Syntactic Theory

Tree-Adjoining Grammar (TAG)

Yi Zhang

Department of Computational Linguistics
Saarland University

November 17th, 2009

Outline

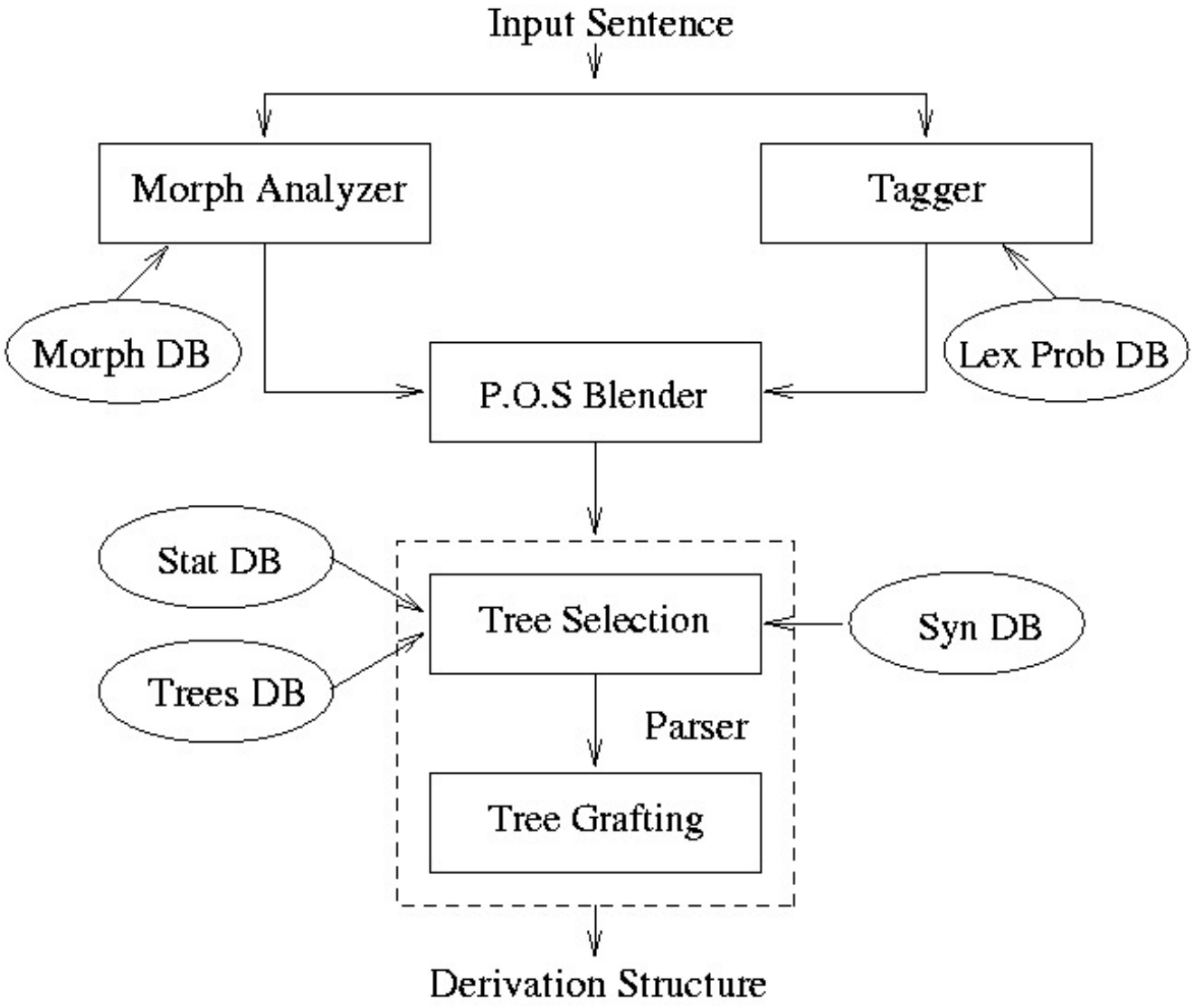
XTAG: A Lexicalized Tree Adjoining Grammar for English

