C-command dependencies as TSL string constraints

Thomas Graf and Nazila Shafiei
Stony Brook University

Contact Information:
Department of Linguistics
Stony Brook University
Stony Brook, NY 11794-4376

Email:
mail@thomasgraf.net
nazila.shafiei@stonybrook.edu

1 Introduction

- Phonological and morphological dependencies belong to the **subregular class IO-TSL**. (Kaplan & Kay 1994, Karttunen et al. 1992)
- With the right representation, syntactic dependencies (binding, NPI licensing) are IO-TSL, too.
- "Right representation"?
  - c-command as primitive relation (cf. Frank and Shankar 2001)
  - tree dependencies translated to constraints on c[ommand]-strings

Main Claim: Syntactic dependencies are simple (subregular) given the right representation.

2 C-command Relations as Strings

- Tree dependencies converted to string dependencies via c[ommand]-strings
- **Intuition**: c-string of X **lists c-commanders** of X
- **Formally**: computed over dependency trees
  - *immediate c-string of X (ics)*: X + all left siblings of X
  - *c-string of X (cs)*: *ics*(X) + *cs*(mother of X)

Example 1: C-strings

To calculate the c-strings of a node: go left ← , and up ↑ in the dependency tree.

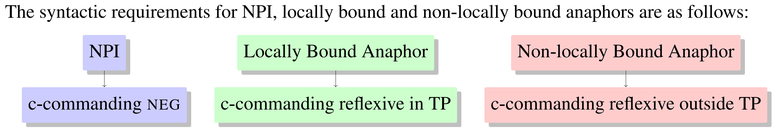
$$\begin{aligned} cs(\text{car}) &= ics(\text{car}) + cs(\text{this}) \\ &= \text{car} + cs(\text{this}) \\ &= \text{car} + ics(\text{this}) + cs(\text{likes}) \\ &= \text{car} + \text{this 's} + cs(\text{likes}) \\ &= \text{car} + \text{this 's} + ics(\text{likes}) + cs(\text{T}) \\ &= \text{car} + \text{this 's} + \text{likes} + cs(\text{T}) \\ &= \text{car} + \text{this 's} + \text{likes} + ics(\text{T}) + cs(\text{C}) \\ &= \text{car} + \text{this 's} + \text{likes} + \text{T} + cs(\text{C}) \\ &= \text{car} + \text{this 's} + \text{likes} + \text{T} + ics(\text{C}) \\ &= \text{car} + \text{this 's} + \text{likes} + \text{T} + \text{C} \\ &= \text{car this 's likes T C} \end{aligned}$$

Figure 1. C-string of *car* over dependency tree

C−strings give us the c-commanders for a specific node
↓
Only a finitely bounded number of elements is needed to determine well-formedness
↓
This ensures their IO-TSL definability

C-strings and Syntactic Dependencies

The syntactic requirements for NPI, locally bound and non-locally bound anaphors are as follows:

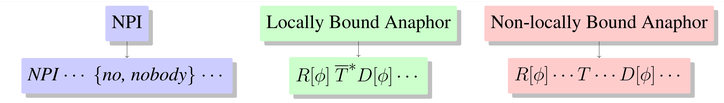


Example 2: C-strings for NPI and Reflexives Example Sentences

- NPI
  1. Nobody saw anybody. → *cs(NPI)*= anybody nobody see T C
  2. *Anybody saw nobody. → *cs(NPI)*= anybody see T C
- Locally Bound Reflexives
  1. John shaved himself. → *cs(NPI)*= himself John shave T C
  2. *John said that himself shaved Bill. → *cs(NPI)*= himself shave T that John say T C
- Non-Locally Bound Reflexives (e.g. Norwegian (Kiparsky 2002))
  1. *John shaved *sig*. → *cs(NPI)*= sig John shave T C
  2. John said that Bill shaved *sig*. → *cs(NPI)*= sig Bill shave T that John say T C

The well-formed c-strings for each constraint form a **regular string language**.

