Contextual Hyperedge Replacement Grammars for the Generation of Abstract Meaning Representations

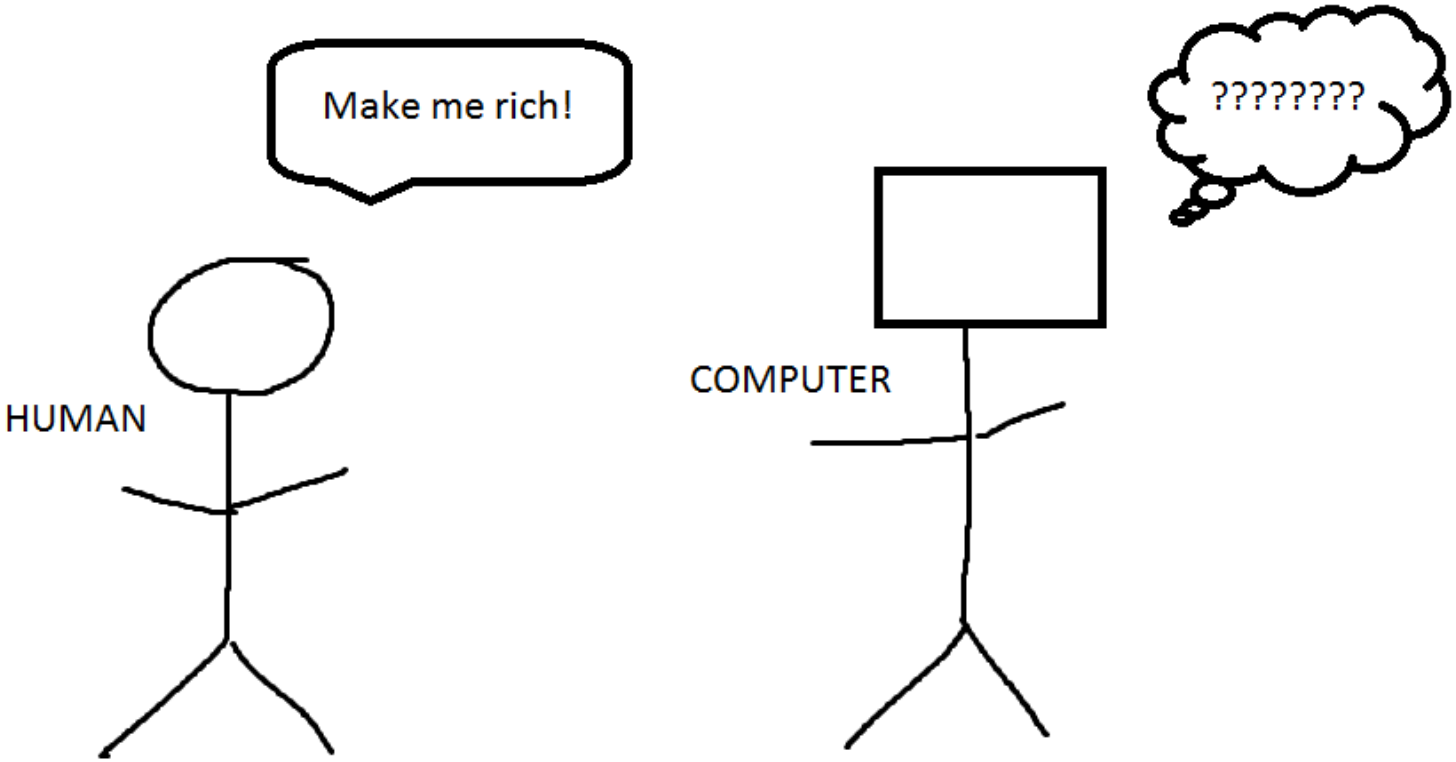
Anna Jonsson

Umeå University

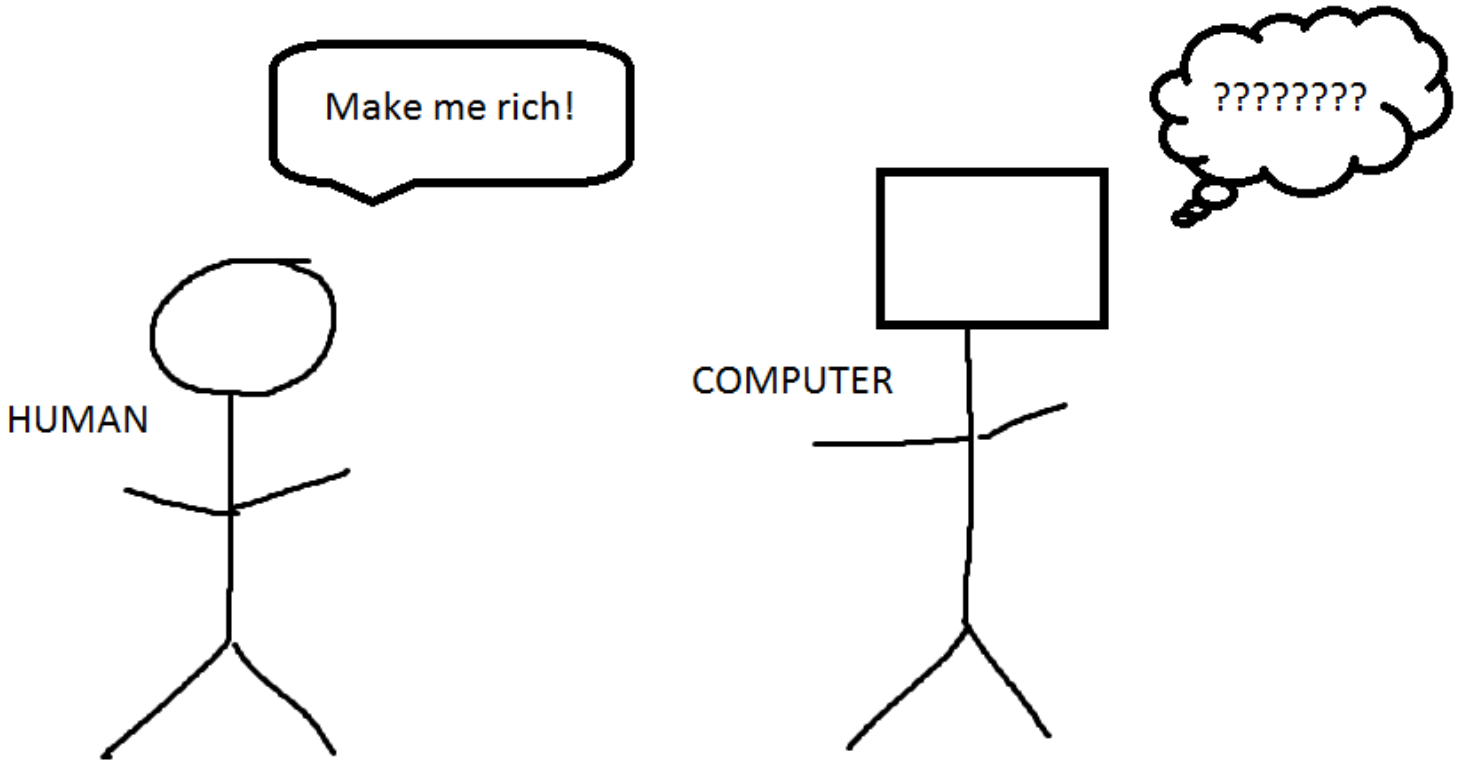
aj@cs.umu.se

January 13, 2017

Aim of natural language processing



Aim of natural language processing



1. Parse
2. Analyse
3. Respond

Natural language in computers

- Natural language is complex – we must use approximations
- A model (e.g. an automaton, a grammar, a neural network) can be used for approximating language
- In, e.g., machine learning and natural language generation

Correct vs. incorrect sentences

What do we mean by “correctness”?

Correct vs. incorrect sentences

What do we mean by "correctness"? Correctness of:

- Syntax – the words and their order
- Semantics – the meaning

Correct vs. incorrect sentences

What do we mean by "correctness"? Correctness of:

- Syntax – the words and their order
- Semantics – the meaning

Examples:

Correct vs. incorrect sentences

What do we mean by "correctness"? Correctness of:

- Syntax – the words and their order
- Semantics – the meaning

Examples:

- "The cat today no have eat."

