verb V and model M, for any role assignments g and g', if $g \in [\![V]\!]_M$, $g' \subset g$, and every argument of V is in the domain of g, then $g' \in [\![V]\!]_M$.

This axiom says that if V holds of a role assignment R, it also holds of any role assignment that is more specific than R, as long as all of the arguments of the verb are kept. This models entailments like the one in (4). However, at this point the question arises whether we have lost something in the translation. Linking semantics is presented by its authors as a logic (!) of verbal modification. But as explained above, one major motivation for event semantics as a logic of verbal modification is to explain entailments like the one in (4) by reducing them to the fact that it is a logical necessity that $p \wedge q$ entails p. When modeling natural language inferences, this basic property of predicate logic is also taken to explain facts that are unrelated to verbal modification, such as the entailment from natural-language conjunctions like *John is happy and Mary is sad* to each of their conjuncts. This motivation for event semantics does not carry over to linking semantics, since the argument

reduction axiom is specific to verbal modification and is not linked to any such general principles.

3 The Treatment of Time

Linking semantics relies on another axiom, this one concerning time. As we have seen above, the surface scope reading of a sentence like (9a) is represented as in (9b).

(9)   a.   A diplomat visited every country.
       b. $\exists t.t < \textbf{now} \wedge \exists y.\textbf{diplomat}(y) \wedge \forall x.\textbf{country}(x) \rightarrow \textbf{visit}([\text{ARG1}, y; \text{ARG2}, x; \text{T}, t])$
             Preliminary paraphrase: "There exist a past time interval t and a diplomat y such that
             y visited every country at t."

Since the role assignments over which the formula quantifies all map T to the same value, namely the time t that is existentially bound by the highest-scope quantifier, this formula by itself requires all the visits to happen simultaneously at that time t. This is too strong, since the sentence is compatible with a (plausible) scenario in which the diplomat in question visits each country in sequence. Beaver and Condoravdi address this problem by introducing a principle they call "temporal closure". This principle makes all verbs temporally upward closed, in the sense that, for example, if x visits y at time t then x also visits y at every superinterval of t. This principle is implemented as follows:

(10)   **Temporal closure axiom.** For any verb V and model M, and for any role assignments g
        and g' that differ only with respect to the value they assign to T, if $g \in \llbracket V \rrbracket_M$ and $g(\text{T})$ is a
        subinterval of $g'(\text{T})$, then $g' \in \llbracket V \rrbracket_M$.

The effect of adding this axiom to the system is the following. In order for a verbal predicate to be true, it no longer has to be true *at* whatever time t it is formally related to, it is merely required to be true at some interval or other *within* that time. Given this axiom, the surface-scope reading of sentence (9a) no longer requires the diplomat to visit all countries simultaneously. Rather, the truth conditions in (9b) can now be paraphrased as "There exist a past time interval t and a diplomat y such that y visited every country at some point or other within t." As long as all the visits took place in the past, there will always be a way to make the time interval t long enough to include all of them. So the sentence comes out as having the right truth conditions after all.

Unfortunately, this approach to temporal semantics does not always work as intended. The temporal closure axiom gives the mapping of T to a given time the effect of a "within" semantics rather than an "at" semantics. In effect, the axiom causes this mapping to behave like an upper bound on the time a given activity takes to unfold, as opposed to being an exact measure. In the previous sentence, this was appropriate, but for some predicates an exact measure is required. One kind of example involves statements like (11).

(11)   It took John exactly five years to learn Russian.

This sentence states an exact measure on the time in which John learned Russian. It does not merely place an upper bound: if John actually learned Russian in three years, the sentence is false. Similarly, the entailment in (12) is not valid, since a model where John learned Russian in five years makes (11) true and (12b) false. Compare this with the entailment in (13), which is valid:

(12)   a.   It took John exactly five years to learn Russian.
        b. $\nRightarrow$ It took John exactly ten years to learn Russian.

(13)   a.   It took John five years or less to learn Russian.
        b. $\Rightarrow$ It took John ten years or less to learn Russian.

The problem with the temporal closure axiom is that it predicts the entailment in (12) to be valid. This is because (14a) entails (14b).

