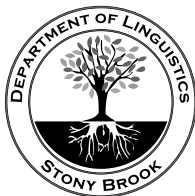


# Strict Locality in Syntax

Kenneth Hanson  
Stony Brook University



SYNC 23  
March 4, 2023



# Introduction (1)

Linguistic patterns belong to very simple classes of formal languages, known as **subregular** languages (Heinz 2018; Graf 2022a).

The **strictly local (SL)** languages are used to model finitely bounded dependencies.

- In phonology: local phonotactics
- In syntax: lexical (category) selection

# Introduction (2)

**This presentation:** there are additional SL phenomena in syntax beyond selection. Together, these instantiate essentially the full range of formal patterns within SL.

- Lexical selection – branching and looping paths
- Functional hierarchies – linear order, optionality
- Adjunct ordering – linear order, optionality, iteration

I use a formalism based on **command strings (c-strings)** in order to enable a direct comparison with phonology, and visualizations using **finite state automata**.

**Proposal:** SL computations are the basis for linguistic structure building across domains.

# Roadmap

1. Introduction to SL
  - Examples from phonology
2. Generalizing SL to trees using c-strings
  - Dependency trees
  - C-strings
3. SL in syntax:
  - Lexical selection
  - Functional hierarchies
  - Adjunct ordering
4. Beyond local dependencies

# Strictly Local Languages (1)

**Defining characteristic:** a string is well-formed if all of its substrings (of some fixed length) are well-formed

**SL-k:** SL for substrings of length  $k$

**Example: CV Alternation (SL-2)**

$\Sigma = \{C, V\}$      $k = 2$      $G = \{\$C, \$V, CV, VC, C\$, V\$\}$

	Word	Substrings ( $k=2$ )
✓	\$CVCVC\$	\$C, CV, VC, CV, VC, C\$
✓	\$VCV\$	\$V, VC, CV, V\$
✗	\$CVCCV\$	\$C, CV, VC, <b>CC</b> , CV, V\$
✗	\$VCVV\$	\$V, VC, CV, <b>VV</b> , V\$

# Strictly Local Languages (2)

## Example: Japanese phonotactics (SL-2)

Syllable template: (C) (j) V (N)

Example words: aoi, kotowaza, sjunkan

$$\Sigma = \{C, j, V, N\} \quad k = 2$$

$$G = \left\{ \begin{array}{ccccc} \$C & & & VC & NC \\ \$j & Cj & & Vj & Nj \\ \$V & CV & jV & VV & NV \\ & & & VN & \\ & & & V\$ & N\$ \end{array} \right\}$$

Note: geminates are omitted. The grammar with geminates is also SL-2.

# Strictly Local Languages (2)

## Example: Japanese phonotactics (SL-2)

Syllable template: (C) (j) V (N)

Example words: aoi, kotowaza, sjunkan

$$\Sigma = \{C, j, V, N\} \quad k = 2$$

$$G = \left\{ \begin{array}{cccccc} \$C & & & & VC & NC \\ \$j & Cj & & & Vj & Nj \\ \$V & CV & jV & & VV & NV \\ & & & & VN & \\ & & & & V\$ & N\$ \end{array} \right\}$$

\$ s j u n k a n \$

Note: geminates are omitted. The grammar with geminates is also SL-2.

# Strictly Local Languages (2)

## Example: Japanese phonotactics (SL-2)

Syllable template: (C) (j) V (N)

Example words: aoi, kotowaza, sjunkan

$$\Sigma = \{C, j, V, N\} \quad k = 2$$

$$G = \left\{ \begin{array}{ccccc} \$C & & & VC & NC \\ \$j & Cj & & Vj & Nj \\ \$V & CV & jV & VV & NV \\ & & & VN & \\ & & & V\$ & N\$ \end{array} \right\}$$

\$ s j u n k a n \$

Note: geminates are omitted. The grammar with geminates is also SL-2.



# Strictly Local Languages (2)

## Example: Japanese phonotactics (SL-2)

Syllable template: (C) (j) V (N)

Example words: aoi, kotowaza, sjunkan

$$\Sigma = \{C, j, V, N\} \quad k = 2$$

$$G = \left\{ \begin{array}{ccccc} \$C & & & VC & NC \\ \$j & \text{Cj} & & Vj & Nj \\ \$V & CV & jV & VV & NV \\ & & & VN & \\ & & & V\$ & N\$ \end{array} \right\}$$

\$ s j u n k a n \$

Note: geminates are omitted. The grammar with geminates is also SL-2.

