A Nanosyntactic Account of Syncretism

in French Personal Pronouns

Sandrine Guilbert

A Thesis

In the Department

of

School of Graduate Studies

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Arts (Individualized Program in Classics, Languages and Linguistics)

Concordia University

Montréal, Québec, Canada

November 2025

CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: **Sandrine Guilbert**

Entitled: **A Nanosyntactic Account of Syncretism in French Personal Pronouns**

and submitted in partial fulfillment of the requirements for the degree of

Master of Arts (Individualized Program in Classics, Languages and Linguistics)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

________________________________ *Chair*
Chair name

________________________________ *Examiner*
Dr. Charles Reiss

________________________________ *Examiner*
Dr. Sigwan Thivierge

________________________________ *Supervisor*
Dr Daniela Isac

Approved by ________________________________
Dr. Felice Yuen, Graduate Program Director

________ **2025** ________________________________
Dr. Geoff Dover, Interim Dean of Graduate Studies

Abstract

A Nanosyntactic Account of Syncretism in French Personal Pronouns

Sandrine Guilbert

This thesis investigates the phenomenon of syncretism, defined as a systematic mismatch between morphophonological forms and their associated syntactic feature content. The specific focus of this thesis is the French pronominal paradigm, which encodes multiple dimensions simultaneously—case, gender, and class (strong, weak, clitic). Several forms, such as *lui* and *elle*, appear in more than one morphosyntactic environment, raising questions about how such syncretisms should be represented in the grammar. Do syncretic forms correspond to a single lexical entry or multiple entries? How can similarities and differences between syncretic items be captured, and what does this reveal about the interface between morphology and syntax? To address these questions, this thesis adopts a nanosyntactic perspective, building on cartographic approaches to the structure of Determiner Phrases (DPs). The analysis explores the internal feature composition of pronouns and uses tools such as pointer-based lexicalization and branching lexical trees to account for French syncretism patterns. This study contributes to our understanding of the relationship between syntax and morphology, as well as the principles that govern the lexicalization of morphosyntactic structures.

Contents

Chapter 1

Introduction

The general goal of this thesis is to contribute to a better understanding of the relationship between syntax and morphology by investigating cases of syncretism, i.e. situations in which there is a mismatch between the morphophonological form of a word and its syntactic feature content.

A well-known example of syncretism is the English pronoun *you*, which is used both in singular and plural contexts. In other words, the morphophonological form *you* is ambiguously mapped to two sets of syntactic features- either second person singular or second person plural.

Crucially, for a form to be syncretic between two or several different syntactic feature contents, their respective contents have to be related. In the case of *you*, both the singular and plural forms share the feature of second person, referring to the addressee. By contrast, lexical homophony—such as the word *bank*, which can refer either to a financial institution or the edge of a river—involves a single morphological form associated with distinct semantic contents, rather than distinct syntactic features. In fact, the syntactic features of bank under both interpretations are the same, i.e. noun, singular and third person. Therefore, such cases are not considered syncretic in the technical sense adopted here.

The cases of syncretism that are the focus of this research are instances of systematic, rather than accidental, syncretism. A good example of both types can be found in the German verbal paradigm, which displays recurring syncretism across person and number combinations in the present tense. Specifically, the first and third person plural verb endings, marked in blue, are syncretic (-*en*), as are the third person singular and second person plural forms, marked in red (-*t*).

Person	**Singular (SG)**	**Plural (PL)**
1st Person	kauf-e	kauf-en
2nd Person	kauf-st	kauf-t
3rd Person	kauf-t	kauf-en

Table 1.1: Present tense paradigm of *kaufen* ‘to buy’ in German

The former is generally considered a case of systematic syncretism, as it appears consistently across different tenses and verbs, including in the past tense, as shown below. In contrast, the latter is typically viewed as accidental syncretism, since the pattern is not preserved across tenses; for instance, the syncretism between third-person singular and second-person plural does not hold in the past tense (Hein et al., 2021).

Person	Singular (SG)	Plural (PL)
1st Person	kauf-t-e	kauf-t–en
2nd Person	kauf-te-st	kauf-t-et
3rd Person	kauf-t-e	kauf-t-en

Table 1.2: Past tense (Präteritum) paradigm of *kaufen* 'to buy' in German

The more specific goal of this thesis is to focus on a particular case of syncretism in the pronominal paradigm in French. This paradigm is multidimensional, as it encodes several morphosyntactic features simultaneously: case (nominative, accusative, dative), gender (masculine, feminine), and class (strong, weak, and clitic pronouns). Table 1.3 below presents an overview of the relevant forms:

	Strong M.	Weak M.	Clitic M.	Strong F.	Weak F.	Clitic F.
Nom	lui	il	–	elle	elle	–
Acc	lui	–	le	elle	–	la
Dat	(à lui)	–	lui	(à elle)	–	lui

Table 1.3: French third-person pronoun

As shown in table 3, several forms appear in more than one morphosyntactic environment. For example, *lui* is used as a strong masculine form in the nominative and accusative case, and as a clitic dative (for both genders), while *elle* appears across nominative and accusative cases in both strong and weak paradigms. Therefore, both *lui* and *elle* are syncretic forms, as they display a mismatch between their morphophonological form and their feature content.

These patterns raise important questions about the underlying syntactic features and the principles governing this syncretism.

(i) How can we capture the fact that two syntactically different sets of features are mapped onto the same morphological form? Should the lexicon include one lexical entry for all instances of *lui* or different ones?

(ii) How can we capture both the similarities and the differences between two syncretic forms?

(iii) What is the dividing line between morphology and syntax? Are these two systems autonomous modules of grammar, or do they operate according to the same underlying principles?

(iv) What syntactic features are responsible for the grammatical behaviour of pronouns in general, and where are these features located in the syntactic tree of a nominal constituent?

(v) How can we account for the observed syncretisms in the French pronominal system?

The remainder of this thesis is organized as follows. Chapter 2 presents the theoretical background, covering Cartography, basic assumptions in Nanosyntax, and the principles governing lexicalization, including the Superset Principle, the Elsewhere Principle, Cyclic Override, and the Lexicalization Algorithm. Chapter 3 introduces syncretism, providing historical context and outlining how syncretism is analyzed within Nanosyntax. Chapter 4 develops the feature sequence relevant to pronominal structures, discussing case, distinctions between strong/weak/clitic pronouns, φ-features, and concluding with the final f-seq for pronouns. Chapter 5 presents the French pronominal data and formulates the empirical problem they pose. Chapter 6 introduces pointers as a theoretical tool and shows how they account for suppletion, idioms, and syncretism in multidimensional paradigms. Chapter 7 turns to branching L-trees following Blix, developing structural representations necessary for the analysis. Chapter 8 proposes a solution to the French patterns, deriving forms such as *lui* and discussing the implications. Finally, Chapter 9 concludes.

Chapter 2

Theoretical background

2.1 Cartography

The theoretical framework adopted in this thesis is that of Nanosyntax (Baunaz et al., 2018; Caha, 2009; Caha et al., 2024; Starke, 2009, 2011), which builds on many of the core assumptions and empirical findings of the cartographic tradition. Cartography is a research program within generative syntax, grounded in the foundational work of Belletti (2004); Benincà (1988); Cinque et al. (1990); Cinque (1999, 2002); Cinque and Rizzi (2008); Rizzi (1997, 2004).

At its core, Cartography is based on the belief that syntactic structures are far more articulated than was previously assumed. For example, Rizzi (1997) decomposes the CP layer into multiple functional projections—ForceP, TopP, FocP, and FinP—arguing that Italian clearly distinguishes separate syntactic positions for each of these. Similarly, Cinque et al. (1990) propose that there are many adverbial positions which are hierarchically ordered, based on cross-linguistic evidence showing that certain types of adverbs appear in fixed relative positions. This led to the assumption that adverbs are merged in specifiers positions, rather than as adjuncts. The fixed ordering of adverbs follows from a fixed ordering of these functional heads. In the nominal domain, cartographic work led to positing a rich functional structure above the NP, including projections for case (Giusti, 1993), number (Ritter, 1988, 1991), and person features (Bonet, 1991), among others. These studies reinforce the idea that the fine-grained architecture of syntax extends to all grammatical domains.

A central tenet of Cartography is the One Feature–One Head hypothesis, which posits that each syntactico-semantic feature corresponds to a distinct functional head in the syntactic hierarchy (Cinque and Rizzi, 2008, pp. 50). This hypothesis drives the cartographic enterprise to uncover the full array of functional projections by analyzing the syntactic behaviour of fine-grained features across languages.