Correct vs. incorrect sentences

What do we mean by "correctness"? Correctness of:

- Syntax – the words and their order
- Semantics – the meaning

Examples:

- "The cat today no have eat."
  - Syntax: incorrect
  - Semantics: correct

Correct vs. incorrect sentences

What do we mean by "correctness"? Correctness of:

- Syntax – the words and their order
- Semantics – the meaning

Examples:

- "The cat today no have eat."
  - Syntax: incorrect
  - Semantics: correct
- "Colourless green ideas sleep furiously." [Chomsky, 1956]

Correct vs. incorrect sentences

What do we mean by "correctness"? Correctness of:

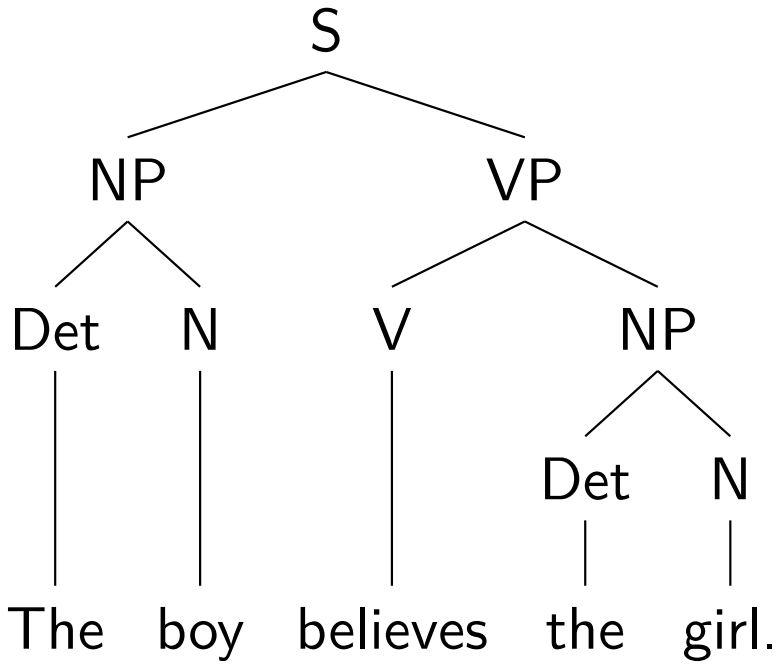
- Syntax – the words and their order
- Semantics – the meaning

Examples:

- "The cat today no have eat."
  - Syntax: incorrect
  - Semantics: correct
- "Colourless green ideas sleep furiously." [Chomsky, 1956]
  - Syntax: correct
  - Semantics: incorrect

Formal representation of syntax

Parse tree:



Formal representation of semantics

- Semantics is more difficult to represent
- There is no equivalence of parse trees
- Do we really need one?

Motivating example: Robot

Input:

"Feed her cat food."

Motivating example: Robot

Input:

"Feed her cat food."

Output:

Motivating example: Robot

Input:

"Feed her cat food."

Output:

1. The robot feeds her cat with food

Motivating example: Robot

Input:

"Feed her cat food."

Output:

1. The robot feeds her cat with food
2. The robot feeds her with cat food

Motivating example: Robot

Input:

"Feed her cat food."

Output:

1. The robot feeds her cat with food
2. The robot feeds her with cat food
3. The robot tries to feed her cat's food with something

Motivating example: Robot

Input:

"Feed her cat food."

Output:

1. The robot feeds her cat with food
2. The robot feeds her with cat food
3. The robot tries to feed her cat's food with something

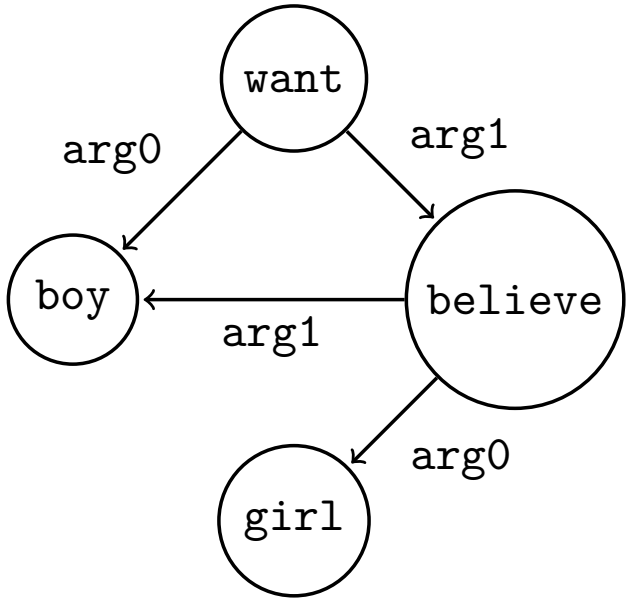
We need an unambiguous semantic representation!

Abstract meaning representation[1]

- Semantic representations of natural language sentences
- Directed graphs
  - Acyclic
  - Rooted
  - Node and edge labelled
- The nodes represent *concepts* ($\approx$ words)
- The edges represent *relations* between the concepts

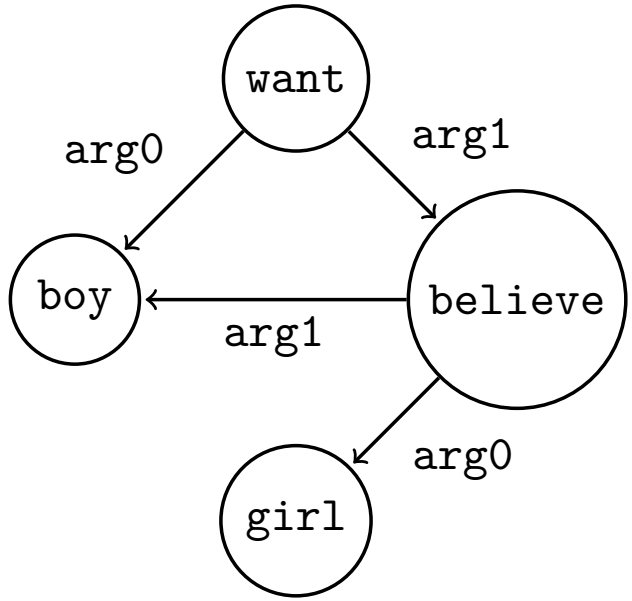
[1] [Banarescu et al., 2013]

Abstract meaning representation[1]



[1] [Banarescu et al., 2013]

Abstract meaning representation[1]



"The boy wants the girl to believe him"

"It is wanted by the boy that the girl has belief in him"

[1] [Banarescu et al., 2013]

Boy-girl AMRs

- From the boy-girl AMR corpus [Braune et al., 2014]
- Domain:
  - Concepts: `boy`, `girl`, `want` and `believe`
  - Relations: `arg0` and `arg1`
- Requirements (for correct meanings):
  - `boy` and `girl` can only occur once each and are leaf nodes
  - There must be at least one occurence of `boy` or `girl`
  - `want` and `believe` can have at most one each of `arg0` and `arg1` as outgoing edges
  - `want` and `believe` can only have `arg1`s as incoming edges

Boy-girl AMRs

- From the boy-girl AMR corpus [Braune et al., 2014]
- Domain:
  - Concepts: `boy`, `girl`, `want` and `believe`
  - Relations: `arg0` and `arg1`
- Requirements (for correct meanings):
  - `boy` and `girl` can only occur once each and are leaf nodes
  - There must be at least one occurence of `boy` or `girl`
  - `want` and `believe` can have at most one each of `arg0` and `arg1` as outgoing edges
  - `want` and `believe` can only have `arg1`s as incoming edges
- **We limit ourselves to boy-girl AMRs**

Correctness of AMRs

- Since AMRs are not (yet) formally defined, we use an informal description (the list of requirements)

Correctness of AMRs

- Since AMRs are not (yet) formally defined, we use an informal description (the list of requirements)
- Framework in which the correctness can be expressed
- As previously mentioned, possible frameworks are automata, grammars and neural networks

Correctness of AMRs

- Since AMRs are not (yet) formally defined, we use an informal description (the list of requirements)
- Framework in which the correctness can be expressed
- As previously mentioned, possible frameworks are automata, grammars and neural networks
- We want a **graph grammar** that generates the **language of all correct AMRs (over the boy-girl domain)**

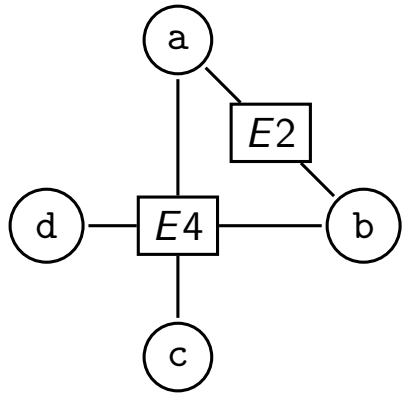
Correctness of AMRs

- Since AMRs are not (yet) formally defined, we use an informal description (the list of requirements)
- Framework in which the correctness can be expressed
- As previously mentioned, possible frameworks are automata, grammars and neural networks
- We want a **graph grammar** that generates the **language of all correct AMRs (over the boy-girl domain)**
- For practical feasibility: Correctness check (parsing) must be done in **polynomial time**

Previous work

- Two graph grammar types were used for building boy-girl grammars [Jonsson, 2016]
  - Predictive top-down parsable grammars [Drewes et al., 2015]
  - Restricted directed acyclic graph grammars [Björklund et al., 2016]
- Both are hyperedge replacement grammars (HRGs) [Drewes et al., 1997] (but with restrictions)
- Both have polynomial-time parsing algorithms (which is not true for all HRGs)

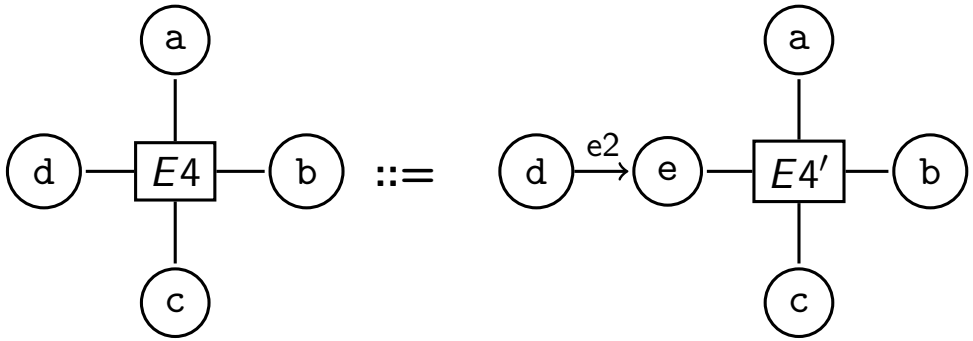
Hyperedges and hypergraphs



- A *hyperedge* is an edge that can be connected to an arbitrary number of nodes; the sequence of connected nodes is called the *attachment* of a hyperedge
- A hyperedge attached to exactly two nodes is equivalent to an ordinary directed edge
- A *hypergraph* is a graph containing hyperedges