# Strictly Local Languages (2)

## Example: Japanese phonotactics (SL-2)

Syllable template: (C) (j) V (N)

Example words: aoi, kotowaza, sjunkan

$$\Sigma = \{C, j, V, N\} \quad k = 2$$

$$G = \left\{ \begin{array}{cc} \$C & VC \quad NC \\ \$j & Cj \quad Vj \quad Nj \\ \$V & CV \quad jV \quad VV \quad NV \\ & VN \\ & V\$ \quad N\$ \end{array} \right\}$$

\$ s j u n k a n \$

Note: geminates are omitted. The grammar with geminates is also SL-2.

# Strictly Local Languages (2)

## Example: Japanese phonotactics (SL-2)

Syllable template: (C) (j) V (N)

Example words: aoi, kotowaza, sjunkan

$$\Sigma = \{C, j, V, N\} \quad k = 2$$

$$G = \left\{ \begin{array}{ccccc} \$C & & & VC & NC \\ \$j & Cj & & Vj & Nj \\ \$V & CV & jV & VV & NV \\ & & & \text{VN} & \\ & & & V\$ & N\$ \end{array} \right\}$$

\$ s j u n k a n \$

Note: geminates are omitted. The grammar with geminates is also SL-2.

# Strictly Local Languages (2)

## Example: Japanese phonotactics (SL-2)

Syllable template: (C) (j) V (N)

Example words: aoi, kotowaza, sjunkan

$$\Sigma = \{C, j, V, N\} \quad k = 2$$

$$G = \left\{ \begin{array}{ccccc} \$C & & & VC & \text{NC} \\ \$j & Cj & & Vj & Nj \\ \$V & CV & jV & VV & NV \\ & & & VN & \\ & & & V\$ & N\$ \end{array} \right\}$$

\$ s j u n k a n \$

Note: geminates are omitted. The grammar with geminates is also SL-2.

# Strictly Local Languages (2)

## Example: Japanese phonotactics (SL-2)

Syllable template: (C) (j) V (N)

Example words: aoi, kotowaza, sjunkan

$$\Sigma = \{C, j, V, N\} \quad k = 2$$

$$G = \left\{ \begin{array}{ccccc} \$C & & & VC & NC \\ \$j & Cj & & Vj & Nj \\ \$V & CV & jV & VV & NV \\ & & & VN & \\ & & & V\$ & N\$ \end{array} \right\}$$

\$ s j u n k a n \$

Note: geminates are omitted. The grammar with geminates is also SL-2.

# Strictly Local Languages (2)

## Example: Japanese phonotactics (SL-2)

Syllable template: (C) (j) V (N)

Example words: aoi, kotowaza, sjunkan

$$\Sigma = \{C, j, V, N\} \quad k = 2$$

$$G = \left\{ \begin{array}{ccccc} \$C & & & VC & NC \\ \$j & Cj & & Vj & Nj \\ \$V & CV & jV & VV & NV \\ & & & \text{VN} & \\ & & & V\$ & N\$ \end{array} \right\}$$

\$ s j u n k a n \$

Note: geminates are omitted. The grammar with geminates is also SL-2.

# Strictly Local Languages (2)

## Example: Japanese phonotactics (SL-2)

Syllable template: (C) (j) V (N)

Example words: aoi, kotowaza, sjunkan

$$\Sigma = \{C, j, V, N\} \quad k = 2$$

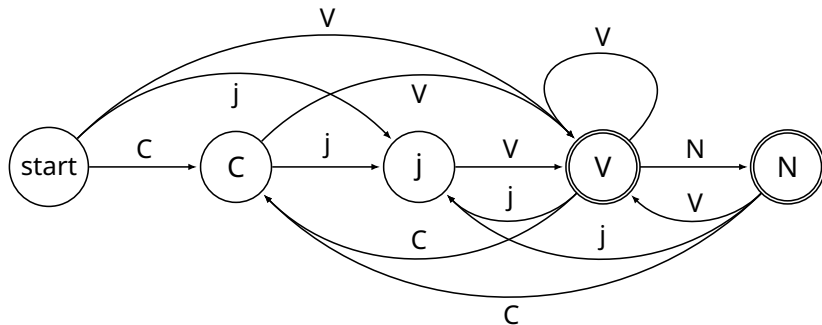
$$G = \left\{ \begin{array}{cccccc} \$C & & & & VC & NC \\ \$j & Cj & & & Vj & Nj \\ \$V & CV & jV & & VV & NV \\ & & & & VN & \\ & & & & V\$ & N\$ \end{array} \right\}$$

\$ s j u n k a n \$

Note: geminates are omitted. The grammar with geminates is also SL-2.

