Unification-based grammars and complexity classes

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

0.1 The landscape

The complexity classes:

$$\text{P} \sqsubseteq \text{NP} \sqsubseteq \text{PSPACE} \sqsubseteq \text{EXPTIME}$$
$$\text{P} \neq \text{EXPTIME}$$

The landscape:

Author(s)	**Formalism**	**Result**	**Comments**
[KR86]		NP-complete	Same as [BS93].
[BBR87]	AG	NP-complete	
[Joh88]	LFG	NP-complete	Off-line parsable.
[Joh91]	LFG	NP-complete	Not stand-alone.
[BS93]		NP-complete	All features deterministic, and no global quantification.
[Tra95]	HPSG/LFG	NP-complete	Various restrictions.
[Tra95]	CUG	NP-complete	
[BS93]		EXPTIME	No path equations.
[Joh88]	LFG	UNDEC	Proof-theoretic perspective.
[BS93]		UNDEC	
[KM03]	HPSG	UNDEC	Indirect result.

Question: Why does the landscape look this way? Why care? *Gains:* A survey (for complexity *and* non-inclusion), some critical points, and some guidelines.

0.2 The problem

Definition 0.1 (Grammar extensions)**.** The extension of a grammar $\mathcal{G}$ is denoted by $\mathcal{L}(\mathcal{G})$. Consider the definition of $\mathcal{L}(\mathcal{G})_p$ and $\mathcal{L}(\mathcal{G})_m$.

$$\begin{aligned} \mathcal{L}(\mathcal{G})_p &= \{x \in \mathcal{V}^* | \exists M \in \mathcal{M}\ \mathtt{start}' \sqsubseteq M \wedge M \Rightarrow x\} \\ \mathcal{L}(\mathcal{G})_m &= \{x \in \mathcal{V}^* | \exists M \in \mathcal{M}\ M, w \models (\mathsf{Axioms} \wedge x')\} \end{aligned}$$

Important differences in methods. Unification-based grammars were conceived of as model-theoretic, and logical methods are used to obtain metaresults. Examples include [KR86, Joh91, BS93, KM03]. Complexities range from NP-complete to (highly) undecidable. The obvious reason is that the model-theoretic paradigm encodes parsing as a search problem. The universal recognition problem is:

INSTANCE:	Some logic L with axioms A, and a string ω.
QUESTION:	$\exists M.M \models_{\mathsf{L}} \mathsf{A} \wedge \omega'$?

0.3 Using logic

$$\begin{bmatrix} \text{FUNCT} \begin{bmatrix} \text{NUM } \boxed{1} \end{bmatrix} \\ \text{ARG} \begin{bmatrix} \text{NUM } \boxed{1} \end{bmatrix} \end{bmatrix}$$

$$\models \langle \text{FUNCT}; \text{NUM} \cap \text{ARG}; \text{NUM} \rangle \top$$

or

$$\langle \text{FUNCT}; \text{NUM} \rangle i \wedge \langle \text{ARG}; \text{NUM} \rangle i$$

or

$$\exists x, y, z, v. R_{\text{FUNCT}}(x, y) \wedge R_{\text{NUM}}(y, z) \wedge R_{\text{ARG}}(x, v) \wedge R_{\text{NUM}}(v, z)$$

or, if bounded

$$\bigwedge_{p \in \mathbb{PROP}} \langle \text{FUNCT}; \text{NUM} \rangle p \rightarrow \langle \text{ARG}; \text{NUM} \rangle p$$

0.4 Complexity of logics

Multimodal $\mathbf{K}.\mathbf{D_c}$	NP-complete	AG
$\mathbf{K}.\mathbf{D_c}^{\Box}$	NP-complete	(HPSG [BS93])
Multimodal $\mathbf{K}$	PSPACE-complete	?
$\mathbf{K}^{\Box}$	EXPTIME-complete	?
PDL	EXPTIME-complete	CFG,HPSG [Kra95]
PDL^{-1}	EXPTIME-complete	CFG
IPDL ($\mathbf{D_c}$)	UNDEC	UCG,AVG
$\mathsf{H}(\forall)$	UNDEC	AVG
First order logic	UNDEC	AVG
IPDL^{*}	UNDEC	HPSG [LS06]
$\mathsf{H}(\forall)^{*}$	UNDEC	HPSG [Rea94]

For a survey of fragments of first order logic, see [Søg06].

0.5 Why all this?

Gains:

- formalization,
- expressivity, and
- verification.

In addition, the expressive gains that are associated with a move up from one complexity class to another are wellstudied, for instance:

Definition 0.2 (Determinism). Determinism is ensured by axiom D_c:

D_c $\Diamond\phi \rightarrow \Box\phi$

In the context of multimodal logic, determinism leads to a small model theorem: If $M \models \phi$ then $\exists M'.|M'| \leq |\phi| + 1$ and $M' \models \phi$. The effect is only positive if we have the tree model property, however.

Definition 0.3 (Tree model property). If $M \models \phi$, then $\exists M'.M' \models \phi$ and M' is a (undirected) tree.

The loss of the tree model property is associated with the move to undecidability in the context of deterministic relations (non-deterministic IPDL is decidable), unless the logic is restricted somehow (simple H is PSPACE-complete).

To be continued ...

Definition 0.4 (Compactness)**.** If M satisfies every subformula of ϕ, it also satisfies ϕ.

The loss of compactness is indirectly related to PDL's move to EXPTIME. In the deterministic case, the polysize model property is lost due to non-deterministic operators.

Definition 0.5 (Generated submodels)**.** If $M \models \phi$ then if M' is the generated submodel in $w \in \mathbb{W}_M$, then $M' \models \phi$.

