A unified analysis of ‘respective’ and summative readings

The ‘respective’ and summative readings pose challenges to theories of the syntax-semantics interface.

(1) a. John and Bill married Sue and Mary (respectively).
   b. John and Bill spent a total of $10,000 last year.

One issue that any adequate account needs to deal with is that these expression can iterate:

(2) a. John and Bill introduced Mary and Sue to Chris and Pat (respectively).
   b. John and Bill introduced Mary and Sue to Chris and Pat on Thursday and Friday (respectively).
   c. John and Bill sent a total of 100 papers to a total of 30 referees.
   d. John and Bill donated a total of $10,000 to Red Cross and Salvation Army (respectively).

Note in particular that (2d) exhibits an interdependency involving both ‘respectively’ construed conjoined terms and a summative term: ‘John donated some amount of money to Red Cross and Bill donated some amount of money to Salvation Army, and they sum up to $10,000’.

Another challenge is that these phenomena are found not only with co-arguments of the same predicate (as in (1) and (2)) but also in more complex syntactic environments such as long-distance dependencies and nonconstituent coordination (NCC; right-node raising (RNR) in (4) and dependent cluster coordination (DCC) in (5)) we omit the examples with summative predicates for the sake of space:

(3) Which book and which magazine did John buy and Bill read (respectively)?
(4) John met, and Bill saw, Sue and Mary (respectively).
(5) John introduced Sue and Mary to Chris on Thursday and (to) Peter on Friday (respectively).

We propose, for the first time in the literature, a compositional analysis of ‘respective’ and summative readings which systematically accounts for both the iteration problem and interaction with NCC noted above. This analysis exploits specific properties of Hybrid Type-Logical Categorial Grammar (Hybrid TLCG; Kubota to appear; Kubota and Levine 2014)—in particular, the recognition of both directional and non-directional modes of implication proven successful in other empirical domains (especially Gapping and NCC)—to capture the interactions between ‘respective’ and summative readings and non-canonical coordination to yield direct compositional interpretations with no special extra mechanisms.

Following Winter (1995) and Bekki (2006), we make extensive use of the notion of product in capturing the ‘respective’ (and summative) readings. The key underlying idea is that conjoined (and plural) terms denote products and that the ‘respective’ readings arise when these product-type terms are related to each other in a pairwise manner with respect to some relational predicate. A prosodically empty ‘respective’ operator with the semantics in (6) is responsible for this ‘respective’ predication as in (7).

(6) \( \text{resp} = \lambda R \lambda X, \lambda Y. \prod_i (\pi_i(X_i) \times \pi_i(Y_i)) \)
(7) \( \text{resp}(\text{marry})(\langle (m, s), (j, b) \rangle) = (\text{marry}(m)(j), \text{marry}(s)(b)) \)

The \( \text{resp} \) operator returns a product of two propositions. We assume that this product of propositions is converted to a boolean conjunction by the boolean conjunction operator \( a \) la Gawron and Kehler (2004).

Since the two product-type terms that enter into the ‘respective’ relation are not necessarily co-arguments of a verb (as they happen to be in (1) and (2); see (3)–(5)), the ‘respective’ operator is given the following polymorphic specification, with the syntactic category \( (Z | X | Y) \times (Z | X | Y) \) (where \( \varphi_1, \varphi_2 \) and \( \sigma \) are prosodic variables of type \text{st} (string) and \text{st} \rightarrow \text{st}, respectively; the use of linguistic expressions with functional prosodies has been preceded in Oehrle (1994) and Muskens (2003)):

(8) \( \lambda \sigma \lambda \varphi_1 \lambda \varphi_2. \sigma(\varphi_1)(\varphi_2); \text{resp}; (Z | X | Y) \times (Z | X | Y) \)

The \( \text{resp} \) operator maps a functor of type \( Z | X | Y \) to another functor of the same type. Phonologically, it is an identity function over \( \text{st} \rightarrow \text{st} \rightarrow \text{st} \) functions. In Hybrid TLCG, hypothetical reasoning with the non-directional slash \(|\) (for which the argument is written to the right) has the effect of \( \lambda \)-binding both in phonology and semantics, and can be used to create functional expressions with any arbitrary type (unlike the directional slashes \(/\) and \( \backslash \), which are sensitive to string positions of the hypothesis in the phonological component). Thus, from the standard lexical type assignment \( \text{NP} \langle S \rangle / \text{NP} \) for the verb married (with a simple string-type phonology), the sign in (9) can be derived, which can then be given as an argument to the ‘respective’ operator in (8) to yield the ‘respectivized’ transitive verb in (10).
For summative terms, we assume that they essentially denote objects that are decomposable to multiple parts adding up to the named amount. Such an object can be formally modelled in terms of products by assuming that each component of the product corresponds to the relevant ‘subparts’ of the whole amount. The ‘respective’ operator in (8) then takes care of the pairwise mapping between the coordinated term and the summative term in examples like (1b) in a way parallel to ‘respective’ readings.

