

### Comparing methods of tree-construction across mildly context-sensitive formalisms

A central constraint on minimalist derivations is cyclicity, as instantiated in the extension condition (**Ext**), which permits trees to grow only at (or near) the root. A second key idea concerns the notion of *derivational state*: how much information about the derivational past can the applicability of a certain operation be contingent on? Conditions like Phase Impenetrability and Shortest Move, for example, can be understood as limiting the amount of information that conditions subsequent derivational steps; some such conditions result in a finite bound (**Fin**). Interestingly, **Ext** and **Fin** define points of variation across mildly context-sensitive grammar formalisms, systems that have been proposed to characterize the computational structure of syntactic derivations. On the one hand, Minimalist Grammars (MGs; Stabler, 2011) abide by **Ext**, making use of the standard minimalist operations of Merge and Move, as do Combinatory Categorical Grammars (CCGs; Steedman, 1996), whose derivations involve bottom-up concatenation of lexical items, and the closely related Linear Indexed Grammars (LIGs; Gazdar, 1988). In contrast, the adjoining operation of Tree Adjoining Grammars (TAGs; Joshi and Schabes, 1997) allows trees to grow “in the middle”, in violation of **Ext**. Turning to **Fin**, both MG and TAG operate with bounded derivational state, whereas CCG, with its unboundedly large categories, and LIG, with its stack-valued non-terminals, permit unbounded state. Our goal here is to use the pattern of wh-island-violating extraction in languages like Bulgarian to argue against the conjunction of **Ext** and **Fin**, and therefore against the viability of the standard version of the MG formalism. Adequately capturing this pattern requires abandoning either **Ext** (as in TAG) or **Fin** (as in CCG and LIG). The ability of TAG/LIG/CCG to capture this structural pattern is striking and intriguing given that they are strictly less powerful than MGs in weak generative capacity.

	<b>Ext</b>	<b>–Ext</b>
<b>Fin</b>	MG	TAG
<b>–Fin</b>	LIG, CCG	

Our empirical starting point is Bulgarian sentences like (1) (Rudin, 1988; Richards, 1997).

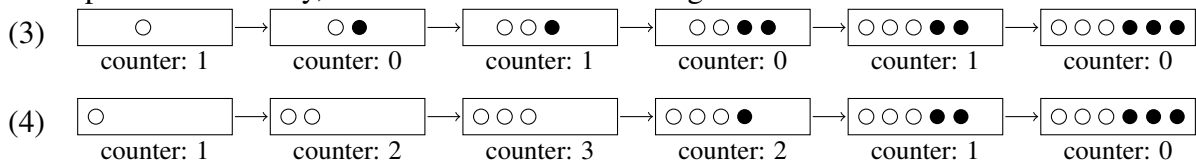
- (1) Koja kniga<sub>1</sub> te popita učitelja [ kogo<sub>2</sub> [ ubedi Ivan t<sub>2</sub> [ da publikiva t<sub>1</sub>]]]  
 which book you asked teacher who convinced Ivan to publish

“Which book did the teacher ask you who Ivan convinced to publish?”

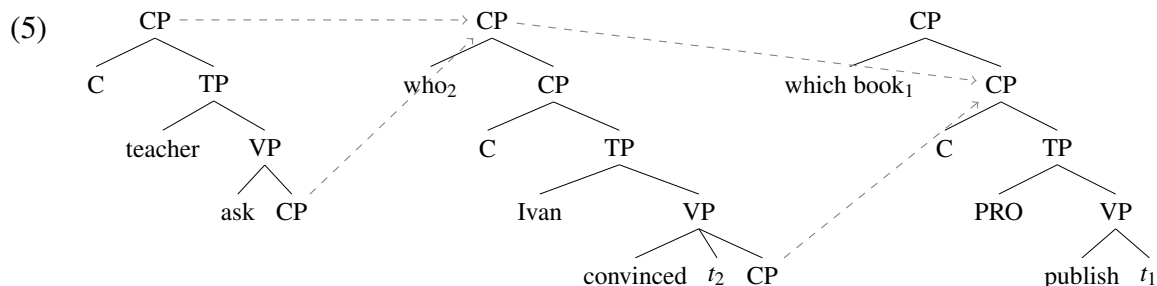
The significant point for our purposes is that this example exemplifies the abstract structure shown in (2a). We assume that the pattern can be extended without bound to (2b), (2c), and so on, modulo performance complications. To be clear, the point here is independent of the linear order of the words in the sentence, and relates only to the structural configuration we wish to assign to (1): (2a) describes a tree in which some subtree (corresponding to the innermost square brackets) contains two traces but neither of the corresponding wh-phrases.

- (2) a. [wh ... [wh [...t ...t ...]]]  
 b. [wh ... [wh ... [wh [...t ...t ...t ...]]]]  
 c. [wh ... [wh ... [wh ... [wh [...t ...t ...t ...t ...]]]]]