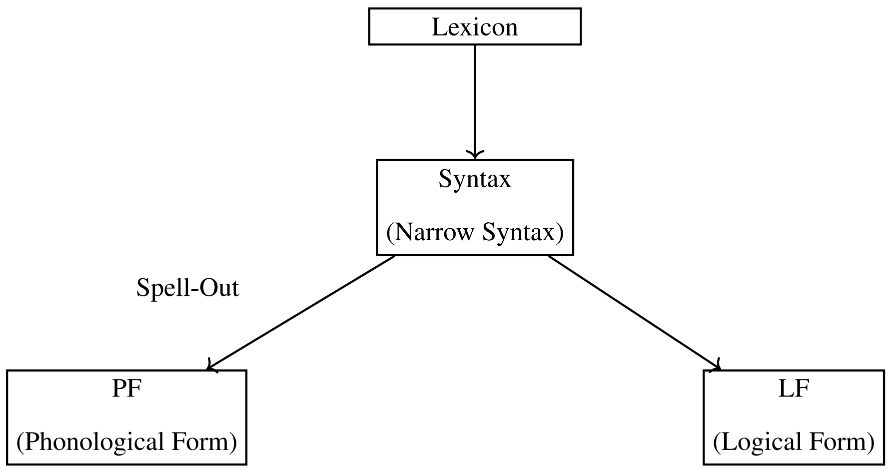
One of the main goals of the cartographic program is to determine which aspects of meaning are grammatical—i.e., encoded in syntax—as opposed to exclusively conceptual or pragmatic, i.e. not relevant to the syntactic module. This involves identifying the precise set of functional projections that make the clausal structure and mapping out their hierarchical organization, which is assumed to be universal. The spine of functional projections is thus shared across languages, even if not always overtly realized in each of them (Cinque, 1999; Rizzi, 1997; Cinque and Rizzi, 2008). This assumption aligns with the Uniformity Principle, which holds that, "in the absence of compelling evidence to the contrary, assume languages to be uniform, with variety restricted to easily detectable properties of utterances" (Chomsky, 2001, 2).

The universality of functional head order is supported by cross-linguistic evidence, much of it uncovered through comparative syntax. For instance, Cinque (1999) shows that adverbs in multiple unrelated languages—such as Italian, English, and Chinese—follow the same underlying relative order, even when they surface in different linear positions due to movement or morphological differences. This suggests that the underlying functional sequence is consistent across languages, regardless of surface variation.

One of the central goals of current linguistic theory, consistent with this view, is to account for cross-linguistic variation by reducing it to differences in the lexicon. This idea is known as the Borer-Chomsky Conjecture (Borer, 1984; Baker, 2008), which holds that all parametric variation stems from the feature composition of lexical items, particularly functional heads. On this view, syntactic operations and structural configurations are invariant across languages. Variation arises solely from the interaction of these universal principles with language-specific lexical entries. This approach therefore situates variation within the lexicon, where crosslinguistic variation is indisputable.

The model of grammar assumed by most generativists, including cartographers, is that of the Y-model Chomsky (1965, 1981, 1986, 1995); Rizzi (2013). There is a lexicon which is presyntactic. This lexicon contains lexical and functional items which are in turn made up of abstract features, such as *singular,past, number*...These feature bundles are then fed to the syntax which computes them and sends the resulting structure to the interfaces for interpretation at PF and LF.

(1) Y-Model of language (Chomsky, 1965, 1981, 1995)



In summary, the cartographic framework posits that syntactic structure is composed of a universal sequence of atomic features. This universal spine underlies all grammatical structures, including clauses, noun phrases, and adpositional phrases. The empirical methodology of Cartography is largely comparative, relying on a detailed analysis of linguistic data from a wide range of languages in order to map out the universal syntactic hierarchy. This detailed mapping of structure serves as the foundation for more fine-grained theories like Nanosyntax, which adopt and extend the cartographic tradition.

2.2 Basic Assumptions in Nanosyntax

Nanosyntax grew from the cartographic tradition and inherited its core assumptions. Like Cartography, it holds that functional structure is highly articulated and aims to identify the full set of atomic syntactic elements along with their universal hierarchical organization—what is referred to as the functional sequence (fseq) (Baunaz et al., 2018, 15). Despite this shared foundation, Nanosyntax diverges from Cartography in several key ways, most notably in its treatment of lexical insertion and the relationship between morphology and syntax. Much of the discussion in this chapter draws on the work of Baunaz et al. (2018) & Caha et al. (2024).

One major distinction between the two frameworks is that Nanosyntax adopts a late insertion model of syntax, in which lexicalization is a strictly post-syntactic operation. Unlike traditional generative models and unlike cartographic approaches—where lexical items are inserted into the derivation and manipulated by syntax—Nanosyntax posits that the syntactic structure is built independently of the lexicon, and lexical items are inserted after the syntax. This approach offers two significant advantages. First, it makes the linguistic system more modular, since the syntactic component does not handle phonological or conceptual information (which are part of lexical items). Second, it maintains the universality of the syntax: since lexical items are language-specific, postponing their insertion allows the syntax to operate only with universal grammatical features and according to universal principles.

A second way in which Nanosyntax differs from Cartography is that Nanosyntax treats morphology and syntax as one and the same: word formation and sentence structure are built in a single, unified system. This follows from a few key assumptions. First, just like cartographic approaches, Nanosyntax assumes that each syntactic head corresponds to a single grammatical feature. However, a single morpheme is typically associated with multiple grammatical features. For example, the English pronoun *her* must minimally be associated with *feminine*, *singular*, and *case* feature. As a result, it is common for a single lexical item to span multiple syntactic heads.

Under the One Feature–One Head assumption—adopted in Nanosyntax from Cartography—the system must permit phrasal lexicalization, whereby a single morpheme can lexicalize not only an individual head but an entire phrase. That is, phonological material may be inserted at a phrasal node that dominates one or more terminal nodes (Starke, 2002). If each syntactic head encodes a single feature, but morphemes often correspond to multiple features, then spell-out must target not only individual heads but also larger constituent structures. This mechanism enables the system to bridge the mismatch between fine-grained syntactic structure and coarser morphological realization.

Another important innovation of Nanosyntax is its rejection of feature bundles in the lexicon. Instead, lexical items are assumed to contain hierarchically organized features. The mapping between the lexicon and syntax then becomes a question of matching two structures: the L(exical)-tree and the S(yntactic)-tree, which I will explain in more detail later in this chapter.

Finally, Nanosyntax adopts the Borer–Chomsky Hypothesis, which holds that cross-linguistic variation arises solely from differences in the lexicon. But if syntax is universal and lexical insertion occurs post-syntactically, how can the theory account for syntactic differences across languages? The answer lies in the architecture of lexicalization: variation emerges through bottom-up cyclic spellout and rescue operations, which determine how syntactic structures are mapped onto language-specific lexical entries. These operations allow for linguistic variation while preserving the universality of syntactic computation.

In the following subsection, I outline how lexicalization proceeds within the nanosyntactic framework, showing how the principles of the theory determine the mapping between syntactic structure and vocabulary items, and provide a step-by-step explanation of the lexicalization algorithm.

2.3 Lexicalization and its constraints

To restate the key points discussed above, Nanosyntax assumes that the inputs to the syntactic component are individual atomic features provided by Universal Grammar. Syntax builds structure by merging these features in a fixed, universal order—referred to as the functional sequence, or fseq. Crucially, each feature must be spelled out before the derivation can proceed. Once a successful lexicalization has occurred, the next feature is merged, and the process continues recursively.

As discussed in section 2.2, lexical items in Nanosyntax are not associated with unordered feature bundles. Rather, they correspond to hierarchically structured features that resemble syntactic trees. These structures—referred to as L-trees (lexical trees)—are not generated by the syntax but are stored in the lexicon. In contrast, the trees constructed by the syntactic derivation are called S-trees (syntactic trees).

An L-tree constitutes one of three components of a lexical entry, alongside its phonological form and its conceptual (or semantic) content. Whether a given lexical item can be inserted at a particular point in the derivation depends on the structural compatibility between its L-tree and the current S-tree. Two principles govern this process: the Superset Principle and the Elsewhere Principle.

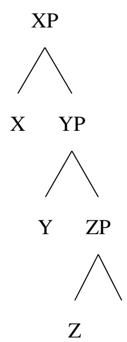
2.3.1 Superset Principle

The Superset Principle (Starke, 2009; Caha, 2009) determines when an L-tree can be inserted to lexicalize an S-tree. According to this principle, an L-tree is a valid match if the S-tree is a subtree of it—meaning the L-tree must contain all the nodes of the S-tree in the correct hierarchical order, though it may also include additional, unused structure. This mechanism enables a single lexical entry to apply in multiple syntactic environments, thereby accounting for patterns of syncretism, which I return to in Chapter 3.

(2) *Superset Principle:* A lexically stored tree L matches a syntactic node S iff L contains the syntactic tree dominated by S as a subtree. (Starke 2009)

To give an example of the Superset principle, consider the L-tree in (3):

(3) *L-tree*

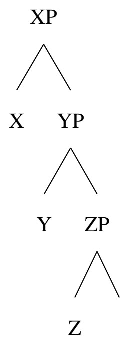


In line with the Superset Principle, this L-tree is a valid match for precisely the three S-trees shown in (4), (5), and

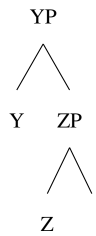
(6), and for no others—since a structure with three hierarchical layers can have only three distinct subtrees.

Matching S-trees for (3)

(4)



(5)

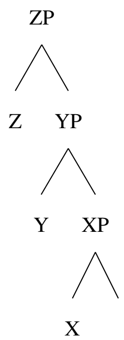


(6) ZP

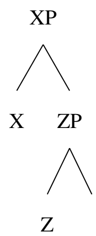
Conversely, none of the trees in (7), (8), and (9) qualify as valid matches for (3), even though they share the same features. Tree (7) differs in hierarchical relationships from (3): XP and ZP are inverted. Tree (8) lacks an intermediate node (i.e., YP). Tree (9) includes the XP node, but does not contain its daughters (YP and ZP), thereby failing to constitute a subtree of (3).

