History of Grammar Formalisms

Grammar Formalisms: 11-722

Spring Term, 2004

Lori Levin

What is a Grammar Formalism?
From Formal Language Theory

- A language
  - A language L is a possibly infinite set of strings.
  - The strings are made from a finite alphabet.
    - The “alphabet” might be the words of English
    - Henceforth, we will call it the “vocabulary”
  - Some strings of language L:
    - Bears live in the forest.
    - Never have I seen such ridiculous beasts.
  - Some strings are not in L:
    - *Never I have seen such ridiculous beasts.
    - *Live bears the in forest.
    - (* means that the string is not a member of the set of strings that comprise the language L.)

What is a Grammar Formalism?
From Formal Language Theory

- A Grammar:
  - A set of production rules.
  - In addition to the vocabulary, the production rules can use other symbols
    - N (noun)
    - V (verb)
    - NP (noun phrase)
    - VP (verb phrase)
  - One symbol is special:
    - S (sentence)

What is a grammar formalism? Production Rules

- S → NP VP
- NP → Det N
- VP → V NP
- DET → the
- DET → a
- N → boy
- N → girl
- V → saw
- V → sees

These production rules have a non-terminal symbol (one that isn't from the vocabuary) on the left, then an arrow, then some terminal (from the vocabulary) and non-terminal symbols on the right.

This is one instance of a grammar formalism.

We will see that other grammar formalisms use other types of symbols and production rules.

What is a grammar formalism? Derivation

- The production rules are interpreted as instructions:
  - Parsing: when you find the string on the right hand side, replace it with the string on the left hand side.
  - Generation: when you find the symbol on the left hand side, replace it with the string on the right hand side.
  - Different grammar formalisms will have different instructions.
- Your job:
  - Generation: get from the special symbol S to a terminal string (only symbols from the vocabulary).
  - Parsing: get from a terminal string to the special symbol S

What is a grammar formalism? Derivation

- A derivation is the ordered list of production rules that you use to get from the special symbol to the terminal string or vice versa.
- S
- NP VP
- Det N VP
- Det N V NP
- Det N V Det N
- The N V Det N
- The girl V Det N
- The girl sees Det N
- The girl sees a N
- The girl sees a boy

S → NP VP
NP → Det N
VP → V NP
DET → the
DET → a
N → boy
N → girl
V → saw
V → sees

Reference

- Hopcroft, Motwani, and Ullman, Introduction to Automata Theory, Languages, and Computation, second edition, Addison-Wesley, 2001.
  - Chapter 5.2

What does a grammar formalism look like?

- Context Free Phrase Structure Grammar:

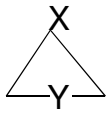
  S → NP VP

- Lexical Functional Grammar:

  S → NP VP

  (↑ SUBJ)=↓ ↑=↓

- Tree Adjoining Grammar



- Categorial Grammar

  X → X/Y Y

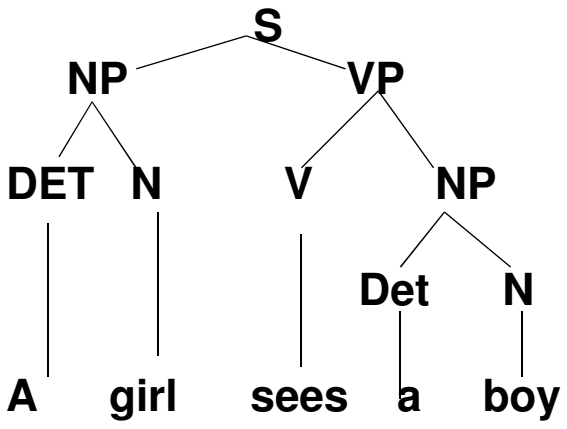
  (an X consists of an X missing a Y and a Y.)

What is a Grammar Formalism for?

- Distinguish strings that are in the language from those that are not in the language.
  - The girl sees a boy.
  - *Girl the the.
    - No derivation exists using the grammar rules.
- Identify a structure for the sentence.