NLP with Tree-Adjoining Grammars

The XTAG Project

- A long-term project to develop a wide-coverage grammar for English using the Lexicalized Tree-Adjoining Grammar (LTAG) formalism
- Provides a grammar engineering platform consisting of a parser, a grammar development interface, and a morphological analyzer
- The project extends to variants of the formalism, and languages other than English

Overview of the XTAG System



Components in the XTAG System

- Morphological Analyzer & Morph DB: 317K inflected items derived from over 90K stems
- POS Tagger & Lex Prob DB: Wall Street Journal-trained 3-gram tagger with *N*-best POS sequences
- Syntactic DB: over 30K entries, each consisting of:
  - Uninflected form of the word
  - POS
  - List of trees or tree-families associated with the word
  - List of feature equations
- Tree DB: 1004 trees, divided into 53 tree families and 221 individual trees

Overview of the Grammar Design

- Tree family: a set of elementary trees for predicative words to represent the linguistic notion of subcategorization
- Complements & Adjuncts
  - Complements are included in the elementary tree anchored by the word that selects them
  - Adjuncts do not originate in the same elementary tree as the head word; they are added to a structure by adjunction
- Non-S constituent trees does not group into families
- Nouns carry case with them, which is checked against the verbs by unification

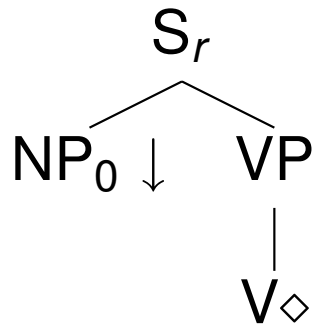
Example Verb Classes

Declarative Intransitive Tree

Example

eat, sleep, dance

- *Al ate.*
- *Seth slept.*
- *Hyun danced.*



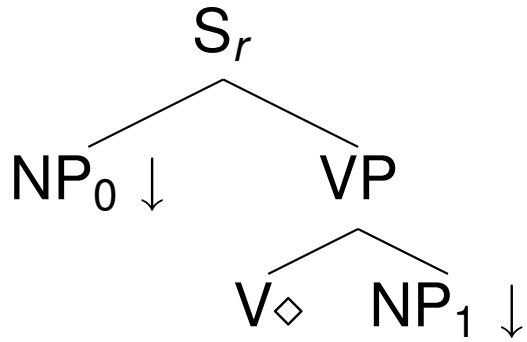
Example Verb Classes

Declarative Transitive Tree

Example

eat, dance, take, like

- *Al ate an apple.*
- *Seth danced the tango.*
- *Hyun is taking an algorithms course.*
- *Anoop likes the fact that the semester is finished.*