(14)   a. $\exists t.t < \textbf{now} \wedge \textbf{years}(t) = 5 \wedge \textbf{learn}([\text{ARG1}, j; \text{ARG2}, r; \text{T}, t])$
        b. $\exists t.t < \textbf{now} \wedge \textbf{years}(t) = 10 \wedge \textbf{learn}([\text{ARG1}, j; \text{ARG2}, r; \text{T}, t])$

For example, take a model where John learned Russian in the five-year interval *2000-2005*. In this model, (14a) will be true because the predicate **learn** will hold of (15a). But by the temporal closure axiom, the same predicate will also hold of (15b), and so (14b) is also true.

(15)   a. $[\text{ARG}1, j; \text{ARG}2, r; \text{T}, \textit{2000-2005}]$
        b. $[\text{ARG}1, j; \text{ARG}2, r; \text{T}, \textit{2000-2010}]$

So while (14a) and (14b) appear to be the most straightforward (and indeed the only) ways in which the sentences in (12) can be represented in linking semantics, the temporal closure axiom causes them to behave as if they had the meanings of the sentences in (13).

Fortunately, there is no need to abandon event semantics even if we want to capture the phenomena for which linking semantics was intended. The rest of this paper develops an event semantic account that eliminates the need for problematic statements like the argument reduction axiom and the temporal closure axiom.

4 Back to Events

The basic insight that leads to the system to be presented is that role assignments are very similar to sets of events. Take for example the role assignment $g_1 = [\text{ARG}1, j; \text{ARG}2, b; \text{T}, 1pm]$. This assignment corresponds to the property of being an event whose agent is John, whose theme is Bill, and which takes place at 1pm. This property could in principle apply to more than one event. For example, if John kicked and slapped Bill at the same time, then it would apply to the slapping event and to the kicking event. So in terms of event semantics, a role assignment corresponds to a set of events and not just to one event.

Traditional implementations of event semantics, such as Carlson (1984) and Landman (2000), model verbal projections (that is, verbs, verb phrases, sentence radicals, and so on) as sets of events, perhaps with additional lambda slots for verbal arguments. But for quantificational arguments, these traditional implementations need to resort to quantifying-in or other devices that Beaver and Condoravdi see as problematic. Let us therefore translate linking semantics into event semantics in a way that minimizes disturbance to their system and does not introduce the need for quantifying-in. Since the verbal projections of linking semantics denote properties of role assignments, and since these properties correspond to sets of events, it can be expected that the least disturbance to the system will result if we move to an event semantics in which verbal projections denote properties of sets of events, rather than properties of events.

The system in Champollion (2010, 2014), which I will call quantificational event semantics, fits the bill and its derivations are quite similar to the ones in linking semantics. This is no accident, because quantificational event semantics resulted from my attempt to reconstruct linking semantics in an event-based framework.

Quantificational event semantics is built around the idea that verbs and their projections denote predicates that hold of sets of events. A verb will be true of any set of events f so long as f contains (possibly among other things) an event that satisfies the relevant event predicate. For example, $[\![\text{rain}]\!]$ can be represented (ignoring tense and aspect) as in (16), where f stands for the sets of events to which the verbal predicate may be applied.

(16) $[\![\text{rain}]\!] = \lambda f_{\langle v, t \rangle} \exists e[\mathbf{rain}(e) \wedge f(e)]$

Assuming that the verb *rain* does not take any semantically visible arguments, and again ignoring tense and aspect for now, the truth conditions of a sentence like *it is raining* can then be obtained by checking whether the set of all events whatsoever, $\lambda e.\mathbf{true}$, has the property denoted by the verb. The meaning of *it is raining* comes out as follows (Schwarzschild 2014):

(17) $[\![\text{It is raining}]\!]$
        a. $= \lambda f \exists e[\mathbf{rain}(e) \wedge f(e)](\lambda e.\mathbf{true})$
        b. $= \exists e[\mathbf{rain}(e) \wedge (\lambda e.\mathbf{true})(e)]$
        c. $= \exists e[\mathbf{rain}(e) \wedge \mathbf{true}]$

    d. $= \exists e[\mathbf{rain}(e)]$

As for verbal arguments like *Mary*, they are assumed to combine with silent thematic heads like *[ag]* and *[th]*, so that, for example, $[\![[\text{ag] Mary}]\!]$ can be represented as $\lambda V \lambda f.V(\lambda e.[f(e) \wedge \text{AG}(e) = m])]$.

