Tree Adjoining Grammars

TAG: The syntax-semantics interface

Laura Kallmeyer & Benjamin Burkhardt

HHU Düsseldorf

WS 2017/2018

Outline

1. LTAG semantics: Overview
    - Synchronous TAGs for semantics
    - Unification-based LTAG semantics with predicate logic
    - Unification-based LTAG semantics with frames
2. Introduction to frame semantics
3. Case study: Directed motion construction

LTAG semantics: overview

Goal: an LTAG architecture of the syntax-semantics interface that

- is compositional: the meaning of a complex expression can be computed from the meaning of its subparts and its composition operation.
- pairs entire elementary trees with meaning components.

LTAG semantics: overview

Three principal approaches:

1. LTAG semantics with synchronous TAG (STAG) (Shieber, 1994; Nesson and Shieber, 2006, 2008)
2. Unification based LTAG semantics with predicate logic (Kallmeyer and Joshi, 2003; Gardent and Kallmeyer, 2003; Kallmeyer and Romero, 2008)
3. Unification based LTAG semantics with frames (Kallmeyer and Osswald, 2013; Kallmeyer et al., 2016)

We will use the third approach in this course and only briefly present the other two.

LTAG semantics: STAG

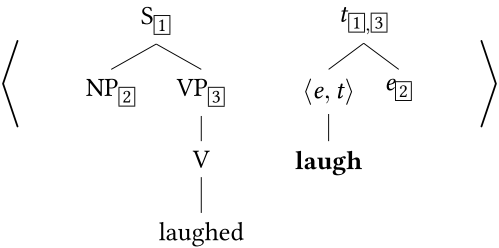
Idea:

- pair two TAGs, one for syntax and one for L(ogical) F(orm) (= typed predicate logic),
- and do derivations in parallel.

Formalism used for this: synchronous TAG (STAG) Shieber and Schabes (1990); Shieber (1994).

STAG = two TAGs G_1, G_2 whose trees are related to each other. More precisely, it contains pairs $\langle \gamma_1, \gamma_2, link \rangle$ where γ_1 is an elementary tree from G_1, γ_2 an elementary tree from G_2, and *link* is a set of pairs of node addresses from γ_1 and γ_2 respectively.

LTAG semantics: STAG



(The links are depicted with boxed numbers.)

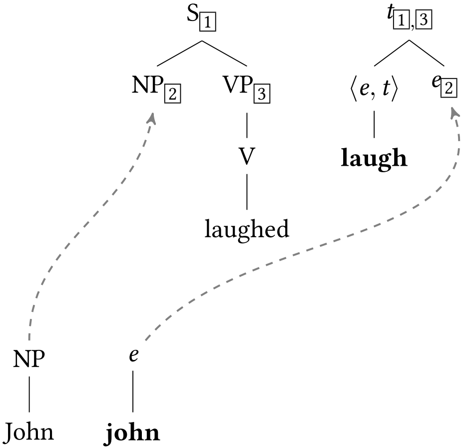
- The non-terminals of the semantic TAG are types t, e, $\langle e, t\rangle$, ...
- The semantic TAG describes the syntactic structure of typed predicate logical formulas.
- The links in this example tell us, for instance, that the subject NP corresponds to the e argument of **laugh**.

LTAG semantics: STAG

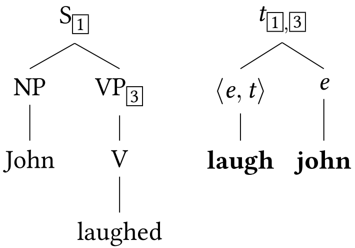
STAG derivation proceeds as in TAG, except that all operations must be paired: In every derivation step:

- A new elementary tree pair $\langle \gamma_1, \gamma_2 \rangle$ is picked.
- γ_1 is attached (substituted or adjoined) to the syntactic tree while γ_2 is attached to the semantic tree.
- The nodes that the two trees attach to must be linked.
- The link that is used in this derivation step disappears while all other links involving the attachment sites are inherited by the root of the attaching tree.

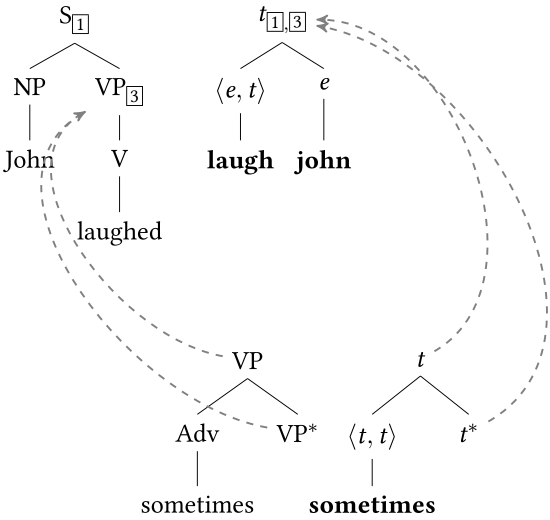
LTAG semantics: STAG



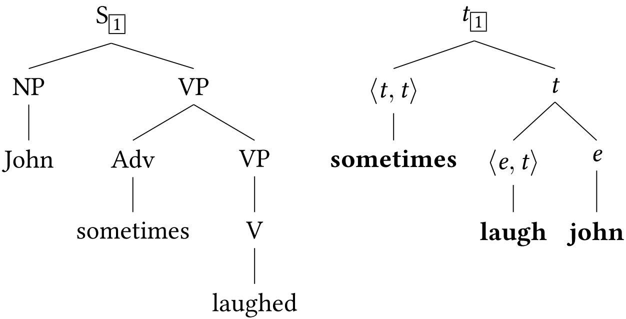
LTAG semantics: STAG



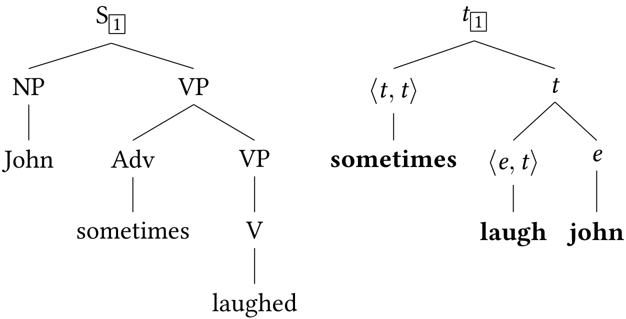
LTAG semantics: STAG



LTAG semantics: STAG



LTAG semantics: STAG



Logical form: **sometimes**(**laugh**(**john**))