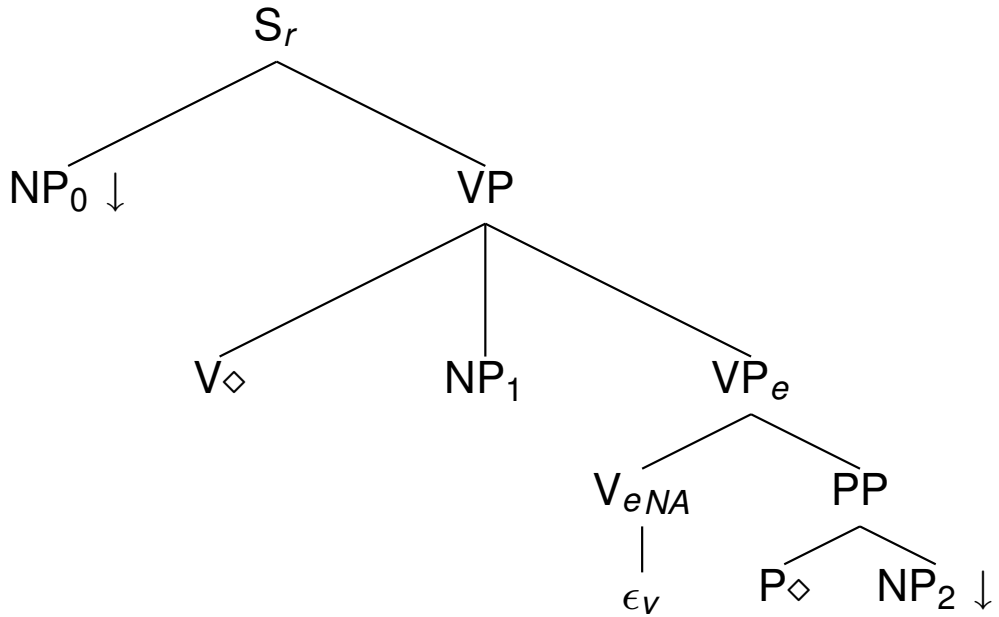
Example Verb Classes

Declarative Multiple Anchor Ditransitive with PP Tree

Example

gear for, give to, remind of

- *The attorney geared his client for the trial.*
- *He gave the book to his teacher.*
- *The city reminded John of his home town.*



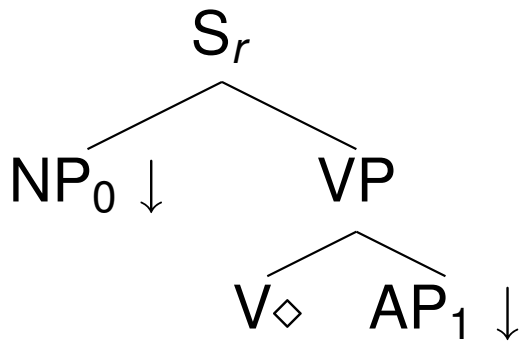
Example Verb Classes

Declarative Intransitive with Adjective Tree

Example

become, grow, smell

- *The greenhouse became hotter.*
- *The plants grew tall and strong.*
- *The flowers smelled wonderful.*



