

# Complexity of reentrancy, unordering and synchronism

Anders Søgaard

Center for Language Technology  
University of Copenhagen  
Njalsgade 80  
DK-2300 Copenhagen S  
Email: anders@cst.dk

July 18 2007

### 3 NP-hardness proofs

- Most NP-hardness proofs of recognition problems relate to reentrancy, unordering or synchronism. The canonical proofs go by, respectively, 3-SAT [BBR87], vertex cover [Bar85] and 3-SAT [SP05].
- Minimal instances:
  - ▶ s-AVG (see below; CFG with partial functions from attributes to values instead of nonterminals)
  - ▶ UCFG ( $\phi_1 A \phi_2 \implies_U \phi_1 \omega \phi_2$  iff  $A \rightarrow \omega'$  and  $\omega' \in \text{permute}(\omega)$ )
  - ▶ SCFG (e.g.  $A \rightarrow \langle B_1 C_2, C_2 B_1 \rangle$ )
- I will present proof sketches and identify more or less reasonable polynomial fragments. Some already known; others obtained by translation into RCG or, orthogonally, by  $k$ -ambiguity constraints (relative to an inheritance hierarchy).

# Simple attribute-value grammars

s-AVGs are defined over simple attribute-value structures (s-AVSs):

## Definition (s-AVS)

An s-AVS  $A$  is defined over a finite signature  $\langle \text{Attr}, \text{Atms}, \rho \rangle$  as a partial function from  $\text{Attr}$  to  $\text{Atms}$ , where  $\rho : \text{Attr} \rightarrow 2^{\text{Atms}}$ , and  $\forall a \in \text{DOM}(A). A(a) \in \rho(a)$ .

## Definition (s-AVG)

A s-AVG is a 5-tuple  $G = \langle \langle \text{Attr}, \text{Atms}, \rho \rangle, \text{AgrAttr}, T, P, S \rangle$  (all sets finite), where  $\text{AgrAttr} \subseteq \text{Attr}$ ,  $\rho : \text{Attr} \rightarrow 2^{\text{Atms}}$ ,  $S$  is an s-AVS, and every production rule in  $P$  is of the form  $A \rightarrow \omega_i$  or  $A_0 \rightarrow A_1 \dots A_n$  where  $n \geq 1$ ,  $A_i$  is an s-AVS, and

$$\forall a \in \text{DOM}(A_0) \cap \text{AgrAttr}. \forall 1 \leq i \leq n. f \in \text{DOM}(A_i) \wedge A_i(a) = A_0(a)$$

where  $A(a)$  is the value of  $a$  in the s-AVS  $A$  with  $A(a) \in \rho(a)$ .

## Theorem

*s*-AVG context-free recognition is NP-complete.

## Proof.

Lower bound by reduction of 3SAT, connectives removed. Introduce an agreement attribute  $p_i$  for each propositional variable in the 3SAT instance such that  $\rho(p_i) = \{0, 1\}$ .

$\text{Attr} = \{p_1, \dots, p_n, \text{STAGE}, \text{LITERAL}\}$ ,  $\text{AgrAttr} = \{p_1, \dots, p_n\}$ .  $\rho(\text{STAGE}) = \{1, \dots, n + 3\}$ ,  $\text{LITERAL}$  Boolean. The first  $n$  stages assign truth values by productions

$$\left[ \text{STAGE } i \right] \rightarrow \left[ \begin{array}{l} \text{STAGE } i + 1 \\ p_i \ 0 \end{array} \right]$$

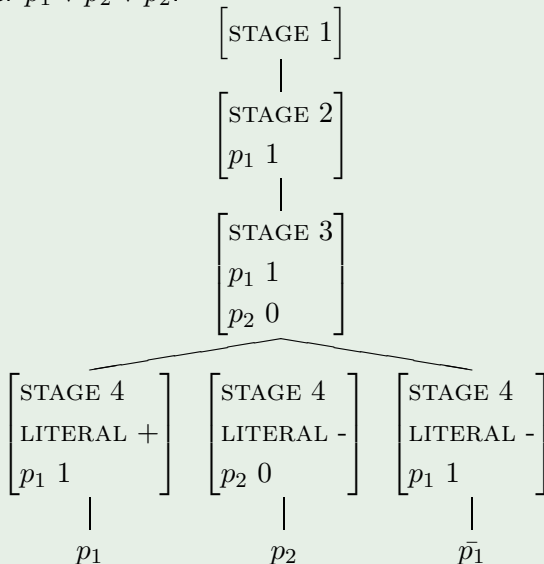
From stage  $n + 1$  to  $n + 2$  tree structure is build for the (3-literal) clauses.  $\text{LITERAL} +$  translates truth value 1. Say  $A_0$  is the *s*-AVS with  $\text{STAGE}$  value  $n + 3$  and  $\text{LITERAL} -$ , and  $A_1$  the same except  $\text{LITERAL} +$ . Now cover all (7) combinations except all false. Finally add productions of the form

