A compositional account of VP ellipsis

Markus Egg and Katrin Erk

Abstract

We present an approach to VP ellipsis that allows the direct derivation of source and target sentences (the former need not be unique) during semantic construction. Specific syntactic constituent structures are associated with ellipsis potential, which can then be discharged by pro-verbs like *did (too)*. The determination of source and target sentence, which is done with semantic features in an HPSG framework, is coupled with a comprehensive analysis of ellipsis, which also handles its interaction with scope and anaphora.

1 Introduction

VP ellipses like (1) consist of two sentences, the *source sentence* (SS) and the *target sentence* (TS), which have the same meaning except for the semantic contributions of the respective subject NPs. In the TS the VP is replaced by an appropriate pro-form, here, *does too*:

(1) John wants to read *Jane Eyre*, and Bill does too.

Sentence (1) is not ambiguous w.r.t. potential SSs. Thus, there is only one feasible way of understanding the TS: 'Bill wants to read *Jane Eyre*'. But this is not always so. E.g., the TS of (2) may be understood as 'Bill reads' or 'Bill wants Max to read'. Such sentences, where the TS is part of a relative clause within an object NP, are called 'antecedent-contained deletions' (ACD).

(2) John wants Max to read everything that Bill does.

There are few previous approaches to an automatic determination of source and target sentences in VP ellipsis. Hardt (1997) considers all VPs within the previous three sentences as potential SS-VPs, and ranks them by several heuristics. We feel, however, that the set of possible SSs can be narrowed down much more. We associate specific syntactic constituent structures with ellipsis potential, i.e. information on where the SS candidates are if a VP ellipsis occurs. E.g., sentence conjunction has ellipsis potential, as do structures that underlie ACD cases (in particular, VPs that consist of a transitive verb and an NP). This potential can only be 'discharged' by a pro-form like *do (too)*. For ellipses like (2) that are semantically underdetermined since there is more than one feasible SS, we intend to model this underdetermination by *semantic underspecification*.
The traditional approach to VP ellipsis (Sag 1976; Fiengo and May 1994) regards it as deletion of the phonological realization of the TS-VP, which is identical (on some level of linguistic representation) to the SS-VP. However, this runs into problems with Hirschbühler (1982) examples like (3). Although both TS and SS are two ways ambiguous, (3) as a whole has only three readings (the third giving *a workshop* scope over both sentences), since the scope position of *a workshop* must be the same in both cases.

(3) Every linguist attended a workshop. Every computer scientist did, too.

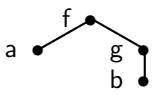
Hence we follow Dalrymple et al. (1991) and relate SS and TS as wholes (not only their VPs): their semantic structures are identical, except for the semantic contributions of their subject NPs.

2 CLLS

CLLS, the *Constraint Language over Lambda Structures* (Egg et al. 2000), is a formalism for underspecified semantic representations, which we employ in our semantic representations of ellipses.

2.1 Introduction to CLLS

CLLS is a language for partial tree descriptions, which we use for partial descriptions of λ-terms. Let us explain this in two steps. First, terms can be represented as trees. The i-th argument of the term becomes the i-th child. E.g., the term $f(a, g(b))$ can be drawn as the tree to the right.



Second, λ-terms can be represented as trees decorated with λ-links, so-called λ-*structures*. E.g., the λ-term $\lambda x.f(x)$ can be drawn as the λ-structure to the right, where λ-abstraction is represented by a lam node. Instead of the variable name, the dashed arrow uniquely identifies the binder of the variable. Furthermore, we represent application by a node labelled @, as in (5) below.

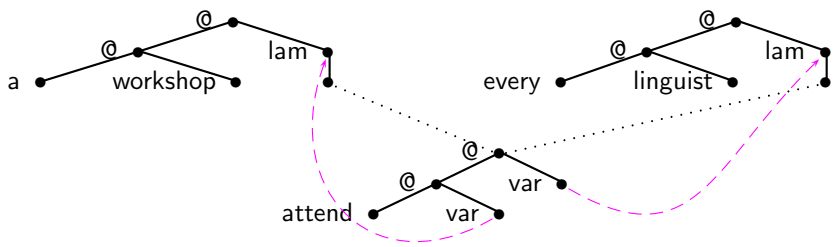
By a partial description of a λ-structure, we can model semantic underdetermination like *scope ambiguity*. For instance, the first sentence of (3) has two closely related readings (either quantifier may outscope the other one) as represented in (4):

