# Tree Adjoining Grammars

## Feature Structure Based TAG

Laura Kallmeyer & Benjamin Burkhardt

HHU Düsseldorf

WS 2017/2018

# Outline

1. Why feature structures?
2. Basics of feature structure logic
3. Feature Structure based TAG (FTAG)

# Why feature structures?

**Idea:** Instead of atomic categorial symbols, feature structures are used as non-terminal nodes.

Two reasons with respect to TAG:

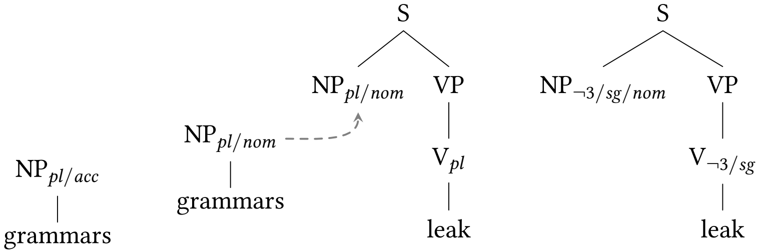
- generalizing agreement, case assignment etc. (via underspecification)
- modelling adjunction constraints

⇒ meaningful generalizations

⇒ smaller grammars that are easier to maintain

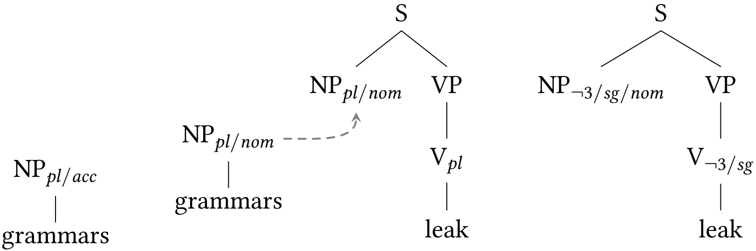
# Why feature structures? Agreement

Example without feature structures:



# Why feature structures? Agreement

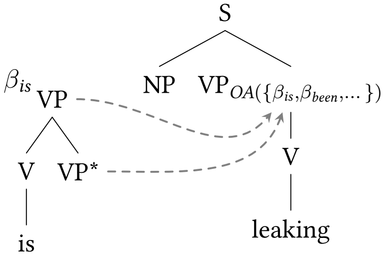
Example without feature structures:




- The generalization that the finite verb and its subject agree in number and person is not captured.
- Every morphological alternative gives rise to a new elementary tree!

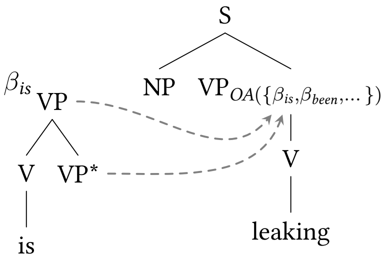
# Why feature structures? Adjunction constraints

Example without feature structures:



# Why feature structures? Adjunction constraints

Example without feature structures:




- The generaliztion that some form of the auxiliary *be* has to be adjoined is not captured.
- Things get even worse when combining agreement and adjunction constraints. (A plural subject forbids for instance adjunction of $\beta_{is}$, ...)

# Features structures

Features structures are

- sets of features (e.g. CASE) and unique values (e.g. nom)
- feature structures are often represented as **attribute value matrices (AVM)**

$$\begin{bmatrix} \textsc{cat} & \text{V} \\ \textsc{vform} & \text{finite} \\ \textsc{agr} & \begin{bmatrix} \textsc{num} & \text{sg} \\ \textsc{pers} & 3 \end{bmatrix} \end{bmatrix}$$

- feature values can be
  - atomic (e.g. for VFORM)
  - feature structures (e.g. for AGR)
- A feature structure is called **recursive** if there is an attribute *attr* that occurs inside the value of a higher attribute *attr*.

TAG uses non-recursive feature structures.

