

Proof Nets and Minimalism

Richard Moot

SIGNES

Workshop in Honour of Alain Lecomte's Birthday
2 – 3 November 2007

Outline

- 1 The Multimodal Lambek Calculus
 - AB Grammars
 - NL Grammars
 - Multimodality

Outline

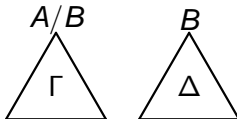
- 1 The Multimodal Lambek Calculus
 - AB Grammars
 - NL Grammars
 - Multimodality
- 2 Minimalist Grammars
 - Merge
 - Comparison with Categorical Grammars
 - Move

Outline

- 1 The Multimodal Lambek Calculus
 - AB Grammars
 - NL Grammars
 - Multimodality
- 2 Minimalist Grammars
 - Merge
 - Comparison with Categorical Grammars
 - Move
- 3 Proof Nets and Minimalist Grammars
 - Ideas
 - Strong and Weak Features
 - Transformations

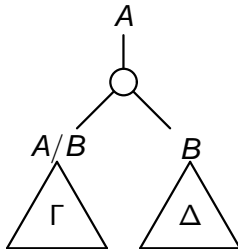
AB Grammars

Logical Rules



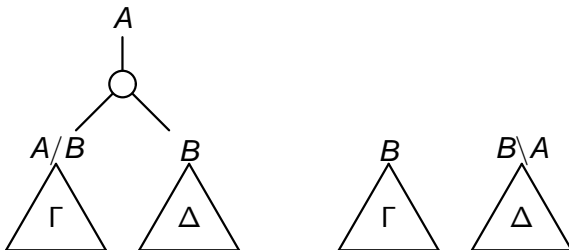
AB Grammars

Logical Rules



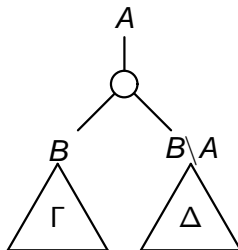
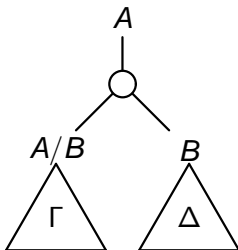
AB Grammars

Logical Rules



AB Grammars

Logical Rules



AB Grammars

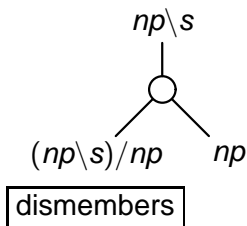
Compiling Away The Lexicon

$(np \backslash s) / np$

dismembers

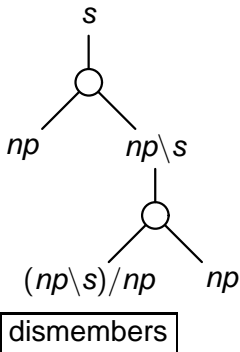
AB Grammars

Compiling Away The Lexicon



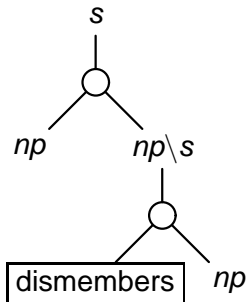
AB Grammars

Compiling Away The Lexicon



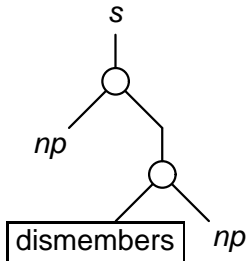
AB Grammars

Compiling Away The Lexicon



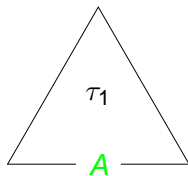
AB Grammars

Compiling Away The Lexicon



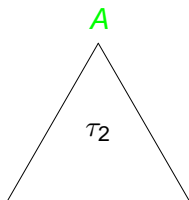
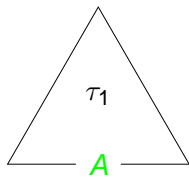
AB Grammars

