# Formalizing Feature Inheritance

**Gregory M. Kobele[1] and Lei Liu[2]**
Institut für Linguistik
Universität Leipzig
[1]gkobele@uni-leipzig.de,[2]lei.liu@uni-leipzig.de

## Abstract

*Feature Inheritance* is a prominent theoretical innovation in minimalist syntax, which takes it further from the formal framework of minimalist grammars, the best understood formalism for reasoning about minimalism. Feature inheritance involves movement targeting non-root positions, as well as simultaneous movement steps. This turns out to require a formally innocuous extension to minimalist grammars, leaving strong generative capacity and worst-case parsing complexity unchanged.

## 1 Introduction

Viewing context-free base rules as structure building operations (a rule $S \rightarrow NP\ VP$ builds an $S$ out of a $NP$ and a $VP$), the transformational cycle in syntax was a principle that governed the interleaving of transformational operations with context-free structure building operations. In particular, (cyclic) transformational rules were applied only once certain categories (always S, often NP, sometimes PP) of expressions were built. In early minimalism, the transformational rule of movement was interleaved with the structure building operation of merge. However, movement could in principle apply at any time, regardless of the categorial status of its input. A mechanism of *feature inheritance*, introduced by Chomsky (2008), in effect delays transformations until a particular category is reached. Thus, minimalism with feature inheritance seems to be a return to the original conception of the syntactic cycle.

In this paper we provide a formalization of the mechanism of feature inheritance in the context of minimalist grammars (MGs), itself a formalization of Chomsky's (1995) Minimalist program. The weak generative capacity and worst-case parsing complexity of *feature inheritance* is then compared to that of vanilla MGs.

## 2 Feature Inheritance

Minimalist orthodoxy assumes a universal hierarchy of functional projections: Complementizers select Tense which selects Voice which selects Verbs. Underlying these lay terms are the abstract heads (categories) 'C', 'T', 'v' ("little-v"), and 'V' ("big-V"). A large body of work assumes a shared property between little-v and C; these two heads are said to define locality domains in the syntax (called *phases*). A basic goal expressed by Chomsky (1995) is to reduce the stipulations needed in the theory. As little-v and C share one non-trivial property already, determining whether more of their properties can be identified would potentially reduce the number of independent stipulations needed to describe the lexicon. Feature Inheritance (FI) is introduced in (Chomsky, 2008) as a way of reconciling a number of related observations with theoretical assumptions, and is made use of by little-v and C, which increases their formal similarity a great deal.

A main theoretical motivation for FI is to give a larger role to phases. Phases are said to coincide with the portion of the syntactic structure that the interfaces can refer to. In other words, they are the units that semantic and phonological interpretation are defined over. Chomsky suggests that both interfaces refer to the same units of syntactic structure. In addition, he suggests that syntactic operations (like movement and agreement) are not distributed throughout the nodes making up a phase, but are rather deferred until the last head in the phase (little-v or C). This desideratum is problematic from the perspective of orthodox analytical assumptions, as the T head is generally considered to trigger movement of and agreement with the surface subject.

One relevant observation is that only finite T heads trigger movement and agreement. A second observation is that the distribution of finite vs nonfinite T is related to the choice of C: for example, the declarative complementizer `that` selects for finite T, whereas `for` selects for non-finite T.

1. John believes that Mary smiled.
2. $*$John believes that Mary to smile.
3. $*$John hopes for Mary smiled.
4. John hopes for Mary to smile.

Chomsky's resolution to the problem is to shift the finite-nonfinite distinction over to C, making T into an underspecified tense head. Then it is C which selects for a generic T head, and it must be C which is responsible for triggering movement and agreement **on T**. FI is the mechanism by which movement triggered by a higher head targets the projection of a lower head, which allows for the idea that movement and agreement is deferred until phase heads are introduced to be realized.

