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

Tim Hunter and Bob Frank

November 6, 2020
NELS 51, UQAM

1 Overview: Fin and Ext as points of variation across derivational systems

Two dimensions:

- How do trees grow?

  Ext: from one end to the other, always building at the root

  ¬Ext: can grow/expand "in the middle" (roughly like late merge)

- Derivational state: how much information about the derivational past can the applicability of an operation be contingent upon?

  Fin: only a finite/bounded amount of information (maybe something like phases?)

  ¬Fin: no bound on the amount of information

Main claims:

- A pattern of extractions in languages like Bulgarian, which lack the wh-island constraint, is incompatible with the conjunction of **Ext** and **Fin**.
- Adequately capturing the relevant pattern requires abandoning either **Ext** or **Fin**; either one is sufficient.

Mildly context-sensitive grammar formalisms have been argued to have formal power appropriate for the characterization of natural language (Joshi, 1985; Joshi et al., 1990; Stabler, 2010).

- (limited) cross-serial dependencies
- constant growth property (semi-linearity)
- polynomial time processing

Yet despite their important similarities (particularly with respect to weak generative capacity), these formalisms do not all fall into the same categories with respect to these dimensions. Our goal today is to classify formalisms along these dimensions, and to understand how this classification relates to the ability of a formalism to treat a specific type of structure.

(1)

	Ext	**¬Ext**
Fin		
¬Fin		

2 The Bulgarian data, the "everyday minimalist" account, and unbounded derivational state

2.1 The data

Our empirical starting point is a subclass of Bulgarian multiple wh-questions, like those is (2) and (3) (Rudin, 1988; Richards, 1997).

(2) Koja kniga$_1$ te popita učitelja kogo$_2$ [ubedi Ivan t_2 da publikiva t_1]
which book you asked teacher who convinced Ivan to publish
"Which book did the teacher ask you who Ivan convinced to publish?"

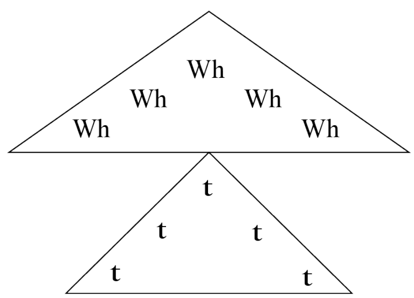
(3) Koj kontinent$_1$ te popita učitelja koj$_2$ [t_2 e otkril t_1]?
which continent you asked teacher who has discovered
"Which continent did the teacher ask you who discovered?"

These sentences exemplify the pattern shown in (4), which we assume generalizes to arbitrary numbers of wh-phrases, even if such examples are difficult to process (Miller and Chomsky 1963, but cf. Joshi et al. 2000):

(4) wh . . . wh . . . [. . . t . . . t . . .]

A couple of points of clarifications:

- This pattern is concerned with *structural configurations*, not *linear order*: what matters is that there is a constituent that includes the base positions of unboundedly many wh-phrases while excluding all of their ultimate landing sites. (This can alternatively be characterized as there being a point during the derivation at which unboundedly many wh-phrases have unchecked featural requirements.)



- The question of whether the wh-phrases and traces are organized in nested or crossing configurations (Pesetsky, 1982) is orthogonal to the question we explore here, though we return to this issue in the conclusion.

2.2 Some familiar/intuitive derivational strategies

An natural minimalist-style bottom-up (**Ext**-satisfying) strategy for deriving (3) would look roughly like this, where highlighting indicates phrases with unchecked featural requirements:

(5) a. discovered which-continent — unchecked: 1
b. who discovered which-continent — unchecked: 2
c. [$_{CP}$ who [$_{TP}$ t discovered which-continent]] — unchecked: 1
d. teacher ask you [$_{CP}$ who [$_{TP}$ t discovered which-continent] — unchecked: 1
e. [$_{CP}$ which-continent teacher ask you [$_{CP}$ who [$_{TP}$ t discovered t]]] — unchecked: 0

The configuration we care about is importantly different from (6), derivations of which *do not* require unbounded derivational state:

(6) Who$_1$ do you think t_1 wonders what$_2$ John bought t_2 yesterday?