# Unification

- Feature structures are combined by **unification**.
- Unification is a (partial) operation on feature structures.
  Intuitively: the operation of combining two feature structures such that the new feature structure contains all the information of the original two, and nothing more
  e.g. $\begin{bmatrix} \textsc{cat} & \text{vp} \\ \textsc{agr} & \begin{bmatrix} \textsc{num} & \text{pl} \end{bmatrix} \end{bmatrix} \sqcup \begin{bmatrix} \textsc{cat} & \text{vp} \\ \textsc{agr} & \begin{bmatrix} \textsc{pers} & 3 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \textsc{cat} & \text{vp} \\ \textsc{agr} & \begin{bmatrix} \textsc{num} & \text{pl} \\ \textsc{pers} & 3 \end{bmatrix} \end{bmatrix}$
- Unification can fail (partial operation).
  e.g. $\begin{bmatrix} \textsc{cat} & \text{np} \\ \textsc{num} & \text{sg} \end{bmatrix} \sqcup \begin{bmatrix} \textsc{cat} & \text{np} \\ \textsc{num} & \text{pl} \end{bmatrix} = \text{FAIL}$

# Unification

**Subsumption ($F_1 \sqsubseteq F_2$)**

A feature structure $F_1$ subsumes ($\sqsubseteq$) another feature structure $F_2$, iff all the information that is contained in $F_1$ is also contained in $F_2$.

# Unification

## Subsumption ($F_1 \sqsubseteq F_2$)

A feature structure $F_1$ subsumes ($\sqsubseteq$) another feature structure $F_2$, iff all the information that is contained in $F_1$ is also contained in $F_2$.

## Example: Subsumption

$$\begin{bmatrix} \textsc{cat} & \text{np} \\ \textsc{num} & \text{sg} \end{bmatrix} \sqsubseteq \begin{bmatrix} \textsc{cat} & \text{np} \\ \textsc{case} & \text{acc} \\ \textsc{num} & \text{sg} \end{bmatrix}$$

$$\begin{bmatrix} \textsc{cat} & \text{np} \\ \textsc{agr} & \begin{bmatrix} \textsc{num} & \text{sg} \end{bmatrix} \end{bmatrix} \sqsubseteq \begin{bmatrix} \textsc{cat} & \text{np} \\ \textsc{agr} & \begin{bmatrix} \textsc{num} & \text{sg} \\ \textsc{pers} & 3 \end{bmatrix} \end{bmatrix}$$

# Unification

**Unification ($F \sqcup G$)**

The unification of two feature structures $F$ and $G$ is (if it exists) the smallest feature structure that is subsumed by both $F$ and $G$: $F \sqcup G$ is the feature structure with the following three properties:

# Unification

## Unification ($F \sqcup G$)

The unification of two feature structures $F$ and $G$ is (if it exists) the smallest feature structure that is subsumed by both $F$ and $G$: $F \sqcup G$ is the feature structure with the following three properties:

(1) $F \sqsubseteq (F \sqcup G)$

(2) $G \sqsubseteq (F \sqcup G)$

(3) If $H$ is a feature structure such that $F \sqsubseteq H$ and $G \sqsubseteq H$, then $(F \sqcup G) \sqsubseteq H$.

# Unification

## Unification ($F \sqcup G$)

The unification of two feature structures $F$ and $G$ is (if it exists) the smallest feature structure that is subsumed by both $F$ and $G$: $F \sqcup G$ is the feature structure with the following three properties:

(1) $F \sqsubseteq (F \sqcup G)$

(2) $G \sqsubseteq (F \sqcup G)$

(3) If $H$ is a feature structure such that $F \sqsubseteq H$ and $G \sqsubseteq H$, then $(F \sqcup G) \sqsubseteq H$.