$$\left[ \begin{array}{l} \text{STAGE } n + 3 \\ \text{LITERAL } - \\ p_i \ 1 \end{array} \right] \rightarrow \bar{p}_i$$

Percolation of agreement values will ensure that  $p_i$  values are consistent, i.e. that 3SAT assignment is consistent. The upper bound is easily proven. □

## Example

For  $p_1 \vee p_2 \vee \bar{p}_2$ :



## Theorem

*Unordered context-free recognition is NP-complete.*

## Proof (1/4).

The decision problem

INSTANCE: A graph  $G$  and a positive integer  $k$ .

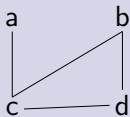
QUESTION: Is there a vertex cover of size  $k$  or less for  $G$ ?

is NP-complete [GJ79]. ...



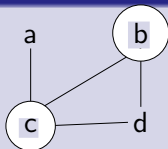
Proof (2/4).

Example (Example (1/2))



Proof (2/4).

Example (Example (1/2))



## Example (Example (2/2))

Say  $k = 2$ ,  $V = \{a, b, c, d\}$ ,  $E = \{(a, c), (b, c), (b, d), (c, d)\}$ . One way to obtain a vertex cover is to go through the edges and underline one endpoint of each edge. If you can do that and only underline two vertex symbols, a vertex cover has been found. Since  $|V| = 4$ , this is equivalent to leaving two vertex symbols untouched. Consequently, the vertex cover problem for this specific instance is encoded by the totally unordered type 2 grammar, where  $\delta$  is a bookkeeping dummy symbol:

$$\begin{array}{ll}
 S & \rightarrow \rho_1 \rho_2 \rho_3 \rho_4 u u \delta \delta \delta \delta & \rho_1 & \rightarrow a|c \\
 \rho_2 & \rightarrow b|c & \rho_3 & \rightarrow b|d \\
 \rho_4 & \rightarrow c|d & \delta & \rightarrow a|b|c|d \\
 & & u & \rightarrow aaaa|bbbb|cccc|dddd
 \end{array}$$

$\rho_i$  captures the  $i$ th edge in  $E$ . The input string  $\omega = aaaa|bbbb|cccc|dddd$ . One derivation tree in our example will have the form:

$$[[aaaa]_U [bbbb]_U [c]_{(b,c)} [c]_{(a,c)} [c]_{(c,d)} [c]_\delta [d]_{(b,d)} [d]_\delta [d]_\delta [d]_\delta]_S$$

## Proof (4/4).

Generally, the first production has as many  $\rho_i$  as there are edges in the graph,  $|V| - k$  many  $U$ 's and  $|E| \times |V| - |E| - |E| \times (|V| - k)$  many  $\delta$ 's, i.e. the length of the string minus the number of edges and the extension of  $|V| - k$  many  $U$ 's. The  $\rho_i$  productions are simple,  $U$  extends into  $|E|$  many  $a$ 's or  $b$ 's or so on, and  $\delta$  extends into all possible vertices. Since the grammar and input string can be constructed in polynomial time from an underlying vertex cover problem  $\langle k, V, E \rangle$ , universal recognition of UCFG must be at least as hard as solving the vertex cover problem. Since the vertex cover problem is NP-complete, the universal recognition problem for totally unordered type 2 grammars is accordingly NP-hard. It is easy to see that recognition is also in NP. Simply guess a derivation and evaluate it in polynomial time.  $\square$

## Theorem

*Synchronous context-free recognition is NP-complete.*

Proof is omitted, for brevity. The lower bound proof goes by reduction of 3SAT. The idea is to recognize only string pairs  $\langle w_1 \dots w_n, w_c \rangle$  where  $w_c$  is a string representation of the clauses in the 3SAT input, and  $w_i$  with  $1 \leq i \leq n$  represents the truth assignment of the variable  $v_i$  in this input. The important thing to note is that it relies on productions of the form  $S \rightarrow \langle A_1 \dots A_l, B_1 \dots B_l \rangle$  where  $l$  is unbounded and polynomial in  $n$ . The upper bound is easily proven.