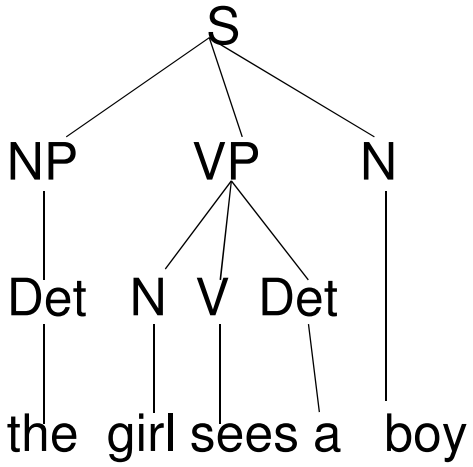
Structure

- S
- NP VP
- Det N VP
- Det N V NP
- Det N V Det N
- The N V Det N
- The girl V Det N
- The girl sees Det N
- The girl sees a N
- The girl sees a boy



This grammar makes wrong structures

- S → NP VP N
- NP → Det
- VP → N V Det
- DET → the
- DET → a
- N → boy
- N → girl
- V → saw
- V → sees



History of Grammar Formalisms

- 500 B.C.
  - Pānini's grammar of Sanskrit, *Astadhyayi*, contains production rules (!!!) for Sanskrit phonology, morphology, and grammar.
  - This is a lasting work of genius, still studied today, and remarkably similar to some modern linguistic theories.
- 7th to 8th centuries A.D.
  - Classical Arabic Grammarians define Classical Arabic.
- 19th century Europe
  - No production rules. They spend all their time on historical and comparative linguistics – finding genetic relationships among languages:
  - The Grimm brothers collect fairy tales, but are actually working on systematic sound correspondences between branches of Indo-European:
    - p-f: père – father; pied – foot; etc.

Ferdinand de Saussure
(1857-1913)

History of Grammar Formalisms
Late 19th to Early 20th centuries

- Ferdinand de Saussure
  - Ferdinand de Saussure starts out as a historical linguist, and then comes back to the present.
    - Distinguishes synchronic linguistics from diachronic linguistics
  - Linguistics gets mental:
    - Signifier and signified:
      - The signifier is a word, like “tree”. The signified is not a tree, but a concept of a tree.
    - *Langue* (language) and *Parole* (speech)
      - http://www.press.jhu.edu/books/hopkins_guide_to_literary_theory/ferdinand_de_saussure.html

      “… he distinguishes between the particular occurrences of language – its particular "speech-events," which he designates as *parole* –and the proper object of linguistics, the system (or "code") governing those events, which he designates as *langue.”*

History of Grammar Formalisms
Late 19th to Early 20th centuries

- Ferdinand de Saussure (continued)
  - Structuralism:
    - Elements of language are like pieces on a chess board. They only make sense when you consider their role in relation to other pieces.
  - De Saussure's students write up his lectures in *Course de Linguistique Générale.*
  - Structuralism becomes big in linguistics, literary theory, art, anthropology, etc.

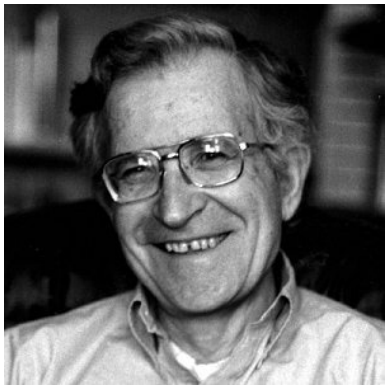
History of Grammar Formalisms

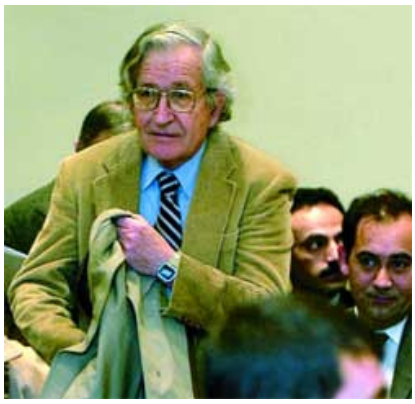
- Early 20th Century
  - Linguistics gets scientific
    - Structuralists focus on discovery procedures for phonemes and morphemes.
    - Structuralism is *empiricist:*
      - Focus is on what is attested in the data
      - The source of knowledge is sensory experience
    - Some linguists use formal notation.
    - But structuralism does not focus on a set of production rules that define which strings are members of a language and which are not.
      - No derivations?

