Linguistic Universals in Grammars and Language Models

Miloš Stanojević

UCL

talk at ILFC seminar 12 Feb 2025

What are linguistic universals?

Linguistic Universal

- Any property of language that is universally true across the observed human languages.

Why study linguistic universals?

- For Cognitive Science:
  - Why do universals appear?
  - Part of the innate universal grammar?
  - Learnable from data?
- For NLP:
  - Evaluation: Our language models should be good for all human languages.
  - Modeling: Inductive biases.

Do LLMs learn a true syntactic universal?

John T. Hale and Miloš Stanojević

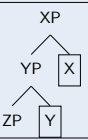
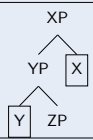
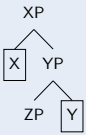
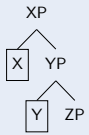
EMNLP 2024

Introduction

- The question of whether large language models (LLMs) display human-level competence has provoked lively discussion.
  - "LLMs have already demonstrated that human-like grammatical language can be acquired without the need for a built-in grammar" (Contreras Kallens et al., 2023).
  - Others are more guarded: "nearly all studies have reported that [deep neural networks'] behavior deviated from the idealized syntactic competence that a linguist might postulate" (Linzen and Baroni, 2021).
- Here we focus on evaluating whether LLMs learn a syntactic universal.
  - If they can, that would underwrite strong nativist arguments for inborn biases (e.g. Chomsky, 1965, 25).
  - Also it would show if multilingual LLMs can learn universal properties holding for all human languages.

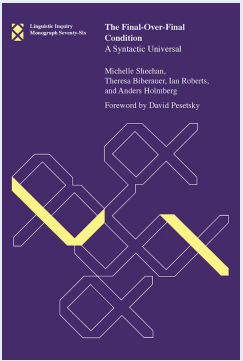
A Syntactic Universal: Final-over-Final Condition (FOFC)

Possible head-projections

YP / XP	Final	Initial
Final over		
Initial over		

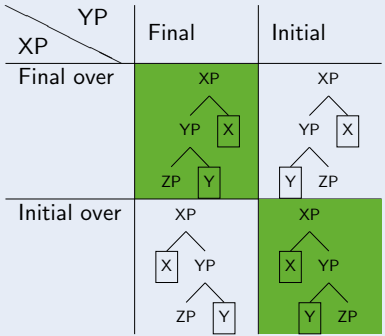
book

(Holmberg, 2000; Sheehan et al., 2017)



A Syntactic Universal: Final-over-Final Condition (FOFC)

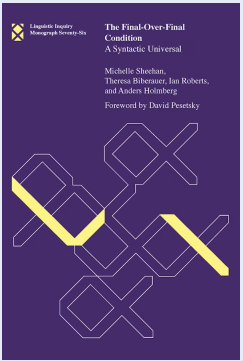
Possible head-projections



- Harmonic orders are all good.

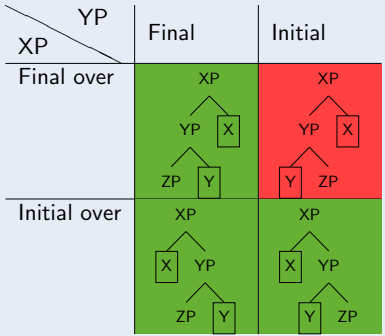
book

(Holmberg, 2000; Sheehan et al., 2017)



A Syntactic Universal: Final-over-Final Condition (FOFC)

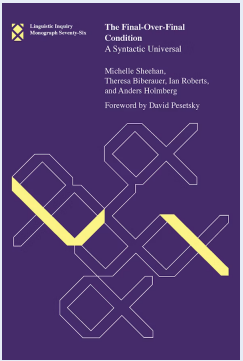
Possible head-projections



- In disharmonic orders Initial-over-Final is (strongly) preferred over Final-over-Initial.

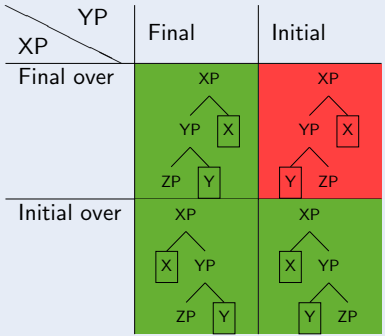
book

(Holmberg, 2000; Sheehan et al., 2017)



A Syntactic Universal: Final-over-Final Condition (FOFC)

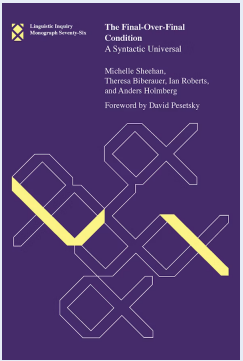
Possible head-projections



- This does not hold for all X and Y. We focus on X=Aux, Y=V, Z=Obj.

book

(Holmberg, 2000; Sheehan et al., 2017)

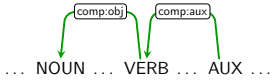


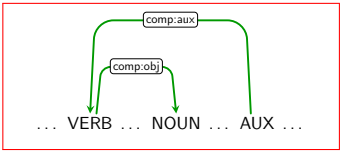
FOFC

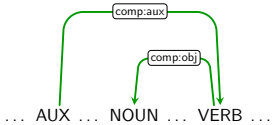
- We don't take FOFC for granted.
  - We first test its empirical validity for langauges under study.
- We do not assume strength of the claim.
  - Both strong formal universal or statistical tendency is consistent with our approach.
- We do not put a claim on the reasons for FOFC universality.
  - Both performance and competence explanations for FOFC existance are consistent with our approach.
- We want to know if it can be acquired from data alone.

