# Formalizing Feature Inheritance

Greg Kobele    Lei Liu

Leipzig University

UNIVERSITÄT LEIPZIG

## Feature Inheritance

Transformational cycle: (cyclic) transformational rules applied only once certain categories (always S, often NP, sometimes PP) of expressions were built

Early Minimalism: movement interleaved with the structure building operation of merge; can in principle apply at any time, regardless of the categorial status of its input

Feature Inheritance: delays movement until a particular category is reached

### *Background*

1. Basic clausal architecture: $C - T - v - V$
2. $C$ and $v$ are *phases* — play a special role in system
   - delimit cyclic domains of interpretation
   - engender island effects; circumventable by movement to *edge*/Spec of head
   - wh-phrases move to (and through) $C$, and also through $v$
3. phase heads are *same-same* but *different*
   - $v$ agrees with object, but $T$ (not $C$) with subject
   - mvt to $C$ edge always A-bar, to $v$ only sometimes

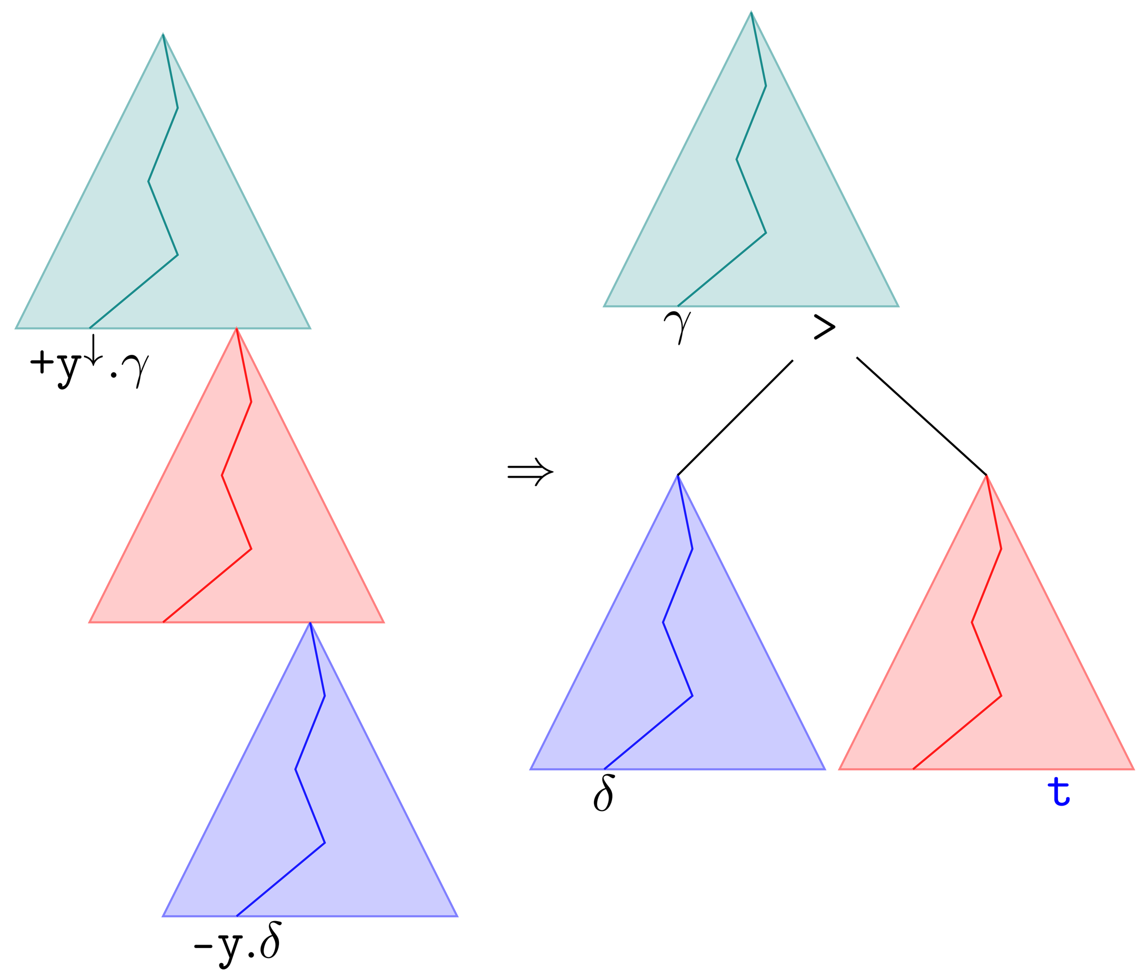
The Idea: phase heads are the (sole) locus of movement and agreement triggers
- phase head bequeaths agreement feature to complement head, triggering movement to specifier *of complement*
- phase head triggers movement to its periphery

