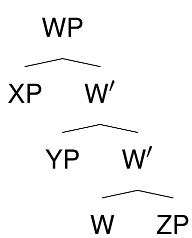
Ordered dependencies **Gautam Ottur (University of Göttingen)**

Much early generative work sought to identify the underlying “shapes” of syntactic structure (e.g., the X-bar schema, c-command, subjacency). The Minimalist Program has shifted focus toward specifying the minimal mechanisms required to generate syntactic structure, displacing the burden of capturing the shape of syntax to stipulated constructs such as terms and phases (Chomsky 1995; Chomsky 2000; Chomsky 2001). While this eliminationist approach has been productive in some ways, it introduces new gaps into the theory which circuitously require new subtheories to account for. As an example, assuming that syntactic structure is generated by recursively nesting (unordered) set-theoretic objects necessitates a *sui generis* subtheory for labeling/projection. Much of this kind of conceptual bloat ultimately arises because the standard theory views constituency as primitive, and hierarchy as a side effect of recursion.

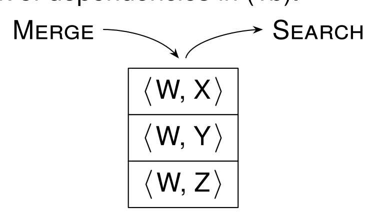
In this poster, I discuss the advantages of a *diatomic* model of syntax, i.e. one where syntactic objects (inputs for syntactic operations) are always atomic. Building on precedents (e.g., Brody 2000; Jayaseelan 2008; Svenonius 2016; Adger 2025), I explore a version of syntax built entirely on asymmetric relations between atomic elements, i.e. dependencies. However, in contrast to most previous diatomic approaches, this model explicitly derives hierarchy from the order in which dependencies are produced and stored in a memory submodule (cf. Krivochen 2023).

Within this system, dependencies are generated by Merge and pushed to a memory stack, which is filled bottom-up. Hierarchy thereby becomes an intrinsic property of storage rather than a derived effect of set inclusion. Dependencies generated by Internal Merge and Agree use a Search algorithm which simply scans the memory stack in the opposite order, i.e. top-down. This renders minimality: the first suitable goal encountered by Search is necessarily the most recent one added to the stack. As an example of how all this would work, the tree structure in (1a) would be underlyingly stored as the stack of dependencies in (1b).

(1) a.



b.



I discuss how the ordered dependencies model offers enriched structural descriptions that are not available in the standard theory, and speculate on how widely-assumed theoretical constructs (e.g. c-command, phases, strong islands) may consequently be reformulated more naturally in this system.

References.

Adger, David. 2025. *Mereological syntax: phrase structure, cyclicity, and islands* (Linguistic Inquiry Monographs). Cambridge, MA: The MIT Press.

Brody, Michael. 2000. Mirror Theory: Syntactic Representation in Perfect Syntax. *Linguistic Inquiry* 31(1). 29–56.

Chomsky, Noam. 1995. *The Minimalist Program*. Cambridge, MA: MIT Press.

Chomsky, Noam. 2000. Minimalist inquiries: The framework. *Step by step: Essays on minimalist syntax in honor of Howard Lasnik*. 89–155.

Chomsky, Noam. 2001. Derivation by phase. In Michael Kenstowicz (ed.), *Ken Hale: A life in language*, vol. 18, 1–52. Cambridge, MA: MIT Press.

Jayaseelan, K.A. 2008. Bare Phrase Structure and Specifier-less Syntax. *Biolinguistics* 2(1). 087–106.

Krivochen, Diego Gabriel. 2023. The search for Minimal Search: A graph-theoretic approach. *Biolinguistics* 17(1). e9793.

Svenonius, Peter. 2016. Spans and words. In Daniel Siddiqi & Heidi Harley (eds.), *Morphological metatheory* (Linguistik Aktuell/Linguistics Today 229), 201–222. Amsterdam: John Benjamins Publishing Company.