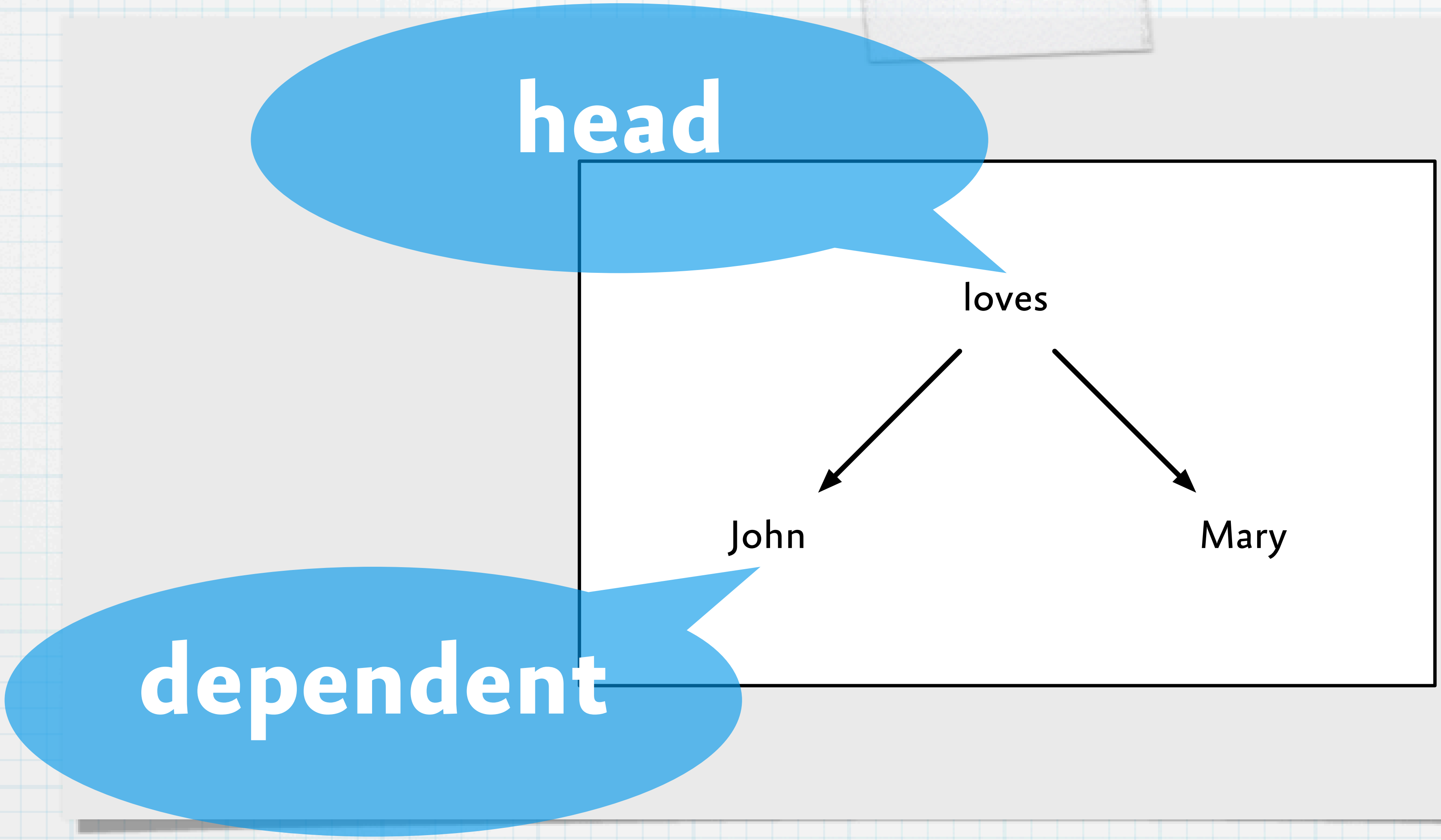
Expressivity and Complexity of Dependency Grammars

Marco Kuhlmann

Programming Systems Lab
Saarland University, Saarbr cken, Germany

IGK Colloquium · Saarbr cken · 2005–11–10



Dependencies

Current interest in DG

* **statistical parsing**
  Eisner 1996; Collins 1997
* **languages with free word order**
  Pl tek et al. 2001
* **syntax-semantics interface**
  Debusmann et al. 2001

The DG diversity

* **grammatical paradigm**
  * rule-based (Gaifman, Dikovsky)
  * constraint-based (CDG, XDG)
* **structural assumptions**
  * projective (Gaifman, Eisner)
  * non-projective (CDG, Nasr, XDG)

Milestones

- **Gaifman 1965:**
  Projective dependency grammars and lexicalised context-free grammars are strongly equivalent.
- **Neuhaus & Br ker 1997:**
  The general word problem for unrestricted dependency grammars is NP-complete.

What this talk is about

* **formal and computational aspects of dependency grammar**
  * language-theoretic expressivity
  * complexity of recognition & parsing
* **intermediate report on ongoing work for my PhD thesis**

Questions asked

* How does DG relate to grammar formalisms other than CFG?
* How could a general framework look like, in which existing DG formalisms can be studied and compared?
* How can parsing techniques for other frameworks be transferred to DG?