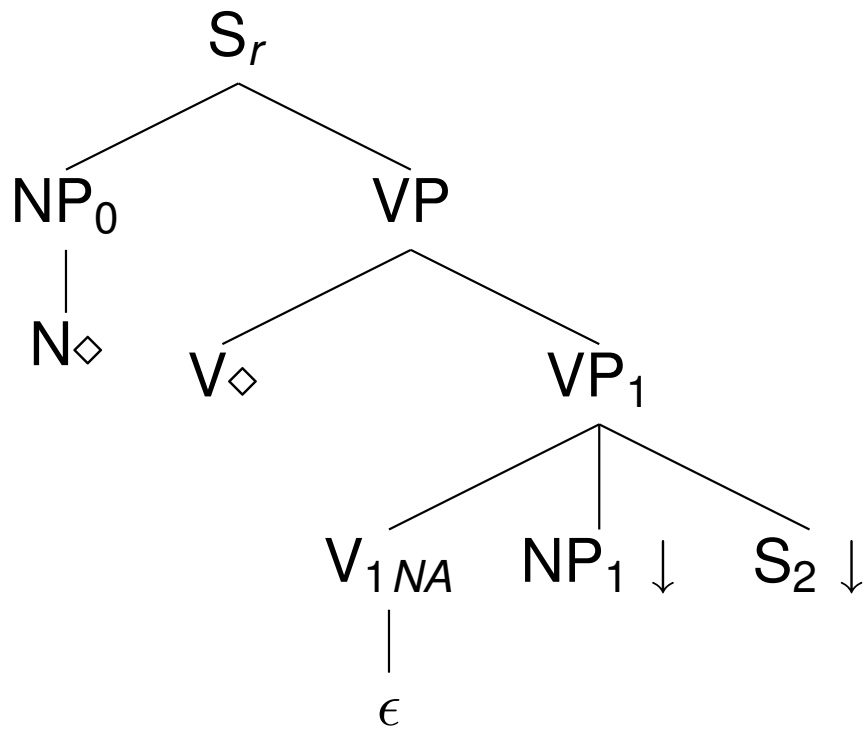
Example Verb Classes

Declarative NP It-Cleft Tree

Example

it be

- *It was yesterday that we had the meeting.*



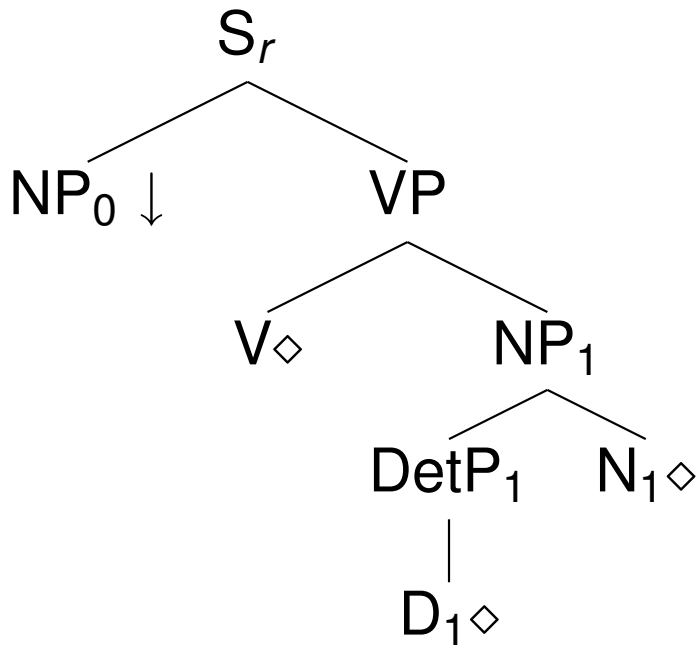
Example Verb Classes

Declarative Transitive Idiom Tree

Example

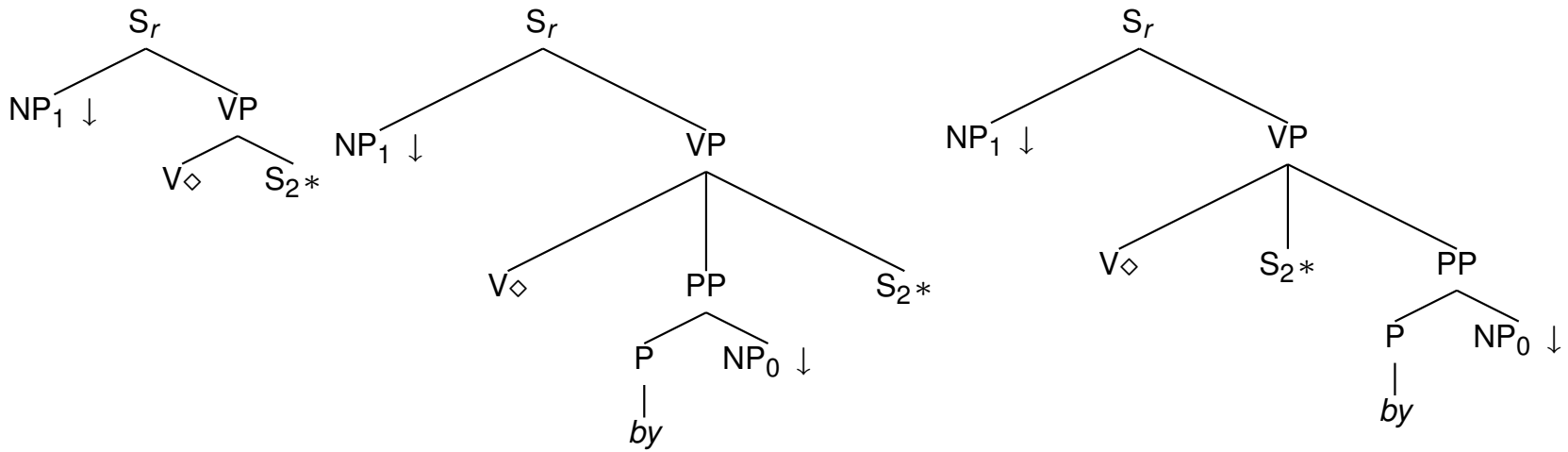
kick the bucket, bury the hatchet, break the ice