Example 3: Generalized Well-formed C-strings for NPIs and Reflexives



- $\overline{T}^*$ matches strings without any T-heads,
- $R[\phi]$ is a reflexive,
- $D[\phi]$ is a matching determiner.

3 Subregular Complexity

- C-string constraints are **also subregular**.
- C-string constraints are *input-output tier-based strictly local* (IO-TSL).
- IO-TSL is also an upper bound on phonotactic complexity (Graf & Mayer 2018).

IO-TSL

- IO-TSL is an extension of the *strictly local* (SL) languages.
- SL-n: well-formedness of string depends only on its substrings of length n
- TSL-n: project a tier that is SL-n

Example 4: German Final Devoicing is SL_2

- **Forbidden Bigrams**: {**z$**, **v$**, **d$**} ($ = word edge).
- *ra**d$** versus ^{OK}ra**t$** → **d$** is in the forbidden Grammar

Example 5: Samala Sibilant Harmony is TSL_2

- No string may contain sibilants that differ in anteriority
- Project tier of sibilants
- **Forbidden Bigrams**: all xy such that x and y differ in anteriority

The more information the tier projection may use, the more powerful the TSL-variant:

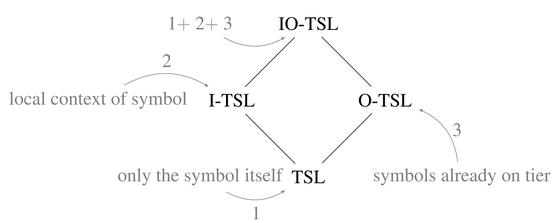


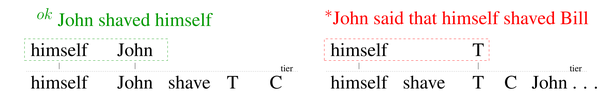
Figure 2. TSL classes by tier projection parameters

Example Grammars for Dependencies

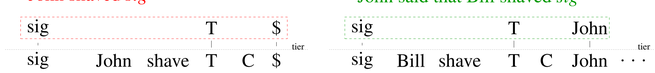
Example 6: Tier-Projections for NPI and Reflexives

- **NPI**
  1. Project the first symbol.
  2. Project an NPI-licensor if the previous tier-symbol is an NPI.
  3. Forbidden: **NPI $**

- **Locally Bound Reflexive**
  1. Project the first symbol.
  2. Project T or D[ϕ] if the previous tier-symbol is R[ϕ].
  3. Forbidden: **Refl T**



- **Non-locally Bound Reflexives**
  1. Project the first symbol.
  2. Project T if the previous tier-symbol is R[ϕ].
  3. Project D[ϕ] if the previous two tier-symbols are R[ϕ] T.
  4. Forbidden: **Refl T $**



4 Conclusion

- C-command dependencies are **subregular string constraints** over c-strings.
- The string constraints all fall within the class IO-TSL.
- The complexity of many syntactic phenomena thus is comparable to dependencies in phonology and morphology.

References

• **Frank, Robert, and K. Vijay-Shanker.** 2001. Primitive C-Command. *Syntax* 4 (3): 164-204. • **Graf, Thomas, and Connor Mayer.** 2018. Sanskrit n-Retroflexion is Input-Output Tier-Based Strictly Local. In *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology*: 151-160. • **Kaplan, Ronald M., and Martin Kay.** 1994. Regular models of phonological rule systems. *Computational linguistics* 20 (3): 331-378. • **Karttunen, Lauri, Ronald M. Kaplan, and Annie Zaenen.** 1992. Two-level morphology with composition. In *Proceedings of the 14th conference on Computational linguistics* 1: 141-148. • **Kiparsky, Paul.** 2002. *Disjoint reference and the typology of pronouns*. • **Shieber, Stuart M.** 1985. Evidence against the context-freeness of natural language. *Philosophy, Language, and Artificial Intelligence*: 79-89.