Constructing a Construction Grammar with LTAG: Linguistic and Computational Perspectives

Timm Lichte & Laura Kallmeyer

University of Düsseldorf, Germany

ICCG8, September 03, 2014

SFB 991

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Preface

LTAG (= Lexicalized Tree-Adjoining Grammar)

- one of the major grammar formalisms (Müller, 2014)
- rich history, dates back to 1975 (Joshi et al., 1975)
- 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 LTAG community so far
- and that's surprising given the rather obvious connections!

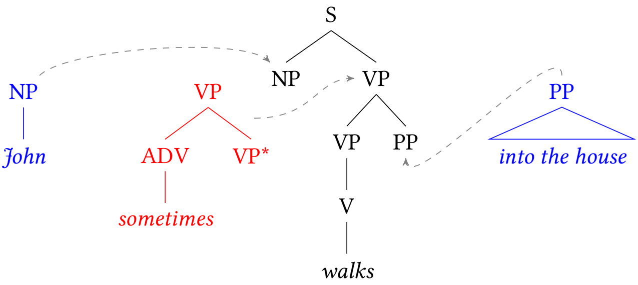
Aims and overview

Aims of this talk:

- present Lexicalized Tree-Adjoining Grammmar (LTAG) as a grammar formalism that shares central ideas with (some versions of) Construction Grammar (CxG):
  1. **grammatical constructions**
  2. **only surface structure**: no transformational or derivational component
  3. **a network of constructions** "which nodes are related by inheritance links" (Goldberg, 2013)
- show that it substantially differs from other explicit implementations of CxG, namely Sign-based Construction Grammar (SBCG), and Fluid Construction Grammar (FCG).

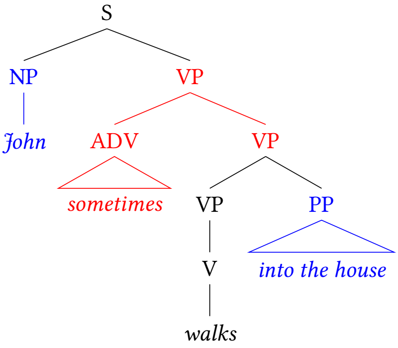
LTAG: basic ingredients

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



LTAG: basic ingredients

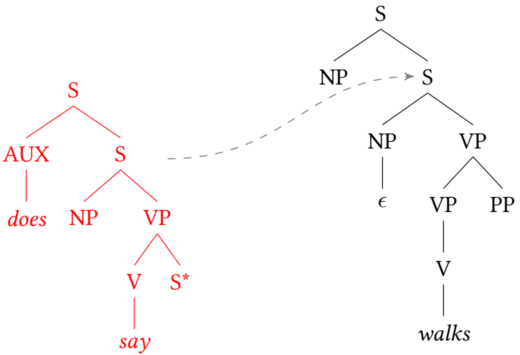
- a set of **elementary trees**
- two combinatorial operations:
  - substitution (replace a leaf node)
  - adjunction (replace an inner node)



LTAG: long distance dependencies

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

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

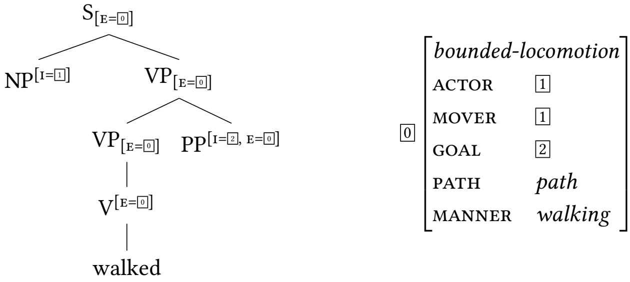


only surface structure ✓

LTAG and frames

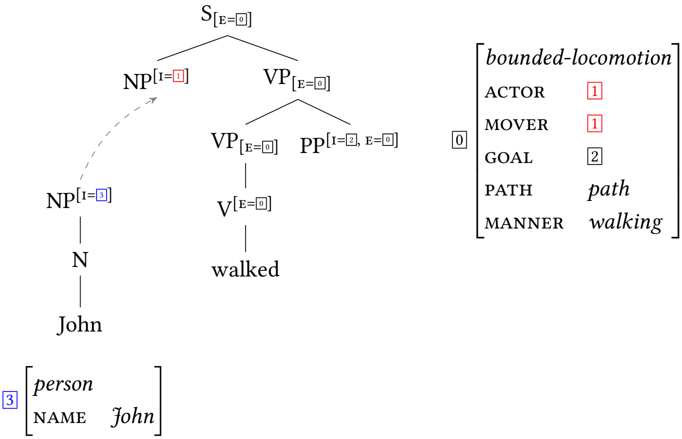
Kallmeyer & Osswald (2013):