agreement and edge movement happen simultaneously; if same item responds to both features, *distinct* chains created

The Goal: just one phase head; call it $v$ or $C$ depending on category of complement

### Towards a formalization

We introduce a '↓' diacritic on a movement feature, which bequeaths that feature to lower head, immediately triggering movement to lower head's specifier:




To account for 'simultaneity' of bequeathed movement and movement to own Spec, we introduce a pairing construct for features: (+y$^{\downarrow}$, +z) which *simultaneously*

1. triggers tucking in movement (+y$^{\downarrow}$) to the spec of its complement
2. triggers movement (+z) to its spec

These new feature types are restricted in their distribution in feature bundles:

- only one ↓ marked feature in a feature bundle: feature inheritance happens just once
- ↓ marked feature must immediately follow first selection feature: feature inheritance immediately follows merger of complement
- the first component of a pairing construct must be ↓-marked: simultaneous movements only happens as a result of feature inheritance

## Results

Feature Inheritance increases neither the generative capacity of the MG formalism, nor its worst case time complexity. (But see the adjacent result on multiple feature inheritance.)

- to establish equivalence, as well as permit easy complexity analysis, operations should be in inference rule format, stated over finite sequences of strings paired with feature bundles [Michaelis, 2001]
- parsers derived from the above notation can have their worst-case time complexity read directly off of the rules themselves [Stanojević, 2019]

### *Analyzing Complexity of Logic Programs*

Stanojević [2019] reminds us that the worst case time complexity of an MG parser given as a set of inference rules can be determined by counting the number of distinct variables in the antecedent of the most complex rule [McAllester, 2002]. A MG inference rule for a final movement step follows.

$$\frac{\langle v, \text{+x}.\alpha, \rangle, \vec{\phi}, \langle u, \text{-x}\rangle, \vec{\psi}}{\langle uv, \alpha\rangle, \vec{\phi}, \vec{\psi}}\mathcal{O}(n^{2k+1})$$

Each of $\vec{\phi}$ and $\vec{\psi}$ are sequences (possibly empty) of string-feature bundle pairs. The **SMC** constraint imposed on minimalist grammars [Stabler and Keenan, 2003] ensures that the total number of such pairs in any minimalist expression is less than or equal to $2k+1$, for $k$ a constant determined by the grammar. To deal with strings and string concatenation in a logic program, a string is replaced by a pair of integer variables. The above inference rule is then given as the following in a logic program:

$$\frac{\langle (j,k), \text{+x}.\alpha, \rangle, \vec{\phi}, \langle (i,j), \text{-x}\rangle, \vec{\psi}}{\langle (i,k), \alpha\rangle, \vec{\phi}, \vec{\psi}}\mathcal{O}(n^{2k+1})$$

Because $u$ and $v$ end up being concatenated, the variables for the right edge of $u$ and for the left edge of $v$ are identical. Altogether, the $u$ and $v$ pairs contribute three unique variables. There can be at most $k-1$ other pairs, each of which could contribute at most two unique variables, resulting in a total of $2k+1$ potentially unique variables. For an input string of length $n$, there are thus $n^{2k+1}$ ways of choosing positions in this string which could match with this rule, and so as a function of string length, the complexity of this rule is on the order of $n^{2k+1}$.