Some polynomial fragments are easily identified:

- $k$ -bound s-AVG recognition,
- $k$ -bound UCFG recognition, and
- $k$ -bound SCFG recognition.

The problem is *weak generative capacity*. The three linguistic theories identified (and their  $k$ -fragments too) generate only context-free languages. In context-sensitive extensions, simple  $k$ -bounds will not do.

# Reentrancy

- Of course the NP-hardness proof for unbounded agreement context-free recognition applies to standard attribute-value grammar (AVG) too. The ability to store information and let it percolate by reentrancy buys us context-sensitivity.
- Context-sensitive fragments with polynomial procedures: [KW95]. Others have identified fragments weakly equivalent to TAG [SNKK93] and [FW06], but translations – at least in [FW06] – are exponential. Complexity vs. generative capacity.

# Unordering

- The NP-hardness proof for unordered context-free recognition applies to ID/LP grammar, FO-TAG, MCTAG, linearization-based HPSG and so on too. The simple  $k$ -bound only works in the context-free case, but there are other restrictions.
- Context-sensitive fragments with polynomial procedures: [Kal05, SLM07]. Complexity vs. generative capacity: For instance, tree-local MCTAG is weakly equivalent to TAG, but NP-hard.

# Synchronism

It is known that recognition for binary SCFG is in  $\mathcal{O}(n^6)$ .

## 2 techniques: RCG and $k$ -ambiguity

- Two techniques investigated for non-synchronous grammars: polynomial translation into RCG and  $k$ -ambiguity. RCG provides a standard hierarchy, but unordering is indirect;  $k$ -ambiguity enables total unordering, but the hierarchy cross-cuts the Chomsky hierarchy.
- RCG technique is extended for synchronism.
- *Motivation*: (1) No fragments for both *reentrancy*, *unordering* and *synchronism*. (2) Little systematicity.

# Range concatenation grammar (RCG)

## Definition

An RCG  $G = \langle N, T, V, P, S \rangle$  is a 5-tuple with  $\dots$  and  $V$  a finite set of variable symbols, and  $P$  a finite set of productions of the form:

$$A_0(\alpha_1, \dots, \alpha_{\rho(A_0)}) \rightarrow A_1(\beta_1, \dots, \beta_{\rho(A_1)}), \dots, A_n(\gamma_1, \dots, \gamma_{\rho(A_n)})$$

with  $\dots$  and  $\alpha_i, \beta_i, \gamma_i \in (V \cup T)^*$ .

and

$$L(G) = \{\omega \mid S(\omega) \xrightarrow{*} \epsilon\}$$

## Theorem ([Bou04])

*A RCG produces a shared parse forest in time  $\mathcal{O}(n^k \times |G|)$  where  $k$  is the bound on arguments in a production rule.*

# From s-AVG to RCG

## Definition (Translation)

A translation  $\tau$  is defined from s-AVG  $G = \langle \langle \text{Attr}, \text{Atms}, \rho \rangle, \text{AgrAttr}, V_T, P, S_A \rangle$ , with  $\text{Attr} = \text{AgrAttr} \cup \{\text{CAT}\}$  and  $S_A = (\text{CAT}, S)$ , into  $R = \langle V_N, V_T, \text{Vars}, P', S_N \rangle$ . For each production in  $P$  of the form

$$A_0 \rightarrow A_1 \dots A_n$$

we introduce a production in  $P'$ :

$$A'_0(X_1 \dots X_n) \rightarrow A'_1(X_1), \dots, A'_n(X_n), \\ \{f(X_1 \dots X_n) \mid f \in \text{AgrAttr} \text{ and } f \in \text{DOM}(A_0)\}$$

where  $X \in \text{Var}$ ,  $A_j(\text{CAT}) = A'_j$ .  $\forall \nu \in \rho(\alpha_i)$ , we then introduce productions

$$\alpha_i(X) \rightarrow \alpha'_i(X) \\ \alpha'_i(\omega X) \rightarrow \alpha'_i(X) \\ \alpha'_i(\epsilon) \rightarrow \epsilon$$

for each production in  $P$

$$A \rightarrow \omega$$

where  $A(\alpha_i) = \nu$ .