# Strictly Local Languages (3)

We can represent an SL grammar visually using a **finite-state automaton (FSA)**.

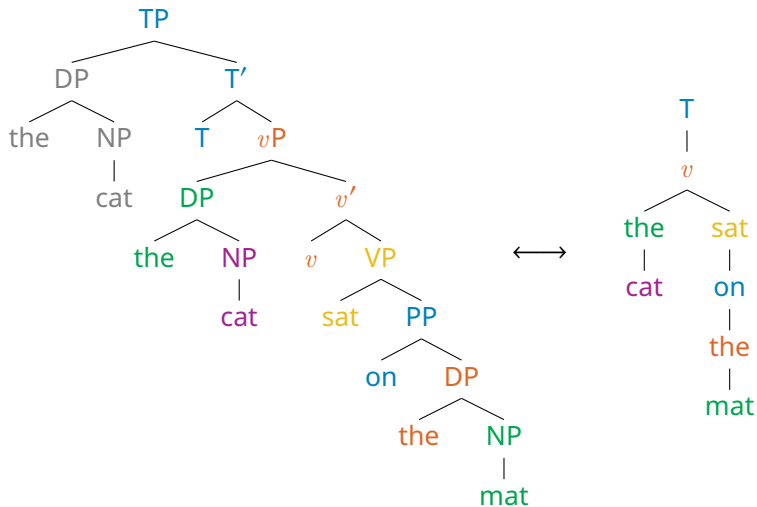


SL is a subclass of the languages expressible by FSAs.

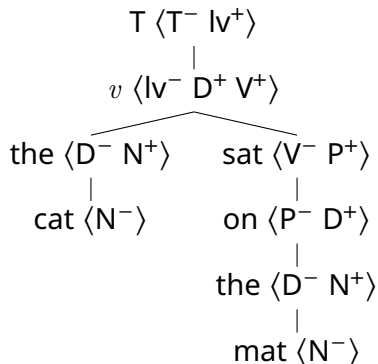




# Dependency Trees

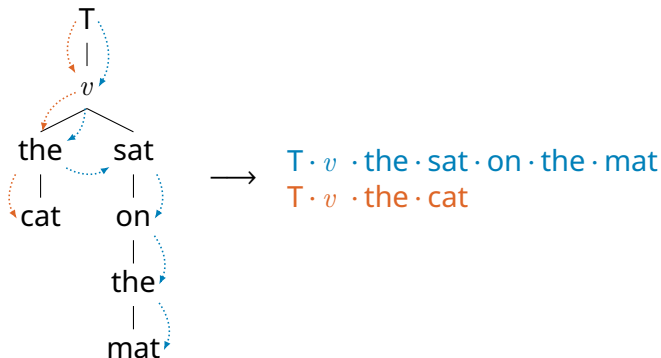


# Dependency Trees



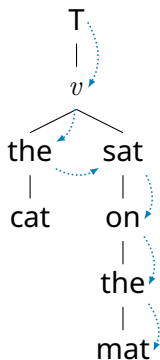
Note: the feature system is based on Minimalist Grammar (Stabler 1997).

# C-Strings



See Graf and Shafiei (2019) for details.

# Lexical Selection (1)



T · v · the · sat · on · the · mat

T	$\langle T^- \text{Iv}^+ \rangle$
v	$\langle \text{Iv}^- D^+ V^+ \rangle$
the	$\langle D^- N^+ \rangle$
sat	$\langle V^- P^+ \rangle$
on	$\langle P^- D^+ \rangle$
the	$\langle D^- N^+ \rangle$
mat	$\langle N^- \rangle$

