We don't need no |A|rboration

Julian Bradfield
University of Edinburgh

we don't need no m-command
no deep structure in the segment
linguist, leave them As alone

with profuse apologies to Pink Floyd

ET and GP

Element Theory (ET) asserts that segments comprise (multi)sets of elements.
Each element conveys some rather concrete phonetic property: I for high, palatal, front; U for round, back (sometimes); A for low, coronal (in some versions), open; H for friction, aspiration; and so on.
Once there were up to ten elements; now there are four to six. Power is added by *heading* elements A 'strong A'.
Government Phonology (GP) expresses phonology by constraints between segments, elements, and licensing or government relations.

The frightening power of GP2.0

GP2.0 [Pö06] is a development of ET and GP. Its main change is to combine elements into binary trees (vs. (multi)sets in classic GP). This allows descriptions of several disparate phenomena by the same mechanism, and the elimination of elements in favour of structure.
This data structure makes GP2.0 very rich in expressive power – several restrictions are imposed to reduce this. Today's question: what does the structure actually do, and is it the simplest way to do it?

Elementary GP2.0

There are (depending on version) as few as three elements: I (high/front), U (round/back), L (voicing/nasality). Elements are combined by a tree-formation rule (version of [Pö18]) using notions from syntactic X-bar theory:

- $\underline{x}$, $[_{\underline{x}'}\ \underline{x}\ \ x]$, $[_{\underline{x}''}\ [_{\underline{x}'}\ \underline{x}\ \ x]\ \ x]$ are trees;
- any of the above may be embedded in an x slot of any of the above;
- the underlined (head) terminals may carry elements.

Various types of openness are represented by deeper structures: lowness of vowels, aperture of consonants, etc.
Additional expressive power, and ways to constrain it, come from stipulating various government-style relations between nodes in the trees.

Representation theory

A standard technique for understanding a class of difficult objects is to map them to simpler (or easier to understand) objects while preserving properties of interest.
We will try to represent the trees of GP2.0 by tuples of multivalent elements, and see how much survives.

Example [Pö18]: vowel reduction

In Brazilian Portuguese. /ɛ/ merges to /e/ in prestressed positions, and further to /i/ in final unstressed position.
In GP2.0 representation, mergers cut down the tree as 'weak positions allow less space', describing both steps uniformly.

$/i/ = [_{\underline{x}'}\ \underline{x}{:}\,\mathsf{I}\ \ x]$ $\quad /e/ = [_{\underline{x}''}\ [_{\underline{x}'}\ \underline{x}{:}\,\mathsf{I}\ \ x]\ \ x]$ $\quad /\varepsilon/ = [_{\underline{x}''}\ [_{\underline{x}'}\ \underline{x}\ \ [_{\underline{x}'}\ \underline{x}{:}\,\mathsf{I}\ \ x]]\ \ x]$

However, in Eastern Catalan, /e, ɛ/ both merge to /ə/ in unstressed position.
Here [Pö18] proposes that reduction is reduction to height 1 trees, and the melodic |I| is in a removed layer. The natural SGP model is /ə/ = |.|, /e/ = |A.I|, /ɛ/ = |$\underline{\mathsf{A}}$.I|, so reduction removes both A and I. Thus GP2.0 unifies the reduction computation but uses different representations for BP and EC front vowels.

$/ə/ = [_{\underline{x}'}\ \underline{x}\ \ x]$ $\quad /e/ = [_{\underline{x}'}\ \underline{x}{:}\,\mathsf{I}\ \ [_{\underline{x}'}\ \underline{x}\ \ x]]$ $\quad /\varepsilon/ = [_{\underline{x}''}\ [_{\underline{x}'}\ \underline{x}{:}\,\mathsf{I}\ \ [_{\underline{x}'}\ \underline{x}\ \ x]]\ \ x]$

Is the *structure* doing the work?

The reduction reduces the height of the tree; and the placing of melody at a particular height determines when it vanishes. The binary tree structure doesn't seem to be doing any work …

Pruning the trees – multivalent elements?

