Order-Preserving DAG Grammars: Parsing, Complexity and Learning

Henrik Björklund

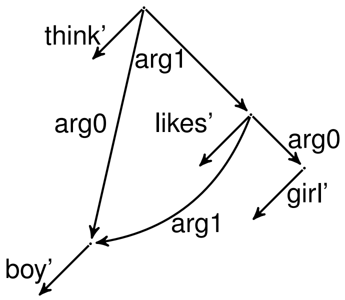
Joint work with
Johanna Björklund, Frank Drewes, and Petter Ericson

Umeå University

Highlights, September 23, 2017

Motivation: Semantic graphs

One type of semantic grapsh: Abstract Meaning Representations



AMR for "the boy thinks that the girl likes him"

Abstract Meaning Representations

Properties of AMRs:

- Directed and acyclic
- Reentrancies (not trees)
- Any number of modifiers (i.e. no fixed rank)
- No formalized grammar

Long term goals

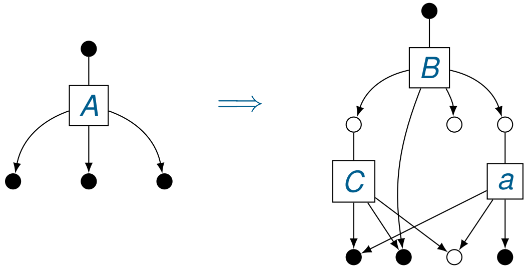
Parallel parsing of natural language sentences, building both syntax trees and semantic graphs.

Long term goals

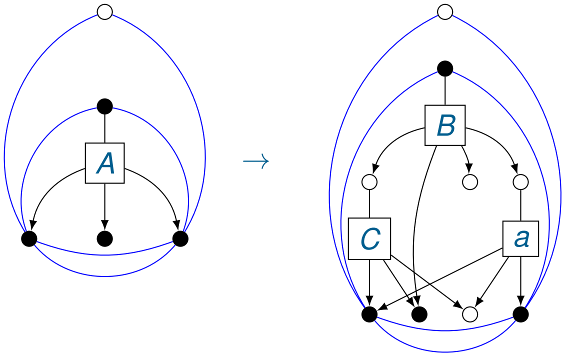
Parallel parsing of natural language sentences, building both syntax trees and semantic graphs.

Trransformation of semantic graphs into natural language sentences.

Hyperedge replacement grammars



Hyperedge replacement grammars



Uniform vs. non-uniform parsing

For database theoreticians: Think data complexity vs. combined complexity

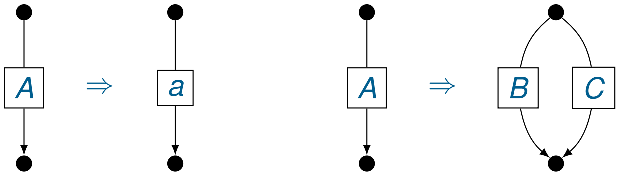
For verification people: Think model complexity vs. combined complexity

Uniform vs. non-uniform parsing

For database theoreticians: Think data complexity vs. combined complexity

For verification people: Think model complexity vs. combined complexity

Consider a grammar where we only have rules of the following forms:



Order-preserving DAG grammars

Graph parsing is hard.

Order-preserving DAG grammars

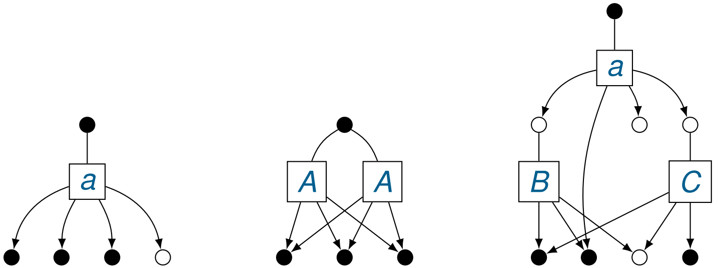
Graph parsing is hard.

To achieve uniform polynomial parsing, we need to heavily restrict the right-hand sides.

Order-preserving DAG grammars

Graph parsing is hard.

To achieve uniform polynomial parsing, we need to heavily restrict the right-hand sides.



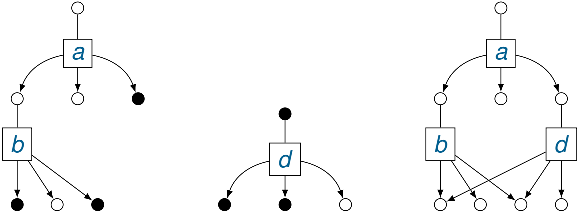
MAT-learning

We develop an algorithm for learning OPDGs from a Minimally Adequate Teacher (Angluin).

The teacher can answer

- equivalence queries (Is this the correct grammar?)
- membership queries (Does this graph belong to the language of the grammar?)

Concatenation



A Myhill-Nerode theorem

Theorem. A DAG language L can be generated by an OPDG if and only if $\equiv_L$ has finite index.
If $\equiv_L$ has finite index, there is a unique minimal unambiguous OPDG for L.

Polynomial time learning

Theorem. An OPDG G can be learned from a MAT in time polynomial in $|G|$ and the combined sizes of the counterexamples provided by the teacher.

The end

Thank you for listening!