The proposed analysis immediately accounts for interactions with NCC. In Hybrid TLCG, a string like to Chris on Thursday in (5) can be treated as a full-fledged constituent via hypothetical reasoning with / and \. The NCC example (5) differs from the simpler (1a) only in the category of one of the two product-type arguments for the resp operator. Instead of a product of entities of type e, we here have a product of dependent cluster constituents. Thus, the derived sign for John introduced in (11) (to which the resp operator has already applied) takes two products of types NP and NP\PTV\VP (where VP and PTV abbreviate NP\S and (NP\S)/PP/NP respectively) and returns a pair of propositions in (12).

\[
\begin{align*}
(11) \quad & \lambda \varphi_3 \lambda \varphi_4 \circ \text{married} \circ \varphi_3; \text{marry}; (S|NP)\NP \\
(12) \quad & \text{thursday}((\text{intr}(s)(c))(j), \text{friday}((\text{intr}(m)(p))(j))
\end{align*}
\]

Iterated ‘respective’ and summative readings in (2) are derived essentially via recursive application of the resp operator. With the three-place resp operator (8), we can derive the following expression of type S|NP for (2a), which is missing the subject argument:

\[
\begin{align*}
(13) \quad & \lambda \varphi_3 \circ \text{ introduced} \circ \text{ mary} \circ \text{ and} \circ \text{ sue} \circ \text{ to} \circ \text{ chris} \circ \text{ and} \circ \text{ pat}; (\text{intr}(m)(c), \text{intr}(s)(p)); S|NP \\
\end{align*}
\]

This product of two predicates can then be related to the product-denoting subject argument John and Bill to yield the right meaning for the whole sentence as in (14) via a two-place ‘respectively’ operator in (15) (which, as we show in the full paper, is a corollary derived from the three-place one in (8)).

\[
\begin{align*}
(14) \quad & \lambda X_{\varphi} \lambda P_{\varphi} \cdot \prod_i \pi_i(P_{\varphi}((\pi_i(X_{\varphi}))(j), b)) = (\text{intr}(m)(c)(j), \text{intr}(s)(p)(b)) \\
(15) \quad & \lambda q_1 \lambda q_1 \circ \varphi_1; \lambda X_{\varphi} \lambda P_{\varphi} \cdot \prod_i \pi_i(P_{\varphi}((\pi_i(X_{\varphi}))); (S|NP)|S|NP)
\end{align*}
\]

We conclude with a brief comparison with related approaches. Our analysis builds on Gawron and Kehler’s (2004) (G&K’s) analysis of ‘respective’ readings (especially in the definition of the resp operator), but simplifies it by modelling the meanings of conjoined and plural terms as products rather than sums. Our account improves on G&K’s account by embedding the analysis of ‘respective’ readings in the general syntax-semantics interface of Hybrid TLCG, thereby accounting for the interactions with RNR and DCC straightforwardly. This is not a mere theoretical elegance. The most recent PSG analysis of ‘respective’ and related predicates partially addresses the RNR case (Chaves, 2012), but still fails to extend to the DCC case. For the syntax-semantics interface, our analysis via hypothetical reasoning for \ resembles Barker’s (2007) analysis of same via ‘parasitic scope’. Our approach extends to the analysis of symmetrical predicates straightforwardly and covers cases in which multiple symmetrical predicates interact with each other (John and Bill bought the same book at different stores), as we demonstrate in the full paper. Barker’s analysis covers only the same/same interactions (Barker, 2012) and fails to extend to these more complex cases. Recasting the analysis of ‘respective’ and related predicates in the logical setup of TLCG moreover has one intriguing conceptual advantage in addition to the empirical advantages noted above. In the formal calculus of Hybrid TLCG, we can model both the local, ‘rule-based’ analysis of ‘respective’ readings by G&K and the nonlocal, ‘movement-like’ analysis (via ‘parasitic scope’) a la Barker, and the two can be shown to yield exactly the same outcome. (Thus, no ‘mischief’ is introduced in our setup than what is already there in G&K’s analysis as far as the basic analysis of ‘respective’ readings is concerned, but the added flexibility directly yields the syntax and semantics that supports a straightforward analysis of NCC cases.) This, we argue, enables us to gain a deeper understanding of the common underlying compositional mechanism for ‘respective’, symmetrical, and summative predicates that have remained elusive in previous attempts at characterizing the semantics of these constructions.

References