Let h be the height of the tree. For element X, let $d(X)$ be its depth in the tree. The representation $(h, X_{d(X)}, \dots)$ is enough to do the work:
BP: /i/ = $(1, \underline{\mathsf{I}}_1)$ /e/ = $(2, \underline{\mathsf{I}}_2)$ /ɛ/ = $(3, \underline{\mathsf{I}}_3)$
EC: /ə/ = (1) /e/ = $(2, \underline{\mathsf{I}}_1)$ /ɛ/ = $(3, \underline{\mathsf{I}}_2)$

Reduction lops $h-1$ levels (from the root!), so reducing h and d; when d goes to zero, the element disappears:
BP: /ɛ/ = $(3, \underline{\mathsf{I}}_3) \to$ /e/ = $(2, \underline{\mathsf{I}}_2) \to$ /i/ = $(1, \underline{\mathsf{I}}_1)$
EC: /ɛ/ = $(3, \underline{\mathsf{I}}_2) \to$ /e/ = $(2, \underline{\mathsf{I}}_1) \to$ /ə/ = (1)

A by the back door

Why do we need to talk about h separately? We could write
BP: /i/ = $(\mathsf{A}_0, \underline{\mathsf{I}}_1)$ /e/ = $(\mathsf{A}_1, \underline{\mathsf{I}}_2)$ /ɛ/ = $(\mathsf{A}_2, \underline{\mathsf{I}}_3)$
Here A is also a multivalent element, and its special property is allowing space for other elements to show different valences – another way of 'making room', but without unnecessary trees.

Example [Pö06]: pre-fortis clipping

One of the original applications: shortening of English vowels before fortis consonants: *whiff* [ʍɪfˑ] vs *give* [gɪˑv]. Analysis:

- fortis/lenis are distinguished by length, not by H ('aspiration'), thus
- /f/ is $[_{\mathrm{O}'}\ x_1\ \ x\mathrm{O}{:}\,\mathsf{U}]$, but is also specified for 'm-command', which spreads the U to give surface $[_{\mathrm{O}'}\ x_1{:}\,\mathsf{U}\ \ x\mathrm{O}{:}\,\mathsf{U}]$ adding length;
- /v/ is like /f/, but without m-command, so $[_{\mathrm{O}'}\ x_1\ \ x\mathrm{O}{:}\,\mathsf{U}]$;
- /ɪ/ is (here) $x\mathrm{N}{:}\,\mathsf{I}$
- in the case of /ɪv/, the /ɪ/ 'm-commands' the /v/'s x_1 node resulting in /v/ surfacing as $[_{\mathrm{O}'}\ x_1{:}\,\mathsf{I}\ \ x\mathrm{O}{:}\,\mathsf{U}]$, and by the magic of 'm-command' the x_1 'slot' is borrowed by /i/ for its melody.

In this example, we see the use of structure to subdivide skeletal slots, mediated by stipulated 'm-command'.

Structure vs length

In our 'pruned representation', what would $(2, \mathsf{I}_2, \mathsf{I}_2)$ mean? We can as well stipulate that each slot is a timing unit. m-command is not needed: /f/ is directly specified as $(2, \mathsf{U}_2, \mathsf{U}_2)$ and normal spreading suffices for /ɪv/.
In [Pö06, pp. 72–84], m-command is carefully constrained to avoid over-generating; the pruned representation can be directly constrained in the length of the tuple.

Summary

GP2.0 uses binary trees for (at least) two purposes: to give the effect of varying strengths of elements, and to give fine control of timing slots. The exponential size of trees requires constraints to avoid over-generation; these are not always transparent.
Representing the slots and multivalent elements directly avoids the over-generation, and achieves most of what the trees do.
We have elided complexity in the versions of GP2.0: the precise rules on which terminals can carry melody, notions of heading, and types of licensing vary from one paper to another,. It remains to be done to formalize all versions and see if pruning preserves enough information.
So what is the argument for trees? One is that in GP2.0, the tree structure of segments is a continuation of the tree structure of prosody. (Does prosody need trees?)

References

[Pö06] Markus A. Pöchtrager. *The Structure of Length*. PhD thesis, Universität Wien, 2006.

[Pö18] Markus A. Pöchtrager. Sawing off the branch you are sitting on. *Acta Linguistica Academica*, 65(1):47–68, 2018.

LaTeX TikZposter