Logic, languages and linguistics

Anders Søgaard
Center for Language Technology
Njalsgade 80
DK-2300 Copenhagen
`anders@cst.dk`

April 17, 2006

The talk includes:

- a brief survey of the literature on descriptive complexity results; this journey takes us from regular to context-sensitive languages,
- a brief survey of the complexity of (realistic) linguistic formalisms; this journey takes us from context-sensitive grammars to fragments of richer grammars,
- a discussion of generative and model-theoretic means to arrive at more adequate formalisms.

My point of departure is the somewhat Chomskyan methodological point that an adequate theory of natural languages lies at the intersection of the descriptively adequate theories (in terms of **extension**), those theories that are adequate in terms of **complexity** and those theories that are adequate in terms of **learnability**. I talk mostly about complexity issues here.

0.1 Extension

Chomsky hierarchy	**Anomaly**	**Assasin**
Regular	Mirror effects	Noam Chomsky
Context-free	Cross-serial dep.	Stuart Shieber (+ others)
Mildly context-sensitive	Suffixaufnahme	Marcus Kracht

Several arguments against context-freeness are found in the literature. Certain forms of reduplication have been argued to transcend context-freeness, and Chinese α-not-α questions are often seen as an argument against mild context-sensitivity. Ordinary scrambling challenge certain mildly context-sensitive grammars, but non-semilinearity, as in Suffixaufnahme, Chinese number names and certain Dutch coordinations, seems to be a stronger constraint.

(1) He is **both** a painter **and** a linguist.

(2) A white male (whom a white male)^n hired^n hired another white male.

(3) mer em Hans.DAT es huus.ACC hälfed.DAT aastriiche.ACC.

(4) govel-i.NOM igi.NOM sisxl-i.NOM saxl-isa-j.GEN.NOM m-is.GEN Sail-is-isa-j.GEN.GEN.NOM ('all (the) blood.NOM (of the) house.GEN (of) Saul.GEN)

Here's the Dutch example, for you guys:

(5) dat Jan Piet Marie Fred^k (horde leren^k uitnodigen)$^+$ en zag leren^k omhelzen.

In the face of these examples, and for reasons of compositionality, it seems a good idea to step up the complexity somewhat, the next level being *context-sensitivity*.

0.2 Why context-sensitivity is not an option

Context-sensitive grammars represent an intermediate step between context-free and totally unrestricted rewrite grammars. No restrictions are placed on the left-hand side of a production, but the length of the right-hand side is required to be at least that of the left. If a language is accepted by a linear-bounded automaton, it is context-sensitive. Complexity of context-sensitive grammars is PSPACE-complete. If context-sensitive grammars are restricted to be acyclic, they turn NP-complete. The decidability of context-sensitive grammars can easily be seen by inspection of the productions they license:

$$\phi_1 \mathrm{A} \phi_2 \rightarrow \phi_1 \omega \phi_2$$

Since ω cannot be empty, the length of the right-hand side of the entailment is always as long as the length of its left-hand side. Consequently, the derivation either terminates or enters a loop, i.e. a projection is cyclic. However, such loops can be ignored in a context-sensitive grammar, since they contribute no important information.

Savitch notes that context-sensitive languages are too permisive to be the natural ones, since they, in a real sense, have all the structural complexity of the recursively enumerable ones. This, and what was presented on the previous slide, supports Wasow's Observation: Natural languages cannot be identified in the Chomsky Hierarchy. Or, in other words, it is necessary to cross-cut the hierarchy to do so.

Other things in favor of Wasow's Observation include linearization and learnability. (Learnability is addressed later on.) Chomsky Hierarchy grammars use phrase structure rules to encode dominance and precedence simultaneously, but there is really no empirical evidence for phrase structure. Whatsoever. Transformations, type raising, functional uncertainty, etc. are ways to accommodate for the limitations inherent to Chomsky Hierarchy grammars, or generative grammars, more broadly. In a sense, all these operators are used to move things "out of order". I will refer to some non-generative proposals of putting things *in* order that are more adequate and more efficient than Chomsky Hierarchy grammars.

0.3 Our goals

Most of this talk, however, is devoted to singling out feasible fragments of (constraint-based) linguistic formalisms that are at least context-sensitive; in fact, some are Turing-complete. The results presented here, in regard to computational complexity, is, first and foremost:

- the identification of a set of constraint-based grammars that account for the phenomena listed in the above, but whose universal recognition problems remain solvable in NPTIME, i.e. these grammars are more expressive than, for instance, the NPTIME fragments defined by Patrick Blackburn and Edith Spaan [BS93] or by Marten Trautwein [Tra95] some 10 years ago.

