Optimal Parsing Strategies for Linear Context-Free Rewriting Systems

Daniel Gildea

Computer Science Department

University of Rochester

Overview

- Factorization lowers rank of LCFRS rules
- Binarization minimizes parsing complexity
- Minimizing fan-out does not minimize parsing complexity

Linear Context-Free Rewriting Systems

LCFRS generalizes CFG, TAG, CCG, SCFG, STAG.

Productions $p \in P$ take the form:

$$p : A \longrightarrow g(B_1, B_2, \ldots, B_r)$$

where $A, B_1, \ldots B_r \in V_N$, and g is a **linear, non-erasing** function

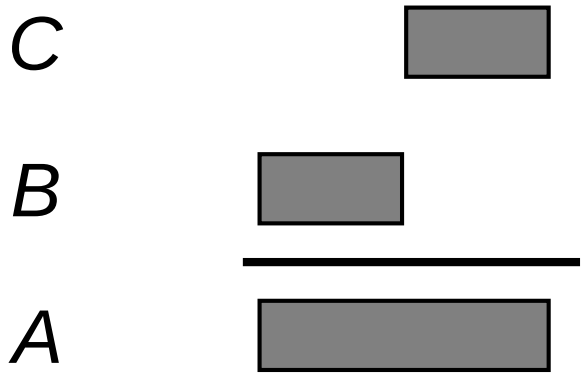
$$g(\langle x_{1,1}, \ldots, x_{1,\varphi(B_1)} \rangle, \ldots, \langle x_{1,1}, \ldots, x_{1,\varphi(B_r)} \rangle) = \langle t_1, \ldots, t_{\varphi(A)} \rangle$$

(Vijay-Shankar et al. ACL 1987)