Non-matching S-trees for (3)

(7)



(8)



(9) XP

The Superset Principle determines which lexical items are eligible matches for a given syntactic structure. However, when multiple items satisfy this condition simultaneously, additional constraints are necessary to select the most appropriate match. The Elsewhere Principle serves this purpose.

2.3.2 Elsewhere Principle

When many lexical items match a syntactic structure, the grammar must decide which item to insert. This selection is governed by the Elsewhere Principle—a term originally coined by Kiparsky (1973)—which, in the context of Nanosyntax, is understood as a preference for the most specific applicable item. That is, the system selects the L-tree with the least amount of unnecessary structure. This principle is also known as the *Best Fit Principle* or *Minimize Junk*.

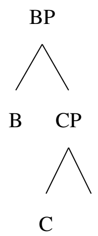
(10) *Elsewhere Principle:*

If more than one L-tree can lexicalize the same S-tree (via the Superset Principle), the L-tree with the least amount of superfluous structure is selected. (Baunaz et al., 2018, 50)

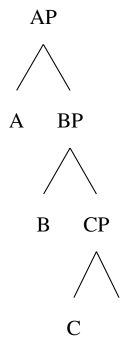
Consider the following case. Suppose there is an S-tree such as the one in (11) and there are two matching L-trees for this structure: (12) and (13). Both L-trees are valid matches under the Superset Principle, but the Elsewhere Principle selects the more specific entry—(13)—as it contains less unnecessary structure (i.e. it does not have the unnecessary AP layer).

An S-tree and two matching L-trees

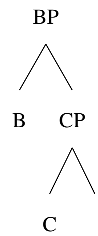
(11) *S-tree*



(12) *L-tree #1*



(13) *L-tree #2*



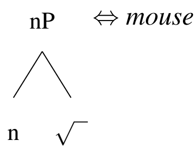
2.3.3 Cyclic Override

So far, I have introduced how the grammar chooses between competing lexical items at a single point in the derivation. But lexicalization is not confined to one step at the end of the derivation—it proceeds cyclically. After each Merge operation, the grammar checks whether the newly formed structure can be matched by an L-tree in the lexicon.

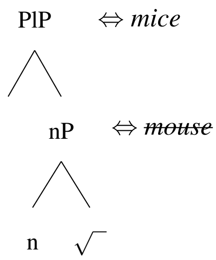
This cyclical process is governed by the Cyclic Override Principle (also referred to as the Biggest Wins Theorem), which states that "previous lexicalizations are overridden or cancelled out by later lexicalizations" (Baunaz et al., 2018, 54). In other words, if a larger chunk of structure becomes lexicalizable at a later stage in the derivation, it will replace smaller segments that may have already been successfully lexicalized. The system thus continuously seeks the largest possible match in the lexicon, even at the cost of retracting earlier insertions.

An example of cyclic override can be seen in the derivation of the irregular English plural *mice*. The structure for the singular form is first constructed and lexicalized as *mouse*, as shown in (14). When the plural feature is subsequently merged, it is added on top of the structure already spelled out as *mouse*. At this point, the grammar has access to the lexical entries *mouse* and *-s*. However, because the lexicon also contains a single entry that lexicalizes the entire structure [mouse + Pl], the exponent *mice* is inserted instead, overriding *mouse*.

(14) *Mouse* singular



(15) *Mouse* Plural



This dynamic interaction between syntax and the lexicon is known as the spellout loop: after each Merge operation, the derivation enters a loop in which it attempts to spell out the current structure using stored lexical items. If no match is found, the derivation cannot continue by simply merging more features. Instead, the syntax attempts to rescue the structure by applying a predefined hierarchy of operations.

2.4 The Lexicalization Algorithm

When a new feature (F) is merged in the syntax, the system immediately tries to lexicalize the resulting structure with a matching lexical item. If no match is found, the derivation doesn't crash immediately—instead, the grammar initiates a structured sequence of repair strategies to rescue the derivation. These steps unfold in a specific order, forming what is known as the lexicalization algorithm, outlined in (16).

(16) a. Merge F and lexicalize.

b. If fail, try a spec-to-spec movement and lexicalize.

c. If fail, try a movement of the complement of the newly inserted feature and lexicalize

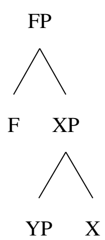
d. If fail, go back to the previous cycle, and try the next option for that cycle.

e. If fail, spawn a new derivation providing feature X and merge that with the current derivation, projecting feature X to the top node.

From Caha (2024), based on Starke (2018)

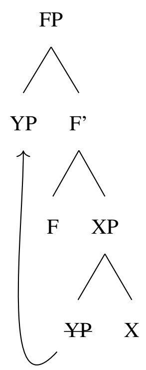
In what follows, I illustrate how each of these operations works. Take, for instance, the structure in (17), where F has just been merged. The first step, as per (16a), is to attempt lexicalization of the newly formed constituent.

(17) Merge F



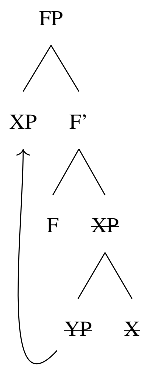
If there is no matching lexical item for the S-tree in (17). The next step is to attempt (16b): *Spec-to-spec movement*. The specifier of XP (i.e., YP) moves to the specifier position of FP (18):

(18) Spec-to-spec movement



If no lexical item matches the resulting structure after spec-to-spec movement, the next rescue operation is *Comp-to-spec movement*(16c), where the complement of F (i.e., XP) moves to the specifier position of FP (19):

(19) Comp-to-spec Movement



If the derivation fails to lexicalize the structure even after attempting both Spec-to-Spec and Comp-to-Spec movement, the algorithm proceeds to backtrack (16d). This involves returning to the previous cycle, before merging F, and trying the next step in the algorithm. The system essentially revisits earlier stages of the derivation to see whether a different structure might allow for successful lexicalization at the current stage. This step in the lexicalization algorithm has been argues again by Blix (2021) which I will revisit in Chapter 7.

Finally, if all lexicalization strategies fail—including after backtracking—the grammar creates a new derivational workspace. Within this secondary space, a structure headed by the problematic feature (F) is constructed. This new derivation is then merged with the original one, with a new feature (F) projecting to the top of the combined structure. Often referred to as "spawning" a new derivation, this operation represents the grammar's ultimate mechanism for salvaging otherwise unlexicalizable structures, while still preserving the intended hierarchical feature ordering. This is the process responsible for deriving pre-elements, such as prepositions or prefixes.

In this chapter, I have established the theoretical foundations essential for the analyses to come by introducing the core principles of Nanosyntax, including the Superset Principle, the Elsewhere Principle, Cyclic Override, and the lexicalization algorithm. With this groundwork in place, the following chapter shifts focus to the phenomenon of syncretism, beginning with an overview of its historical treatment in linguistic theory and then exploring how Nanosyntax explains syncretic patterns, especially through its structural constraints and predictions regarding morphological paradigms.

Chapter 3

Syncretism

3.1 History of syncretism

The term syncretism was first introduced by Pott (1836). Initially, it carried a diachronic connotation, referring to the historical merger of two previously distinct forms. In such cases, the shared morpho-phonological form of two syncretic form was explained in terms of a historical change. Beginning in the 20th century, however, the term began to be used in a synchronic context. For structuralist linguists such as Hjelmslev (1935) and Jakobson (1971), syncretism did not presuppose a contrast between an earlier and a later form, but between an underlying form and a surface one.

In this thesis, I adopt a synchronic view of syncretism, as understood within the Nanosyntactic framework. Under this approach, syncretism refers to instances where two or more distinct grammatical categories are realized by a single phonological form in one syncretic grammar.

Syncretism has received considerable attention in the Nanosyntactic literature, as it represents an area where the framework has shown particular explanatory strength. Nanosyntax offers a systematic approach to accounting for syncretic patterns by deriving them from structural containment and the principles of lexical insertion. This makes it especially well-suited to capturing cross-linguistic generalizations about syncretism that might appear arbitrary under other frameworks. The phenomenon has been explored extensively in a range of studies (Caha, 2009, 2010, 2013; Taraldsen, 2009; Pantcheva, 2011; Clercq, 2013; Rocquet, 2013; Øystein A. Vangsnes, 2014; Baunaz, 2015, 2016).

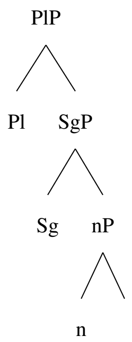
3.2 Syncretism in Nanosyntax

Syncretism has received considerable attention in nanosyntactic research. The framework offers a structure-based account of syncretic patterns, deriving them from the interaction between fine-grained syntactic representations and constraints on lexical insertion. Syncretisms and the feature structures underlying them have been extensively explored in the literature (Caha, 2009, 2010, 2013; Taraldsen, 2009; Pantcheva, 2011; Clercq, 2013; Rocquet, 2013; Øystein A. Vangsnes, 2014; Baunaz, 2015, 2016).

