Tree-Adjoining Grammars

Miro Lehtonen

Department of Computer Science

University of Helsinki

Outline

- **Introduction**: formalisms for linguistic purposes.
- **Basics of TAGs:** elementary structures and operations, derivation.
- **Formal properties of grammars and TAGs**
- **TAG variants**
  - **Multicomponent TAGs (MC-TAG)**
  - **Synchronous TAGs (S-TAG)**
- **TAG parsing**

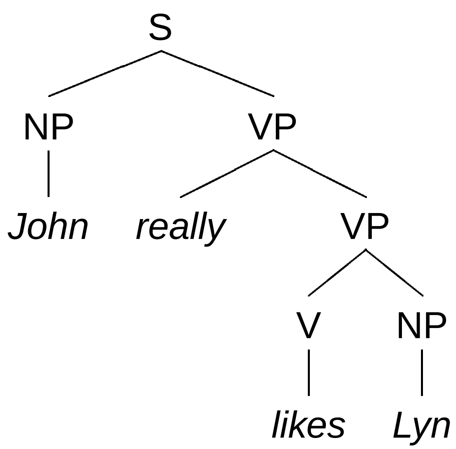
Formal systems for linguistic theories

- Basis of any formal system: elementary structures and combining operations.
- Context-free grammars (CFG): terminal and nonterminal symbols, and rewrite rules.
- CFG example – rules as elementary structures.

1. S $\longrightarrow$ NP VP
2. VP $\longrightarrow$ *really* VP
3. VP $\longrightarrow$ V NP
4. V $\longrightarrow$ *likes*
5. NP $\longrightarrow$ *John*
6. NP $\longrightarrow$ *Lyn*

Derivation in CFGs

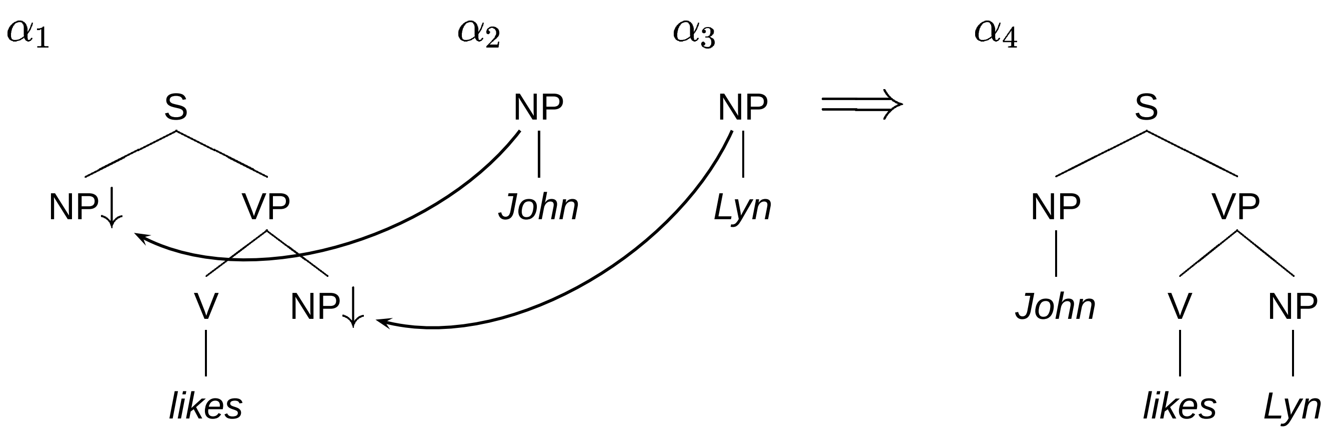
- The phrase structure tree



- For each nonterminal node, the daughters record which rule was used to rewrite it.

Tree Substitution Grammars (TSG)

- Both elementary objects and derivations are trees.
- TSG example.