    The parallel to linking semantics is very close. For comparison, I show a linking semantic derivation in (18) and its quantificational event semantic counterpart in (19). Here as above, g, g' range over role assignments, and L over sets of role assignments. In addition, f ranges over sets of events, and V over sets of sets of events.

(18)   a. $[\![\text{Mary}]\!] = \lambda P.P(m)$
          b. $[\![\text{Mary:ARG1}]\!] = \lambda L \lambda g.L(g + [\text{ARG1}, m])$
          c. $[\![\text{-ed}]\!] = \lambda L \lambda g.L(g) \wedge g(\text{T}) < \mathbf{now}$
          d. $[\![\text{laugh -ed}]\!] = \lambda g.\mathbf{laugh}(g) \wedge g(\text{T}) < \mathbf{now}$
          e. $[\![\text{Mary:ARG1 laugh -ed}]\!] = \lambda g.\mathbf{laugh}(g + [\text{ARG1}, m]) \wedge g(\text{T}) < \mathbf{now}$
          f. $M \models \text{Mary laughed iff } \exists t[\mathbf{laugh}([\text{T}, t; \text{ARG1}, m]) \wedge t < \mathbf{now}]$

(19)   a. $[\![\text{Mary}]\!] = \lambda P.P(m)$
          b. $[\![[\text{ag] Mary}]\!] = \lambda V \lambda f.V(\lambda e.[f(e) \wedge \text{AG}(e) = m])$
          c. $[\![\text{-ed}]\!] = \lambda V \lambda f \exists t[t < \mathbf{now} \wedge V(\lambda e[f(e) \wedge \tau(e) \subseteq t])]$
          d. $[\![[\text{closure}]]\!] = \lambda V.V(\lambda e.\mathbf{true})$
          e. $[\![\text{laugh}]\!] = \lambda f \exists e[\mathbf{laugh}(e) \wedge f(e)]$
          f. $[\![[\text{closure] [ag] Mary laugh -ed}]\!] = \exists t[t < \mathbf{now} \wedge \exists e[\mathbf{laugh}(e) \wedge \text{AG}(e) = m \wedge \tau(e) \subseteq t]]$

In (19), I have deviated from linking semantics in distinguishing between the runtime of the event, $\tau(e)$, and the reference time interval of the sentence, t. Following standard practice, the morpheme *-ed* contributes both past tense and perfective aspect, so it relates $\tau(e)$ and t by temporal inclusion, written as $\subseteq$ (19c). This removes the need for the temporal closure principle.

    I represent (9a) (repeated below as (20)) as in (21). The underlined bit requires each visit to be contained in the reference interval, but does not require all visits to take place at the same time. For consistency with linking semantics, I also give the tense quantifier widest scope.

(20)   A diplomat visited every country.

(21) $\exists t.t < \mathbf{now} \wedge \exists x.\mathbf{diplomat}(x) \wedge \forall y.\mathbf{country}(y) \rightarrow \exists e.\mathbf{visit}(e) \wedge \mathbf{ag}(e) = x \wedge \mathbf{th}(e) = x \wedge \underline{\tau(e) \subseteq t}$

As for the matrix clauses of (12), I translate them as in (22).

(22) $[\![\text{It took John } n \text{ years to}]\!]$
        $= \lambda V \exists t.t < \mathbf{now} \wedge \mathbf{years}(t) = n \wedge V(\lambda e.\mathbf{ag}(e) = j \wedge \tau(e) = t)$

The embedded clause does not have past tense, and therefore does not contribute $\subseteq$. The underlined parts of (23a) and (23b) block the undesired inference in (12). This is so because we have $=$ where in linking semantics it is as if we had $\subseteq$ because of the temporal closure principle. Unlike in linking semantics, there is no overgeneration problem: (23a) does not entail (23b).

(23)   a. $\exists t.t < \mathbf{now} \wedge \underline{\mathbf{years}(t) = 5} \wedge \exists e[\mathbf{learn}(e) \wedge \mathbf{ag}(e) = j \wedge \mathbf{th}(e) = r \wedge \underline{\tau(e) = t}]$
          b. $\exists t.t < \mathbf{now} \wedge \underline{\mathbf{years}(t) = 10} \wedge \exists e[\mathbf{learn}(e) \wedge \mathbf{ag}(e) = j \wedge \mathbf{th}(e) = r \wedge \underline{\tau(e) = t}]$