Unification-based LTAG semantics with predicate logic

Kallmeyer and Romero (2008), Gardent and Kallmeyer (2003):
Syntax-Semantics Interface for LTAG

Idea: Each elementary tree is paired with

Unification-based LTAG semantics with predicate logic

Kallmeyer and Romero (2008), Gardent and Kallmeyer (2003):
Syntax-Semantics Interface for LTAG

Idea: Each elementary tree is paired with

- A set of typed predicate logic expressions and of scope constraints (i.e., constraints on sub-term relations)

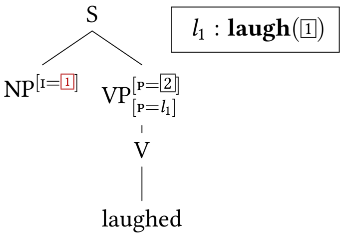
Unification-based LTAG semantics with predicate logic

Kallmeyer and Romero (2008), Gardent and Kallmeyer (2003):
Syntax-Semantics Interface for LTAG

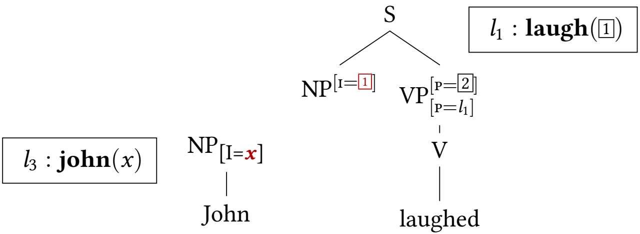
Idea: Each elementary tree is paired with

- A set of typed predicate logic expressions and of scope constraints (i.e., constraints on sub-term relations)
- interface features that characterizes a) which arguments need to be filled, b) which elements are available as arguments for other elementary trees and c) the scope behaviour.
  The features are linked to positions in the elementary tree.

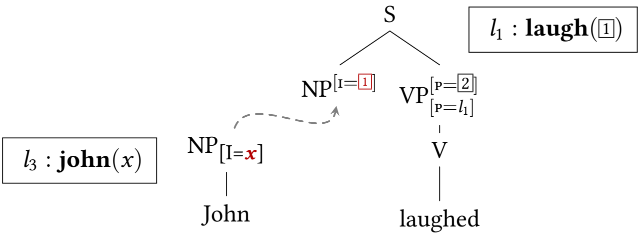
Unification-based LTAG semantics with predicate logic



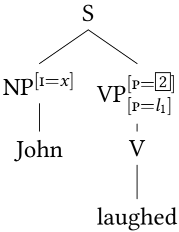
Unification-based LTAG semantics with predicate logic



Unification-based LTAG semantics with predicate logic

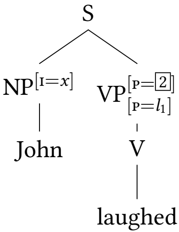


Unification-based LTAG semantics with predicate logic

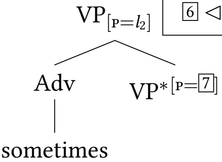


$$\begin{array}{l} l_1 : \textbf{laugh}(x), \\ l_3 : \textbf{john}(x) \end{array}$$

Unification-based LTAG semantics with predicate logic

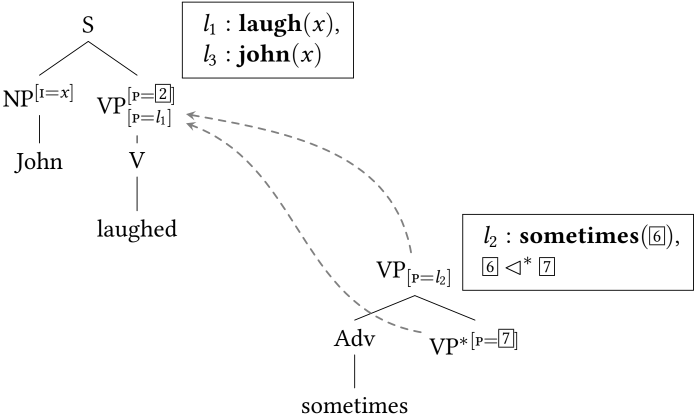


l_1 : **laugh**(x),
l_3 : **john**(x)

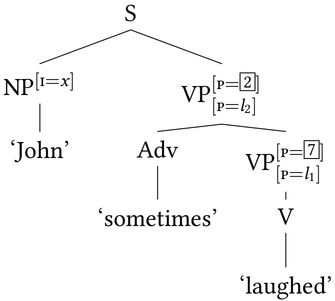


l_2 : **sometimes**$(\boxed{6})$,
$\boxed{6} \lhd^* \boxed{7}$

Unification-based LTAG semantics with predicate logic

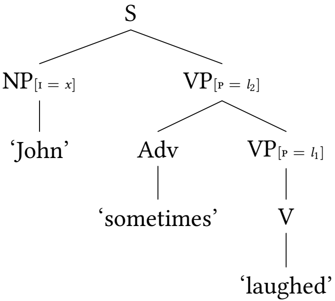


Unification-based LTAG semantics with predicate logic



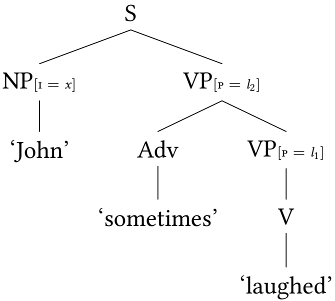
l_1 : $\mathbf{laugh}(x)$,
l_3 : $\mathbf{john}(x)$,
l_2 : $\mathbf{sometimes}(\boxed{6})$,
$\boxed{6} \lhd^* \boxed{7}$