- Elementary structures are combined by **substitution**.
- Condition: The nonterminal node must have the same label as the root node of the substituted tree.

Domain of locality

- CFGs and TSGs are **weakly equivalent**.
  - → They generate the same string languages, but
  - → the derived structures have a different **Domain of locality**.
- Local restrictions are valid in the domain of locality:
  - → a CFG rule or a tree grammar tree.
- Examples: NP – V agreement, subcategorisation.
- TSGs (and other tree grammars) have an **Extended domain of locality**.

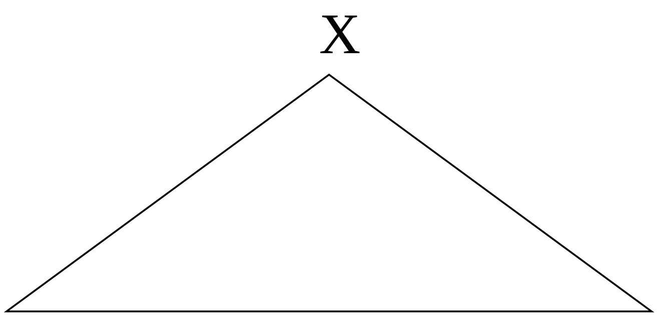
Lexicalisation

- A grammar is **lexicalised**, if
  - every elementary structure is associated with exactly one lexical item, and
  - every lexical item of the language is associated with a finite set of elementary structures in the grammar.
- CFGs cannot be lexicalised in a linguistically meaningful manner, but let's try.
  - S $\longrightarrow$ NP *likes* NP
  - No place for *really* ?
  - Instead of merging two rules into one, we can combine them into a tree structure $\Rightarrow$ TSG.
  - Still no place for *really*.
- Solution: **Adjunction** operation.
- A formalism in which the elementary structures of a grammar are trees and in which the combining operations are adjunction and substitution is called a **Tree Adjoining Grammar (TAG)**.
- When lexicalised, we have a Lexicalised Tree Adjoining Grammar (LTAG).

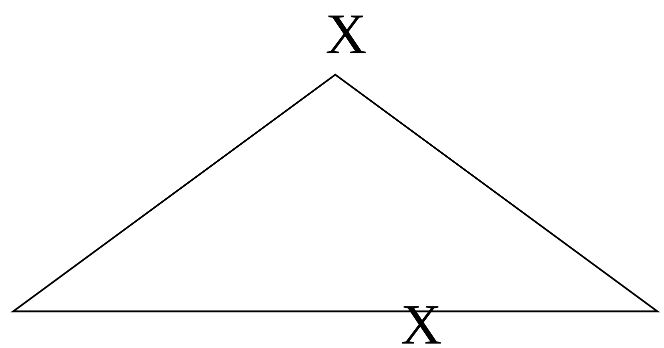
Elementary structures

- Elementary trees are maximal syntactic projections of lexical items.

Initial tree:



Auxiliary tree:

