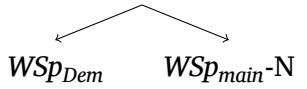
Appendix 2 - Cases of gaps

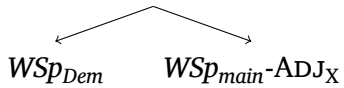
Appendix of the article: Pinzin, Francesco & Mattiuzzi, Tommaso. 2025. Word-order information in the Lexicon. *Glossa: a journal of general linguistics* 10(1), 1–49.
DOI: https://doi.org/10.16995/glossa.18521

As discussed in Section 4.3, merger of a DEMP on top of a configuration derived via roll-up like N - Adj - Num creates a structure compatible with two subextraction options. We argued there that a difference in the LIs involved can model a difference between languages that have the basic word-order N - Dem - Num - Adj (Language type 1), and those that have the order NP - Adj - Dem - Num (Language type 2). Respectively, Language type 1 was argued to involve (1), and Language type 2 (2). The crucial difference lies in the category of the constituent of which they dictate evacuation via lexicalisation-driven movement, respectively N and Adj.

(1) LI_{78}: ⟨ DEMP ⟩



(2) LI_{79}: ⟨ DEMP ⟩



Here, we show how the format adopted for the relevant LIs allows to correctly capture the expected behaviour of the two types languages in cases in which one of the categories of the nominal phrase is missing. Specifically, we discuss two such cases of gaps.

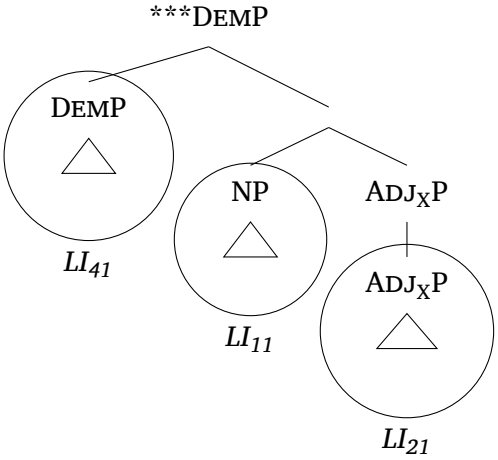
These example derivations help highlight a more general point: the power granted to LIs by allowing them to explicitly specify a "cut off point" is motivated by the attempt to capture with a single LI all relevant strings obtained by merging a given category (here, Dem) to a nominal phrase, regardless of the presence or absence of other lower categories. The same cannot be obtained by making reference to more specific configurations of LIs or previous derivational stages, since both alternatives would make the insertion of LIs like LI_{78} or LI_{79} systematically sensitive to the specific derivational history prior to the insertion of Dem.

No Num

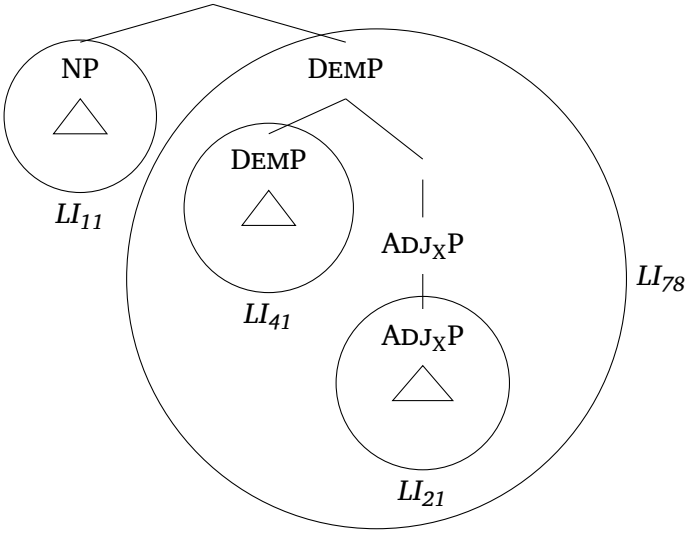
Absent a Num, we expect Language type 1 (with basic order N - Dem - Num - Adj) to show the surface order N - Dem - Adj, and Language type 2 (with basic order N - Adj - Dem - Num) N - Adj - Dem.