## Example: Unification

$$\begin{bmatrix} \textsc{cat} & \text{np} \\ \textsc{agr} & \begin{bmatrix} \textsc{num} & \text{sg} \end{bmatrix} \end{bmatrix} \sqcup \begin{bmatrix} \textsc{case} & \text{acc} \\ \textsc{agr} & \begin{bmatrix} \textsc{pers} & 3 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \textsc{cat} & \text{np} \\ \textsc{case} & \text{acc} \\ \textsc{agr} & \begin{bmatrix} \textsc{num} & \text{sg} \\ \textsc{pers} & 3 \end{bmatrix} \end{bmatrix}$$

$$\begin{bmatrix} \textsc{cat} & \text{np} \\ \textsc{agr} & \begin{bmatrix} \textsc{num} & \text{sg} \end{bmatrix} \end{bmatrix} \sqcup \begin{bmatrix} \textsc{cat} & \text{vp} \\ \textsc{agr} & \begin{bmatrix} \textsc{pers} & 3 \end{bmatrix} \end{bmatrix} = \bot$$

# Reentrancies

- Several paths can lead to the same node ⇒ to the same value.
  ⇒ hence, they share that value.
- This property of sharing value(s) is called **reentrancy**
- In AVMs: expressed by coindexing the shared values (boxed numbers).

$$\begin{bmatrix} \text{ATTR}_1 & \boxed{1} \\ \text{ATTR}_2 & \boxed{1} \end{bmatrix} \qquad \begin{bmatrix} \text{ATTR}_1 & \boxed{1}\,\text{val}_1 \\ \text{ATTR}_2 & \boxed{1} \end{bmatrix} \qquad \begin{bmatrix} \text{ATTR}_1 & \boxed{1} \begin{bmatrix} \text{ATTR}_2 & \boxed{1} \end{bmatrix} \end{bmatrix}$$

FTAG uses acyclic reentrancies!

Reentrancies can occur between features structures (in a tree):

$$\begin{bmatrix} \text{ATTR}_1 & \boxed{1} \end{bmatrix} \qquad \begin{bmatrix} \text{ATTR}_1 & \boxed{1} \end{bmatrix}$$

# Reentrancies

Note that

- Feature structucs in FTAG are untyped.
- The feature geometry is such that there is only a finite number of possible feature structures.
- Therefore, FTAG can be shown to be strongly equivalent to TAG without feature structures.

# Unification: examples

- $\begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{NUM} & \text{sg} \end{bmatrix} \\ \text{SUBJ} & \begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{NUM} & \text{sg} \end{bmatrix} \end{bmatrix} \end{bmatrix} \sqcup \begin{bmatrix} \text{SUBJ} & \begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{PERS} & 3 \end{bmatrix} \end{bmatrix} \end{bmatrix} =$
  $\begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{NUM} & \text{sg} \end{bmatrix} \\ \text{SUBJ} & \begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{NUM} & \text{sg} \\ \text{PERS} & 3 \end{bmatrix} \end{bmatrix} \end{bmatrix}$

# Unification: examples

- $$\begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{NUM} & \text{sg} \end{bmatrix} \\ \text{SUBJ} & \begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{NUM} & \text{sg} \end{bmatrix} \end{bmatrix} \end{bmatrix} \sqcup \begin{bmatrix} \text{SUBJ} & \begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{PERS} & 3 \end{bmatrix} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{NUM} & \text{sg} \end{bmatrix} \\ \text{SUBJ} & \begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{NUM} & \text{sg} \\ \text{PERS} & 3 \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

- $$\begin{bmatrix} \text{AGR} & \boxed{1} \begin{bmatrix} \text{NUM} & \text{sg} \end{bmatrix} \\ \text{SUBJ} & \begin{bmatrix} \text{AGR} & \boxed{1} \end{bmatrix} \end{bmatrix} \sqcup \begin{bmatrix} \text{SUBJ} & \begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{PERS} & 3 \end{bmatrix} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \text{AGR} & \boxed{1} \begin{bmatrix} \text{NUM} & \text{sg} \\ \text{PERS} & 3 \end{bmatrix} \\ \text{SUBJ} & \begin{bmatrix} \text{AGR} & \boxed{1} \end{bmatrix} \end{bmatrix}$$

