

Natural Language Processing, 60 years after the Chomsky-Schützenberger hierarchy

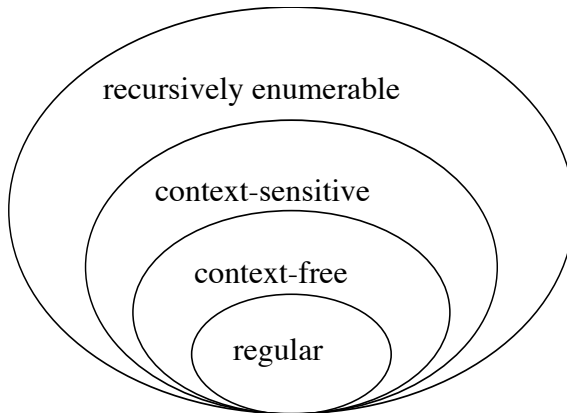
Laurence Danlos

(avec Benoît Crabbé)

Université Paris Diderot-Paris 7, Alpage, IUF

21 Mars 2016

Chomsky-Schützenberger hierarchy



Chomsky-Schützenberger hierarchy

Class	Grammars	Languages	Automaton
Type-0	Unrestricted	Recursively enumerable (Turing-recognizable)	Turing machine
Type-1	Context-sensitive	Context-sensitive	Linear-bounded
Type-2	Context-free	Context-free	Pushdown
Type-3	Regular	Regular	Finite

Computer programs *versus* Natural language texts in the 50'

Computer programs

The syntax analysis of a computer program can be based only on a CFG (with procedures to construct meaning)

Natural language texts

Linguistic research in Chomsky (1957, 1965) lead to a more complex formal system: the model is both generative (CFG) and transformational

Transformational grammars

- Chomsky (1957, 1965) posits that each sentence in a language has two levels of representation:
 - deep structure: canonical structure, from which semantics can be computed
 - surface structure: syntactic representation, from which phonology can be computed
- Deep structures are mapped onto surface structures via transformations

Transformational model in the 60'

Two components

- The generative component based on a CFG generates only deep structures for canonical clauses such as (1a)
- The transformational component generates surface changes from canonical structures, passive transformation (1b), WH transformation (1c), two transformations (1d)

- (1) a. The student put the book on the shelf
b. The book was put on the shelf
c. Who put the book on the shelf?
d. Which book was put on the shelf?

2016 Lecture on formal grammars

by Bob Hardin (Western Michigan University)



- The syntax of most programming languages is context-free (or very close to it)
- Natural language is almost entirely definable by type-2 tree structures
- Syntax of some natural languages (Germanic) is type-1

Is it true?

What are the results in NLP after 60 years of research?

Outline

- 1 Research in the 70'
- 2 Research in the 80'
- 3 Extension of LTAG for semantics and discourse
- 4 Nowadays NLP applied research
- 5 Schützenberger and AI

Leaving aside transformational model in NLP

The model has poor computational properties

Peters and Ritchie (1973) establish its undecidability

Formalism GPSG (Generalized Phrase Structure Grammar)
(Gazdar et al 1985)

no transformational component but use of features and a metagrammar (to automatically generate new rules)
GPSG inspired by computer science development

The hypothesis is still that natural language syntax can be described with a CFG

(although GPSG actually defines a more general class of languages than CFG)

Rehabilitation of lexical information

Chomsky's model

Nearly nothing about lexical information: just the arguments of verbal predicates, e.g. *sleep* is intransitive, *eat* is transitive

Importance of lexical information

- development of electronic lexicons
 - Maurice Gross for French
 - Beth Levin for English
- development of grammars with lexical information, e.g. categorial grammars (Lambek 1958)

Outline

- 1 Research in the 70'
- 2 **Research in the 80'**
- 3 Extension of LTAG for semantics and discourse
- 4 Nowadays NLP applied research
- 5 Schützenberger and AI

Existence of cross dependencies

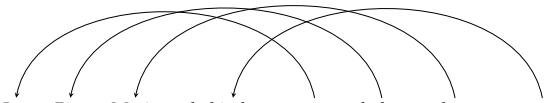
Swiss German (Shieber 1985)