We avoid tucking-in movement by performing the simultaneous movement *during the merge of the complement*.

### *Implementation*

Just the new inference rules are presented below. In inference rule notation, to each term corresponds a sequence of string-feature bundle pairs. Each pair beyond the first corresponds to a maximal proper subterm whose head begins with negative features. The first pair corresponds to the term minus these moving pieces.

**MrgFI1a** single mover, last movement step, and is pronounced in its highest position.

$$\frac{\langle m, \text{+x}.(\text{+y}^{\downarrow}, \text{+z}).\alpha\rangle \quad \langle n, \text{-x}\rangle, \vec{\phi}, \langle o, \text{-y}.\text{-z}\rangle, \vec{\psi}}{\langle omn, \alpha\rangle, \vec{\phi}, \vec{\psi}}\mathcal{O}(n^{2k+2})$$

**MrgFI1b:** the single mover has features left over, and thus continues moving.

$$\frac{\langle m, \text{+x}.(\text{+y}^{\downarrow}, \text{+z}).\alpha\rangle \quad \langle n, \text{-x}\rangle, \vec{\phi}, \langle o, \text{-y}.\text{-z}.\beta\rangle, \vec{\psi}}{\langle mn, \alpha\rangle, \vec{\phi}, \langle o, \beta\rangle, \vec{\psi}}\mathcal{O}(n^{2k+3})$$

**MrgFI2a:** two movers, last movement step, and so are pronounced here. The tucking-in mover *o* is between head *m* and its complement *n*.

$$\frac{\langle m, \text{+x}.(\text{+y}^{\downarrow}, \text{+z}).\alpha\rangle \quad \langle n, \text{-x}\rangle, \vec{\phi}, \langle o, \text{-y}\rangle, \vec{\psi}, \langle p, \text{-z}\rangle, \vec{\chi}}{\langle pmon, \alpha\rangle, \vec{\phi}, \vec{\psi}, \vec{\chi}}\mathcal{O}(n^{2k+1})$$

**MrgFI2b:** two movers, but there is continuing movement by second.

$$\frac{\langle m, \text{+x}.(\text{+y}^{\downarrow}, \text{+z}).\alpha\rangle \quad \langle n, \text{-x}\rangle, \vec{\phi}, \langle o, \text{-y}\rangle, \vec{\psi}, \langle p, \text{-z}.\beta\rangle, \vec{\chi}}{\langle mon, \alpha\rangle, \vec{\phi}, \vec{\psi}, \langle p, \beta\rangle, \vec{\chi}}\mathcal{O}(n^{2k+2})$$

We see that the rule **MrgFI1b** contributes the most to the worst case time complexity of the new rules. To put this in perspective, the worst case time complexity of minimalist grammars *without* feature inheritance is also $\mathcal{O}(n^{2k+3})$ [Fowlie and Koller, 2017, Stanojević, 2019]. Thus minimalist grammars with feature inheritance have the same worst case time complexity as vanilla MGs.

## Multiple Feature Inheritance

Branigan [2020] proposes that features from C can be inherited by multiple *different* lower heads.

- We feared `that` **such a whiskey**, *we* might never meet a distiller of

Here **such a whiskey** has moved to a topic position below C (`that`), but higher than the subject position (of *we*) in Spec-TP.

This is analyzed in terms of C passing a topic feature to an intermediate head '$\delta$', and at the same time a subject feature to the lower T head, thus, *multiple feature inheritance*. Another potential analysis (not pursued by Branigan) has it that both **such a whiskey** and *we* are specifiers of the same T head. This analysis allows for viewing this as simply an instance of normal feature inheritance (but where C is passing two features down to T). We here call this the *batch* analysis of feature inheritance.