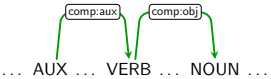
Part 1: FOFC corpus study for Aux>V>O

1. Reinterpret FOFC with Surface-UD dependencies.
2. Parse mC4 with Stanza for mixed-head languages.
   - Basque, Russian, German, Hungarian, Serbian.
3. Look for FOFC breaking examples.
4. Do human eval because parsing can be inaccurate (78% – 90%).









Part 1: FOFC corpus study for Aux>V>O

Hungarian

	AUX <V	V < AUX
V < O	20754	320
O < V	9530	4401

Basque

	AUX <V	V < AUX
V < O	79119	1632
O < V	291281	7099566

German

	AUX <V	V < AUX
V < O	3	13
O < V	74	34498

Russian

	AUX <V	V < AUX
V < O	1335724	20690
O < V	77485	40055

Serbian

	AUX <V	V < AUX
V < O	212251	2197
O < V	13928	2442

Part 1: FOFC corpus study for Aux>V>O

Hungarian

	AUX <V	V < AUX
V < O	20754	320
O < V	9530	4401

Basque

	AUX <V	V < AUX
V < O	79119	1632
O < V	291281	7099566

German

	AUX <V	V < AUX
V < O	3	13
O < V	74	34498

Russian

	AUX <V	V < AUX
V < O	1335724	20690
O < V	77485	40055

Serbian

	AUX <V	V < AUX
V < O	212251	2197
O < V	13928	2442

- count(FOFC abiding) $\gg$ count(FOFC breaking)
- Manual inspection of FOFC breaking examples:
  - parsing errors, tagging errors, sentence segmentation errors
  - topicalization,
  - poetic expressions (e.g. 15th centuary religious German song).

Part 1: FOFC corpus study results

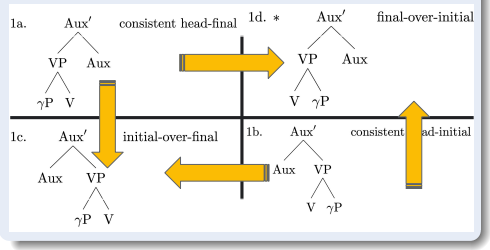
language	sentence count	χ^2	significance level
Hungarian	182M	6498	$p < 10^{-16}$
Basque	41M	1499197	$p < 10^{-16}$
Russian	483M	290076	$p < 10^{-16}$
Serbian	124M	14906	$p < 10^{-16}$
German	20M	clearly follows from Biberauer (2017)	

Part 2: Testing if LLMs learn FOFC

Experiment plan

- Find sentences with harmonic word-order.
- Convert them to both disharmonic word-orders.
- Compute log-probabilities of each sentences under LLM.
- Find the difference between FOFC-abiding and FOFC-breaking disharmonic sentences.
- The more positive the value, the better the model has learned FOFC.

Conversion direction.

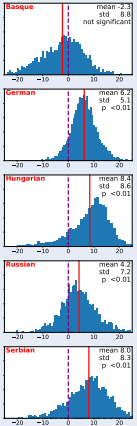


Do Gemini Pro and PaLM learn FOFC?

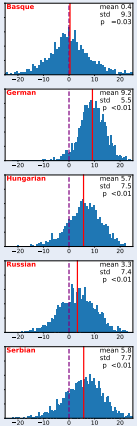
Result

- Almost – 4 out of 5 language
- But not universally – Basque is clearly a difficult language for both models.

Gemini Pro



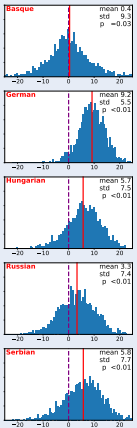
PaLM 8 Billion



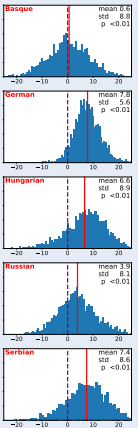
PaLM – Model Size Effect

- Increasing model size helps.
- However, doesn't solve problem completely:
  - Almost half of the sentence pairs in Basque are misclassified even in the largest half-trillion PaLM model.

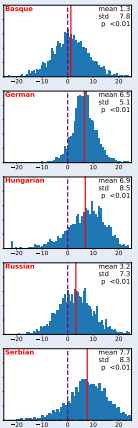
PaLM 8 Billion



PaLM 64 Billion



PaLM 540 Billion



PaLM – Data Size Effect

- Amount of data reflects how well does the model learn universal language properties.
- Basque is most difficult to learn likely because:
  - It has small amount of training data and
  - It is an isolate so transfer from other lanaguages is less likely.
- Serbian may seem a counter-example but:
  - Serbian is very similar (or even same) as Croatian and Bosnian so amount of data is much larger.
  - Transfer from other related Slavic languages may have an effect as well.