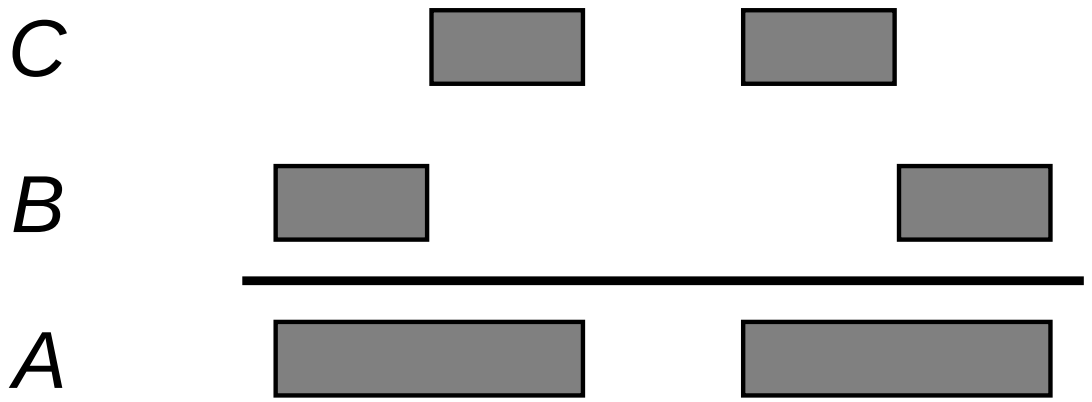
Context-Free Grammar

$$g(\langle x_B \rangle, \langle x_C \rangle) = \langle x_B x_C \rangle$$

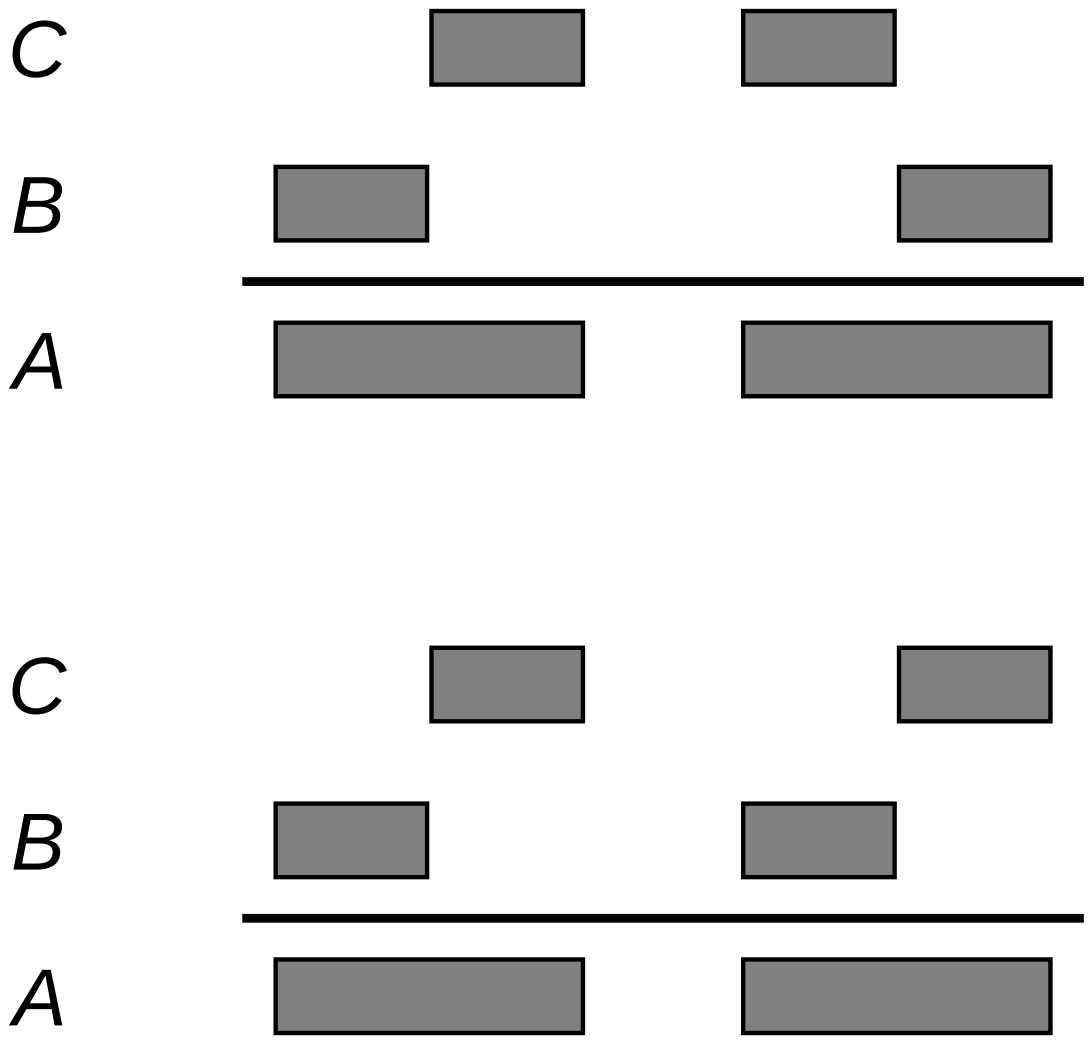
$$A \longrightarrow B\ C$$



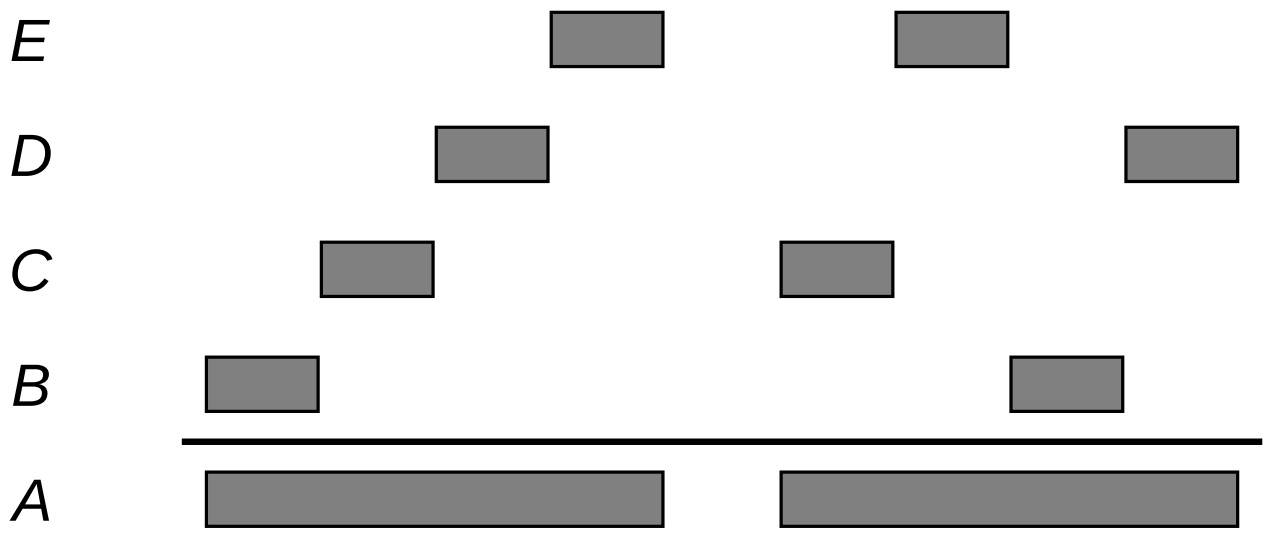
Tree-Adjoining Grammar



Inversion Transduction Grammar



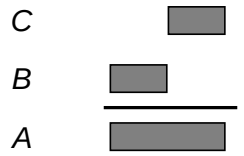
Synchronous Context-Free Grammar (SCFG)



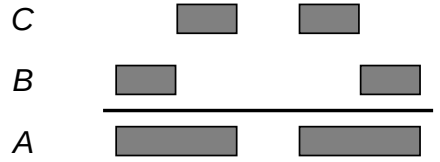