Unification-based LTAG semantics with predicate logic



l_1 : **laugh**(x),
l_3 : **john**(x),
l_2 : **sometimes**$(\boxed{6})$,
$\boxed{6} \lhd^* l_1$

Unification-based LTAG semantics with predicate logic



l_1 : $\mathbf{laugh}(x)$,
l_3 : $\mathbf{john}(x)$,
l_2 : $\mathbf{sometimes}(\boxed{6})$,
$\boxed{6} \lhd^* l_1$

$\boxed{6} \lhd^* l_1$ signifies that the formula labeled l_1 is a subformula of the formula that has to be placed in the hole $\boxed{6}$.

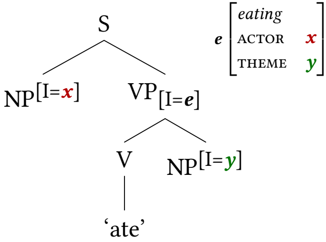
Disambiguation leads to $\mathbf{john}(x) \wedge \mathbf{sometimes}(\mathbf{laugh}(x))$

Unification-based LTAG semantics with frames

- Semantic representations are linked to entire elementary trees (as in the previous approaches).
- Semantic representations: frames, expressed as typed feature structures.
- Interface features relate nodes in the syntactic tree to nodes in the frame graph.
- Frame composition by unification, triggered by the unifications on the interface features that are in turn triggered by substitution, adjunction and final top-bottom unification on the derived tree.

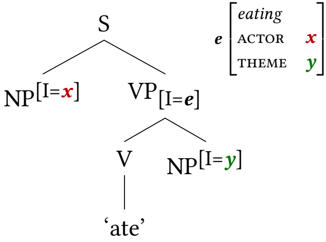
Unification-based LTAG semantics with frames

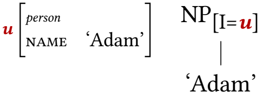
(1) Adam ate an apple.



Unification-based LTAG semantics with frames

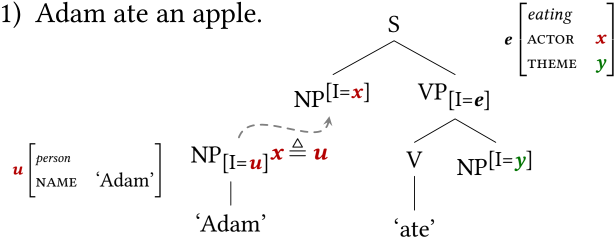
(1) Adam ate an apple.





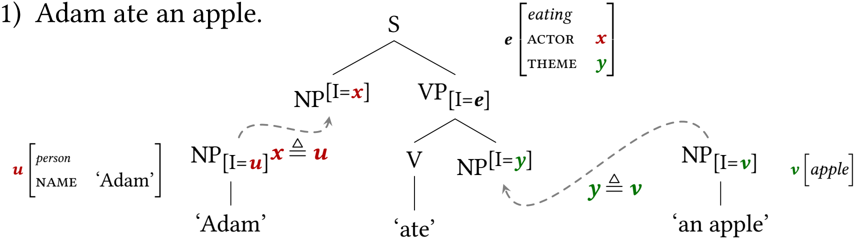
Unification-based LTAG semantics with frames

(1) Adam ate an apple.



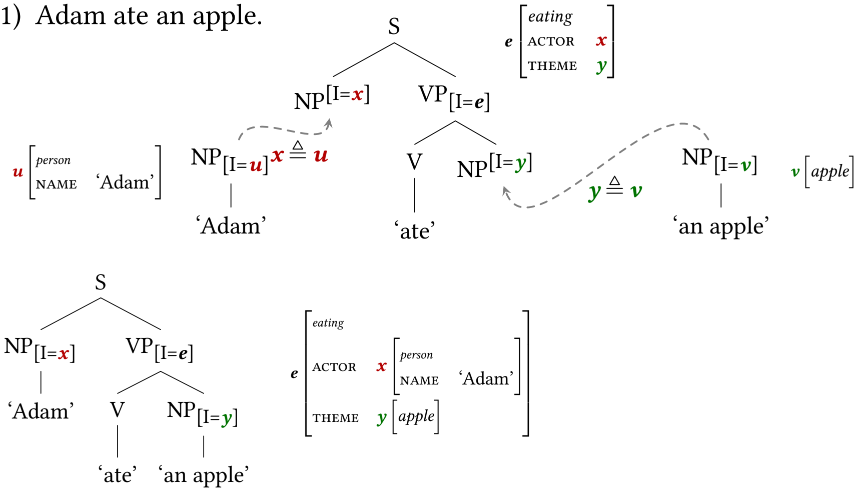
Unification-based LTAG semantics with frames

(1) Adam ate an apple.



Unification-based LTAG semantics with frames

(1) Adam ate an apple.



Introduction to frame semantics

Frames as used in LTAG

- A representation format for **rich lexical** and **constructional content**.
- Can nicely capture **semantic composition** and **decomposition**.
- Can be formalized as **generalized feature structures** with **types**, **relations** and **node labels**.

Introduction to frame semantics

Frames as used in LTAG

- A representation format for **rich lexical** and **constructional content**.
- Can nicely capture **semantic composition** and **decomposition**.
- Can be formalized as **generalized feature structures** with **types**, **relations** and **node labels**.