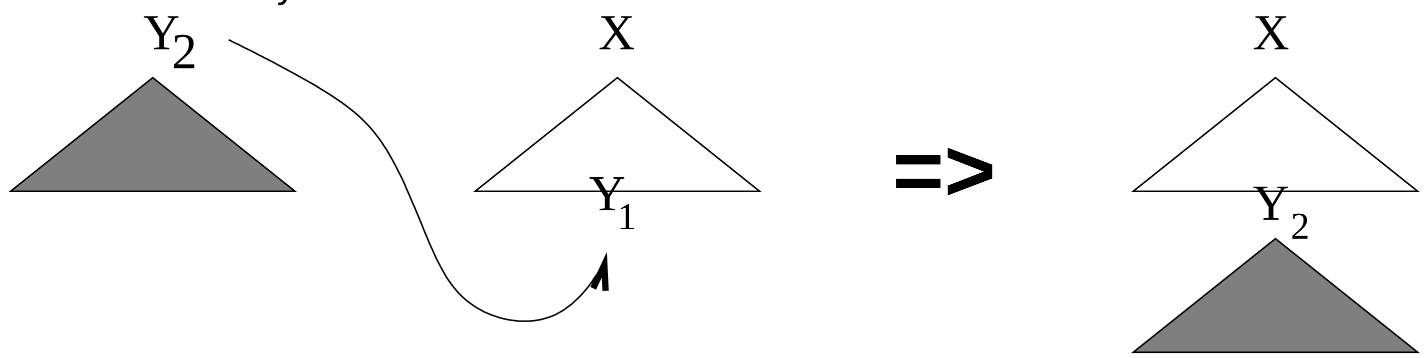


- *Alpha* trees.
- Recursion is not allowed in initial trees.
- Lexicalised trees have **anchors** on the frontier of the tree.

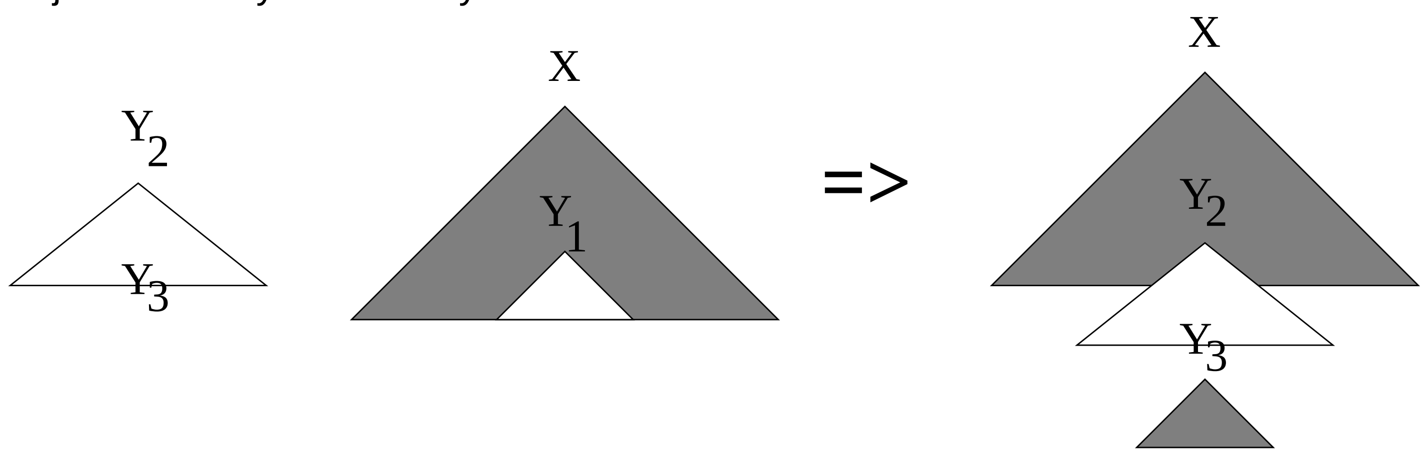
- *Beta* trees.
- Recursion allowed.
- Root and foot node must have the same label.

Operations

- Substitution: only for initial trees or lexical items.