References

- R. H. Robbins, *A Short History of Linguistics*, Indiana University Press, 1967.
- Frederick Newmeyer, *Lingiustic Theory in America*, Academic Press, 1982.

Noam Chomsky
1928 –





History of Grammar Formalisms

- Mid to Late 20th century: Generative Grammar
- Before Chomsky:
  - Post
  - Bar Hillel
  - Adjukiewicz
  - Zellig Harris (Chomsky's professor)
    - "Kernel" sentences become the model for Chomsky's deep structures. Derived sentences are the model for surface structures.
      - The police arrested the thief. (Kernel)
      - The thief was arrested by the police.
      - It was the thief who was arrested by the police.
  - Shannon and Weaver: information theory
    - Structuralists pay some attention to this, but then it falls out of fashion for human language until about 1980.
    - Becomes the basis for statistical NLP.

History of Grammar Formalisms

- 1950's: Chomsky's book *The Logical Structure of Linguistic Theory* is not published.
  - It was published in 1975
- 1957: Chomsky's *Syntactic Structures* is published.
  - Generative Grammar is *rationalist*:
    - The source of knowledge is reason, not experience.
      - http://www.whps.org/schools/norfeldt/libraryweb/MediaResources/TermPaper.PDF
    - Grammaticality judgments are legitimate data.
    - Generalization beyond what is in a corpus.
    - Language is an infinite set of sentences defined by a finite set of rules.

History of Grammar Formalisms

- *Syntactic Structures* (continued)
  - Human language cannot be described adequately by finite state machines.
    - If….either…or… then…
    - If either it's raining or it's snowing, then we won't go outside.
      - You could get a given finite number of these right, but your internal grammar tells you that the embedding is potentially infinite.
    - *If …either….either…then… or….

History of Grammar Formalisms

- *Syntactic Structures* (continued)
  - Human language cannot be described well by context free grammars
    - Agreement
      - S → NP-sg VP-sg
      - S → NP-pl VP pl
      - The girl sings.
      - The girls sing.
      - *The girl sing.
      - *The girls sings.

History of Grammar Formalisms

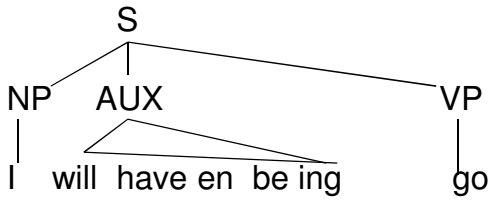
- *Syntactic Structures* (continued)
  - Add transformations – tree-to-tree mappings – to generative grammars.
  - Structuralism:
    - Order of English auxiliary verbs:
    - Modal have be (prog) be (pass)
    - Modal must be followed by infinitive
    - *Have* must be followed by past participle
    - *Be* (prog) must be followed by present participle
    - *Be* (pass) must be followed by a past participle in passive voice.

History of Grammar Formalisms

- I will go.
- *I will gone/going/went.
- I have gone.
- *I have go/gone/going/went.
- I am going.
- *I am go/went.
  - I am gone (adjective).
- I will have been being arrested.
- I will have been singing.
- *I will be having sung. (wrong order)
- *I will be sung. (wrong verb form.)
- Etc.

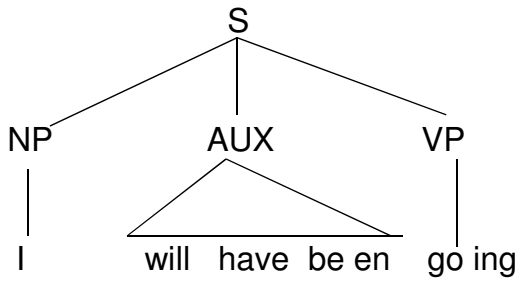
Affix Hopping

- Base rule:
  - AUX → modal (have –en) (be –ing)
  - Parentheses mean optionality
  - Deep Structure



History of Grammar Formalisms

- Surface structure



There was a special formalism for transformations.

