Investigating Different Graph Representations of Semantics

Petter Ericson

Department of Computing Science
Umeå University, Sweden
901 87 Umeå
pettter@cs.umu.se

Abstract

Combinatory Categorial Grammar is a generic approach to the mechanical understanding of language, where movement is minimised in favour of using combinators such as **B** (composition) and **T** (type lifting) to clearly define in which ways various constituents can refer to each other. Taking the tree languages induced by the syntactic derivations and connecting the various leaves linked through the semantics, one ends up with a class of graph languages. The present work aims to point out promising avenues of research in order to investigate this class, specifically in terms of similarities with other graph-based semantic representations, such as Abstract Meaning Representations (AMR), and furthermore what graph generating or recognising formalism would be most suitable to define the class characteristics.

1. Introduction

There is a long history in computational linguistics of using graph representations for modelling semantics. However, as with many other linguistic tasks, the computational power to make reasonable experiments (using large corpora and retaining relatively short response times) has only recently become widely available, spurring research in new directions.

In this extended abstract, we focus on semantic representations that does not adhere too closely to the specific wording of a sentence. Notably, *"Murder!", she said*, and *She said"Murder!"* would receive the same representation in both of the formalisms we have chosen. We will briefly describe the formalisms, and propose a number of research questions from a formal grammar perspective.

2. Abstract Meaning Representation

A relatively recent development in semantic representation, the research effort centred around Abstract Meaning Representation (AMR) (Banarescu et al., 2013) is data-driven and focused on modelling English-language semantics, specifically. In short, an AMR is an edge-labelled graph representing the meaning of a sentence, where the labels are derived from the PropBank framesets. A major achievement of the project is the variety of real-world manually annotated data available for download. (Knight et al., 2014) An example is shown in figure 1.

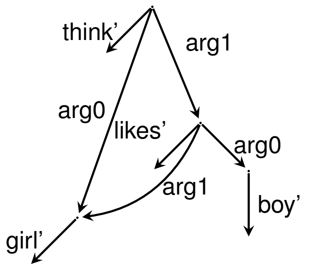

Figure 1: AMR for *"The girl thinks the boy likes her"*

Though there is no specified algorithmic way of constructing a sentence from a graph, or vice versa, there is extensive documentation (Banarescu et al., 2012) on what tags are appropriate representations for what concepts, and how they are to be combined to express meaning. There have been recent attempts to formalise the graph languages defined by AMR, however, notably using various restrictions on Hyperedge Replacement Grammars (HRG) (Björklund, Drewes and Ericson, 2016).

3. Combinatory Categorial Grammar

The development of Combinatory Categorial Grammar (CCG) (Steedman and Baldridge, 2011) began in the mid-eighties, and has progressed roughly parallel to the Minimalist program of Chomsky et. al. In particular, CCG aims to model the universal grammar, but to do so in a theory that derives both syntactic and semantic information in the same operation. This is done through assigning each word in the lexicon one or (commonly) several *categories*, which are essentially types or syntactic constituents that define not only what kind of thing the word is by itself, but also how it interacts with other types. A simple intransitive verb may for example have the category S/NP, meaning it takes an NP from the right and produces an S. Additionally, the lexical entry has an associated logical form that defines the semantics of the entry, for example $\lambda x.px$. The lambda terms of the various constituents get applied during the course of derivation, yielding a final logical form describing the semantics of the sentence, modelled on, for example, Discourse Representation Theory.

What sets CCG apart, and what makes it powerful, is that categories can combine in some rather surprising ways, essentially treating 'non-standard' parts of the sentence, such as 'the boy likes' as constituents for the purpose of derivation. This makes it possible to defer resolution of various semantic arguments until the actual value is available (through some other branch of the derivation tree), while still deriving the semantic and syntactic structures in an integrated manner. There are a number of complications that require more space than afforded in this extended abstract to explain the exact procedure in a satisfactory manner, but the simplified derivation in figure 2 may help give an intu-

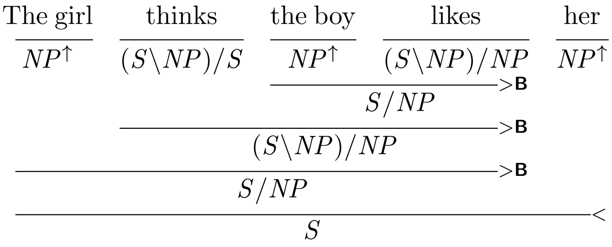

Figure 2: A simplified CCG derivation

ition of the process.[1]

3.1 Working from Discourse Representation Structures

The logical forms produced by CCG can themselves be used to construct a semantic graph, in various ways. An obvious way is to first construct the tree induced by the logical form, and then link any leaves that refer to the same variable, giving rise (for the example sentence) to a graph that is almost identical to figure 1. However, there are also many potential translations, for example converting to a Semantic Web-style RDF document, or using the framework of Conceptual Graphs.

