Approaches to Computational Grammar of Natural Language

Roussanka Loukanova

Institute of Mathematics and Informatics (IMI)
Bulgarian Academy of Sciences (BAS), Sofia, Bulgaria

EuroProofNet School on Natural Formal Mathematics
June 3rd–5th 2025 in Bonn, Germany

1. Computational Efficiency and Linguistics Generalizations: CFGs

2. Some Ambiguities: Syntactic vs Semantic Ambiguities
   - Syntactic Ambiguities

3. A Glimpse of Approaches to Formal and Computational Grammar
   - Some Approaches to Formal or Computational Syntax
   - Overview of Approaches to Computational Semantics

CFG as a Candidate for Computational Grammar of Natural (Human) Language (NL)

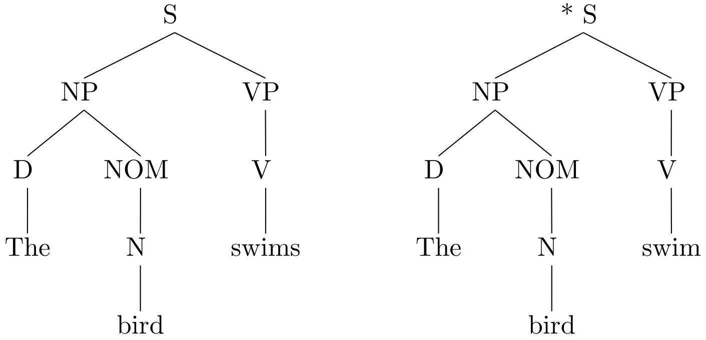
- CFGs of NL specify co-occurrence properties of expressions by using nonterminal symbols S, NP, VP, NOM, N, Det, V, Adj, ect.
  We can call them syntactic categories.
- N, Det, V, Adj, etc., are lexical categories, by rules for lexemes, e.g., for generating sets of words
- A given CFG defines possible combinations between the categories, by CF rules, e.g., like the following Gr1, a toy CFG:

Gr1

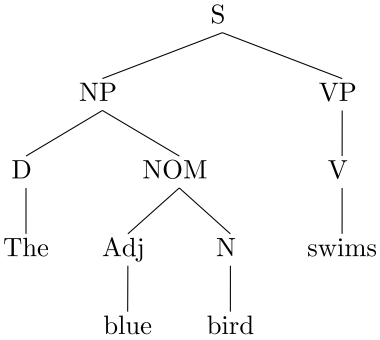
S	$\rightarrow$	NP VP	VP	$\rightarrow$	V
NP	$\rightarrow$	D NOM	VP	$\rightarrow$	V NP
NOM	$\rightarrow$	N	NOM	$\rightarrow$	Adj NOM
VP	$\rightarrow$	V NP NP	VP	$\rightarrow$	V NP PP
NOM	$\rightarrow$	NOM PP	PP	$\rightarrow$	P NP
N	$\rightarrow$	bird $\mid$ birds ...	Det	$\rightarrow$	every $\mid$ all $\mid$ some $\mid$ a $\mid$ the
Adj	$\rightarrow$	blue $\mid$ big ...	Det	$\rightarrow$	one $\mid$ two $\mid$ three $\mid$ most ...
V	$\rightarrow$	swims $\mid$ swim ...			

CFG as a Candidate for Computational Grammar of NL

- Overgenerated Tree Structure: *S
  by missing grammatical agreement between the subject NP and the predicate VP

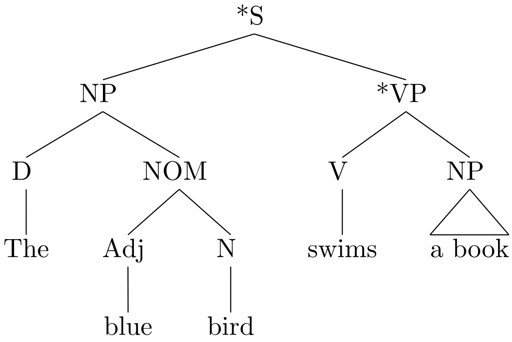


CFGs as Computational Grammar of Natural (Human) Language



CFGs as Computational Grammar of Natural (Human) Language

- Overgenerated Tree Structure: *S
  by violation of the complement requirements / constraints of the head word "swims" of the syntactic category V inside the predicate VP