The Axiom Rule



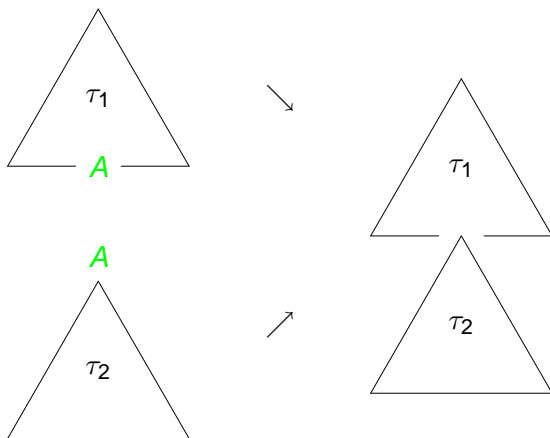
AB Grammars

The Axiom Rule



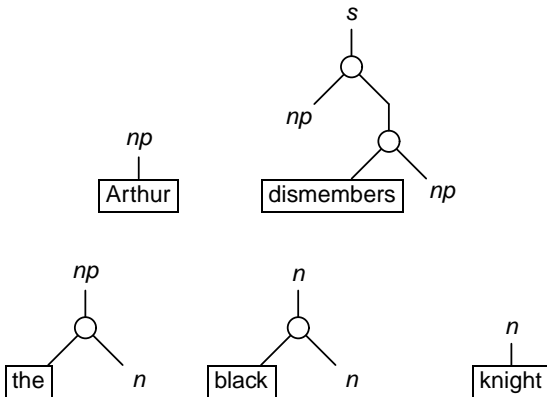
AB Grammars

The Axiom Rule



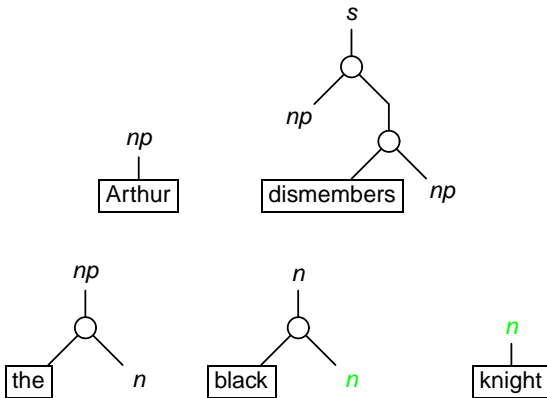
AB Grammars

Example: 'Arthur dismembers the black knight'



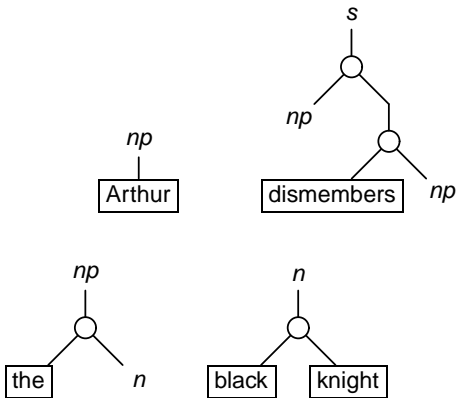
AB Grammars

Example: 'Arthur dismembers the black knight'



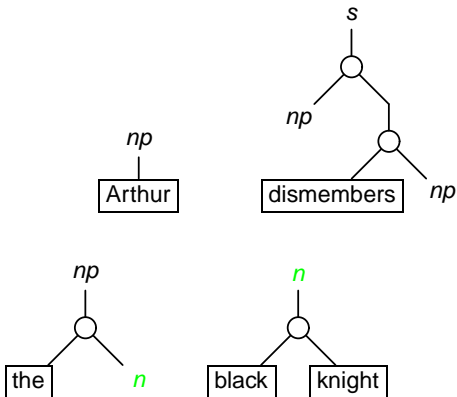
AB Grammars

Example: 'Arthur dismembers the black knight'



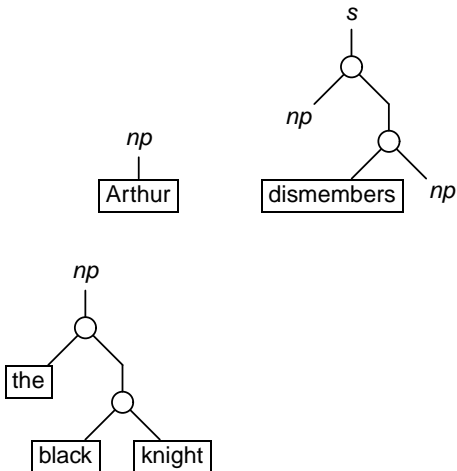
AB Grammars

Example: 'Arthur dismembers the black knight'



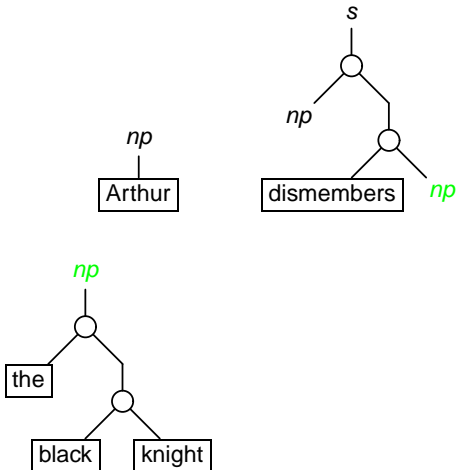
AB Grammars

Example: 'Arthur dismembers the black knight'



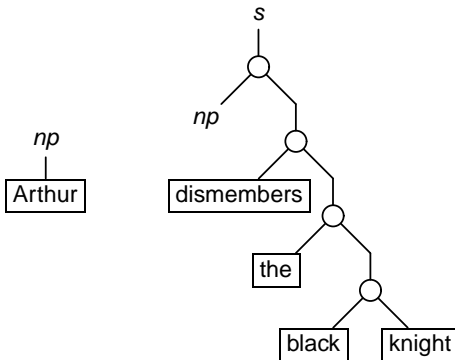
AB Grammars

Example: 'Arthur dismembers the black knight'



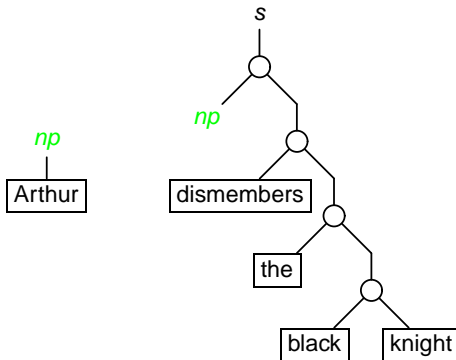
AB Grammars

Example: 'Arthur dismembers the black knight'



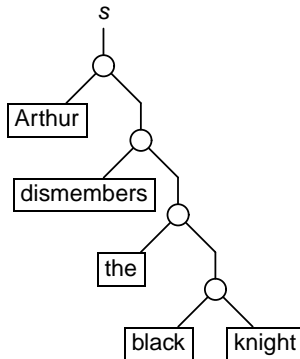
AB Grammars

Example: 'Arthur dismembers the black knight'