# Unification: examples

- $$\begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{NUM} & \text{sg} \end{bmatrix} \\ \text{SUBJ} & \begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{NUM} & \text{sg} \end{bmatrix} \end{bmatrix} \end{bmatrix} \sqcup \begin{bmatrix} \text{SUBJ} & \begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{PERS} & 3 \end{bmatrix} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{NUM} & \text{sg} \end{bmatrix} \\ \text{SUBJ} & \begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{NUM} & \text{sg} \\ \text{PERS} & 3 \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

- $$\begin{bmatrix} \text{AGR} & \boxed{1} \begin{bmatrix} \text{NUM} & \text{sg} \end{bmatrix} \\ \text{SUBJ} & \begin{bmatrix} \text{AGR} & \boxed{1} \end{bmatrix} \end{bmatrix} \sqcup \begin{bmatrix} \text{SUBJ} & \begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{PERS} & 3 \end{bmatrix} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \text{AGR} & \boxed{1} \begin{bmatrix} \text{NUM} & \text{sg} \\ \text{PERS} & 3 \end{bmatrix} \\ \text{SUBJ} & \begin{bmatrix} \text{AGR} & \boxed{1} \end{bmatrix} \end{bmatrix}$$

- for any feature structure $F$: $F \sqcup [\,] = [\,] \sqcup F = F$

# Unification: examples

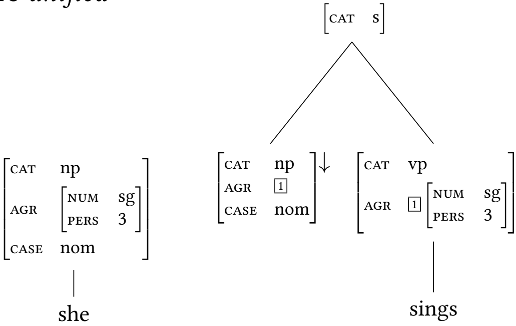
- $\begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{NUM} & \text{sg} \end{bmatrix} \\ \text{SUBJ} & \begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{NUM} & \text{sg} \end{bmatrix} \end{bmatrix} \end{bmatrix} \sqcup \begin{bmatrix} \text{SUBJ} & \begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{PERS} & 3 \end{bmatrix} \end{bmatrix} \end{bmatrix} =$
$\begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{NUM} & \text{sg} \end{bmatrix} \\ \text{SUBJ} & \begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{NUM} & \text{sg} \\ \text{PERS} & 3 \end{bmatrix} \end{bmatrix} \end{bmatrix}$

- $\begin{bmatrix} \text{AGR} & \boxed{1} \begin{bmatrix} \text{NUM} & \text{sg} \end{bmatrix} \\ \text{SUBJ} & \begin{bmatrix} \text{AGR} & \boxed{1} \end{bmatrix} \end{bmatrix} \sqcup \begin{bmatrix} \text{SUBJ} & \begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{PERS} & 3 \end{bmatrix} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \text{AGR} & \boxed{1} \begin{bmatrix} \text{NUM} & \text{sg} \\ \text{PERS} & 3 \end{bmatrix} \\ \text{SUBJ} & \begin{bmatrix} \text{AGR} & \boxed{1} \end{bmatrix} \end{bmatrix}$

- for any feature structure $F$: $F \sqcup [\,] = [\,] \sqcup F = F$
- the empty feature structure is the **identity element** for unification

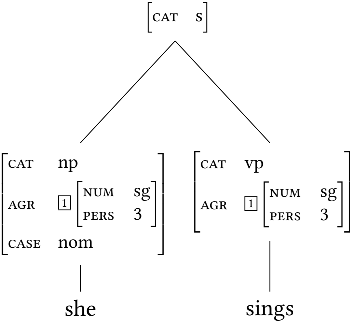
# TAG with feature structures

**Idea:** feature structures as non-terminal nodes.
At substitution/adjunction the feature structures of the participating nodes are *unified*



# TAG with feature structures

**Idea:** feature structures as non-terminal nodes.
At substitution/adjunction the feature structures of the participating nodes are *unified*



# FTAG

Feature-structure based TAG (FTAG Vijay-Shanker & Joshi, 1988):