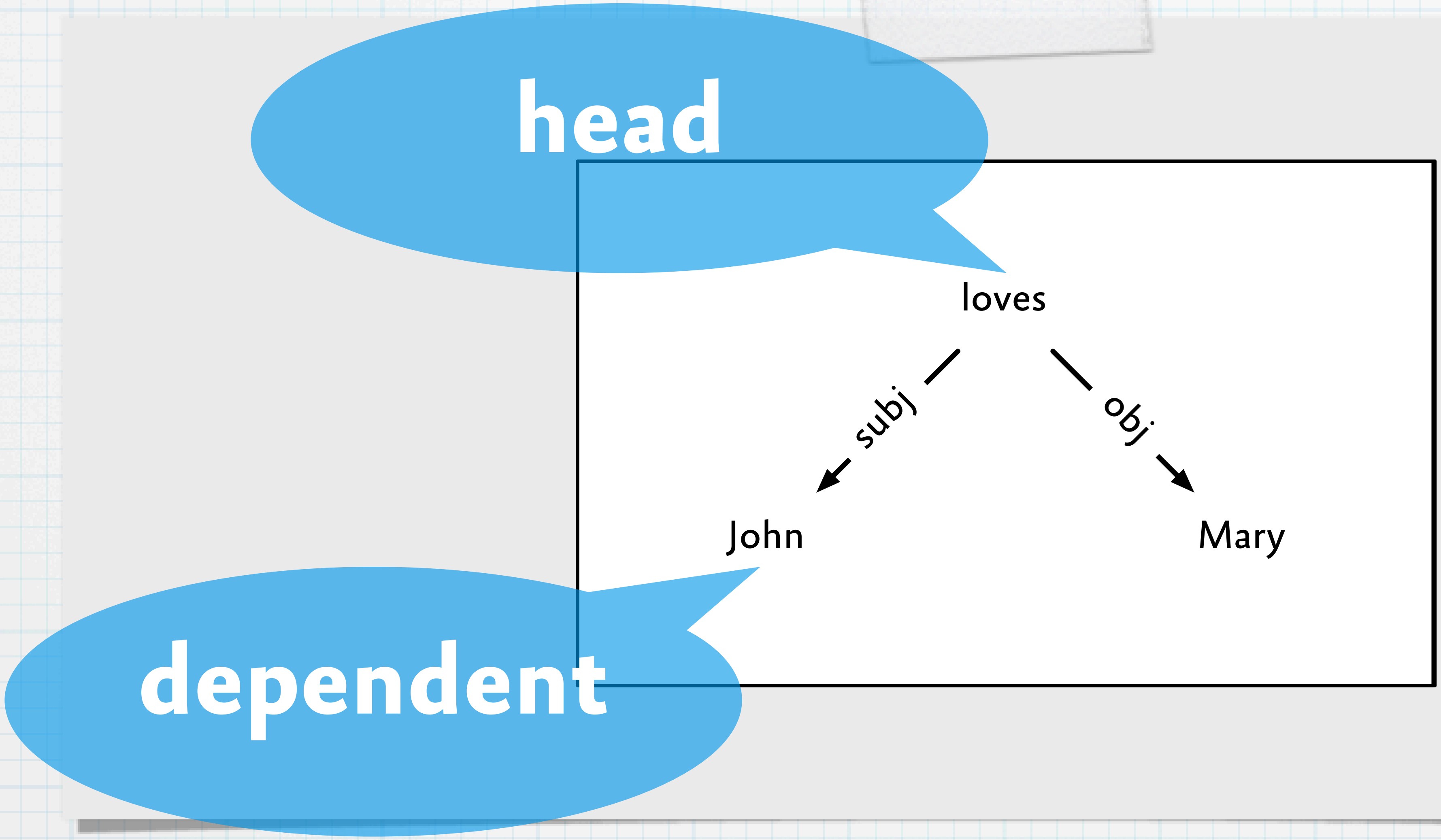
Focus of this talk

- **dependency structures**
  - structural constraints
  - non-projectivity
- **dependency languages**
  - mild context-sensitivity
  - extensional perspective

Structure

* Introduction
* **Dependency structures**
* Dependency languages
* Complexity
* Conclusion & Future Work

Dependency structures



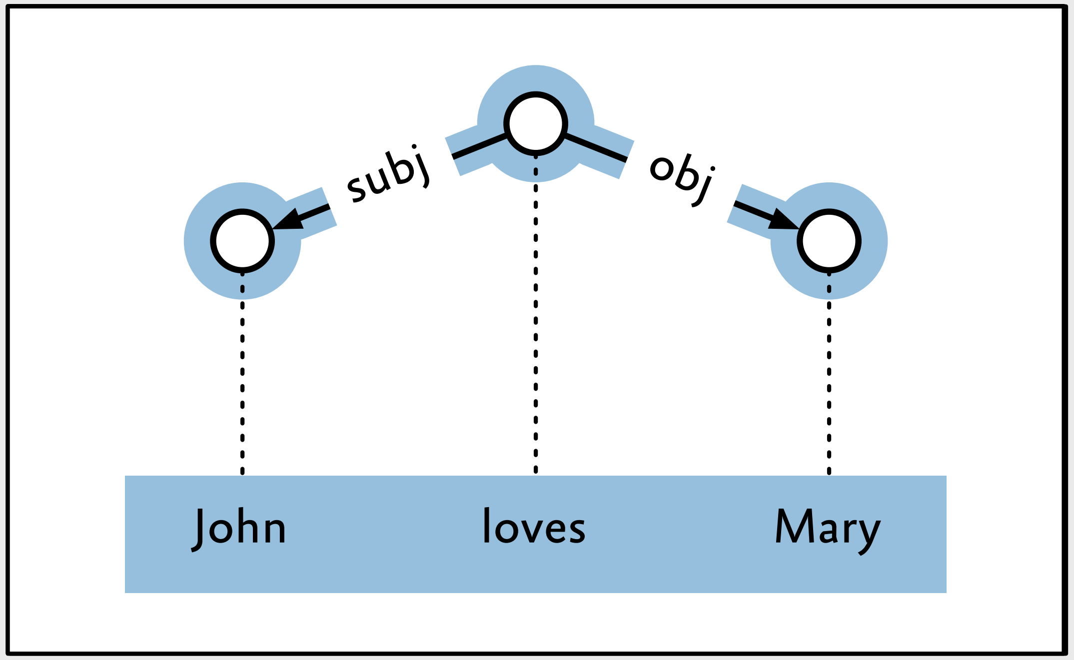
Dependencies

Common constraints (1)

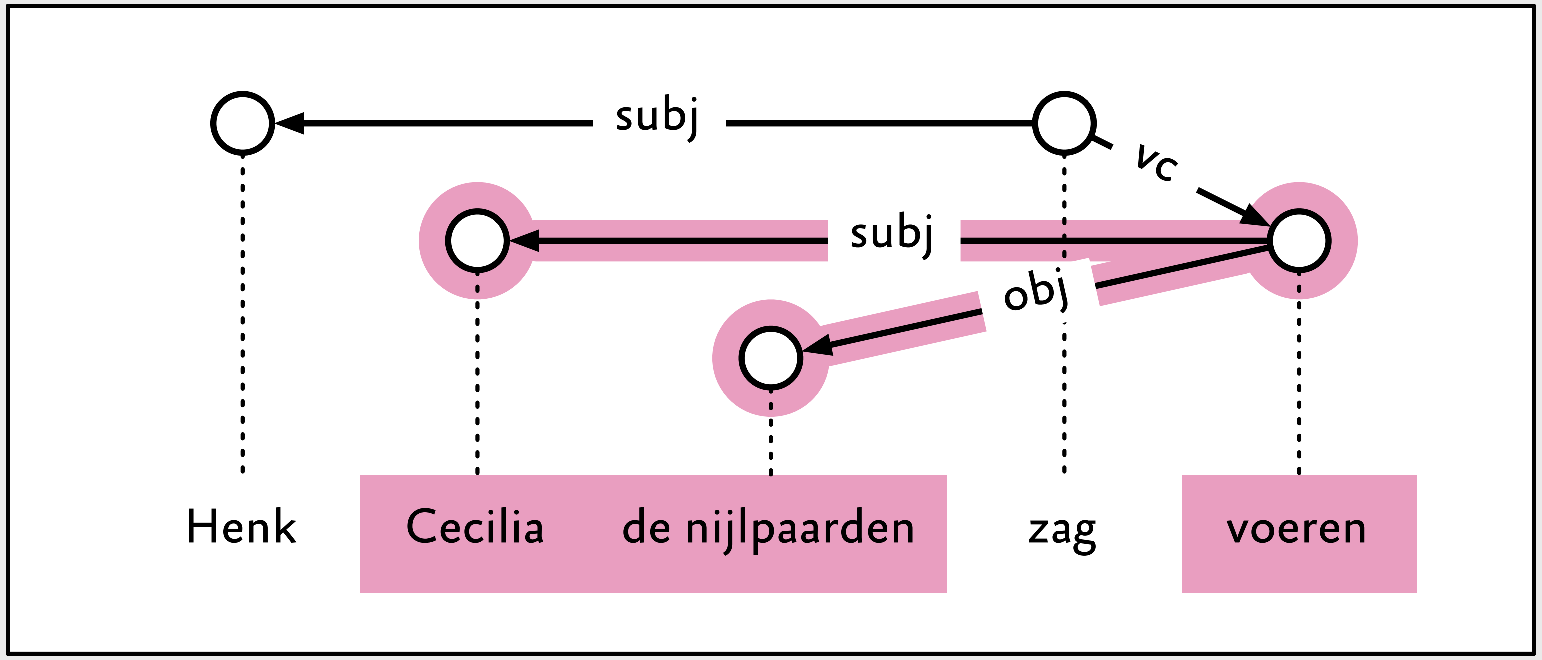
- **acyclicity:**
  no word depends on itself
- **indegree at most 1:**
  each word has at most one head
- **single root:**
  exactly one word without a head