- 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 ⇝ unification of frames
- parallel composition of derived trees and larger frames



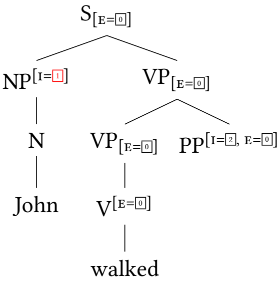
LTAG and frames: example

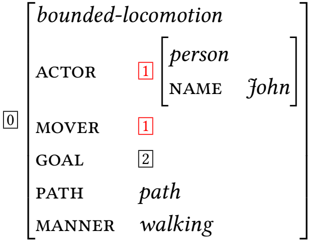
(2) John walked into the house.



LTAG and frames: example

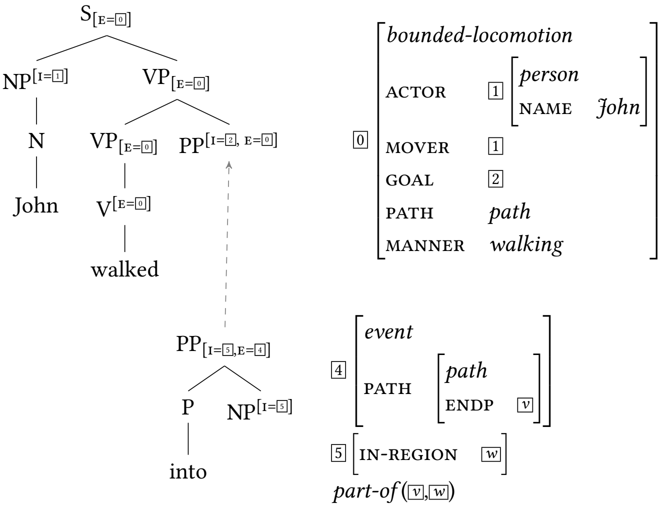
(2) John walked into the house.





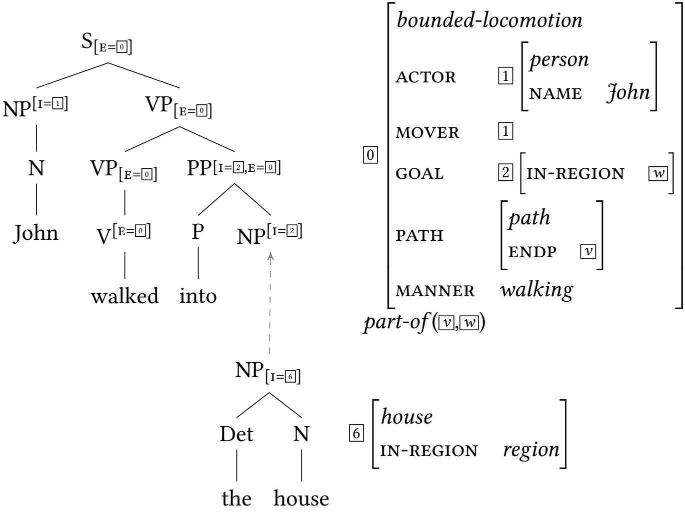
LTAG and frames: example

(2) John walked into the house.

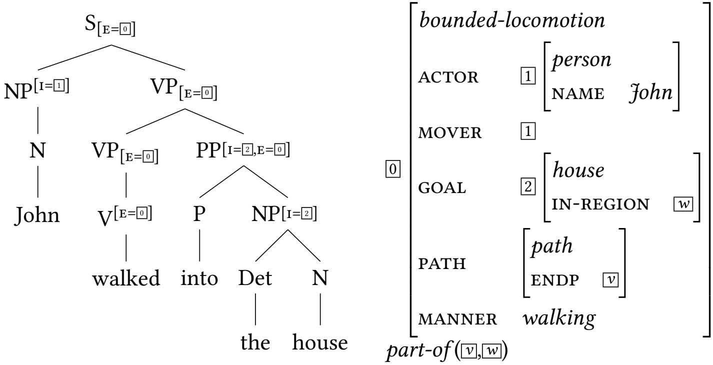


LTAG and frames: example

(2) John walked into the house.



