

Two-*(cL) structures for the price of one-(cL) in spoken Persian

Yasaman Sanei & Gautam Ottur

University of Göttingen & University of Kassel



References

Basic pattern

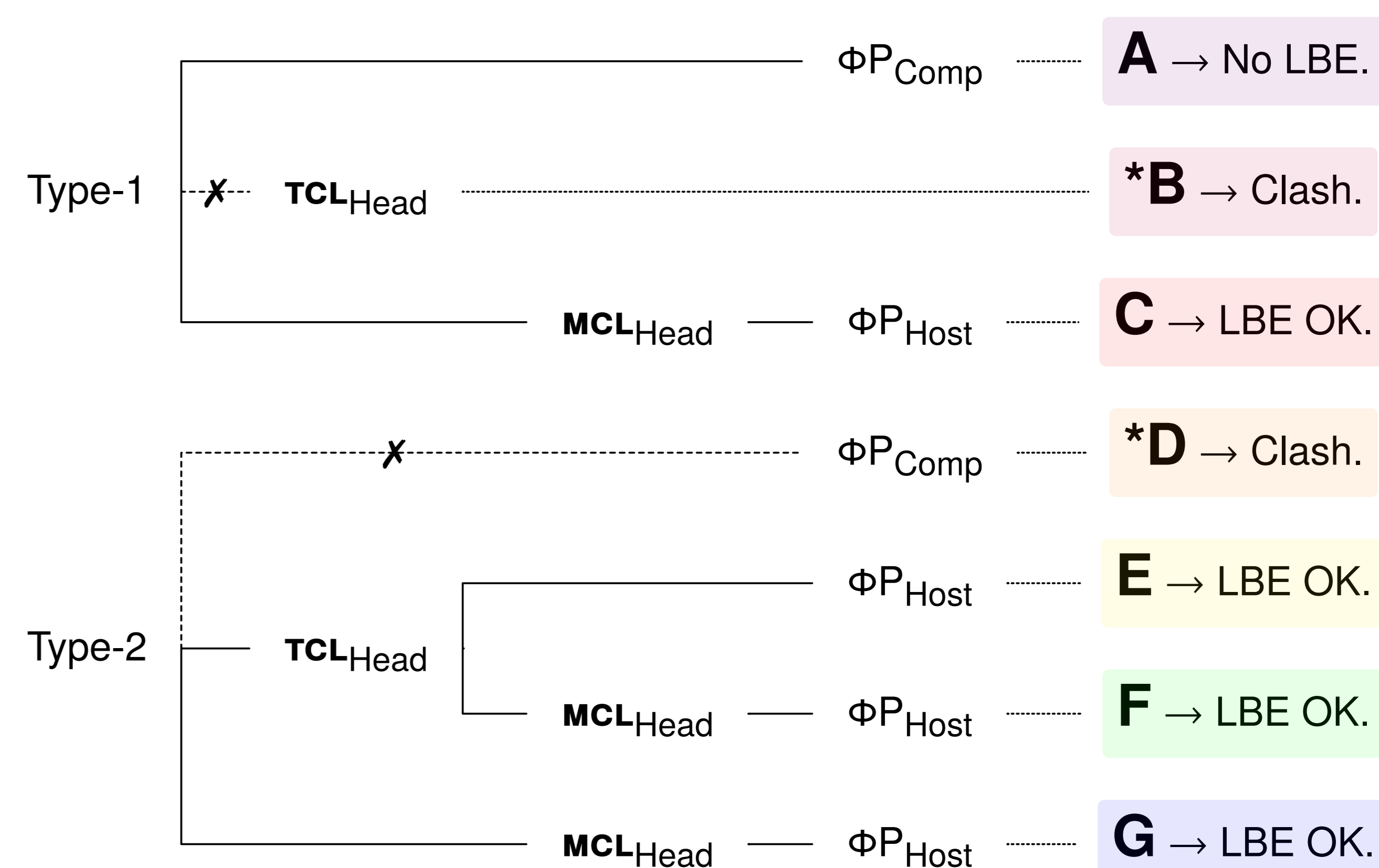
- Spoken Persian numerals generally need **classifiers** to modify nouns.
- The modified nouns are generally **marked singular**.
- Classifiers come in two types: the “true” classifier *tā* (the **TCL**) and a number of measure classifiers (**MCLs**).
- Most numerals require the **TCL**, and **MCL**, and allow both, with the order **TCL** < **MCL**.
- But *yek* ‘one,’ and numerals with *yek* at the right edge, i.e. *bist-o-yek* ‘twenty-and-one,’ *si-o-yek* ‘thirty and one,’ etc. **share a peculiar distribution**:
 - No classifier required, compatible with **MCLs**, incompatible with the **TCL** *tā*.
- We call these *yek*-numerals **Type-1 numerals**, and the others **Type-2 numerals**.
- Our claim: Classifiers are not heads on the nominal spine.**
 - Numeral+classifier complexes (NCCs) can undergo left-branch extraction (LBE).
 - Bare Type-1s cannot undergo LBE!