Fan-Out

Number of spans in nonterminal.

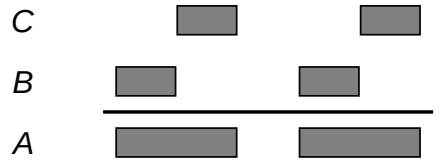
CFG: fan-out 1



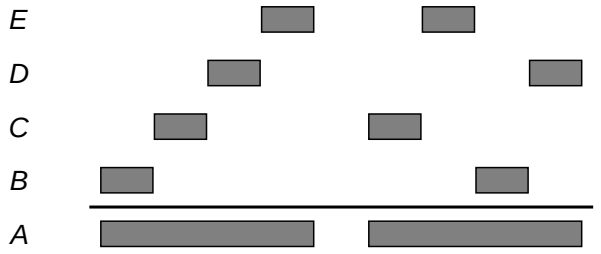
TAG: fan-out 2



ITG: fan-out 2



SCFG: fan-out 2



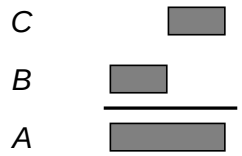
$$\varphi(G) = \max_{N \in G} \varphi(N)$$

(Rambow & Satta, 1999)

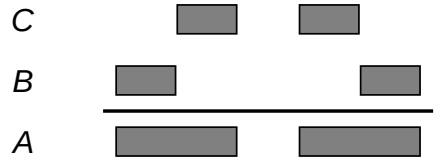
Rank

Number of nonterminals on righthand side of rule.