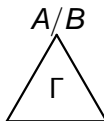
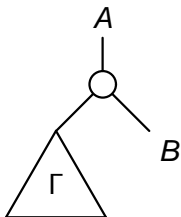
AB Grammars

Example: 'Arthur dismembers the black knight'



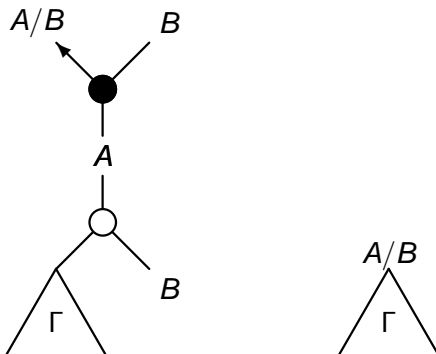
NL Grammars

Logical Rules



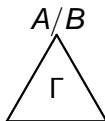
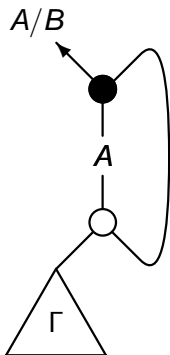
NL Grammars

Logical Rules



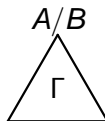
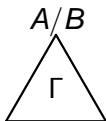
NL Grammars

Logical Rules



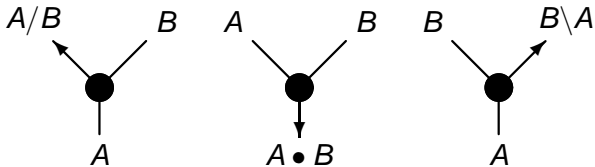
NL Grammars

Logical Rules



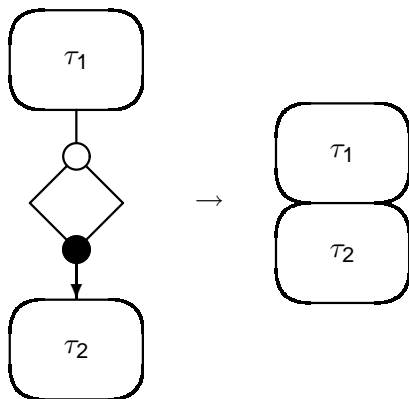
NL Grammars

Links



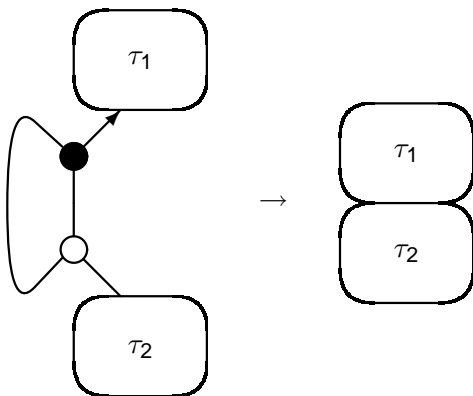
NL Grammars

Graph Contractions



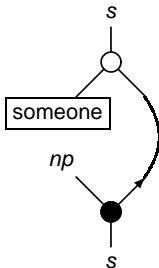
NL Grammars

Graph Contractions



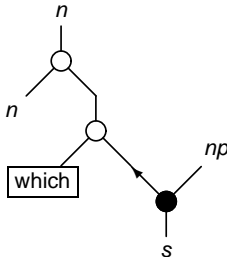
NL Grammars

Example: Quantifiers



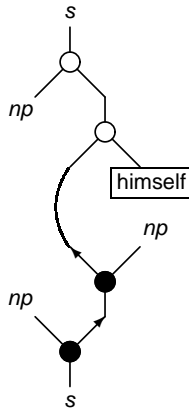
NL Grammars

Example: Extraction



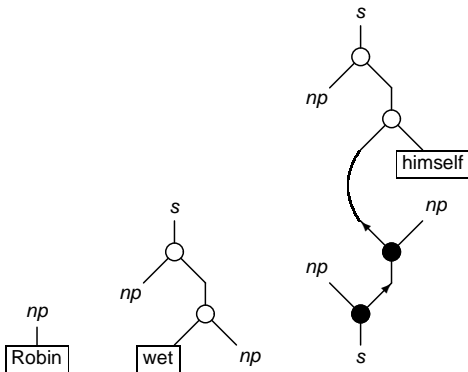
NL Grammars

Example: Reflexives



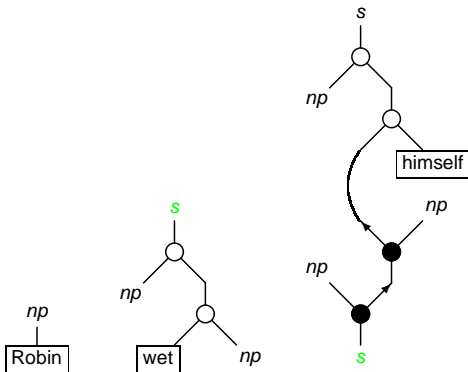
NL Grammars

Example: 'Robin wet himself'