# Lexical Selection (2)

## What the SL grammar looks like

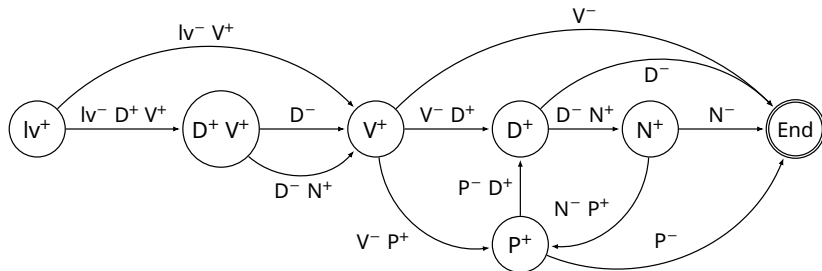
$$\Sigma = \{ \langle T^- \text{Iv}^+ \rangle, \langle \text{Iv}^- \text{V}^+ \rangle, \langle \text{Iv}^- \text{D}^+ \text{V}^+ \rangle, \langle \text{V}^- \rangle, \langle \text{V}^- \text{D}^+ \rangle, \langle \text{V}^- \text{P}^+ \rangle, \dots \}$$

$$k = 3$$

$$G = \left\{ \begin{array}{ccc} \dots & \langle \text{V}^- \text{D}^+ \rangle & \langle \text{D}^- \rangle \\ \dots & \langle \text{V}^- \text{D}^+ \rangle & \langle \text{D}^- \text{N}^+ \rangle \\ \dots & \langle \text{D}^- \text{N}^+ \rangle & \langle \text{N}^- \rangle \\ \dots & \langle \text{D}^- \text{N}^+ \rangle & \langle \text{N}^- \text{P}^+ \rangle \\ \dots & \langle \text{Iv}^- \text{D}^+ \text{V}^+ \rangle & \langle \text{D}^- \text{N}^+ \rangle \\ \langle \text{Iv}^- \text{D}^+ \text{V}^+ \rangle & \langle \text{D}^- \text{N}^+ \rangle & \langle \text{V}^- \text{D}^+ \rangle \\ \vdots & \vdots & \vdots \end{array} \right\}$$

# Lexical Selection (3)

## FSA Representation



# Functional Hierarchies (1)

## Example: English clausal hierarchy

$T < (\text{Neg}) < (\text{Perf}) < (\text{Prog}) < (\text{Pass}) < v < V$

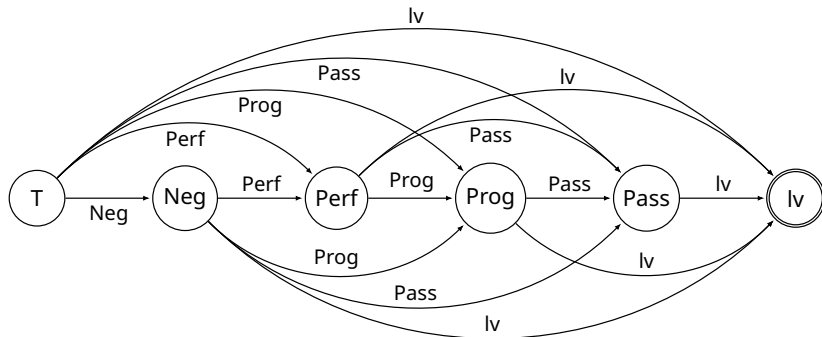
Ex. “We might<sub>T</sub> not<sub>Neg</sub> have<sub>Perf</sub> been<sub>Prog</sub> being<sub>Pass</sub> watched.”

This is also SL-2!

$$G = \left\{ \begin{array}{ccccc} T \text{ Neg} & & & & \\ T \text{ Perf} & \text{Neg Perf} & & & \\ T \text{ Prog} & \text{Neg Prog} & \text{Perf Prog} & & \\ T \text{ Pass} & \text{Neg Pass} & \text{Perf Pass} & \text{Prog Pass} & \\ T \text{ lv} & \text{Neg lv} & \text{Perf lv} & \text{Prog lv} & \text{Pass lv} \end{array} \right\}$$



# Functional Hierarchies (2)



# Adjunct Ordering (1)

Adjectives and adverbs often have a preferred order.

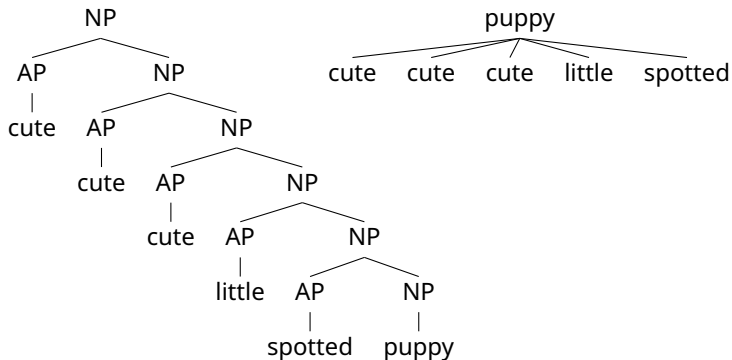
- |             |                                     |
|-------------|-------------------------------------|
| 1. opinion  | ✓ <i>cute little spotted puppy</i>  |
| 2. size     | ? <i>little cute spotted puppy</i>  |
| 3. shape    | ? <i>cute spotted little puppy</i>  |
| 4. age      | ?? <i>little spotted cute puppy</i> |
| 5. color    |                                     |
| 6. origin   |                                     |
| 7. material |                                     |
| 8. purpose  |                                     |

Items in the same group can be iterated.