Jan saït das mer em Hans es huus hälfed aastriiche
 John says that nous Hans.DAT the house.ACC help+DAT paint+ACC
 (Jean says we help Hans to paint the house)

Jan saït das mer em Hans es huus hälfed aastriiche



... dat Jan Piet Marie de kinderen zag helpen leren zwemmen



Context sensitive phenomenon

$$L(G) = \{ww \mid w \in \{a, b\}^*\}$$

Tree Adjoining Grammar (TAG) (Joshi 1986)

Two sets of trees

- initial trees
- auxiliary trees

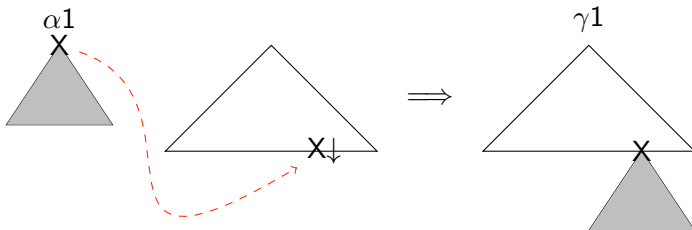
Two operations

- substitution
- adjunction

Substitution operation

Substitution of the initial tree $\alpha 1$ (root node X)

in a tree with a substitution node X on the frontier marked with a $\downarrow$

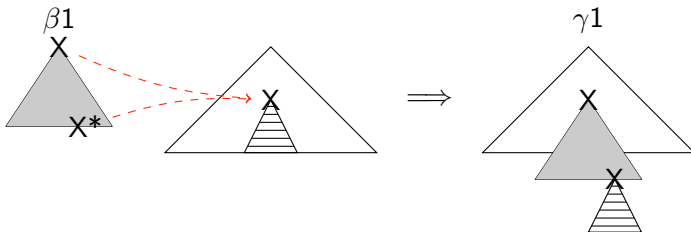


Adjunction operation

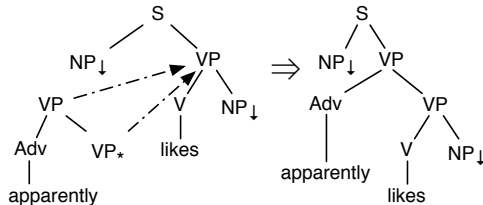
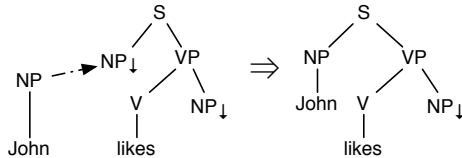
Adjunction of the auxiliary tree β_1

root node: labelled X (non terminal)

on the frontier: “foot node” also labelled X and marked with *



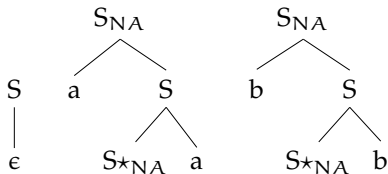
Example of substitution and adjunction operations



Cross dependencies in TAG

Grammar G with $L(G) = \{ww \mid w \in \{a, b\}^*\}$

NA: non adjunction



Midly context-sensitive languages

TAG is part of a class of languages called midly context-sensitive

This class is a superset of context-free languages and a subset of context-sensitive languages

(3-copy language $L3 = \{www | w \in \{a, b\}^*\}$ cannot be generated by a TAG)

Parsing in TAG is made in polynomial time $O(n^6)$

Embedded Pushdown Automata (Vijay-Shanker, 1987)

While CFGs are associated with pushdown automata (PDA), TAGs are associated with the so-called Embedded Pushdown Automata (EPDA)

Lexicalized grammars

Definitions

- A rule is lexicalized if it has a lexical (terminal) anchor
- A grammar is lexicalized if all its rules are lexicalized

Lexicalization of a CFG?

Can a CFG be lexicalized?

i.e., given a CFG, G , can we construct another CFG, G' , such that

- every rule in G' is lexicalized,
- and G and G' are strongly equivalent?

