

# Tree-Adjoining Grammar: A tree-based constructionist grammar framework for natural language understanding

Timm Lichte & Laura Kallmeyer

University of Düsseldorf, Germany

AAAI Spring Symposium, March 27–29, 2017



## **Tree-Adjoining Grammar (TAG)**

- one of the major grammar formalisms (Müller [13])

## **Tree-Adjoining Grammar (TAG)**

- one of the major grammar formalisms (Müller [13])
- rich history, dates back to 1975 (Joshi, Levy & Takahashi [8])

## **Tree-Adjoining Grammar (TAG)**

- one of the major grammar formalisms (Müller [13])
- rich history, dates back to 1975 (Joshi, Levy & Takahashi [8])
- originally developed by engineers, further studied by theoretical computer scientists and computational linguists, finally discovered by linguists

## **Tree-Adjoining Grammar (TAG)**

- one of the major grammar formalisms (Müller [13])
- rich history, dates back to 1975 (Joshi, Levy & Takahashi [8])
- originally developed by engineers, further studied by theoretical computer scientists and computational linguists, finally discovered by linguists
- large implemented grammars for several languages (e. g. XTAG at UPenn)

## **Tree-Adjoining Grammar (TAG)**

- one of the major grammar formalisms (Müller [13])
- rich history, dates back to 1975 (Joshi, Levy & Takahashi [8])
- originally developed by engineers, further studied by theoretical computer scientists and computational linguists, finally discovered by linguists
- large implemented grammars for several languages (e. g. XTAG at UPenn)
- parsers, implementation tools, grammar induction tools, ...

## **Tree-Adjoining Grammar (TAG)**

- one of the major grammar formalisms (Müller [13])
- rich history, dates back to 1975 (Joshi, Levy & Takahashi [8])
- originally developed by engineers, further studied by theoretical computer scientists and computational linguists, finally discovered by linguists
- large implemented grammars for several languages (e. g. XTAG at UPenn)
- parsers, implementation tools, grammar induction tools, ...

## **Tree-Adjoining Grammar (TAG)**

- one of the major grammar formalisms (Müller [13])
- rich history, dates back to 1975 (Joshi, Levy & Takahashi [8])
- originally developed by engineers, further studied by theoretical computer scientists and computational linguists, finally discovered by linguists
- large implemented grammars for several languages (e. g. XTAG at UPenn)
- parsers, implementation tools, grammar induction tools, ...

## **Construction Grammar?**

- not really in the focus of the TAG community so far
- and that's surprising given the rather obvious connections!



# Aims of this talk

- present TAG as a grammar formalism that shares “major tenets” with (some versions of) Construction Grammar (following Goldberg [6]):
  - 1 **only surface structure**
  - 2 **grammatical constructions**
  - 3 **a network of constructions**

# Aims of this talk

- present TAG as a grammar formalism that shares “major tenets” with (some versions of) Construction Grammar (following Goldberg [6]):
  - 1 **only surface structure**
  - 2 **grammatical constructions**
  - 3 **a network of constructions**
- highlight some differences and commonalities regarding other explicit implementations of CxG (namely SBCG, ECG and FCG);

# Aims of this talk

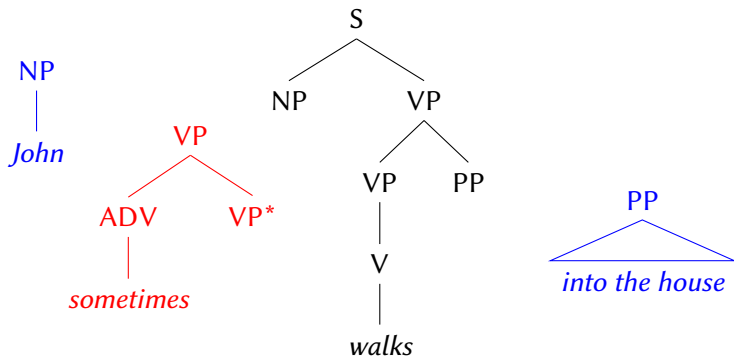
- present TAG as a grammar formalism that shares “major tenets” with (some versions of) Construction Grammar (following Goldberg [6]):
  - 1 **only surface structure**
  - 2 **grammatical constructions**
  - 3 **a network of constructions**
- highlight some differences and commonalities regarding other explicit implementations of CxG (namely SBCG, ECG and FCG);
- summarize computational applications based on TAG;

# Aims of this talk

- present TAG as a grammar formalism that shares “major tenets” with (some versions of) Construction Grammar (following Goldberg [6]):
  - 1 **only surface structure**
  - 2 **grammatical constructions**
  - 3 **a network of constructions**
- highlight some differences and commonalities regarding other explicit implementations of CxG (namely SBCG, ECG and FCG);
- summarize computational applications based on TAG;
- ask (and not answer) some general questions.