Common constraints (2)

* **projectivity:**
  the reflexive-transitive dependents
  of a word form a contiguous substring
  of the full sentence
* controversial; must be abandoned
  for languages with freer word order



Projectivity

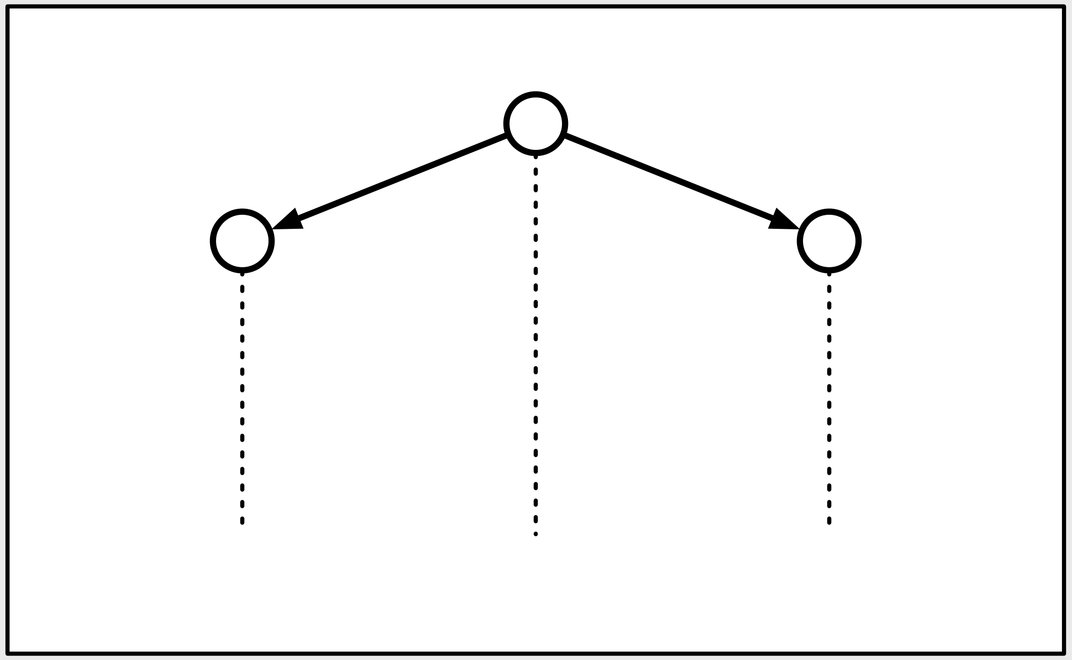


Crossing edges

Drawings

* **simple models of syntactic structure**
  * relational structure
  * forest + linear order on the nodes
* **dimensions of non-projectivity**
  * quantitative aspect
  * qualitative aspect

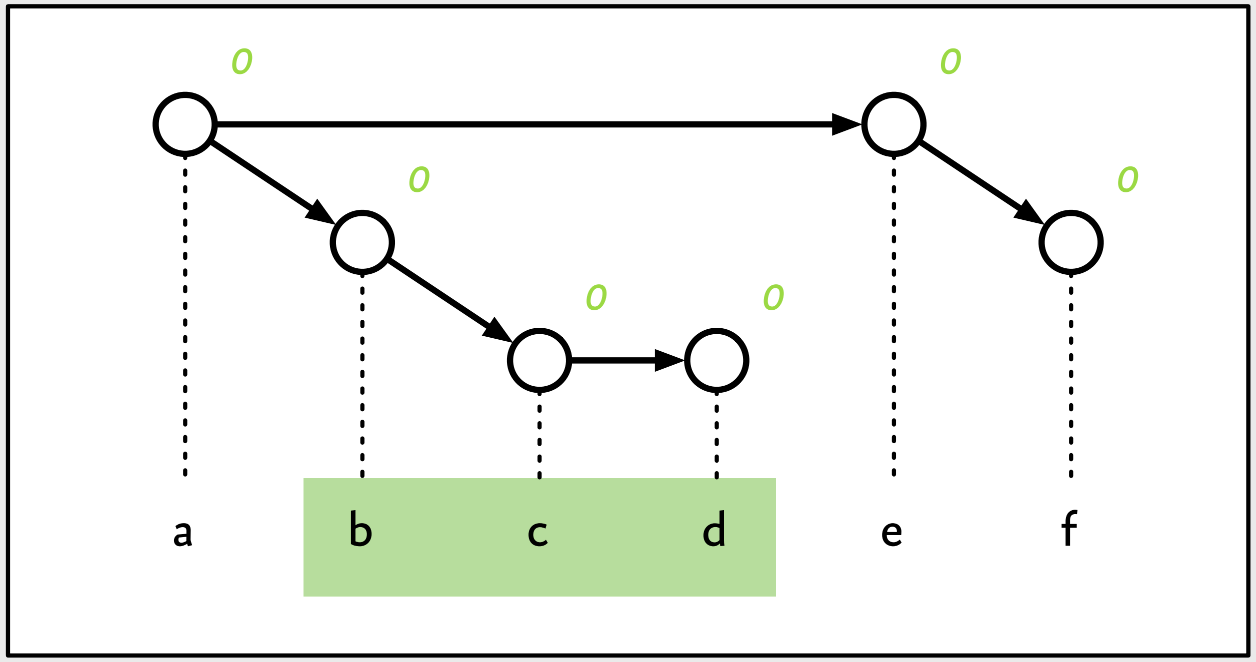
not an ordered tree



Drawings

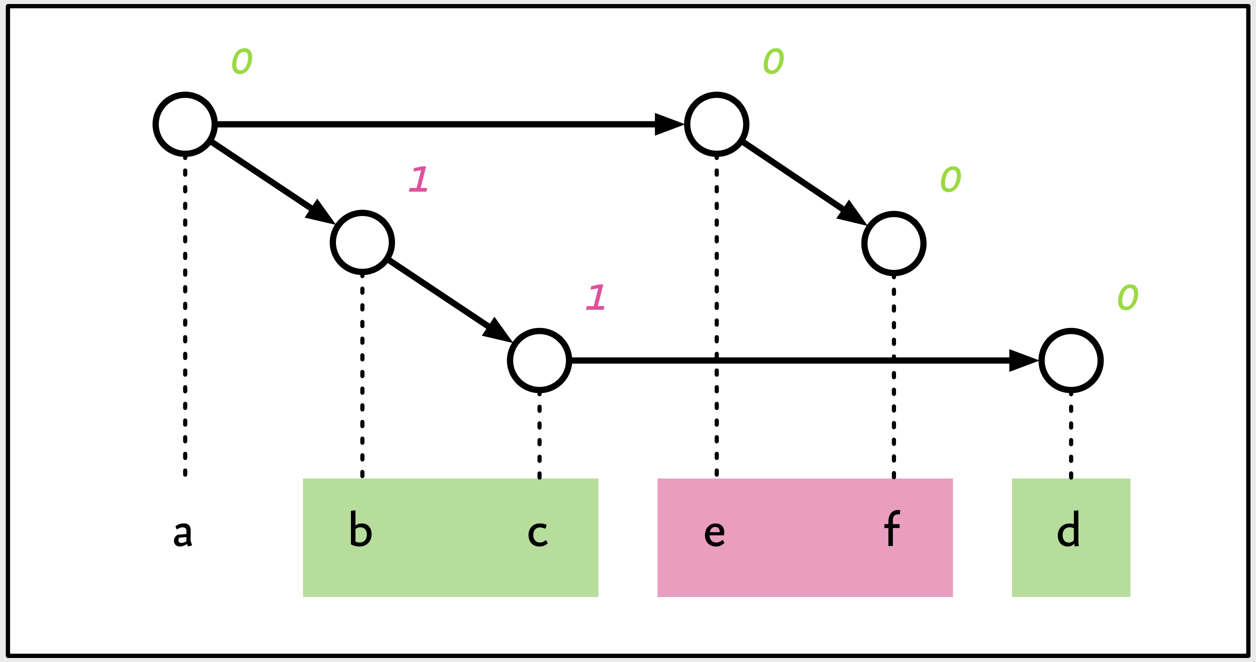
Gap degree

* **gap degree of a node:**
  number of interruptions
  in the projection of that node
* **gap degree of a drawing:**
  maximum over the gap degrees
  of the nodes in the drawing