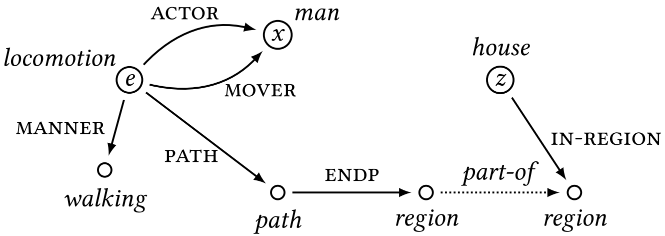
Basic assumptions

- **Attributes** (features, functional roles/relations) play a central role in the organization of semantic and conceptual knowledge and representation.
- Semantic components (participants, subevents) can be (recursively) addressed via **attributes** (from some "**base**" node).

$\rightsquigarrow$ inherently **structured representations** (models); composition by **unification** (under **constraints**)

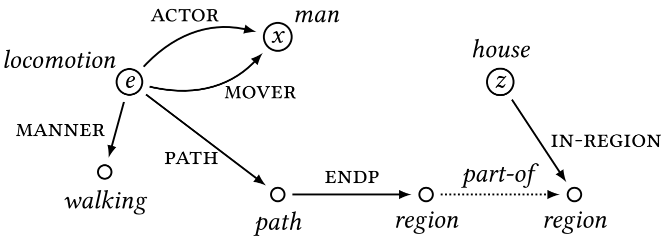
Introduction to frame semantics

Example



Introduction to frame semantics

Example

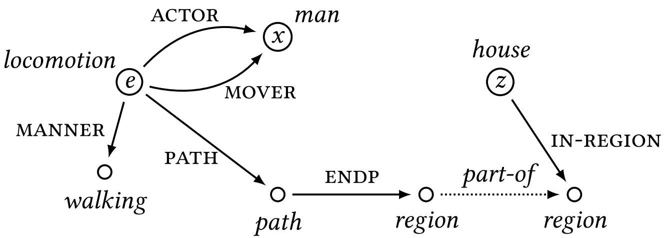


Ingredients

- Attributes (funct. relations): ACTOR, MOVER, PATH, MANNER, IN-REGION, …

Introduction to frame semantics

Example

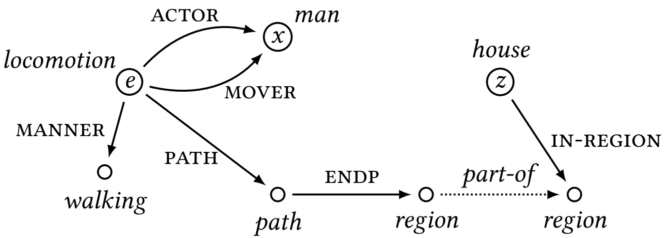


Ingredients

- Attributes (funct. relations): ACTOR, MOVER, PATH, MANNER, IN-REGION, …
- Type symbols: *locomotion*, *man*, *path*, *walking*, *region*, …

Introduction to frame semantics

Example

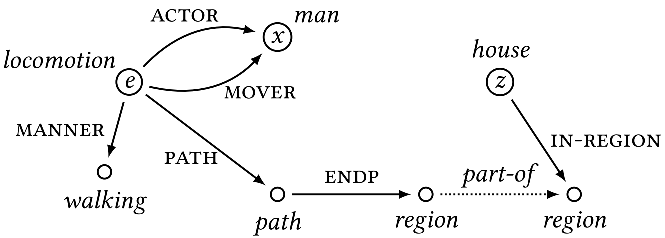


Ingredients

- Attributes (funct. relations): ACTOR, MOVER, PATH, MANNER, IN-REGION, ...
- Type symbols: *locomotion*, *man*, *path*, *walking*, *region*, ...
- Proper relations: *part-of*

Introduction to frame semantics

Example

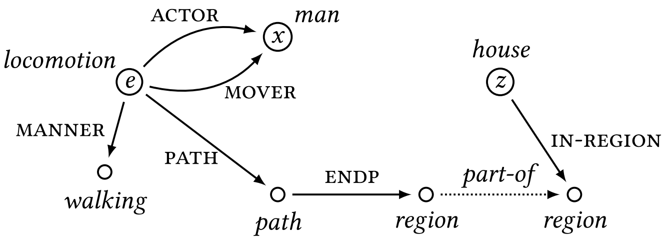


Ingredients

- Attributes (funct. relations): ACTOR, MOVER, PATH, MANNER, IN-REGION, …
- Type symbols: *locomotion*, *man*, *path*, *walking*, *region*, …
- Proper relations: *part-of*
- Node labels (variables): *e*, *x*, *z*

Introduction to frame semantics

Example



Ingredients

- Attributes (funct. relations): ACTOR, MOVER, PATH, MANNER, IN-REGION, ...
- Type symbols: *locomotion*, *man*, *path*, *walking*, *region*, ...
- Proper relations: *part-of*
- Node labels (variables): e, x, z

Core property

- Every node is reachable from some labeled "base" node via attributes.

Introduction to frame semantics

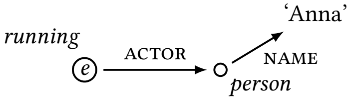
Example

(2) Anna ran

Introduction to frame semantics

Example

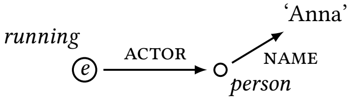
(2) Anna ran



Introduction to frame semantics

Example

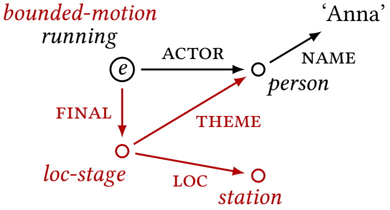
(2) Anna ran to the station.



Introduction to frame semantics

Example

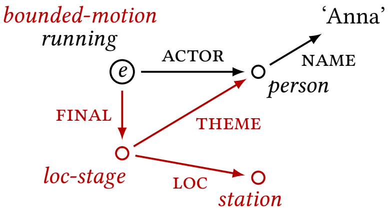
(2) Anna ran to the station.