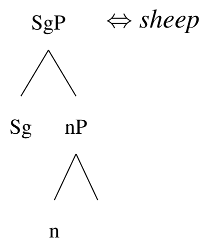
In Nanosyntax, syncretism arises when a single lexical item can be inserted in multiple syntactic environments—i.e., when one L-tree lexicalizes more than one S-tree. For example, the English noun *sheep* is syncretic in its singular and plural forms. This is accounted for by the assumption that the plural feature structurally contains the singular one. As a result, the L-tree for *sheep*, shown in (20), is an exact match for the plural S-tree in (21), but it can also be inserted into the singular S-tree in (22), since the latter is a proper subtree of the former. This pattern follows from the Superset Principle.

An L-tree and two matching S-trees for Sheep

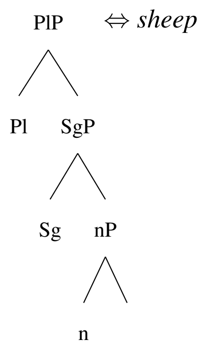
(20) *L-tree for sheep*



(21) *S-tree #1*



(22) *S-tree #2*



This type of structural containment—where one form (e.g., the plural) is built upon another (e.g., the singular)—is often supported by morphological evidence, known as *morphological containment* or *nesting*, observed across languages (Bobaljik, 2007, 2012). For instance, the plural marker is often added onto a singular form, e.g. the regular English plural nouns (*"sg noun"* + *-s*).

Under this Nanosyntax, approach to syncretism, no additional mechanisms are required to account for syncretic forms. In contrast, frameworks like Distributed Morphology often explain syncretism through post-syntactic operations such as impoverishment rules or feature underspecification (Keine and Müller, 2019). Nanosyntax, by comparison, derives syncretism as a natural consequence of hierarchical feature structure and the lexicalization algorithm.

Chapter 4

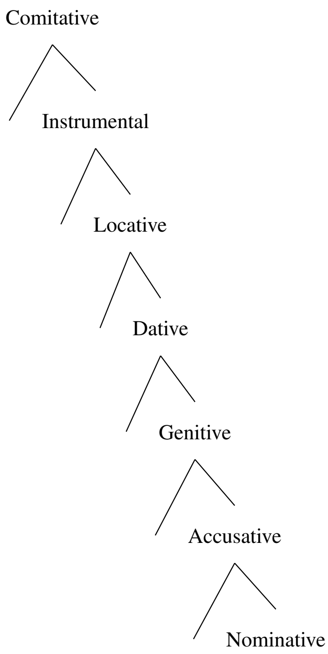
Feature structure of pronouns

As outlined in Chapter 2.2, the framework assumes that all features are hierarchically ordered according to a universal functional sequence. In what follows, I examine the features that make up the functional sequence of pronouns, focusing on three domains in particular: case features, class-defining features (which determine whether a pronoun is weak, clitic, or strong), and φ-features. I then propose a (non-exhaustive) functional sequence that will serve as the basis for my analysis in Chapter 8.

The structure of the functional sequence (fseq) can be identified primarily through two sources of evidence (Baunaz et al., 2018). (i) Syncretism provides evidence for the adjacency of features: cross-linguistic patterns of syncretism reveal which features are present in the structure and how they are linearly ordered. (ii) Morphological containment (or nesting) goes further by establishing hierarchical relations: while syncretism shows adjacency, containment patterns indicate how features are embedded within one another. A third source of evidence is semantic compositionality, which may support the identification of the functional sequence, though on its own it is generally insufficient .

4.1 Case

Caha (2009) proposes that grammatical cases stand in a universal containment relationship, where less marked cases (such as the nominative) are structurally embedded within more marked ones (such as the genitive, accusative, or dative). By more marked, Caha refers to cases that are structurally more complex—those that involve a greater number of functional projections. In (23), is the hierarchical ordering Caha (2009) proposes for all cases.

(23)


This proposal is supported by cross-linguistic evidence from patterns of syncretism and morphological containment (or nesting) (Bobaljik, 2007, 2012). Syncretism suggests that certain cases share contiguous features, but it does not on its own determine the hierarchical ordering of those features. Morphological containment, by contrast, provides overt evidence about structural relationships, revealing which cases are closer to the nominal base and which project higher in the functional sequence.

Consider a few illustrative examples that support Caha's containment hierarchy:

- Cross-linguistically, nominative is often unmarked—lacking overt morphological expression. This suggests that it is structurally minimal, and thus occupies the lowest position in the case hierarchy (Caha, 2009, p. 42).
- In West Tocharian, the genitive plural form consists of two morphemes: *-em* and *-ts*, where *-em* corresponds to the accusative plural. This nesting indicates that the genitive morpheme contains the accusative, consistent with a structural hierarchy where accusative is embedded within genitive (Caha, 2009, p. 69).
- In Russian, the dative plural ending *-am* is overtly contained within the instrumental plural *-ami*, suggesting that instrumental contains dative.

Similar containment effects are also observed with prepositions. Caha (2009) and subsequent work argue that prepositions, like case morphemes, are composed of stacked case-features. In each of the examples below, a preposition that realizes a higher case selects a complement that bears a lower case in the hierarchy:

(24) a. In English, the genitive-marking preposition *of* selects an accusative complement:

of him (English)

b. In Arabic, the dative preposition *li* selects a dative-marked complement:

li-l-binti

to the girl.GEN (Arabic)

c. In German, the instrumental preposition *mit* selects a dative complement:

mit einem Hammer

with a.DAT hammer (German)

Similar containment effects are observed with prepositions. Caha (2009: 37) argues that prepositions, like case morphemes, are composed of stacked case-features. For example:

(25) a. In English, the genitive-marking preposition *of* selects an accusative complement

b. In Arabic, the dative preposition *li* selects a dative-marked complement

c. In German, the instrumental preposition *mit* selects a dative complement:

These selectional patterns are consistent with a view in which functional structure builds up in a fixed sequence, where higher K-heads select complements that instantiate lower portions of the case spine. For instance, the German preposition *mit* realizes that Instrumental head, while its dative complement realizes the cases below the instrumental as illustrated in (26).

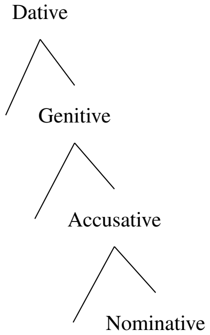
(26) Adapted from Baunaz et al., 2018, 43

Ins Dat Gen Acc Nom DP

mit *–dative*

Together, these patterns provide empirical support for a hierarchically organized case system in which more complex cases structurally contain simpler ones. In my analysis of the French pronominal system, I adopt Caha's case hierarchy but restrict attention to a relevant subtree of the full structure, shown in (23), focusing specifically on the subset of cases illustrated in (27).

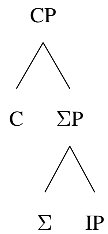
(27)



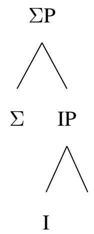
4.2 Strong, weak & clitic pronouns

Cardinaletti and Starke (1994), hereafter C&S, propose a widely accepted three-way distinction among pronouns: strong pronouns, weak pronouns, and clitics. Their analysis builds on Kayne (1975) and argues that these three classes of pronouns stand in a containment relationship. Strong pronouns are the most structurally complex (CP > ΣP > IP), weak pronouns are less complex (ΣP > IP), and clitics are the least complex (IP).

(28) Strong



(29) Weak



(30) Clitic

IP

Weak and clitic pronouns are collectively referred to as *deficient* pronouns, since they share a set of properties that distinguish them from strong pronouns. According to C&S, this contrast is tied to the presence of C. The relevant diagnostics are summarized in (Cardinaletti and Starke, 1994, 15) and can be illustrated with data from French. In what follows, I provide examples of coordination, modification, and cleft constructions, contrasting the strong pronoun *lui* with the weak pronoun *il*. .

(a) Strong, but not deficient pronouns, can be coordinated. This is shown in (31)–(32): weak pronouns like *il* cannot be coordinated, while strong pronouns like *lui* allow it.

(31) a. *Il est beau*
He is pretty
'He is pretty'

b. *Il et Jean sont beaux*
He and John are pretty
'John and him are pretty'

(32) a. *Lui est beau*
He is pretty
'He is pretty'

b. *Lui et celui de Jean sont beaux*

(b) Strong, but not deficient pronouns, can be modified by adverbs such as *seulement*, *seul*, or *juste*. This contrast is shown in (33).

(33) a. **Seulement il est beau*
Only he is pretty
'Only he is pretty'

b. *Seulement lui est beau*
Only he is pretty
'Only he is pretty'

(c) Strong, but not deficient pronouns, can occur in base/theta positions at surface structure, they can also occur in dislocations or clefts, as illustrated in (34).

(34) a. *C'est lui qui est beau*
It is him who is pretty
'It is him who is pretty'

b. **C'est il qui est beau*
It is he who is pretty
'It is he who is pretty'

(d) Strong pronouns, but not deficient pronouns, carry their own range-restriction. They can introduce discourse prominence through ostension or contrastive focus, and are the only forms that can appear as expletives, impersonals, non-referential datives, or with exclusive reference to human entities.