C (and little-v) also permit generic movement to their edges, for example, to break long distance movement into phase-sized chunks. Thus C can trigger movement multiple times, both to its edge, as well as to the edge of the T head immediately below it. However, the movements that C now triggers are typically thought to be of two fundamentally different kinds: the movement to T is A-movement, and that targeting C is A-bar-movement. These kinds of movements have importantly different properties (pronouns can be bound after moving over them with A-movement, but not with A-bar-movement, for example), and Chomsky (1995) has proposed that movement steps between the highest A-bar position and the lowest base-merge position of expressions be invisible to various well-formedness conditions. Making the A and A-bar movements which C triggers happen simultaneously (as opposed to serially) structures the movement dependencies entered into by DPs as trees (ordered by derivational order), rather than sequences. This then eliminates the need to postulate an independent operation which deletes intermediate elements in a sequence of movement dependencies — these are no longer on a single branch of the tree.

Feature Inheritance thus paves the way for 1. phase heads to be the locus of movement and agreement triggers, and 2. a novel approach to the distinction between A and A-bar movements.

## 3 Formal background

We couch our formalization of feature inheritance in the formal framework of minimalist grammars (Stabler, 1997, 2011), an extensible and well-understood grammar formalism capable of transparently representing minimalist analyses. Minimalist grammars are a lexicalized grammar formalism, like categorial grammars, with universal grammatical rules and complex lexical entries. The categories of lexical entries take the form of lists of features, written with lower case greek letters, called *feature bundles*, where a list is a data structure where only the first element is directly accessible. Removing ('checking') the first element of a nonempty list $\alpha$ results in the remainder of the list $\alpha'$ (so $\alpha = a.\alpha'$). Features have one of two polarities (positive and negative), and come in different kinds, represented as different names ($k$, $wh$, $q$, $d$, $\ldots$). Two features `+x` and `-y` of opposite polarity *match* iff they are of the same kind (i.e. $x = y$).

A syntactic expression is either a pair $\langle w, \alpha \rangle$ consisting of a string of phonemes $w$ and a feature bundle $\alpha$ (written `w:`$\alpha$), or a term $\bullet(t_1, t_2)$, where $t_1$ and $t_2$ are syntactic expressions, and $\bullet$ is either $<$ or $>$. The *head* of a syntactic expression $t$ is $t$ itself, if a pair, and the head of $t_H$ if $t = \bullet(t_1, t_2)$, where $t_H = t_1$ if $\bullet = <$, and $t_H = t_2$ if $\bullet = >$.

Given a syntactic expression $t$, the result of checking the first feature of its head is written $t'$. When $t$ is a term, it represents a tree, and the internal nodes 'point' in the direction of the head. A trace is a pair of the empty string and the empty feature bundle, written `t`.

There are two syntactic operations, **Merge** and **Move**. **Merge** is binary, and **Move** unary. They are both restricted in their application by the feature bundles present in their arguments. The head of the first argument of both operations must be a positive feature. **Merge** applies to two expressions $t$ and $s$ just in case the heads of both have matching first features. **Move** applies to its single argument just in case this argument contains a unique leaf whose first feature matches the first feature of the head.

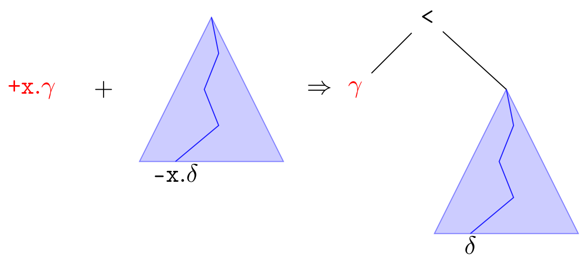

Figure 1: Merge of a complement

The output of **Merge** depends on whether its first argument is a leaf or a complex term. If a leaf $\ell$, then **Merge**$(\ell, s) = {<}(\ell', s')$, and if a proper term $t$, **Merge**$(t, s) = {>}(t', s')$, as is depicted in figures 1 and 2.

