Lexicalized Tree Adjoining Grammar

Petr Horáček, Eva Zámečníková and Ivana Burgetová

Department of Information Systems
Faculty of Information Technology
Brno University of Technology
Božetěchova 2, 612 00 Brno, CZ

BRNO UNIVERSITY OF TECHNOLOGY

FACULTY OF INFORMATION TECHNOLOGY

FRVŠ MŠMT FR97/2011/G1

Outline

- **Introduction**
- **Tree Adjoining Grammar**
- **Some Important TAG Properties**
- **Lexicalized Tree Adjoining Grammar**

- **Introduction**
- Tree Adjoining Grammar
- Some Important TAG Properties
- Lexicalized Tree Adjoining Grammar

Motivation

Motivation is of linguistic and formal nature.

- Elementary objects are trees - structured objects and not strings.
- Structured objects are related with strong generative capacity. $\Rightarrow$ More relevant to linguistic description.
- TAG allow *factoring recursion* from the statement of linguistic dependencies
- Lexicalization of grammar formalism.

- TAG is tree-generating system $\Rightarrow$ the set of trees constitute the *object language*
- One well known normal form of grammars - Greibach Normal Form (GNF) is a kind of lexicalization.

Topic

- Introduction
- **Tree Adjoining Grammar**
- Some Important TAG Properties
- Lexicalized Tree Adjoining Grammar

Definition

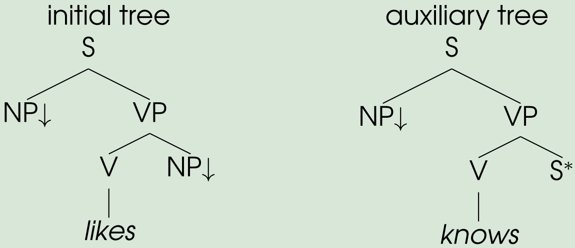
Tree Adjoining Grammar (TAG) is a quintuple (T, N, I, A, S).

- $T \ldots$ a finite set of terminal symbols
- $N \ldots$ a finite set of nonterminal symbols; $T \cap N = \emptyset$
- $I \ldots$ a finite set of initial trees
  - An initial tree is a phrase structure tree
- $A \ldots$ a finite set of auxiliary trees
  - An auxiliary tree is a phrase structure tree that has a leaf nonterminal node that is the same as its root symbol
- $S \ldots$ start symbol, $S \in N$

- Trees in I and A are called elementary trees.
- Parsing is done by two operations: substitution and adjunction.

An example of an initial and an auxiliary tree

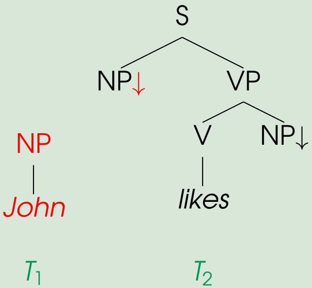
Example



- A nonterminal symbol marked by * is the foot node of an auxiliary tree.
- A nonterminal symbol marked by ↓ is a nonterminal node for substitution.

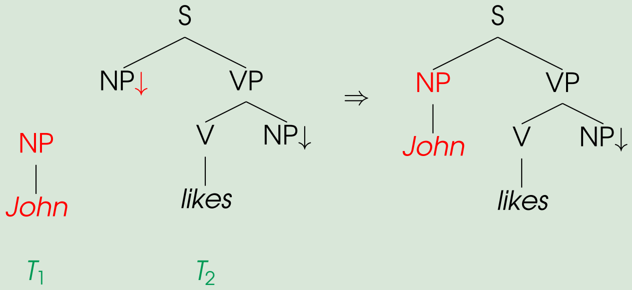
Substitution of an initial tree T_1 into a tree T_2 is to replace a substitution node in T_2 with T_1.

Example



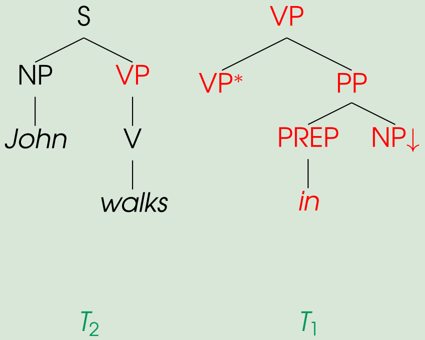
Substitution of an initial tree T_1 into a tree T_2 is to replace a substitution node in T_2 with T_1.