Hyperedge replacement[2]

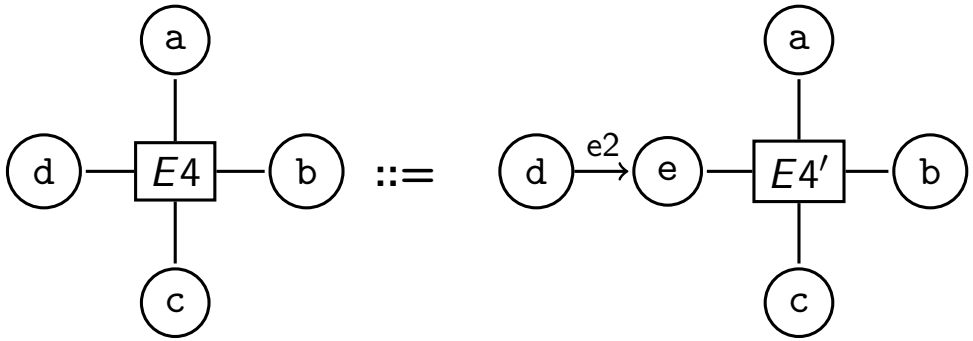
Rule for the *nonterminal* $E4$:



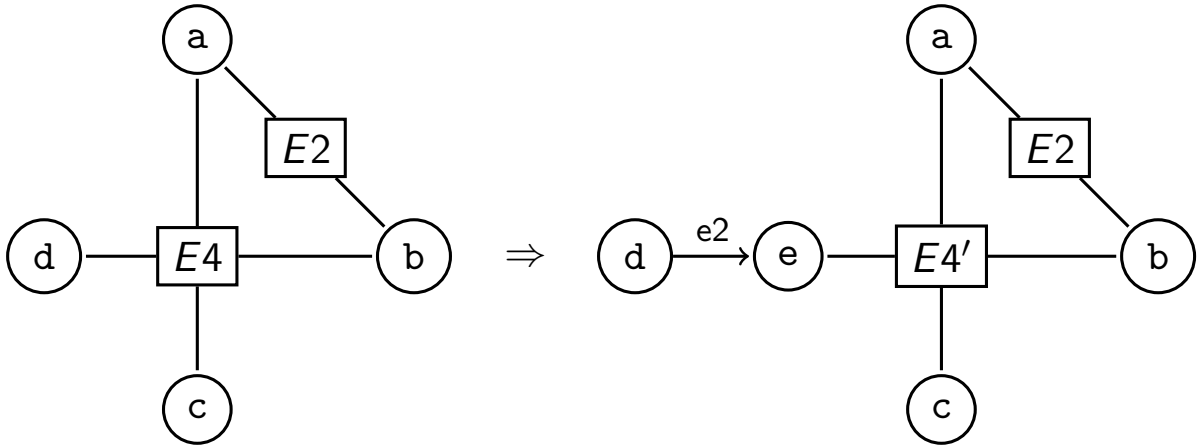
[2] [Drewes et al., 1997]

Hyperedge replacement[2]

Rule for the *nonterminal* $E4$:



Rule application (derivation step):



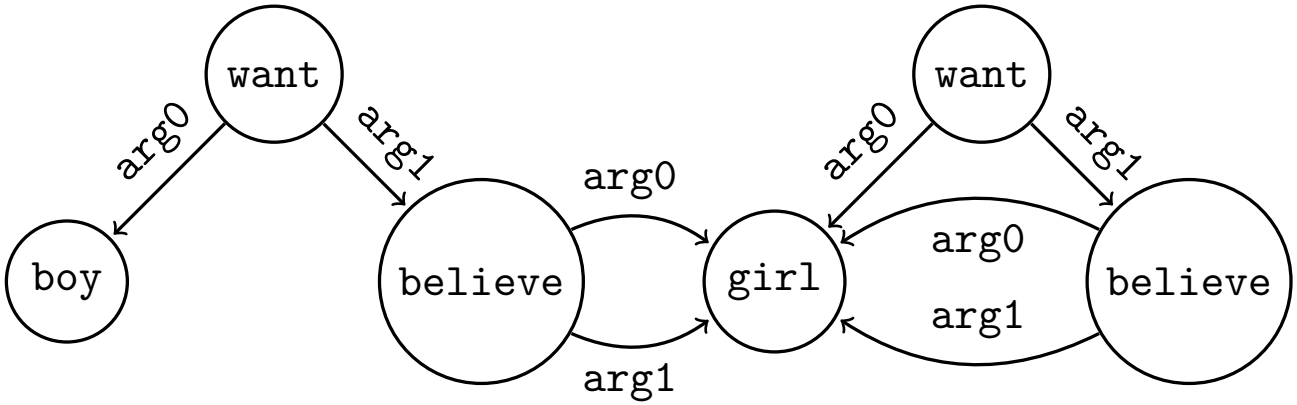
[2] [Drewes et al., 1997]

A boy-girl HRG

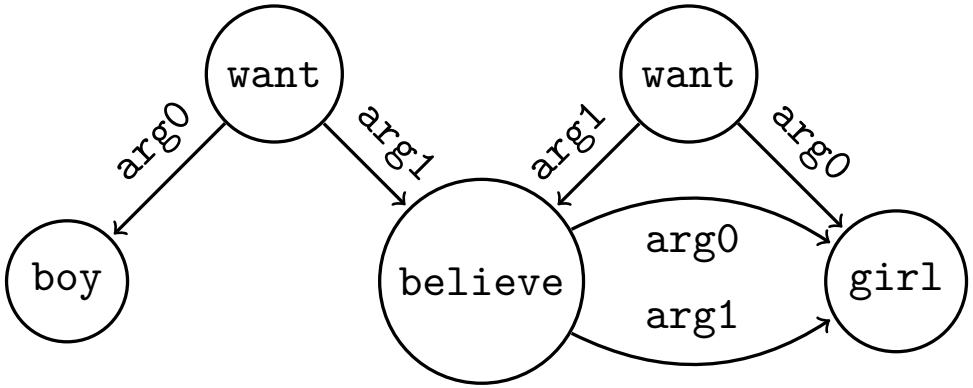
A set of rules specifying hyperedge replacements intended to generate boy-girl AMRs.

Results

Observation (same meaning – two AMRs):