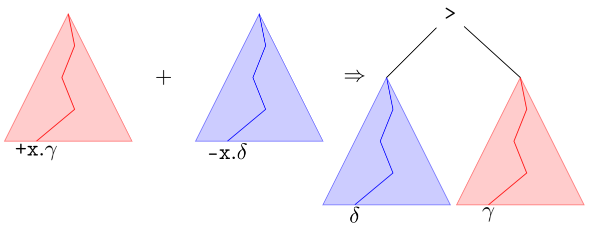

Figure 2: Merge of a specifier

**Move** replaces a subterm of the input with a trace, and so we need a notation which simplifies referring to subterms. We define *maximal projection contexts* $C[x]$ to be either a variable $x$, or a structure of one of the two forms: ${>}(C[x], t)$ or ${<}(t, C[x])$. A maximal projection context $C[x]$ is a term where $x$ occurs without any arrows pointing to it, and replacing the variable $x$ with a term $s$ is written $C[s]$. **Move** applies to $t$ iff $t = C[s]$, where $s$ is a term whose head begins with a negative feature which matches that of $t$. **Move**$(C[s]) = {>}(s', C[\mathtt{t}]')$, as is depicted in figure 3.

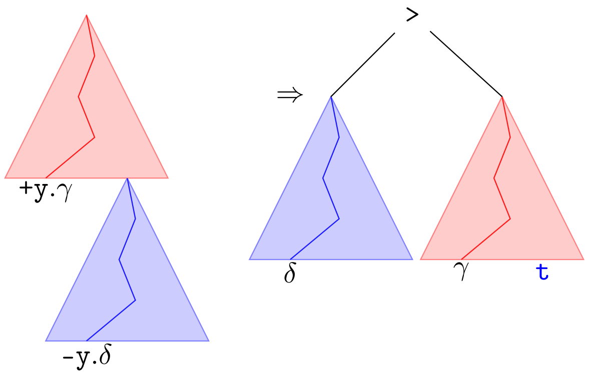

Figure 3: Movement leaves a trace

Both operations have the effect of removing features from feature bundles one at a time, and features in feature bundles are checked one at a time from left to right.

## 4 Features for Feature Inheritance

Feature inheritance diverges from minimalist grammars as they have been defined above in two ways. First, movement can target not the top of an expression, but rather some node embedded inside it. Second, two features can be checked at the same time.

To deal with the first difference, we allow positive features to take a diacritic (written: +x$^{\downarrow}$) indicating that they should target the sister node to the head. We can augment the **Move** operation so that it can deal with these new feature types. For example, given a term $t$ the first feature of the head of which begins with +y$^{\downarrow}$, whose complement $C[s]$ contains a unique term $s$ with matching first feature, write $t = D[C[s]]$. Then **Move**$(D[C[s]]) = D[{>}(s', C[\mathtt{t}])]'$. This is shown in figure 4.

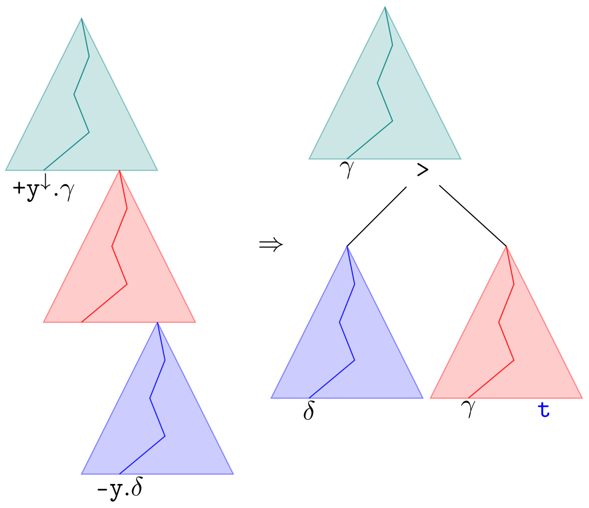

Figure 4: Inherited movement

To allow two features to be checked simultaneously, we allow feature bundles to contain not just individual features, but also pairs of features. Given a pair of features ⟨+x, +y⟩, it is intended that they be checked during the same derivational step. This allows us to write lexical items with the desired behaviour; Chomsky's C head would have feature bundle +T.⟨+k$^{\downarrow}$, +wh⟩.-C, indicating that it first merges with a TP, after which it simultaneously triggers k-movement to TP and wh-movement to itself, and then is itself a CP. Introducing two new feature types (+x$^{\downarrow}$ and $\langle f, g\rangle$) would allow for lexical feature bundles of the following forms:

1. +a.+b.+c$^{\downarrow}$.-d
2. +a.+b$^{\downarrow}$.+c$^{\downarrow}$.-d
3. +a.⟨+b, +c⟩.-d