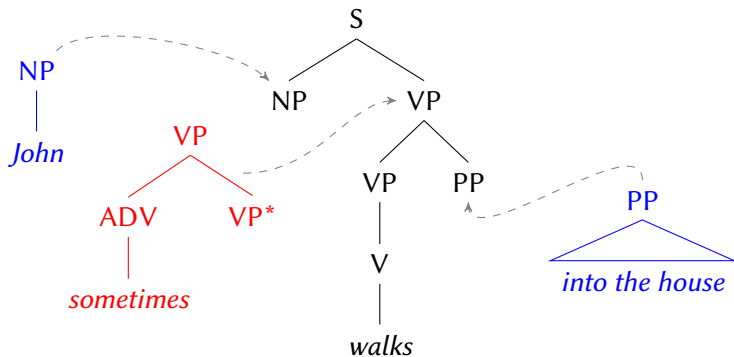
# TAG: basic ingredients

- a set of **elementary trees**



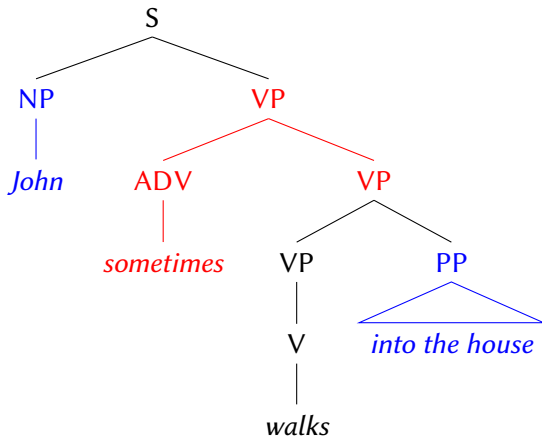
# TAG: basic ingredients

- a set of **elementary trees**
- two combinatorial operations:
  - substitution (replace a leaf node)
  - adjunction (replace an inner node)



# TAG: basic ingredients

- a set of **elementary trees**
- two combinatorial operations:
  - substitution (replace a leaf node)
  - adjunction (replace an inner node)



**In principal, the structure of elementary trees can be flexibly chosen!** (However, there are certain conventions as to how to represent, e.g., valency.)



**In principal, the structure of elementary trees can be flexibly chosen!** (However, there are certain conventions as to how to represent, e.g., valency.)

**Derivations are unordered!** (There is no prescribed order of when to apply substitution and adjunction.)

## TAG: some important properties

**In principal, the structure of elementary trees can be flexibly chosen!** (However, there are certain conventions as to how to represent, e.g., valency.)

**Derivations are unordered!** (There is no prescribed order of when to apply substitution and adjunction.)

**The structure of derivations resembles (but is not fully congruent with) dependency graphs!** (This holds if the tree design is reflective of certain linguistically motivated principles.)

# TAG: some important properties

**In principal, the structure of elementary trees can be flexibly chosen!** (However, there are certain conventions as to how to represent, e.g., valency.)

**Derivations are unordered!** (There is no prescribed order of when to apply substitution and adjunction.)

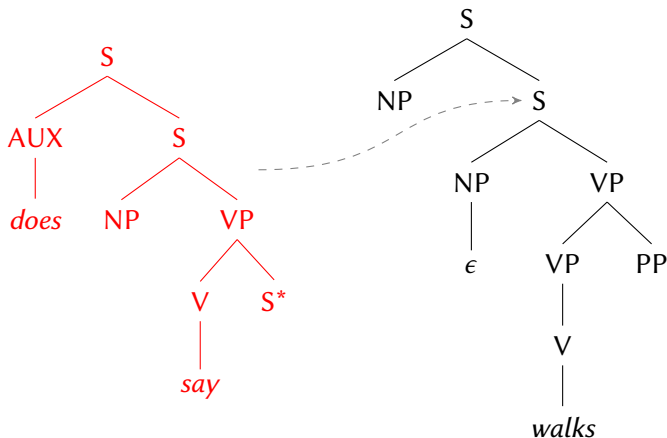
**The structure of derivations resembles (but is not fully congruent with) dependency graphs!** (This holds if the tree design is reflective of certain linguistically motivated principles.)