- annotate each substitution node with one and each other node with two feature structures
- adjunction splits the feature structures
  - top features: the relation of the node to the tree above it
  - bottom features: the relation of the node to the tree below it

**FTAG description of node $\eta$**

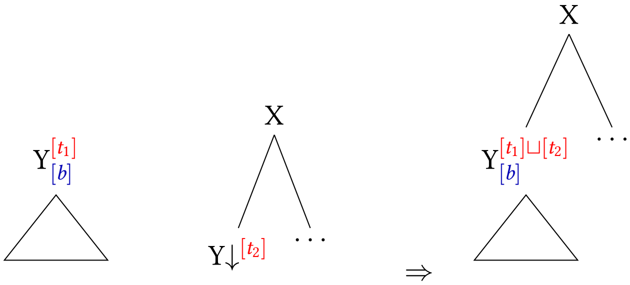
1. The relation of $\eta$ to its supertree is called feature structure $t_\eta$.
2. The relation of $\eta$ to its descendants is called feature structure $b_\eta$.

In the final derived tree top and bottom features are unified for all nodes

# FTAG: Substitution

**Substitution in FTAG**

The top features of the root of the tree to substitute unify with the top features of the substitution node.

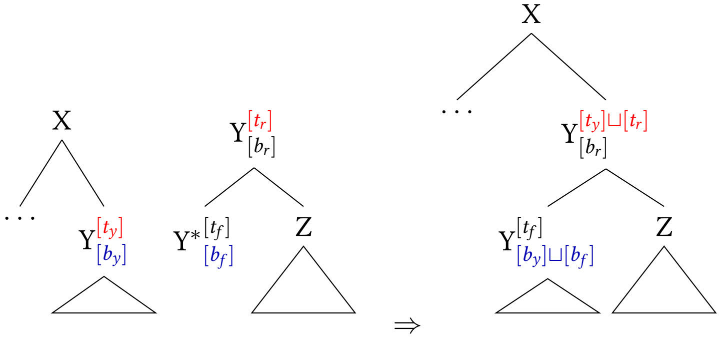


- substitution nodes (Y↓) have only top features

# FTAG: Adjunction

## Adjunction in FTAG

The top features of the root of the auxiliary tree unify with the top features of the adjunction node, and the bottom features of the footnode of the auxiliary tree unify with the bottom features of the adjunction node.



# Adjunction constraints

Modeling adjunction constraints with features:

# Adjunction constraints

Modeling adjunction constraints with features:

- SA: top and bottom are unifiable

$$\begin{bmatrix}\textsc{cat} & \text{vp}\end{bmatrix}$$
$$\begin{bmatrix}\textsc{cat} & \text{vp}\end{bmatrix}$$

# Adjunction constraints

Modeling adjunction constraints with features:

- SA: top and bottom are unifiable

$$\begin{bmatrix}\textsc{cat} & \text{vp}\end{bmatrix}$$
$$\begin{bmatrix}\textsc{cat} & \text{vp}\end{bmatrix}$$

- OA + SA: feature mismatch between top and bottom

$$\begin{bmatrix}\textsc{cat} & \text{vp}\\ \textsc{mode} & \text{ind}\end{bmatrix}$$
$$\begin{bmatrix}\textsc{cat} & \text{vp}\\ \textsc{mode} & \text{ger}\end{bmatrix}$$

# Adjunction constraints

Modeling adjunction constraints with features:

- SA: top and bottom are unifiable

$$\begin{bmatrix}\textsc{cat} & \text{vp}\end{bmatrix}$$
$$\begin{bmatrix}\textsc{cat} & \text{vp}\end{bmatrix}$$

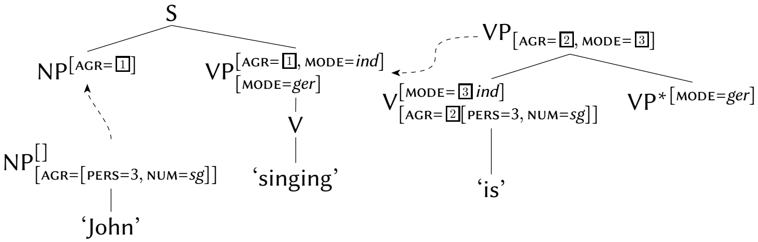
- OA + SA: feature mismatch between top and bottom

$$\begin{bmatrix}\textsc{cat} & \text{vp} \\ \textsc{mode} & \text{ind}\end{bmatrix}$$
$$\begin{bmatrix}\textsc{cat} & \text{vp} \\ \textsc{mode} & \text{ger}\end{bmatrix}$$