History of Grammar Formalisms

- Robert Lees writes an influential positive review of Syntactic Structures.
- Many structuralists are impressed by the insightful treatment of English auxiliary verbs (affix hopping).
- Chomsky debates Piaget.
- Chomsky criticizes Skinner.
- Chomsky becomes one of the leading intellectuals of the 20th century.
  - Some people say *the* leading intellectual

Anecdotes

- Debating style of philosophers: How to prove a point P:
  - Davidson: As I have said before, P.
  - Chomsky: P. Because what else? Q?
    - Human language capabilities must be innate because, what else? Something that complicated could be learned so well by babies in such a short time with corrupt input?
- Chomsky's followers get a reputation for being arrogant and obnoxious.

History of Grammar Formalisms
Transformation Grammar in the 1960's and 1970's

- Chomsky's theories move from a set of rules to a set of principles which predict which rules are possible and which are not.
  - X-bar theory: Chomsky (1970) "Remarks on Nominalizations"; Jackendoff (1977) *X-bar Theory*.
    - NP → V PP (not likely)
    - NP → Det N (likely)
  - John Robert (Haj) Ross (1967) *Constraints on transformations (not sure of the title);* Chomsky (1977) "On Wh Movement"

History of Grammar Formalisms
Transformation Grammar in the 1960's and 1970's

- Some of Chomsky's followers spin off in different directions:
  - Generative Semantics
    - ***All*** aspects of meaning are represented in deep structure.
    - Speech acts are represented in deep structure. There has to be a transformation to remove "I say to you" at the beginning of every statement. (But "I promise you" usually does not get deleted.)
    - There is some debate about whether the word "bachelor" has to be derived from the deep structure "unmarried man."
  - Relational Grammar
    - (Re-)introduced subject and object into generative grammar.
    - Chomsky says that grammatical relations can't be primitives of the theory because the primitives have to be things that babies could perceive like suffixes, word order, and agentivity/cause.
- Anecdote:
  - People leave MIT. There are raging debates with the vehemence of the times (late 60's and early 70's). Careers are ruined.

No more about Chomsky

- His systems of principles moved away from formalization in a way that cannot be easily implemented.
- But there are some principle based parsers:
  - See papers by these people: Weinberg, Wehrli, Abney, Fong, Berwick

Montague Grammar
Syntax and Formal Semantics

- http://www-unix.oit.umass.edu/~partee/docs/MontagueGrammarElsevier.PDF
- "**Montague**'s idea that a natural language like English could be formally described using logicians' techniques was a radical one at the time. Most logicians believed that natural languages were not amenable to precise formalization, while most linguists doubted the appropriateness of logicians' approaches to the domain of natural language semantics."
- Precursor to Categorial Grammar