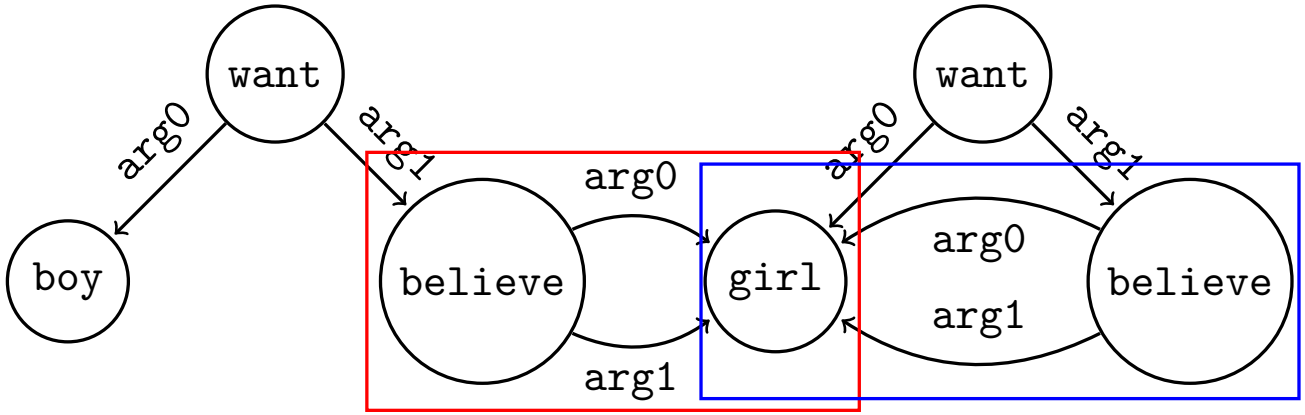
Could be generated



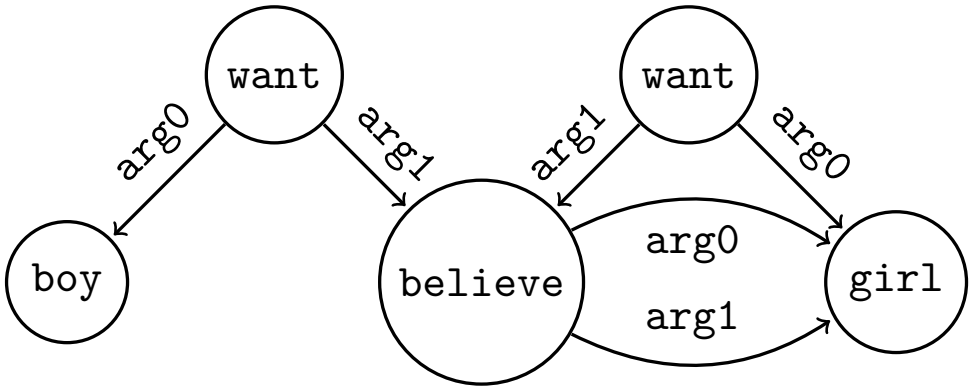
Could not be generated

Minimal AMRs

Here, we say that an AMR that contains no duplicate substructures is minimal



Non-minimal



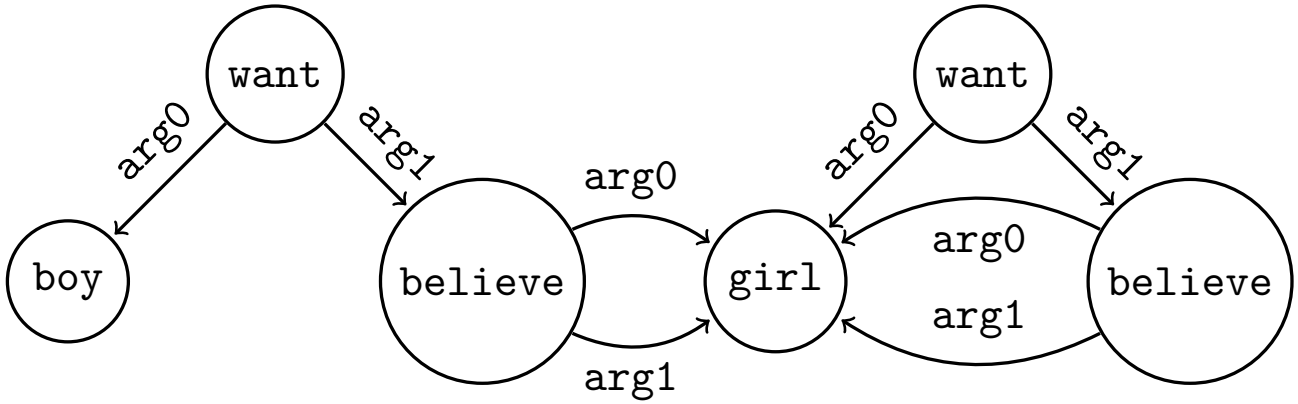
Minimal

Question 1

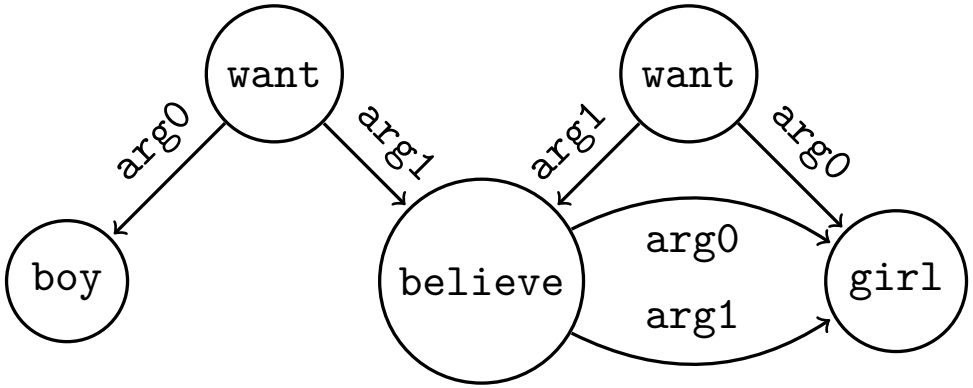
Why does not hyperedge replacement seem to be powerful enough for the generation of minimal AMRs?

Question 1

Why does not hyperedge replacement seem to be powerful enough for the generation of minimal AMRs?



Non-minimal

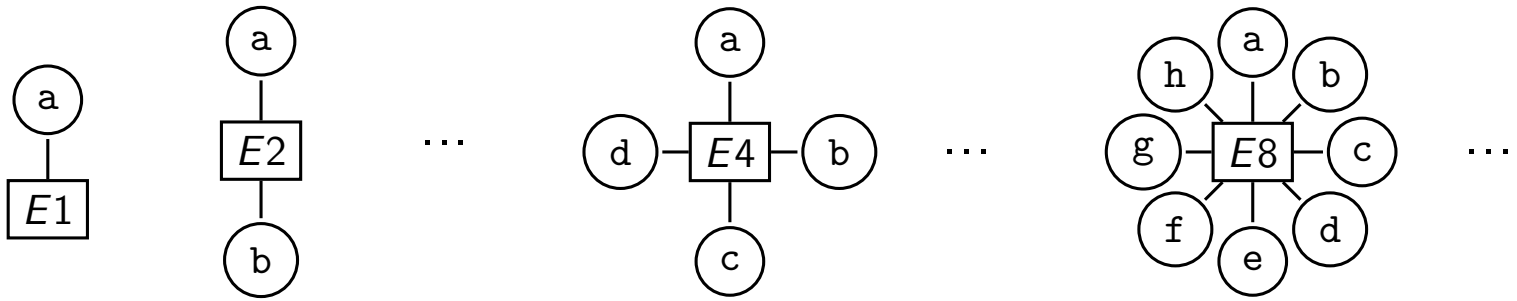


Minimal

Question 1

Why does not hyperedge replacement seem to be powerful enough for the generation of minimal AMRs?

Intuitively, infinitely many nonterminals – and thereby rules – would be needed.

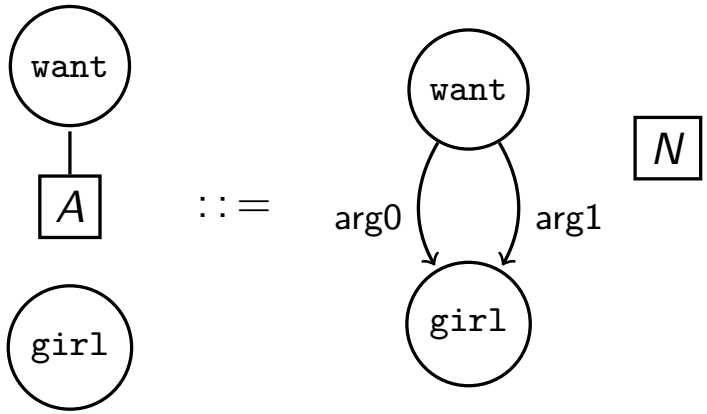


Question 2

Is there another grammar type that can generate minimal AMRs in general?

Contextual hyperedge replacement[3]

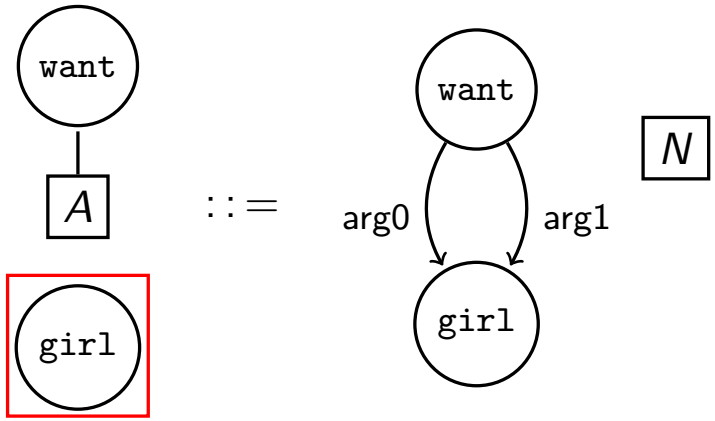
Contextual rule:



[3] [Drewes et al. 2015]

Contextual hyperedge replacement[3]

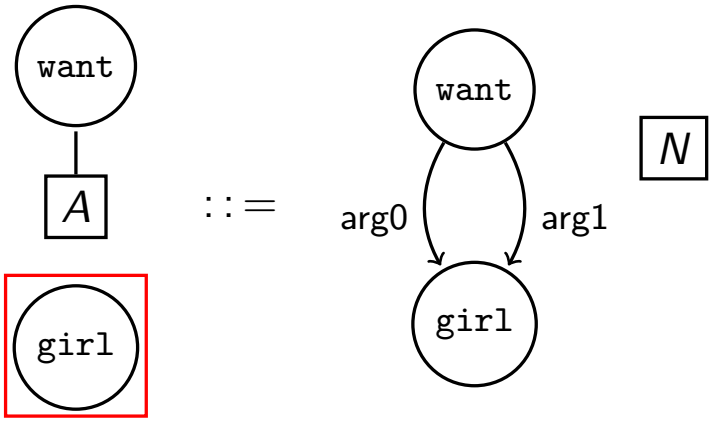
Contextual rule:



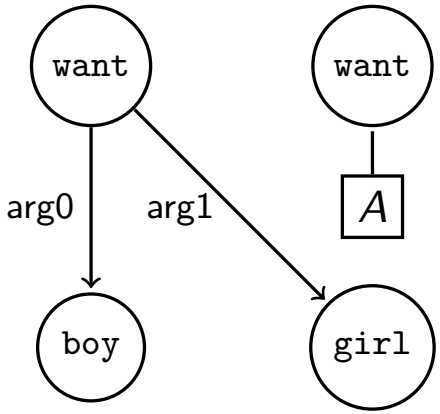
[3] [Drewes et al. 2015]

Contextual hyperedge replacement[3]

Contextual rule:



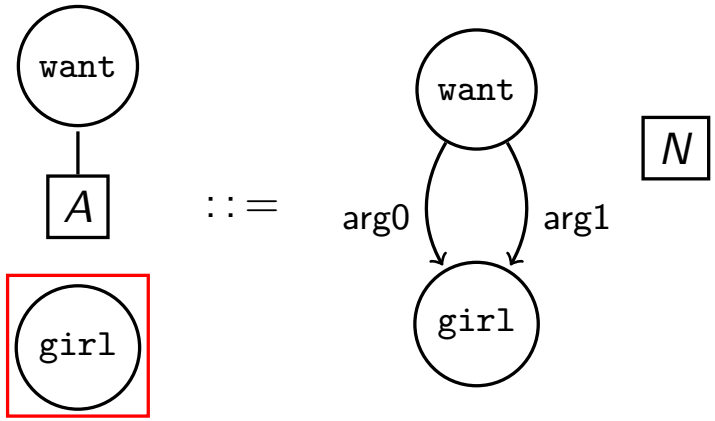
Rule application:



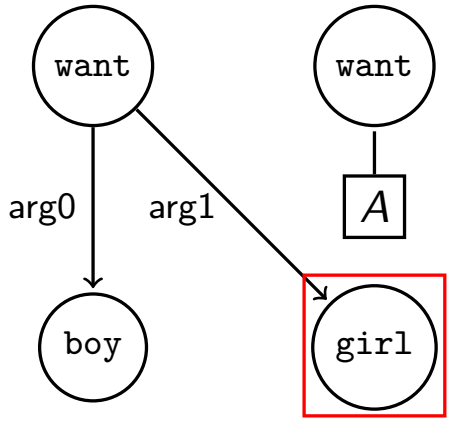
[3] [Drewes et al. 2015]

Contextual hyperedge replacement[3]

Contextual rule:



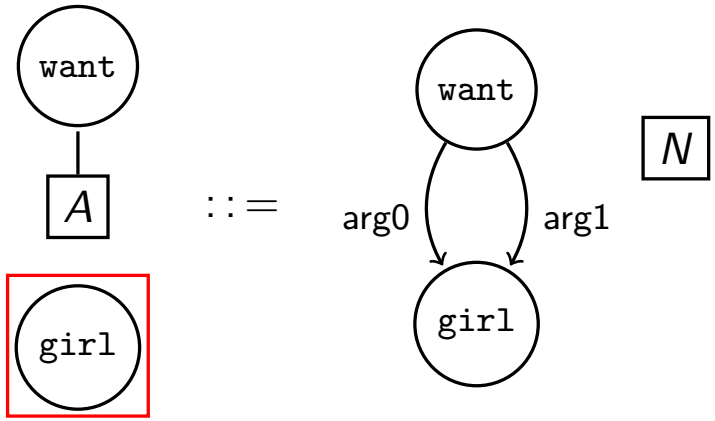
Rule application:



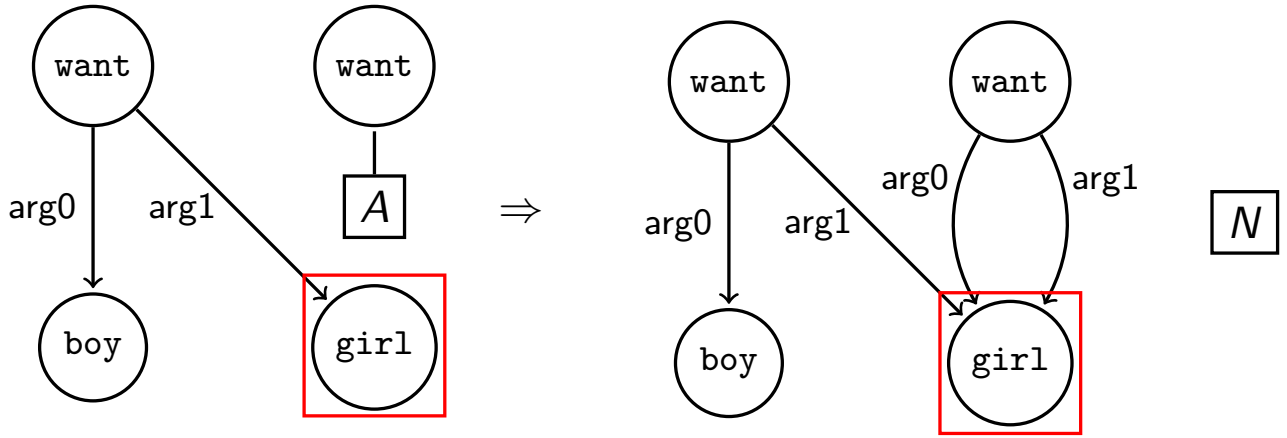
[3] [Drewes et al. 2015]

Contextual hyperedge replacement[3]

Contextual rule:

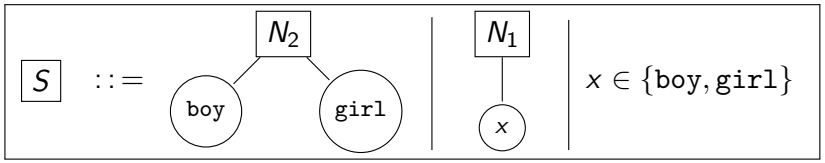


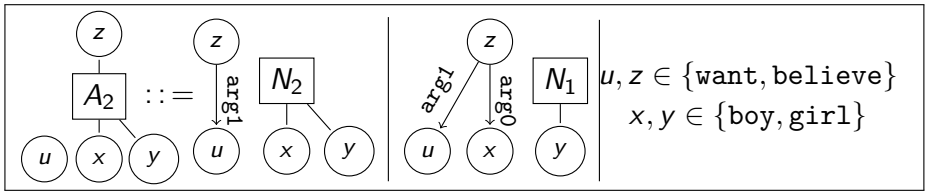
Rule application:

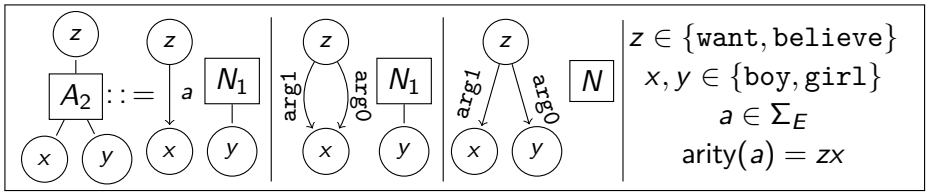


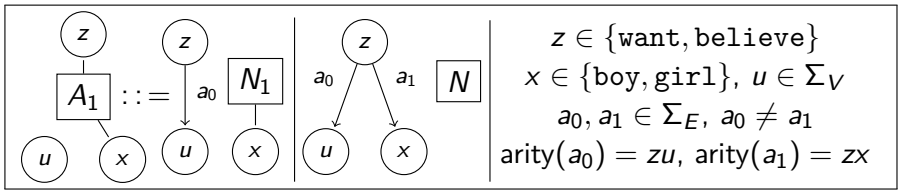
[3] [Drewes et al. 2015]