This accounts for the difference in complexity between $\mathbf{K}$ and $\mathbf{K}^{\Box}$.

0.6 Polysize model property (NP)

Definition 0.6 (Polysize model property)**.** If $M \models \phi$, then $\exists M'.M' \models \phi$ and $|M'| \leq |\phi^k|$.

Theorem 0.7. *If a logic has the polysize model property and tractable model checking, satisfiability is in NP.*

Example 0.8. The theorem applies to AG, CUG and off-line parsable LFG, for instance.

Theorem 0.9. *If a logic has the polysize model property and PSPACE model checking, satisfiability is in PSPACE.*

Example 0.10. The theorem applies to HPSG defined on polynomial models [LS06].

Gain: Since the proofs only rely on model size and the complexity of verification, certain things come for free, for instance the dissociation of dominance and linear precedence.

0.7 Alternating Turing machines (PSPACE)

An ATM is a Turing Machine in which the state set is partitioned into existential and universal states. Transitions from existential states correspond to non-deterministic choice. Hence, the ATM accepts an input word from an existential state iff there is a successor configuration from which it accepts. Universal choice dualises this, and the ATM accepts an input from a universal configuration iff all possible successor configurations accept.

Theorem 0.11. *If something is solvable in alternating polynomial time, then there is also a deterministic PSPACE algorithm for it.*

0.8 Finite model property (DEC)

Definition 0.12 (Finite model property)**.** If $M \models \phi$, then $\exists M'.M' \models \phi$ and M' is finite.

If a logic has the finite model property and an axiom system, it is possible to construct a Turing machine that uses the axioms to enumerate the validities of the logic. Second, construct a Turing machine that recursively enumerates all the finite models. The two machines can effectively test the validity of any formula ϕ: If ϕ is valid it will eventually be generated by the first machine; if it is not, it is falsified by a model generated by the second machine.

Definition 0.13 (Strong finite model property)**.** If $M \models \phi$, then $\exists M'.M' \models \phi$ and an upper bound on $|M'|$ is computable by f.

The Turing machine now takes ϕ as input. It simply generates all the models smaller than $f(|\phi|)$ and tests ϕ on each of them.

0.9 Ambiguity reduction (P)

The tractable unification-based fragments known in the literature either reduce trivially to context-free grammar (or sometimes mildly context-sensitive formalisms), or they restrict ambiguity in one way or another. Two examples:

Example 0.14. The PTIME result in [SNKK93] is obtained under a proof-theoretic perspective too by syntactic restrictions on the productions of LFG. The productions are first restricted to be either of the form $(\uparrow \texttt{attr}) = \texttt{val}$ (an immediate value schema) or $(\uparrow \texttt{attr}) = \downarrow$ (a structure synthesizing schema). Each pair of rules $r_1 : A \rightarrow \alpha_1$ and $r_2 : A \rightarrow \alpha_2$ whose left-hand sides are the same, is said to be inconsistent in the sense that there exists no f-structure that locally satisfies both of the functional schemata of r_1 and r_2.

Example 0.15. It is also possible to single out a tractable fragment of HPSG by ensuring that

- recursive feature geometry is polynomially bound in the length of the string, and
- ambiguity is bound by some constant, i.e at some point in the derivation, signs begin to combine unambiguously.

0.10 Guidelines

- Lower bound proofs highlight (possibly unnecessary) complexities.
- Complexity increases are often associated with specific properties.
- Restricting model size is an efficient and linguistically sound way of reducing complexity.
- Ambiguity reduction seems necessary to obtain tractability.
- Non-inclusion: HPSG $\not\subseteq$ [BS93] $\not\subseteq$ CUG, for instance.
- Combining the insights of lower bound proofs and invariance results with non-inclusion results, it is relatively easy to provide partial (optimal) translations from one grammar formalism to another.

References

[BBR87] Edward Barton, Robert Berwick, and Erik Ristad. *Computational complexity and natural language*. MIT Press, Cambridge, Massachusetts, 1987.

[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.

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

[Joh91] Mark Johnson. Features and formulae. *Computational LInguistics*, 17:131–152, 1991.

[KM03] Stephan Kepser and Uwe Mönnich. Undecidability results for HPSG. In Anton Nijholt and Giuseppe Scollo, editors, *Algebraic Methods in Language Processing*, pages 1–12, 2003.

[KR86] Robert Kasper and William Rounds. A logical semantics for feature structures. In *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics*, pages 257–265, New York, NY, 1986.

[Kra95] Marcus Kracht. Is there a genuine modal perspective on feature structures? *Linguistics & Philosophy*, 18:401–458, 1995.

[LS06] Martin Lange and Anders Søgaard. Polyadic dynamic logics for deep parsing. *X*, X:X–X, 2006. Under preparation.

[Rea94] Mike Reape. A feature value logic with intensionality, nonwellfoundedness and functional and relational dependencies. In C. J. Rupp, M. Rosner, and R. Johnson, editors, *Constraints, language and computation*, pages 77–110. Academic Press, San Fransisco, California, 1994.

[SNKK93] Hiroyuki Seki, Ryuichi Nakanishi, Yuichi Kaji, and Sachiko Andoand Tadao Kasami. Parallel multiple context-free grammars, finite-state translation systems, and polynomial-time recognizable subclasses of lexical-functional grammars. In *Proceedings of the 31st Annual Meeting on Association for Computational Linguistics*, pages 130–139, Columbus, Ohio, 1993.

[Søg06] Anders Søgaard. *HPSG and model generation*, volume 7 of *Center for Language Technology Working Papers*. University of Copenhagen Press, Copenhagen, Denmark, 2006. In press.

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