Simple example of Lexicalization of a CFG by a TSG

Tree Substitution Grammar

TSG is TAG without the adjunction operation (only initial trees)

CFG G

$S \rightarrow NP VP$

$VP \rightarrow V NP$

$NP \rightarrow$

Harry

$NP \rightarrow$

peanuts

$V \rightarrow$

likes

TSG G'

$\alpha 1$

S

$NP \downarrow$

VP

V

likes

$\alpha 2$

NP

peanuts

$\alpha 3$

NP

Harry

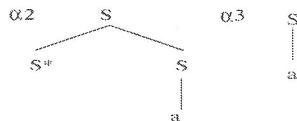
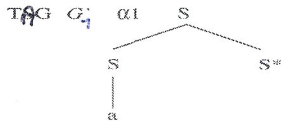
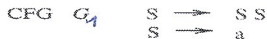
No Lexicalisation of a CFG by a TSG

CFG G

$$\begin{array}{lcl} S & \longrightarrow & S S \text{ (non-lexical)} \\ S & \longrightarrow & a \text{ (lexical)} \end{array}$$


G and G' are weakly but not strongly equivalent

TAG strong lexicalisation of a CFG

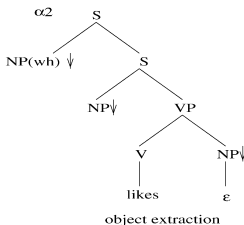
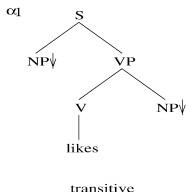


G and G' are strongly equivalent

Lexicalized TAG (LTAG)

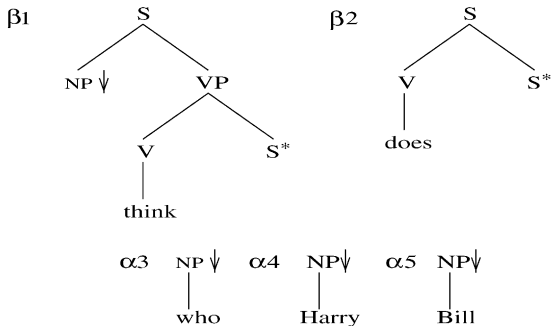
Two elementary trees for *likes*

- transitive : *Harry likes Mary*
- object extraction: *Who does (Bill think) Harry likes?*

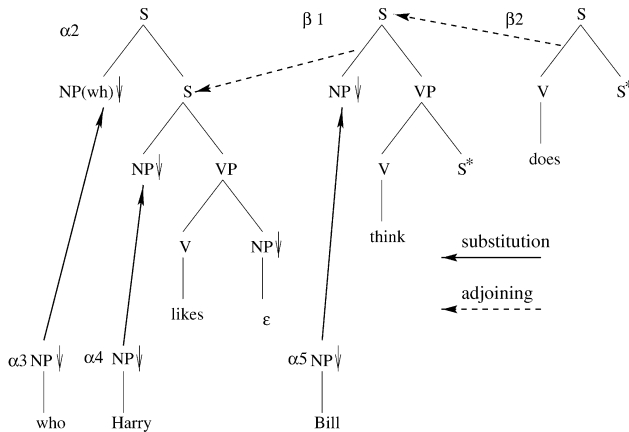


- all recursion has been factored away because dependencies are localized in the elementary trees
- $\Rightarrow$ no long distance dependencies as such