PaLM training data

language	tokens
Basque	153M
German	25,954M
Hungarian	555M
Russian	3,932M
Serbian	373M
Croatian	198M
Bosnian	427M

So, can LLMs learn FOFC?

- Final-Over-Final Condition is universally true.
  - At least in languages and constructions we have tested.
- Larger parameter count and amount data helps, but doesn't solve the problem.
  - Even half-trillion parameters and trillion tokens is not enough for universal FOFC learning.
- LLMs don't generalize universaly.
  - They learn FOFC constraints for each language separately.
- Possible solution:
  - LLM learning of FOFC from plausible amount of data humans are exposed to requires stronger inductive biases.
  - Those biases could come from:
    - targetted pretraining (Papadimitriou and Jurafsky, 2023; Lindemann et al., 2024) or
    - pre-wiring the neural architecture (Sartran et al., 2022; Murty et al., 2023).

But, what if LLMs could learn FOFC?

- If LLM could learn FOFC they would be a good tool.
- But does it make them a good theory of language?
  - Mark Steedman: "Neural Networks work in practice, but do they work in theory?"
  - NN practice:
    Performance is great but on its own provides no explanation for why universals are there.
  - NN theory:
    There is progress (Hahn, 2020; Weiss et al., 2021; Hao et al., 2022; Strobl et al., 2024)
- Syntactic theory can provide explanation much more clearly.
  - FOFC performance explanation (Hawkins, 2014).
  - FOFC competence explanation (Biberauer et al., 2014; Clem, 2022).
  - Greenberg Universal 20 – what follows.

Formal Basis of a Language Universal

Miloš Stanojević and Mark Steedman

Computational Linguistics 2021

Introduction

- Language Universals
  - statistical universals (i.e. Greenberg-style universals).
  - **formal universal** that explains statistical ones.
- Stems from grammar expressivity and type of automata.
- Theoretical and practical importance.
  - Theoretical: If CCG is the true Universal Grammar, what would be the predictions?
  - Practical: constrains the space of possible word orders

What is CCG?

Categories

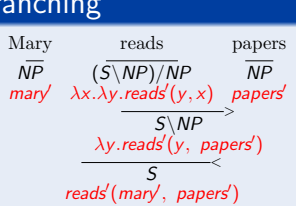
- basic categories: N, NP, S
- functional categories: $S\backslash NP$
- higher-order categories: $S/(S\backslash NP)$

Combinators

$>$	forward application:	A/B	B	$\Rightarrow$	A
$<$	backward application:	B	$A\backslash B$	$\Rightarrow$	A
B$>$	forward composition:	A/B	B/C	$\Rightarrow$	A/C
B$<$	backward composition:	$B\backslash C$	$A\backslash B$	$\Rightarrow$	$A\backslash C$
T$>$	type raising:	A		$\Rightarrow$	$B/(B\backslash A)$