5 Extension to Temporal Anaphora through Dynamic Semantics

There is one point in which linking semantics does afford additional formal expressivity compared to quantificational event semantics as I have introduced it so far. In their treatment of (24), Beaver and Condoravdi model the dependency between the two temporal modifiers by letting *in July* write its denotation into the value of T, and letting *last year* read it out and replace it with its own value which is computed as a function of the previous value. This is an instance of destructive assignment (van Eijck 2001). I give Beaver and Condoravdi's derivation in (25). This derivation assumes the existence of two partial functions from intervals to intervals, **the-july-in** and **the-year-before**, which

have the meanings you would expect. The notation $g[\text{T},m]$ is only defined when g already contains an entry for T. It can be read as the result of updating the entry for T in g by overwriting its previous value with m. The destructive assignment step just described occurs between steps (25d) and (25e), where the time of g is overwritten from "July" to "July of last year". (To be precise, prior to this one there is another instance of destructive assignment. In step (25d), *in July* overwrites the time of g from whatever interval it was to the July of that interval.)

(24)   Last year, it rained in July.

(25)   a. $[\![\text{in July}]\!] = \lambda L \lambda g. L(g[\text{T}, \textbf{the-july-in}(g(\text{T}))])$
        b. $[\![\text{last year}]\!] = \lambda L \lambda g. L(g[\text{T}, \textbf{the-year-before}(g(\text{T}))])$
        c. $[\![\text{it rained}]\!] = \lambda g. \textbf{rain}(g) \wedge g(\text{T}) < \textbf{now}$
        d. $[\![\text{it rained in July}]\!]$
              $= \lambda g. \textbf{rain}(g[\text{T}, \textbf{the-july-in}(g(\text{T}))]) \wedge \textbf{the-july-in}(g(\text{T})) < \textbf{now}$
        e. $[\![\text{Last year, it rained in July}]\!]$
              $= \lambda g. \textbf{rain}(g[\text{T}, \textbf{the-july-in}(\textbf{the-year-before}(g(\text{T})))])$
              $\wedge \textbf{the-july-in}(\textbf{the-year-before}(g(\text{T}))) < \textbf{now}$
        f. $M \models \text{Last year, it rained in July}$ iff
              $\exists t [\textbf{rain}([\text{T}, \textbf{the-july-in}(\textbf{the-year-before}(g(\text{T})))])$
              $\wedge \textbf{the-july-in}(\textbf{the-year-before}(g(\text{T}))) < \textbf{now}]$

There is no way to reproduce this behavior one-to-one in quantificational event semantics, as we cannot compositionally map a predicate of sets of events to another one in a way that would simulate the destructive assignment (the overwriting of runtimes) that B&C rely on. The problem already becomes clear when we try to translate the denotation of *last year* in (25a) to a format that is suitable for quantificational event semantics:

(26)   *Problematic translation:*
        $[\![\text{last year}]\!] = \lambda V \lambda f. V(\lambda e.[f(e) \wedge \tau(e) = \textbf{the-year-before}(\tau(e))])$

The problem with this translation is the statement $\tau(e) = \textbf{the-year-before}(\tau(e))$. This does not have the intended effect of "overwriting" the value of $\tau(e)$ with the year before it, but instead requires the two intervals to be identical. Since the mapping from times to years preceding them has to be irreflexive, this constraint cannot be satisfied. Worse, when two such modifiers are stacked, these equalities pile up instead of replacing one another. For example, here is what happens when we apply $[\![\text{in July}]\!]$ (defined analogously to (26)) to $[\![\text{it rained}]\!]$, defined as in (16), and then apply $[\![\text{last year}]\!]$ to the result. I continue to ignore tense for now, but including it would not make things better.

(27)   *Problematic derivation:* $[\![\text{last year}]\!]([\![\text{in July}]\!]([\![\text{it rained}]\!]))$
        $\lambda f. \exists e [\textbf{rain}(e) \wedge f(e) \wedge \tau(e) = \textbf{the-year-before}(\tau(e)) \wedge \tau(e) = \textbf{the-july-in}(\tau(e))]$

The problem here is that $\tau(e)$ is simultaneously constrained to be the year before itself and the July inside itself. The first conjunct is already contradictory, and the second conjunct is added to the first instead of replacing it. The problem arises from the fact that the analogy between sets of events and role assignments is not perfect. Sets are fundamentally different from partial functions in that there is no operation on sets that corresponds to re-assigning a variable.