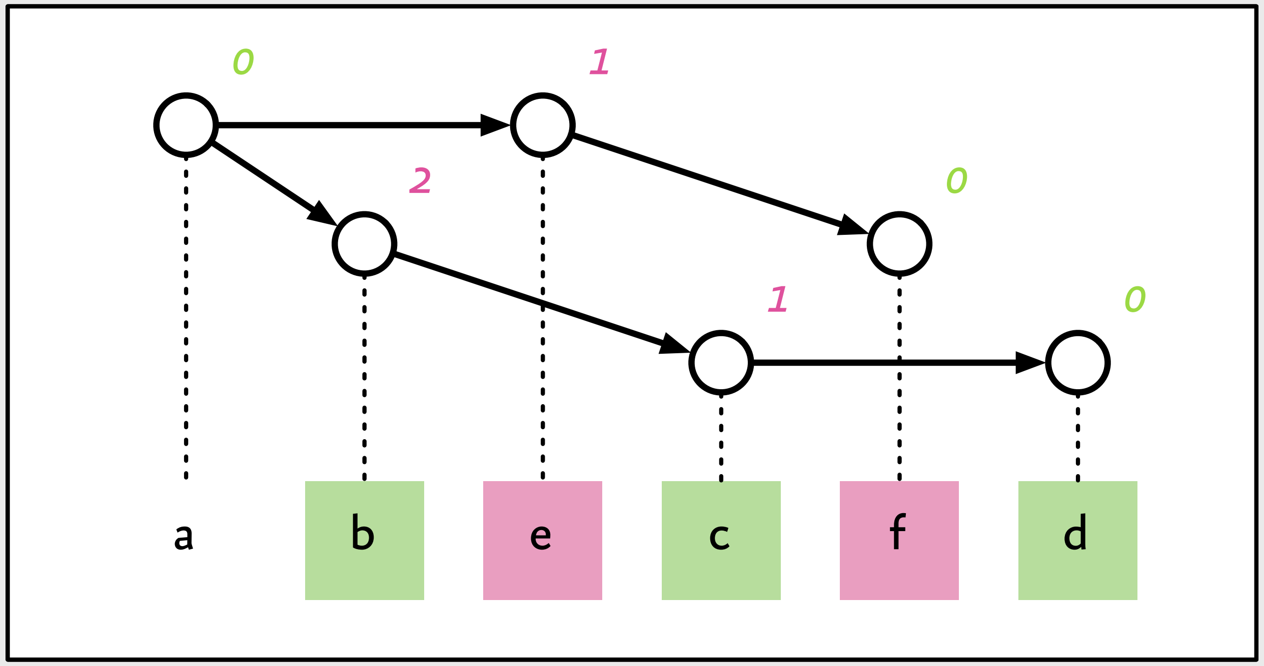


projective

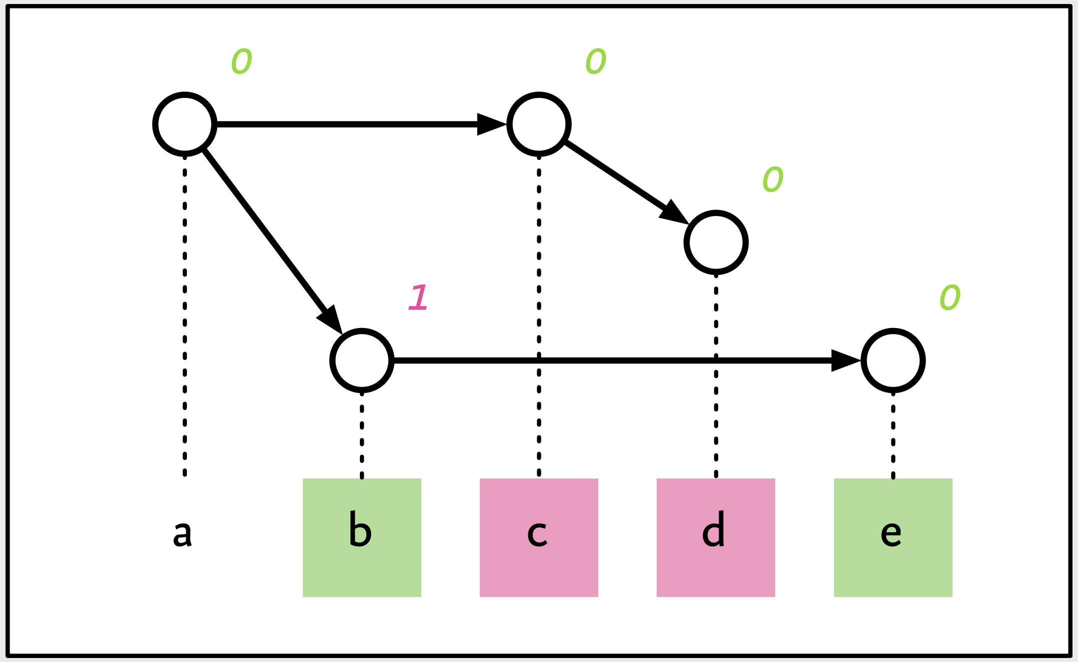
Gap degree 0



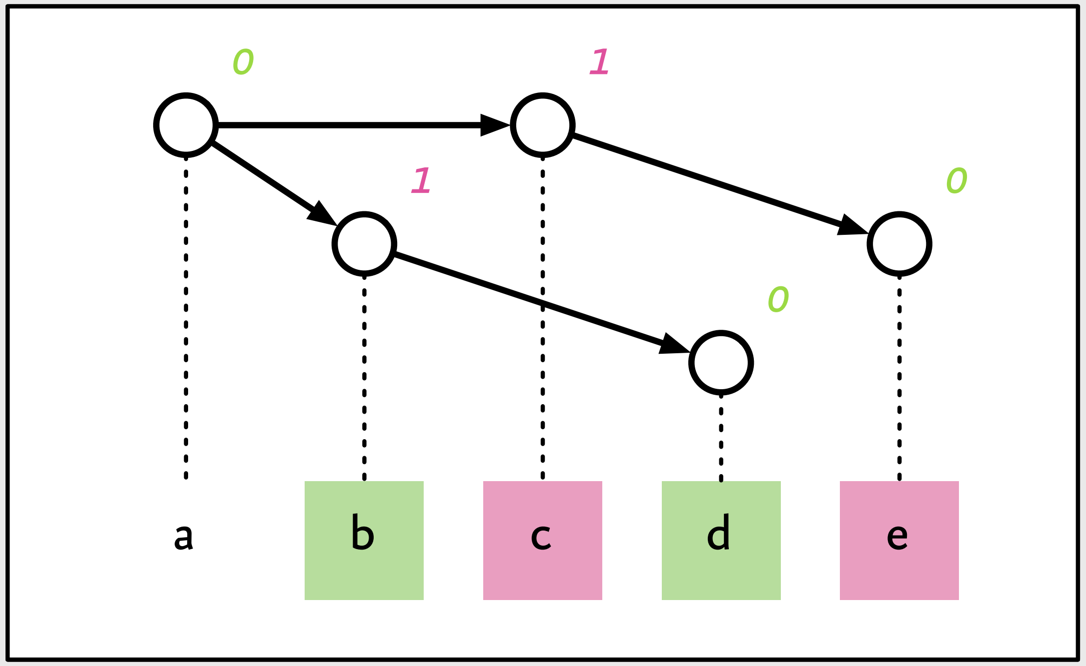
Gap degree 1



Gap degree 2



Well-nested drawings (1)