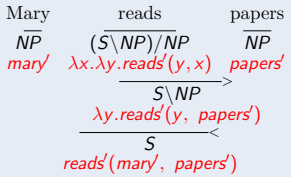
Example 1

Right Branching

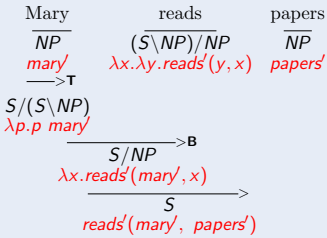


Example 1

Right Branching

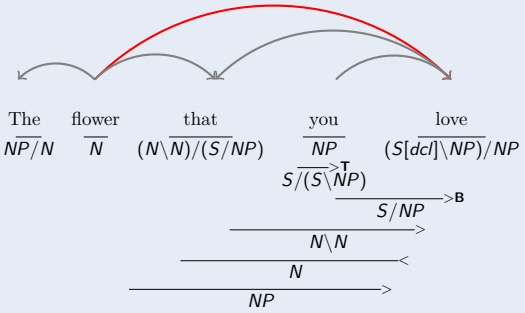


Left Branching



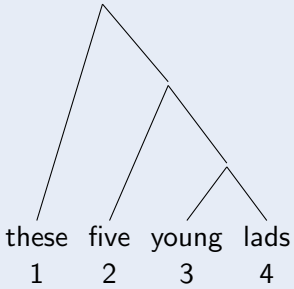
Example 2

Non-Local Dependencies

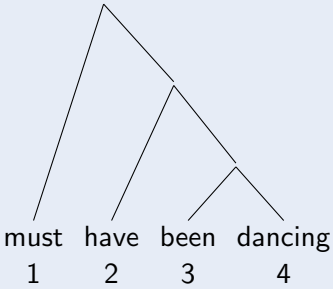


Core elements of NP and VP

Noun Phrase



Verb Phrase



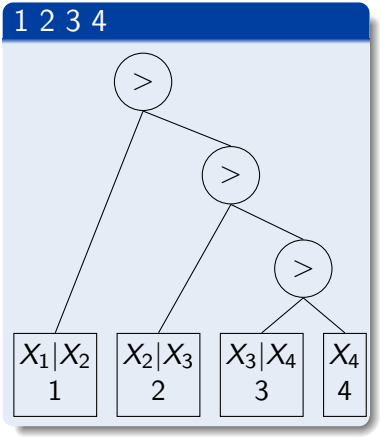
Typology of Permutations

		NP	NP	VP
		Cinque (2005)	Nchare (2012)	Abels (2016)
a.	1 2 3 4	✓	✓	✓
b.	1 2 4 3	✓	✓	✓
c.	1 4 2 3	✓	✗	✓
d.	4 1 2 3	✓	✗	✓
e.	2 1 3 4	✗	✓	✗
f.	2 1 4 3	✗	✓	✓
g.	2 4 1 3	✗	✗	✗
h.	4 2 1 3	✗	✗	✓
i.	3 1 2 4	✗	✓	✗
j.	3 1 4 2	✗	✗	✗
k.	3 4 1 2	✓	✓	✓
l.	4 3 1 2	✓	✓	✓
m.	1 3 2 4	✓	✓	✓
n.	1 3 4 2	✓	✓	✓
o.	1 4 3 2	✓	✓	✓
p.	4 1 3 2	✓	✓	✓
q.	2 3 1 4	✗	✓	✗
r.	2 3 4 1	✓	✓	✓
s.	2 4 3 1	✓	✓	✓
t.	4 2 3 1	✓	✓	✓
u.	3 2 1 4	✗	✓	✗
v.	3 2 4 1	✗	✓	✗
w.	3 4 2 1	✓	✓	✓
x.	4 3 2 1	✓	✓	✓

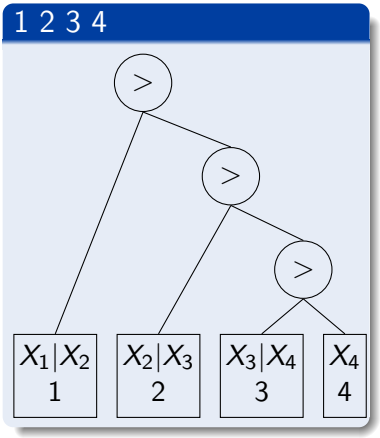
Typology of Permutations

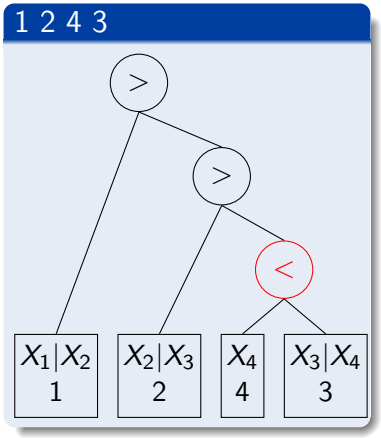
		NP	NP	VP
		Cinque (2005)	Nchare (2012)	Abels (2016)
a.	1 2 3 4	✓	✓	✓
b.	1 2 4 3	✓	✓	✓
c.	1 4 2 3	✓	✗	✓
d.	4 1 2 3	✓	✗	✓
e.	2 1 3 4	✗	✓	✗
f.	2 1 4 3	✗	✓	✓
g.	2 4 1 3	✗	✗	✗
h.	4 2 1 3	✗	✗	✓
i.	3 1 2 4	✗	✓	✗
j.	3 1 4 2	✗	✗	✗
k.	3 4 1 2	✓	✓	✓
l.	4 3 1 2	✓	✓	✓
m.	1 3 2 4	✓	✓	✓
n.	1 3 4 2	✓	✓	✓
o.	1 4 3 2	✓	✓	✓
p.	4 1 3 2	✓	✓	✓
q.	2 3 1 4	✗	✓	✗
r.	2 3 4 1	✓	✓	✓
s.	2 4 3 1	✓	✓	✓
t.	4 2 3 1	✓	✓	✓
u.	3 2 1 4	✗	✓	✗
v.	3 2 4 1	✗	✓	✗
w.	3 4 2 1	✓	✓	✓
x.	4 3 2 1	✓	✓	✓