In the following, I will extend quantificational event semantics by adding a dynamic layer on top of it that will afford us a simple treatment of temporal anaphora. Beyond making sure that we do not lose any coverage compared with linking semantics, my intention is to show that phenomena relating to temporal anaphora do not stand in contradiction with adopting an event semantic view of verbal denotations. There are many ways we could add a dynamic component. Here I consider one whose particular advantage is its modularity. It can be built on top of quantificational event semantics without disturbing its inner workings or without having to switch to another logic. Following Shan (2005), Barker and Shan (2014) and Charlow (2014), I consider binding relations to be linguistic side effects, a term introduced in Shan (2005) in analogy to the corresponding term in computer science. The basic idea of the analogy, applied to temporal modifiers like *last year* and *in July*, is that in addition to their core meaning, they retrieve and provide temporal values "on the side". In

fact, in the implementation to follow, I will give them a trivial core meaning: the identity function on verbal denotations. So temporal modifiers make their semantic contribution only via side effects.

In order for these these values to be communicated, I will enrich quantificational event semantics with a dynamic semantic layer that provides the necessary infrastructure for anaphoric links (Groenendijk and Stokhof 1990). The implementation here follows Charlow (2014). In technical terms, it can essentially be seen as lifting quantificational event semantics into the state monad (Wadler 1992). In the following, I will not assume prior knowledge with monads, though Charlow discusses the connection extensively.

We equip each lexical item with an additional lambda slot λt, which supplies the "input time", the temporal interval at which the lexical item is evaluated. We also form a pair of the ordinary denotation of each lexical item and the "output time". This can serve as an antecedent to subsequent modification. By default, the input time can be returned unchanged as the output time. This is the case, for example, in the denotation of *it rained*, given below:

(28) $[\![\text{it rained}]\!] = \lambda t.\langle \lambda f \exists e.\mathbf{rain}(e) \wedge f(e) \wedge \tau(e) \subseteq t \wedge \tau(e) < \mathbf{now} \, , \, t \rangle$

In this denotation, the argument λt is the way through which a temporal antecedent supplies the value at which the proposition "it rained" is evaluated. Once this argument has been supplied, a pair is returned whose first argument corresponds to what would be considered in quantificational event semantics the ordinary denotation of the verb (cf. (19e)) and whose second argument contains the time that serves as an antecedent to subsequent denotations. In this case, it is the input time t. (In this paper, I will not actually study any constructions where the output time of the verb serves as an antecedent, so this choice is mainly for convenience.) Tense supplies the requirement that $\tau(e)$ be located before now. This requirement should be properly handled as a definedness condition rather than as a part of the assertion. Beaver and Condoravdi make an analogous comment in a footnote. Like them, I will ignore this issue.

The second element of the pair in (28) can be seen as keeping track of the current state of the discourse. In the present implementation, it is an extremely impoverished state: the only thing that we keep track of is a single time interval. This corresponds to the way in which linking semantics manipulates the value of just one temporal variable at a time. We could enrich the picture in various ways, for example by passing assignment functions from variables to times. I come back to this point in Section 6.

As seen above, I write pairs of denotations a and b as $\langle a,b \rangle$. These pairs do not play an essential role in the system. A pair like $\langle a,b \rangle$ can be seen as a shorthand for $\lambda k.k(a)(b)$, where k is of the polymorphic type $\langle \alpha, \langle \beta, t \rangle \rangle$. The latter term is known as the Church encoding of pairs (Church 1936, Charlow 2014). Likewise, the type $\alpha \times \beta$ can be seen as a shorthand for the type $\langle \langle \alpha, \langle \beta, t \rangle \rangle, t \rangle$.

Temporal modifiers can be treated quite simply as identity functions whose side effect is to return as output time the result of applying the constant function **the-july-in** to the input time:

(29)   a. $[\![\text{in July}]\!] = \lambda t.\langle \lambda V.V, \ \mathbf{the\text{-}july\text{-}in}(t) \rangle$ $= \lambda t.\lambda k.k(\lambda V.V)(\mathbf{the\text{-}july\text{-}in}(t))$
        b. $[\![\text{last year}]\!] = \lambda t.\langle \lambda V.V, \ \mathbf{the\text{-}year\text{-}before}(t) \rangle$ $= \lambda t.\lambda k.k(\lambda V.V)(\mathbf{the\text{-}year\text{-}before}(t))$