Well-nested drawings (2)

Results

* The derivations of Lexicalised Tree Adjoining Grammar (LTAG) can be interpreted as drawings in a natural way.
* **The drawings induced by LTAG are well-nested and have a gap degree of at most 1.**

Bodirsky, Kuhlmann & Möhl – FG/MoL 2005

Summary

* I propose **drawings** as a simple class of models for dependency structure.
* Drawings allow us to formalise and reason about various forms of **non-projectivity**.

Structure

* Introduction
* Dependency structures
* **Dependency languages**
* Complexity
* Conclusion & Future Work

Dependency languages

Definitions

* **dependency language:**
  set of dependency structures
* **string language:**
  image of a dependency language under projections
* **dependency grammar:**
  specifies a dependency language

CF languages

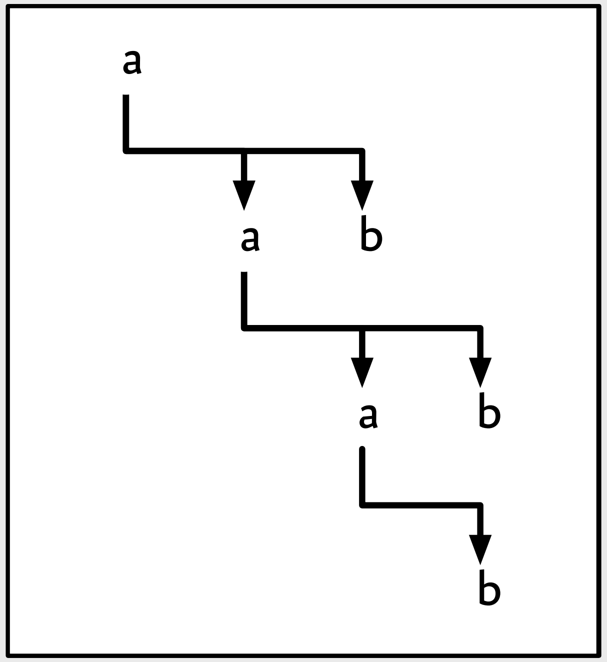
* Gaifman proved that lexicalised context-free grammars and projective dependency grammars are equivalent.
* **Context-free dependency language:** image of a context-free set of trees under projective closure

$$S \to a\,S\,B$$

$$S \to a\,B$$

$$B \to b$$

CF grammar



CF tree set

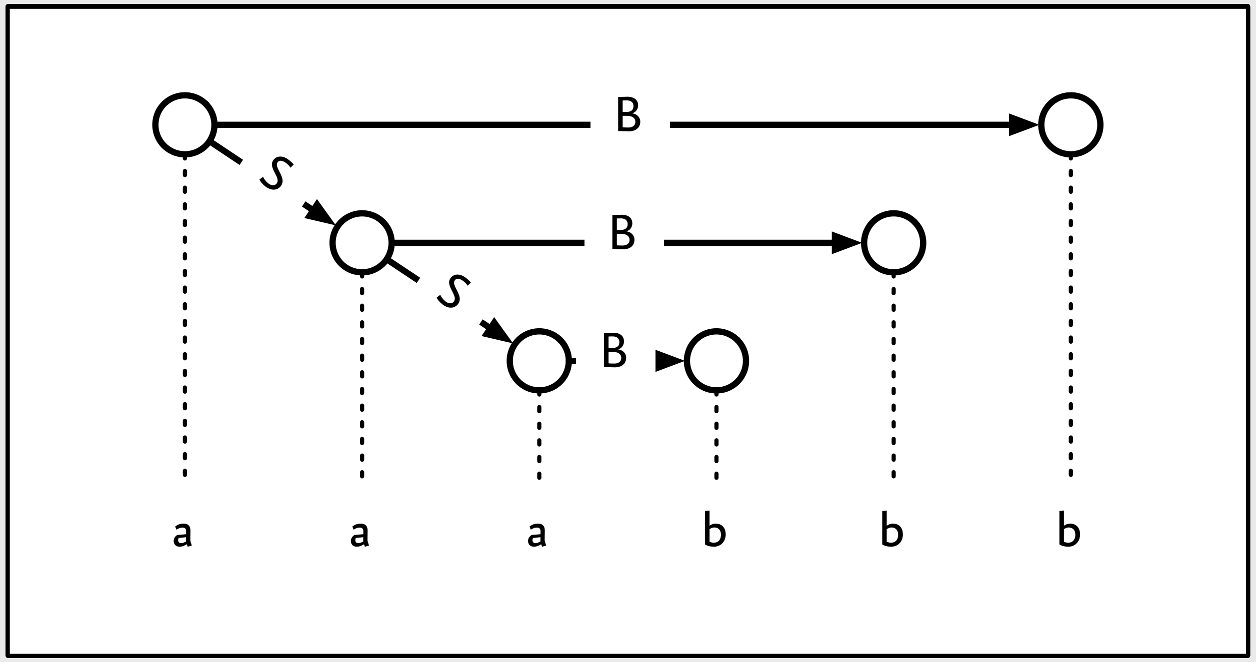
$$\frac{v \prec_u w}{v \prec w}$$

local order

$$\frac{Suv \qquad v \prec_u w \qquad Suw}{S^+ v \prec S^+ w}$$

global order

Projective closure



CF dependency language

Dissecting CF languages

* an underlying set of unordered trees: **forest structure**
* global linearisation constraints: **projectivity, non-projectivity**
* local linearisation constraints: **grammar-specific**

MCS languages

* **mildly context-sensitive languages**
  * extend context-free languages
  * limited form of non-projectivity
* **mildly context-sensitive formalisms**
  * TAG, CCG, Minimalist Grammar
  * corpora and practical parsers exist

Properties of MCSL

* **limited crossing dependencies:**
  crossed-serial dependencies in Dutch
* **constant growth property:**
  the size progression of the language is bounded by a constant
* **recogniseable in polynomial time**

The plan

* Take the sets of (unordered) trees specified by context-free grammars.
* Define a class of deterministic automata that transform those trees into dependency structures.
* Prove that the resulting output languages are mildly context-sensitive.