CFG as a Candidate for Computational Grammar of Natural and Formal Languages

- The above small CFG does not pretend to be any complete CFG of NL
- I'm using it to point to fundamental problems related to using CFGs as a computational grammar theory of NL
- The above rules explicitly show that they are not sufficiently good to represent:
  - agreement between syntactic structures
  - complement requirements of the syntactic category V (of verbs), etc.
- CFGs and rewriting grammars are the major computational facility for parsing of
  - Natural (Human) Language (NL) of everyday life
  - Natural (Human) Language (NL) of mathematical texts, i.e., of Natural Formal Mathematics
  - Formal Languages (FL), including programming and logic languages

CFGs as Computational Grammar of NL: How to Repair Gr1

By considering pairs of proliferated grammatical categories:

Gr2

$$\begin{array}{lll} \mathrm{S} & \rightarrow & \mathrm{NP\text{-}SG\ VP\text{-}SG} \\ \mathrm{S} & \rightarrow & \mathrm{NP\text{-}PL\ VP\text{-}PL} \\ \mathrm{NP\text{-}SG} & \rightarrow & \mathrm{D\text{-}SG\ N\text{-}SG} \\ \mathrm{NP\text{-}PL} & \rightarrow & \mathrm{D\text{-}PL\ N\text{-}PL} \\ \mathrm{VP\text{-}SG} & \rightarrow & \mathrm{V\text{-}SG\ NP\text{-}SG} \end{array}$$

$$\begin{array}{lll} \mathrm{D\text{-}SG} & \rightarrow & \mathrm{the} \\ \mathrm{N\text{-}SG} & \rightarrow & \mathrm{bird} \\ \mathrm{V\text{-}SG} & \rightarrow & \mathrm{swims} \\ \mathrm{Adj} & \rightarrow & \mathrm{blue} \end{array} \qquad \begin{array}{lll} \mathrm{D\text{-}PL} & \rightarrow & \mathrm{the} \\ \mathrm{N\text{-}PL} & \rightarrow & \mathrm{birds} \\ \mathrm{V\text{-}PL} & \rightarrow & \mathrm{swim} \end{array}$$

Gr2 encodes linguistic info, implicitly via cooccurrences of nonterminals:

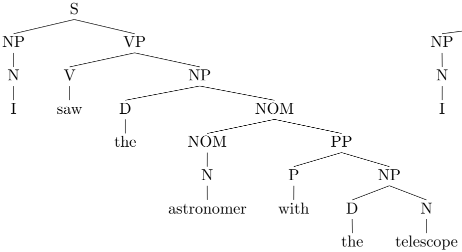
1. N-SG and N-PL are both of (type) part of speech *noun*
2. V-SG and V-PL are both part of speech *verb*
3. D-SG and D-PL are both part of speech *determiner*

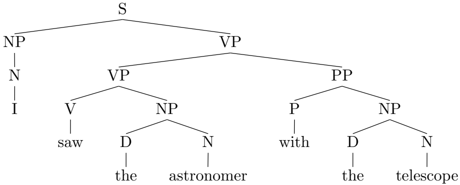
(Some) Objectives of Adequate Linguistic Theory

- Computational theory of linguistic information as theory of language phenomena
- The target is adequateness
  - Correctness: distinguishing syntactically well-formed from ill-formed expressions, e.g., in a given HL
  - Completeness: potential representation of all well-formed expressions that can be expressed in a given HL
  - Semantic coverage
  - Linguistically significant generalizations: Why? E.g., at minimum:
    1. to understand the nature of human languages, i.e., natural languages
    2. for adequate mathematical and computational theory of linguistic information
    3. for efficient language processing (LP) by computers, AI, and other computational systems
    4. in formal languages of programming and proof systems
    5. for effective language learning, via:
       - Learning of Rules of Computational Grammar
       - ML, LLM, etc.