Some temporal modifiers will require us to abandon the pair notation because they contain quantifiers that bind across both the first and the second element of a pair. Although I will not have space to demonstrate this in detail here, temporal quantifiers like *every day* are an example. As Beaver and Condoravdi discuss, such quantifiers can serve as antecedents to other temporal modifiers in their scope, as in *Every day^i, it rained in the afternoon$_i$*. So the output variable of *every day* should be bound by its universal quantifier. The entry in (30a) tries to achieve this but fails because of a type mismatch: a pair does not have the right kind of type to appear on the right hand side of the implication arrow. The entry in (30b) fixes this by using Church encoding and moving the variable k above the universal quantifier.

(30)   a.   *Problematic entry:* $[\![\text{every day}]\!] = \lambda t.\forall t'[\mathbf{day\text{-}in}(t',t) \rightarrow \langle \lambda V.V, \ t' \rangle]$
        b.   *Correct entry:* $[\![\text{every day}]\!] = \lambda t.\lambda k.\forall t'[\mathbf{day\text{-}in}(t',t) \rightarrow k(\lambda V.V)(t')]$

The way to combine such paired meanings is a bit more complicated than function application. I first present the algorithm informally in procedural terms, and then more formally below. For sister nodes F and A (mnemonics for *function* and *argument*, respectively), and an initial input time t which will eventually be lambda-abstracted over, the result R of applying F to A, or in other words the denotation of their mother node, is obtained in the following way (see Charlow (2014) on how to derive this procedure from first principles):

1. Apply $[\![F]\!]$ to t. This results in a pair $\langle F_{core}, F_{out} \rangle$, which may be thought of as the core meaning of F and the output time F_{out} respectively. For example, in the case of $F = [\![\text{in July}]\!]$, as indicated in (29a), the core meaning F_{core} will be the identity function $\lambda V.V$, and the output time F_{out} will be **the-july-in**(t), the July contained in the initial input time t.

2. Now apply $[\![A]\!]$ to that output time F_{out}. The output time of F thus becomes the input time of A. This again results in a pair of a core meaning and an output time, $\langle A_{core}, A_{out} \rangle$. For example, in the case of $A = [\![\text{it rained}]\!]$, as indicated in (28), the core meaning A_{core} will be $\lambda k \exists e.\textbf{rain}(e) \wedge k(e) \wedge \tau(e) \subseteq \textbf{the-july-in}(t) \wedge \tau(e) < \textbf{now}$, and the output time A_{out} will be the event time of e, that is, **the-july-in**(t).

3. Like F and A, the result R should be a function from input times t to a pair $\langle R_{core}, R_{out} \rangle$ of core meanings and output times. We set R_{core} to the result of applying F_{core} to A_{core} via ordinary function application, and we set R_{out} to the output time A_{out} of $[\![A]\!]$. In our current example, R_{core} is (the result of applying the identity function $\lambda V.V$ to) $\lambda k \exists e.\textbf{rain}(e) \wedge k(e) \wedge \tau(e) \subseteq \textbf{the-july-in}(t) \wedge \tau(e) < \textbf{now}$. As for R_{out}, it is **the-july-in**(t). Thus the result of applying *in July* to *it rained* is as follows:

(31) $[\![\text{it rained in July}]\!]$
        $= \lambda t.\langle \lambda f \exists e.\textbf{rain}(e) \wedge f(e) \wedge \tau(e) \subseteq \textbf{the-july-in}(t) \wedge \tau(e) < \textbf{now}, \textbf{the-july-in}(t) \rangle$

More formally, this procedure can be stated as in (32). Here, i is the type of temporal intervals, α is the type of a, and β the type of b. I write $\alpha \times \beta$ for the type of the pair $\langle a, b \rangle$.[1]

(32)   *Stateful function application:*
        Given a binary-branching node R whose daughters F (function) and A (argument, may be to the left or right of F) have the types $\langle i, (\langle a, b \rangle \times i) \rangle$ and $\langle i, (a \times i) \rangle$, the value of R, written $[\![F]\!]([\![A]\!])$, is given by $\lambda t.\langle f(a), t'' \rangle$ where $\langle f, t' \rangle = [\![F]\!](t)$ where $\langle a, t'' \rangle = [\![A]\!](t')$.

