



HAL
open science

On the semantics of dependencies: relative clauses and open clausal complements - extended abstract -

Philippe de Groote

► To cite this version:

Philippe de Groote. On the semantics of dependencies: relative clauses and open clausal complements - extended abstract -. Logic and Engineering of Natural Language Semantics 20 (LENLS20), Nov 2023, Osaka, Japan. hal-04402969

HAL Id: hal-04402969

<https://inria.hal.science/hal-04402969>

Submitted on 18 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

On the semantics of dependencies: relative clauses and open clausal complements

— extended abstract —

Philippe de Groote

LORIA, UMR 7503, Université de Lorraine, CNRS, Inria,
54000 Nancy, France

1 Introduction

Dependency grammars, which stem from a long linguistic tradition [12, 7], have seen a resurgence of interest in recent years, especially due to the Universal Dependency project [9]. Their principle is to make explicit the dependency relations existing between the words of a sentence, these relations being identified by the syntactic functions they represent.

Dependency parsing offers an interesting alternative to constituent parsing. In particular, parsing an agrammatical sentence does not result in a failure, but in a partial dependency structure that still contains some information. For this reason, dependency parsing is considered to be more robust than constituent parsing.

Formal compositional semantics, in the tradition of Montague [8], is based on a homomorphism between syntactic structure and semantic representation. As a result, it relies heavily on the notion of constituent, which does not appear explicitly in dependency structures. For this reason, providing a dependency grammar with a Montagovian semantic interpretation is not straightforward.

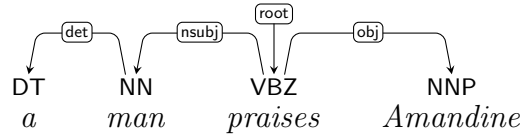
Nonetheless, in recent years, several ways of adapting Montague's semantics to the case of dependency grammars have been proposed in the literature. In a series of papers, Haug and his co-authors show how to assign a formal interpretation to a dependency structure using glue semantics [6, 3, 5, 4]. Other authors, including myself, rely on a compositionality principle close to Montague's by exploiting a functional representation of dependency structures [10,

11, 2]. In this paper, we continue the latter line of thought. In particular, we aim to show that the basic principles that we have stated in [2] allow one to deal with more advanced syntactic phenomena. We consider two cases: the relative clauses (which depend on the `acl:relcl` relation) and the open clausal complements (which depend on the `xcomp` relation).

2 Relative clauses and wh-extraction

In this section, we propose a solution to the problem of the semantic treatment of wh-extraction. We deal with relative clauses as a paradigmatic example.

First, consider the following sentence together with its dependency tree:



Following the approach that we have advocated in [2], the above sentence may be assigned a meaning by interpreting the following term:

$$\text{root (nsubj (obj PRAISES AMANDINE) (det MAN A))}$$

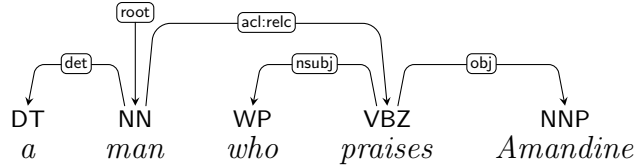
using the following semantics recipes:

$$\begin{aligned}
A &= \text{SOME} = \lambda pq. \exists x. (p x) \wedge (q x) : \text{DET} \\
\text{MAN} &= \lambda da. d(a \mathbf{man}) : \text{NP} \\
\text{AMANDINE} &= \lambda dap. p \mathbf{a} : \text{NP} \\
\text{PRAISE} &= \lambda p. \exists e. (\mathbf{praise} e) \wedge (p e) : \text{VP} \\
\text{det} &= \lambda nd. \lambda ea. n d a : \text{NP} \rightarrow \text{DET} \rightarrow \text{NP} \\
\text{nsubj} &= \lambda vn. \lambda p. n \text{SOME} (\lambda x. x) (\lambda x. v (\lambda e. (\mathbf{agent} e x) \wedge (p e))) : \text{VP} \rightarrow \text{NP} \rightarrow \text{VP} \\
\text{obj} &= \lambda vn. \lambda p. n \text{SOME} (\lambda x. x) (\lambda x. v (\lambda e. (\mathbf{theme} e x) \wedge (p e))) : \text{VP} \rightarrow \text{NP} \rightarrow \text{VP} \\
\text{root} &= \lambda v. v (\lambda e. \mathbf{true}) : \text{VP} \rightarrow \text{S}
\end{aligned}$$

where:

$$\begin{aligned}
\text{DET} &= (\mathbf{e} \rightarrow \mathbf{t}) \rightarrow (\mathbf{e} \rightarrow \mathbf{t}) \rightarrow \mathbf{t} \\
\text{NP} &= ((\mathbf{e} \rightarrow \mathbf{t}) \rightarrow (\mathbf{e} \rightarrow \mathbf{t}) \rightarrow \mathbf{t}) \rightarrow ((\mathbf{e} \rightarrow \mathbf{t}) \rightarrow \mathbf{e} \rightarrow \mathbf{t}) \rightarrow (\mathbf{e} \rightarrow \mathbf{t}) \rightarrow \mathbf{t} \\
\text{VP} &= (\mathbf{v} \rightarrow \mathbf{t}) \rightarrow \mathbf{t} \\
\text{S} &= \mathbf{t}
\end{aligned}$$