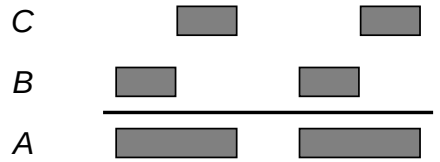
CFG: rank 2



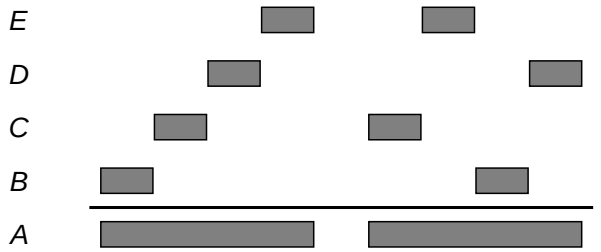
TAG: rank 2



ITG: rank 2



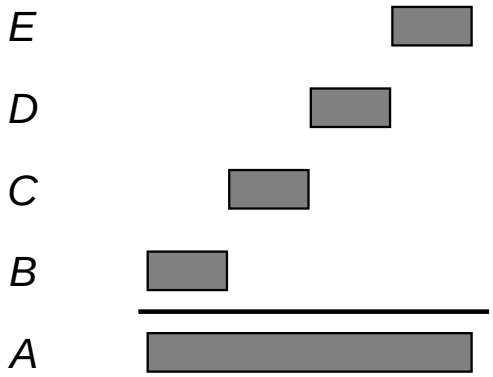
SCFG: rank *r*



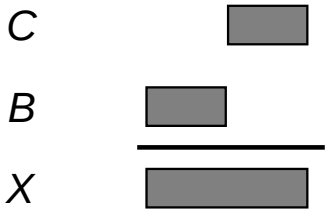
$$\rho(G) = \max_{P \in G} \rho(P)$$

Factorization

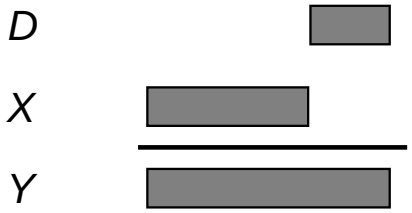
Reduces rank



$A \rightarrow B\ C\ D\ E$



$X \rightarrow B\ C$

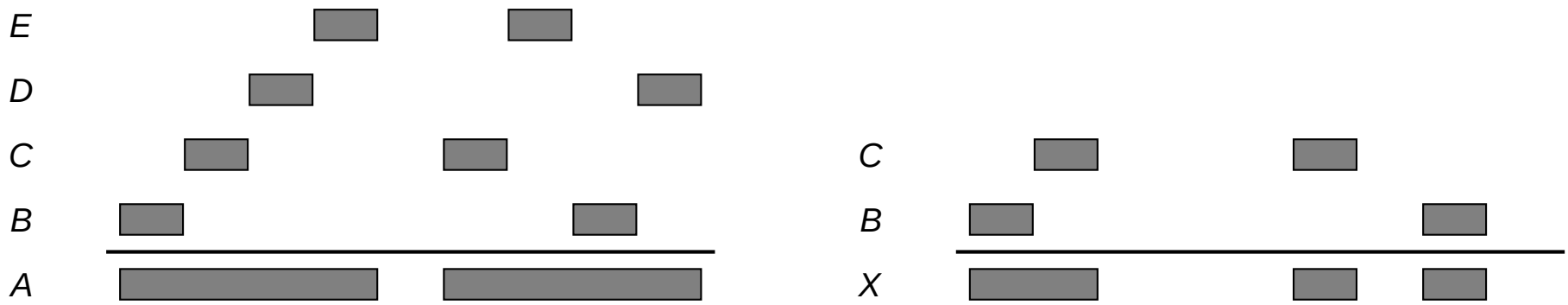


$Y \rightarrow X\ D$

$A \rightarrow Y\ E$

Factorization

Reduces rank, may increase fan-out



Factorization Algorithms

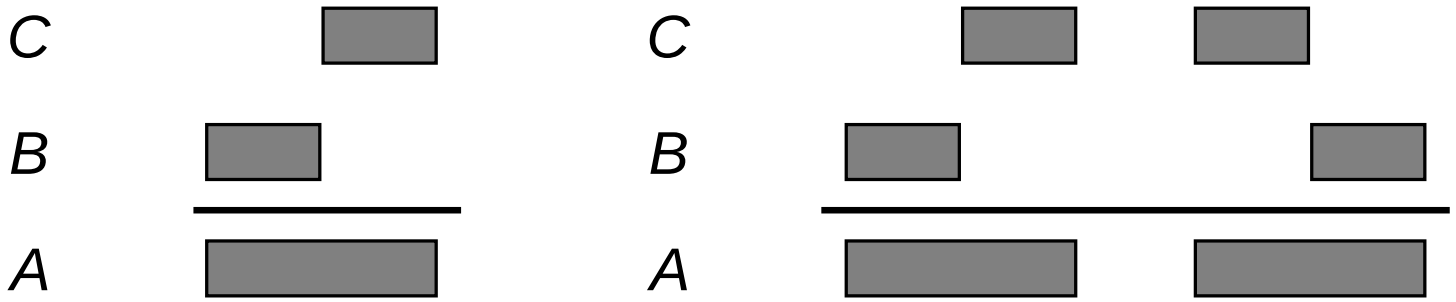
- SCFG $\longrightarrow$ rank 2 (Zhang et al., NAACL 2006)
- SCFG $\longrightarrow$ minimum rank in $O(n)$

  (Zhang & Gildea, SSST 2007)
- LCRFS fan-out 2 $\longrightarrow$ rank 2, fan-out 2 in $O(n)$

  (Sagot & Satta, ACL 2010)
- LCRFS $\longrightarrow$ rank 2, min fan-out in $O(n^{\varphi})$

  (Gomez-Rodriguez et al., NAACL 2009)

Parsing Complexity



$O(n^3)$ $O(n^6)$

For $p : A \longrightarrow g(B_1, \ldots B_r),\ O(n^{c(p)})$

$c(p) = \varphi(A) + \sum_{i=1}^{r} \varphi(B_i)$ (Seki et al. 1991)