### *(Multiple) FI as tucking in*

Movement which 'tucks in' to a non-root position requires that this position be accessible to it. This is done in a string based inference rule by utilizing *pairs* of strings $[u, v]$, where $u$ represents the yield of the green tree to the left of the red tree, and $v$ represents the yield of the red tree and the rest of the green tree. The blue tree (with yield $w$) tucks into the specifier of the red tree, yielding the string $uwv$. This is shown below.

$$\frac{\langle [u,v], \text{+y}^{\downarrow}.\gamma, \rangle, \vec{\phi}, \langle w, \text{-y}\rangle, \vec{\psi}}{\langle uwv, \gamma\rangle, \vec{\phi}, \vec{\psi}}\mathcal{O}(n^{2k+2})$$

The problem with this lies in the merge step which creates this pair, shown below.

$$\frac{\langle u, \text{=x}.\alpha\rangle \quad \langle v, \text{x}\rangle, \vec{\phi}}{\langle [u,v], \alpha\rangle, \vec{\phi}}\mathcal{O}(n^{2k+4})$$

With multiple feature inheritance, we need to keep track of multiple insertion points. Each additional insertion point requires two additional unique variables. The parsing complexity of (at most) $m$-multiple FI (for $m > 1$) is thus $\mathcal{O}(n^{2k+2m})$.

### *Batch FI*

We extend our pairing construct on features to relate a *sequence* of features to another: $(\delta, \text{+z})$. Here, the entire sequence $\delta$ is bequeathed to the lower head. *The order of the specifiers created via this inherited movement is given by the order of features in* $\delta$. While there is no *formal* bound on the number of simultaneous movements, **each lexicon will have an upper bound on the length of the feature sequences bequeathed in this way**. Although it would be formally natural to break the bequathed feature sequences up into a sequence of beqeathed features, the present approach directly and faithfully implements the linguist's intuitions about simultaneity of movement.

Batch FI can be implemented by extending the inference rules for single FI. We present the inference rule **MrgFI1b**$^{(2)}$, where the superscript indicates the length of the bequeathed feature sequence.

$$\frac{\langle m, \text{+w}.(\text{+x}^{\downarrow}.\text{+y}^{\downarrow}, \text{+z}).\alpha\rangle \quad \langle n, \text{-w}\rangle, \vec{\phi}, \langle o, \text{-x}.\text{-y}.\text{-z}.\beta\rangle, \vec{\psi}}{\langle mn, \alpha\rangle, \vec{\phi}, \langle o, \beta\rangle, \vec{\psi}}\mathcal{O}(n^{2k+3})$$

We see that the complexity of rule **MrgFI1b**$^{(2)}$ is the same as that of **MrgFI1b**$^{(1)}$ (the original). This is because the movers (however many there may be) are all targeting the same two positions (the left edges of *m* and of *n*). **We conclude that batch FI does not increase the parsing complexity of the MG formalism.**

## References

P. Branigan. Multiple feature inheritance and the phase structure of the left periphery. In R. Woods and S. Wolfe, editors, *Rethinking Verb Second*, chapter 7, pages 150–176. Oxford University Press, 2020. ISBN 9780198844303.

M. Fowlie and A. Koller. Parsing minimalist languages with interpreted regular tree grammars. In M. Kuhlmann and T. Scheffler, editors, *Proceedings of the 13th International Workshop on Tree Adjoining Grammars and Related Formalisms*, pages 11–20, Umeå, Sweden, Sept. 2017. Association for Computational Linguistics. URL https://aclanthology.org/W17-6202/.

D. McAllester. On the complexity analysis of static analyses. *Journal of the Association for Computing Machinery*, 49(4):512–537, 2002.

J. Michaelis. *On Formal Properties of Minimalist Grammars*. PhD thesis, Universität Potsdam, 2001.

E. P. Stabler and E. L. Keenan. Structural similarity within and among languages. *Theoretical Computer Science*, 293:345–363, 2003.

M. Stanojević. On the computational complexity of head movement and affix hopping. In R. Bernardi, G. Kobele, and S. Pogodalla, editors, *Formal Grammar*, pages 101–116, Berlin, Heidelberg, 2019. Springer Berlin Heidelberg. ISBN 978-3-662-59648-7.