In addition, the adequacy of linearization-based, non-generative proposals is illustrated by an account of the Danish subordinate clause, which for your convenience, will be compared to a similar account of Dutch subordinate clauses. Finally, the issue of learnability is addressed; is it likely to construct a tractable, learnable linearization-based theory of natural language syntax?

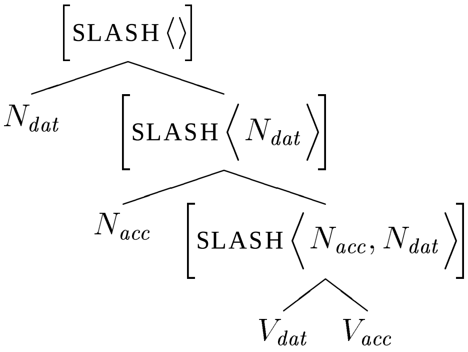
Example 0.1. Let A be the start symbol of

- $A \rightarrow a, A$
- $A \rightarrow a, b$

which then generates $\{a^+b\}$ and corresponds to

- $\begin{bmatrix}\textsc{type } \textit{non-terminal}\\ \textsc{cat } A\end{bmatrix} \rightarrow \begin{bmatrix}\textsc{type } \textit{terminal}\\ \textsc{phon } a\end{bmatrix}, \begin{bmatrix}\textsc{type } \textit{non-terminal}\\ \textsc{cat } A\end{bmatrix}$
- $\begin{bmatrix}\textsc{type } \textit{non-terminal}\\ \textsc{cat } A\end{bmatrix} \rightarrow \begin{bmatrix}\textsc{type } \textit{terminal}\\ \textsc{phon } a\end{bmatrix}, \begin{bmatrix}\textsc{type } \textit{terminal}\\ \textsc{phon } b\end{bmatrix}$

Unification-based CFGs extend CFGs, however. Reentrancies allow for generalizations (e.g. agreement), and information can be stored (as in indexed languages):


$\left[\text{SLASH}\ \langle\rangle\right]$
N_{dat}
$\left[\text{SLASH}\ \left\langle N_{dat} \right\rangle\right]$
N_{acc}
$\left[\text{SLASH}\ \left\langle N_{acc}, N_{dat} \right\rangle\right]$
V_{dat}
V_{acc}

0.4 Parsing algorithms

Parsing algorithms are constructed under two main paradigms; the generative one, and the model-theoretic one. The NPTIME-results obtained here are obtained under the latter paradigm, while the PTIME-fragments we discuss, refer to generative algorithms. The difference amounts to how languages are defined relative to grammars, i.e. how the extension of a grammar is defined:

$$\begin{array}{lcl}\mathcal{L}(\mathcal{G}) & = & \{\sigma \in \mathcal{V}^* | \exists M \in \mathcal{M}\ \texttt{start}' \sqsubseteq M \wedge M \overset{*}{\Rightarrow} \sigma\} \\ \mathcal{L}(\mathcal{G}) & = & \{\sigma \in \mathcal{V}^* | \exists M \in \mathcal{M}\ M, w \models (\mathsf{Axioms} \wedge \sigma')\}\end{array}$$

The generative definition is fully standard; the model-theoretic one is mine. In a constraint-based context, the generative algorithms that are employed, are often bottom-up (for good reasons). Model-theoretic algorithms are eclectic; usually the state-of-the-art algorithms for theorem proving or model generation in the particular logic in which the axioms are defined, are just adopted. The generative algorithms still employ phrase structure. Model-theoretic algorithms are agnostic on this matter; one can simulate phrase structures in the axioms, or one can change horses completely.

The advantages of model-theoretic algorithms include, apart from this flexibility wrt. grammar design: a more flexible notion of grammaticality, an open lexicon, and an agnostic stand on the size or cardinality of natural languages.

