How does the word-order complexity of Australian languages compare with that of other “flexible” word-order languages?

Recent computational research by Wedekind & Kaplan [WK20] builds a bridge between:

- the LFG framework, used in much research on Australian languages, and
- the LCFRS formalism, used in much research on “free word order” in other kinds of languages (e.g. German, Dutch, Turkish, . . .).

I use this connection to identify two apparent differences between the Australian type of word-order complexity and the type that has been the focus of LCFRS work, where the Australian type is less complex and one where it is more complex.

<table>
<thead>
<tr>
<th>Depth of zippers</th>
<th>Width of zippers</th>
<th>Language type</th>
</tr>
</thead>
<tbody>
<tr>
<td>bounded</td>
<td>unbounded</td>
<td>Australian-style discontinuity</td>
</tr>
<tr>
<td>bounded</td>
<td>unbounded</td>
<td>Germanic-style discontinuity</td>
</tr>
</tbody>
</table>

These observations add to the theoretical significance of specific empirical questions concerning:

- clausal subordination
- discontinuous nominal expressions

Discontinuity in LFG

LFG incorporates some substantive hypothesised limitations on the patterns of discontinuity that can be produced, motivated by observations of what is and isn’t found in certain (non-Australian) languages.

For example, it allows the famous Dutch crossing-dependencies pattern [H76], but rules out many conceivable unattested options [K10].

(1) ...dat Jan Piet Marie zag helpen zwemmen 
   ...that Jan saw Piet help Marie swim

 Comparison: Dutch crossing-dependencies pattern

- Clauses appear contiguously as in (6), but generally not in a discontinuous form like (7), which would be parallel to (5).

Comparison: LFG analyses generally predict that more than two pieces should be possible: any number of horizontally-related NP nodes see in familiar examples like (4) and (8) (from Kalkatungu [B83, cited in N14]).

This would mean that analyses of Australian-style discontinuous NPs require zippers of unbounded width.

An important empirical question, therefore, is whether discontinuous NPs can be split up into more than two pieces we see in familiar examples like (4) and (8) from Kalkatungu [B83, cited in N14]).

LFG analyses generally predict that more than two pieces should be possible: any number of horizontally-related NP nodes might contribute to a unified f-structure (e.g. via a set-valued ADV/NTs attribute) in a structure like (9).

Difference 1: Depth of zippers

Languages like Warlpiri show very free word order within clauses [HS3], to the extent of allowing discontinuous NPs:

(4) wawirri kapi-ma panti-mi yalumpu
    kangaroo AUX spear-NONPAST that
    “I will spear that kangaroo”

But when a multi-word grammatical element is a subordinate clause, it is (almost?) always contiguous [N06].

Or stated in LFG terms: we do not find pairs of horizontally-related S nodes that map to a shared f-structure: subordinate clauses appear continguously as in (6), but generally not in a discontinuous form like (7), which would be parallel to (5).

Difference 2: Width of zippers

LCFG allows intermingling between unboundedly many discontinuous elements (deep zippers), but requires that there is a bound on the number of pieces into which a discontinuous element is “split” (the width of a zipper).

An important empirical question, therefore, is whether discontinuous NPs can be split up into more than the two pieces we see in familiar examples like (4) and (8) (from Kalkatungu [B83, cited in N14]).

LFG allows intermingling between unboundedly many discontinuous elements (deep zippers), but requires that there is a bound on the number of pieces into which a discontinuous element is “split” (the width of a zipper).