The expression of number (#) within the noun phrase has been argued to vary between a high (num) and a low position, which Kramer (2014), i.a., associates with n, providing the root with a syntactic category. We argue that Linking Morphemes (L) in Dutch provide new evidence for such a split, and moreover, for a low expression of # in a language that is normally considered to have high #. We argue that L is realized in the absence of num when n bears a plural and an additional feature. In the presence of num, n is typically not spelled out. Under this approach, the different properties, i.e. containment and variation across different contexts, can be accommodated. By taking L to instantiate n, the presence or absence of L can be taken as a diagnostic of the size of non-head elements. Combined with recent work on Germanic compounds (Harðarson 2016, De Belder 2017) this makes a prediction about the order of modifiers in Dutch compounds, which we show is borne out.

**Ls are systematic:** The most relevant properties of L are (i) the set of Ls in Dutch is the same as the set of plural suffixes (-en, -s) but the L for any given noun need not be the same as the plural affix: (2), -s linkers can occur with -en plurals, but -en linker only appears on -en plural nouns outside of compounds. (ii) usually, the choice of L depends on the non-head element, (1c-d) (Hoekstra 1996, Krott et al. 2001, De Belder 2013, 2017, a.o.). (iii) although Ls are homophous to plural markers, plural interpretation of non-heads is not obligatory (Booij 2001), (1d).

(iv) Double #-marking can be observed outside of compounds, (3): Some nouns takes -eren as plural, instead of -en (3b). In compounds, -en is absent but -er is present, (3c).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>cat</td>
<td>‘cat’</td>
<td>cat-PL</td>
<td>cat-L-luik</td>
<td>cat-L-turd</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(2)</th>
<th>a. hond</th>
<th>b. hond-en</th>
<th>c. hond-s-dol</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>‘dogs’</td>
<td>dog-PL</td>
<td>dog-L-crazy</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(3)</th>
<th>a. kalf</th>
<th>b. kalv-er-en</th>
<th>c. kalv-er-liefde</th>
</tr>
</thead>
<tbody>
<tr>
<td>calf</td>
<td>‘calves’</td>
<td>calf-PL-PL</td>
<td>calf-L-love</td>
</tr>
</tbody>
</table>

Dutch n#: Under (i-iv), we argue that Ls express #-features on n (class markers in De Belder 2017) that usually appear in compounds. The Dutch NP structure is as (4).

| (4) | [DP [D [numP num[#] [nP n[#] [ROOT ]]]] |

Dutch has two positions to express #: n and num. In general, Dutch is considered a num-# language (Acquaviva 2008). Extensive typological work shows that languages differ in where # is expressed (Kramer 2014, a.o.). n-# has (i) selectional restrictions, (ii) can have multiple #-marking, (iii) occurs in derivational contexts; num does not, and (iv) n-# is non-deterministic, where num-# is. These are consistent with Dutch Ls: (i) L only appears with nominal (and adverbial) stems (De Belder 2013), (ii) some nouns show multiple #-marking, (3), and (iii) L can be used to derive adverbs, (5). We argue that the #-typology can be extended by examining compounds in Dutch and taking Ls as an instantiation of n-#.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>She sang soft-DIM-L</td>
<td>he walks to front-L</td>
<td></td>
</tr>
<tr>
<td>‘She was singing quietly’ (De Belder 2013:18)</td>
<td>‘He walks to the front’</td>
<td></td>
</tr>
</tbody>
</table>

**Deriving Dutch L-compounds:** We argue that compounds are formed by adjoining the non-head element directly to the head of the compound in syntax after being formed in separate workspaces.
(e.g. Piggott & Travis 2013). Following e.g. Ritter 1991, i.a., *num* is then merged, resulting in the structure in (6). At the point of vocabulary insertion, the rules in (7)–(9) apply.

(6) \[
\begin{array}{c}
N & num \\
\text{Root} & \text{n} & \text{root} & \text{n} \\
\sqrt{\text{CAT}} & \text{u[PL][β]} & \sqrt{\text{TURD}} & \text{u[PL][β]}
\end{array}
\]

(7) VI roots
a. \(\sqrt{\text{CAT}} \rightarrow \text{kat}\)
b. \(\sqrt{\text{TURD}} \rightarrow \text{drol}\)

(8) VI #-n
a. \(\text{n} \rightarrow /\text{a}/ /\text{u[+PL], γ}\)
b. \(\text{n} \rightarrow /\text{an}/ /\text{u[+PL], β}\)
c. \(\text{n} \rightarrow /\text{s}/ /\text{u[+PL], α}\)

(9) VI #-num
a. \(\text{num} \rightarrow /\text{an}/ /\text{[β]a i[+PL]}\)
b. \(\text{num} \rightarrow /\text{s}/ /\text{i[+PL]}\)

We argue that the difference between #-n and #-num comes from interpretability, which gives rise to different meanings: #-n is uninterpretable, #-num is interpretable, and both are in an Agree relation (Kramer 2016). Interpretable features are always semantically plural, whereas as in 1d, uninterpretable features are not (Smith 2015). Because of the Agree relation, the VI-s overlap, but are spelled out differently on *n* due to additional features. Following De Belder (2013), we assume that the subset relation between the linkers and the plural suffix is a result of accumulation of features, i.e. that the feature combination resulting in -s is a subset of the feature combination resulting in -en, (10). We set aside for the time being the precise identity of the relevant features.

We assume that double plural marking is rarely observed because # on *n* is deleted in the presence of num-#. In cases like (3), deletion does not apply, hence plural marking on *n* and num is observed.

### Presence/Absence of linker:
Our proposal offers a test for the size of compounds. It has been argued that compounding takes place at different layers and the size of the element determines the layer at which it can be attached (Harðarson 2016, De Belder 2017), e.g. only elements of the same size can be compound. Hence roots (11a-b) can occur in root-root compounds: (10), but they cannot occur outside of categorized material. If the absence of L indicates the absence of *n*, bracketing effects are expected: (11a) should not appear outside of a compound with L, since L signals *n*. This is borne out, (13).

(10) \(\alpha^+ \quad -s \quad -en \quad -er\)

(11) a. *kleer b. kle(e)r-en
\(\text{cloth} \quad \text{cloth-L/PL}\)
\‘clothes’

(12) \([\text{jurk-en}]x [\text{kleer}]\text{root-[maker]}\text{root} ]x\)
\x\
\‘clothing maker of dresses’

\(\text{cat-L} \quad \text{curtain - pattern}\)
\‘curtain patterns with cats on them’

int.: ‘pattern maker for clothing’

### Conclusion:
We develop a theoretical analysis of L in Dutch arguing that it is *n-#, and contribute to the typology of #. Furthermore, under this approach, the distribution of L in Dutch can be accommodated. Finally, taking L as *n-#* offers a test for compound size.

### Selected references:
De Belder, Marijke. 2013. Linking phonemes are class-markers.
—. 2017. The Root and Nothing but the Root: Primary Compounds in Dutch
Harðarson, Gisli Runar. 2016. Peeling away the layers of the onion.