Example



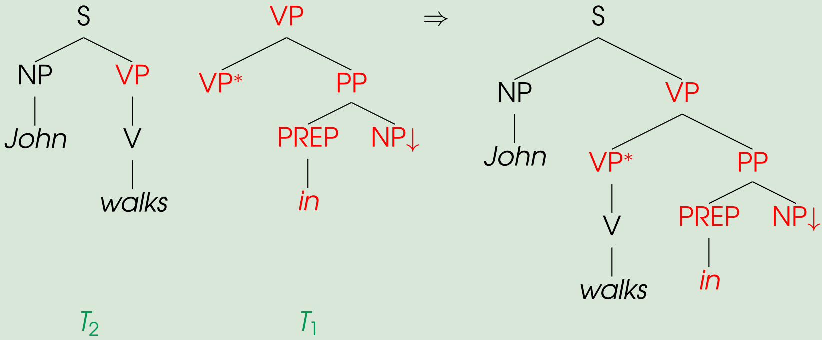
Adjoining an auxiliary tree T_1 into a tree T_2 is to inset T_1 into T_2 at the node that is the same as the root (and the foot) of T_1.

Example



Adjoining an auxiliary tree T_1 into a tree T_2 is to inset T_1 into T_2 at the node that is the same as the root (and the foot) of T_1.

Example



T_1 is adjoined at VP in T_2

Adjunction (Adjoining)

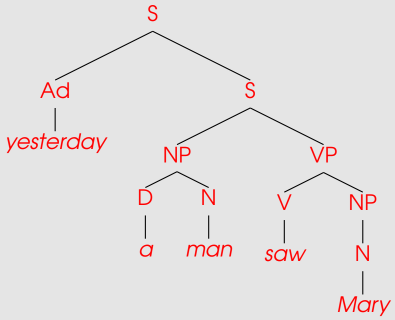
- Any adjunction on a node marked for substitution is *disallowed*.

Adjoining Constraints

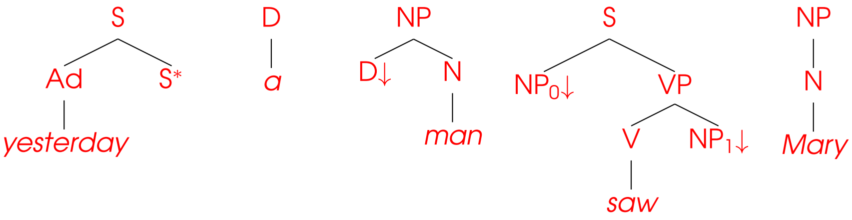
to have more precision for specifying which auxiliary trees can be adjoined at a given node.

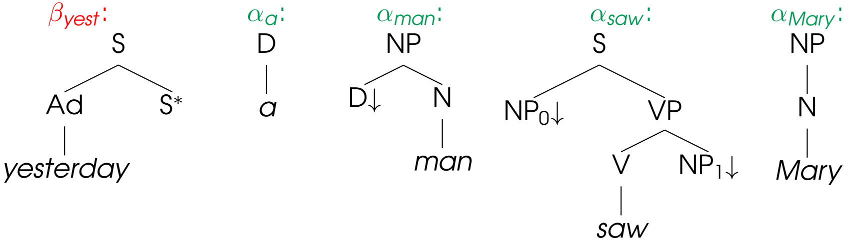
1. *Selective Adjunction* (S A(T)) - only members of a set $T \subseteq A$ of auxiliary trees can be adjoined on the given node, the adjunction of an auxiliary is not mandatory on the given node.
2. *Null Adjunction* (N A) - disallows any adjunction on the given node.
3. *Obligatory Adjunction* (O A(T)) - an auxiliary tree member of the set $T \subseteq A$ must be adjoined on the given node.

- These constraints on adjoining are needed for formal reasons in order to obtain some closure properties.

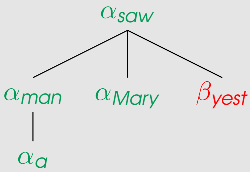


This tree yields the sentence *Yesterday a man saw Mary* and is derived from the following *elementary* trees:





Derivation tree for *Yesterday a man saw Mary.*



The order in which the derivation tree is interpreted has no impact on the resulting derived tree.

Derivation in TAG

Derived Tree

A tree built by composition of two others trees.

- the derived tree does not give enough information to determine how it was constructed
- adjunction and substitution are considered in a TAG derivation

Derivation Tree

It is an object that specifies uniquely how a derived tree was constructed.

Topic

- Introduction
- Tree Adjoining Grammar
- **Some Important TAG Properties**
- Lexicalized Tree Adjoining Grammar

Tree Sets and String Languages

Tree Set of a TAG T_G

- Defined as the set of completed initial trees derived from some *S-rooted initial trees*.

$$T_G = \{t | t \text{ is derived from some S-rooted initial tree}\}$$

- Note that completed initial tree is an initial tree with no substitution nodes.