- (1) a. [*ye(k) ja'be*]_i *man har ruz* [_{—i} *širini*] *mi-xor-am*
 one **MCL** 1SG every day sweet DUR-eat:PRS-1SG
 ‘I eat *one* box of sweets every day.’
 b. **ye(k)*_i *man har ruz* [_{—i} *širini*] *mi-xor-am*
 one 1SG every day sweet DUR-eat:PRS-1SG
 Intended: ‘I eat *one* sweet every day.’

	Type-1 numeral	Type-2 numeral
Bare	A → <i>ye(k) gol</i> one flower ‘one flower’	*D → <i>do gol</i> two flower ‘two flowers’
TCL	*B → <i>ye(k) tā gol</i> one TCL flower ‘one flower’	E → <i>do tā gol</i> two TCL flower ‘two flowers’
TCL+MCL	*B → <i>ye(k) tā šāxe gol</i> one TCL MCL flower ‘one flower’	F → <i>do tā šāxe gol</i> two TCL MCL flower ‘two flowers’
MCL	C → <i>ye(k) šāxe gol</i> one MCL flower ‘one flower’	G → <i>do šāxe gol</i> two MCL flower ‘two flowers’

General proposal

- Idea 1:** the distributions of different kinds of numerals/classifiers can be explained by **c(ategory)-selection**: heads select their complements, and adjuncts select their hosts based on their formal features.
- Idea 2:** the LBE asymmetry is because **NCCs are adjuncts** (cf. Bale et al. 2019), but bare numerals are **projecting heads**.
- Numerals and classifiers share a feature, but with **different subfeatures**, and carry variously specified probes which trigger/license Merge.
- There are two kinds of probes for Merge: [**•F•**] allows something to **take an argument** that bears [F]. [**◦F◦**] allows something to **adjoin to a host** that bears [F].
- These probes/goals may be specified [**•F: va/•**] or underspecified [**•F•**].



Derivations

- The noun is **nested** within a ΦP , which has the feature [ϕ : SG].
- Type-2s** have [ϕ : PL], and can’t attach to ΦP without a classifier (***D**).
- Classifiers have a probe [$\circ\phi$: SG \circ], so they can **adjoin** to ΦP .
- The **TCL** has $\{[\phi], [\circ\phi: PL\circ]\}$, so it can **select Type-2s** as complements (**E**).
- The **MCL** has $\{[\phi], [\circ\phi\circ]\}$, so it can **select either Type-2s (G) or the TCL (F)** as its complement.
- NCCs, as **adjoined constituents**, can undergo LBE.
- Type-1s** have $\{[\phi], [\circ\phi: SG\circ]\}$, so they can only be **selected as complement by MCLs (C)**, and **not the TCL (*B)**.
- Type-1s** can also select ΦP as their **complement** directly (**A**), without a classifier. They then **project**, so they naturally **can’t undergo LBE**.

Consequences

- Two intersecting factors are at play: **projectional differences** (adjunction vs. complementation) and **overlapping distributions**.
- We show that two theoretical revisions can efficiently capture this:
 - Separating projective and non-projective selection** (Ottur 2025, Zyman 2023).
 - Using subfeatures/values**, similar to Agree (e.g. Zeijlstra 2020).
- Of 8 conceivable structures, this correctly derives **just the attested 5** while also capturing other facts, viz. the hierarchy of **TCL** and **MCL** and the asymmetry in **LBE**.
- This allows for a **more straightforward explanation of syntactic distribution** than having arbitrary features for every unique distribution, e.g. [TCL], [MCL], etc.