These "where" statements are shorthands. They expand as in (33). I write π_{core} for the first element of a pair π (its core meaning) and π_{out} for its second element (its output time):[2]

(33)   *Stateful function application (expanded):*
        $[\![F]\!]([\![A]\!]) \stackrel{\text{def}}{=} \lambda t.\langle (\pi_{core}([\![F]\!](t)))(\pi_{core}([\![A]\!](\pi_{out}([\![F]\!](t))))), \pi_{out}([\![A]\!](\pi_{out}([\![F]\!](t)))) \rangle$

We can now continue the derivation begun in the example. To do this, we first use stateful function application in order to apply $[\![\text{last year}]\!]$, shown in (29b), to (31). In step 1, we retrieve the (trivial) core meaning of $[\![\text{last year}]\!]$ and its output time, **the-year-before**(t). In step 2, we apply (31) to that output time. This results in the core meaning $\lambda f \exists e.\textbf{rain}(e) \wedge f(e) \wedge \tau(e) \subseteq \textbf{the-july-in}(\textbf{the-year-before}(t))$. Here we see that the functions introduced by the temporal modifiers have been stacked in the right way. We also get **the-july-in**(**the-year-before**(t)) as the output time. In step 3, we pack these two things into a pair, and abstract over t. The result is as follows:

(34) $[\![\text{last year}]\!]([\![\text{it rained in July}]\!])$
        $= \lambda t.\langle \lambda f \exists e.\textbf{rain}(e) \wedge f(e) \wedge \tau(e) \subseteq \textbf{the-july-in}(\textbf{the-year-before}(t)) \wedge \tau(e) < \textbf{now},$
        the-july-in(**the-year-before**(t))$\rangle$

---

[1]I thank Simon Charlow for discussion of this point.

[2]If one uses the Church encoding of pairs, this rule should be replaced by:

$[\![F]\!]([\![A]\!]) \stackrel{\text{def}}{=} \lambda t.\lambda k.[\![F]\!](t)(\lambda f.\lambda t'.[\![A]\!](t')(\lambda a.\lambda t''.k(f(a))(t'')))$.

We can now apply a closure operation that maps this to a truth condition. The resulting operator is shown below. Since its result type has to be t, it applies via ordinary function application, not via stateful function application. I use S for variables of type $\langle i, (\langle vt, t \rangle \times i) \rangle$.

(35) $[\![\text{past-closure}]\!] = \lambda S. \exists t. \pi_{core}(S(t))(\lambda e.\textbf{true})$

Applied to (34), this rule yields the right truth conditions:

(36) $\exists t \exists e.\textbf{rain}(e) \wedge f(e) \wedge \tau(e) \subseteq \textbf{the-july-in}(\textbf{the-year-before}(t)) \wedge \tau(e) < \textbf{now}$

6 Enriching the System with Variable Assignments

Linking semantics, as well as the extension of quantificational event semantics that I have presented so far, can only handle temporal anaphora between two temporal modifiers of the same event or role assignment. But in order to account for temporal anaphora across sentence boundaries, as in (37), it will be necessary to extend the system in ways that are familiar from dynamic semantics:

(37)   Last year^i, I lived$_i$ in Rome. I visited Bill in July$_i$.

The ability to cross sentence boundaries is just one of many ways in which temporal anaphora is parallel to nontemporal anaphora (Champollion 2012). To mention just one example, quantificational subordination appears to occur in the domain of temporal dependencies:

(38)   a.   Most books contain a table of contents. In some, it is at the end. (Heim 1990)
          b.   Every year, Arnim spends a week in the mountains. In some years, he hikes every day.

The present system can be easily extended to keep track of more than one time at once. In dynamic semantic fashion, one can equip all items that are anaphoric on a time with subscripts, and all items that provide antecedents with superscripts, as follows:

(39) $[\![\text{rain}_i^j]\!]_{alternative} = \lambda g. \langle \lambda f \exists e.\textbf{rain}(e) \wedge f(e) \wedge \tau(e) \subseteq g(i) \, , \, g + [j, g(i)] \rangle$

(40) $[\![\text{in July}_i^j]\!]_{alternative} = \lambda t. \langle \lambda V.V, \, g + [j, \textbf{the-july-in}(g(i))] \rangle$

7 Discussion

When I dynamicized quantificational event semantics for the treatment of temporal anaphora, I set up the system so that the output time of the function is always the input time of the argument. This is not the only logical possibility. I could have gone the other way round, or I could have implemented a default left-to-right evaluation order by giving the output time of the left daughter as an input time to the right daughter. Other empirical domains, such as pronoun binding, require left-to-right evaluation, and weak crossover phenomena show that right-to-left evaluation order is more than just a default preference (for details and qualifications, see Barker and Shan (2008, 2014)). But in the case of temporal anaphora, the importance of left-to-right evaluation order is more doubtful. As Beaver and Condoravdi (2007) observe, fronted modifiers are interpreted with wider scope than clause-internal ones:

(41)   a.   Last year, it rained { in July, every day }.
          b.   ?{ In July, Every day }, it rained last year.