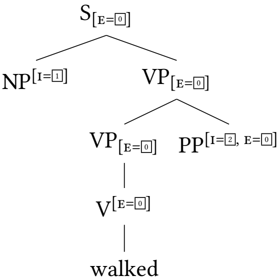
(2) John walked into the house.



Nice, but where are the constructions ???

Constructions in LTAG

Elementary trees:

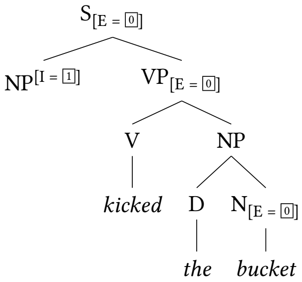


0

bounded-locomotion	
ACTOR	1
MOVER	1
GOAL	2
PATH	*path*
MANNER	*walking*

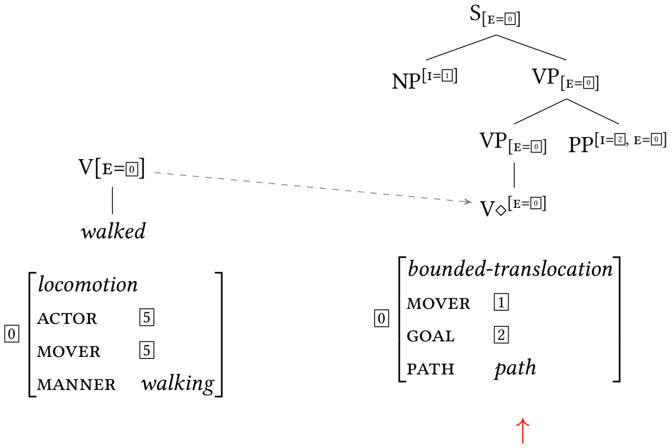
Constructions in LTAG

Elementary trees with multiple lexical anchors:



Constructions in LTAG

Lexical anchoring:

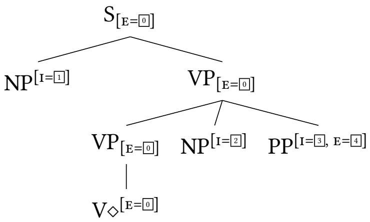


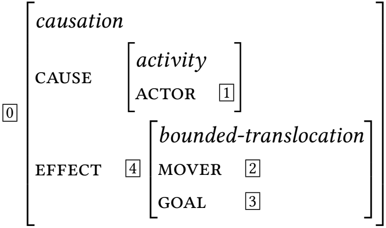
↑ argument structure construction

Constructions in LTAG

Transitive motion construction:

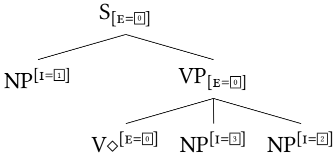
(3) John rolls the ball into the goal





Dative alternation: DO and PO construction

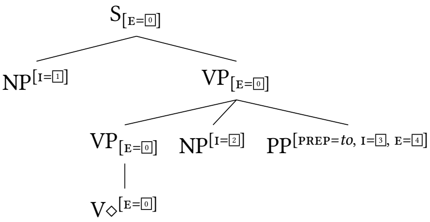
(4) John gives/sends Mary the book

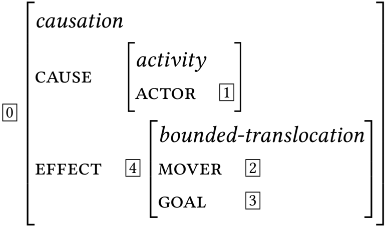


$$\boxed{0}\begin{bmatrix} \textit{causation} & \\ \text{CAUSE} & \begin{bmatrix} \textit{activity} & \\ \text{ACTOR} & \boxed{1} \end{bmatrix} \\ \text{EFFECT} & \begin{bmatrix} \textit{change-of-possession} & \\ \text{THEME} & \boxed{2} \\ \text{RECIPIENT} & \boxed{3} \end{bmatrix} \end{bmatrix}$$

Constructions in LTAG

(5) John gives/sends the book to Mary



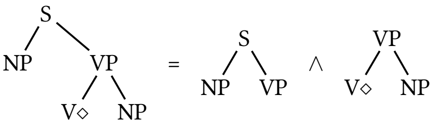


grammatical constructions

a network of constructions ???