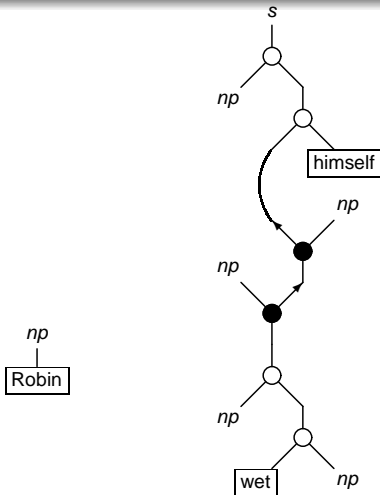
NL Grammars

Example: 'Robin wet himself'



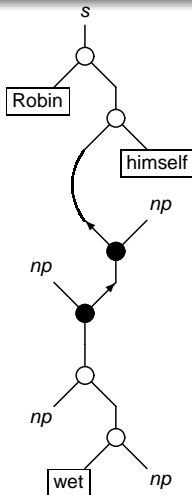
NL Grammars

Example: 'Robin wet himself'



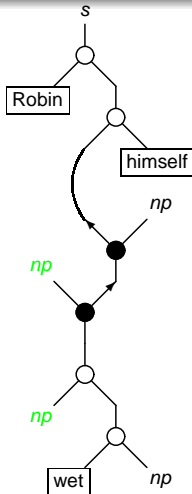
NL Grammars

Example: 'Robin wet himself'



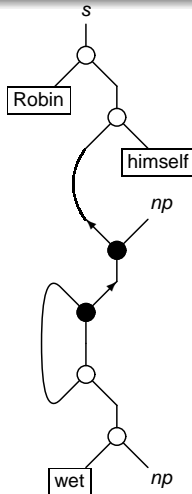
NL Grammars

Example: 'Robin wet himself'



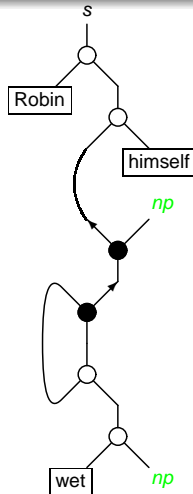
NL Grammars

Example: 'Robin wet himself'



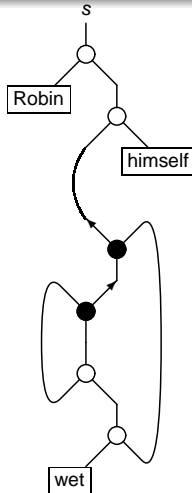
NL Grammars

Example: 'Robin wet himself'



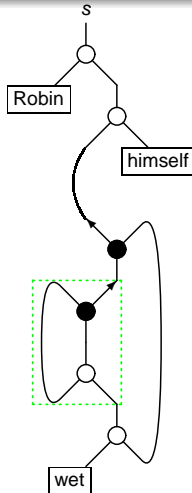
NL Grammars

Example: 'Robin wet himself'



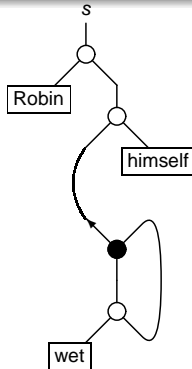
NL Grammars

Example: 'Robin wet himself'



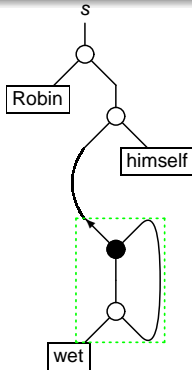
NL Grammars

Example: 'Robin wet himself'



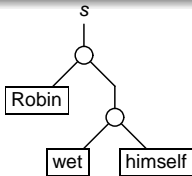
NL Grammars

Example: 'Robin wet himself'



NL Grammars

Example: 'Robin wet himself'

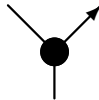
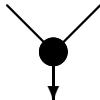
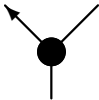
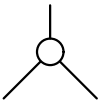


Multimodal Categorical Grammars

- NL is an interesting formalism, but it recognizes only context free languages.
- To boost its expressive power, as well as increase the level of detail in the analyses, we will look at multimodal extensions of NL (Moortgat 1997).

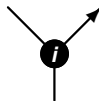
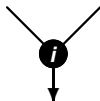
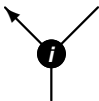
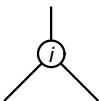
Multimodal Categorical Grammars

Mode Information



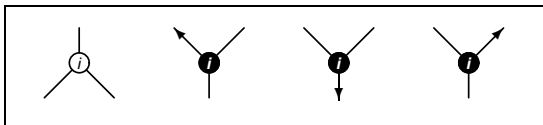
Multimodal Categorical Grammars

Mode Information



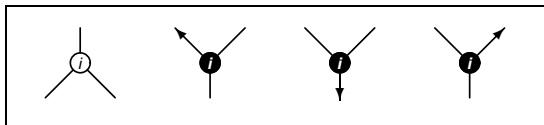
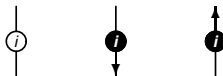
Multimodal Categorical Grammars