Whichever translation is chosen, the question then becomes how the restrictions defined for CCG and their resulting logical forms impacts the resulting graph languages, in formal terms, and what kind of graph formalism would be the most useful for further investigating this class, and yield useful and efficient algorithms for tasks such as graph parsing, generation and transformation.

3.2 Incorporating derivation trees

An alternative approach to working exclusively from the finished logical form is to also incorporate the CCG derivation tree. This could allow for investigating the CCG generative power in more detail. In particular, the original sentence can be recovered and used for analyses, something which is, as noted, in general not possible with either the logical forms or AMR. Additionally, the direct correspondence between various parts of the logical form and the sentence could also be directly and clearly indicated, which may be useful for further processing (e.g. using word order to identify the principal theme of a sentence).

Again, the formal questions would center around proper formalisms, expressiveness, restrictions, and complexity results. In figure 3, we show an example of a similar 'direct' translation to a graph as in the previous section.

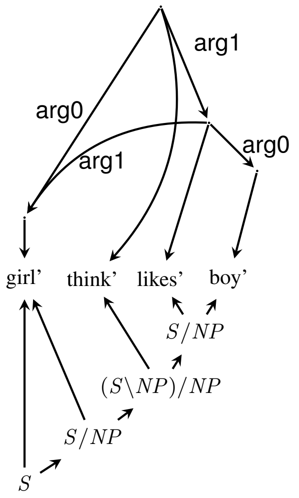

Figure 3: Graph sketch incorporating CCG structure

3.3 Word-word dependency graphs

Another approach to CCG graphs was taken by Hockenmaier and Steedman (2007) in translating the Penn Treebank into CCGbank, where the phrase structure syntax trees have been transformed into dependency graphs. However, as this approach uses a tree rather than a string as its input, it is not anticipated to be used in the present work.

3.4 Research questions

A final aim for this program is to not only define and discuss a graph formalism for CCG-induced semantic graphs, but to compare and contrast the class with various candidates for AMRs, especially with an eye to useful unification. Ideally, we would like to find a usable translation mechanism between AMR graph and these graphs, which could then allow for using the AMR bank for CCG machine learning tasks, and vice versa. Before we reach a point where that task is viable however, there are a number of questions to if not solve, then at least investigate rather more deeply:

- Under standard CCG restrictions, what is the class of logical forms that can be derived? What are its formal properties? Does it require further restrictions to be practically useful?
- Can CCG logical form be implemented using the AMR concepts directly or do we need another translator layer?

Additionally, it is likely that investigating and formalizing modern variants of CCG, as used in practise, will constitute at least part of the output of the proposed program.

4. Related work

Recent work by Artzi et. al. (2015) into practical AMR parsing using CCG as a 'middle step' is a typical example of the intended applications of the results of the theoretical work proposed in this abstract. However, as the work is highly practical, the precise definition of the AMR-CCG interface remain underspecified, and its formal properties even moreso.

5. Acknowledgements

The very helpful comments given by the anonymous reviewers are gratefully acknowledged.

[1] The up-arrow indicates 'type-raising', where $Y^{\uparrow}$ can denote a category like $X/(X\backslash Y)$. The **B** stands for the composition combinator, which can derive a category such X/Y from X/Z and Z/Y. In the first derivation in figure 2, take NP for Y, S for X and $(S\backslash NP)$ for Z

References

Yoav Artzi, Kenton Lee, and Luke Zettlemoyer. Broad-coverage ccg semantic parsing with amr. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1699–1710, 2015.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract meaning representation (amr) 1.0 specification. In *Parsing on Freebase from Question-Answer Pairs. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Seattle: ACL*, pages 1533–1544, 2012.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract meaning representation for sembanking. In *Proc. 7th Linguistic Annotation Workshop, ACL 2013*, 2013.

Henrik Björklund, Frank Drewes, and Petter Ericson. Between a rock and a hard place–uniform parsing for hyperedge replacement dag grammars. In *International Conference on Language and Automata Theory and Applications*, pages 521–532. Springer, 2016.

Kevin Knight et. al. Abstract meaning representation (amr) annotation release 1.0 ldc2014t12. Web Download, Philadelphia: Linguistic Data Consortium, 2014.

Julia Hockenmaier and Mark Steedman. Ccgbank: a corpus of ccg derivations and dependency structures extracted from the penn treebank. *Computational Linguistics*, 33(3):355–396, 2007.

Mark Steedman and Jason Baldridge. Combinatory categorial grammar. *Non-Transformational Syntax: Formal and Explicit Models of Grammar. Wiley-Blackwell*, 2011.