a. John bought what yesterday — unchecked: 1

b. [$_{\text{CP}}$ what [$_{\text{TP}}$ John bought *t* yesterday]] — unchecked: 0

c. you think who wonders [$_{\text{CP}}$ what [$_{\text{TP}}$ John bought *t* yesterday]] — unchecked: 1

d. [$_{\text{CP}}$ who do you think *t* wonders [$_{\text{CP}}$ what [$_{\text{TP}}$ John bought *t* yesterday]]] — unchecked: 0

Note that imposing the PIC on the derivation in (5) and requiring successive cyclic movement does not change the upper limit on how many phrases must be “highlighted” during the derivation:

(7)

a. discovered which-continent — unchecked: 1

b. who discovered which-continent — unchecked: 2

c. [$_{\text{CP}}$ who [$_{\text{TP}}$ *t* discovered which-continent]] — unchecked: 1

d. [$_{\text{CP}}$ which-continent who [$_{\text{TP}}$ *t* discovered *t*]] — unchecked: 1

e. teacher ask you [$_{\text{CP}}$ which-continent who [$_{\text{TP}}$ *t* discovered *t*] — unchecked: 1

f. [$_{\text{CP}}$ which-continent teacher ask you [$_{\text{CP}}$ who [$_{\text{TP}}$ *t* discovered *t*]]] — unchecked: 0

The PIC constrains *where* the highlighted things are allowed to be, but not *how many* phrases can be highlighted at a single point. So, with or without the PIC, in step b, *all* of the derivation’s wh-phrases (here, two) have remaining unchecked featural requirements.

Consequently, as the pattern is extended, either of these two **Ext**-satisfying strategies ((5) and (7)) will require unbounded derivational state (i.e., ¬**Fin**).

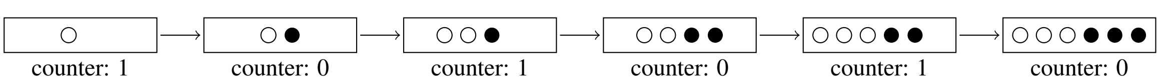
Is this the only conceivable approach?

3 The trade-off between Ext and Fin, abstractly

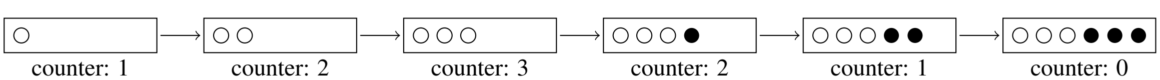
Suppose we need to arrange white and black pebbles in a line, so that there are an equal number of each, with all white pebbles preceding all black.

We can do this using one of two strategies:

(8) Work inside-out/outside-in (≈ ¬**Ext**), finite memory (**Fin**)



(9) Work from one end to the other (≈ **Ext**), unbounded memory (¬**Fin**)



The relationship between these two strategies is analogous to the relationship between context-free grammars (CFGs) and pushdown automata (PDAs):

(10) A CFG generates $a^n b^n$ inside-out/outside-in using finite derivational state, like (8) (≈ [**Fin**,¬**Ext**])

a. S → a X
X → S b
S → ε

b. S ⇒ a X ⇒ a S b ⇒ a a X b ⇒ a a S b b ⇒ a a a X b b ⇒ a a a S b b b ⇒ a a a b b b

(11) A PDA processes $a^n b^n$ from one end to the other using an unbounded stack-based memory to condition the derivation, like (9) (≈ [**Ext**,¬**Fin**])

a. $(\cdots, \cdots) \Rightarrow (\cdots\ \text{a}, \cdots\ \text{X})$ ("read an a, push an X")
$(\cdots, \cdots\ \text{X}) \Rightarrow (\cdots\ \text{b}, \cdots)$ ("read a b, pop an X")

b. (a, [X]) ⇒ (aa, [X X]) ⇒ (aaa, [X X X]) ⇒ (aaab, [X X]) ⇒ (aaabb, [X]) ⇒ (aaabbb, [ ])