More Branches



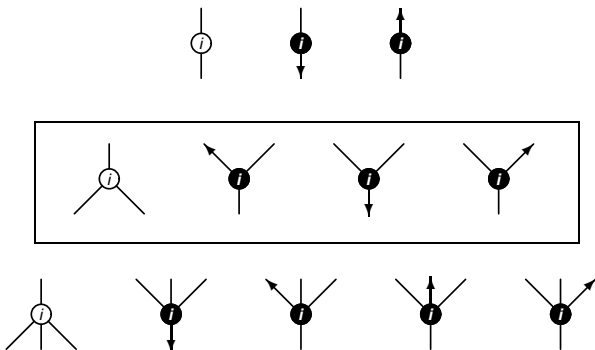
Multimodal Categorical Grammars

More Branches



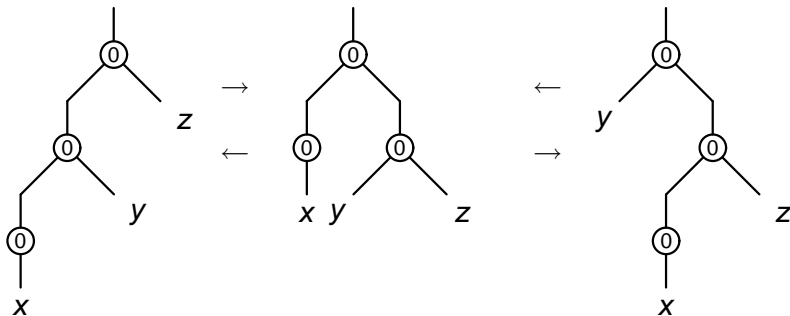
Multimodal Categorical Grammars

More Branches



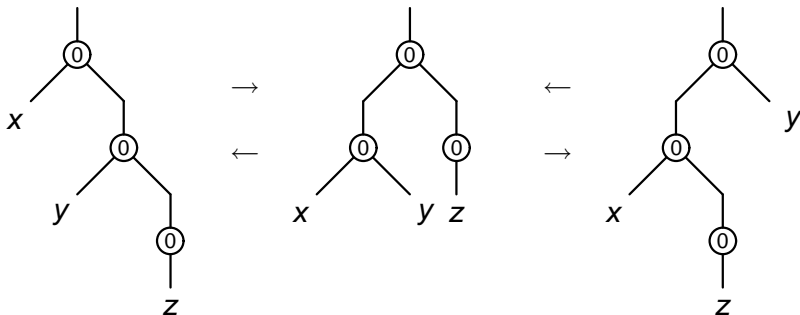
Multimodal Categorical Grammars

Structural Rules: Left Branches

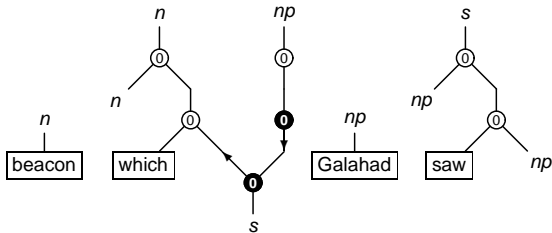


Multimodal Categorical Grammars

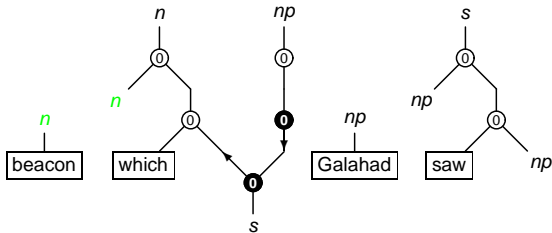
Structural Rules: Right Branches



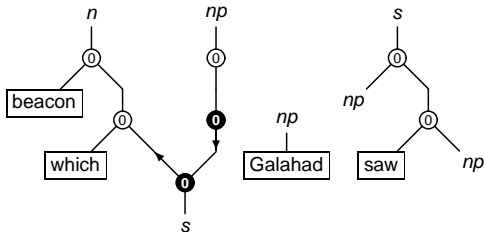
Example: Modified Lexicon



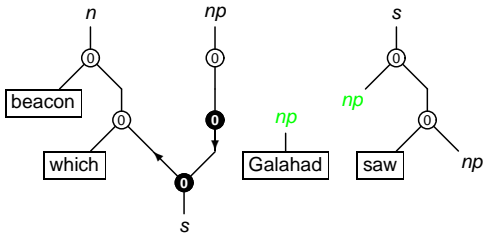
Example: 'beacon which Galahad saw'



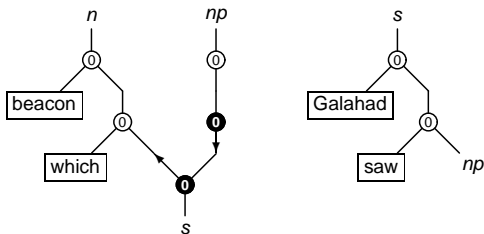
Example: 'beacon which Galahad saw'



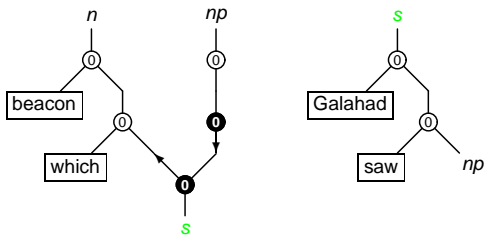
Example: 'beacon which Galahad saw'



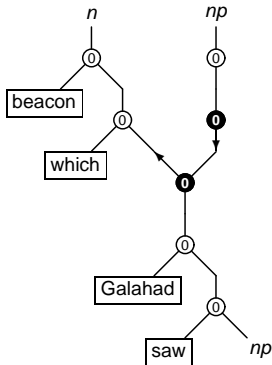
Example: 'beacon which Galahad saw'



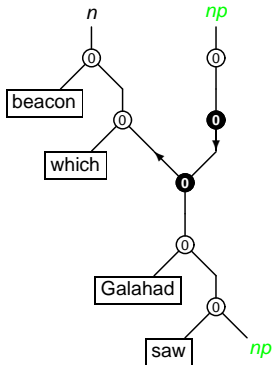
Example: 'beacon which Galahad saw'