**There is an abundant number of TAG variants!** (For, e.g., free word order languages, extra flat trees, or incremental parsing)

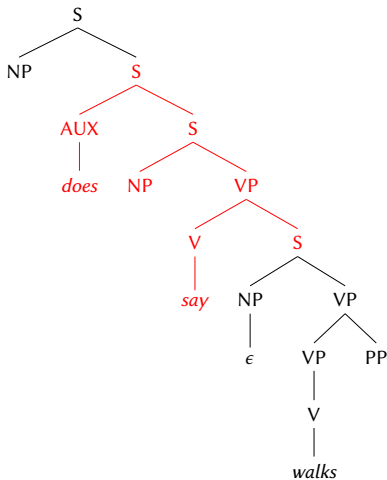
# TAG: only surface structure

By virtue of adjunction, cases of **long-distance dependencies** can be immediately captured:

(1) Who **does Mary say** sometimes walks into the house.



# TAG: only surface structure



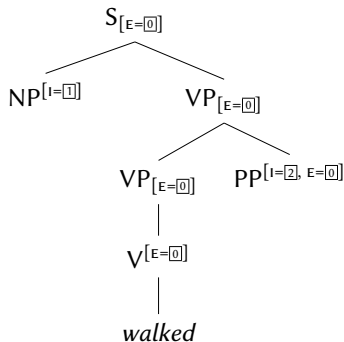
only surface structure



# TAG and frames

Kallmeyer & Osswald [10]:

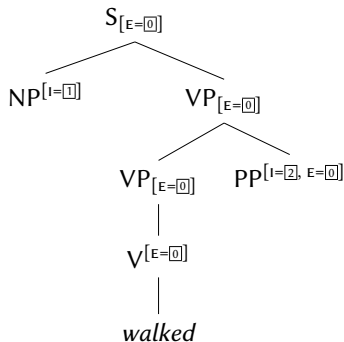
- lexicon: **pairs of elementary trees and frames**  
(= typed feature structures)



	<i>bounded-locomotion</i>
ACTOR	1
MOVER	1
GOAL	2
PATH	<i>path</i>
MANNER	<i>walking</i>

Kallmeyer & Osswald [10]:

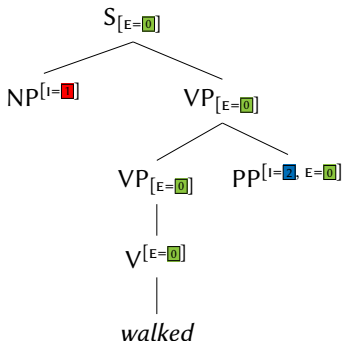
- lexicon: **pairs of elementary trees and frames**  
(= typed feature structures)
- Elementary trees are enriched with **interface features**, which contain base labels from the frame representation.
  - unification of interface features  $\leadsto$  unification of frames



<i>bounded-locomotion</i>	
ACTOR	1
MOVER	1
GOAL	2
PATH	<i>path</i>
MANNER	<i>walking</i>

Kallmeyer & Osswald [10]:

- lexicon: **pairs of elementary trees and frames**  
(= typed feature structures)
- Elementary trees are enriched with **interface features**, which contain base labels from the frame representation.
  - unification of interface features  $\leadsto$  unification of frames

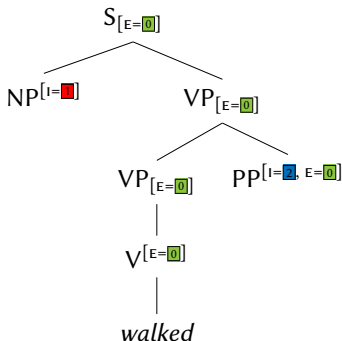


	<i>bounded-locomotion</i>
ACTOR	1
MOVER	1
GOAL	2
PATH	<i>path</i>
MANNER	<i>walking</i>



Kallmeyer & Osswald [10]:

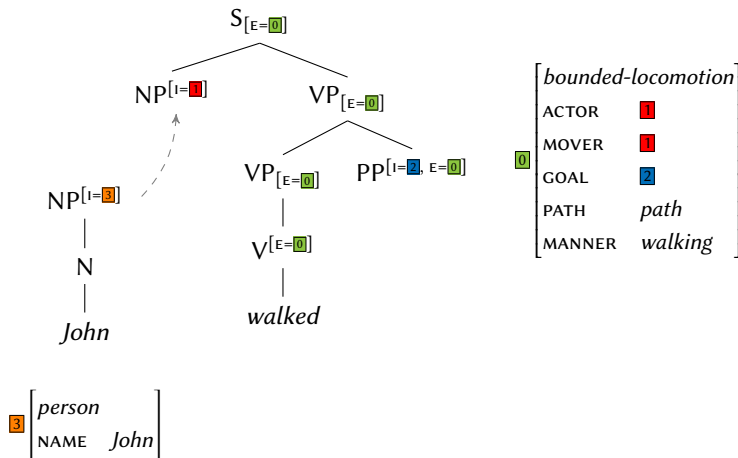
- lexicon: **pairs of elementary trees and frames**  
(= typed feature structures)
- Elementary trees are enriched with **interface features**, which contain base labels from the frame representation.
  - unification of interface features  $\leadsto$  unification of frames
- parallel composition of derived trees and larger frames



<i>bounded-locomotion</i>	
ACTOR	1
MOVER	1
GOAL	2
PATH	<i>path</i>
MANNER	<i>walking</i>

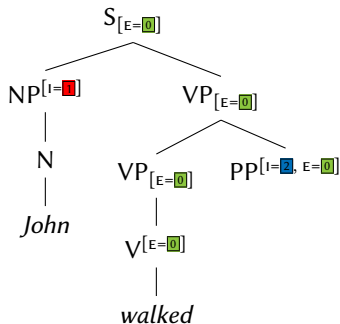
# TAG and frames: example

(1) John walked into the house.



# TAG and frames: example

(2) John walked into the house.

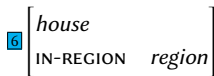
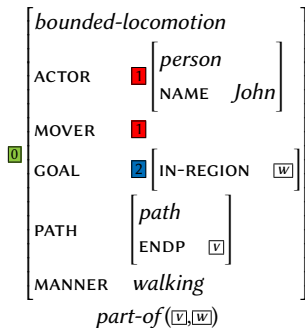
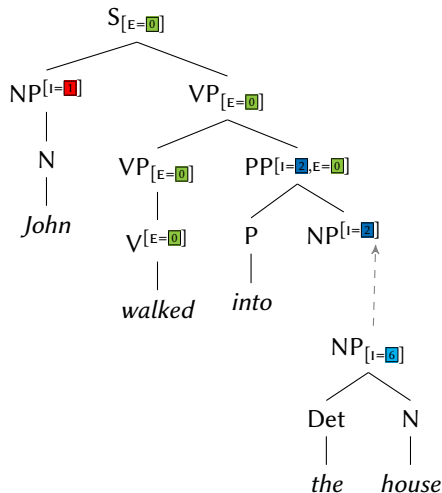


		<i>bounded-locomotion</i>
	ACTOR	<b>1</b> $\left[ \begin{array}{l} \textit{person} \\ \textit{NAME John} \end{array} \right]$
<b>0</b>	MOVER	<b>1</b>
	GOAL	<b>2</b>
	PATH	<i>path</i>
	MANNER	<i>walking</i>



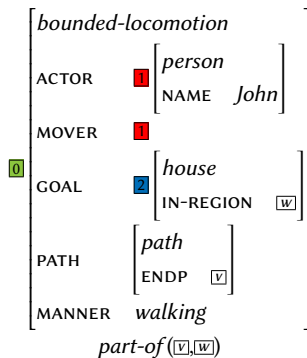
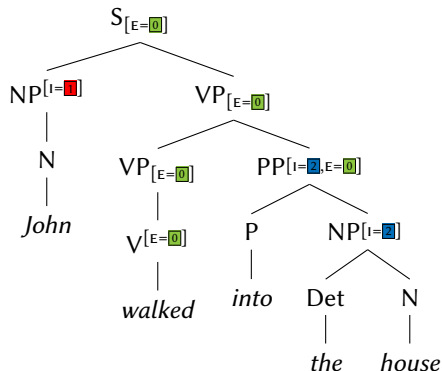
# TAG and frames: example

(2) John walked into the house.