CCG view of permutations

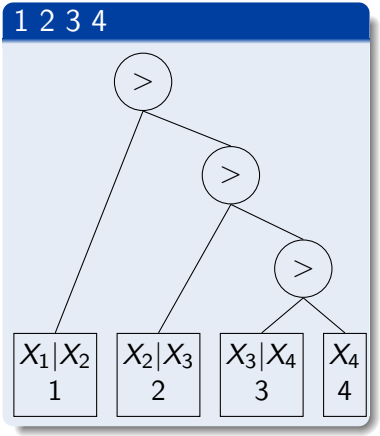


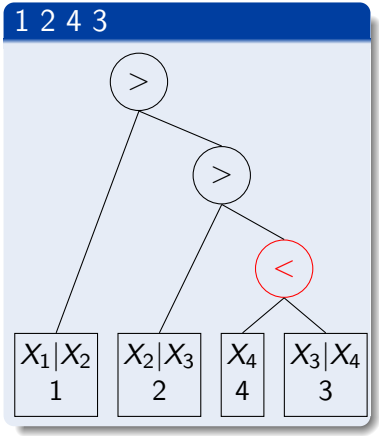
CCG view of permutations

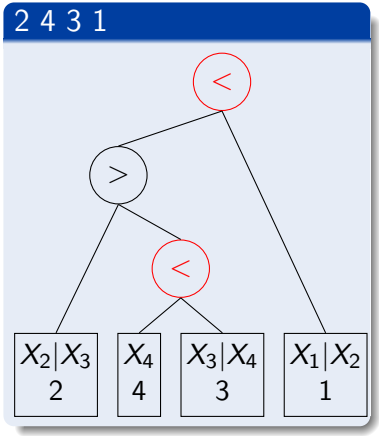




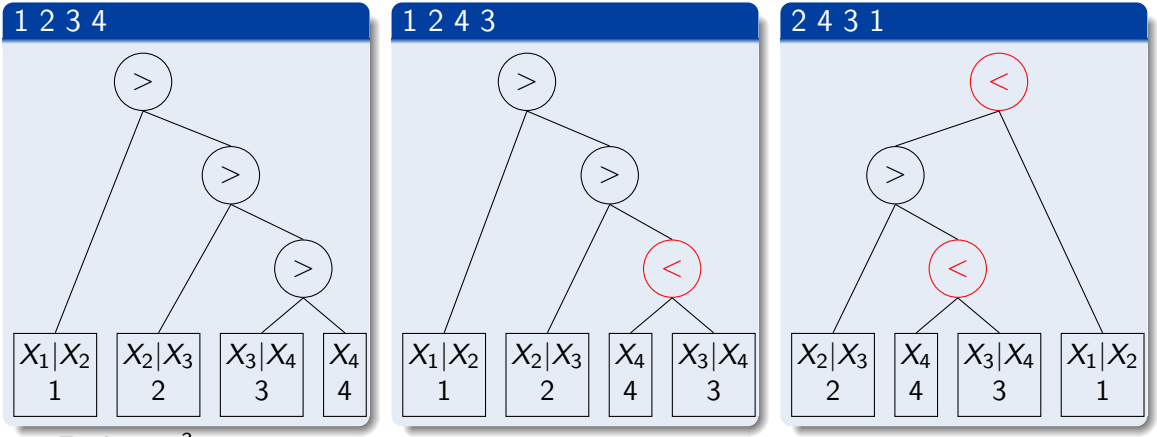
CCG view of permutations





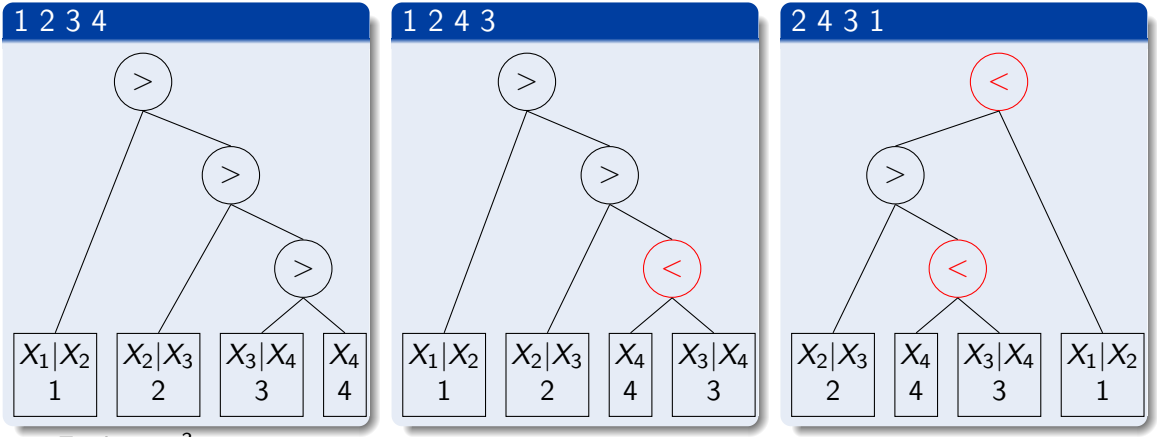


CCG view of permutations



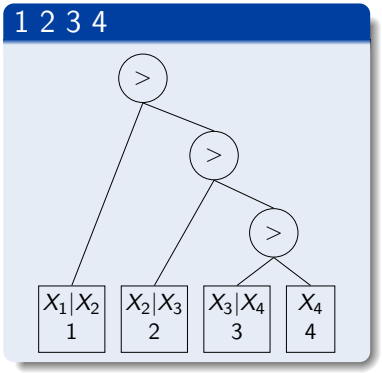
- Explains $2^3 = 8$ permutations.