(4) (a)
(a workshop)(λx
(every linguist)(λy
(attend x) y))

(b)
(every linguist)(λy
(a workshop)(λx
(attend x) y))

Two λ-structures correspond to these λ-terms. We describe them both in a single CLLS constraint:

(5)



(5) is satisfied by all λ-structures into which the graph can be embedded in such a way that no labelled nodes of the graph overlap. Dotted edges in (5) signify *dominance*: the upper node must be above the lower one in the λ-structure. The description leaves the ordering between the quantifier fragments unspecified. But since both fragments dominate the nuclear scope and, like trees, λ-structures cannot branch upwards, one of the quantifier fragments must dominate the other. Pictures like (5) have a formal meaning as *constraints over λ-structures*, see Egg et al. (2000).

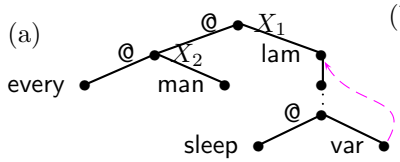
2.2 Representing ellipsis in CLLS

In CLLS, ellipsis is modelled by *parallelism constraints*. They express structural equality between pieces of λ-structures. Consider e.g. the elliptical (6):

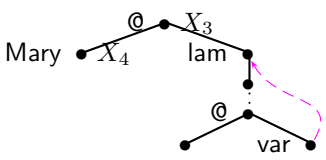
(6) Every man slept, and Mary did, too

The meaning of its TS *so does Mary* is equal to the one of its SS *every man slept*, except that the semantic contribution of the source parallel element *every man* is replaced by the one of the target parallel element *Mary*. The constraints for SS and TS are (7a) and (7b). The first part of the constraint for sentence (6) conjoins the SS and TS structures, as expressed in (7c). The X_i are *node variables*. Two occurrences of the same variable name, like X_1 in (7a) and (7c), describe the same variable, so the two constraints (7a) and (7c) are to be joined at X_1.

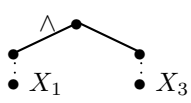
(7) (a)



(b)



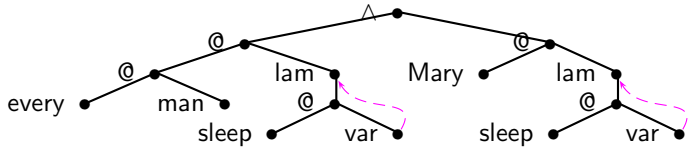
(c)



The second part of the constraint is the *parallelism constraint* (8), which (roughly) states that the constraint part between X_1 and X_2 must have the same structrure as that between X_3 and X_4.[1] (9), the intended semantic representation of (6), satisfies all constraints in (7a), (7b), (7c), and (8).

(8) $X_1/X_2 \sim X_3/X_4$

(9)

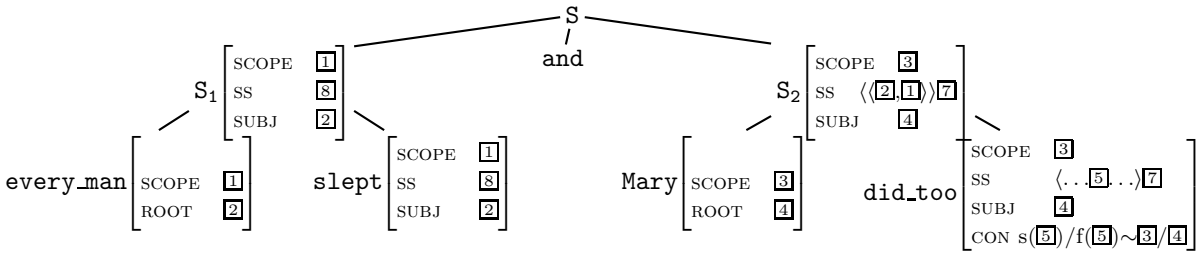


CLLS can also handle the interaction of ellipsis and scope (as in (3)) and the interaction of ellipsis and anaphora that arise from 'strict' and 'sloppy' reconstructions of pronouns. We refer to Egg et al. (2000) for analyses of these issues in CLLS.