4 The trade-off between Ext and Fin, concretely

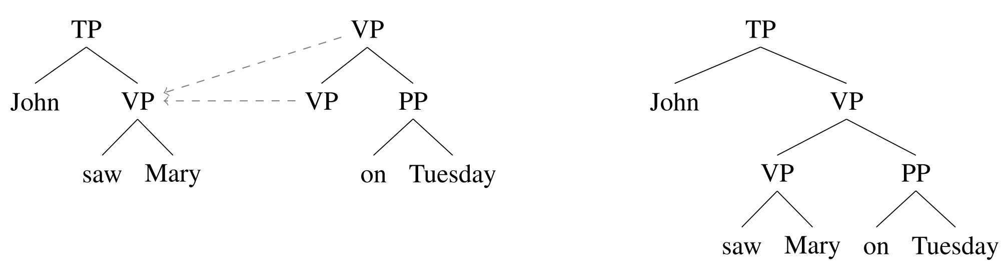
4.1 Tree adjoining grammar (TAG) (Joshi and Schabes, 1997)

(12)

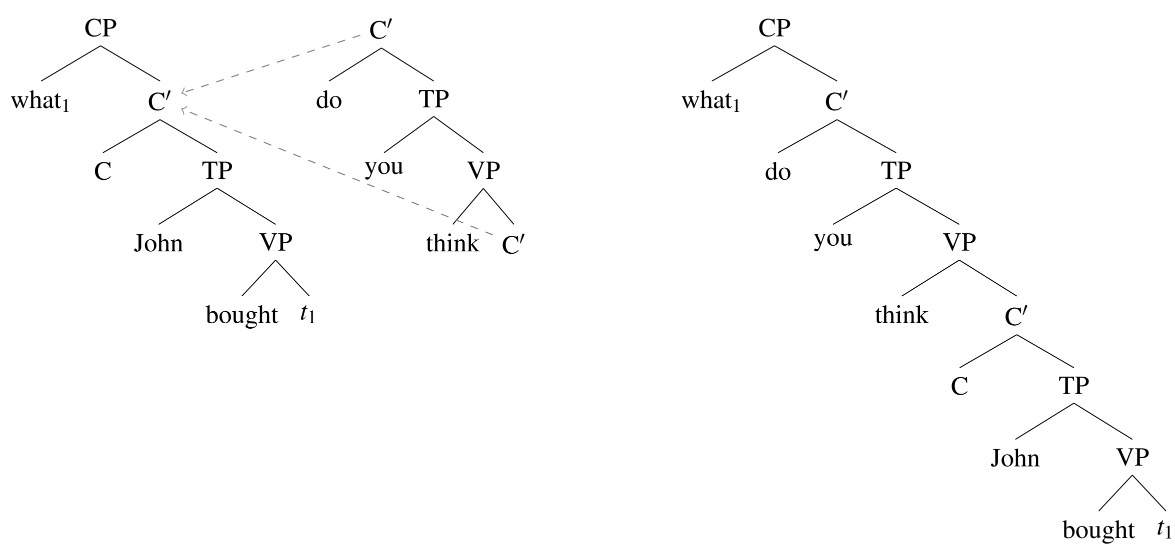
	Ext	**¬Ext**
Fin		TAG
¬Fin		

TAG's **¬Ext** adjoining mechanism is used to create unbounded long-distance dependencies.

(13)



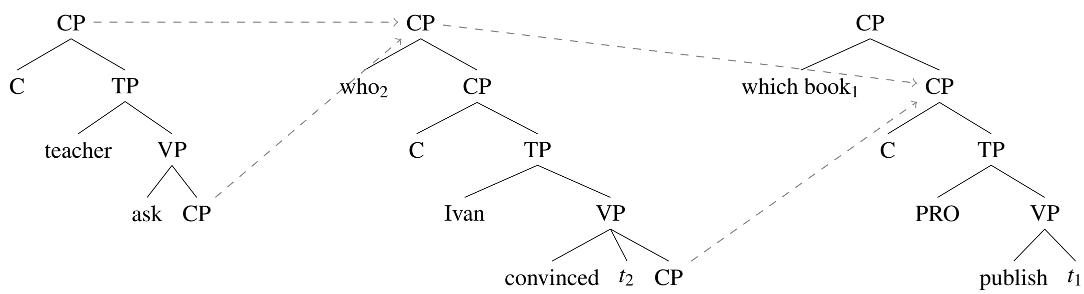
(14)



TAG, being relevantly similar to a certain kind of context-free tree grammar (Kepser and Rogers, 2011), can generate the crucial Bulgarian pattern using a form of the "pebble-pairing" strategy in (8).