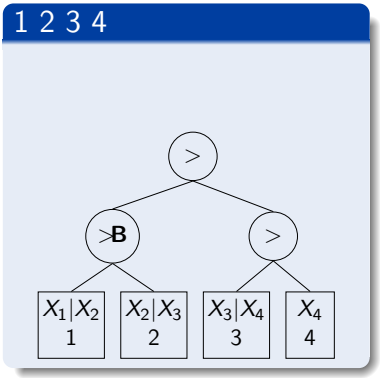
CCG view of permutations



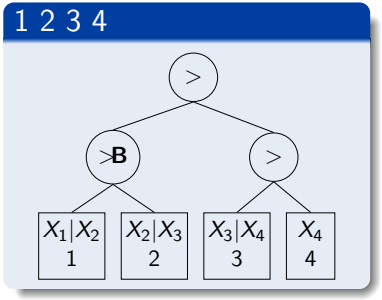
- Explains $2^3 = 8$ permutations.
- How about other permutations?

CCG alternative derivations

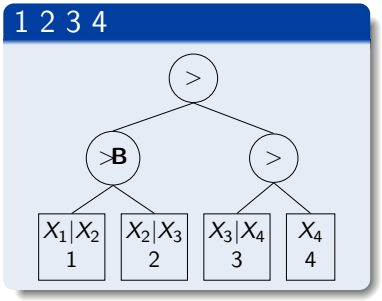


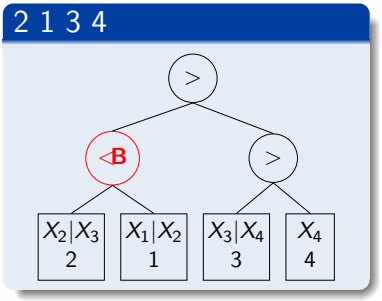


CCG derivation of 2 1 3 4



CCG derivation of 2 1 3 4





CCG is not permutation complete

- Can we now generate all $4! = 24$ permutations?

CCG is not permutation complete

- Can we now generate all 4! = 24 permutations? No

2 4 1 3

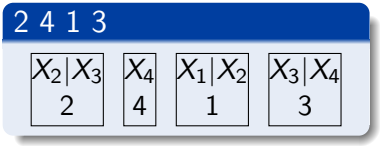
$X_2 \mid X_3$	X_4	$X_1 \mid X_2$	$X_3 \mid X_4$
2	4	1	3

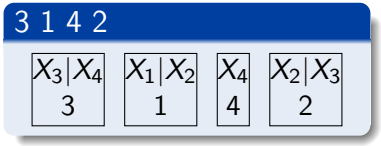
3 1 4 2

$X_3 \mid X_4$	$X_1 \mid X_2$	X_4	$X_2 \mid X_3$
3	1	4	2

CCG is not permutation complete

- Can we now generate all 4! = 24 permutations? No

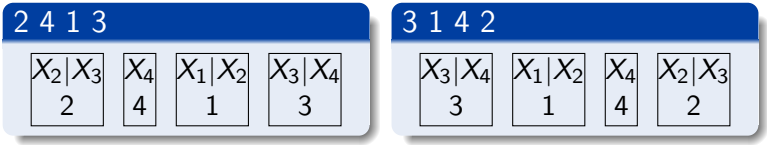




		NP	NP	VP
		Cinque (2005)	Nchare (2012)	Abels (2016)
a.	1 2 3 4	√	√	√
b.	1 2 4 3	√	√	√
c.	1 4 2 3	√	—	√
d.	4 1 2 3	√	—	√
e.	2 1 3 4	—	√	—
f.	2 1 4 3	—	√	√
g.	2 4 1 3	—	—	—
h.	4 2 1 3	—	—	√
i.	3 1 2 4	—	√	—
j.	3 1 4 2	—	—	—
k.	3 4 1 2	√	√	√
l.	4 3 1 2	√	√	√
m.	1 3 2 4	√	√	√
n.	1 3 4 2	√	√	√
o.	1 4 3 2	√	√	√
p.	4 1 3 2	√	√	√
q.	2 3 1 4	—	√	—
r.	2 3 4 1	√	√	√
s.	2 4 3 1	√	√	√
t.	4 2 3 1	√	√	√
u.	3 2 1 4	—	√	—
v.	3 2 4 1	—	√	—
w.	3 4 2 1	√	√	√
x.	4 3 2 1	√	√	√

CCG is not permutation complete

- Can we now generate all 4! = 24 permutations? No

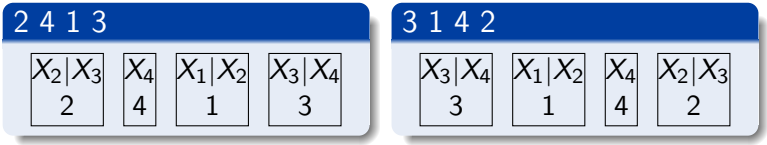


- 4 words $\longrightarrow$ 22 CCG permutations

		NP	NP	VP
		Cinque (2005)	Nchare (2012)	Abels (2016)
a.	1 2 3 4	√	√	√
b.	1 2 4 3	√	√	√
c.	1 4 2 3	√	—	√
d.	4 1 2 3	√	—	√
e.	2 1 3 4	—	√	—
f.	2 1 4 3	—	√	√
g.	2 4 1 3	—	—	—
h.	4 2 1 3	—	—	√
i.	3 1 2 4	—	√	—
j.	3 1 4 2	—	—	—
k.	3 4 1 2	√	√	√
l.	4 3 1 2	√	√	√
m.	1 3 2 4	√	√	√
n.	1 3 4 2	√	√	√
o.	1 4 3 2	√	√	√
p.	4 1 3 2	√	√	√
q.	2 3 1 4	—	√	—
r.	2 3 4 1	√	√	√
s.	2 4 3 1	√	√	√
t.	4 2 3 1	√	√	√
u.	3 2 1 4	—	√	—
v.	3 2 4 1	—	√	—
w.	3 4 2 1	√	√	√
x.	4 3 2 1	√	√	√