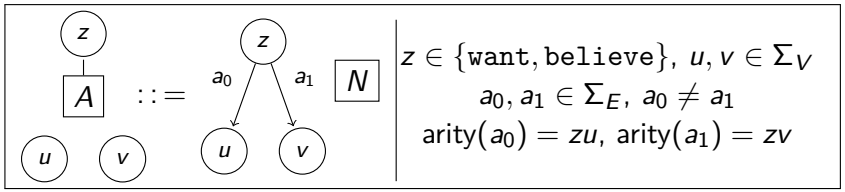
A contextual boy-girl HRG

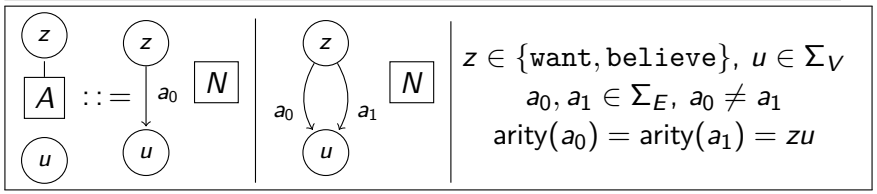


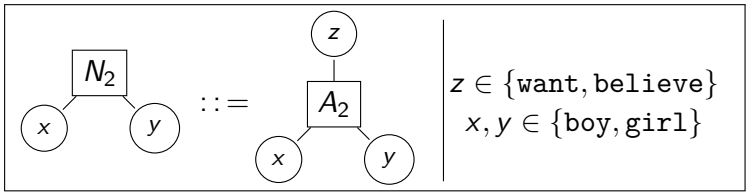


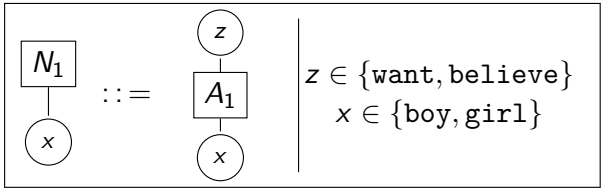


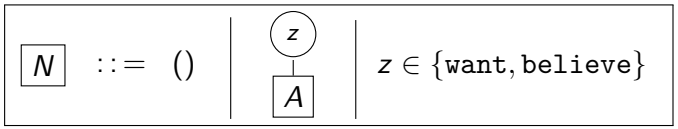


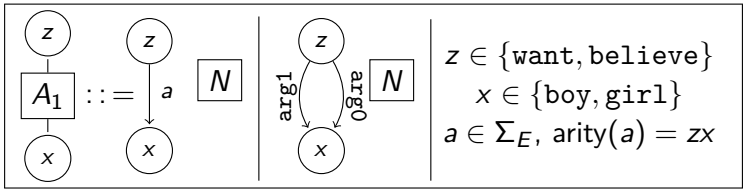


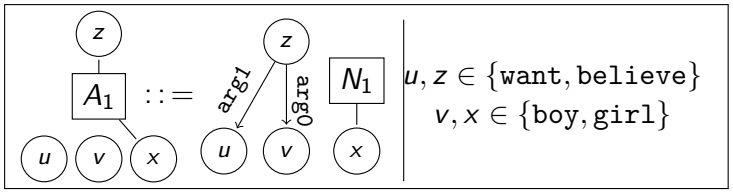




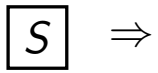




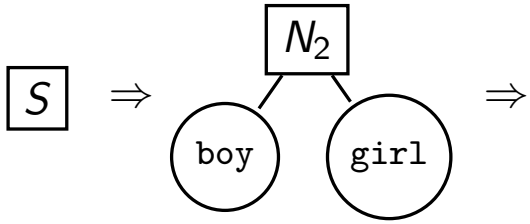


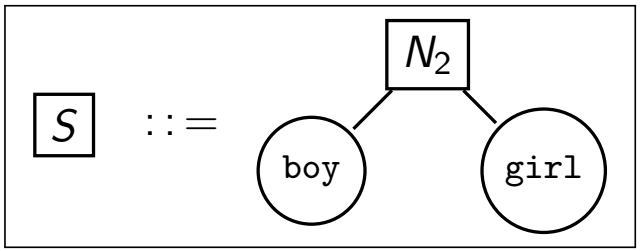


An example derivation

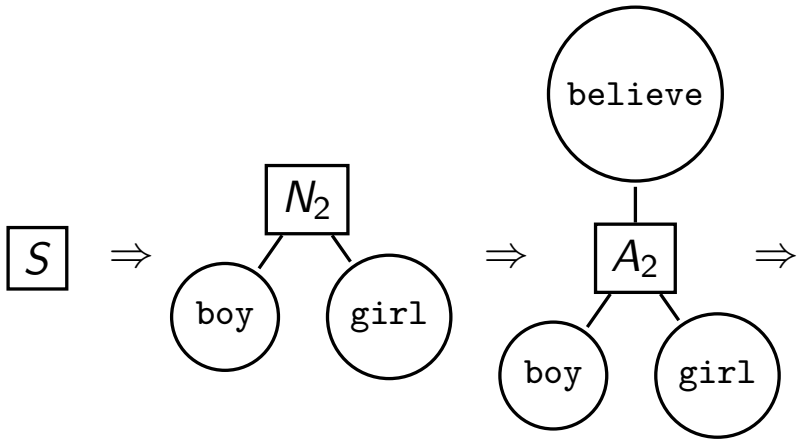


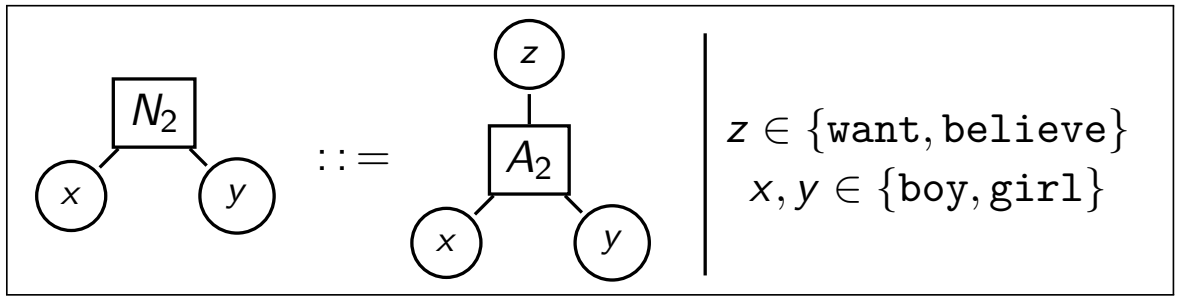
An example derivation



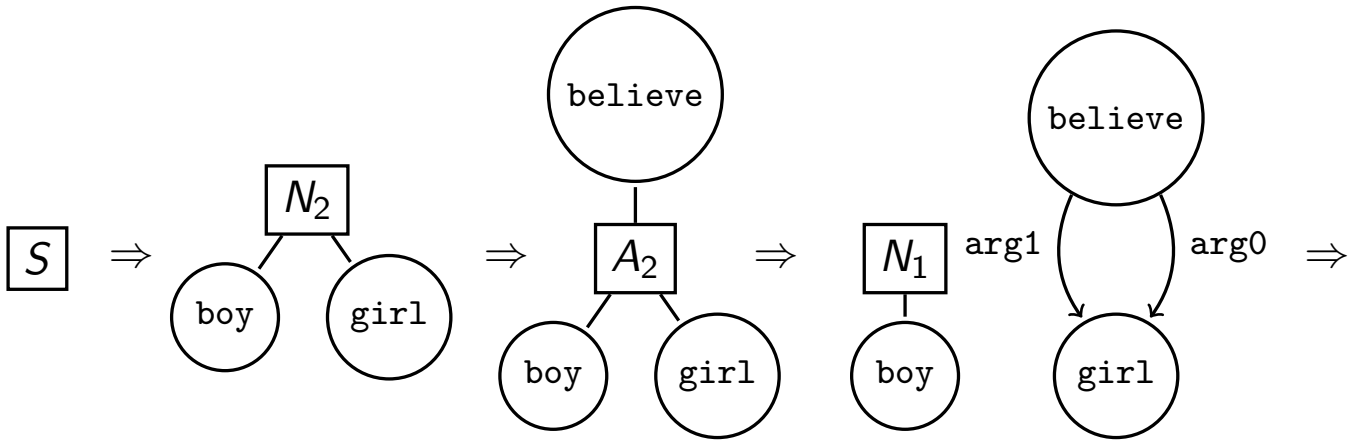


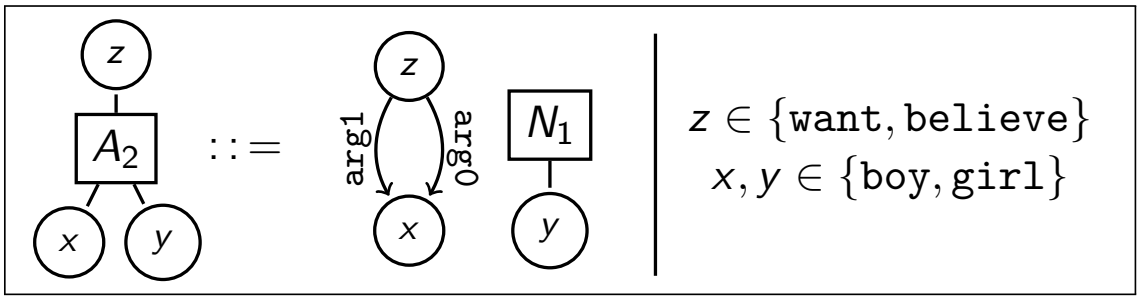
An example derivation



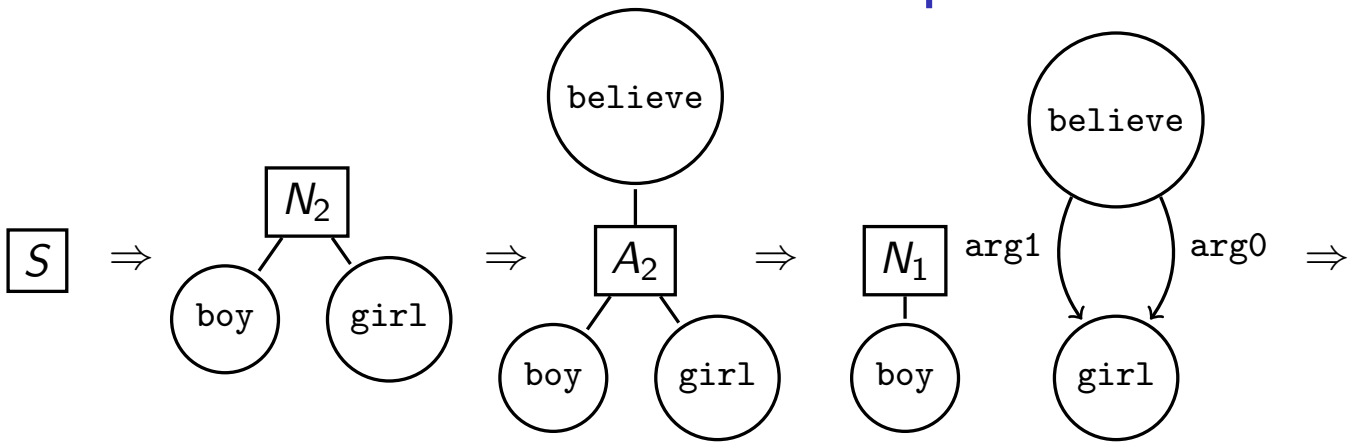


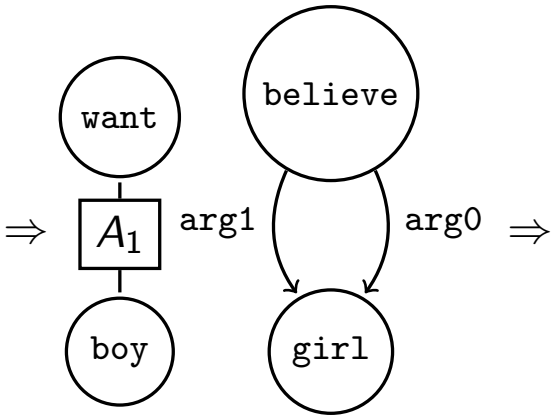
An example derivation

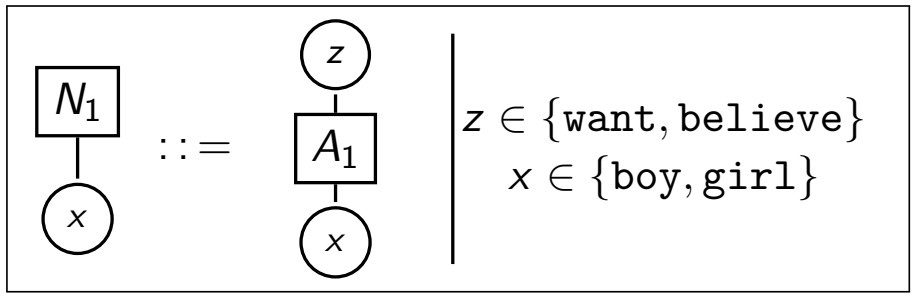




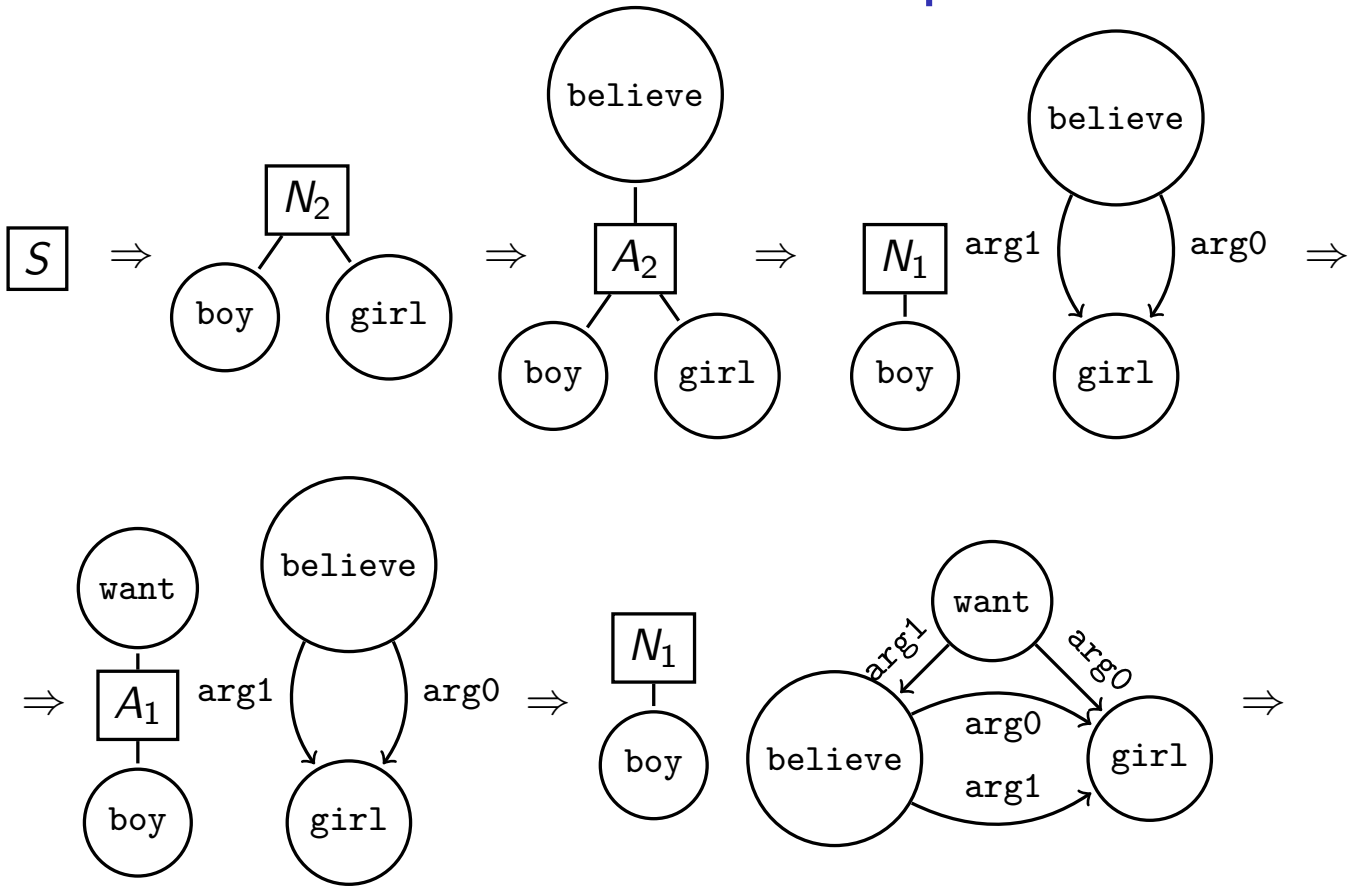
An example derivation

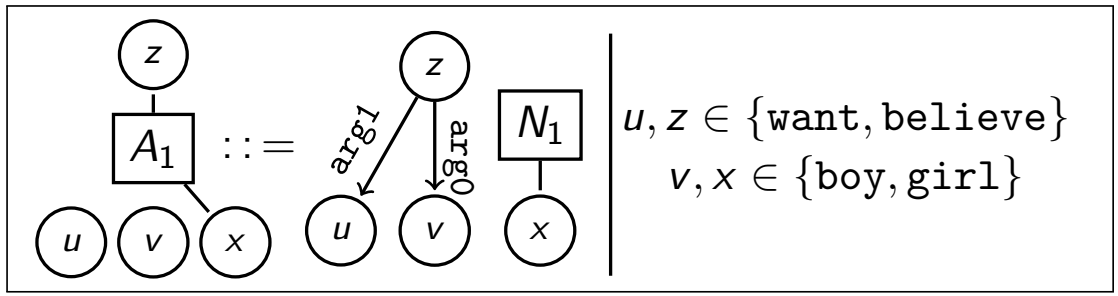




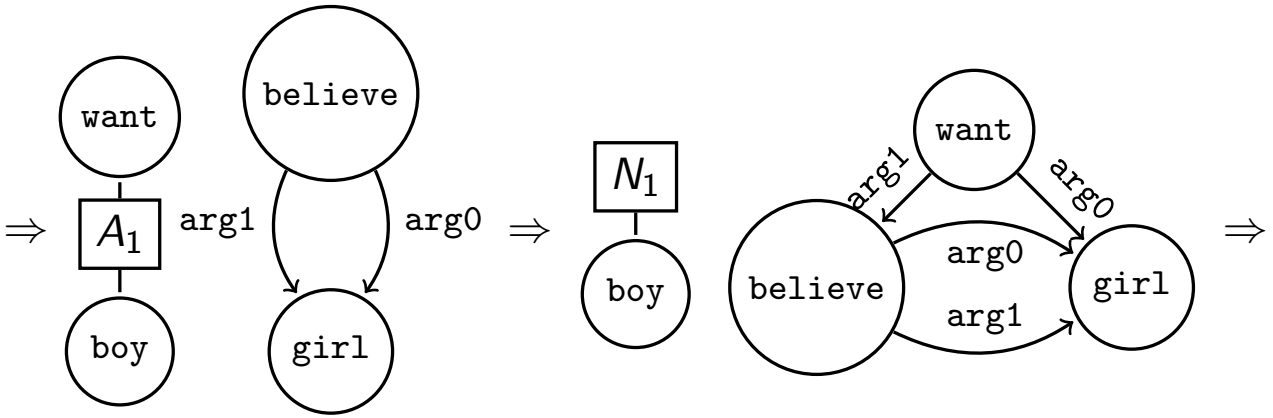


An example derivation

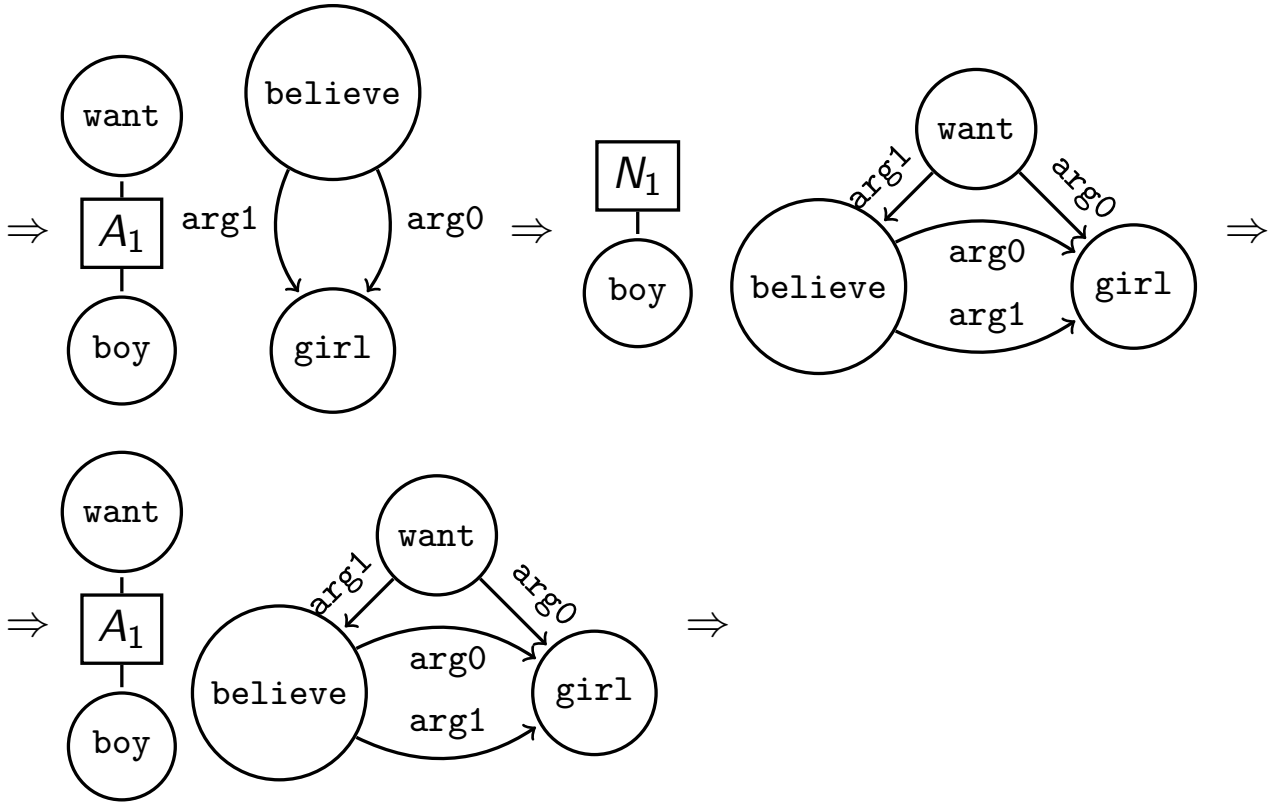




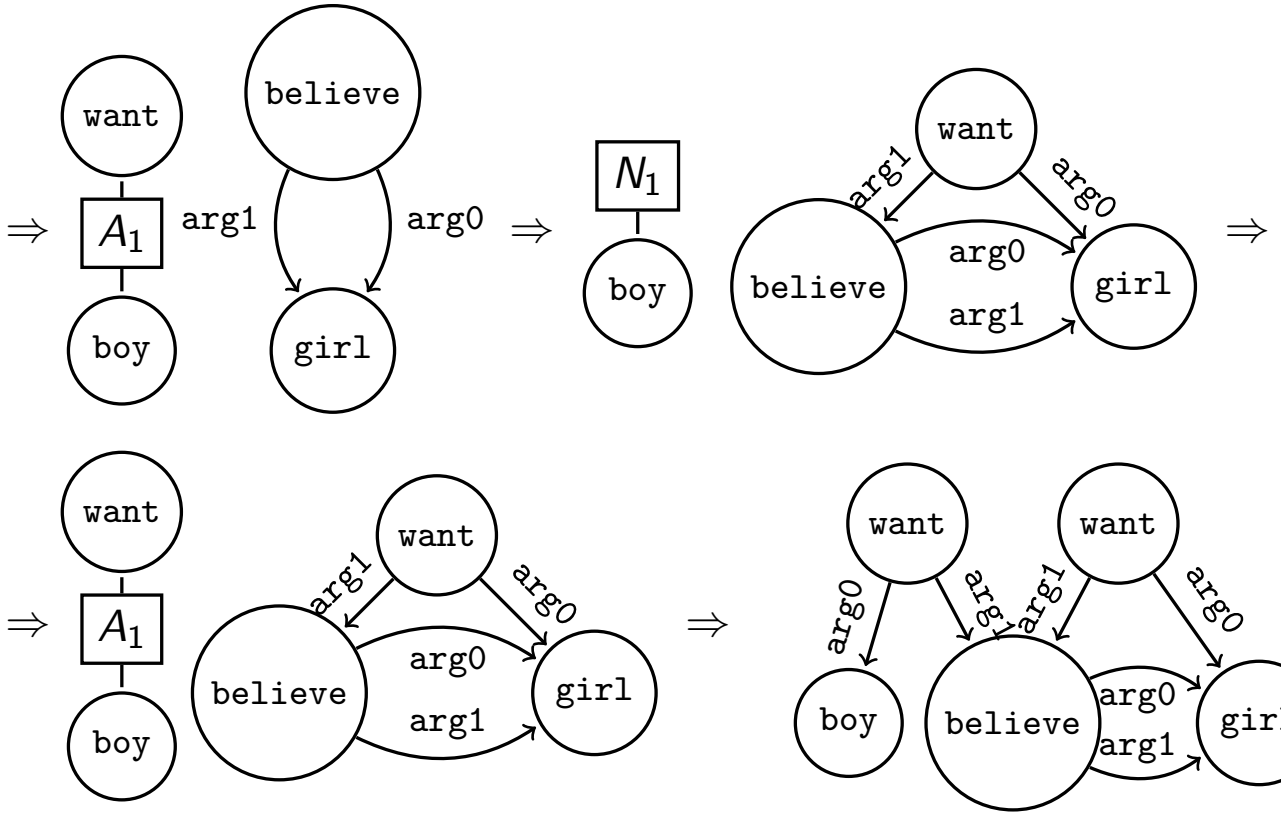
An example derivation

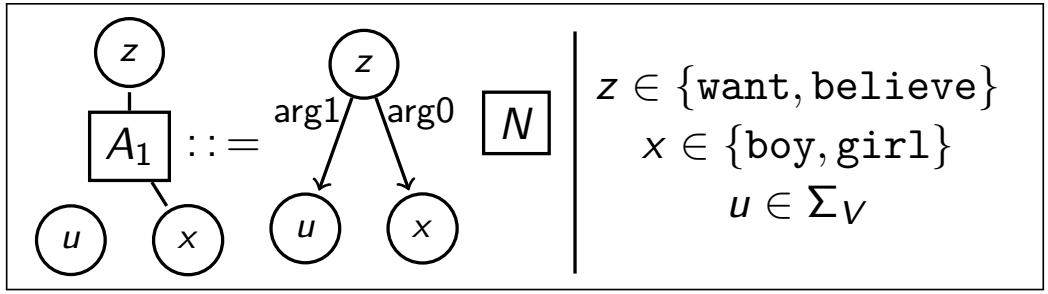


An example derivation



An example derivation





$N ::= ()$

Are CHRGs more suitable than HRGs for AMR generation?

HRGs

Contextual HRGs

Are CHRGs more suitable than HRGs for AMR generation?

HRGs

- Cannot generate minimal AMRs in general

Contextual HRGs

- Can generate minimal AMRs

Are CHRGs more suitable than HRGs for AMR generation?

HRGs