Note that the number of productions in  $P'$  is bound by  $|P| + 2|\text{AgrAttr}| \times |\text{Atms}| + |P| \times |\text{AgrAttr}| \times |\text{Atms}|$ . Complexity for 2-s-AVG in RCG:  $\mathcal{O}(n^{3+|\text{AgrAttr}|})$ .

## MIX in RCG

$$\begin{aligned}S(X) &\rightarrow M(X, X, X) \\M(aX, bY, cZ) &\rightarrow M(X, Y, Z) \\M(TX, Y, Z) &\rightarrow \text{len}(1, T)a(\overline{T})M(X, Y, Z) \\M(X, TY, Z) &\rightarrow \text{len}(1, T)b(\overline{T})M(X, Y, Z) \\M(X, Y, TZ) &\rightarrow \text{len}(1, T)c(\overline{T})M(X, Y, Z) \\M(\epsilon, \epsilon, \epsilon) &\rightarrow \epsilon \\a(a) &\rightarrow \epsilon \\b(b) &\rightarrow \epsilon \\c(c) &\rightarrow \epsilon\end{aligned}$$

It is easy to see that RCG provides *total* unordering up to  $k$ . If  $k$  is unbounded, RCG turns NP-hard. Complexity: linear, but  $\mathcal{O}(n^9)$  for the class.

# From SCFG to RCG

A similar translation exists for SCFGs. Complexity for 2-SCFG:  $\mathcal{O}(n^6)$ .

## Example

$S$	$\rightarrow$	$\langle NP_1 PP_2 VP_3, NP_1 VP_3 PP_2 \rangle$
$NP$	$\rightarrow$	$\langle Baoweier, Powell \rangle$
$PP$	$\rightarrow$	$\langle yu Shalong, with Sharon \rangle$
$VP$	$\rightarrow$	$\langle juxing le huitan, held a meeting \rangle$
$VP$	$\rightarrow$	$\langle juxing le huitan, met \rangle$

translates into

$S(C_1 C_2 C_3, E_1 E_3 E_2)$	$\rightarrow$	$NP(C_1, E_1) PP(C_2, E_2) VP(C_3, E_3)$
$NP(Baoweier, Powell)$	$\rightarrow$	$\epsilon$
$PP(yu Shalong, with Sharon)$	$\rightarrow$	$\epsilon$
$VP(juxing le huitan, held a meeting)$	$\rightarrow$	$\epsilon$
$VP(juxing le huitan, met)$	$\rightarrow$	$\epsilon$

- RCG provides a systematic method to map out the complexity of reentrancy, unordering and synchronism.
- RCG is modular with respect to intersection.

## $k$ -ambiguity

Say  $\Longrightarrow_1$  denotes ordinary type 2 (context-free) derivability, then  $\Longrightarrow_2$  is totally unordered type 2 derivability iff:

### Definition (Derivability)

If  $A \xRightarrow{*}_1 \omega$  and  $\omega' \in \text{permute}(\omega)$ , then  $A \xRightarrow{*}_2 \omega'$ .

- Polynomial fragment of totally unordered s-AVG (AVG)?
- The NP-hardness proofs rely on reentrancy/unordering and ambiguity, so why not restrict ambiguity instead?

## Definition ( $\omega$ -grammar)

Say you have a type 2 grammar in Chomsky normal form  $G = \langle N, T, P, S \rangle$  and some string  $\omega_1 \dots \omega_n$ . Construct  $G_\omega = \langle N_\omega, T_\omega, P_\omega, \{S_\omega\} \rangle$  such that

$$T_\omega = \{\omega_1, \dots, \omega_n\}$$

and, recursively

- (a) ( $\omega_i \in T_\omega$  and  $A \rightarrow \omega_i \in P$ )  $\Rightarrow$  ( ${}_i A_i \in N_\omega$  and  ${}_i A_i \rightarrow \omega_i \in P_\omega$ )
- (b) ( ${}_i B_j, {}_{j+1} C_k \in N_\omega$  and  $A \rightarrow BC$ )  $\Rightarrow$  ( ${}_i A_k \in N_\omega \wedge {}_i A_k \rightarrow {}_i B_j, {}_{j+1} C_k \in P_\omega$ )

## Lemma

Say  $G$  is a type 2 grammar in Chomsky normal and  $\omega \in T^*$ . It now holds that  $|C_{G,\omega}| \leq ((|N| \times \frac{n^2+n}{2} - |N| \times n) \times (n-1) \times |N|^2) + (|N| \times n) + n$ .