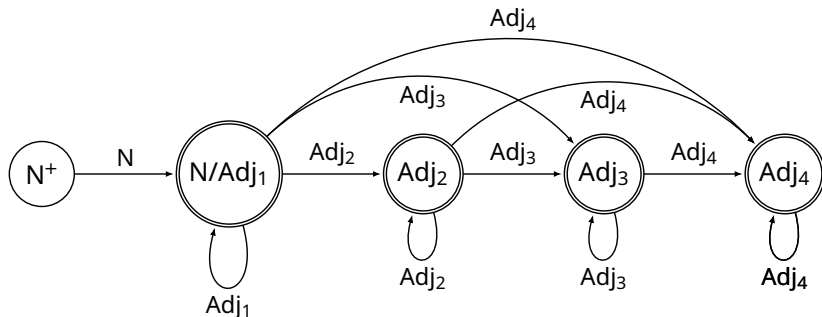
✓ *cute cute cute little spotted puppy*

## Adjunct Ordering (2)

## PS tree and dependency tree for "cute cute cute little spotted puppy"

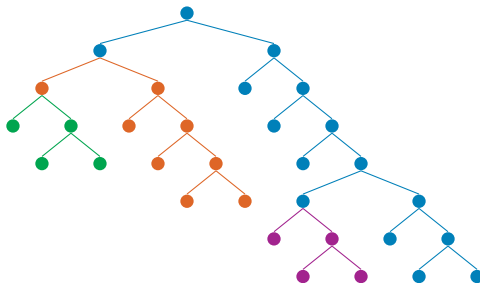


# Adjunct Ordering (3)



# Which c-strings do we use?

**Answer:** those that trace the **complement spine** of the tree, or of a subtree. See Graf and De Santo (2019) for details.



# What about long-distance phenomena?

Most long-distance phonological dependencies are in the class **tier-based strictly local (TSL)**, a generalization of SL in which non-salient items are ignored (Heinz 2018).

Most long-distance syntactic phenomena are TSL (or a close variant of TSL) over trees.

- Movement (Graf 2022b)
- Case (Vu et al. 2019)
- Anaphora and NPI licensing (Graf and Shafiei 2019)
- Agreement (work in progress)

# Conclusion

Functional hierarchies and adjunct hierarchies are unsurprising from a computational perspective — they are just further examples of SL patterns.

Syntax and phonology are very similar in computational terms, as highlighted by the c-string perspective.

SL computations are a good candidate for the basis of linguistic structure building.

# Thank you!

## Acknowledgments

- This work is partly funded by Thomas Graf's NSF CAREER Award: *Abstract Universals in (Morpho)Syntax: Computational Characterizations and Empirical Implications*.
- Thanks to Thomas Graf and the Stony Brook Mathematical Linguistics Reading Group for comments and discussion.



# References

- Graf, Thomas (2022a). "Subregular linguistics: bridging theoretical linguistics and formal grammar". In: *Theoretical Linguistics* 48.3-4. DOI: doi:10.1515/tl-2022-2037.
- (2022b). "Typological implications of tier-based strictly local movement". In: *Proceedings of the Society for Computation in Linguistics 2022*.
- Graf, Thomas and Aniello De Santo (2019). "Sensing Tree Automata as a Model of Syntactic Dependencies". In: *Proceedings of the 16th Meeting on the Mathematics of Language*. Toronto, Canada: Association for Computational Linguistics.
- Graf, Thomas and Nazila Shafiei (2019). "C-command dependencies as TSL string constraints". In: *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*.
- Heinz, Jeffrey (2018). "The computational nature of phonological generalizations". In: *Phonological Typology, Phonetics and Phonology*.
- Stabler, Edward P. (1997). "Derivational minimalism". In: *Logical Aspects of Computational Linguistics*. Springer.
- Vu, Mai Ha et al. (2019). "Case assignment in TSL syntax: A case study". In: *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*.