Multiple PPs as Modifiers: SynSem Ambiguities

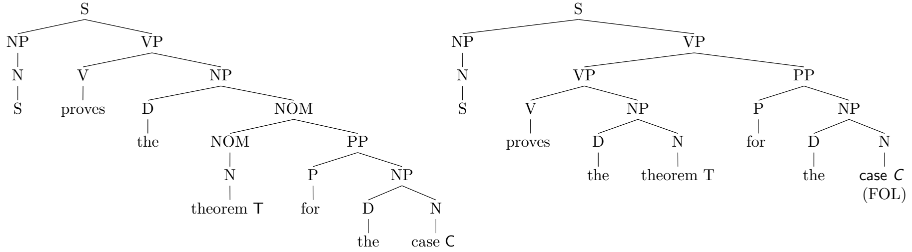
Different syntactic parses determine different denotations:



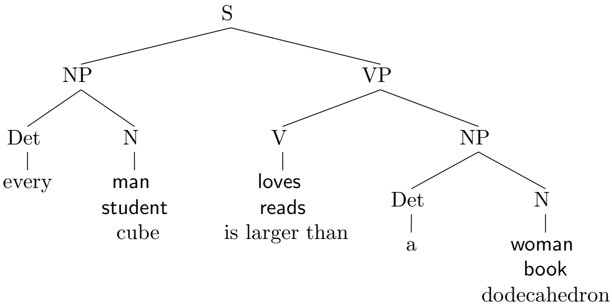


Multiple PPs as Modifiers: SynSem Ambiguities

SynSem ambiguities of NL propagate into mathematical texts:

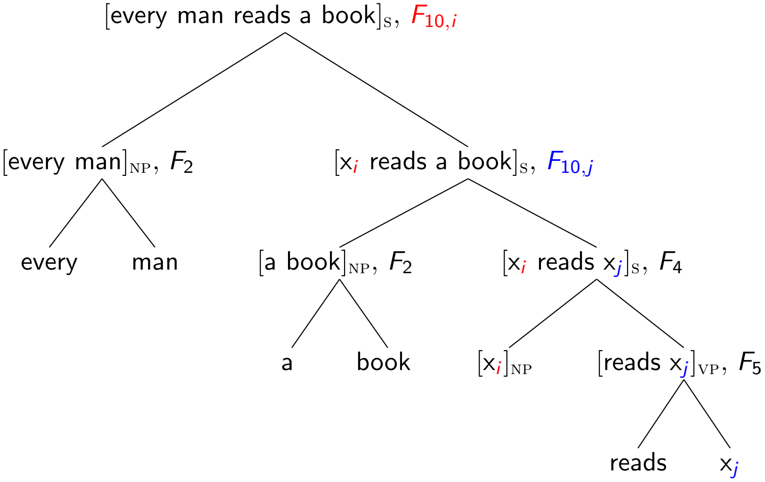


Multiple Quantifiers: Patterns of Scope Ambiguity, see LPL [1]; Syntactic or Semantic?