- Cannot generate minimal AMRs in general
- Polynomial-time parsing for subclasses

Contextual HRGs

- Can generate minimal AMRs
- No polynomial-time parsing for subclasses (yet)

Are CHRGs more suitable than HRGs for AMR generation?

HRGs

- Cannot generate minimal AMRs in general
- Polynomial-time parsing for subclasses
- Can be used with node-joining algorithms

Contextual HRGs

- Can generate minimal AMRs
- No polynomial-time parsing for subclasses (yet)

Are CHRGs more suitable than HRGs for AMR generation?

HRGs

- Cannot generate minimal AMRs in general
- Polynomial-time parsing for subclasses
- Can be used with node-joining algorithms

Contextual HRGs

- Can generate minimal AMRs
- No polynomial-time parsing for subclasses (yet)
- Likely to be efficiently parsable subclasses (parsing problem at least not harder than for HRGs)

Conclusion:

- Contextual HRGs can generate minimal boy-girl AMRs
- ... and seemingly also minimal AMRs in general

Future work:

- Proof!
- Is there a unique minimal AMR for each meaning?
- Parsing algorithm for subclasses of CHRG
- Other semantic representations and other formalisms

References I

Chomsky, N. (1956)

Three models for the description of language

Information Theory, IRE Transactions on, 2(3):113–124.

Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., and Schneider, N (2013)

Abstract Meaning Representation for Sembanking

Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse 178 – 186

Braune, F., Bauer, D. and Knight, K. (2014)

Mapping Between English Strings and Reentrant Semantic Graphs