- Elementary trees introduce matched pairs of a wh-phrase and trace.
- The fact that the trees are not constrained to grow only at one end (**¬Ext**) allows the tree-building system to operate with a finite amount of derivational state (**Fin**).

(15)



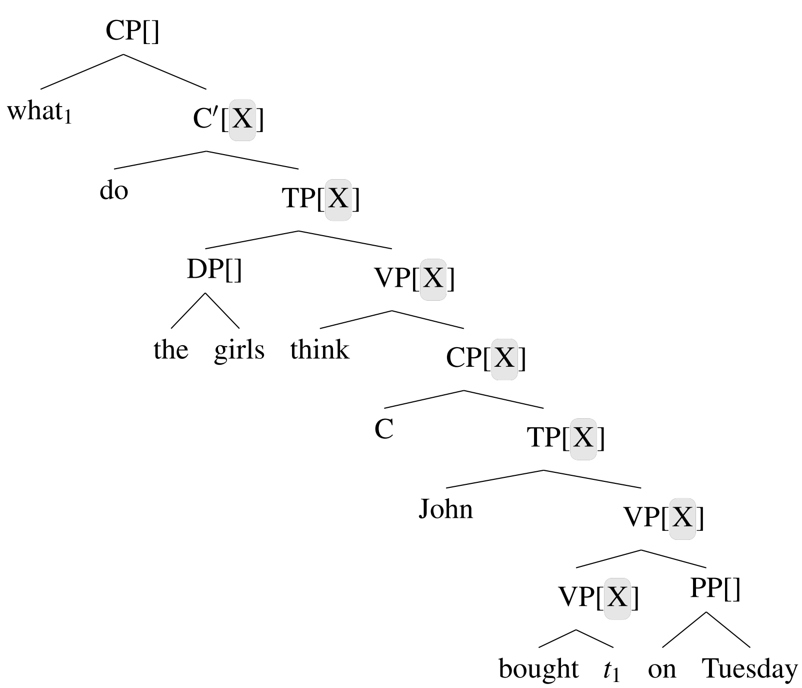
4.2 Linear Indexed Grammars (LIG) (Gazdar, 1988) and Combinatory Categorial Grammars (CCG) (Steedman, 1996)

(16)

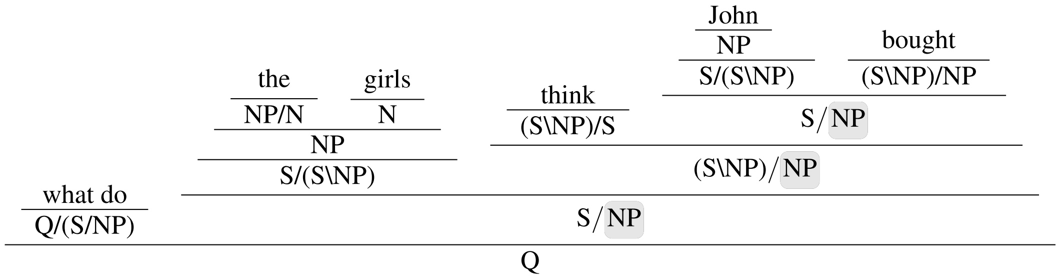
	Ext	**¬Ext**
Fin		TAG
¬Fin	LIG, CCG	

Linear Indexed Grammars (LIGs) add to CFGs the ability to store information in an unbounded stack at each node of the tree.