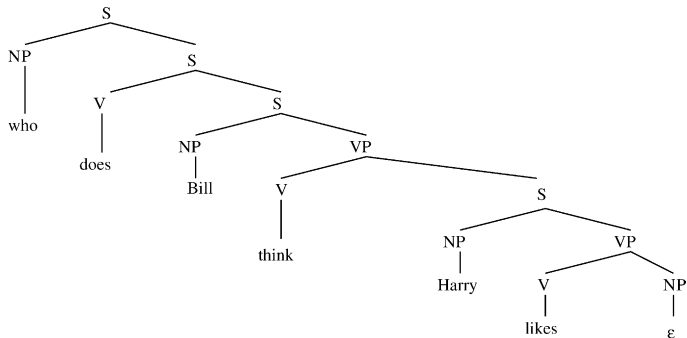
Other elementary trees in a LTAG



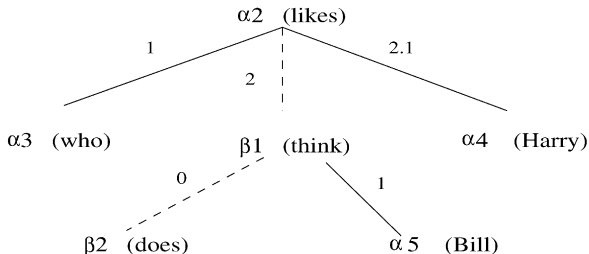
LTAG derivation for *who does Bill think Harry likes*



LTAG derived tree for *who does Bill think Harry likes*



LTAG derivation tree for *who does Bill think Harry likes*



Summary

The class of mildly context-sensitive grammars appears appropriate for modeling natural languages

- Development of LTAGs for many languages: English, French, German, Korean, etc.
- Study of mildly context-sensitive grammars, e.g. LCFRS (Weir 1988) or RCG (Boullier 2003)

Outline

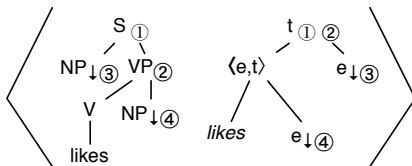
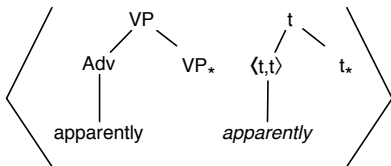
- 1 Research in the 70'
- 2 Research in the 80'
- 3 Extension of LTAG for semantics and discourse
- 4 Nowadays NLP applied research
- 5 Schützenberger and AI

Semantics: Synchronous TAG (STAG) (Shieber 1994)

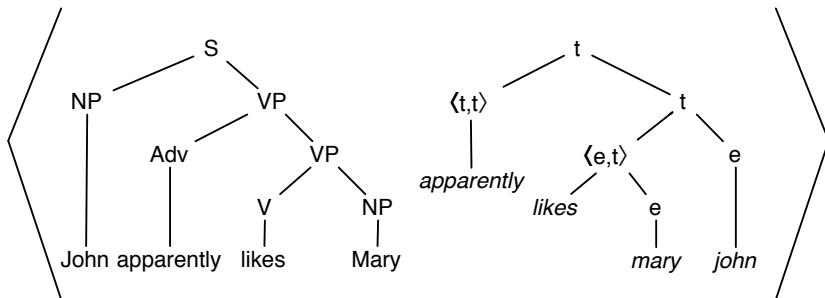
- Synchronous TAG (STAG) extends TAG by taking the elementary structures to be **pairs of TAG trees** with links between particular nodes in those trees.
- An STAG is a set of triples, $\langle t_L, t_R, \curvearrowright \rangle$ where t_L and t_R are elementary TAG trees and $\curvearrowright$ is a linking relation between nodes in t_L and nodes in t_R
Links are marked with circled indices (e.g. ①)
- Derivation proceeds as in TAG except that all operations must be paired. That is, a tree can only be substituted or adjoined at a node if its pair is simultaneously substituted or adjoined at a linked node.

An English syntax/semantics STAG fragment for *John apparently likes Mary*

(From Nesson and Shieber 2006)



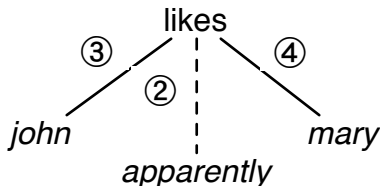
Derived tree pair for *John apparently likes Mary.*



Resulting semantic representation can be read off the semantic derived tree by treating the leftmost child of a node as a functor and its siblings as its arguments: *apparently(likes(john, mary))*

Derivation tree for *John apparently likes Mary.*

- Only one derivation tree for both the syntactic and semantic representations.
- Each link in the derivation tree specifies a link number in the elementary tree pair.



Extension of STAG to discourse (D-STAG)

(Danlos 2009)

- At the discourse level, sentences and propositions are related by *discourse relations* (DRs).
- DRs can either be *implicit* — semantically inferred — (2a), or *explicit* —lexically signalled — (2b).
- The most common markers of explicit DRs are *discourse connectives* (DCs), a group mainly composed of conjunctions, prepositions and adverbs.