3 The analysis

Our syntax-semantics interface assigns each syntactic constituent a CLLS constraint: the combination of the constraints of its immediate constituents with a constraint associated with the phrase structure rule itself. For the construction of complex constraints, each constituent makes accessible two distinguished node variables of its constraint by auxiliary features in the SEM value. (MRS (Copestake et al. 1997) uses the same technique.) The value of SEM|SCOPE models the local scope domain. The value of SEM|ROOT is the uppermost node variable of the semantic contribution of the constituent. As an example, consider the phrase structure rule (15b) and the CLLS constraint (15c) that is associated with it. The node variable $X^r_{\texttt{NP}}$ is the ROOT variable of the (semantic contribution of the) NP constituent, and analogously for $X^r_{\texttt{TV}}$ and $X^r_{\texttt{VP}}$. The constraints for TV, NP, and VP are connected through the variables they share, viz., $X^r_{\texttt{NP}}$ and $X^r_{\texttt{TV}}$.
Our analysis of ellipsis uses these features and two additional ones in the SEM value: SEM|SUBJ contains the ROOT node of the constraint for the subject of the current sentence. SEM|SS is a list of SS candidates. An SS candidate is a pair consisting of a SCOPE variable (for the local scope domain of the SS candidate) and the SUBJ variable for the candidate sentence.

(10)



Consider e.g. sentence (6). We show how its constraint, the conjunction of (7a-c) and (8), is derived from its constituent structure (10). (The irrelevant NP subtree in the SS is omitted.) The relevant parts of the interface rules are shown in (11). They are simplified in that the respective immediate constituents appear as DTR1 ... DTRn; dots in paths abbreviate SYNSEM|LOC.

[1] Since proper names and quantifier NPs can be parallel elements in an ellipsis (as in (6)), proper names are type-raised; the constant Mary in (7b) corresponds to a λ-term of type $\langle\langle e,t\rangle,t\rangle$.

(11) (a) lex. entry *do (too)*

$$\ldots \mid \textsc{sem} \begin{bmatrix} \textsc{scope} & \boxed{1} \\ \textsc{ss} & \langle \ldots \boxed{2} \ldots \rangle \\ \textsc{subj} & \boxed{3} \\ \textsc{con} & \mathrm{s}(\boxed{2})/\mathrm{f}(\boxed{2}) \sim \boxed{1}/\boxed{3} \end{bmatrix}$$

(b) `NP VP` → `S`

$$\begin{bmatrix} \textsc{dtr1} \mid \ldots \mid \textsc{sem} \mid \textsc{root} & \boxed{1} \\ \textsc{dtr2} \mid \ldots \mid \textsc{sem} & \begin{bmatrix} \textsc{ss} & \boxed{2} \\ \textsc{subj} & \boxed{1} \end{bmatrix} \\ \ldots \mid \textsc{sem} & \begin{bmatrix} \textsc{ss} & \boxed{2} \\ \textsc{subj} & \boxed{1} \end{bmatrix} \end{bmatrix}$$

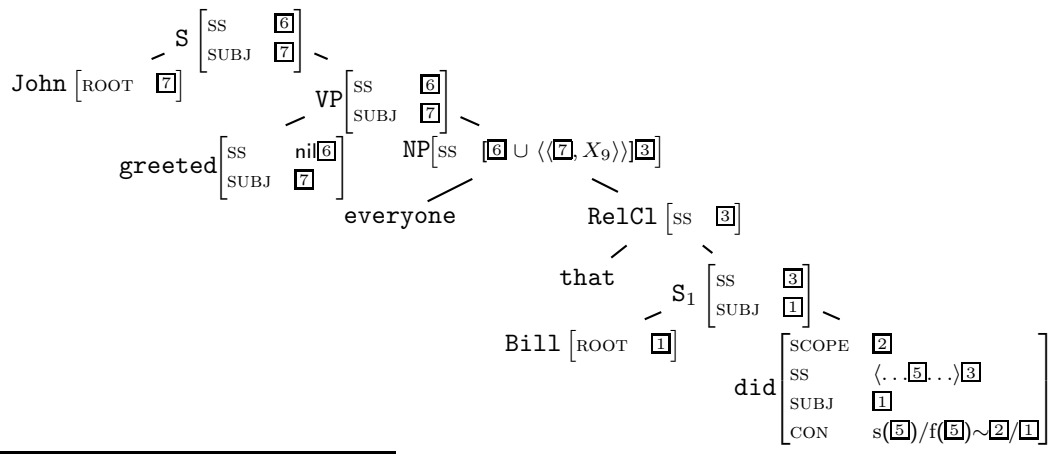
(c) `S Conj S` → `S`

$$\begin{bmatrix} \textsc{dtr1} \mid \ldots \mid \textsc{sem} & \begin{bmatrix} \textsc{scope} & \boxed{1} \\ \textsc{subj} & \boxed{2} \end{bmatrix} \\ \textsc{dtr3} \mid \ldots \mid \textsc{sem} \mid \textsc{ss} & \langle \langle \boxed{2}, \boxed{1} \rangle \rangle \end{bmatrix}$$