Proceedings of the Ninth International Conference on Language Resources and Evaluation 4493 – 4498

References II

Jonsson, A. (2016)

Generation of Abstract Meaning Representations by Hyperedge Replacement Grammars – A Case Study

Master's thesis, Umeå University

Drewes, F., Hoffmann, B. and Minas, M. (2015)

Predictive Top-down Parsing for Hyperedge Replacement Grammars

Proceedings of the 8th International Conference on Graph Transformation 19 – 34

Björklund, H. and Drewes, F. and Ericson, P.

Between a rock and a hard place — Parsing for Hyperedge Replacement DAG Grammars

10th International Conference on Language and Automata Theory and Applications (LATA 2016)

Drewes, F., Kreowski, H.-J. and Habel, A. (1997)

Hyperedge Replacement Graph Grammars

Handbook of Graph Grammars and Computing by Graph Transformation 95 – 162

References III

Drewes, F. and Hoffman, B. (2015)

Contextual Hyperedge Replacement

Acta Informatica, 52(6):497–524

Thank you for listening!

Summary:

- AMRs are semantic representations of natural language sentences in the form of directed graphs
- Minimal AMRs do not have any repeated substructures
- We want to be able to check the correctness of AMRs
- Previously, hyperedge replacement was used, but minimal AMRs were not fully captured
- Here, contextual hyperedge replacement was used, and even the minimal AMRs could be generated
- Contextual rules cannot be used in practice since there is no efficient parsing algorithm for such grammars, but it is likely that there will be