(35) a. *J' aime LUI, pas elle*
I like HIM, not her
'I like HIM, not her'

b. **J' aime il, pas elle*
I like HIM, not her
'I like HIM, not her'

These four properties — along with others not discussed here — lead C&S to argue that deficient pronouns (weak and clitic) differ systematically from strong pronouns.

Furthermore, C&S distinguish weak pronouns from clitics by proposing that clitics are heads at surface structure. One piece of evidence comes from their distribution: clitic pronouns cannot appear in XP-positions, such as verb-second initial position in Germanic, the specifier of an intermediate functional projection in Italian, or the sentence-initial position in Slovak (Cardinaletti and Starke, 1994, 21). Weak pronouns, by contrast, may occupy these XP-positions, which motivates treating them as maximal projections at surface structure.

A further diagnostic is clitic doubling. Doubling is always clitic doubling, in the sense that at least one of the doubled elements must be a clitic. This contrasts with weak pronouns, which do not participate in doubling constructions.

Prosodic evidence also supports the distinction. Both weak and clitic pronouns can receive phrasal or contrastive accent, but only weak pronouns may bear lexical word stress (Cardinaletti and Starke, 1994, 24). This difference aligns with their structural status: weak pronouns are deficient elements that surface as maximal projections, while clitic pronouns are deficient elements that surface as heads.

Taken together, these diagnostics suggest that weak and clitic pronouns, while both deficient, diverge in their structural realization at surface structure: weak pronouns as maximal projections, and clitics as heads.

For my analysis, I adopt Cardinaletti and Starke's (1999) tripartite distinction among strong, weak, and clitic pronouns, which assumes a containment relationship between the three classes. I propose that the features distinguishing these pronoun types are located below case features, in a way that parallels the structural relationship between nouns and case. Finally, I turn to φ-features, which I take to be situated below the IP layer.

4.3 Phi-features

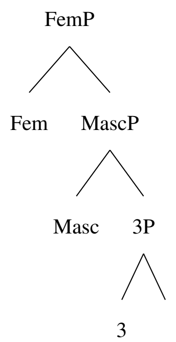
Cardinaletti and Starke (C&S) argue that the IP layer is composed entirely of ϕ-features. Following their approach, I assume that gender is located immediately below IP.

Picallo (1991) was the first to introduce the notion of a Gender Phrase (GenP) as the functional projection hosting grammatical gender. Within nanosyntax, several authors (Taraldsen, 2009, 2019) further decompose gender into distinct features: one for Masculine (*Masc*) and one for Feminine (*Fem*). These features stand in a containment relationship: the feminine form spells out the span [Masc + Fem], the masculine form lexicalizes only [Masc], and the neuter form emerges when neither feature is present.

Since this thesis focuses exclusively on singular pronouns, the Number projection will be set aside. At the lowest level of the hierarchy lies the Person feature, represented here as *3* for third person.

The feature structure for ϕ-features assumed in this thesis is shown in (36).

(36)



4.4 Final f-seq for pronouns

Building on the discussion above, the functional sequence of pronouns can be organized into three main domains. At the top are the case features, which stand in a containment relationship (Dat > Gen > Acc > Nom). Below case are the class-defining features that distinguish strong, weak, and clitic pronouns, which also form a containment hierarchy. Finally, at the lowest level of the structure are the ϕ-features, including gender and person.

The projections in (37) represent the resulting functional sequence of pronouns, though it is not intended to be exhaustive. For instance, the various classes of pronouns likely require additional structural layers to capture their full range of properties. Moreover, I have omitted the Number projection, since this thesis focuses exclusively on singular pronouns.

(37) Dat $>$ Gen $>$ Acc $>$ Nom $> CP > \Sigma P > IP >$ Fem $>$ Masc > 3

In Chapter 5, I show that the feature sequence in (37) does not, on its own, capture the full range of syncretisms observed in the French pronominal system.

Chapter 5

French pronoun data and the problem they pose

In this Chapter, I present the French third-person pronoun paradigm (repeated below for convenience) and discuss why, at first glance, Nanosyntax is unable to account for the observed syncretism. In Chapters 6 and 7, I introduce two tools within the nanosyntactic framework that help resolve this issue. Finally, in Chapter 8, I propose five structural representations that generate the observed data.

Below, in Tableau 1.3, I summarize the third-person pronoun paradigm. This paradigm is multidimensional, involving grammatical features for case (nominative, accusative, and dative), gender (feminine and masculine), and class-defining distinctions (i.e., whether a pronoun is strong, weak, or clitic). For now, I set aside the strong dative forms (*à lui* and *à elle*).

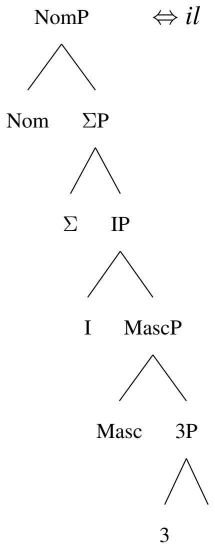
(38)

	Strong M.	**Weak M.**	**Clitic M.**	**Strong F.**	**Weak F.**	**Clitic F.**
Nom	lui	il	-	elle	elle	-
Acc	lui	-	le	elle	-	la
Dat	(à lui)	-	lui	(à elle)	-	lui

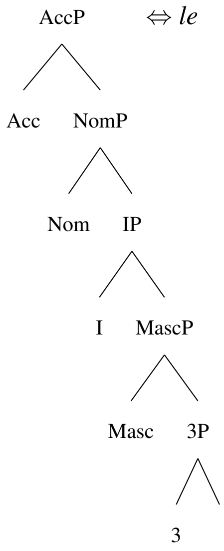
Il is nominative, weak, and masculine. In contrast, *le* is accusative, clitic, and masculine, while *la* is accusative, clitic, and feminine. Assuming the functional sequence (fseq) outlined in Chapter 4, and in the absence of any spellout-driven movement, our system predicts the three L-trees in (39), (40) and (41) for *il*, *le*, and *la*, respectively.

Tentative L-tree for il, la and le

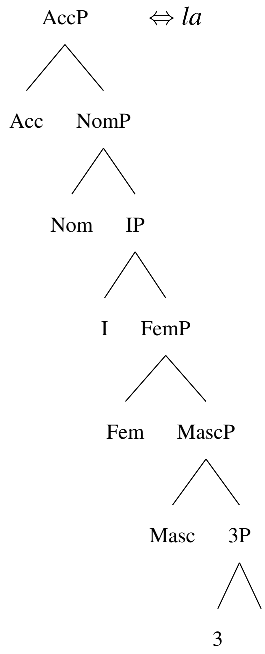
(39)



(40)



(41)



Il, *le*, and *la* each appear in only one position within the paradigm and are therefore relatively straightforward to analyze. In contrast, *elle* and *lui* occur in multiple environments, making their analysis more complex.

Elle appears in three distinct contexts: (i) nominative, strong, and feminine; (ii) accusative, strong, and feminine; and (iii) nominative, weak, and feminine. To account for this syncretism, the L-tree for *elle* must minimally contain all the features associated with the three *elle*'s to satisfy the requirements of the Superset Principle. However, it must also allow for certain features to be absent in specific S-trees—namely, ACC in (i) and (iii), and C in (iii).

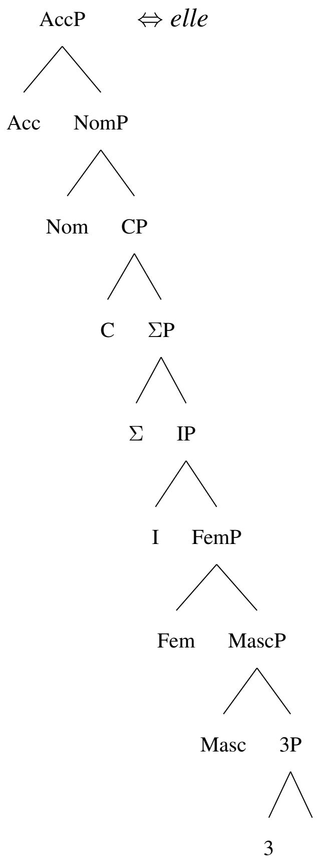
If we assume the L-tree in (42), then (i) — nominative, strong & feminine— can be derived by omitting only the topmost feature ACC, which yields a valid match under the Superset Principle. Similarly, (ii) — accusative, strong & feminine— can be derived straightforwardly, as it is an exact match. However, (iii) —nominative, weak, feminine— cannot be derived from this structure: it lacks not only ACC, but also C—a feature that occurs in the middle of the L-tree proposed in 42. Since intermediate nodes cannot be omitted, the S-tree for (iii) would not be a valid match.