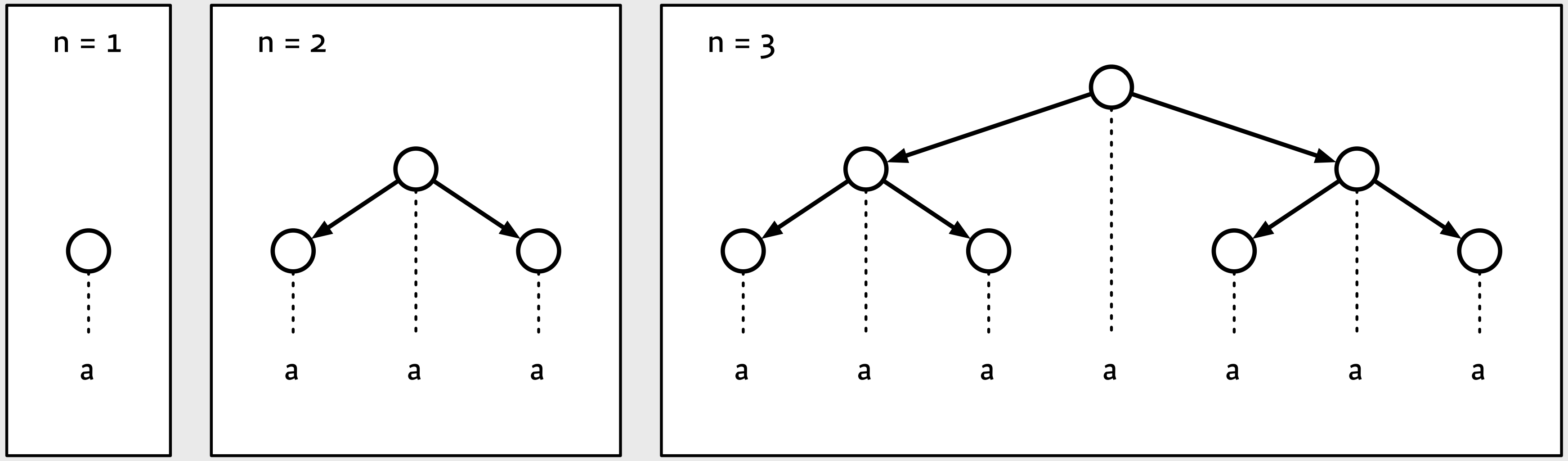
Semilinearity

* Semilinearity is a property that implies the constant growth property.
* It may be too strong a constraint to be put on natural language.
* A language is **semilinear**, if its Parikh image can be decomposed into a finite union of linear sets.

Parikh images

* **Parikh vector for a string:**
  function that maps terminal symbols
  to their numbers of occurrences
* **Parikh image of a language:**
  set of Parikh vectors
  for the strings in that language

$$\{\, a^{2^n - 1} \mid n \geq 1 \,\}$$



Semilinearity

Tree linearisers

* **ingredients:** context free grammar + specification of linearisation
* **output:** a set of dependency structures (labelled drawings)
* specialisation of deterministic tree-walking transducers (Weir 1992)

Tree linearisers

* finite set of **context-free rules**
* finite set of **states**
* finite set of **actions**: *up*, *down n*, *mark*
* **transition function:**
given a current rule and state,
what is the action to take,
and what is the new state?

deterministic

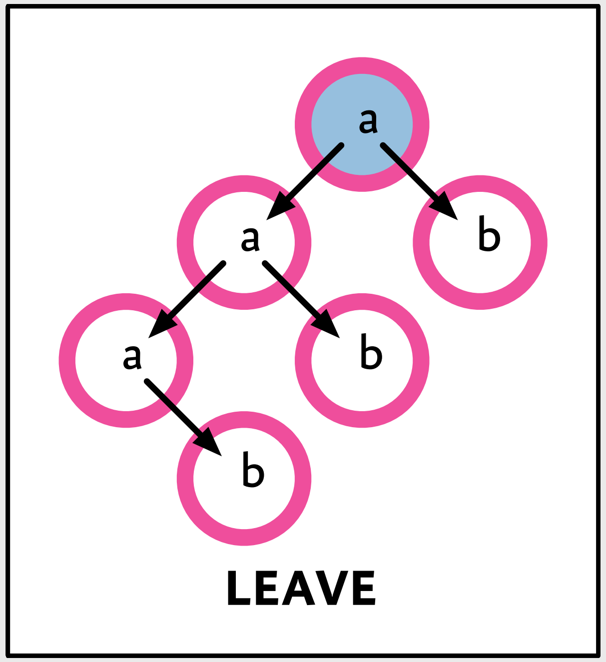
$$S \to a\,S\,B$$
$$S \to a\,B$$
$$B \to b$$

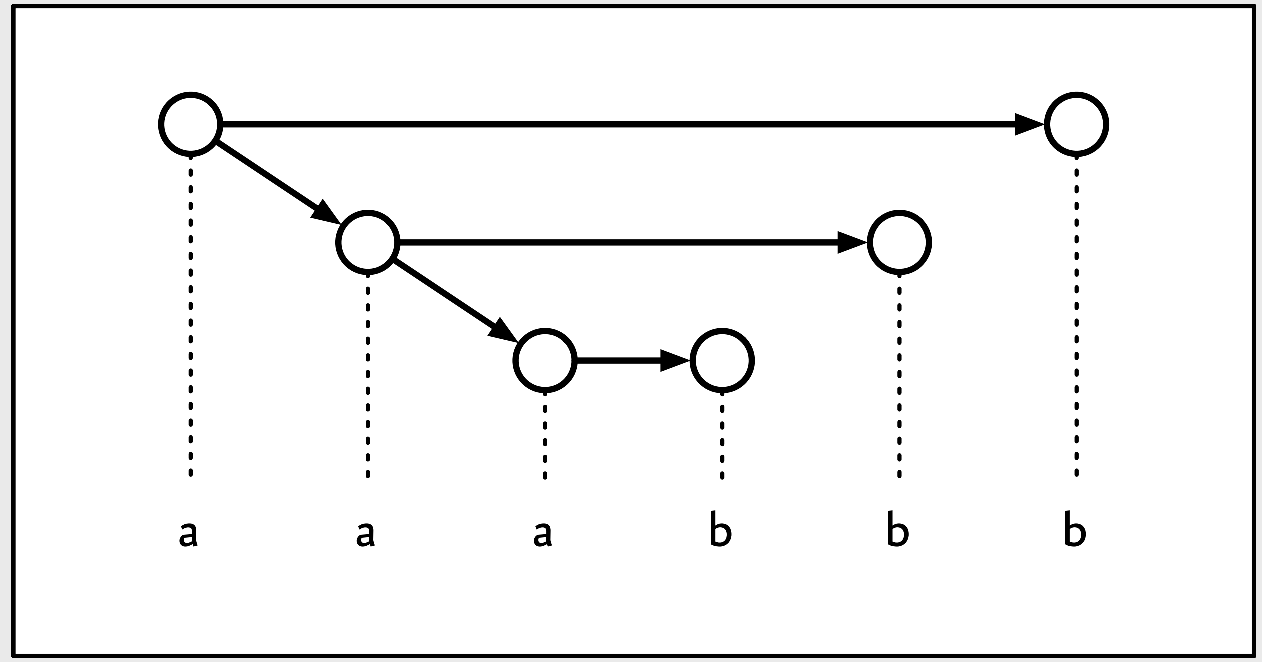
CF grammar

$$S \to a\ S\ B$$

$$\text{entered} \to (\textbf{MARK}, \text{marked})$$
$$\text{marked} \to (\textbf{ENTER S}, \text{entered})$$
$$\text{left S} \to (\textbf{ENTER B}, \text{entered})$$
$$\text{left B} \to (\textbf{LEAVE}, \text{left S})$$