We start at $\mathtt{S}_1$. Here (11b) enforces head percolation of SS and SUBJ values and identifies the SUBJ value of the sentence with the `NP`'s ROOT value, here, the root $\boxed{2}$ of `every_man`. Now at `S`, (11c) builds up ellipsis potential for the SS candidate $\mathtt{S}_1$. It identifies the SS value of its DTR3 (later to be determined as $\mathtt{S}_2$) with a singleton list.[2] Its member is a pair consisting of the SUBJ and SCOPE values of $\mathtt{S}_1$. Due to (11b), the SS value $\boxed{7}$ and the SCOPE value $\boxed{3}$ of $\mathtt{S}_2$ percolate down to `did_too`, and the SUBJ of `did_too` is identified with the ROOT value $\boxed{4}$ of `Mary`. Now in `did_too`, the built-up ellipsis potential is discharged (11a): The CON value is the semantic contribution of the pro-form, the parallelism constraint (8) ('f' and 's' are functions that map a pair to its first and second member, respectively). The four node variables involved are the local scope and subject root variables of the SS and the TS. The SS values are a tuple from the SS list, which in this case possesses only one entry anyway. The TS local scope and subject node variables are the SCOPE and SUBJ values of `did_too`. Thus, the meaning of $\mathtt{S}_2$ with the exception of the semantic contribution of *every man* is the same as the one of $\mathtt{S}_1$ except for the semantics of *Mary*.
Next we analyse the ACD sentence (12). Its constituent structure is (13), the CLLS constraint for its semantics, (14). (In (13) the syntactic structure of the elliptical NP is simplified.) This analysis uses additional interface rules, depicted in (15). The rule (15b) is applied at `VP`, it expresses head percolation of SS and SUBJ values. The SS value $\boxed{6}$ of the finite verb *greeted* is the empty set (which terminates the derivation of the SS value of the sentence), so this sets the SS value of `VP` to nil. Furthermore (15b) adds a new element to the `NP`'s set of potential source sentences. The first element of this tuple is the SUBJ of `VP`, which is the ROOT of `John`. The second element of the tuple, the local scope variable for this SS candidate, is a node variable X_9 that is introduced in the semantic construction for `VP`, shown in (15c). This constraint reappears as part of (14).

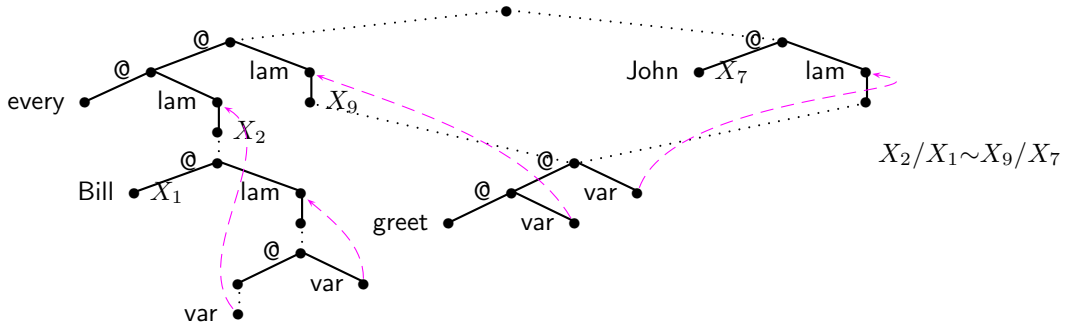
(12) John greeted everyone that Bill did.

(13)



(14)

[2] This is a simplification, as there may be more than one SS candidate also in the case of sentence conjunction – see (16). We envisage that in a future analysis, the lexical entries of conjunctions will carry information to that respect.