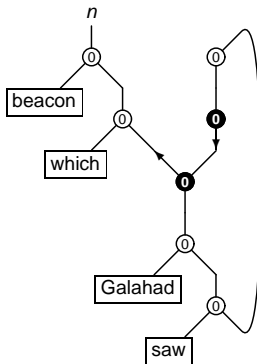
Example: 'beacon which Galahad saw'



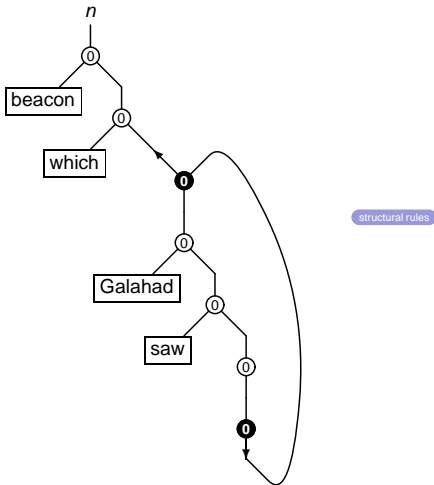
Example: 'beacon which Galahad saw'



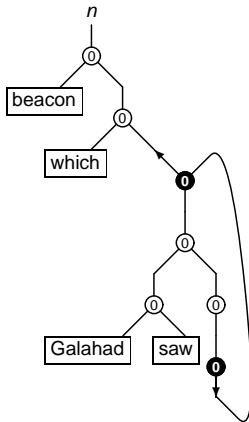
Example: 'beacon which Galahad saw'



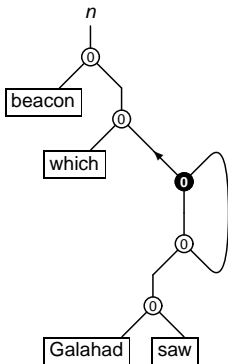
Example: 'beacon which Galahad saw'



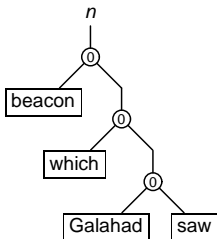
Example: 'beacon which Galahad saw'



Example: 'beacon which Galahad saw'



Example: 'beacon which Galahad saw'



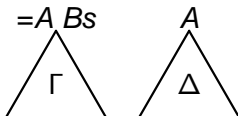
Minimalist Grammars

Introduction

- Stabler's (1997, 1999) formalisation of Chomsky (1995).
- Close to categorial grammars in many respects (Amblard 2007, Lecomte 2000, Lecomte & Retoré 1999, Retoré & Stabler 1999, Vermaat 1999)
- Let's explore why.