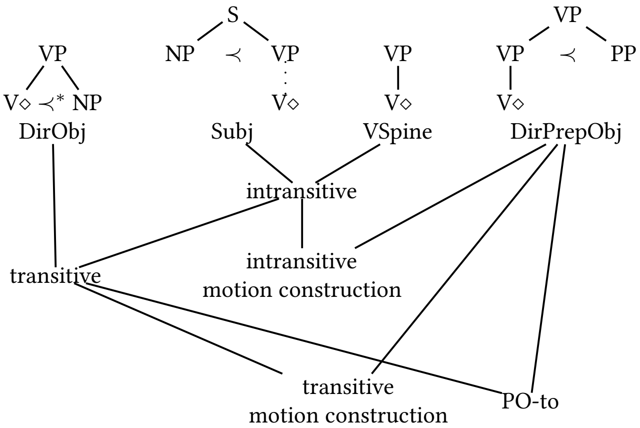
Inheritance hierarchies and metagrammatical factorization

- In order to produce and maintain a consistent LTAG of a considerable coverage, one uses a **metagrammar** (MG, Candito 1996; Crabbé & Duchier 2005).
- An MG contains factorized descriptions of unanchored elementary trees. It defines a set of tree fragments (MG classes) that can be used in other MG classes.
- This way, an unachored elementary tree family is the denotation of an MG class that makes use of a series of other, smaller tree fragments in the MG.



- Advantage of MGs for TAG from a linguistic point of view: The MG allows to express and implement lexical generalizations.

Class hierarchy in the MG (fragment):



inheritance network of constructions ✓

Points of comparison

Fundamental distinction between two classes of grammar frameworks:

- limited domain of locality (LDL)
  - list-like valency that is processed stepwise
  - movement, type raising, valency merge
  - examples: CG, (binarized) HPSG, SBCG, MG
- extended domain of locality (EDL)
  - set-like valency without predetermined order
  - capability to immediately access arbitrarily distant parts of a sentence within one lexical entry or syntactic rule
  - examples: LTAG, RRG, *some* versions of CxG, Dependency Grammar

Another recently discussed distinction that is orthogonal:

- lexical vs. phrasal (Müller & Wechsler, 2014)

Comparison

Lexicalized Tree-Adjoining Grammar:

- EDL
- tree rewriting + unification of typed feature structures
- inheritance network based on classes of the metagrammar

Sign-based construction grammar:

- LDL
- constraint-based architecture à la HPSG
- inheritance network based on types

Fluid Construction Grammar:

- EDL
- “match” (of conditional parts) and “merge” (of contributional parts) on non-functional untyped feature structures
- no inheritance, but conditioned unifiability

Summary

LTAG incorporates central ideas of CxG:

only surface structure ✓

grammatical constructions ✓

inheritance network of constructions ✓

LTAG differs substantially from other implementations of CxG.

⇒ different empirical predictions or theoretical ramifications?

Candito, Marie-Hélène. 1996. A principle-based hierarchical representation of LTAGs. In *Proceedings of the 16th international conference on computational linguistics (COLING 96)*, Copenhagen. `http://aclweb.org/anthology-new/C/C96/C96-1034.pdf`.

Crabbé, Benoit & Denys Duchier. 2005. Metagrammar redux. In Henning Christiansen, Peter Rossen Skadhauge & Jørgen Villadsen (eds.), *Constraint solving and language processing*, vol. 3438 Lecture Notes in Computer Science, 32–47. Springer.

Goldberg, Adele. 2013. Constructionist approaches. In Thomas Hoffmann & Graeme Trousdale (eds.), *The Oxford handbook of Construction Grammar*, 15–31. Oxford, UK: Oxford Univ. Press.

Joshi, Aravind K., Leon S. Levy & Masako Takahashi. 1975. Tree Adjunct Grammars. *Journal of Computer and System Science* 10. 136–163.

Kallmeyer, Laura & Rainer Osswald. 2013. Syntax-driven semantic frame composition in Lexicalized Tree Adjoining Grammar. *Journal of Language Modelling* 1. 267–330.

Müller, Stefan. 2014. *Grammatical theory: From transformational grammar to constraint-based approaches* (Lecture Notes in Language Sciences 1). Berlin: Language Science Press. `http://hpsg.fu-berlin.de/~stefan/Pub/grammatical-theory.html`. In Preparation.

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`.