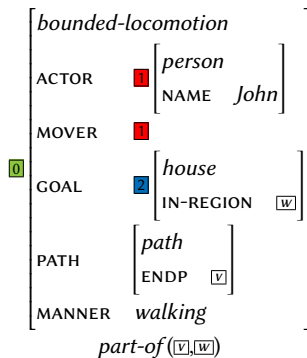
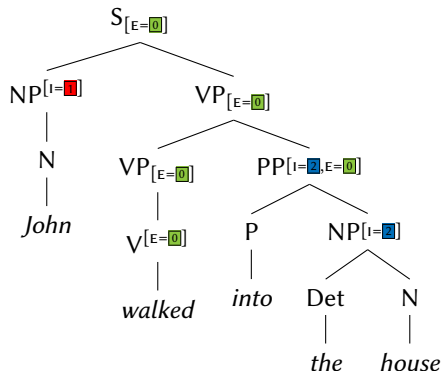
# TAG and frames: example

(2) John walked into the house.



# TAG and frames: example

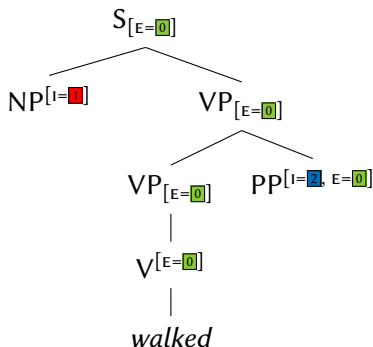
(2) John walked into the house.



Nice, but where are the constructions ???

# Constructions in TAG

Elementary trees:

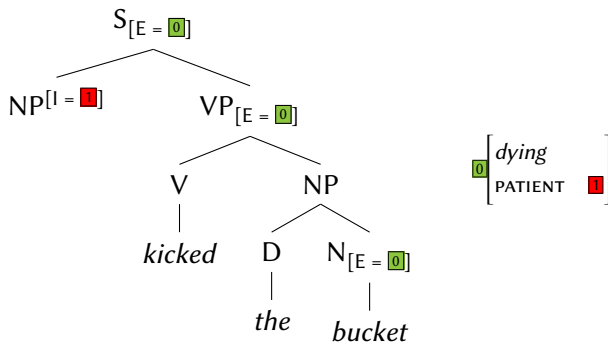


<i>bounded-locomotion</i>	
ACTOR	1
MOVER	1
GOAL	2
PATH	<i>path</i>
MANNER	<i>walking</i>



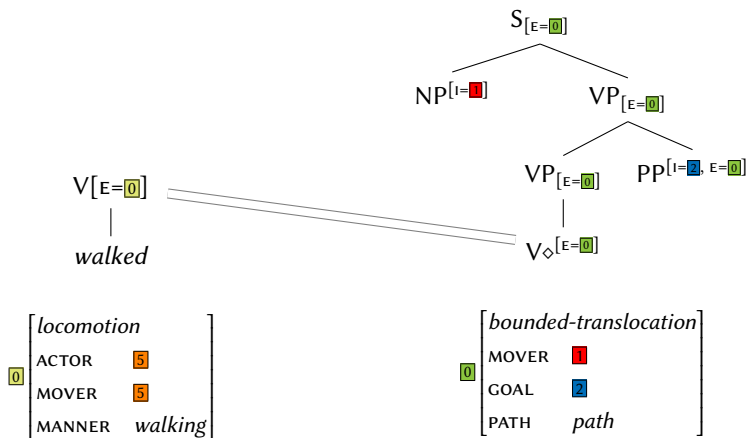
# Constructions in TAG

Elementary trees with multiple lexical anchors:



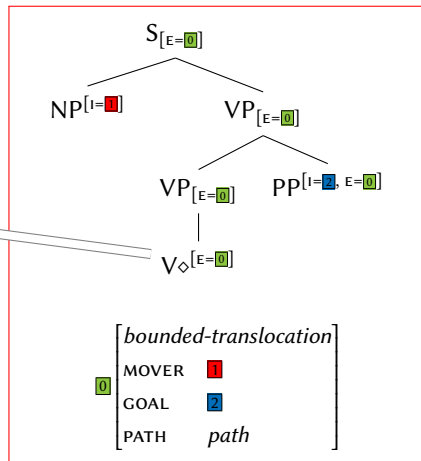
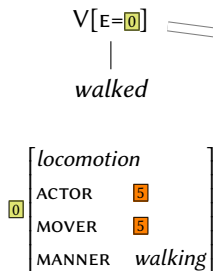
# Constructions in TAG