These bundles express sequences of lexically driven derivational steps which we view as not in the spirit of Chomsky (2008), which we summarize with the following principles:

**FIUniq** Feature inheritance happens just once

**FIEarly** Feature inheritance happens immediately after the complement is merged

**FISimul** Simultaneous feature checking happens only in the context of feature inheritance

Feature bundle 1 violates the earliness principle (**FIEarly**), which requires feature inheritance to happen immediately after the complement is merged. Here, feature inheritance of $\mathsf{+c}^{\downarrow}$ was deferred until after $\mathsf{+b}$ was checked. Feature bundle 2 violates both the uniqueness principle (**FIUniq**), which requires feature inheritance to occur just once, and the earliness principle. Here, feature inheritance occurs both via $\mathsf{+b}^{\downarrow}$ and $\mathsf{+c}^{\downarrow}$, and in addition $\mathsf{+c}^{\downarrow}$ was deferred until after $\mathsf{+b}^{\downarrow}$ was checked. Feature bundle 3 violates the simultaneity principle (**FISimul**), which requires that simultaneous feature checking occur in conjunction with feature inheritance. Here, features $\mathsf{+b}$ and $\mathsf{+c}$ are checked simultaneously, neither of which involve feature inheritance. These principles conspire to enforce that lexical feature bundles are drawn from the following regular set, where $\mathbb{P} := \{\mathsf{+x} \mid x \in \mathbb{F}\}$, $\mathbb{D} := \{\mathsf{+x}^{\downarrow} \mid x \in \mathbb{F}\}$, $\mathbb{S} := \{\langle d, p\rangle \mid d \in \mathbb{D} \wedge p \in \mathbb{P}\}$ and $\mathbb{N} := \{\mathsf{-x} \mid x \in \mathbb{F}\}$:

$$(\mathbb{P}(\mathbb{D} + \mathbb{S})^?)^?\mathbb{P}^*\mathbb{N}^+$$

That is, an inheritance feature occurs only after the first positive feature, either on its own or as part of a simultaneous feature. With respect to the requirement that exactly one of the pair of simultaneous features must be an inheritance feature has a certain coherence to it. Note that with any other combination of simultaneous features (i.e. where both are of the same kind) it would be unclear how to depict the derived tree which should result after the simultaneous features are checked: as both target the same position (either the complement to the head, or the specifier of the same) one mover would need to c-command the other, from which one could reconstruct a checking order, belying the simultaneity of checking.

## 5 Implementing Feature Inheritance

A naïve implementation of inherited movement as in figure 4 is destructive, in the sense that constructing the output requires changing immediate dominance relations which held in the input. (In particular, the immediate dominance between the mother '<' of the head of the tree and the root of its complement.) For reasons discussed in the next section, this is to be avoided when possible.

Taken together, the constraints on feature bundles presented above allow for an alternative implementation of feature inheritance. As feature inheritance targets the first merged argument of the head, and takes place immediately after this argument is merged, it is simple to deal with feature inheritance during this very **Merge** step, where the top of the second argument is still accessible. This avoids the problem of destructivity, as the target position of the inherited movement has not yet been assigned an immediate dominance relation. Let $\ell$ be a lexical item whose feature bundle begins with the following two features: $\mathsf{+x}$ and $\langle \mathsf{+y}^{\downarrow}, \mathsf{+z}\rangle$. There are two cases to consider, depending on whether one mover matches both features in the pair, or whether they are matched by different movers. For the first case, let $C[s]$ be a term with first feature $\mathsf{-x}$, and where the first two features of $s$ are $\mathsf{-y}$ and $\mathsf{-z}$. Then $\mathbf{Merge}(\ell, C[s]) = \mathsf{>}(s'', \mathsf{<}(\ell'', \mathsf{>}(\mathsf{t}, C[\mathsf{t}]')))$, as is depicted in figure 5. In the other case, let $C[x, y]$ be a maximal projection context with *two* variables, and let $C[r, s]$ be a term whose first feature is $\mathsf{-x}$, and where the first features of $r$ and $s$ are $\mathsf{-y}$ and $\mathsf{-z}$ respectively. Then $\mathbf{Merge}(\ell, C[r, s]) = \mathsf{>}(s', \mathsf{<}(\ell'', \mathsf{>}(r', C[\mathsf{t}, \mathsf{t}]')))$, as is depicted in figure 6.