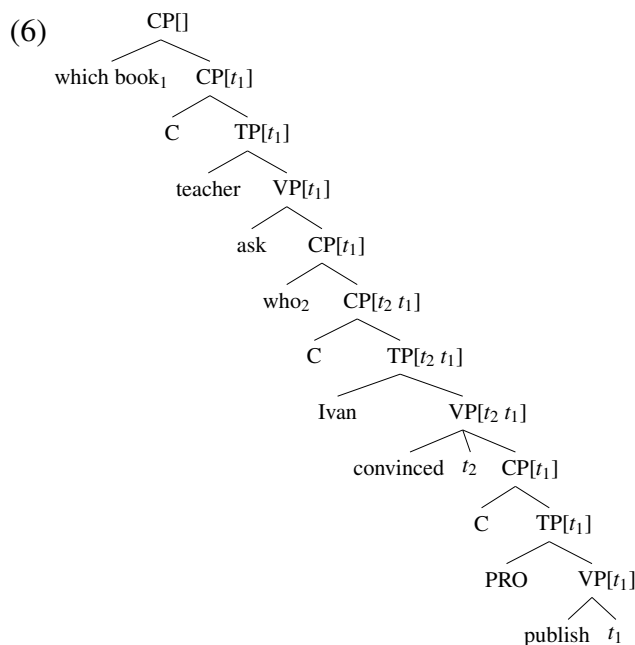
We consider two derivational strategies for generating this structural pattern. For illustration, consider mechanisms for generating a sequence of white and black pebbles whose first half contains only white pebbles ( $\approx$  traces) and whose second half contains the same number of black pebbles ( $\approx$  wh-phrases). One strategy starts in the middle and works outwards, alternating between white and black pebbles, as in (3). This way, a finite amount of memory suffices to track everything necessary to guide future placements of pebbles ( $\approx$  derivational operations); one needs to keep track of only the color of the last pebble, or equivalently a counter with a maximum value of one that tracks the number of unmatched white pebbles. Another strategy works from one end to the other, as in (4), placing (say) all the white pebbles and then all the black pebbles. This way, the information needed can grow without bound.



TAG generates the pattern in (2) using a version of the first strategy. Elementary trees introduce matched pairs of a wh-phrase and trace, as shown in (5). The fact that the trees are not constrained to grow only at one end ( $\neg$ **Ext**) allows the tree-building system to operate with a finite amount of memory (**Fin**). LIG, like its close relative CCG, generates the pattern in (2) using a version of the second strategy. Derivations construct trees from bottom to top (**Ext**), and therefore require an unbounded amount of memory to ensure that wh-phrases and traces are paired up ( $\neg$ **Fin**); see (6).



MGs construct trees bottom up, but the finite bound on their state means that after a certain number of traces have been generated it becomes impossible to retain the corresponding wh-phrase requirements. MGs can produce a certain variant of the pattern of interest (Gärtner and Michaelis, 2010), namely one where the number of wh-phrases is unbounded but the number of contiguous *clusters* of wh-phrases (i.e. wh-phrases that surface together at the edge of a single clause) remains bounded, by using one of the boundedly many units of derivational state to store each cluster. But in (2), where each wh-phrase constitutes its own cluster, there is no bound on the number of clusters. Note also that no operation excorporating a wh-phrase from a cluster is possible: this would require the recovery of information that had been abandoned when the cluster was formed in order to satisfy **Fin**.



Non-standard variants of MGs that are not restricted in this way include Graf (2012) (which rejects **Ext**) and Gärtner and Michaelis (2005) (which rejects **Fin**). Kobele and Michaelis (2005) showed that the latter has the power of a Turing machine. In contrast, LIG rejects **Fin** in a restricted way: the stack-structured memory generates wh-trace dependencies that are structurally nested, in accordance with the empirically-motivated Path Containment Condition (Pesetsky, 1982), allowing only a limited degree of deviation from the nesting pattern that we believe suffices for the non-nested examples reported in the literature. Furthermore, it turns out that the pattern of wh-trace dependencies predicted by TAG's  $\neg$ **Ext,Fin** system is closely analogous to that predicted by this stack-based **Ext,Fin** system (Joshi et al., 1990). This raises the question of how to import an analogous relaxation of **Fin** in MGs' derivational framework.

**Gärtner & Michaelis (2005)**. A note on the complexity of constraint interaction. • **Gärtner & Michaelis (2010)**. On the treatment of multiple-wh interrogatives in Minimalist Grammars. • **Graf, T. (2012)**. Movement-generalized Minimalist Grammars. • **Joshi et al. (1990)**. The convergence of mildly context-sensitive grammar formalisms. • **Kobele & Michaelis (2005)**. Two type-0 variants of Minimalist Grammars. • **Pesetsky (1982)**. *Paths and categories*. • **Richards (1997)**. *What moves where when in which language?*. • **Rudin (1988)**. On multiple questions and multiple wh-fronting. • **Stabler (2011)**. Computational perspectives on minimalism.