Unanchored elementary trees:



# Constructions in TAG

Unanchored elementary trees:

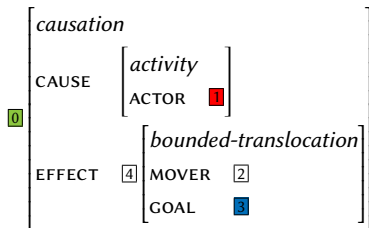
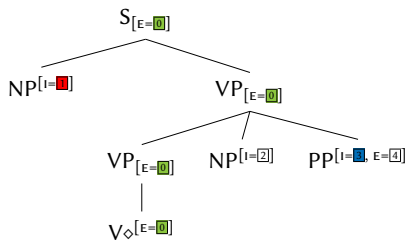


↑  
argument structure construction

# Constructions in TAG

Unanchored elementary trees:

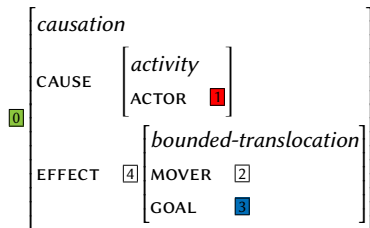
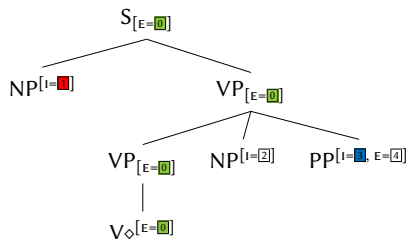
(2) John rolls the ball into the goal



# Constructions in TAG

Unanchored elementary trees:

(2) John rolls the ball into the goal



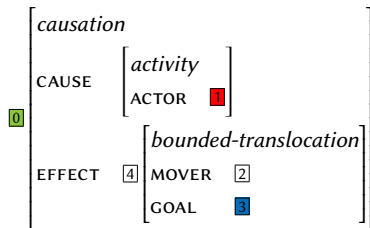
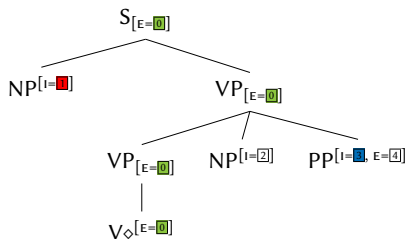
grammatical constructions




# Constructions in TAG

Unanchored elementary trees:

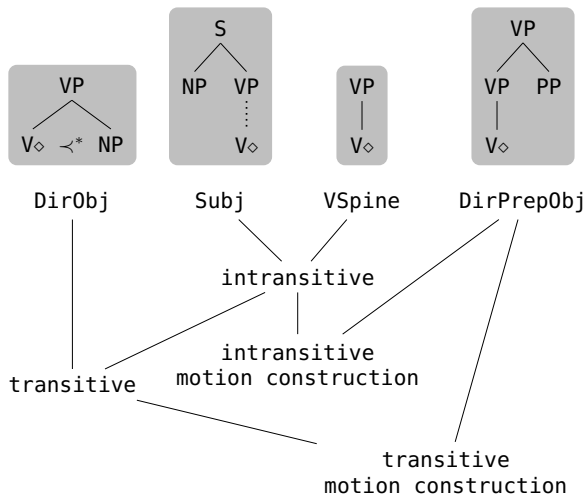
(2) John rolls the ball into the goal



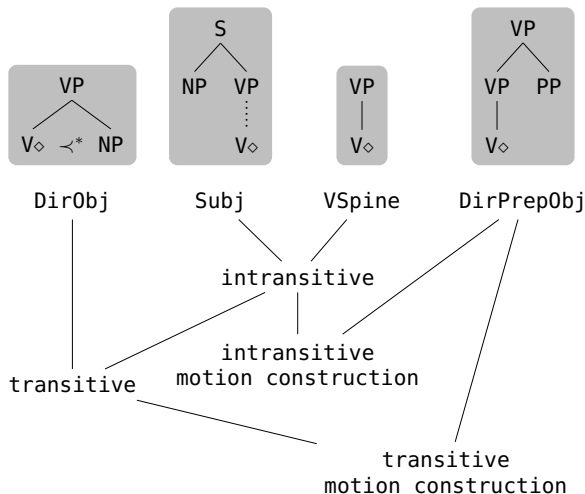
grammatical constructions 

a network of constructions ???