Introduction to frame semantics

Example

(2) Anna ran to the station.

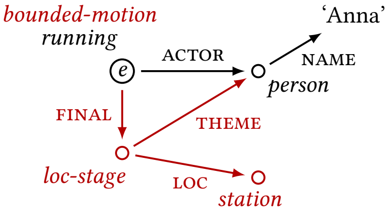


$$E \begin{bmatrix} \textit{running} \wedge \textit{bounded-motion} & \\ \textsc{actor} & \boxed{1} \begin{bmatrix} \textit{person} & \\ \textsc{name} & \text{‘Anna’} \end{bmatrix} \\ \textsc{final} & \begin{bmatrix} \textit{loc-stage} & \\ \textsc{theme} & \boxed{1} \\ \textsc{loc} & [\textit{station}] \end{bmatrix} \end{bmatrix}$$

Introduction to frame semantics

Example

(2) Anna ran to the station.



$$E \left[\begin{array}{lll} \multicolumn{3}{l}{\textit{running} \wedge \textit{bounded-motion}} \\ \text{ACTOR} & \boxed{1} & \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{person}} \\ \text{NAME} & \text{'Anna'} \end{array}\right] \\ \text{FINAL} & & \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{loc-stage}} \\ \text{THEME} & \boxed{1} \\ \text{LOC} & [\textit{station}] \end{array}\right] \end{array}\right]$$

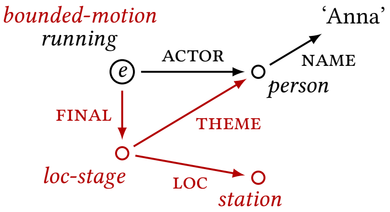
Attribute-value logic

$$e \cdot (\textit{running} \wedge \textit{bounded-motion} \wedge \text{ACTOR} : (\textit{person} \wedge \text{NAME} \triangleq \text{'Anna'}) \\ \text{ACTOR} \doteq \text{FINAL THEME} \wedge \text{FINAL} : (\textit{loc-stage} \wedge \text{LOC} : \textit{station}))$$

Introduction to frame semantics

Example

(2) Anna ran to the station.



$$E\begin{bmatrix} \textit{running} \wedge \textit{bounded-motion} & \\ \text{ACTOR} & \boxed{1}\begin{bmatrix} \textit{person} & \\ \text{NAME} & \text{`Anna'} \end{bmatrix} \\ \text{FINAL} & \begin{bmatrix} \textit{loc-stage} & \\ \text{THEME} & \boxed{1} \\ \text{LOC} & [\textit{station}] \end{bmatrix} \end{bmatrix}$$

Attribute-value logic

$$e \cdot (\textit{running} \wedge \textit{bounded-motion} \wedge \text{ACTOR} : (\textit{person} \wedge \text{NAME} \triangleq \text{`Anna'})$$
$$\text{ACTOR} \doteq \text{FINAL THEME} \wedge \text{FINAL} : (\textit{loc-stage} \wedge \text{LOC} : \textit{station}))$$

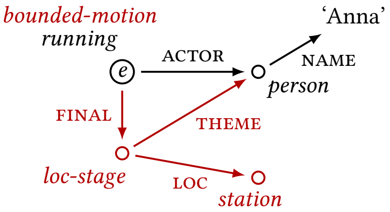
Translation into first-order logic

$$\exists x \exists s \exists y (\textit{running}(e) \wedge \textit{bounded-motion}(e) \wedge \text{ACTOR}(e, x) \wedge \textit{person}(x) \wedge \text{NAME}(x, \text{`Anna'})$$
$$\wedge\ \text{FINAL}(e, s) \wedge \textit{loc-stage}(s) \wedge \text{THEME}(s, x) \wedge \text{LOC}(s, y) \wedge \textit{station}(y))$$

Introduction to frame semantics

Example

(2) Anna ran to the station.



$$E \begin{bmatrix} \textit{running} \wedge \textit{bounded-motion} & & \\ \textsc{actor} & \boxed{1} \begin{bmatrix} \textit{person} & \\ \textsc{name} & \text{`Anna'} \end{bmatrix} \\ \textsc{final} & \begin{bmatrix} \textit{loc-stage} & \\ \textsc{theme} & \boxed{1} \\ \textsc{loc} & [\textit{station}] \end{bmatrix} \end{bmatrix}$$

Attribute-value logic

$$e \cdot (\textit{running} \wedge \textit{bounded-motion} \wedge \textsc{actor} : (\textit{person} \wedge \textsc{name} \triangleq \text{`Anna'})$$
$$\textsc{actor} \doteq \textsc{final theme} \wedge \textsc{final} : (\textit{loc-stage} \wedge \textsc{loc} : \textit{station}))$$

Constraints

$$\textit{running} \Rrightarrow \textit{activity} \quad (\text{short for } \forall e(\textit{running}(e) \rightarrow \textit{activity}(e))),$$
$$\textit{loc-stage} \Rrightarrow \textsc{theme} : \top \wedge \textsc{loc} : \top, \quad \ldots$$

Case study: directed motion construction

Intransitive:

(3) a. Mary walked to the house.
    b. The ball rolled into the goal.

Case study: directed motion construction

Intransitive:

(3) a. Mary walked to the house.
  b. The ball rolled into the goal.

Transitive:

(4) a. John threw/kicked the ball into the goal.
  b. John pushed/pulled the cart to the station.
  c. John rolled the ball into the hole.

Case study: directed motion construction

Intransitive:

(3) a. Mary walked to the house.
    b. The ball rolled into the goal.

Transitive:

(4) a. John threw/kicked the ball into the goal.
    b. John pushed/pulled the cart to the station.
    c. John rolled the ball into the hole.

Directional specifications are not restricted to **goal** expressions but can
also describe the **source** or the **course of the path** in more detail.

Case study: directed motion construction

Intransitive:

(3) a. Mary walked to the house.
b. The ball rolled into the goal.