For Language type 1, LI_{78} (1) dictates lexicalisation-driven movement of N out of the complement of the DemP branch. The base merge configuration is (3a) is therefore not licensed. Lexicalisation-driven extraction of N as in (3b) instead lead to a match, licensing the structure that is then linearised as the expected string N - Dem - Adj.

(3) a.

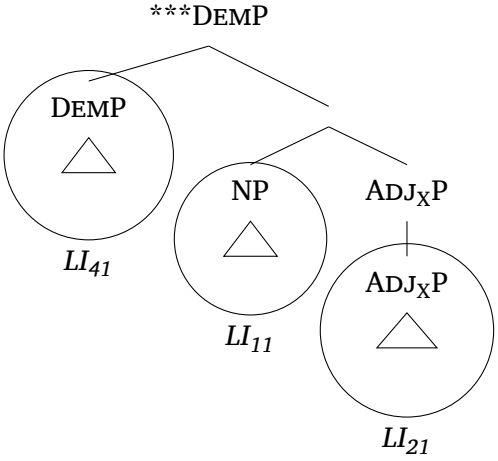


b.

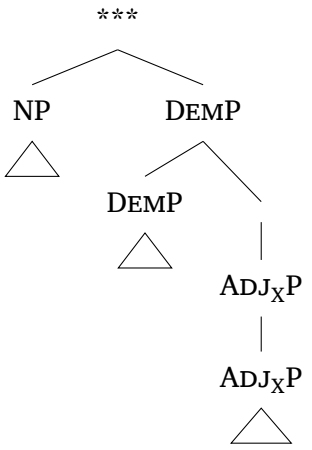


For Language type 2, the first application of lexicalisation-driven movement (producing (4b) from (4a)) fails to yield a match, since LI_{79} (2) requires movement of any item in the structure up to category Adj_XP. Therefore, the licensed configuration is the one obtained by the application of the subsequent step in the Lexicalisation Algorithm (4c), which underlies the expected string NP - Adj - Dem.

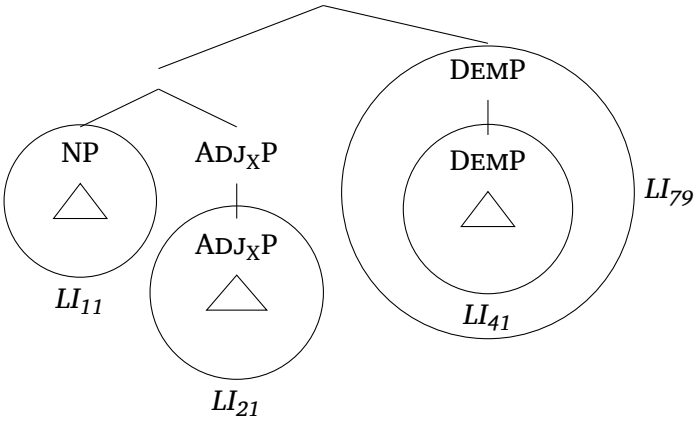
(4) a.



b.



c.