Parsing Complexity

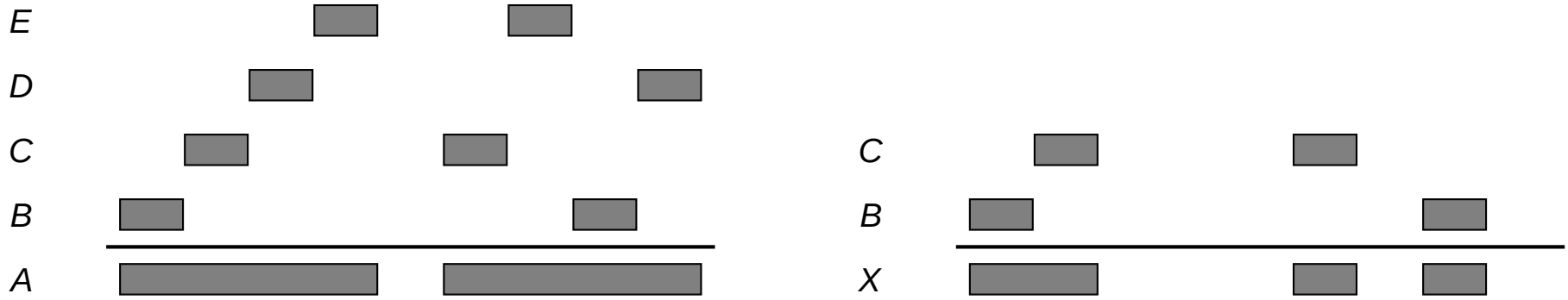
$$c(p) = \varphi(A) + \sum_{i=1}^{r} \varphi(B_i)$$

$$c(G) = \max_{p \in G} c(p)$$

$$c(G) \leq (\rho(G) + 1)\varphi(G)$$

Factorization

Never increases parsing complexity.



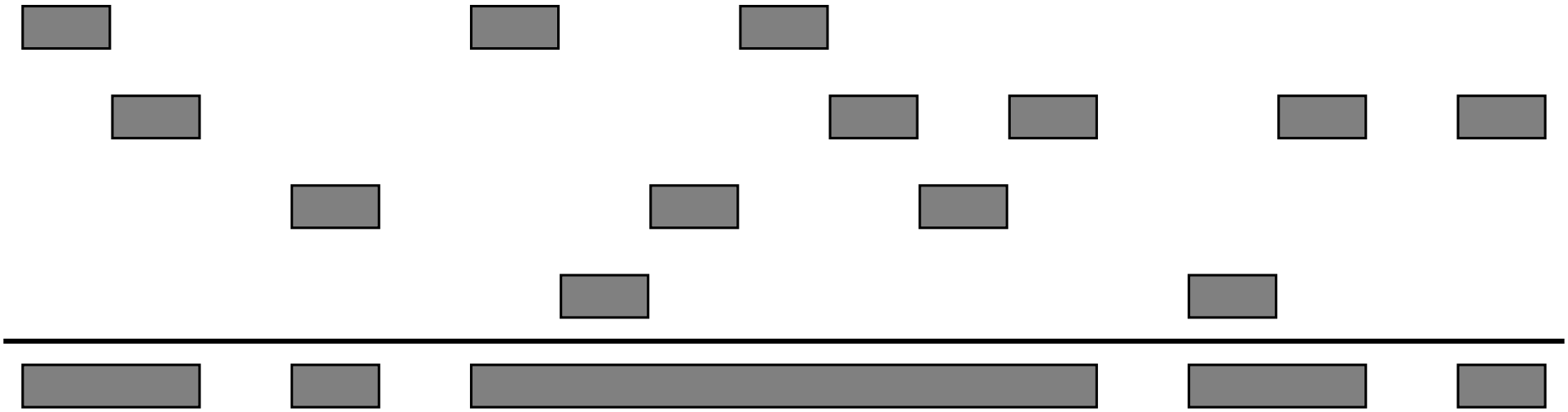
Binarization minimizes parsing complexity.