CCG is not permutation complete

- Can we now generate all 4! = 24 permutations? No

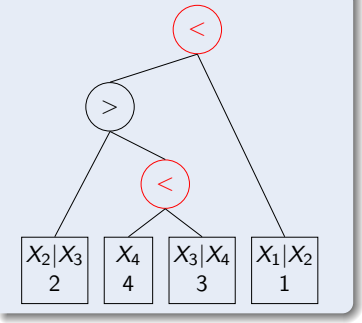


- 4 words $\longrightarrow$ 22 CCG permutations
- n words $\longrightarrow$? CCG permutations

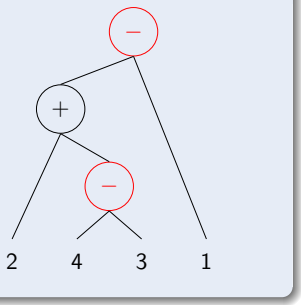
		NP	NP	VP
		Cinque (2005)	Nchare (2012)	Abels (2016)
a.	1 2 3 4	√	√	√
b.	1 2 4 3	√	√	√
c.	1 4 2 3	√	—	√
d.	4 1 2 3	√	—	√
e.	2 1 3 4	—	√	—
f.	2 1 4 3	—	√	√
g.	2 4 1 3	—	—	—
h.	4 2 1 3	—	—	√
i.	3 1 2 4	—	√	—
j.	3 1 4 2	—	—	—
k.	3 4 1 2	√	√	√
l.	4 3 1 2	√	√	√
m.	1 3 2 4	√	√	√
n.	1 3 4 2	√	√	√
o.	1 4 3 2	√	√	√
p.	4 1 3 2	√	√	√
q.	2 3 1 4	—	√	—
r.	2 3 4 1	√	√	√
s.	2 4 3 1	√	√	√
t.	4 2 3 1	√	√	√
u.	3 2 1 4	—	√	—
v.	3 2 4 1	—	√	—
w.	3 4 2 1	√	√	√
x.	4 3 2 1	√	√	√

CCG Trees ≈ Separating Trees (Bose et al., 1998)

CCG Tree

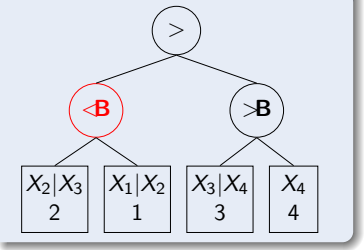


Separating Tree

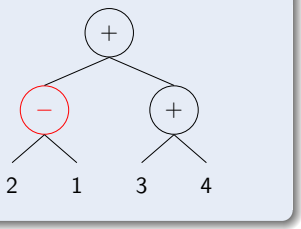


CCG Trees ≈ Separating Trees (Bose et al., 1998)

CCG Tree



Separating Tree



Main Finding

Only subset of permutations are possible

CCG Trees $\leftrightarrow$ Separating Trees $\leftrightarrow$ **Separable Permutations**

(Avis and Newborn, 1981; Shapiro and Stephens, 1991; West, 1996; Wu, 1996)

Main Finding

Only subset of permutations are possible

CCG Trees $\leftrightarrow$ Separating Trees $\leftrightarrow$ **Separable Permutations**

(Avis and Newborn, 1981; Shapiro and Stephens, 1991; West, 1996; Wu, 1996)

Number of allowed permutations is Large Schröder Number $\ll$ factorial

$$S_n = S_{n-1} + \sum_{k=0}^{n-1} S_k S_{n-k-1}$$

Main Finding

Only subset of permutations are possible

CCG Trees $\leftrightarrow$ Separating Trees $\leftrightarrow$ **Separable Permutations**

(Avis and Newborn, 1981; Shapiro and Stephens, 1991; West, 1996; Wu, 1996)

Number of allowed permutations is Large Schröder Number $\ll$ factorial

$$S_n = S_{n-1} + \sum_{k=0}^{n-1} S_k S_{n-k-1}$$

Applications in Machine Translation

- Stanojević and Sima'an (2015, EMNLP) preordering for SMT
- Wang et al. (2021, NeurIPS) neural MT

Conclusion

- Language Universals in LLMs:
  - Depend on the amount of training data for each language.
  - Parameter count helps but doesn't solve everything.
  - Model accuracy (perplexity, ranking, etc.) does not explain anything.
- Language Universals in Grammars:
  - Explains why universals exist.
  - Makes predictions about word orders.
- Path for bridging the LLM/Grammar divide:
  - LLMs with stronger linguistic inductive biases. (pretraining, tree architectures, BabyLM)
  - Treat LLM like a Grammar – have formal analysis with linguistic predictions.