Tree String language of a TAG L_G

- Defined as the set of yields of all trees in the tree set.

$$L_G = \{w | w \text{ is the yield of some } t \text{ in } T_G\}$$

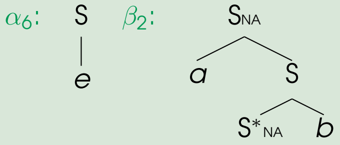
Some Properties of the Tree Sets and String Languages

- All closure properties of context-free languages (CFL) also hold for tree-adjoining languages (TAL).
- $CFL \subset TAL$
- TAL can be parsed in polynomial time.
- Tree-adjoining grammars generate some context-sensitive languages.

Some Properties of the Tree Sets and String Languages: Example 1

Example

Consider following TAG $G_1 = (\{a, e, b\}, \{S\}, \{\alpha_6\}, \{\beta_2\}, S)$

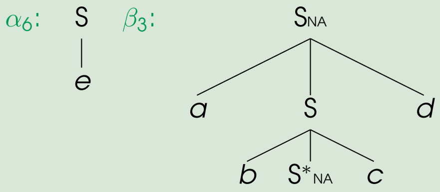


- G_1 generates the language $L_1 = \{a^n e b^n | n \geq 1\}$

Some Properties of the Tree Sets and String Languages: Example 2

Example

Consider following TAG $G_1 = (\{a, b, c, d, e\}, \{S\}, \{\alpha_6\}, \{\beta_3\}, S)$

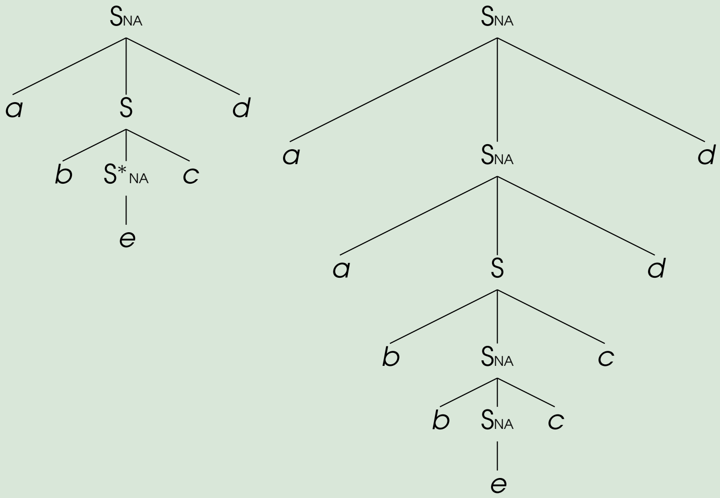


- G_1 generates the language $L_1 = \{a^n b^n e c^n d^n | n \geq 1\}$

Some Properties of the Tree Sets and String Languages: Example 2

Example

Some derived trees of G_2



Topic

- Introduction
- Tree Adjoining Grammar
- Some Important TAG Properties
- **Lexicalized Tree Adjoining Grammar**

Lexicalized Grammar

- Each elementary structure is associate with a lexical item.
- The grammar consists of *lexicon*, where:
  - each lexical item is associated with a finite number of structures and
  - there are operations which tell how these structures are composed.

Definition

- A grammar is lexicalized if it consists of a finite set of structures each associated with a lexical item.
- Each lexical item is called the anchor of the corresponding structure.
- Grammar contains an operation or operations for composing the structure.

- LTAG is a TAG in which every elementary (initial and auxiliary) tree is anchored with a lexical item.

Notes

- The *anchor* must be overt (= not empty string).
- The structures defined by the lexicon are called *elementary structures*.
- Structures built up by combination of others are called *derived structures*.

The definition of Lexicalized Grammar implies the following proposition:

Proposition

Lexicalized grammars are finitely ambiguous.

Further, this closure property holds:

Closure under lexicalization

TAGs are closed under lexicalization.

Conclusion

Notes

- Lexicalization of grammars is of linguistic and formal interest.
- Rules should not be separated from their lexical realization.
- By using TAGs we can lexicalize the CFGs.
- Substitution and adjunction gives this possibility to lexicalize CFG.

- James Allen:
  Natural Language Understanding,
  The Benjamin/Cummings Publishing Company. Inc., 2005
- Yuji Matsumoto:
  Syntax and Parsing: Phrase Structure and Dependency Parsing Algorithms,
  SSLST, 2011
- Yves Schabes, Aravind K. Joshi:
  Tree-Adjoining Grammars and Lexicalized Grammars, 1991