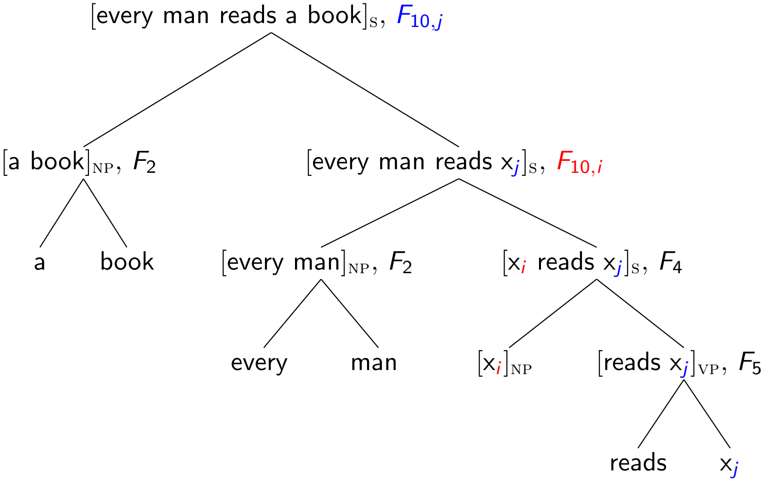


1. [[Every student]$_i$ reads [a book]$_j$]$_S$
2. [[every student]$_i$ [[a book]$_j$ [e$_i$ reads e$_j$]$_S$]$_S$]$_S$
3. $\forall x(\mathit{student}(x) \rightarrow \exists y(\mathit{book}(y) \wedge \mathit{read}(x, y)))$
4. [[a book]$_j$ [[every student]$_i$ [e$_i$ reads e$_j$]$_S$]$_S$]$_S$
5. $\exists y(\mathit{book}(y) \wedge \forall x(\mathit{student}(x) \rightarrow \mathit{read}(x, y)))$

Montague Syntactic Disambiguation: Non-Specific / De Dicto Reading

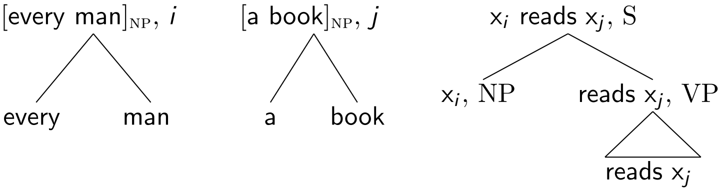


Montague Syntactic Disambiguation: Specific / De Re Reading



Scope Underspecification: Syntax?

What is the common between the trees representing the above two different scope readings?



- See Loukanova [5] for algorithmic syntax-semantics interface

Generalised Computational Grammar

Therefore, I target:

- Computational Grammar
  based on mathematical theory that can be applied to natural and formal languages
- The above objective requires meeting criteria of adequateness of a linguistic theory:
  - Chomsky Hierarchy of Formal Grammars coveres major language phenomena by syntax separated from semantics
  - Remedies: syntactic parsing that supports semantic structures, e.g., interpretations:
    - directly, by semantic models of NL parses,
      or
    - indirectly, via translations into formal languages of logic

Approaches to formal and computational syntax of natural language (NL)

All of the following approaches are at least partly active

CFGs, Phrase Structure Grammars (PHSG): initiated by Chomsky 1950s

Transformational Grammars: initiated by Chomsky 1955, 1957, with versions to the present

Generative Semantics: 1967-74 Lakoff, McCawley, Postal, Ross

Government and Binding Theory (GBT): initiated by Chomsky 1981

Principles and Parameters: initiated by Chomsky 1981 with GBT

Minimalist Program initiated by Chomsky 1995 (major work)

Constraint-Based, Lexicalist Approaches: CBLG

- *GPSG:* Gazdar et al. 1979-87 to the present
- *LFG:* 1979 to the present
- *HPSG:* 1984 to the present

Categorial Grammars: Ajdukiewicz 1935 to the present

Dependency Grammar (DG): active

Grammatical Framework (GF): Multi-Lingual, Chalmers, 1998,
Aarne Ranta, Krasimir Angelov (25 years on, in Mar 2023)
(open development)

...