Richard Montague

1930-1971

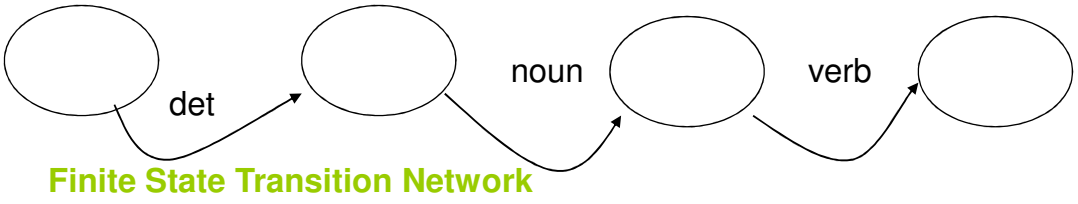
History of Grammar Formalisms

Some events in computational linguistics

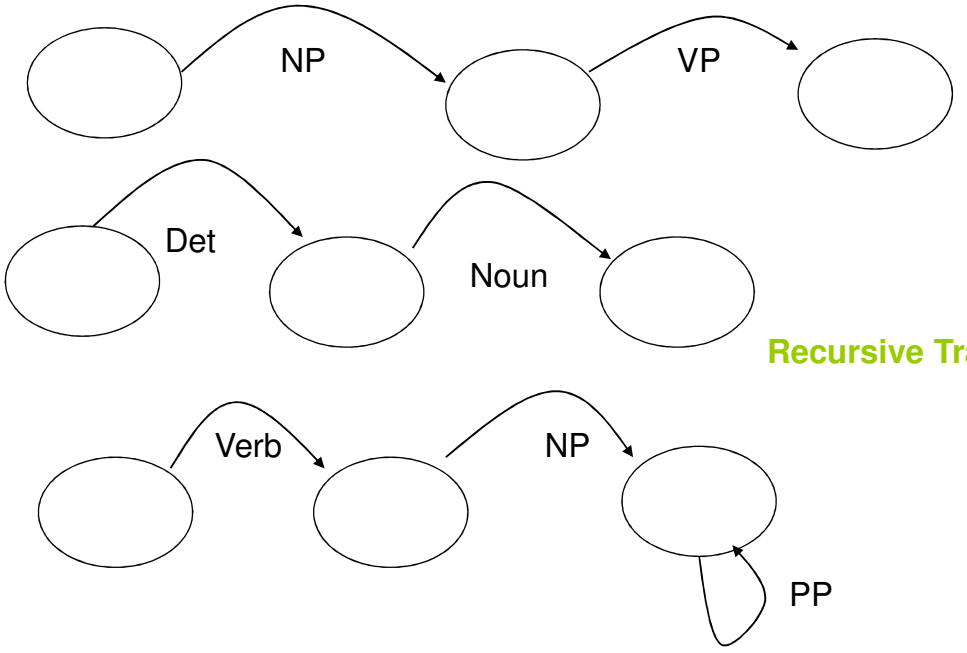
- William Woods (1970): “Transition Network Grammars for Natural Language Analysis” in Grosz, Sparck Jones, and Webber (eds.) *Studies in Natural Language Processing*, Morgan Kaufmann, 1986.

Bill Woods

Sun Microsystems



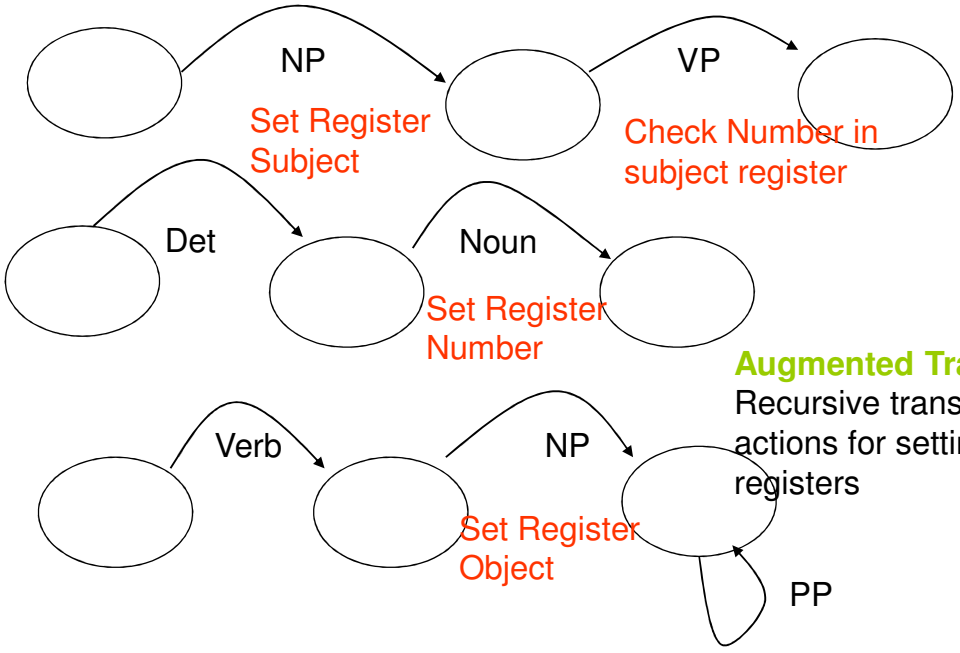
Finite State Transition Network



Recursive Transition Network

Subject: Number singular
Person 3
Root girl

Register Structure



Augmented Transition Network
Recursive transition network with actions for setting and checking registers

History of Grammar Formalisms

Some events in computational linguistics

- Martin Kay, “Parsing in Functional Unification Grammar”, in Grosz, Sparck Jones, and Webber (eds.)
  - Not sure what year Kay’s paper was originally published

Martin Kay
PARC (?)