- NA: top and bottom are unifiable, but there is no auxiliary tree in the grammar that can be unified with them

# FTAG example for OA

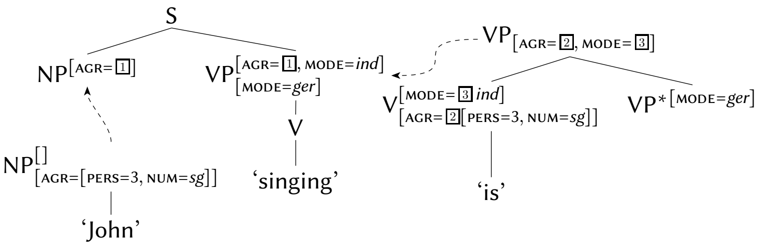
(1) John is singing.



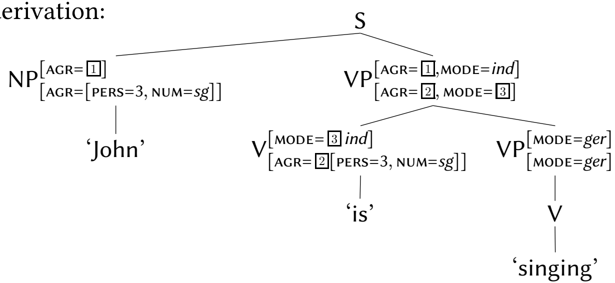

- The features are inspired by the XTAG grammar (XTAG Research Group, 2001).
- The CAT feature is taken to be special, in particular it is usually the same in top and bottom. We therefore notate it as the main category of a node, outside the feature structures.

# FTAG example for OA

(1) John is singing.

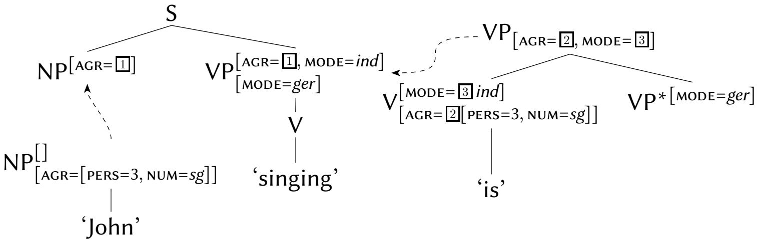



Result of derivation:

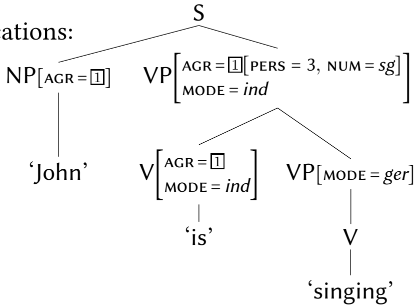


# FTAG example for OA

(1) John is singing.




After top-bottom unifications:



# Summary

- Feature structures as nodes allow to abstract away from agreement properties by underspecification. Linguistic generalizations an be expressed more conveniently.
- Adjunction constraints can be enoded into feature structures.
- The feature structures of FTAG do not add expressive power, hence FTAG and TAG are weakly equivalent.

# References

Vijay-Shanker, K. & Aravind K. Joshi. 1988. Feature structures based tree adjoining grammar. In Proceedings of coling, 714–719. Budapest.

XTAG Research Group. 2001. A Lexicalized Tree Adjoining Grammar for English. Tech. rep. Institute for Research in Cognitive Science Philadelphia. Available from `ftp://ftp.cis.upenn.edu/pub/xtag/release-2.24.2001/tech-report.pdf`.