Example 0.2.

$$\begin{bmatrix}\text{TYPE} & \textit{non-terminal}\end{bmatrix} \Rightarrow \begin{bmatrix}\text{CAT} & \mathrm{A}\ \boxed{1} \\ \text{LEFT} & \begin{bmatrix}\text{TYPE} & \textit{terminal} \\ \text{PHON} & \mathrm{a}\end{bmatrix} \\ \text{RIGHT} & \begin{bmatrix}\text{TYPE} & \textit{non-terminal} \\ \text{CAT} & \boxed{1}\end{bmatrix}\end{bmatrix} \vee$$

$$\begin{bmatrix}\text{CAT} & \mathrm{A} \\ \text{LEFT} & \begin{bmatrix}\text{TYPE} & \textit{terminal} \\ \text{PHON} & \mathrm{a}\end{bmatrix} \\ \text{RIGHT} & \begin{bmatrix}\text{TYPE} & \textit{terminal} \\ \text{PHON} & \mathrm{b}\end{bmatrix}\end{bmatrix}$$

0.5 Complexity of universal recognition

LFG	Undecidable	[Joh88]
CUG	NP-complete	[Tra95]
CFG	$\mathcal{O}(n^3)$	[Ear70]

CUG has some obvious disadvantages. It does not encode free word order, scrambling and discontinuous constituency in any obvious way; it allows no unary extensions, and it cannot quantify over attribute paths, i.e. it does not have anything like LFG-style functional uncertainty.

CFG does not capture cross-serial dependencies and other linguistic phenomena. The number of rules tend to explode in applications; for instance, CFGs do not let the linguist generalize over agreement features.

[Joh88] defines an NP-complete fragment of LFG that does not allow detours, i.e. cyclic unary projections, and which does not incorporate functional uncertainty. This fragment is almost equivalent to the NP-complete HPSG fragment in [Tra95] (in fact, it is a bit weaker, since it does not allow for inheritance). The two fragments have some obvious short-comings, discussed in the next slides. One aim of my work is to present a richer fragment of exactly the same complexity.

0.6 Categorial Unification Grammar

We present a new proof of NP-completeness for CUG, different from the one obtained in [Tra95]. CUG is to A/B grammars what PATR is to CFG. You may recall that an A/B grammar consists of a set of two combinatory rules:

$$\begin{array}{lcll} X\,(X/Y) & \rightarrow & Y & \text{(left application)} \\ (Y/X)\,X & \rightarrow & Y & \text{(right application)} \end{array}$$

The simple move into CUG is to replace the syntactic categories with attribute-value matrices (feature structures) and turn each production step into a two-step procedure of *instantiation* and *stripping*, where instantiation is just unification, and stripping amounts to a kind of resource-sensitive follow-up, i.e. it removes the saturated Xes (in the above) from the syntactic category of the mother of the functor-argument structure.

Since nothing in the derivation procedure adds information (features), and since all rules are binary, the smallest model (the shortest derivation) of a string σ is at most $(2|\sigma|-1) \times \mathbf{paths}$. This is in fact enough to establish an NPTIME result. The general lemma that I'll employ a couple of times in this talk can be formulated as [BdRV01]:

Lemma 0.3. *If Λ (a consistent, normal modal logic) has the polysize model property (pmp), i.e. the smallest model of a formula is at most polynomial in its size, and if model-checking can be solved in PTIME, then satisfiability of Λ is in NPTIME.*

In fact, any consistent, normal modal logic that has the pmp and PTIME model-checking is NP-*complete*, since any such logic by definition encodes propositional satisfiability. If a theory, say of natural language syntax, can be formulated in such a language, i.e. the problem can be reduced to modal satisfiability, it is in NPTIME, but not necessarily NP-complete.

CUG can be encoded in a fragment of Kasper-Rounds logic with implication, where attributes are unary modalities, values are propositions, and derivations are Kripke structures. Since Kasper-Rounds logic is a consistent, normal modal logic with a PTIME model-checking problem [Lan06], and since CUG ensures the polysize model property (see above), it follows that

Theorem 0.4. *The universal recognition problem of CUG can be solved on a polynomial time bound non-deterministic Turing machine.*

0.7 Generalization of our proof

Note that our proof relies only on the size of the smallest model that satisfies a string. It can easily be derived from this that similar results can be obtained for linearization-based grammars. Why? Consider first non-linearization rules as they are represented in modal logic. The $\langle\ll\rangle$-modality encodes immediate precedence.

$$xy\_phrase \quad \rightarrow \quad \langle\texttt{down}\rangle(x \wedge \langle\texttt{right}\rangle y \wedge \langle\ll\rangle y)$$

This rule encorces the x- and y-constituents to immediately precede each other. If you remove the $\langle\ll\rangle$-constraint, this can no longer be ensured, and the constituents can intermingle with others. However, no additional structure can be introduced, if the derivation (Kripke structure) is connected. Consequently, the proof of solvability in NPTIME still applies.