# Inheritance hierarchies in the metagrammar



# Inheritance hierarchies in the metagrammar



inheritance network of constructions





# TAG and other CxG incarnations

Where do grammar formalisms/frameworks/analyses **substantially** differ?

# TAG and other CxG incarnations

Where do grammar formalisms/frameworks/analyses **substantially** differ?

Treatment of non-local dependencies (“discontinuous constituents”)

- **Direct:** TAG, FCG
- **Indirect:** SBCG, ECG?

# TAG and other CxG incarnations

Where do grammar formalisms/frameworks/analyses **substantially** differ?

Treatment of non-local dependencies (“discontinuous constituents”)

- **Direct:** TAG, FCG
- **Indirect:** SBCG, ECG?

Stipulation of grammatical primitives (features & types)

- **Open (“fluid”):** FCG
- **Closed:** TAG, SBCG, ECG

# TAG and other CxG incarnations

Where do grammar formalisms/frameworks/analyses **substantially** differ?

Treatment of non-local dependencies (“discontinuous constituents”)

- **Direct:** TAG, FCG
- **Indirect:** SBCG, ECG?

Stipulation of grammatical primitives (features & types)

- **Open (“fluid”):** FCG
- **Closed:** TAG, SBCG, ECG

Other points of comparison:

- monotonic vs. non-monotonic
- declarative vs. procedural
- lexical vs. phrasal (Müller & Wechsler [14])

## **Manual grammar implementation and symbolic parsing**

- metagrammar compiler (XMG, Crabbé et al. [3])
- TuLiPA parser (Kallmeyer et al. [11])

## **Manual grammar implementation and symbolic parsing**

- metagrammar compiler (XMG, Crabbé et al. [3])
- TuLiPA parser (Kallmeyer et al. [11])

## **Grammar induction and probabilistic/data-driven parsing on multiple languages**

- English (Chiang [2]), German (Kaeshammer & Demberg [9]), Korean (Park [15])
- competitive results for constrained forms of TAG (Carreras, Collins & Koo [1], Hayashi, Suzuki & Nagata [7], Shen [16], Swanson et al. [17], and Yamangil & Shieber [18])

## **Manual grammar implementation and symbolic parsing**

- metagrammar compiler (XMG, Crabbé et al. [3])
- TuLiPA parser (Kallmeyer et al. [11])

## **Grammar induction and probabilistic/data-driven parsing on multiple languages**

- English (Chiang [2]), German (Kaeshammer & Demberg [9]), Korean (Park [15])
- competitive results for constrained forms of TAG (Carreras, Collins & Koo [1], Hayashi, Suzuki & Nagata [7], Shen [16], Swanson et al. [17], and Yamangil & Shieber [18])

## **Generation and bidirectionality**

- Gardent & Kow [5] and Kow, Parmentier & Gardent [12]

## **Manual grammar implementation and symbolic parsing**

- metagrammar compiler (XMG, Crabbé et al. [3])
- TuLiPA parser (Kallmeyer et al. [11])

## **Grammar induction and probabilistic/data-driven parsing on multiple languages**

- English (Chiang [2]), German (Kaeshammer & Demberg [9]), Korean (Park [15])
- competitive results for constrained forms of TAG (Carreras, Collins & Koo [1], Hayashi, Suzuki & Nagata [7], Shen [16], Swanson et al. [17], and Yamangil & Shieber [18])

## **Generation and bidirectionality**

- Gardent & Kow [5] and Kow, Parmentier & Gardent [12]

## **Psycholinguistically realistic incremental processing**

- Demberg, Keller & Koller [4]



# Summary

TAG incorporates central tenets of CxG:

only surface structure



grammatical constructions



inheritance network of constructions



# Summary

TAG incorporates central tenets of CxG:

only surface structure



grammatical constructions



inheritance network of constructions



TAG also differs substantially from other implementations of CxG.  
TAG comes with a wealth of software tools.

# Summary

TAG incorporates central tenets of CxG:

only surface structure ✓

grammatical constructions ✓

inheritance network of constructions ✓

TAG also differs substantially from other implementations of CxG.  
TAG comes with a wealth of software tools.

- ⇒ What is the formal core of constructionist approaches?
- ⇒ Where do constructionist approaches really make a difference?
  - different empirical predictions or theoretical ramifications?
  - benefit in terms of computational applications?