- *Nixon kicked the bucket.*
- *The opponents finally buried the hatchet.*
- *The group activity really broke the ice.*



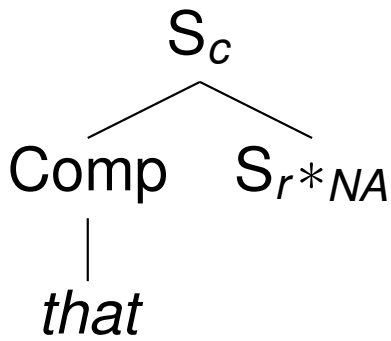
Passives

- The subject NP is interpreted as having the same role as the direct object NP in the active declarative



Complementizer

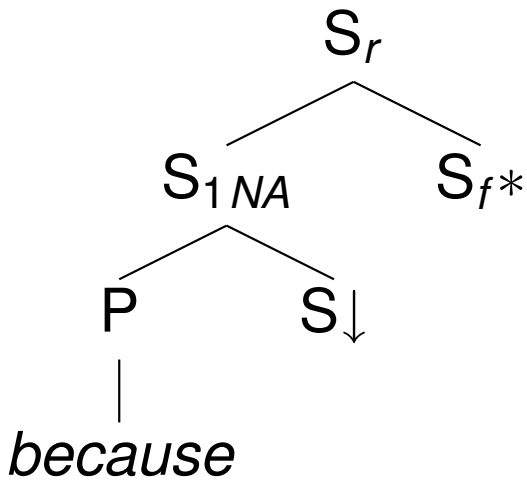
- **mode** feature constrains the type of clauses to which the complementizer adjoins
- **comp** and **wh** feature receives their root node value from the complementizer and ensure no stacked complementizer with **comp=nil** on the foot node



(Feature structures on the whiteboard)