The NPTIME proof does not apply when unary extensions are allowed, since cyclic unary extensions can make the derivations arbitrarily large. If unary extensions are restricted to be *acyclic*, however, the smallest model of a satisfiable string is at most:

$$(2|\sigma| - 1) \times (u + 1) \times \mathbf{paths}$$

where u is the number of unary rules. Consequently, the NPTIME proof still applies.

0.8 Adequacy of linearization-based proposals

The Dutch subordinate clause exibits a bit of free word order:

(6) (a) dat hij [alleen zijn vader]$_1$ [zulke dingen]$_2$ durft te vertellen
(b) dat hij [zulke dingen]$_2$ [alleen zijn vader]$_1$ durft te vertellen

Danish, where the only restrictions are that complementizers are initial and sentential adverbs precede finite verbs and are preceded by subjects, is – apparently? – a bit more radical in this respect:

(7)

. . . at	*han*	*kun*	*tør*	*at*	*sige*	*sådanne*	*ting*	*til*	*sin*	*far*
that.COMPL	he	only	dares	to	say	such	things	to	his	father

’that he only dares to tell his father such things’

(8) (a) at til sin far sådanne ting han kun tør at sige
(b) at sådanne ting til sin far han kun tør at sige
(c) at han kun sådanne ting at sige tør til sin far
(d) . . .

Compare also Dutch and Zürich German:

(9) dat Piet Jan Marie zag helpen zwemmen (Dutch)

(10) das er sini Chind laat Mediziin schtudiere (Zürich German)

(11) das er sini Chind Mediziin laat schtudiere

(12) das er sini Chind Mediziin schtudiere laat

(13) das er Mediziin sini Chind laat schtudiere

(14) das er Mediziin sini Chind schtudiere laat

(15) das er Mediziin schtudiere sini Chind laat

The generalization: (NP) Complements precede their governing verbs. It should be obvious why this data favors a linearization-based account.

0.9 Learnability

Another motivation for Wasow's Observation is learnability. In the late sixties, already, Gold [Gol67] proved that not even the regular languages are identifiable in the limit from positive data. Several things can (reasonably) be hypothesized on this background: (i) The Innateness Hypothesis is true, (ii) L1 acquisition is guided by negative data (in a way that psychologists have not yet noticed), or (iii) natural languages cross-cut the Chomsky Hierarchy. (ii) is widely conjectured to be false. Universal grammar (UG) can be implemented in many ways, e.g. as operations on a type hierarchy [Vil02] or as feature cooccurence restrictions.

UG-	
k-reversible FSA	[Ang82]
rigid A / B	[Kan98]

UG+	
CUG	[Vil02]

References

[Ang82] D. Angluin. Inference of reversible languages. *Journal of the Association for Computing Machinery*, 29:741–765, 1982.

[BdRV01] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal logic*. Cambridge University Press, Cambridge, England, 2001.

[BS93] Patrick Blackburn and Edith Spaan. A modal perspective on the computational complexity of attribute value grammar. *Journal of Logic, Language and Information*, 2(2):129–169, 1993.

[Ear70] Jay Earley. An efficient context-free parsing algorithm. *Communications of the Association for Computing Machinery*, 26(6):400–8, 1970.

[Gol67] Mark Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.

[Joh88] Mark Johnson. *Attribute-value logic and the theory of grammar*. CSLI Publications, Stanford, California, 1988.

[Kan98] Makoto Kanazawa. *Learnable classes of categorial grammars*. CSLI Publications, Stanford, California, 1998.

[Lan06] Martin Lange. Model checking propositional dynamic logic with all extras. *Journal of Applied Logic*, 4:39–49, 2006.

[Tra95] Marten Trautwein. *Computational pitfalls in tractable grammar formalisms*. PhD thesis, University of Amsterdam, Amsterdam, the Netherlands, 1995. IILC Dissertation Series DS-1995-15.

[Vil02] Aline Villavicencio. *The acquisition of a unification-based generalised categorial grammar*. PhD thesis, University of Cambridge, Cambridge, England, 2002. Technical report UCAM-CL-TR-533.