Now consider the following noun phrase, where the `acl:relc` relation indicates that a noun governs a verb that is the head of a relative clause:



A usual way to give a semantic account of wh-movement is by means of λ -abstraction and, possibly, combinators to simulate it. This is the approach we follow. Accordingly, we posit:

$$\begin{aligned} \text{WNP} &= e \rightarrow \text{NP} \\ \text{WVP} &= e \rightarrow \text{VP} \end{aligned}$$

This allows our semantic lexicon to be extended as follows:

$$\begin{aligned} \text{WHO} &= \lambda x d a p . p x : \text{WNP} \\ \text{acl:relc1} &= \lambda n v d a . n d (\lambda p . a (\lambda x . (p x) \wedge (v x (\lambda e . \mathbf{true})))) : \text{NP} \rightarrow \text{WVP} \rightarrow \text{NP} \end{aligned}$$

But then, the expression below gives rise to a type mismatch:

nsubj PRAISES WHO

Indeed, (`nsubj PRAISES`) is of type $\text{NP} \rightarrow \text{VP}$ while `WHO` is of type `WNP`. One way out of this typing problem is to coerce the types of the dependency relations using combinators similar to the `B` and `C` combinators of combinatory logic [1]:

$$\begin{aligned} \text{warg1} &= \lambda r v n x . r (v x) n : (\text{VP} \rightarrow \text{NP} \rightarrow \text{VP}) \rightarrow \text{WVP} \rightarrow \text{NP} \rightarrow \text{WVP} \\ \text{warg2} &= \lambda r v n x . r v (n x) : (\text{VP} \rightarrow \text{NP} \rightarrow \text{VP}) \rightarrow \text{VP} \rightarrow \text{WNP} \rightarrow \text{WVP} \end{aligned}$$

Then, the semantic interpretation of the noun phrase, *a man who praises amandine*, may be computed by evaluating the following expression:

$$\text{root} (\text{det} (\text{acl:relc1} \text{MAN} (\text{warg2} \text{nsubj} (\text{obj} \text{PRAISES} \text{AMANDINE}) \text{WHO}))) \text{A}$$

which reduces to the following formula:¹

$$\lambda q . \exists x . (\mathbf{man} x) \wedge (\exists e . (\mathbf{praise} e) \wedge (\mathbf{theme} e \mathbf{a}) \wedge (\mathbf{agent} e x) \wedge \mathbf{true}) \wedge (q x)$$

¹ When applied to a noun phrase, `root` is defined as `root = \lambda n . n SOME (\lambda x . x)`.

The solution we have sketched is consistent with the approach that we have advocated in [2]. In particular, it satisfies our coherence principle. Indeed, all the following expressions provide the same semantic interpretation:

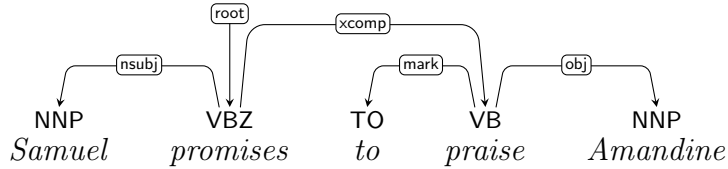
```

root (det (acl:relcl MAN (warg2 nsubj (obj PRAISES AMANDINE) WHO)) A)
root (det (acl:relcl MAN (warg1 obj (warg2 nsubj PRAISES WHO) AMANDINE)) A)
root (acl:relcl (det MAN A) (warg2 nsubj (obj PRAISES AMANDINE) WHO))
root (acl:relcl (det MAN A) (warg1 obj (warg2 nsubj PRAISES WHO) AMANDINE))

```

3 Control verbs and open clausal complements

Open clausal complements are clauses that are missing their own subject and that are typically governed by a control verb or a raising verb.² The dependency relation between a control verb and its open clausal complement is marked by the `xcomp` relation, like in the following example:



The puzzle here is that the `nsubj` dependency relation must provide a subject not only for *promises* but also for *praise*.