Lui presents a challenge similar to that of *elle*, as it appears in four distinct contexts: (i) nominative, strong, and masculine; (ii) accusative, strong, and masculine; (iii) dative, clitic, and masculine; (iv) dative, clitic, and feminine. To capture this syncretism, the L-tree for *lui* must include all the features relevant to these four uses. However, as with *elle*, some of these features (e.g., DAT, GEN, ACC, C, Σ, and FEM) would need to be absent from certain

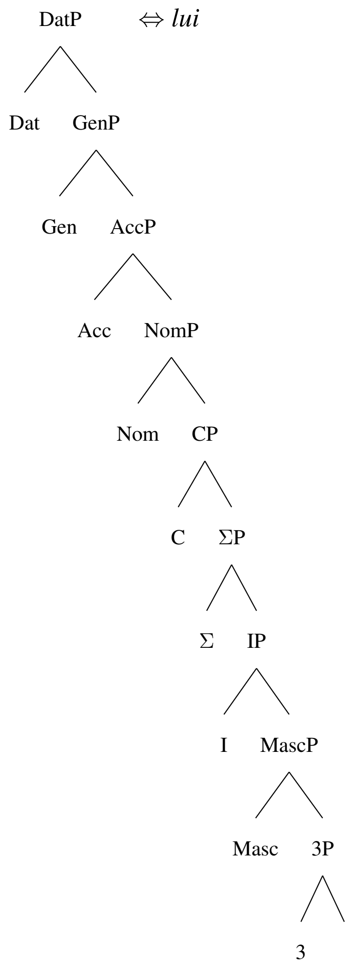
S-trees, which the current system cannot account for.

Tentative L-trees for lui and elle

(42)



(43)



In the following chapter, I examine the solution proposed by Wyngaerd (2018), which I ultimately argue is unsatisfactory. I then turn, in Chapters 6 and 7, to two fairly recent nanosyntactic advancements (pointers and branching L-trees) that account for the French data without requiring any modifications to the Superset Principle.

5.1 Wyngaerd revised superset principle

Wyngaerd (2018) addresses a comparable issue in Slave, an Athabascan language that—like French pronouns—exhibits syncretism in a multidimensional paradigm. In this paradigm, certain syncretic forms require the presence of a feature located in the middle of the L-tree, even though this feature is absent in some corresponding S-trees. When an L-tree must lexicalize an S-tree that lacks a middle feature present in the L-tree, Wyngaerd refers to the resulting mismatch as "shrinking in the middle."

To resolve this problem, Wyngaerd proposes a revision of the Superset Principle. In his reformulation, the Superset Principle no longer requires the S-tree to be a literal subtree of the L-tree. Instead, it evaluates whether the feature set of the L-tree constitutes a superset of the features in the S-tree, thereby permitting "shrinking in the middle":

(44) A lexical entry L may spell out a syntactic node SN iff the features of L are equal to or a superset of the features dominated by SN. (Wyngaerd 2018: 11)

This revision is far from trivial. It effectively treats L-trees not as hierarchical structures, but merely as unordered sets of features. If feature order and structure no longer matter, much of the movement observed in nanosyntactic derivations would likely become unnecessary or unmotivated.

While this revised Superset Principle does generate the French pronoun data, it represents a substantial departure from the original formulation. Moreover, Wyngaerd (2018) himself has since moved away from this view. In what follows, I argue that such a drastic revision is unnecessary: by drawing on independently motivated tools already available in Nanosyntax—such as pointers and branching lexical items—it is possible to capture the French data while preserving the original Superset Principle.

Chapter 6

Pointers

6.1 Pointers to account for suppletions

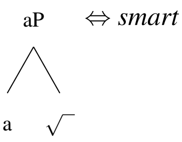
This Chapter introduces a key mechanism in Nanosyntax—*pointers*. Initially proposed to account for suppletion, pointers have since been extended to capture other phenomena and will play a central role in the analysis of the French data. Before turning to their application in the French pronominal paradigm, I first outline why pointers are independently motivated in Nanosyntax to account for suppletion (see Vanden Wyngaerd et al. 2021 for a more in-depth discussion).

Since Nanosyntax adopts a Late Insertion model—where lexical items are inserted only after the syntactic structure has been built—it does not distinguish between open-class lexical items during the syntactic derivation. For instance, nouns like *cat* and *dog* are associated with the same structural representation, meaning the grammar must permit the free insertion of any root that satisfies the relevant structural conditions.

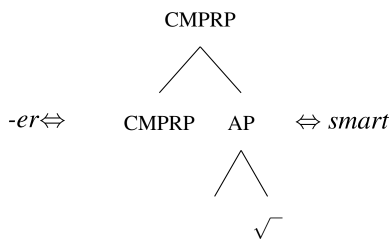
At the same time, the system must also allow for competition between lexical items. This competition is governed by the Superset Principle and the Elsewhere Principle, as outlined in Chapter 2.3. However, these same principles raise challenges when accounting for patterns of root suppletion—such as in *bad–worse*—as opposed to *nice–nicer*.

The structure of comparative adjectives has been widely discussed in the literature, most notably in Bobaljik (2012), who proposes that the comparative and positive adjectives are in a containment relationship. Specifically, the comparative phrase (CMPRP) is projected above the positive adjective phrase (aP). The syntactic trees for an adjective without a suppletive comparative root (e.g., *smart* and *smarter*) are illustrated in example (45) and (46). Although I do not represent spell-out movement here, the adjective would move up and to the left of the comparative suffix *-er*.

(45)



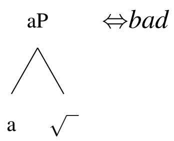
(46)



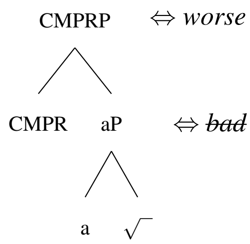
For adjectives with suppletive roots—such as *bad* and *worse*—the underlying syntactic structure remains the same. The S-tree in (47) could be lexicalized by many different lexical items (e.g. bad, nice, happy, sad...). However, in the comparative form, the root *worse*, unlike *nice* for instance, is able to lexicalize the entire CMPRP structure. One might initially propose the L-tree in (48) where *worse* can lexicalize the whole structure and override *bad*, but I will explain below why this structure will need to be revised:

Bad and Worse (to be revised)

(47)



(48)



Although the tree in (48) could in principle lexicalize the positive form in (47), the root *bad* is inserted instead, in accordance with the Elsewhere Principle: it provides a closer match by lacking the unnecessary CMPR feature. In the comparative context, however, *worse* is preferred over *bad-er* because it lexicalizes the entire structure without requiring movement.

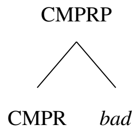
While this analysis successfully derives the contrast between *bad* and *worse*, it also raises a significant problem: if *worse* is treated as the most specific lexicalization of CMPRP, the Elsewhere Principle would predict that it should also be inserted in any comparative context. This would incorrectly block regular forms like *nicer*, which are composed of a root and a comparative suffix. Thus, the L-tree in (48) overgenerates and must ultimately be revised.

The proposed solution by Starke (2011, 2014) is to allow lexical items to include, within their lexical entry, a reference to another lexical item.

(49) *Pointer:* A node within the L-tree of a lexical item that refers to another existing lexical item. (Starke, 2014)

Pointers serve to link irregular forms to their corresponding base forms, while also preventing the overgeneration of regular morphology. The lexical item *worse* can still lexicalize the full comparative structure (including both the CMPR and aP projections), but this structure is not entirely contained within its own L-tree. Instead, part of it is supplied via a pointer to another lexical item (which itself has its own L-tree). The L-tree for *worse*, therefore, includes a pointer to *bad*, as shown in (50). This ensures that *worse* can only be inserted in contexts where the aP has already been lexicalized by *bad*, and not by any other adjective.

(50) L-tree for *worse* (Revised, with a pointer)



6.2 Pointers to account for idioms

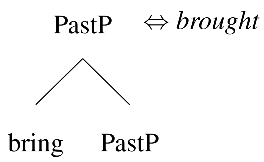
A second major use of pointers, as discussed by Starke (2011, 2014), is in the lexicalization of idiomatic expressions. By definition, idioms are non-compositional in meaning—that is, the meaning of the whole cannot be derived from the meanings of its individual parts. Pointers offer a way to account for this mismatch between syntax and semantics. If we assume that an idiom is represented as a single lexical item whose L-tree contains pointers to each of its constituent parts, then its internal structure can be preserved without requiring a fully compositional interpretation. For instance, the idiom *break a leg* would involve at least three pointers, linking the idiomatic meaning to the lexical items *break*, *a*, and *leg*. This mechanism allows the syntax to remain strictly compositional while accommodating the idiosyncratic meanings associated with idioms.

6.3 Pointers to account for syncretism in multidimensional paradigms

Caha and Pantcheva were the first to suggest that pointers can be useful to account for syncretism in complex paradigms involving multiple dimensions of grammatical features (see also Starke (2013); Wyngaerd (2018)). To illustrate this I will use the irregular past tense form *brought*.

As discussed in Chapter 6.1, suppletion in Nanosyntax is accounted for by using pointers in L-trees which link a suppletive root to its base form. Thus, the irregular past tense *brought* must contain, within its L-tree, a node that stores the lexical item *bring*.

(51) Bring-Brought with one pointer



Given that there is syncretism between Past Perfect and Passive illustrated in (52) it has been proposed that the lexical item -*ed* has more internal structure as shown in (51) and it contains the features shown in 53

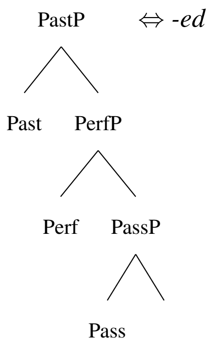
(52) *Past, Perfect and Passive Syncretism*

a. They prepared food / They brought food

b. They have prepared food / They have brought food

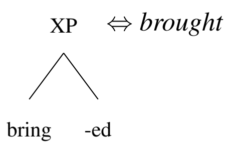
c. Food was prepared / Food was brought

(53) *L-tree for -ed*



A natural consequence of the mechanism of pointers is that the item that is pointed to is an independent lexical item and can match an S-tree that the L-tree contains as a subtree. Informally put, an item with a pointer can shrink not just at the top but also in the middle of the tree at the top of the item being pointed to. (p.8). For this reason if we assume that the lexical item *brought* contains a pointer not only to *bring*, but also to *-ed* then it is able to generate all cases of syncretism with the past, perfect and passive and thus allows *shrinking in the middle*.