Lineariser 1





Lineariser 1

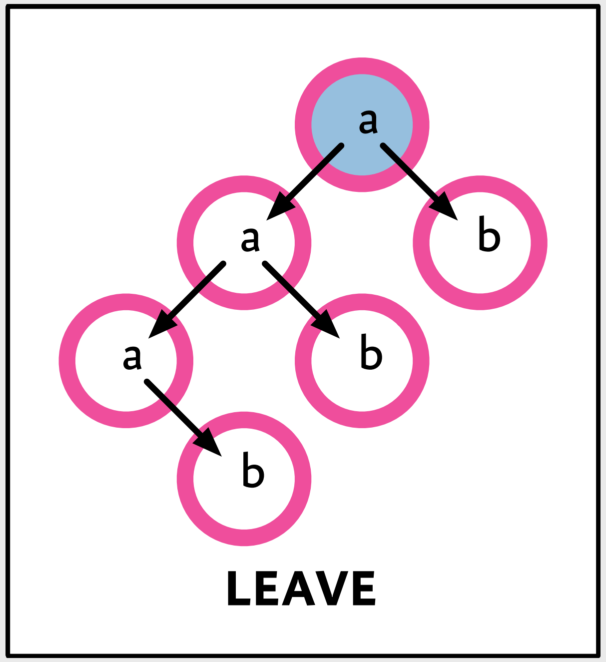
Beyond context-freeness

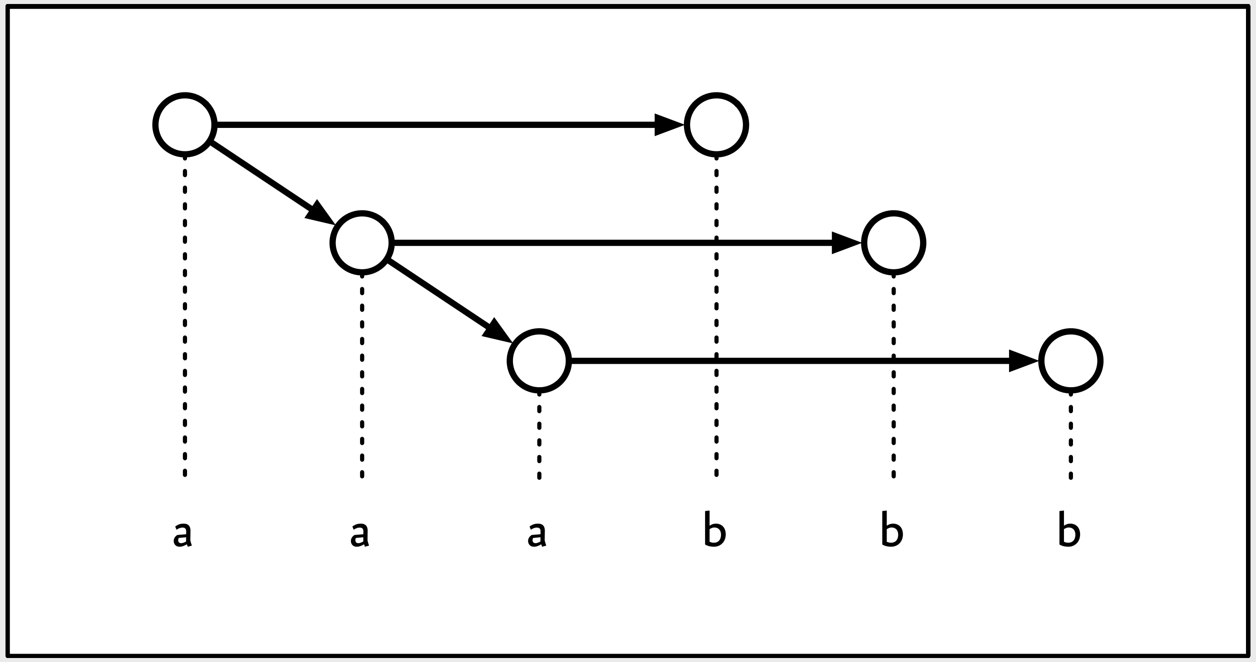
* **reentrancies:** allow a subtree to be entered and left more than once
* number of times that this happens is called the **crossing number**
* extension introduces a finite number of additional states

S $\rightarrow$ a S B

entered 1 $\rightarrow$ (**MARK**, marked)
marked $\rightarrow$ (**ENTER S**, entered 1)
left S 1 $\rightarrow$ (**LEAVE**, left S 1)
entered 2 $\rightarrow$ (**ENTER B**, entered)
left B $\rightarrow$ (**LEAVE**, left S 2)

Lineariser 2





Lineariser 2

DTL – Finite gap degree

* Each subtree can be visited at most a finite number of times.
* Therefore, the yield of each node can be split into at most a finite number of convex blocks, with a finite number of gaps.

DTL – Semi-linearity

* **Parikh's Theorem:**
  A language is semi-linear if and only if its Parikh image is the Parikh image of a context-free language.
* Deterministic tree linearisers merely order the nodes of a context-free set of trees.

Summary

* **basic idea:**
  characterise mildly context-sensitive
  dependency languages
  by different tree linearisation regimes
* **still to be done:**
  show that the output languages
  of tree linearisers
  can be parsed in polynomial time

A note on grammars

* I have taken a completely **extensional approach** to dependency languages.
* Ultimately, I also want to be able to explain what a mildly context-sensitive **dependency grammar** is.
* I have taken first steps into this direction (Grabowski et al., 2005).

Structure

* Introduction
* Dependency structures
* Dependency languages
* **Complexity**
* Conclusion & Future Work

Complexity

Fundamental results

* **Eisner 1996:**
  Projective dependency grammars can be parsed in time $O(n^3)$.
* **Neuhaus & Br ker 1997:**
  The general word problem for unrestricted dependency grammars is NP-complete.

Contribution

* **general parsing schema**
  for gap-restricted
  dependency languages
* **parsing schema:**
  abstract specification of a parsing
  algorithm as an inference system
  (Sikkel 1997)

span

type

$$s : \langle I, \Omega \rangle$$

Parse items

$$\frac{\langle I, \Omega \rangle \in Lex(w_i)}{\{i\} : \langle I, \Omega \rangle}$$