Functional Structure
Martin Kay

SUBJ	PRED	‘bear’
	NUM	sg
	PERS	3
	DEF	+
VERB	EAT	
TENSE	past	
OBJ	PRED	‘sandwich’
	NUM	sg
	PERS	3
	DEF	-

Unification: Combining information in feature structures

Num sg
Pers 3
Gend f

Unified with

Num sg
Case nom

equals

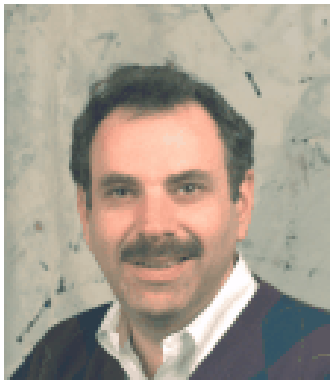
Num sg
Pers 3
Gend f
Case nom

- http://www.authorstream.com/Presentation/Jacqueline-27895-Lecture8Unification-Features-Unification-Solving-Agreement-Problem-Subcategorization-Verbs-Example-Specifying-as-Entertainment-ppt-powerpoint/

That leads to Grammar Formalisms as we know them today

Lexical Functional Grammar (around 1978)

Joan Bresnan
Stanford University



Ron Kaplan
PARC

Lexical Functional Grammar

- Bresnan (linguist) wanted to create a "realistic" transformational grammar.
  - Grammatical relations (subject and object)
  - No empty categories
  - Passivization as a lexical rule rather than a tree-to-tree mapping.
- Kaplan (computational psycholinguist) was working at Xerox PARC with Martin Kay.

Local co-description of *partial* structures

- S → NP VP
  (↑ SUBJ) = ↓ ↑= ↓

NP says: My mother's f-structure has a SUBJ feature whose value is my f-structure.

VP says: My mother's f-structure is my f-structure.

This rule simultaneously describes a piece of c-structure and a piece of f-structure.

It is local because each equation refers only to the current node and its mother. (page 119-120)

Levels of Representation

- Transformational Grammar
  - Deep structure: represents meaning
    - Sentences that mean the same thing have the same deep structure
  - Surface structure: represents word order and represents grammatical relations indirectly
- Lexical Functional Grammar
  - Constituent structure: represents word order and groupings of words into phrases.
  - Functional structure: represents grammatical relations explicitly.
  - Argument structure: represents meaning

Unification Based Formalisms

- Shieber and Pereira
  - Prolog and Natural Language Analysis, CSLI, 1987.
    - Prolog has built in unification and backtracking.
  - PATR formalism

S → NP VP

(X0 SUBJ) = X1

X0 = X2

Fernando Pereira
University of Pennsylvania

Stuart Shieber
Harvard

The Tomita Parser

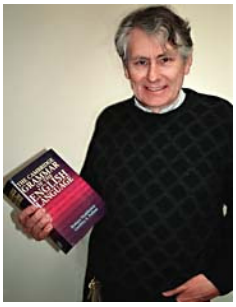
- Around 1985
- Based on PATR and LFG
- One of the first unification based parsers that was fast enough to use for real applications.
- Tomita was co-founder of the CMT, which later became the LTI.
- Several of you are using the Tomita parser or a descendent of it in your current research.