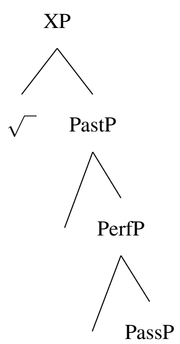
(54) *L-tree for brought (final)*



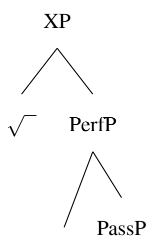
If *brought* contains a pointer to *-ed*, then it does not matter whether *-ed* has been lexicalized by the full structure in (53) or by a smaller substructure, such as *PassP*. This results in what appears to be *shrinking in the middle*, as all the S-trees in (55), (56) and (57) can be lexicalized as *brought*, even though some of them lack features that are present in the middle of the larger S-trees which can be lexicalized by the same lexical item.

3 S-trees for brought

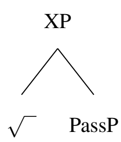
(55)



(56)



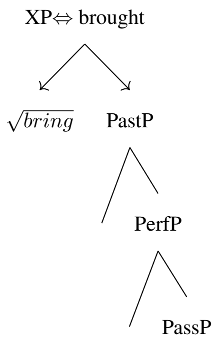
(57)



Crucially, if pointers were not used, then the Superset Principle would not be able to account for this syncretism, as it would require a single lexical item to match all of the structures in (55), (56) and (57) which is not possible due to the nature of the Superset Principle.

There are two common notational conventions for representing pointers in L-trees. The first, illustrated in (54), involves writing the lexical item directly into the tree—such as *bring* and *-ed* in that example. An alternative convention, which I will adopt for the remainder of this paper, uses an arrow pointing to the full L-tree structure associated with the lexical item. Under this approach, (58) is equivalent to (54), and conveys the idea that the higher features in the structure can be omitted via a pointer to a previously stored lexical entry.

(58)



Chapter 7

Branching L-trees

In this Chapter, I discuss Complex Left Branches (CLBs)—a recent development in Nanosyntax that proves particularly useful for analyzing the French pronoun data.

7.1 Complex Left Branches

Blix (2021) was the first to propose that lexical items can be associated with L-trees that include a Complex Left Branch. A CLB refers to a lexical tree in which one or more features appear on the left branch. This branch is considered complex because, due to how Nanosyntax forms specifiers, the features in this position cannot be directly merged but must instead be derived through spellout-driven movement. Since its introduction, the CLB proposal has been adopted by several authors, including Caha (2021, 2023); Cortiula (2023); Janků (2022).

The primary motivation for positing CLBs, as argued by Blix (2021), is that they eliminate the need for backtracking, a computationally expensive step in the lexicalization algorithm. This is possible when CLBs are used in conjunction with pointers, which are already independently motivated within the theory. As described in Chapter 2.3, backtracking is a rescue strategy in which the derivation returns to a previous cycle and revises the structure of the complement of a newly merged feature. This process turns spellout into a recursive algorithm (Blix, 2021, 2), greatly increasing the number of potential derivational paths and resulting in exponential computational complexity.

CLBs offer a more efficient alternative. By encoding movement-derived structure directly into lexical items and relying on tools such as pointers and partial overwriting, the derivation can proceed linearly, without revisiting earlier stages.

7.2 Iron Ossetic Personal Pronouns

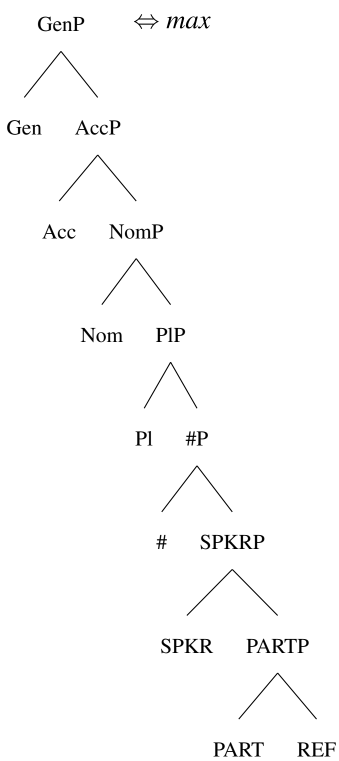
To illustrate the effectiveness of CLBs and pointers, Blix revisits an example originally presented by Caha (2009): the Iron Ossetic first-person plural pronoun max and the noun fyd 'father', shown in (59). While Caha used this example to motivate the necessity of backtracking, Blix argues that the same data can be accounted for using CLBs and pointers alone.

(59) Iron Ossetic

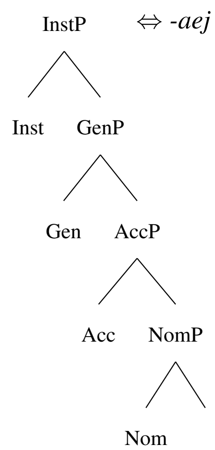
	1PL	**Father,SG**
NOM	max	fyd-
ACC	max	fyd-y
GEN	max	fyd-y
INS (ABL)	max-æj	fyd-æj
DAT	max-æn	fyd-æn

As shown above, the noun *fyd* surfaces with the suffix *-y* in both accusative and genitive, while the pronoun *max* appears unmarked in these cases. This syncretism suggests that *max* spells out a structure at least as large as GenP, as represented in (60). The suffix *-aej*, which lexicalizes the instrumental feature, also overrides the *-y* suffix that realizes accusative and genitive in the nominal paradigm. Therefore, the L-tree for *-aej* must include the full substructure up to GenP, as shown in (61).

(60) *L-tree for max (Caha)*

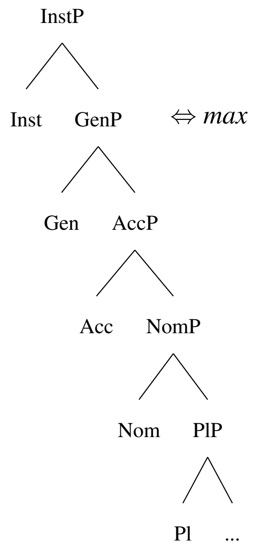


(61) *L-tree for aej (Caha)*



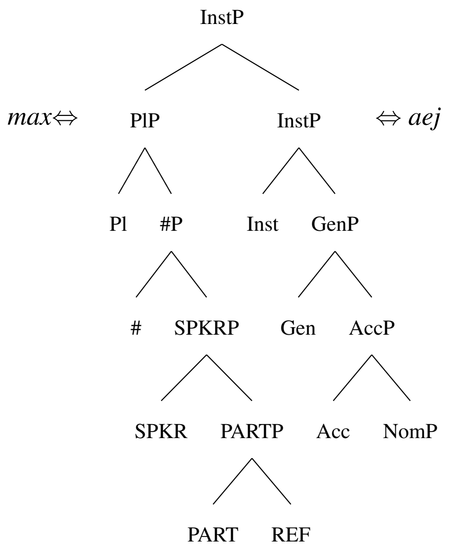
According to Caha (2009), backtracking becomes necessary once the Inst feature is merged. At that point, *max* has already spelled out the subtree below InstP (see 62), but *-aej* requires a structure that includes not only Inst but also Gen, Acc, and Nom.

(62) *S-tree that cannot be lexicalized*



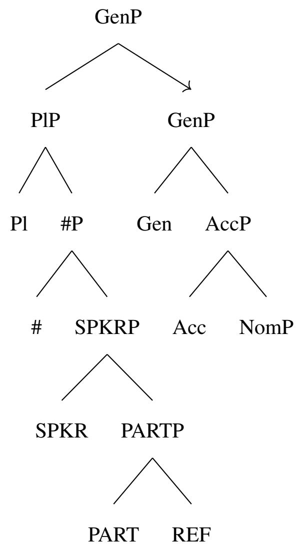
Caha proposes a solution involving backtracking: the derivation returns three cycles earlier and performs repeated spec-to-spec movement of PlP to produce the desired structure shown in (63). For a step-by-step derivation, see Blix (2021).

(63) *S-tree arrived at by backtracking (Caha 2009)*



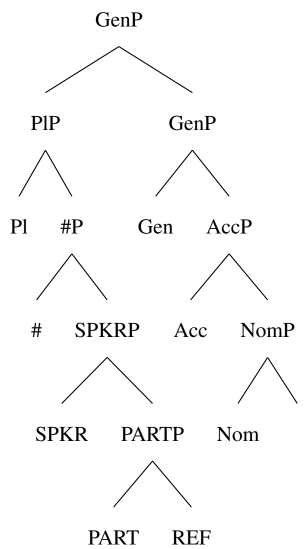
Blix agrees that the tree in (63) is correct, but argues that backtracking is unnecessary. Instead, he proposes an L-tree for *max* that incorporates a Complex Left Branch (CLB) and a pointer, as shown in (64).