It only really matters that the movement steps be simultaneous if the same mover is targeted in both cases. This is because Chomsky analyzes the twin movements as creating different chains — sequential movements of the same item would simply extend a single chain. If two different movers are targeted, each is going to extend its own chain, regardless of whether this happens simultaneously or sequencially.

## 6 Complexity Analysis

Michaelis (2001) (see also Harkema (2001)) proves the equivalence between minimalist grammars and multiple context-free grammars, providing a scaffolding for future demonstrations that extensions do not increase generative capacity. To establish such an equivalence, we need to present the modified operations in inference rule format, stated over finite sequences of strings paired with feature bundles. As noted by Stanojević (2019), parsers

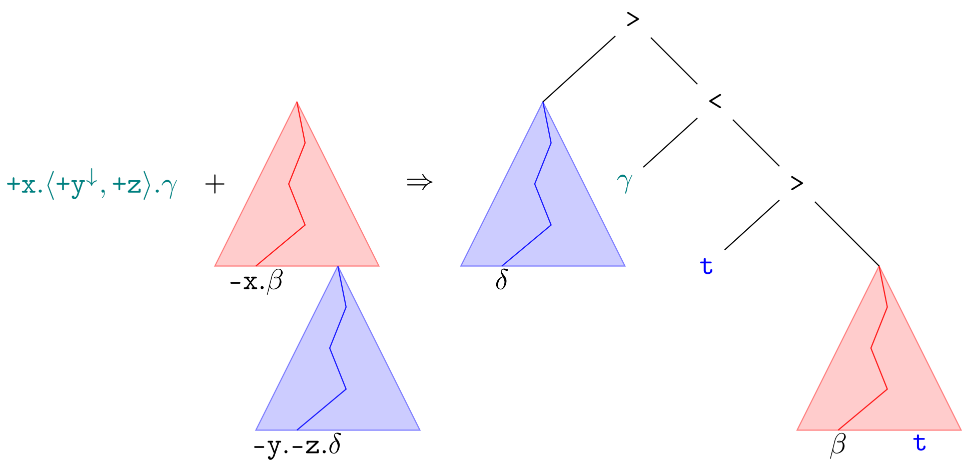

Figure 5: Feature inheritance involving a single mover

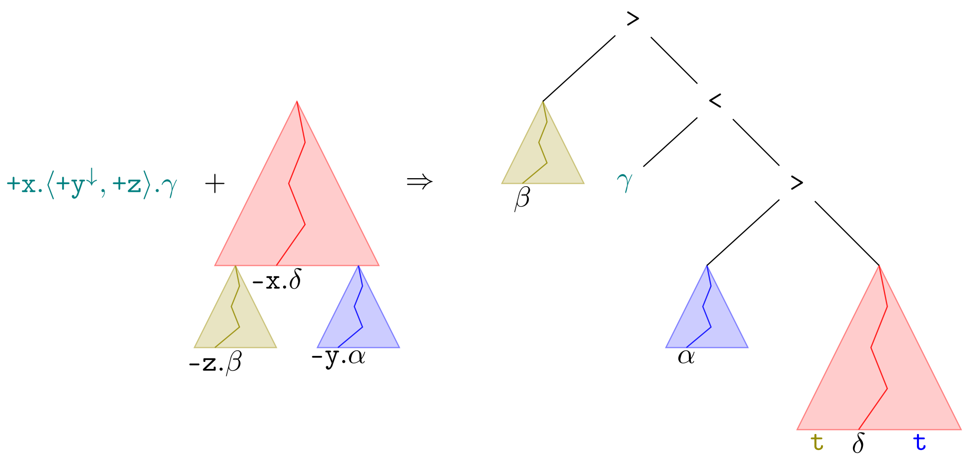

Figure 6: Feature inheritance involving two different movers

derived from this inference rule notation can have their worst-case time complexity read directly off of the rules themselves. Representing each string as a *span*, a pair of integer variables indicating what portion of the input string that string should cover, the number of distinct variables in the antecedents of a rule polynomially bounds its contribution to worst case complexity. Our revised implemention of feature inheritance only modifies the **Merge** rule (by adding to it two new cases), and so we present just these in inference rule format (see Stabler and Keenan (2003) for the others). 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.