Acknowledgment

Co-authors

John Hale

Mark Steedman

Expert linguists

András Kornai

Vera Lee-Schoenfeld

Tibor Laczkó

Ricardo Etxepare

Asya Pereltsvaig

Thank you.

Bibliography I

Abels, K. (2016). The fundamental left–right asymmetry in the Germanic verb cluster. The Journal of Comparative Germanic Linguistics, 19:179–220.

Avis, D. and Newborn, M. (1981). On pop-stacks in series. Utilitas Mathematica, 19:129–140.

Biberauer, T. (2017). The Final-over-Final Condition and Particles. In Sheehan et al. (2017), chapter 9.

Biberauer, T., Holmberg, A., and Roberts, I. (2014). A syntactic universal and its consequences. Linguistic Inquiry, 45(2):169–225.

Bose, P., Buss, J. F., and Lubiw, A. (1998). Pattern matching for permutations. Inf. Process. Lett., 65(5):277–283.

Chomsky, N. (1965). Aspects of the Theory of Syntax. MIT Press.

Cinque, G. (2005). Deriving Greenberg's universal 20 and its exceptions. Linguistic Inquiry, 36:315–332.

Clem, E. (2022). Disharmony and the final-over-final condition in amahuaca. Linguistic Inquiry, 53(4):809–822.

Contreras Kallens, P., Kristensen-McLachlan, R. D., and Christiansen, M. H. (2023). Large language models demonstrate the potential of statistical learning in language. Cognitive Science, 47(3):e13256. Letter to the Editor.

Hahn, M. (2020). Theoretical limitations of self-attention in neural sequence models. Transactions of the Association for Computational Linguistics, 8:156–171.

Hao, Y., Angluin, D., and Frank, R. (2022). Formal language recognition by hard attention transformers: Perspectives from circuit complexity. Transactions of the Association for Computational Linguistics, 10:800–810.

Hawkins, J. A. (2014). Cross-linguistic variation and efficiency. Oxford linguistics. Oxford University Press.

Holmberg, A. (2000). Deriving OV order in Finnish. In Svenonius, P., editor, The derivation of VO and OV, Linguistik aktuell: v. 31. John Benjamins Publishing CompanyBenjamins.

Lindemann, M., Koller, A., and Titov, I. (2024). Strengthening structural inductive biases by pre-training to perform syntactic transformations. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 11558–11573, Miami, Florida, USA. Association for Computational Linguistics.

Linzen, T. and Baroni, M. (2021). Syntactic structure from deep learning. Annual Review of Linguistics, 7(1):195–212.

Murty, S., Sharma, P., Andreas, J., and Manning, C. D. (2023). Pushdown layers: Encoding recursive structure in transformer language models. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 3233–3247.

Bibliography II

Nchare, A. L. (2012). The Grammar of Shupamem. PhD thesis, New York University.

Papadimitriou, I. and Jurafsky, D. (2023). Injecting structural hints: Using language models to study inductive biases in language learning. In Bouamor, H., Pino, J., and Bali, K., editors, Findings of the Association for Computational Linguistics: EMNLP 2023, pages 8402–8413, Singapore. Association for Computational Linguistics.

Sartran, L., Barrett, S., Kuncoro, A., Stanojević, M., Blunsom, P., and Dyer, C. (2022). Transformer grammars: Augmenting transformer language models with syntactic inductive biases at scale. Transactions of the Association for Computational Linguistics, 10:1423–1439.

Shapiro, L. and Stephens, A. B. (1991). Bootstrap percolation, the Schroder numbers, and the *N*-kings problem. SIAM Journal on Discrete Mathematics, 4(2):275–280.

Sheehan, M., Biberauer, T., Roberts, I., and Holmberg, A. (2017). The Final-over-Final Condition: A Syntactic Universal. MIT Press.

Stanojević, M. and Sima'an, K. (2015). Reordering Grammar Induction. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 44–54, Lisbon, Portugal. Association for Computational Linguistics.

Strobl, L., Merrill, W., Weiss, G., Chiang, D., and Angluin, D. (2024). What formal languages can transformers express? a survey. Transactions of the Association for Computational Linguistics, 12:543–561.

Wang, B., Lapata, M., and Titov, I. (2021). Structured reordering for modeling latent alignments in sequence transduction. In Thirty-Fifth Conference on Neural Information Processing Systems.

Weiss, G., Goldberg, Y., and Yahav, E. (2021). Thinking like transformers. In International Conference on Machine Learning, pages 11080–11090. PMLR.

West, J. (1996). Generating trees and forbidden subsequences. Discrete Mathematics, 157(1-3):363–374.

Wu, D. (1996). A polynomial-time algorithm for statistical machine translation. In Proceedings of the 34th Annual Meeting on Association for Computational Linguistics, ACL '96, pages 152–158, Stroudsburg, PA, USA. Association for Computational Linguistics.