Transitive:

(4) a. John threw/kicked the ball into the goal.
b. John pushed/pulled the cart to the station.
c. John rolled the ball into the hole.

Directional specifications are not restricted to **goal** expressions but can
also describe the **source** or the **course of the path** in more detail.
Moreover, path descriptions can be **iterated** to some extent:

(5) a. John walked through the gate along the fence to the house.
b. John threw the ball over the fence into the yard.

Case study: directed motion construction

Question: Syntactic treatment of directional PPs ?

- Construction ($\rightsquigarrow$ elementary tree)
- Syntactic composition ($\rightsquigarrow$ adjunction)

Case study: directed motion construction

Question: Syntactic treatment of directional PPs ?

- Construction ($\rightsquigarrow$ elementary tree)
- Syntactic composition ($\rightsquigarrow$ adjunction)

Arguments for treating goal (or **bounded**) PPs constructionally, in contrast to path (or **unbounded**) PPs:

- Goal PPs cannot be iterated.

Case study: directed motion construction

Question: Syntactic treatment of directional PPs ?

- Construction ($\rightsquigarrow$ elementary tree)
- Syntactic composition ($\rightsquigarrow$ adjunction)

Arguments for treating goal (or **bounded**) PPs constructionally, in contrast to path (or **unbounded**) PPs:

- Goal PPs cannot be iterated.
- They affect the Aktionsart of the expression:

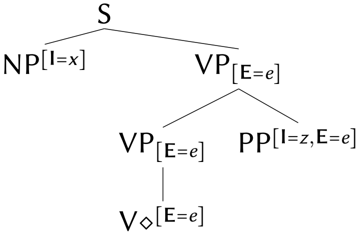
(6) a. She walked (*in half an hour/for half an hour).

b. She walked to the brook (in half an hour/*for half an hour).

c. She walked along the brook (*in half an hour/for half an hour).

Case study: directed motion construction

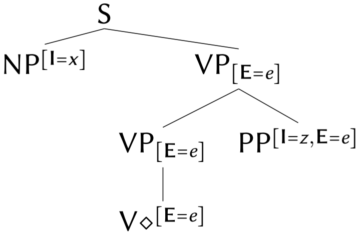
Unanchored construction for intransitive directed motion (*n0Vpp(dir)*):



$$e\begin{bmatrix} \textit{bounded-translocation} & \\ \textsc{mover} & x \\ \textsc{goal} & z \\ \textsc{path} & [\textit{path}] \end{bmatrix}$$

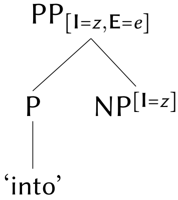
Case study: directed motion construction

Unanchored construction for intransitive directed motion (*n0Vpp(dir)*):



$$e\begin{bmatrix} \textit{bounded-translocation} & \\ \text{MOVER} & x \\ \text{GOAL} & z \\ \text{PATH} & [\textit{path}] \end{bmatrix}$$

Elementary tree for ‘into’:



$$e\begin{bmatrix} \textit{bounded-translocation} & \\ \text{PATH} & \begin{bmatrix} \textit{path} & \\ \text{ENDP} & \boxed{1} \end{bmatrix} \end{bmatrix}$$

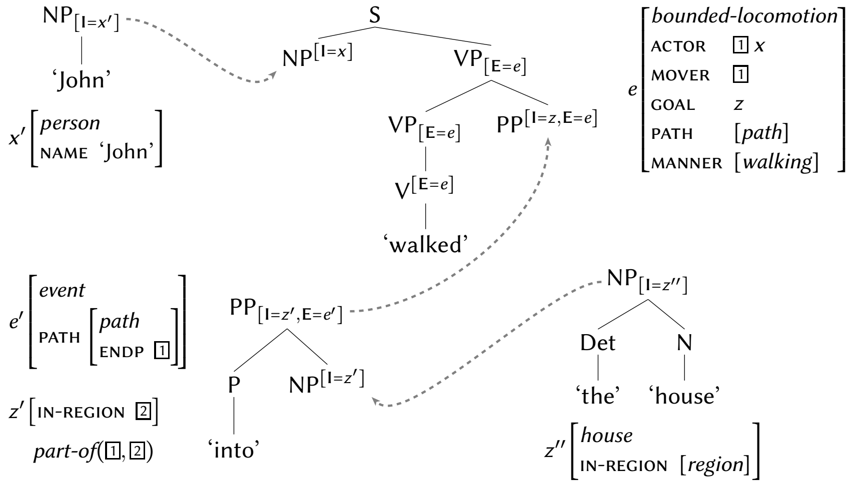
$$z\begin{bmatrix} \text{IN-REGION} & \boxed{2} \end{bmatrix}$$

$$\textit{part-of}(\boxed{1}, \boxed{2})$$

Case study: directed motion construction

Example (intransitive directed motion)

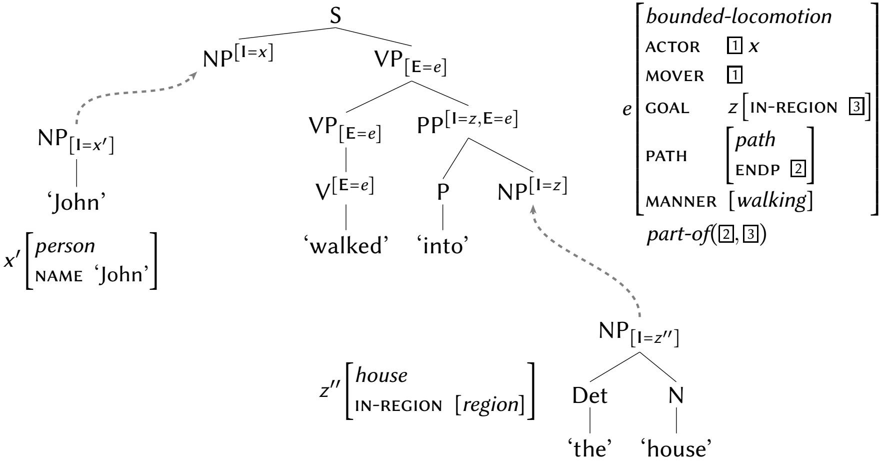
(7) John walked into the house.