I have followed Beaver and Condoravdi in interpreting and implementing this constraint in terms of scope (more specifically, function-argument directionality) rather than in terms of left-to-right surface order. If, as has been implicitly assumed here, the relation between the reference time of the verb and the temporal modifier that is adjacent to it is anaphoric in nature, then left-to-right surface order cannot constrain it. Otherwise there would be no way for temporal modifiers to appear to the right of the verb, as they do in (41b).

   In conclusion, there remains a strong motivation for Davidsonian event semantics as a logic of verbal modification. Once we move the event quantifier into the verb, we are able to provide an account of quantification that does not make use of quantifier raising or quantifying-in (two devices that Beaver and Condoravdi see as problematic). As for temporal anaphora, we have seen that a dynamic system can be layered on top of quantificational event semantics. This system provides room for extension and shows promise as a possible framework within which to account for temporal equivalents of phenomena like quantificational and modal subordination.

References

Barker, Chris, and Chung-chieh Shan. 2008. Donkey anaphora is in-scope binding. *Semantics and Pragmatics* 1:1–46.

Barker, Chris, and Chung-chieh Shan. 2014. *Continuations and Natural Langauge*. Oxford, UK: Oxford University Press.

Beaver, David, and Cleo Condoravdi. 2007. On the logic of verbal modification. In *Proceedings of the Sixteenth Amsterdam Colloquium*, ed. Maria Aloni, Paul Dekker, and Floris Roelofsen, 3–9. Amsterdam, Netherlands: University of Amsterdam.

Carlson, Gregory N. 1984. Thematic roles and their role in semantic interpretation. *Linguistics* 22:259–279.

Champollion, Lucas. 2010. Quantification and negation in event semantics. *The Baltic International Yearbook of Cognition, Logic and Communication* 6.

Champollion, Lucas. 2012. Temporal dependencies: anaphora vs. movement. Poster presentation at the 35th GLOW colloquium, Potsdam.

Champollion, Lucas. 2014. The interaction of compositional semantics and event semantics. To appear in *Linguistics and Philosophy*.

Charlow, Simon. 2014. On the semantics of exceptional scope. Doctoral dissertation, New York University.

Church, Alonzo. 1936. An unsolvable problem of elementary number theory. *American Journal of Mathematics* 345–363.

Davidson, Donald. 1967. The logical form of action sentences. In *The Logic of Decision and Action*, ed. Nicholas Rescher, 81–95. Pittsburgh, PA: University of Pittsburgh Press.

van Eijck, Jan. 2001. Incremental dynamics. *Journal of Logic, Language and Information* 10:319–351.

Groenendijk, Jeroen, and Martin Stokhof. 1990. Dynamic Montague grammar. In *Papers from the Second Symposium on Logic and Language*, ed. Lászlo Kálman and Lászlo Polos. Budapest, Hungary: Akadémiai Kiadó.

Heim, Irene. 1990. E-type pronouns and donkey anaphora. *Linguistics and Philosophy* 13:137–177.

Landman, Fred. 2000. *Events and Plurality: The Jerusalem Lectures*. Dordrecht, Netherlands: Kluwer.

Parsons, Terence. 1990. *Events in the Semantics of English*. Cambridge, MA: MIT Press.

Schwarzschild, Roger. 2014. Distributivity, negation and quantification in event semantics: Recent work by L. Champollion. Manuscript. Rutgers University/MIT, July 2014.

Shan, Chung-chieh. 2005. Linguistic side effects. Doctoral dissertation, Harvard University.

Wadler, Philip. 1992. Comprehending monads. *Mathematical Structures in Computer Science* 2:461–493.

Department of Linguistics
New York University
10 Washington Pl
New York, NY 10003
champollion@nyu.edu