Minimalist Grammars

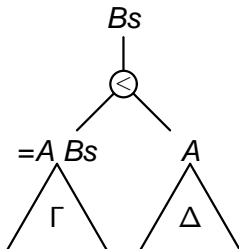
Merge



Minimalist Grammars

Merge

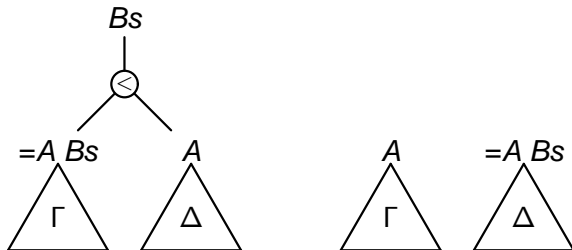
Lexical



Minimalist Grammars

Merge

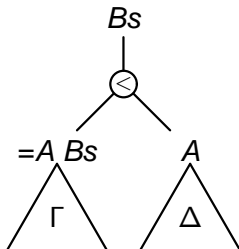
Lexical



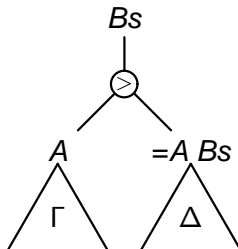
Minimalist Grammars

Merge

Lexical



Derived



Minimalist Grammars

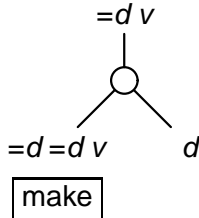
Comparison with Categorical Grammars

$=d = d v$

make

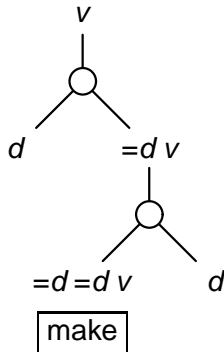
Minimalist Grammars

Comparison with Categorical Grammars



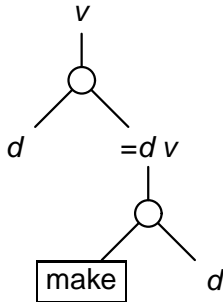
Minimalist Grammars

Comparison with Categorical Grammars



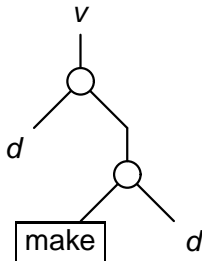
Minimalist Grammars

Comparison with Categorical Grammars



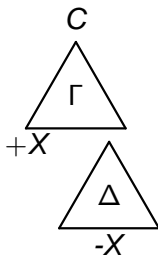
Minimalist Grammars

Comparison with Categorical Grammars



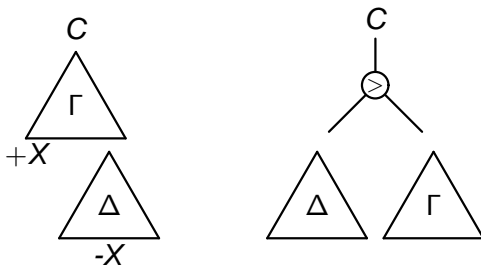
Minimalist Grammars

Move



Minimalist Grammars

Move



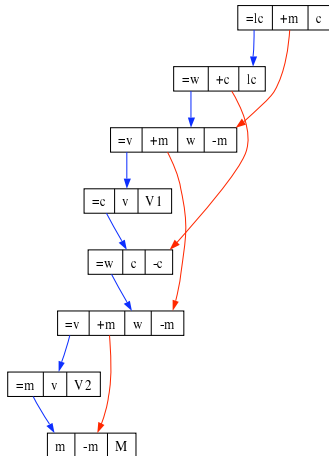
Minimalist Grammars

Move

- The merge operation is easily modeled in categorial grammars: it corresponds directly to the application rules.
- For the move operation, the correspondence is less clear and there are multiple possibilities (Lecomte 2000, Lecomte & Retoré 1999, Vermaat 1999).
- However, a convergence between the different proposals seems possible and I'll investigate this in the rest of this presentation.

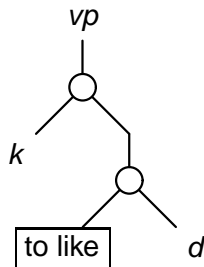
Proof Nets and Minimalist Grammars

Stabler (1998)



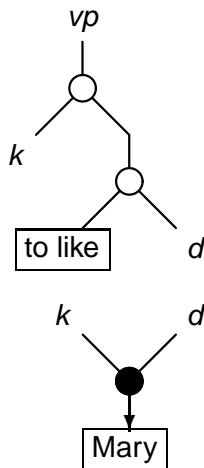
Proof Nets and Minimalist Grammars

Lecomte (2000)



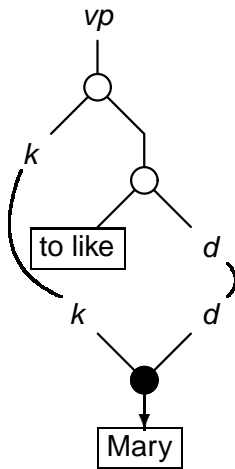
Proof Nets and Minimalist Grammars

Lecomte (2000)



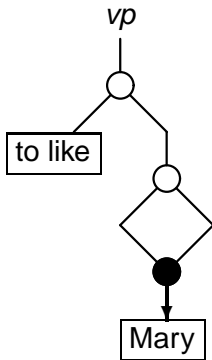
Proof Nets and Minimalist Grammars

Lecomte (2000)



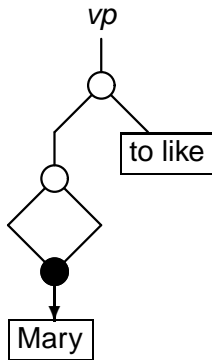
Proof Nets and Minimalist Grammars

Lecomte & Retoré (1999) — Strong and Weak Features



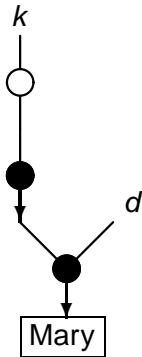
Proof Nets and Minimalist Grammars

Lecomte & Retoré (1999) — Strong and Weak Features



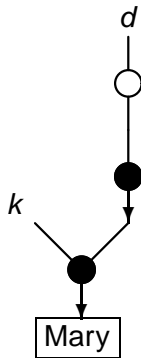
Proof Nets and Minimalist Grammars

Strong and Weak Features — Multimodal Implementation



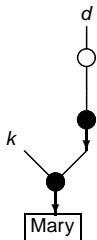
Proof Nets and Minimalist Grammars

Strong and Weak Features — Multimodal Implementation



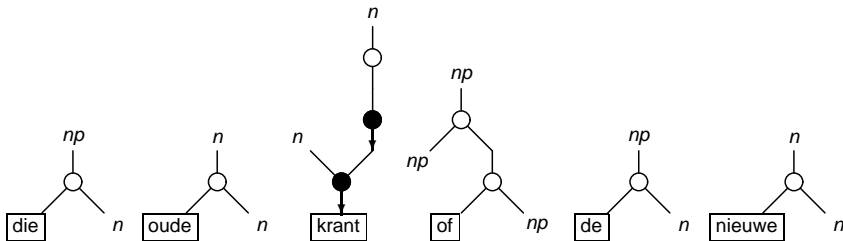
Proof Nets and Minimalist Grammars

Comparison With Extracted Grammars



Proof Nets and Minimalist Grammars

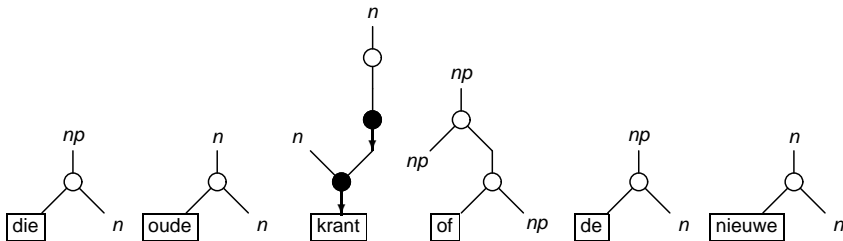
Comparison With Extracted Grammars



Proof Nets and Minimalist Grammars

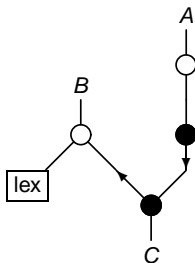
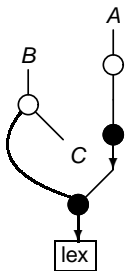
Comparison With Extracted Grammars