Case study: directed motion construction

Example (intransitive directed motion)

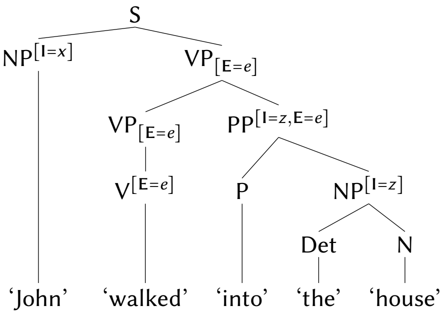
(7) John walked into the house.



Case study: directed motion construction

Example (intransitive directed motion)

(7) John walked into the house.



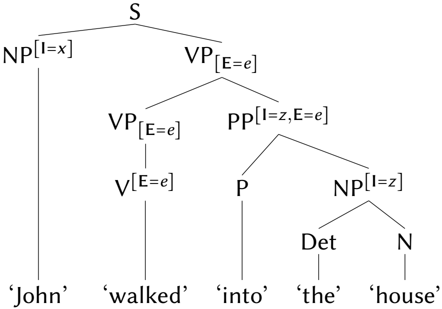
$$
e\begin{bmatrix}
\textit{bounded-locomotion} & \\
\text{ACTOR} & \boxed{1}\; x \begin{bmatrix} \textit{person} & \\ \text{NAME} & \text{`John'} \end{bmatrix} \\
\text{MOVER} & \boxed{1} \\
\text{GOAL} & z \begin{bmatrix} \textit{house} & \\ \text{IN-REGION} & \boxed{3}\; [\textit{region}] \end{bmatrix} \\
\text{PATH} & \begin{bmatrix} \textit{path} & \\ \text{ENDP} & \boxed{2} \end{bmatrix} \\
\text{MANNER} & [\textit{walking}]
\end{bmatrix}
$$

$\textit{part-of}(\boxed{2}, \boxed{3})$

Case study: directed motion construction

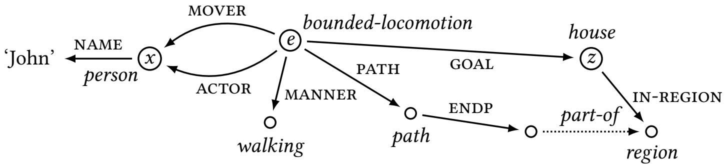
Example (intransitive directed motion)

(7) John walked into the house.



e [*bounded-locomotion*
ACTOR [1] x [*person* ; NAME 'John']
MOVER [1]
GOAL z [*house* ; IN-REGION [3] [*region*]]
PATH [*path* ; ENDP [2]]
MANNER [*walking*]]

part-of([2], [3])



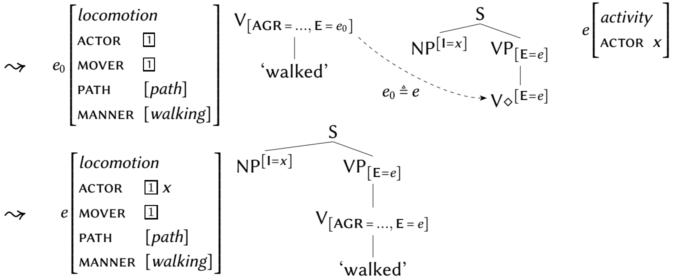
Case study: directed motion construction

Lexical anchoring (non-directed case)

morph entry +
'walked'
pos: V
Syn_1:
$\left[\text{AGR} = \begin{bmatrix}\text{PERS} = 3 \\ \text{NUM} = sg\end{bmatrix}\right]$
lemma: walk

lemma entry +
walk:
FAM: n0V, …
Syn_2:
$[\text{E} = e_0]$
Sem:
$e_0\begin{bmatrix}\textit{locomotion} & \\ \text{MANNER} & [\textit{walking}]\end{bmatrix}$

Constraints:
$\textit{locomotion} \Rrightarrow \textit{activity} \wedge \textit{translocation}$
$\textit{translocation} \Rrightarrow \textit{motion} \wedge \text{PATH} : \textit{path}$
$\textit{activity} \Rrightarrow \text{ACTOR} : \top$
$\textit{motion} \Rrightarrow \text{MOVER} : \top$
$\textit{activity} \wedge \textit{motion} \Rrightarrow \text{ACTOR} \doteq \text{MOVER}$



Case study: directed motion construction

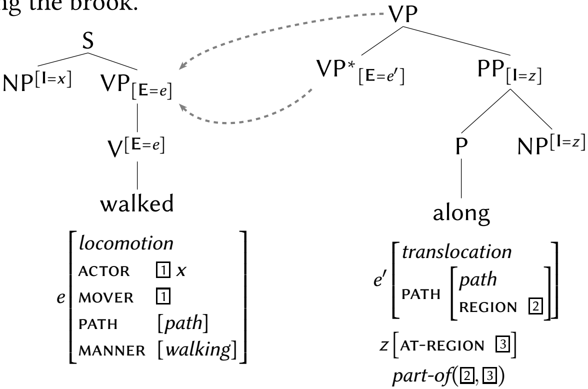
Example

(8) John walked along the brook.