The solution we propose is based on a new interpretation of verb phrases that is obtained by type-raising the type of events not on `t` but on `e → t`:

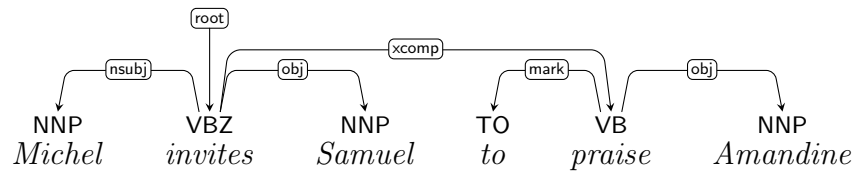
$$VP = (\mathbf{v} \rightarrow \mathbf{e} \rightarrow \mathbf{t}) \rightarrow \mathbf{e} \rightarrow \mathbf{t}$$

This idea, the details and rationale of which would be too long to develop in this extended abstract, leads to the following interpretation of `xcomp` together with a revisited interpretation of `nsubj` to be used for control verbs:

$$\begin{aligned}
\text{xcomp} &= \lambda v c p x. v (\lambda e y. (\mathbf{topic} e (c (\lambda f z. \mathbf{agent} f z) y)) \wedge (p e y)) x \\
\text{nsubj} &= \lambda v n p x. n \text{ SOME } (\lambda x. x) (\lambda y. v (\lambda e z. (\mathbf{agent} e y) \wedge (p e y)) y)
\end{aligned}$$

But this is not the end of the story, because it is not always the subject of the control verb that provides the open clausal complement with a semantic subject. It can also be the object, as the following sentence illustrates:

² An open clausal complement may also be the governee of an adjective.



It is even the case that a same control verb admits both constructions:

Michel wants to praise Amandine
Michel wants Samuel to praise Amandine

Our solution can account for these phenomena by relying on a kind of ad hoc polymorphism, which we discuss in the next section.

4 Ad hoc polymorphism and typing

The semantic developments we have outlined in the previous sections make it necessary to assign several different interpretations to the same dependency relation. For instance, we have given two interpretations to **nsubj**, and these two interpretations need to co-exist. This need for some kind of polymorphism can be satisfied by refining the typing system used at the syntax-semantics interface level. In a full version of this paper, we will propose and discuss several ways of doing so, explaining how they corresponds to encoding notions of lexical semantics in the typing systems.

5 Conclusions

We have shown how dependency relations involving implicit semantic arguments can be treated along the lines we developed in [2]. We argue that this helps to demonstrate of the feasibility and adequacy of the formal theory of dependency semantics, whose basic principles we laid out in [2].

References

1. H.B. Curry and R. Feys. *Combinatory Logic, Vol. I*. North-Holland, 1958.

2. Ph. de Groote. Deriving formal semantic representations from dependency structures. In D. Bekki, K. Mineshima, and E McCready, editors, *19th International Conference, LENLS19, Tokyo, Japan, November 19–21, 2022, Revised Selected Papers*, volume 14213 of *Lecture Notes in Computer Science*. Springer, 2023.
3. J. Y. Findlay and D. T. T. Haug. How useful are enhanced Universal Dependencies for semantic interpretation? In *Proceedings of the Sixth International Conference on Dependency Linguistics (Depling, SyntaxFest 2021)*, pages 22–34. Association for Computational Linguistics, 2021.
4. J. Y. Findlay, S. Salimifar, A. Yıldırım, and D. T. T. Haug. Rule-based semantic interpretation for Universal Dependencies. In *Proceedings of the Sixth Workshop on Universal Dependencies (UDW, GURT/SyntaxFest 2023)*, pages 47–57. Association for Computational Linguistics, 2023.
5. D. T. T. Haug and J. Y. Findlay. Formal semantics for dependency grammar. In *Proceedings of the Seventh International Conference on Dependency Linguistics (Depling, GURT/SyntaxFest 2023)*, pages 22–31. Association for Computational Linguistics, 2023.
6. D. T. T. Haug and M. Gotham. Glue semantics for universal dependencies. In *Proceedings of the LFG-conference*, pages 208–226. CSLI Publications, 2021.
7. Igor Mel’čuk. *Dependency Syntax: Theory and Practice*. State University of New York Press, 1988.
8. R. Montague. *Formal Philosophy: selected papers of Richard Montague, edited and with an introduction by Richmond Thomason*. Yale University Press, 1974.
9. J. Nivre, M.-C. de Marneffe, F. Ginter, Y. Goldberg, J. Hajič, C. D. Manning, R. McDonald, S. Petrov, S. Pyysalo, N. Silveira, R. Tsarfaty, and D. Zeman. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pages 1659–1666, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA).
10. S Reddy, O. Täckström, M. Collins, T. Kwiatkowski, D. Das, M. Steedman, and M. Lapata. Transforming dependency structures to logical forms for semantic parsing. *Transactions of the Association for Computational Linguistics*, 4:127–140, 2016.
11. S Reddy, O. Täckström, S. Petrov, M. Steedman, and M. Lapata. Universal semantic parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 89–101. Association for Computational Linguistics, 2017.
12. L. Tesnière. *Éléments de syntaxe structurale*. Klincksieck, Paris, 1959.