The inference rules are given in the figures 7–12. This summation and the associated computational complexity is indicated next to the names of each of the rules above. We see that the rules **MrgFI1b** and **MrgFI2d** contribute 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.

## 7 Conclusion

We have presented a formalization of Chomsky's ((2008)) mechanism of feature inheritance, which has played an important role in minimalist syntactic theory over the intervening nearly two decades. It is formally innocuous: it increases neither the weak generative capacity nor the worst case time complexity of the MG formalism.

Another route to this result is to simply note that lexica containing the new lexical items with feature bundles of the form $\mathtt{+x.+y^{\downarrow}}.\alpha$ and $\mathtt{+x.\langle +y^{\downarrow}, +z\rangle}.\alpha$ can be transformed into strongly and weakly equivalent lexica containing only standard feature bundles: given a lexical item $\mathtt{u{:}+x.\langle +y^{\downarrow}, +z\rangle}.\alpha$, replace it with a lexical item $\mathtt{u{:}+x'.+z}.\alpha$, where $x'$ is a fresh feature name, and for every lexical item $\mathtt{v{:}}\beta\mathtt{.-x}.\gamma$ add to the lexicon the new lexical item $\mathtt{v{:}}\beta\mathtt{.+x.-x'}.\gamma$. This transformation simply pushes down the inherited features onto the lexical items which will ultimately inherit them, and ensures that they subsequently combine with their benefactors.

Like many proposals in minimalism, the substance of this one seems to lie in things not so easily measured, like: 1. providing a formal foundation for the distinction between movement types: two independent chains branching off of a single element, one of which c-commands the other, gives a scaffolding over which different clusters of properties can be assigned to each, and 2. giving a formal unification of lexical items of a certain type: $\forall\mathtt{x.+x.\langle +}\phi^{\downarrow}\mathtt{, +epp\rangle.-x'}$ is the general format for phasal heads, where *epp* is a feature permitting movement, and $\phi$ are agreement related features (and we have used object-level quantification over feature names to express polymorphism, and $x'$ is the next category up in the extended projection of $x$).

## References

Noam Chomsky. 1995. *The Minimalist Program*. MIT Press, Cambridge, Massachusetts.

Noam Chomsky. 2008. On phases. In Robert Freidin, Carlos P. Otero, and Maria Luisa Zubizarreta, editors, *Foundational Issues in Linguistic Theory*, pages 133–166. MIT Press, Cambridge, Massachusetts.

Meaghan Fowlie and Alexander Koller. 2017. Parsing minimalist languages with interpreted regular tree grammars. In *Proceedings of the 13th International Workshop on Tree Adjoining Grammars and Related Formalisms*, pages 11–20, Umeå, Sweden. Association for Computational Linguistics.

Henk Harkema. 2001. *Parsing Minimalist Languages*. Ph.D. thesis, University of California, Los Angeles.

Jens Michaelis. 2001. *On Formal Properties of Minimalist Grammars*. Ph.D. thesis, Universität Potsdam.

Edward P. Stabler. 1997. Derivational minimalism. In Christian Retoré, editor, *Logical Aspects of Computational Linguistics*, volume 1328 of *Lecture Notes in Computer Science*, pages 68–95. Springer-Verlag, Berlin.

Edward P. Stabler. 2011. Computational perspectives on minimalism. In Cedric Boeckx, editor, *The Oxford Handbook of Linguistic Minimalism*, Oxford Handbooks in Linguistics, chapter 27, pages 616–641. Oxford University Press, New York.