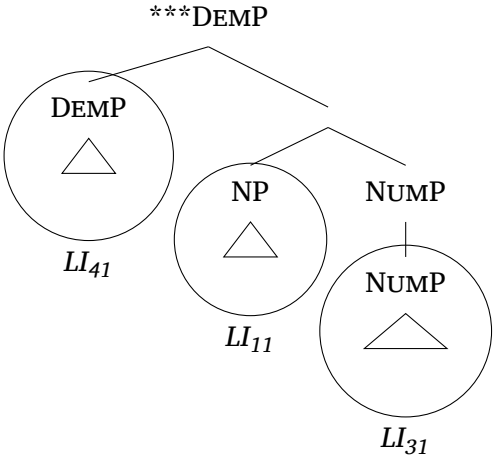


No Adj

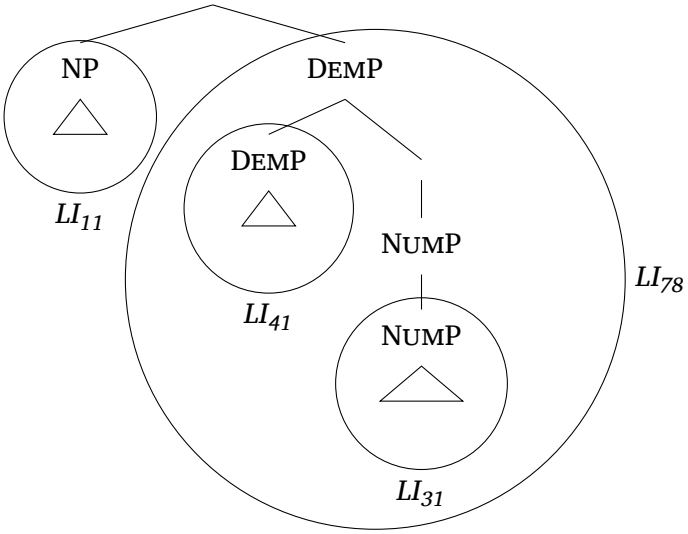
In the absence of any adjective, the output for both type of languages is the same, namely, N - Dem - Num. The logic of LIs discussed in Section 4.3 allows to capture this superficial identity, despite the two types involve distinct LIs, namely (1) and (2).

In neither type of language is the configuration derived by merging Dem directly licensed (5a). As a consequence, lexicalisation-driven movement is triggered, whose initial target is NP. This produces a configuration in which the sister node of Dem does not contain N, allowing the insertion of LI_{78} (1) for Language type 1.

(5) a.

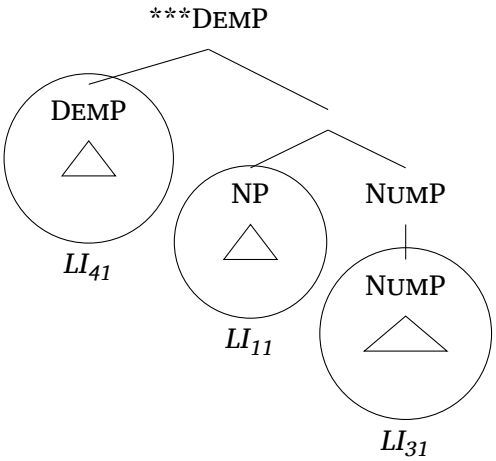


b.

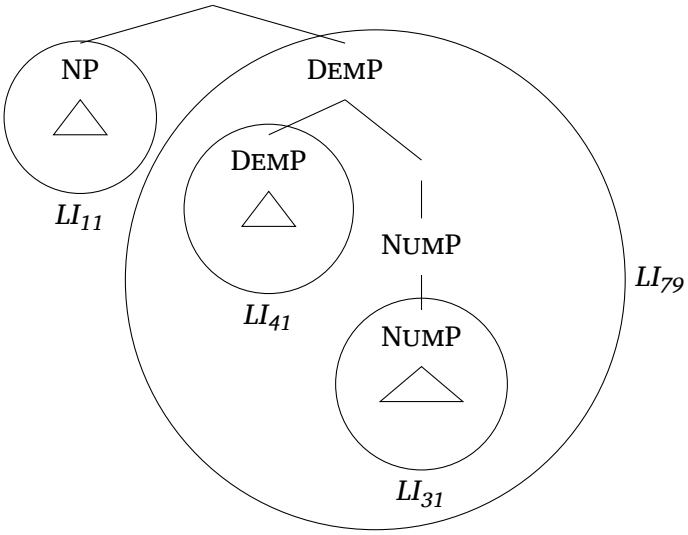


The same initial operation is triggered for Language type 2. Note that the configuration obtained is such that the sister node of Dem does not contain Adj. This allows the insertion of LI_{79} (2).

(6) a.



b.



In both scenarios, the same configuration is obtained, which corresponds to the expected string N - Dem - Num.