(17)

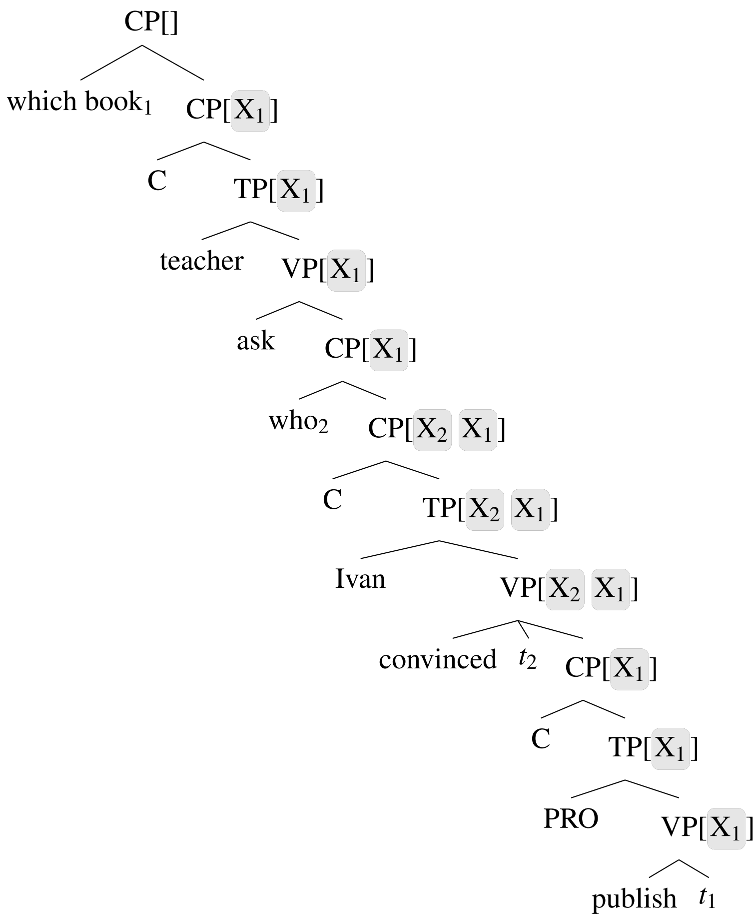


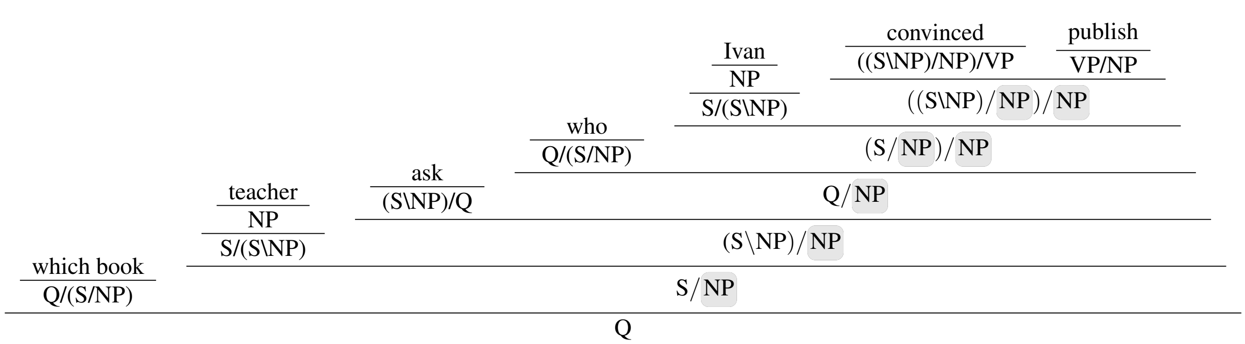
CCGs do something similar by allowing arbitrarily complex categories to be derived through function composition.

(18) 

LIG and CCG generate the crucial Bulgarian pattern using a tree-based version of the left-to-right pebble strategy in (9).

Derivations construct trees from bottom to top (**Ext**), and therefore require an unbounded amount of derivational state to ensure that wh-phrases and traces are paired up (¬**Fin**).