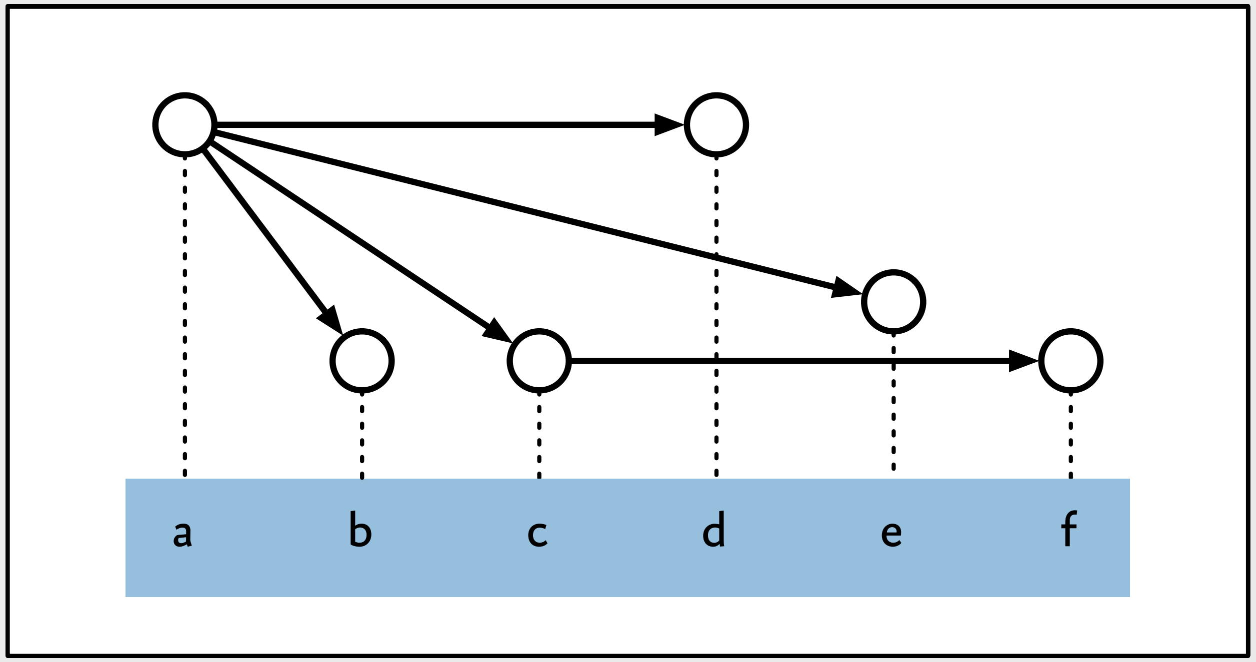
The LOOKUP rule

$$\frac{s_1 : \langle I_1, \Omega + I_2 \rangle \qquad s_2 : \langle I_2, \emptyset \rangle}{s_1 \oplus s_2 : \langle I_1, \Omega \rangle}$$

The PLUG rule

$$\frac{s_1 : \langle I_1, \emptyset \rangle \quad \cdots \quad s_m : \langle I_m, \emptyset \rangle}{s_1 \oplus \cdots \oplus s_m : \langle I_1 + \cdots + I_m, \emptyset \rangle}$$

The GROUP rule



Example

Results

- The parsing schema proves that **arbitrary gap-restricted drawings** can be parsed in polynomial time.

g^{m+2}

- In the special case of **well-nested gap-restricted drawings** a binary group rule suffices, independently of the gap degree.

$3(g+1)$

Conclusion & Future Work

Summary

* I have proposed a general framework in which existing DG formalisms can be studied and compared.
* I have presented evidence that this framework will provide the notion of mildly context-sensitive dependency grammars.

Future work (1)

* **formal aspects**
  * dependency *grammars*
  * embedding more formalisms
* **processing issues**
  * general gap-restricted drawings
  * issues related to lexicalisation

Future work (2)

* **linguistic relevance**
  * study more phenomena
  * corpus study on non-projectivity
* **implementation**
  * using constraint programming
  * using dynamic programming

Polynomial DGs

- There are linguistic phenomena that are beyond the expressive power of mildly context-sensitive languages.
- It might be interesting to study dependency correspondents of more powerful grammar formalisms, such as Literal Movement Grammar.

Thank you for listening!

Backup Slides

Affiliations

* Programming Systems Lab
  Department of Computer Science
* Sonderforschungsbereich 378
  Deutsche Forschungsgemeinschaft
* International Post-Graduate College
  Language Technology
  and Cognitive Systems

Collaborators

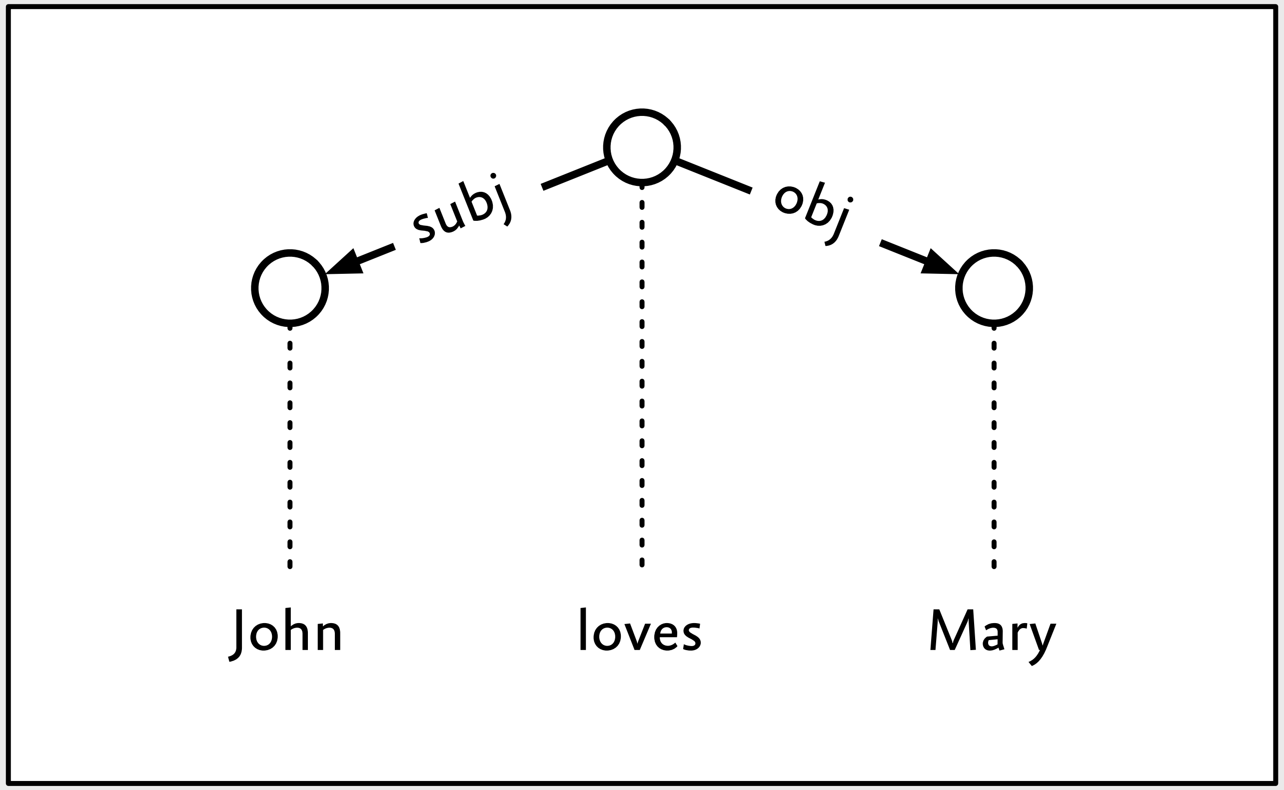
- **Manuel Bodirsky**
  does research in constraint solving at the Humboldt-Universit t zu Berlin
- **Robert Grabowski** and
  Mathias M hl
  are diploma students at
  Saarland University, Saarbr cken

Labelled drawings

* **Labelled drawings**
  are drawings equipped with
  two labelling functions.
* **Node labels** correspond
  to terminal symbols in LCFG.
* **Edge labels** correspond
  to non-terminal symbols.

$$\begin{aligned}
\mathsf{sentence} &\to \mathsf{subj}\ \text{loves}\ \mathsf{obj} \\
\mathsf{subj} &\to \text{John} \\
\mathsf{obj} &\to \text{Mary}
\end{aligned}$$

Labelled drawings



Labelled drawings

Properties

* Each context-free dependency language is a subset of the set of **projective** dependency structures.
* Context-free dependency languages are **closed under (consistent) permutation of subtrees**.