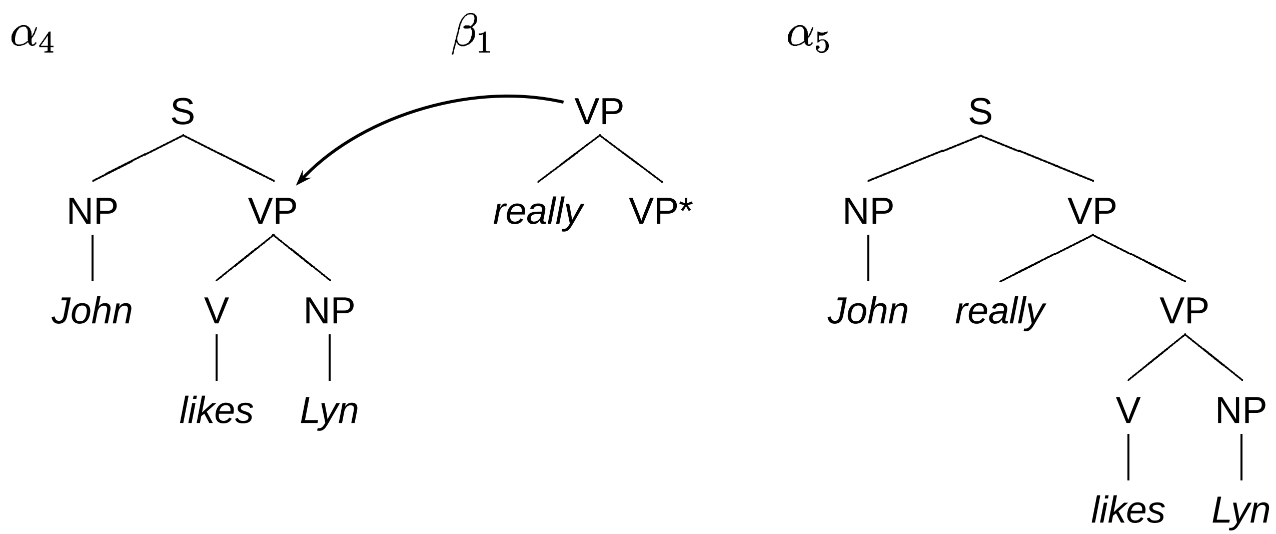


- Adjunction: only for auxiliary trees.



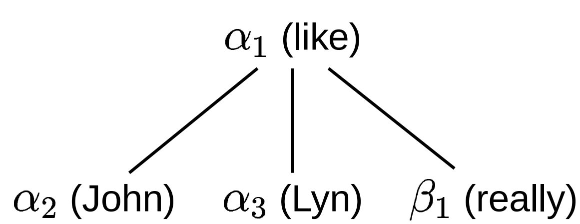
Adjunction example

- Adjunction of *really* into initial tree:



Derived trees and derivation trees

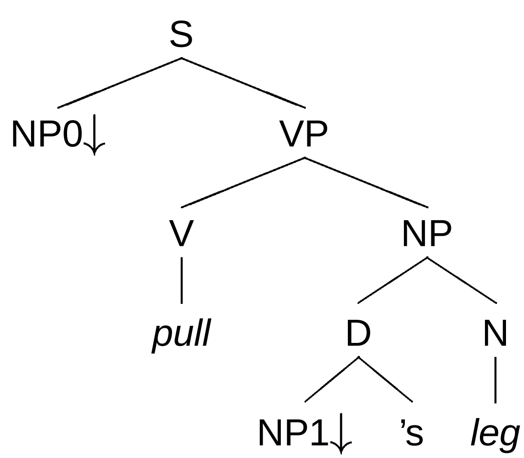
- A string-rewriting formalism, e.g. a CFG, derives a set of strings.
- A tree-rewriting formalism, e.g. a TAG, derives a tree: **derived tree**.
  - → Linguistic TAGs derive phrase structure trees.
- A **derivation tree** records how the derived string (CFG) or derived tree (TAG) was assembled from elementary rules (CFG) or elementary tree (TAG).
- Derivation tree for *John really likes Lyn*:



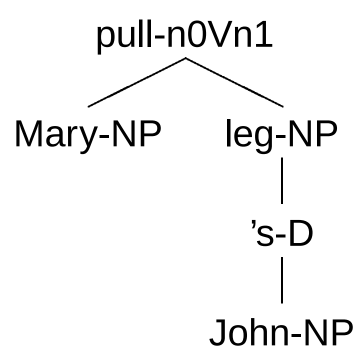
Derivation tree examples

- When derived treed are ambiguous, derivation trees might show the difference.
- Elementary tree for an idiomatic expression and two derivation trees for *Mary pull John’s leg*:

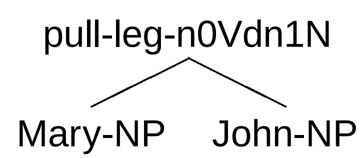
To pull NP’s leg



Literal reading



Idiomatic reading



Adjunction constraints and features

- Elementary tree nodes can be annotated with **adjunction constraints**.
  - → Selective adjoining constraint (SA): list of accepted trees.
  - → Null adjoining constraint (NA): empty list.
  - → Obligatory adjunction constraint (OA): boolean value.
- Nonterminal and terminal nodes ?
  - → NA nodes are nonterminal nodes that are not rewritten.
  - → OA nodes are nonterminal nodes that must be rewritten.
  - → SA nodes are either terminal or nonterminal nodes for tree rewriting.

Comparison of formal grammars

- Chomsky hierarchy for string rewriting systems

Grammar	Languages	Automaton	Production rules
Type-0	Recursively enumerable	Turing machine	No restrictions
Type-1	Context-sensitive	Linear-bounded non-deterministic Turing machine	$\alpha A \beta \rightarrow \alpha \gamma \beta$
Type-2	Context-free	Nondeterministic pushdown automaton	$A \rightarrow \gamma$
Type-3	Regular	Finite state automaton	$A \rightarrow aB$ $A \rightarrow a$

- Tree Adjoining Grammars are sronger than CFGs, but weaker than Context-sensitive grammars.

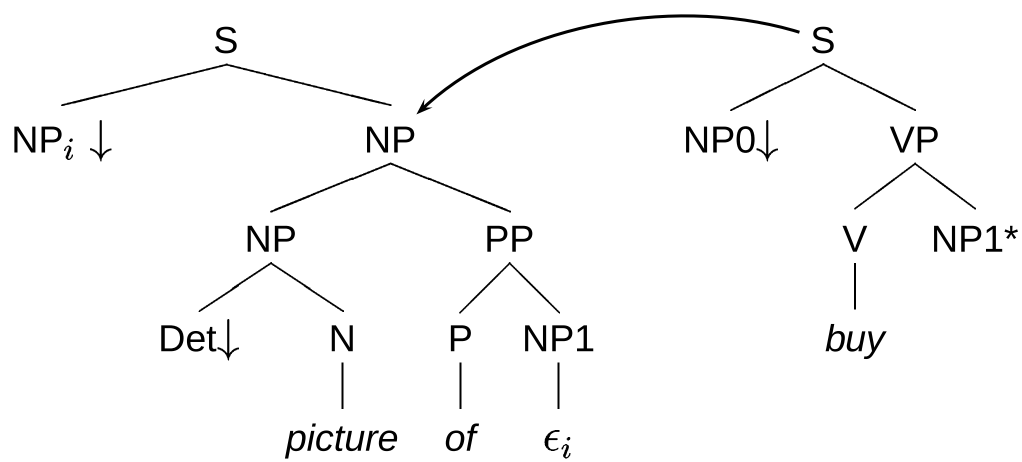
Formal properties of TAGs

- The set of languages generated by a TAG, $\mathcal{L}$(TAG), includes the set of languages generated by a context-free grammar, $\mathcal{L}$(CFG).
- Inclusion is proper, e.g. COUNT-4=$\{a^n b^n c^n d^n \mid n \geq 0\} \subset \mathcal{L}(TAG) \setminus \mathcal{L}(CFG)$
- Moreover, $\mathcal{L}$(TAG)$\subset$ $\mathcal{L}$(CSG), e.g. COUNT-5 $\subset$ $\mathcal{L}(CSG) \setminus \mathcal{L}(TAG)$
- Automaton: Embedded Pushdown Automaton with a stack of stacks of stack symbols as the push-down store.
- Tree-Adjoining Languages (TAL) are polynomially parsable, time complexity $O(n^6)$.