structural rules



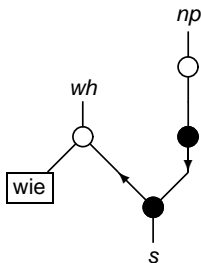
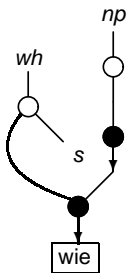
Proof Nets and Minimalist Grammars

Transformation of Extracted Grammars



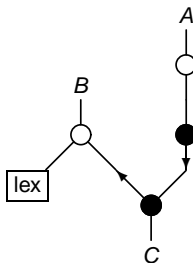
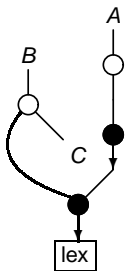
Proof Nets and Minimalist Grammars

Transformation of Extracted Grammars



Proof Nets and Minimalist Grammars

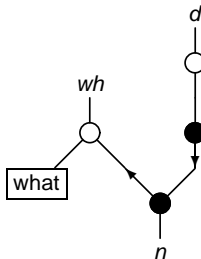
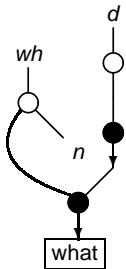
Transformation of Minimalist Grammars



Proof Nets and Minimalist Grammars

Transformation of Minimalist Grammars

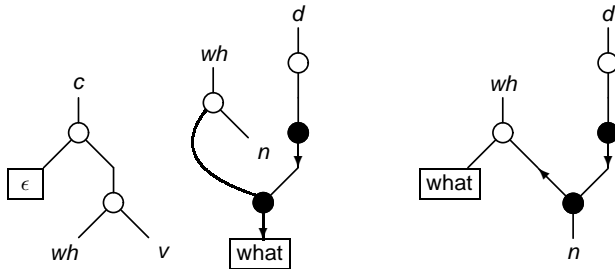
=n -wh d *what*



Proof Nets and Minimalist Grammars

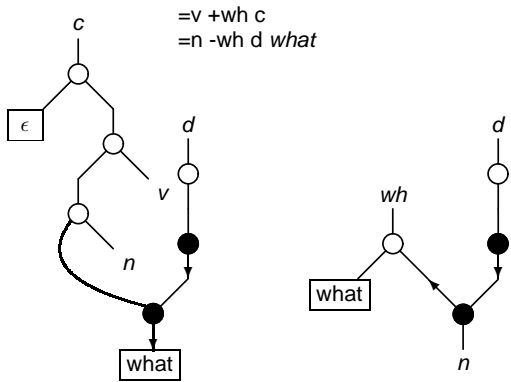
Transformation of Minimalist Grammars

$=v +wh c$
 $=n -wh d \textit{ what}$



Proof Nets and Minimalist Grammars

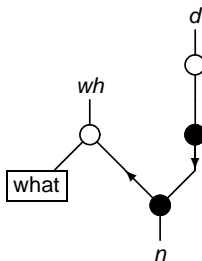
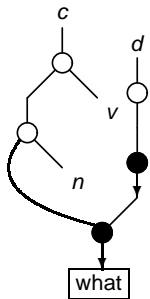
Removing assignments to the empty string



Proof Nets and Minimalist Grammars

Removing assignments to the empty string

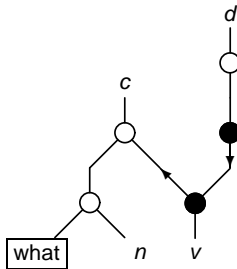
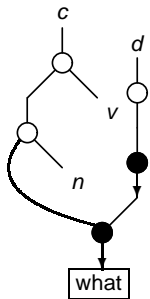
$=v +wh c$
 $=n -wh d \textit{ what}$



Proof Nets and Minimalist Grammars

Removing assignments to the empty string

$=v +wh c$
 $=n -wh d \textit{ what}$



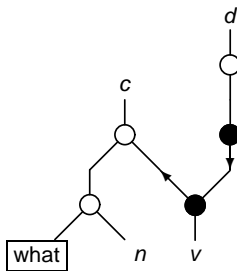
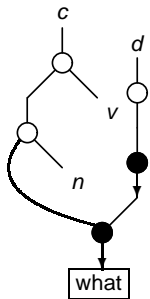
Proof Nets and Minimalist Grammars

Removing assignments to the empty string

structural rules

$=v +wh c$

$=n -wh d \textit{ what}$



Conclusions

- The Lecomte & Retoré-style and the Vermaat-style embeddings of minimalist grammars into categorial grammars are more similar than first impressions would suggest.
- In either case, a unimodal categorial grammar with a permute modality suffices.

Open Questions

- As the example shows, in order to find a correct product-less type, we need to find an appropriate category from where the moving element is extracted. Is this a reasonable restriction? Does it limit the class of languages we can generate?
- What are the conditions under which we can ‘compile away’ the assignments to the empty string?
- Can we show a true embedding and use this result to get a polynomial parsing algorithm for the second order Lambek calculus with the controlled extraction postulates?