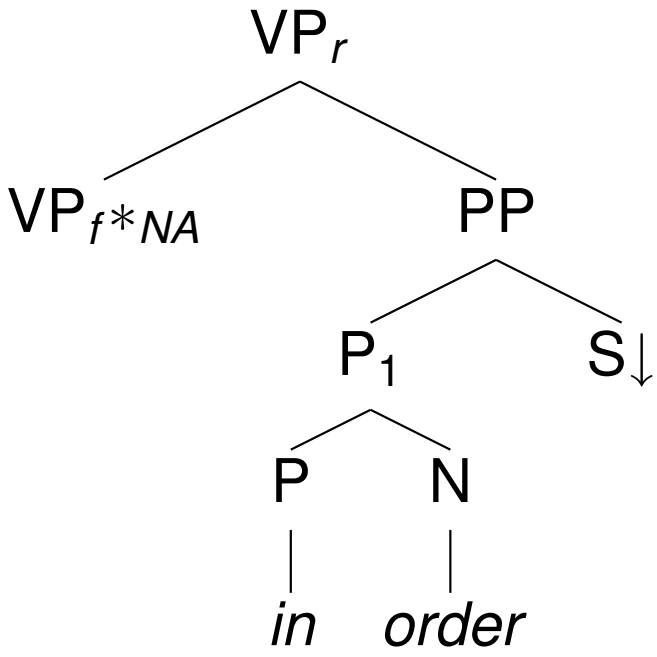
Subordinate Clauses

- Before the matrix clause



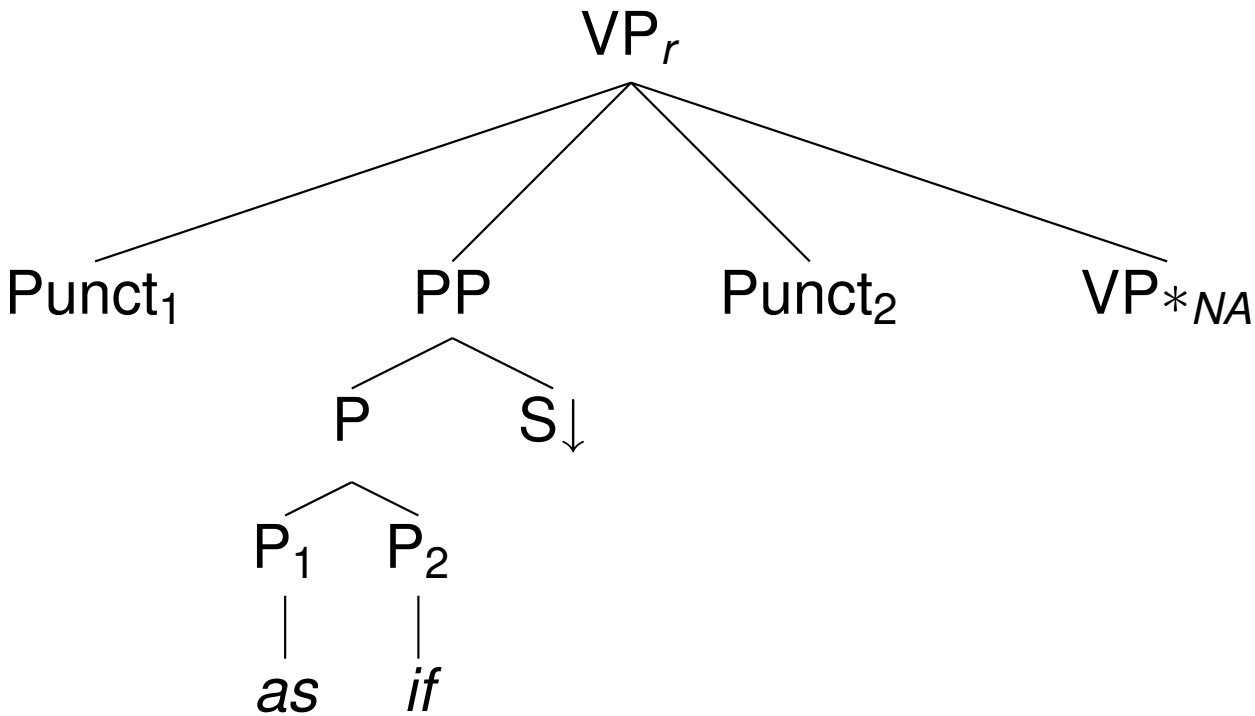
Subordinate Clauses

- After the matrix clause



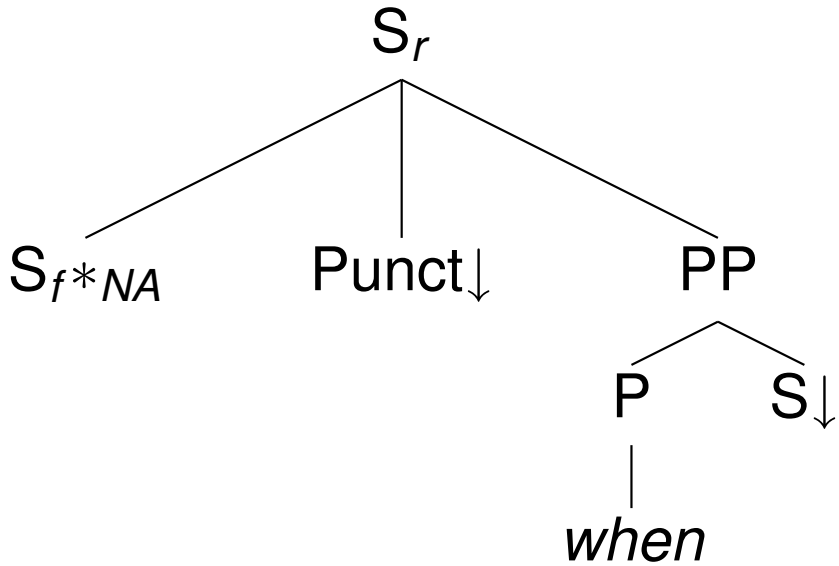
Subordinate Clauses

- Before the VP, surrounded by two punctuation marks



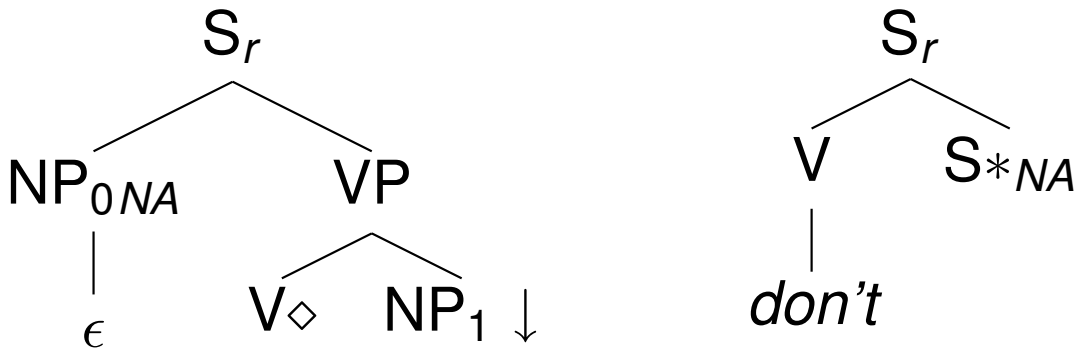
Subordinate Clauses

- After the matrix clause, separated by a punctuation mark



Imperatives

- An overt subject with second person
- Negative imperative is done by adjoining *don't* to the root



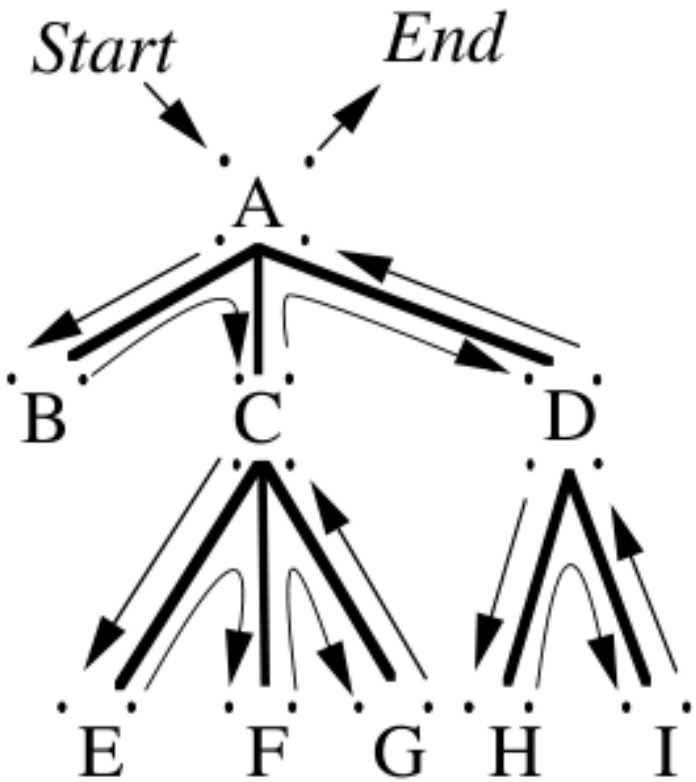
Outline

XTAG: A Lexicalized Tree Adjoining Grammar for English

NLP with Tree-Adjoining Grammars

Parsing Tree-Adjoining Grammar

- A CKY-like algorithm runs in $O(n^6)$ time
- An Earley-like algorithm (with 4 operations: SCAN, PREDICT, COMPLETE, ADJOIN) can reduce the average time complexity by top-down prediction



Supertagging

- The large number of elementary trees pose a computational challenge
- One can view an elementary tree as a *supertag*, and use statistical models to assign the most likely (n) supertag(s) for a given word within particular context
- Efficiency can be greatly enhanced without visible loss in parsing accuracy

References I


Joshi, A. and Schabes, Y. (1997).
Tree-adjoining grammars.