Extending the Power of TAG

- TAG cannot always provide a satisfactory analysis for linguistic constructions, e.g. *This building, John bought a picture of*.
- *This building* is the complement of the noun *picture* and should be substituted into an NP node in the same elementary tree as the head noun *picture*.

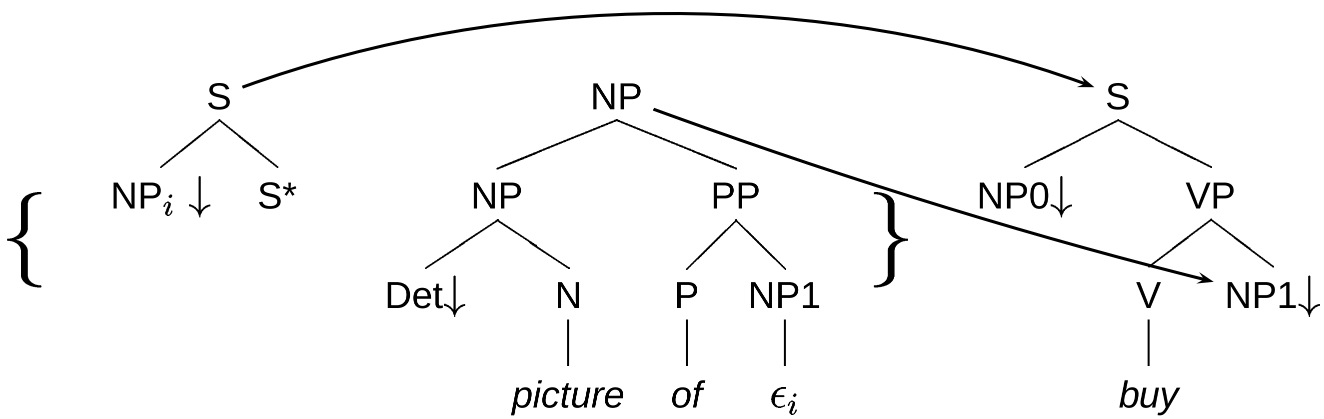
Illegal adjuntion:



Illegal auxiliary tree

Multicomponent TAGs (MC-TAG)

- Elementary sets are sets of trees rather than single trees.
  - → In a **tree-local multicomponent TAG**, all members of an elementary set must adjoin simultaneously into a single elementary tree.
  - → In a **set-local multicomponent TAG**, all members of a derived set of trees must adjoin simultaneously into trees from a single elementary set.



Synchronous TAGs (STAG)

- A Synchronous TAG relates the tree-adjoining grammars of two different languages.
- Definitions for node to node correspondence, lexical entries, feature transfer.
  - Application areas include machine translation, language generation, semantic analysis, etc.
- A typical transfer algorithm for machine translation:
  - Parse the source sentence according to the source grammar.
  - Map each elementary tree in the source derivation tree with a tree in the target derivation tree according to the **transfer lexicon**.
  - Read the target sentence off the target derivation tree.
- Example.

TAG recognition and parsing

- A bottom-up chart parser proceeds bottom-up in recognising the elementary trees used in a derivation and assembling the elementary trees into a derivation. Worst and best case time complexity $O(n^6)$.
- Earley-style algorithms combine bottom-up parsing with top-down prediction on derived trees. Worst case time complexity $O(n^9)$ – $O(n^6)$, faster in an average case.
- Head-driven algorithms extends parses along the path from the anchor of an elementary tree to its root by performing adjunctions. Worst case time complexity $O(n^6)$.
- Algorithms based on kernel grammars (a CFG) parse the input twice. In the second step, TAG-incompatible derivations are eliminated from the context-free parse forest. Worst case time complexity $O(n^6)$.
- Several other parsing algorithms exist.

Today...

- Project work topics — introduction and selection.
- Presentation schedule.
- Delivery of exercises for next week.