Masaru Tomita

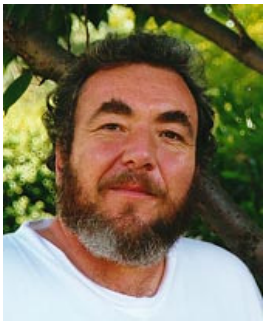
Keio University

Generalized Phrase Structure Grammar (GPSG)
Precursor to HPSG

Ewan Klein
Edinburgh



Geoffrey Pullum
UC Santa Cruz



Ivan Sag
Stanford

Gerald Gazdar

Sussex

Generalized Phrase Structure Grammar (GPSG)
Precursor to HPSG

The nature of syntactic categories: GPSG and HPSG

- Generalized Phrase Structure Grammar (GPSG, Gazdar, Pullum, Sag, and Klein):
  - Claimed that human language syntax could be handled with context free rules.
- Categories in GPSG:
  - Handling dependencies like subject-verb agreement using context free rules (hypothetical language)
    - S $\rightarrow$ NP-sg-fem VP-sg-fem
    - S $\rightarrow$ NP-pl-fem VP-sg-fem
    - S $\rightarrow$ NP-[num x, gen y] VP-[num x, gen y]
    - Node labels like NP and VP turned into feature structures.
  - Immediate dominance and linear precedence
    - Context phrase structure rules specify dominance and precedence
    - For languages with free word order, it makes sense to separate dominance from precedence.
      - S dominates a V, some NPs, and some PPs
    - Generalizations about precedence can hold over all the rules.
      - Japanese is head-final: V is final in S. N is final in NP. P is final in PP.

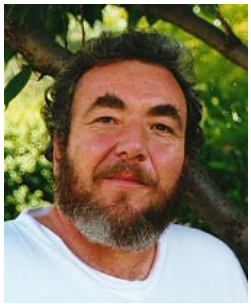
GPSG and HPSG

- Since context free rules aren't very pretty, GPSG introduced meta-rules and rule schemata to make the grammar more elegant.
- Then it was discovered that some of the meta rules weren't context free (Shieber 1985).

Head Driven Phrase Structure Grammar

- By that time, Carl Pollard had started working on HPSG using some ideas that followed naturally from GPSG:
  - Node labels were feature structures instead of atomic categories like NP and VP
  - Generalizing from GPSG rule schemata lead to the special treatment of phrasal heads
  - Unification was becoming popular with LFG and PATR.
- Pollard added:
  - Typed feature structures
  - HPSG is a system of principles that predict which rules occur.
- Carl Pollard did some of this work at CMU in the old Computational Linguistics Program in the Philosophy Department.

Carl Pollard
Ohio State Univ.



Ivan Sag

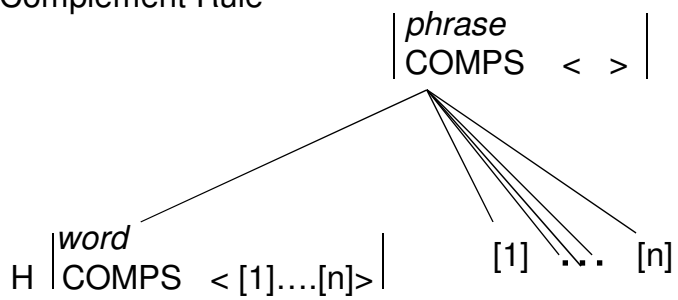
Stanford

HPSG: Things not handled well by context free grammars

- Headedness
  - Some rules are more likely than others
    - NP → V PP (not likely)
    - NP → Det N (likely)
  - The head determines the properties of the phrase.
    - The smart **students** in the class **studied** hard.
    - *Girls* determines that the noun phrase is plural.
    - **Studied** determines that the sentence is past tense.
- Subcategorization
  - Subcategories of verbs that occur in different contexts:
    - No direct object: Some problems exist.
    - One direct object: The children ate.
    - Direct and indirect objects: The teacher handed books to the students.
    - Etc.

HPSG Principles/Rule Schemata

Head Complement Rule



Summarizes many rules:
VP → V-intrans
VP → V-trans NP
AP → Adj VP (happy to be here)
NP → N PP (student of linguistics)

Categorial Grammar

- History
  - appears around 1980
  - Fits well with Montague Grammar
- Key features
  - Compositionality
    - The meaning of a phrase is a function of the meanings of its parts.
  - put more information in lexical items rather than rules
    - E.g., “devour” has to occur with an NP to the left and an NP to the right.
  - Does "non-constituent coordination" well.
    - John wrote and Bill signed the letter.
    - I gave a book to Mary and a CD to Sue.