Rule (15a) is applied at `NP` and `RelCl`. It expresses percolation of the SS value to the semantic head of the constituent. Finally, the pro-form again discharges the built-up ellipsis potential by (11a). It picks the only element $\boxed{5}$ of the SS list $\boxed{3}$ to determine the SS and thus contributes the parallelism constraint $X_2/X_1 \sim X_9/X_7$.

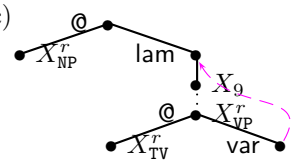
(15) (a) `NP RelCl` → `NP`
`RelPron S` → `RelCl`

$$\begin{bmatrix} \text{DTR2} \mid \ldots \mid \text{SEM} \mid \text{SS} & \boxed{1} \\ \ldots \mid \text{SEM} \mid \text{SS} & \boxed{1} \end{bmatrix}$$

(b) `TV NP` → `VP`

$$\begin{bmatrix} \text{DTR1} \mid \ldots \mid \text{SEM} & \begin{bmatrix} \text{SS} & \boxed{1} \\ \text{SUBJ} & \boxed{2} \end{bmatrix} \\ \text{DTR2} \mid \ldots \mid \text{SEM} \mid \text{SS} & \boxed{1} \cup \langle \boxed{2}, X_9 \rangle \\ \ldots \mid \text{SEM} & \begin{bmatrix} \text{SS} & \boxed{1} \\ \text{SUBJ} & \boxed{2} \end{bmatrix} \end{bmatrix}$$

(c)



The more complicated ACD sentence (2) is analyzed in a similar way, with two differences. First, the interface rule for control verbs adds a tuple to the argument VP's SS list. In the case of (2) it consists of the ROOT of `Max` and a local scope dominated by the semantics of `wants`. Second, `to read` does not set its SS value to `nil`, in contrast to `greeted` above. This encodes the observation that the SS list of a VP whose head is infinite can be further augmented if the VP is the argument of another verb, e.g. a control or raising verb. So in this case, the SS list from which the SS of `Bill does` is determined has two elements, corresponding to the two potential SSs in (2).

4 Further research

Our approach paves the way for analyses of many more elliptical constructions. E.g., the sentence conjunction rule (11c) could be extended to deal with cases like (16) (Hardt 1997). Here the conjunction *as* seems to build an ellipsis potential with more than one potential SS; in (16), either the main clause or the *when*-clause. This illustrates the question raised above: do different conjunctions build up different ellipsis potential?

(16) Gold still acts as a haven when uncertainty prevails in the markets, as it did yesterday.

Our analysis also allows us to tackle sentences with multiple ACD like (17). There is some disagreement about its meaning, but one possible interpretation is 'for all person-thing pairs x and y: if Bill gave x to y and Max gave x to y, John gave x to y'. But, then, (17) resembles Bach-Peters sentences: either quantifier comprises in its restriction a variable to be bound by the other one.

(17) John gave everything that Bill did to everyone that Max did

(18) John likes golf, and Mary too.

We also envisage an extension of our approach to ellipsis phenomena beyond VP ellipsis, like gapping as illustrated by (18) (the verb as well as a complement is missing in the TS). Here the source parallel element need not be the subject of the SS, which for (18) yields two interpretations of the TS, 'John likes Mary' and 'Mary likes golf'.

References

Copestake, A., D. Flickinger, and I. Sag (1997). Minimal Recursion Semantics. An introduction. Available from `ftp://ftp-csli.stanford.edu/linguistics/sag/mrs.ps.gz`.

Dalrymple, M., S. Shieber, and F. Pereira (1991). Ellipsis and higher-order unification. *Linguistics & Philosophy 14*, 399–452.

Egg, M., A. Koller, and J. Niehren (2000). The constraint language over lambda-structures. To appear in *Journal of Logic, Language, and Information* 10; available from `http://www.coli.uni-sb.de/~egg/Papiere/clls2000.ps.gz`.

Fiengo, R. and R. May (1994). *Indices and Identity*. Cambridge: MIT Press.

Hardt, D. (1997). An empirical approach to VP ellipsis. *Computational Linguistics 23*, 525–541.

Hirschbühler, P. (1982). VP deletion and across the board quantifier scope. In *Proc. NELS 12*.

Sag, I. (1976). *Deletion and logical form*. Ph. D. thesis, MIT, Cambridge.