Among binarizations,

minimizing fan-out and

minimizing parsing complexity

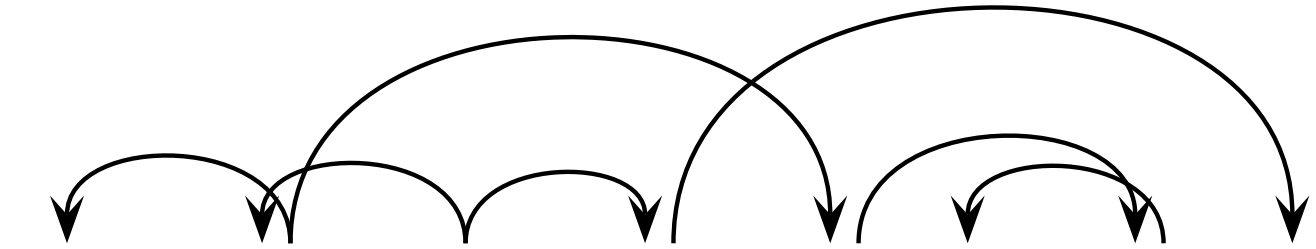
are INCONSISTENT.



Parsing complexity 14 w/ fan-out 6.

Minimum fan-out among binarization = 5.

Dependency Treebank Experiments



$$\text{nmod} \rightarrow g_1 \qquad g_1 = \langle\, A \,\rangle$$

$$\text{sbj} \rightarrow g_2(\text{nmod}, \text{pp}) \qquad g_2(\langle x_{1,1}\rangle, \langle x_{2,1}\rangle) = \langle x_{1,1}\ hearing\ , x_{2,1}\rangle$$

$$\text{root} \rightarrow g_3(\text{sbj}, \text{vc}) \qquad g_3(\langle x_{1,1}, x_{1,2}\rangle, \langle x_{2,1}, x_{2,2}\rangle) = \langle x_{1,1}\ is\ x_{2,1} x_{1,2} x_{2,2}\rangle$$

$$\text{vc} \rightarrow g_4(\text{tmp}) \qquad g_4(\langle x_{1,1}\rangle) = \langle\, scheduled\ , x_{1,1}\rangle$$

$$\text{pp} \rightarrow g_5(\text{tmp}) \qquad g_5(\langle x_{1,1}\rangle) = \langle\, on\ x_{1,1}\rangle$$

$$\text{nmod} \rightarrow g_6 \qquad g_6 = \langle\, the \,\rangle$$

$$\text{np} \rightarrow g_7(\text{nmod}) \qquad g_7(\langle x_{1,1}\rangle) = \langle x_{1,1}\ issue\,\rangle$$

$$\text{tmp} \rightarrow g_8 \qquad g_8 = \langle\, today \,\rangle$$

Dependency Treebank Experiments

Kuhlmann and Nivre (ACL 2006) define “mildly non-projective dependency structures”.

Gomez-Rodriguez et al. (ACL 2009) define “mildly ill-nested dependency structures” parsed in $O(n^{3k+4})$.

Treebank Parsing Complexity

complexity	arabic	czech	danish	dutch	german	port	swedish
20							1
18							1
16							1
15							1
13							1
12						2	3
11					1	1	1
10		2			6	16	3
9					7	4	1
8		4		7	129	65	10
7		3		12	89	30	18
6		178	11	362	1811	492	59
5	48	1132	93	411	1848	172	201
4	250	18269	1026	6678	18124	2643	1736
3	10942	265202	18306	39362	154948	41075	41245

Conclusion

- Parsing complexity $\neq$ fan-out
-

Conclusion

- Parsing complexity $\neq$ fan-out
- Parsing complexity = 20

Space Complexity

- space complexity = $O(n^{2\varphi(G)})$
- Factorization **never** improves space complexity.

1: **function** MINIMAL-BINARIZATION(p, $\prec$)
2: workingSet $\leftarrow \emptyset$;
3: agenda $\leftarrow$ priorityQueue($\prec$);
4: **for** i from 1 to $\rho(p)$ **do**
5: workingSet $\leftarrow$ workingSet $\cup \{B_i\}$;
6: agenda $\leftarrow$ agenda $\cup \{B_i\}$;
7: **while** agenda $\neq \emptyset$ **do**
8: $p' \leftarrow$ pop minimum from agenda;
9: **if** nonterms(p') = $\{B_1, \ldots B_{\rho(p)}\}$ **then**
10: **return** p';
11: **for** $p_1 \in$ workingSet **do**
12: $p_2 \leftarrow$ newProd(p', p_1);
13: find $p_2' \in$ workingSet : nonterms(p_2') = nonterms(p_2);
14: **if** $p_2 \prec p_2'$ **then**
15: workingSet $\leftarrow$ workingSet $\cup \{p_2\} \setminus \{p_2'\}$;
16: push(agenda, p_2);