Mark Steedman
Edinburgh

Categorial Grammar Example

Example Rules
  Forward application : X/Y Y => X
  Backward application: Y X\Y -> X
Lexical items
  John np
  Mary np
  loves (s\np)/np
Derivations

```
 John   likes    Mary
  np   (s\np)/np)  np
         ---------- Forward
          s\np
   ---------- Backward
       s
```

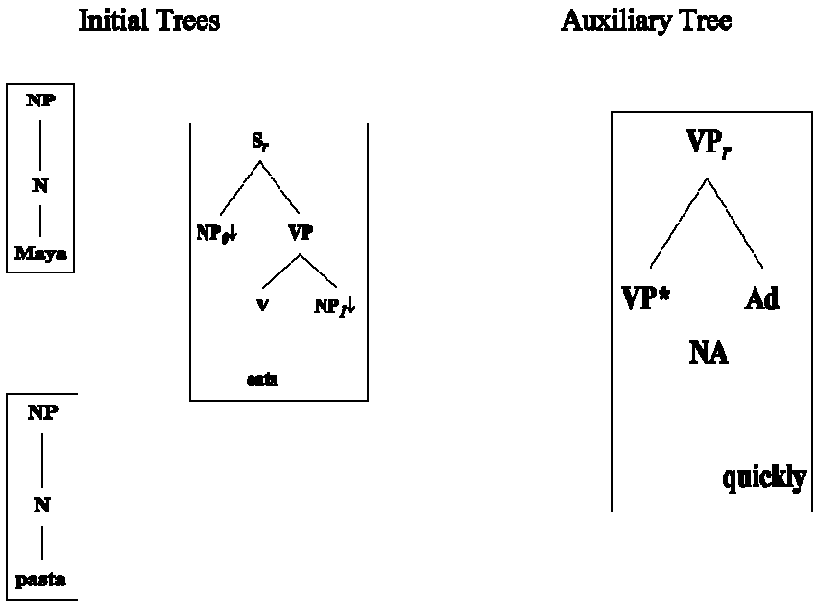
Tree Adjoining Grammar

- Describe a language of trees rather than strings (which trees are licensed by the grammar)
- Easier to encode linguistically well-motivated grammars and constraints:
  - Separation of recursion from linguistic constraints and dependencies (constraints can be specified within the scope of individual elementary trees)
  - Each elementary tree can be lexicalized
- Can describe “mildly-context-sensitive” languages.

Aravind Joshi

University of Pennsylvania

TAGs: Elementary Trees



Summary

- Grammar formalisms consist of production rules
  - The production rules are declarative – a description, not an procedure.
  - The production rules specify which strings are in the language and which aren't.
  - The production rules specify a structure.
  - There may be more than one kind of structure (Levels of representation), for example a phrase structure tree, a feature structure, and a logical formula.
- Feature structures and unification play a big role in most modern formalisms.
- Lexical Functional Grammar focuses on the role of grammatical relations in universal grammar, and how they are encoded in phrase structure.
  - Rules take the form of co-descriptions of pieces of constituent structures and feature structures.
- HPSG focuses on the nature of syntactic categories – typed feature structures.
  - Rules take the form of schemata that are actually summaries of many rules.
- Tree Adjoining Grammars consist of elementary trees that can be combined.
- Categorial grammars focus on compositionality of syntax and semantics.
- Lexicalization is an important concept.
  - Many grammar rules can be stated as requirements of lexical items.

Grammar formalisms and other topics not covered in this class
(Can be used for term projects)

- Link grammar (Sleator and Lafferty)
- Dependency Grammar
- Construction Grammar
- Formalisms for morphology
- Treebanks (English, Chinese, Czech)
- Generative capacity
- Applications – machine translation, IR, etc.
- Chomsky's current theories