- [1] Carreras, Xavier, Michael Collins & Terry Koo. 2008. TAG, dynamic programming, and the perceptron for efficient, feature-rich parsing. In *Proceedings of the twelfth conference on computational natural language learning (CoNLL '08)*, 9–16. Manchester, United Kingdom.
- [2] Chiang, David. 2004. Statistical parsing with an automatically extracted Tree Adjoining Grammar. In Rens Bod, Remko Scha & Khalil Sima'an (eds.), *Data-oriented parsing*, 301–318. CSLI Publications.
- [3] Crabbé, Benoit, Denys Duchier, Claire Gardent, Joseph Le Roux & Yannick Parmentier. 2013. XMG: eXtensible MetaGrammar. *Computational Linguistics* 39(3). 591–629.
- [4] Demberg, Vera, Frank Keller & Alexander Koller. 2013. Incremental, predictive parsing with psycholinguistically motivated Tree-Adjoining Grammar. *Computational Linguistics* 39(4). 1025–1066.
- [5] Gardent, Claire & Eric Kow. 2005. Generating and selecting grammatical paraphrases. In *Enlg.* Aberdeen.
- [6] Goldberg, Adele. 2013. Constructionist approaches. In Thomas Hoffmann & Graeme Trousdale (eds.), *The Oxford handbook of Construction Grammar*, 15–31. Oxford, UK: Oxford Univ. Press.
- [7] Hayashi, Katsuhiko, Jun Suzuki & Masaaki Nagata. 2016. Shift-reduce spinal TAG parsing with dynamic programming. *Transactions of the Japanese Society for Artificial Intelligence*.
- [8] Joshi, Aravind K., Leon S. Levy & Masako Takahashi. 1975. Tree Adjunct Grammars. *Journal of Computer and System Science* 10. 136–163.
- [9] Kaeshammer, Miriam & Vera Demberg. 2012. German and english treebanks and lexica for Tree-Adjoining Grammars. In *Proceedings of the international conference on language resources and evaluation (Irec 2012)*. Istanbul.

- [10] Kallmeyer, Laura & Rainer Osswald. 2013. Syntax-driven semantic frame composition in Lexicalized Tree Adjoining Grammar. *Journal of Language Modelling* 1. 267–330.
- [11] Kallmeyer, Laura, Wolfgang Maier, Yannick Parmentier & Johannes Dellert. 2010. TuLiPA - parsing extensions of TAG with Range Concatenation Grammars. *Bulletin of the Polish Academy of Sciences* 58(3). 377–392.
- [12] Kow, Eric, Yannick Parmentier & Claire Gardent. 2006. SemTAG, the LORIA toolbox for TAG-based parsing and generation. In *Proceedings of the 8th international workshop on tree adjoining grammar and related formalisms*, 115–120. Sydney.
- [13] Müller, Stefan. 2016. *Grammatical theory: From Transformational Grammar to constraint-based approaches*. (Textbooks in Language Sciences 1). Berlin: Language Science Press. <http://langsci-press.org/catalog/book/25>.
- [14] Müller, Stefan & Stephen M. Wechsler. 2014. Lexical approaches to argument structure. *Theoretical Linguistics* 40(1–2). 1–76.  
<http://hpsg.fu-berlin.de/~stefan/Pub/arg-st.html>.
- [15] Park, Jungyeul. 2006. Extraction of Tree Adjoining Grammars from a treebank for Korean. In *Proceedings of the COLING/ACL 2006 student research workshop*, 73–78. Sydney, Australia: Association for Computational Linguistics.  
<http://www.aclweb.org/anthology/P/P06/P06-3013>.
- [16] Shen, Libin. 2006. *Statistical LTAG parsing*. University of Pennsylvania dissertation.
- [17] Swanson, Ben, Elif Yamangil, Eugene Charniak & Stuart Shieber. 2013. A context free TAG variant. In *Proceedings of the 51st annual meeting of the association for computational linguistics (volume 1: long papers)*, 302–310. Sofia, Bulgaria: Association for Computational Linguistics.

- [18] Yamangil, Elif & Stuart M. Shieber. 2012. Estimating compact yet rich Tree Insertion Grammars. In *Proceedings of the 50th annual meeting of the Association for Computational Linguistics*, 110–114. Jeju, Republic of Korea.