## Definition ( $\omega$ -grammar)

Say you have a totally unordered type 2 grammar in Chomsky normal form  $G = \langle N, T, P, S \rangle$  and some string  $\omega_1 \dots \omega_n$ . Construct  $G_\omega = \langle N_\omega, T_\omega, P_\omega, S \rangle$  such that

$$T_\omega = \{\omega_1, \dots, \omega_n\}$$

and, recursively

- (a)  $(\omega_i \in T_\omega \text{ and } A \rightarrow \omega_i \in P) \Rightarrow (A_{\{i\}} \in N_\omega \text{ and } A_{\{i\}} \rightarrow \omega_i \in P_\omega)$
- (b)  $(B_\Sigma, C_{\Sigma'} \in N_\omega \text{ and } \Sigma \cap \Sigma' = \emptyset \text{ and } A \rightarrow BC) \Rightarrow (A_{\Sigma \cup \Sigma'} \in N_\omega \wedge A_{\Sigma \cup \Sigma'} \rightarrow B_\Sigma C_{\Sigma'} \in P_\omega)$

## Lemma (Chart size, upper bound)

Say  $G$  is a totally unordered type 2 grammar in Chomsky normal and  $\omega \in T^*$ . It now holds that

$$|C_{G,\omega}| \leq ((|N| \times 2^n - |N| \times n)^2 \times (n-1)^2 \times |N|^2) + (n \times |N|) + n.$$

## Definition

A grammar is said to be  $k$ -ambiguous iff all signs are combined unambiguously after  $k$  steps.

## Lemma

*Recognition for rigid and  $k$ -ambiguous totally unordered  $s$ -AVG in  $P$ .*

## Proof.

Since composition is totally unordered, you check  $n \times n - 1$  combinations in the first step.  $k$  many times, you have to check  $n \times n - i$  combinations for each solution you obtained. It is easy to see, however, that the total

number of steps is smaller than  $\sum_{\substack{i < n \\ 1 \leq i}} (n^k (n - i))$ . □

Since s-AVSs are of fixed size, and since unification is linear:

### Theorem

*Totally unordered  $k$ -ambiguous s-AVG recognition is in  $P$ .*

The same holds for totally unordered  $k$ -ambiguous AVGs if AVSs are at most polynomial in  $n$ , even relative to an inheritance hierarchy.

### Theorem

*$k$ -ambiguous grammars are non-superfinite.*



Edward Barton.

The computational difficulty of ID/LP parsing.

In *Proceedings of the 23th Annual Meeting of the Association for Computational Linguistics*, pages 76–81, Chicago, Illinois, 1985.



Edward Barton, Robert Berwick, and Erik Ristad.

*Computational complexity and natural language*.

MIT Press, Cambridge, Massachusetts, 1987.



Pierre Boullier.

Range concatenation grammars.

In *New developments in parsing technology*, pages 269–289. Kluwer, 2004.



Daniel Feinstein and Shuly Wintner.

Highly constrained unification grammars.

In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 1089–1096, Sydney, Australia, 2006.



Michael Garey and David Johnson.

*Computers and intractability*.

W. H. Freeman & Co., New York, New York, 1979.



Laura Kallmeyer.

Tree-local multicomponent tree-adjoining grammars with shared nodes.

*Computational Linguistics*, 31(2):187–225, 2005.



Bill Keller and David Weir.

A tractable extension of linear indexed grammars.

In *Proceedings of the 7th European Chapter of the Association for Computational Linguistics*, pages 75–82, Dublin, Ireland, 1995.



Anders Søgaard, Timm Lichte, and Wolfgang Maier.

On the complexity of linguistically motivated extensions of tree-adjoining grammar.

In *Proceedings of Recent Advances in Natural Language Processing 2007*, Borovets, Bulgaria, 2007.

To appear.



Hiroyuki Seki, Ryuichi Nakanishi, Yuichi Kaji, and Sachiko Ando and 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 the Association for Computational Linguistics*, pages 130–139, Columbus, Ohio, 1993.



Giorgio Satta and Enoch Peserico.

Some computational complexity results for synchronous context-free grammars.

In *Proceedings of Human Language Technology Conference and Conference for Empirical Methods in Natural Language Processing*, pages 803–810, Vancouver, Canada, 2005.