Case study: directed motion construction

Example

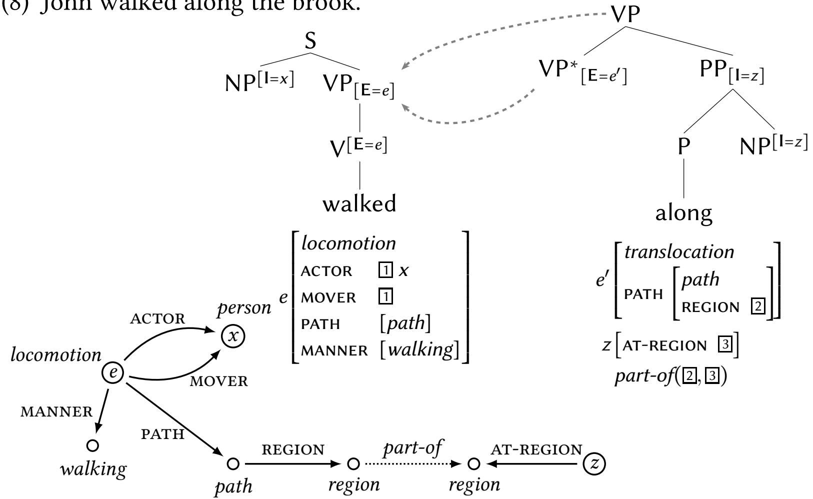
(8) John walked along the brook.



Case study: directed motion construction

Example

(8) John walked along the brook.



Case study: directed motion construction

Example (causative directed motion)

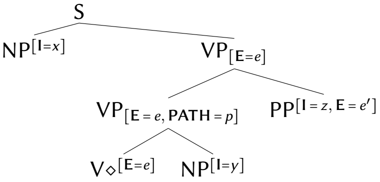
(9) Mary threw/kicked/rolled the ball into the room.

Case study: directed motion construction

Example (causative directed motion)

(9) Mary threw/kicked/rolled the ball into the room.

Unanchored construction (*n0Vn1pp(dir)*):



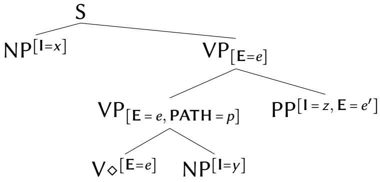
$$e\begin{bmatrix} \textit{causation} & & \\ \text{CAUSE} & \begin{bmatrix} \textit{activity} & \\ \text{ACTOR} & x \\ \text{THEME} & y \end{bmatrix} \\ \text{EFFECT} & e'\begin{bmatrix} \textit{bounded-translocation} & \\ \text{MOVER} & y \\ \text{GOAL} & z \\ \text{PATH} & p \end{bmatrix} \end{bmatrix}$$

Case study: directed motion construction

Example (causative directed motion)

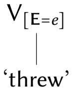
(9) Mary threw/kicked/rolled the ball into the room.

Unanchored construction (*n0Vn1pp(dir)*):



$$e\begin{bmatrix} \textit{causation} & \\ \text{CAUSE} & \begin{bmatrix}\textit{activity} & \\ \text{ACTOR} & x \\ \text{THEME} & y\end{bmatrix} \\ \text{EFFECT} & e'\begin{bmatrix}\textit{bounded-translocation} & \\ \text{MOVER} & y \\ \text{GOAL} & z \\ \text{PATH} & p\end{bmatrix}\end{bmatrix}$$

(Partial) lexical entry for 'threw':



$$e\begin{bmatrix} \textit{onset-causation} & \\ \text{CAUSE} & \begin{bmatrix}\textit{activity} & \\ \text{ACTOR} & \top \\ \text{THEME} & \boxed{1} \\ \text{MANNER} & [\textit{throwing}]\end{bmatrix} \\ \text{EFFECT} & \begin{bmatrix}\textit{translocation} \wedge \textit{undergoing} & \\ \text{THEME} & \boxed{1} \\ \text{MOVER} & \boxed{1}\end{bmatrix}\end{bmatrix}$$

Barsalou, L. W. (1992). Frames, concepts, and conceptual fields. In Lehrer, A. and Kittay, E. F., editors, Frames, Fields, and Contrasts, New Essays in Semantic and Lexical Organization, chapter 1, pages 21–74. Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Gardent, C. and Kallmeyer, L. (2003). Semantic Construction in FTAG. In Proceedings of EACL 2003, pages 123–130, Budapest.

Kallmeyer, L. and Joshi, A. K. (2003). Factoring Predicate Argument and Scope Semantics: Underspecified Semantics with LTAG. Research on Language and Computation, 1(1–2):3–58.

Kallmeyer, L. and Osswald, R. (2013). Syntax-driven semantic frame composition in Lexicalized Tree Adjoining Grammar. Journal of Language Modelling, 1:267–330.

Kallmeyer, L., Osswald, R., and Pogodalla, S. (2016). **For**-adverbials and aspectual interpretation: An LTAG analysis using Hybrid Logic and Frame Semantics. In Empirical Issues in Syntax and Semantics 11, pages 61–90.

Kallmeyer, L. and Romero, M. (2008). Scope and situation binding in LTAG using semantic unification. Research on Language and Computation, 6(1):3–52.

Nesson, R. and Shieber, S. M. (2006). Simpler TAG semantics through synchronization. In Proceedings of the 11th Conference on Formal Grammar, Malaga, Spain.

Nesson, R. and Shieber, S. M. (2008). Synchronous vector tag for syntax and semantics: Control verbs, relative clauses, and inverse linking. In Proceedings of the Ninth International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+ 9), Tübingen, Germany.

Osswald, R. and Van Valin, Jr., R. D. (2014). FrameNet, frame structure, and the syntax-semantics interface. In Gamerschlag, T., Gerland, D., Osswald, R., and Petersen, W., editors, Frames and Concept Types, number 94 in Studies in Linguistics and Philosophy, pages 125–156. Springer, Dordrecht.

Shieber, S. M. (1994). Restricting the weak-generative capacity of synchronous Tree-Adjoining Grammars. Computational Intelligence, 10(4):271–385.

Shieber, S. M. and Schabes, Y. (1990). Synchronous Tree-Adjoining Grammars. In Proceedings of COLING, pages 253–258.