- (2) a. *Fred was sick.* **He stayed at home.**
b. *Fred was sick.* But **he came to work.**

D-STAG

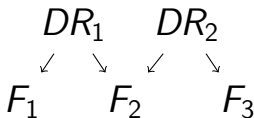
- syntactic trees anchored by discourse connectives (DCs)
- semantic trees anchored by discourse relations (DRs)

Discourse parsing should produce non tree-shaped semantic graphs

[Fred is in a bad mood]₁ because [he didn't sleep well]₂. [He had nightmares]₃.

Interpretation

$Explanation(F_1, F_2) \wedge Explanation(F_2, F_3)$

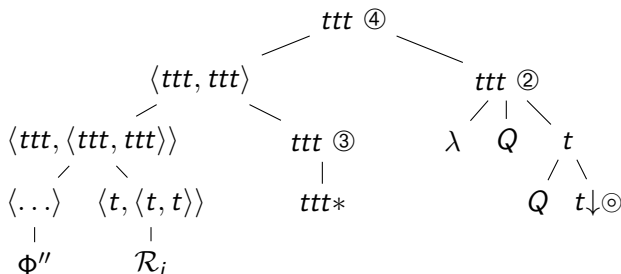


Functor which allows a copy of F2

$$\Phi'' = \lambda \mathcal{R}_i X Y P. X(\lambda x. Y(\lambda y. \mathcal{R}_i(x, y) \wedge P(x)))$$

$$\Phi''(\mathcal{R}_i) = \mathcal{R}''_i = \lambda X Y P. X(\lambda x. Y(\lambda y. \mathcal{R}_i(x, y) \wedge P(x)))$$

with $X, Y : ttt = \langle \langle t, t \rangle, t \rangle$, $P : \langle t, t \rangle$ and $x, y : t$



$$\lambda P. (\mathcal{R}_i(F_2, F_3) \wedge P(F_2))$$

Outline

- 1 Research in the 70'
- 2 Research in the 80'
- 3 Extension of LTAG for semantics and discourse
- 4 Nowadays NLP applied research**
- 5 Schützenberger and AI

Competence *versus* Performance Grammars

All the (symbolic) grammars presented so far are essentially *competence* grammars

They do not model *performance*

they are not robust enough to deal with the phenomena that are found in "real" texts, e. g. journalistic texts.

Nowadays research in NLP

work on *real* texts for applications such as Information Retrieval and Text Mining, Social Media and Sentiment Analysis, Machine Translation ...

Main techniques for *applied work*

Machine learning techniques

Probabilistic context free grammars (PCFGs)

Basic principles

- The probability of a derivation (parse) is the product of the probabilities of the productions used in that derivation.
- These probabilities are typically computed by machine learning on annotated corpora (PTB, FTB).

Big problems

- The size of the grammar increases with the size of the explored data (quite numerous rules with low frequency)
- The ambiguity (number of parse trees for the same input) increases drastically with the size of the grammar

New trends of theoretical research with cognitive aspects

Outline

- 1 Research in the 70'
- 2 Research in the 80'
- 3 Extension of LTAG for semantics and discourse
- 4 Nowadays NLP applied research
- 5 **Schützenberger and AI**

M.P. Schützenberger's talk on AI around 1985

AI goal

How to create computer programs that simulate intelligent human behavior?

M.P. Schützenberger's prediction

It is not computers which are going to simulate human behavior but humans who are going to simulate and adapt to computer behavior

by comparison with alchimistry

M.P. Schützenberger's prediction for NLP 30 years later

Right

For any booking task or alike, we don't use anymore natural language
but fill drop-down menus on the Web

Wrong

Human beings have never written so much (with their thumbs)

So computers should understand their natural language texts (for machine translation, for example)

Which formal grammars for so-called noisy user-generated texts?

English

@Hii_ImFruitiy nuin much at all juss chillin waddup w yu ?

French

jlaime trp ste meuf on stape trp des bar tmtc