Existing Approaches to Large-Scale Computational Grammar: GCBLG

- The most complete works on formalization of HPSG by mathematical logic:
  - Mark Johnson, 1988 [3]
  - Paul J. King, 1989 [4]
  - Bob Carpenter, 1992 [2]
  - Gerald B. Penn, 2000 [9]
  - Frank Richter, 2004 [12]
- HPSG provides semantic representations via Syntax-Semantics
- The formal syntax in CBLG, by Sag et al., 2003 [13], is based on models of typed functions, see:
  - Ch.9 [13]
  - Roussanka Loukanova [6]
- MultiLingual Grammatical Framework: Chalmers GF
  The most complete works on formalization of GF:
  - Aarne Ranta, 1994 [10, 11]

  Note: I would classify GF as a new direction in (a new kind of) GCBLG, under active development

- *Categorial Grammars:* Ajdukiewicz 1935 — formal logic for syntax for NL to the present, with initiations for syntax-semantics
- *Type-Theoretical Grammars* in many varieties
- Montague Grammars started by Montague 1970 to the present
- Situation Theory and Situation Semantics, Jon Barwise 1980s
  SitSem inspired partiality in computational syntax of LFG and HPSG;
  Since start, HPSG approaches, 1984, have been using Situation Semantics in syntax-semantics interfaces;
- *Minimal Recursion Semantics* (MRS) in HPSG since 2000-2002
  MRS is a technique combining (simple) Situation Semantics with major characteristics of Moschovakis recursion
- Moschovakis [7] Formal Language of full recursion, untyped;
  Typed Acyclic and Full Recursion, introduced by Moschovakis [8] (2006) and myself
- Algorithmic Dependent-Type Theory of Situated Information (DTTSitInfo): situated data including context assessments (open)
  Myself
- Other Approaches to Computational Semantics:
  many combinations and variants of HOL, FOL, e.g., Prolog, Definite Clause Grammars, etc.

Some References I

- Barker-Plummer, D., Barwise, J., Etchemendy, J.: Language, Proof and Logic, second edition.
  CSLI Publications, Stanford, California (September 2011).
  URL https://www.gradegrinder.net
- Carpenter, B.: The Logic of Typed Feature Structures, *Cambridge Tracts in Theoretical Computer Science*, vol. 32.
  Cambridge University Press (1992)
- Johnson, M.: Attribute-Value Logic and the Theory of Grammar.
  No. 16 in CSLI Lecture Notes. CSLI Publications, Stanford, California (1988)
- King, P.J.: A logical formalism for Head-Driven Phrase Structure Grammar.
  Ph.D. thesis, University of Manchester (1989)

Some References II

- Loukanova, R.: Relationships between Specified and Underspecified Quantification by the Theory of Acyclic Recursion.
  ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal **5**(4), 19–42 (2016).
  URL https://doi.org/10.14201/ADCAIJ2016541942
- Loukanova, R.: An approach to functional formal models of constraint-based lexicalized grammar (CBLG).
  Fundamenta Informaticae **152**(4), 341–372 (2017).
  URL https://doi.org/10.3233/FI-2017-1524
- Moschovakis, Y.N.: The formal language of recursion.
  Journal of Symbolic Logic **54**(4), 1216–1252 (1989).
  URL https://doi.org/10.1017/S0022481200041086
- Moschovakis, Y.N.: A Logical Calculus of Meaning and Synonymy.
  Linguistics and Philosophy **29**(1), 27–89 (2006).
  URL https://doi.org/10.1007/s10988-005-6920-7

Some References III

- Penn, G.B.: The algebraic structure of attributed type signatures.
  Ph.D. thesis, School of Computer Science, Carnegie Mellon University (2000)

- Ranta, A.: Type Theoretical Grammar.
  Oxford University Press (1994)

- Ranta, A.: Grammatical framework.
  Journal of Functional Programming **14**, 145–189 (2004).
  URL `https://doi.org/10.1017/S0956796803004738`

- Richter, F.: A mathematical formalism for linguistic theories with an application in Head-driven Phrase Structure Grammar.
  Ph.D. thesis, Eberhard Karls Universität Tübingen (2000)

- Sag, I.A., Wasow, T., Bender, E.M.: Syntactic Theory: A Formal Introduction.
  CSLI Publications, Stanford, California (2003)