(19) 

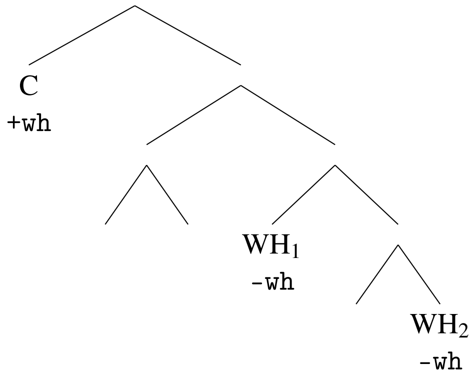
(20) 

5 Minimalist Grammars (MGs) (Stabler, 2011) vs. “everyday minimalism”

(21)

	Ext	**¬Ext**
Fin	MG	TAG
¬Fin	LIG, CCG	

(22)



A common assumption is that WH_1 prevents WH_2 from moving in this configuration, but beyond that versions of “Shortest Move” differ:

- Stabler’s MGs implement a simple conception of Shortest Move, according to which this configuration dooms the derivation (since WH_2 has been prevented from moving to its closest potential attractor).
  - This places a bound on stored information that can condition derivational operations (**Fin**).
  - Given **Ext** as well, the system will be unable to generate the Bulgarian *tree* pattern for the same reason that LIG with a bound on the stack cannot.
    (Even though the latter is weakly CF, and MGs are not.)
- Richards’ version within “everyday minimalism” says that WH_1 can move, which frees up WH_2 for subsequent attractors.
  - This involves unbounded storage (¬**Fin**).
  - Given **Ext** as well, the system looks something like LIG/CCG.

MG’s simple version of Shortest Move makes any form of multiple wh-movement look initially problematic.

(23) Koj$_1$ kŭde$_2$ misliš [če Boris iska [da kažeš [če šte otide t_1 t_2]]] ?
who where you think that Boris wants to you say that will go
Who do you think Boris wants you to say will go where?

This kind of derivation of (23) would involve maintaining unbounded derivational state:

(24) a. . . . wh . . .
b. . . . wh . . . wh . . .
c. wh wh [. . . *t* . . . *t* . . .]
d. . . . [wh wh [. . . *t* . . . *t* . . .]]
e. [wh . . . [wh [. . . *t* . . . *t* . . .]]]
f. [wh wh . . . [[. . . *t* . . . *t* . . .]]]

But cases like (23) can be treated by collapsing/clustering the wh-phrases that surface together at the edge of a single clause keeping a bound on the amount of state to be maintained (Grewendorf, 2001; Gärtner and Michaelis, 2010):

(25) a. . . . wh . . .

b. . . . wh . . . wh . . .

c. . . . wh wh . . . *t* . . .

d. wh wh [. . . *t* . . . *t* . . .]

e. . . . [wh wh [. . . *t* . . . *t* . . .]]

f. [wh wh . . . [[. . . *t* . . . *t* . . .]]]

The pattern in (4) that we have focused on is significant because it involves an unbounded number of "unclusterable" movers.

6 Conclusion

(26)

	Ext	**¬Ext**
Fin	MG	TAG
¬Fin	LIG, CCG	

Main claims:

- A pattern of extractions in languages like Bulgarian, which lack the wh-island constraint, is incompatible with the conjunction of **Ext** and **Fin**, and therefore incompatible with (standard versions of) the MG formalism.
- Adequately capturing the relevant pattern requires abandoning either **Ext** (as in TAG) or **Fin** (as in CCG and LIG).

Broader consequences:

- We aim to highlight ways of comparing mildly context-sensitive formalisms that are *not* based on weak generative capacity.
  - The ability of TAG/LIG/CCG to capture the relevant structural pattern is striking and intriguing given that they are *strictly less powerful* than MGs in weak generative capacity!
- "Everyday minimalism" corresponds better to LIG/CCG (**Ext**,**¬Fin**) than to MG. So what we're raising is a challenge for MG's fit to the empirical pattern, and also for MG's fit to everyday minimalism — not a challenge for everyday minimalism.

Open questions:

- Nesting vs. crossing of hierarchical wh-trace dependencies? LIG/CCG and TAG both predict nesting, but there are other ways to relax **Ext** and **Fin** (Rogers, 2003; Weir, 1992).
- What sort of adjustment to MGs will best let them account for the Bulgarian pattern?
- What does the additional power of MGs in weak generative capacity correspond to in structural terms? Is it necessary?