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

Miloš Stanojević. 2019. On the computational complexity of head movement and affix hopping. In *Formal Grammar*, pages 101–116, Berlin, Heidelberg. Springer Berlin Heidelberg.

$$\frac{\langle m, \mathsf{+x.(+y^{\downarrow}, +z)}.\alpha\rangle \qquad \langle n, \mathsf{-x}\rangle, \vec{\phi}, \langle o, \mathsf{-y.-z}\rangle, \vec{\psi}}{\langle omn, \alpha\rangle, \vec{\phi}, \vec{\psi}}\ \text{MrgFI1a} \qquad \mathcal{O}(n^{2k+2})$$

The inference rule **MrgFI1a** describes the situation where there is a single mover, for whom this is the last movement step, and therefore is pronounced in its highest position.

Figure 7: **MrgFI1a**

$$\frac{\langle m, \mathsf{+x.(+y^{\downarrow}, +z)}.\alpha\rangle \qquad \langle n, \mathsf{-x}\rangle, \vec{\phi}, \langle o, \mathsf{-y.-z}.\beta\rangle, \vec{\psi}}{\langle mn, \alpha\rangle, \vec{\phi}, \langle o, \beta\rangle, \vec{\psi}}\ \text{MrgFI1b} \qquad \mathcal{O}(n^{2k+3})$$

The inference rule **MrgFI1b** describes the situation where the single mover has features left over, and thus continues moving.

Figure 8: **MrgFI1b**

$$\frac{\langle m, \mathsf{+x.(+y^{\downarrow}, +z)}.\alpha\rangle \qquad \langle n, \mathsf{-x}\rangle, \vec{\phi}, \langle o, \mathsf{-y}\rangle, \vec{\psi}, \langle p, \mathsf{-z}\rangle, \vec{\chi}}{\langle pmon, \alpha\rangle, \vec{\phi}, \vec{\psi}, \vec{\chi}}\ \text{MrgFI2a} \qquad \mathcal{O}(n^{2k+1})$$

The inference rule **MrgFI2a** describes the situation where there are two movers, for both of which this is the last movement step, and therefore are pronounced in their highest positions. In the result, we see that the phonetic part *o* of the tucking-in mover is sandwiched between the head *m* selecting the complement, and the pronunciation *n* of this complement.

Figure 9: **MrgFI2a**

$$\frac{\langle m, \mathsf{+x.(+y^{\downarrow}, +z)}.\alpha\rangle \qquad \langle n, \mathsf{-x}\rangle, \vec{\phi}, \langle o, \mathsf{-y}.\beta\rangle, \vec{\psi}, \langle p, \mathsf{-z}\rangle, \vec{\chi}}{\langle pmn, \alpha\rangle, \vec{\phi}, \langle o, \beta\rangle, \vec{\psi}, \vec{\chi}}\ \text{MrgFI2b} \quad \mathcal{O}(n^{2k+2})$$

The inference rule **MrgFI2b** describes the situation where there are two movers, but the first one continues moving.

Figure 10: **MrgFI2b**

$$\frac{\langle m, \mathsf{+x.(+y^{\downarrow}, +z)}.\alpha\rangle \qquad \langle n, \mathsf{-x}\rangle, \vec{\phi}, \langle o, \mathsf{-y}\rangle, \vec{\psi}, \langle p, \mathsf{-z}.\gamma\rangle, \vec{\chi}}{\langle mon, \alpha\rangle, \vec{\phi}, \vec{\psi}, \langle p, \gamma\rangle, \vec{\chi}}\ \text{MrgFI2c} \quad \mathcal{O}(n^{2k+2})$$

The inference rule **MrgFI2c** describes the situation where there are two movers, but the second one continues moving.

Figure 11: **MrgFI2c**

$$\frac{\langle m, \mathsf{+x.(+y^{\downarrow}, +z)}.\alpha\rangle \qquad \langle n, \mathsf{-x}\rangle, \vec{\phi}, \langle o, \mathsf{-y}.\beta\rangle, \vec{\psi}, \langle p, \mathsf{-z}.\gamma\rangle, \vec{\chi}}{\langle mn, \alpha\rangle, \vec{\phi}, \langle o, \beta\rangle, \vec{\psi}, \langle p, \gamma\rangle, \vec{\chi}}\ \text{MrgFI2d} \quad \mathcal{O}(n^{2k+3})$$

The inference rule **MrgFI2d** describes the situation where there are two movers, and both continue moving.

Figure 12: **MrgFI2d**