(64) L-tree for *max* (Blix)



In this configuration, once the Nom feature is merged, Comp-to-Spec movement is triggered and PlP moves to the specifier of Nom. The same movement applies at each subsequent projection: when Acc is merged, PlP moves again this time by Spec-to-spec movement, and likewise for Gen.

(65)



Once the Inst feature is merged, *max* can no longer lexicalize the full structure, since its L-tree does not contain Inst. At this point, spec-to-spec movement applies again, and *-aej* partially overwrites *max*. Crucially, the overwriting is partial: *max* continues to lexicalize the PlP constituent. This results in the same structure proposed by Caha in (63), but it is derived without backtracking.

In the following chapter, I show how the same mechanisms—CLBs and pointers—can account for the French pronoun paradigm. While this analysis is not concerned with the debate over the necessity of backtracking, backtracking will not be required to derive the data, thus perhaps favouring the view that backtracking is not necessary.

Chapter 8

Proposed solution

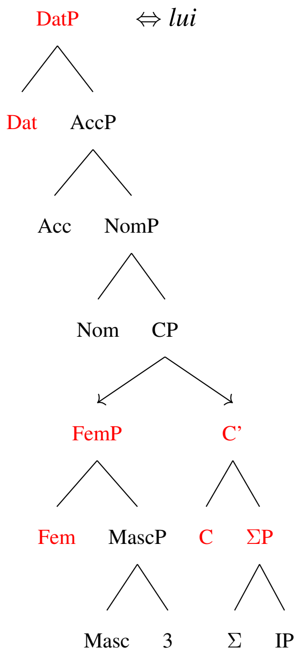
My proposal combines pointers with branching lexical items to derive the French pronoun paradigm. Case features occupy the highest position in the L-trees, and syncretism across different cases follows straightforwardly from the Superset Principle. Below the case nodes, the structure branches: the left branch contains the φ-features (later displaced in the S-tree by spellout-driven movement), and the right branch hosts the features defining pronoun classes.

To capture the "shrinking in the middle" observed with *lui*, two pointers are required: one targeting the φ-features, allowing *lui* to lexicalize both masculine and feminine S-trees; and one targeting the class-defining feature, allowing it to lexicalize both strong and clitic S-trees.

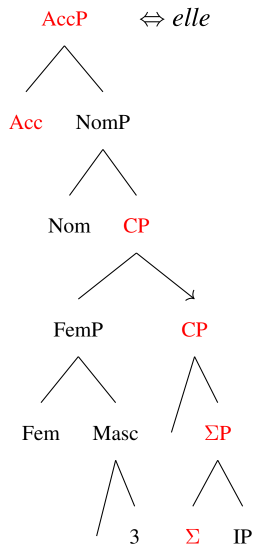
For *elle*, a similar structure is assumed: case features at the top, followed by a branching node with φ-features on the left and class features on the right. However, in this case, only a single pointer is needed—one that targets the class feature—allowing *elle* to lexicalize both strong and weak forms. In (66) and (67) below, the L-trees for *elle* and *lui* are presented, with features that may be omitted highlighted in red.

Proposed trees for lui and elle

(66)

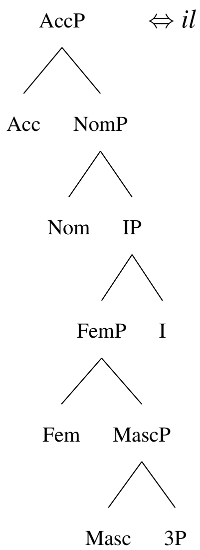


(67)

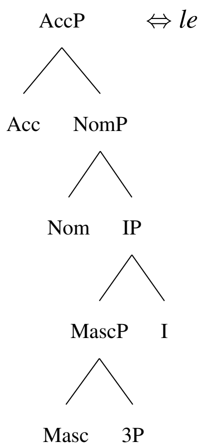


The trees for *il*, *le*, and *la* do not require any pointers, as these lexicalize only a single grammatical form and never involve the omission of features. However, they still require the same branching structure that separates the φ-features from the class features. This branching is crucial: without it, the spellout-driven movements (such as spec-to-spec and comp-to-spec) required by the more complex L-trees for forms like *lui* and *elle* would not be triggered. This is because, in earlier derivational steps, the S-tree could already be lexicalized in a simpler, flatter structure—thus blocking the need for further movement and preventing the correct derivation of those forms. These trees are represented in (68)-(70).

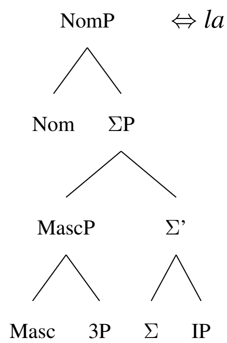
(68)



(69)



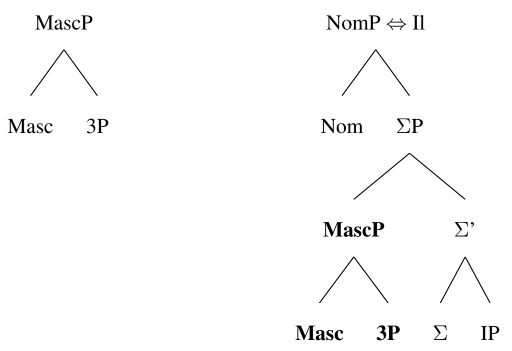
(70)



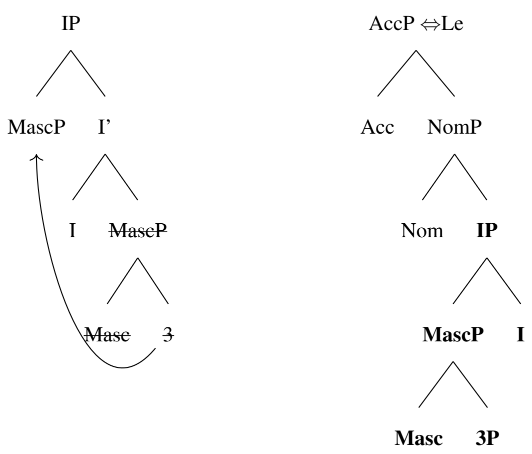
8.1 Deriving *Lui* (Masc, Nom, Str)

In this Chapter, I break down the derivations of several pronouns in the paradigm. For each example, the S-tree under construction will be shown on the left, with the matching L-tree on the right. As a reminder, at every merge step, the structure must be matched by a lexical item. If no match is found, the derivation proceeds as follows: (1) spec-to-spec movement is triggered; if that still fails to yield a match, (2) comp-to-spec movement applies; and if that also fails, (3) backtracking would occur—though this final step will not be necessary in the cases discussed here.

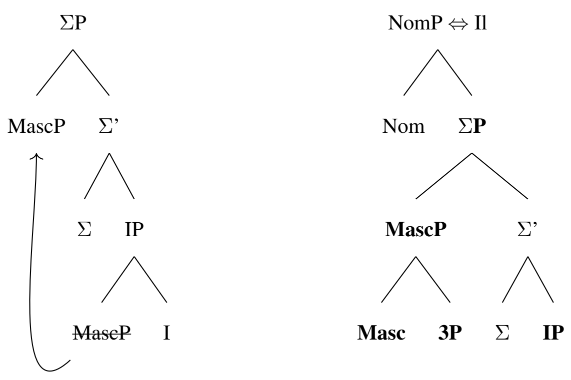
1. The first step is to merge the feature *Masc* to 3P. This structure can be lexicalized by the L-tree for *il*.



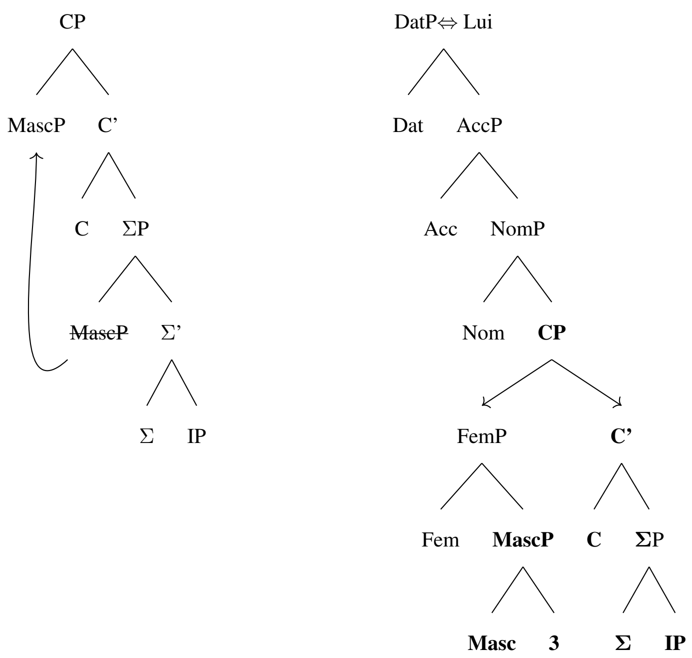
2. IP is merged, but it does not find a matching lexical item in the lexicon. Spec-to-Spec movement is not an option at this point, since the category Masc does not have a specifier. As a result, the only available operation is Comp-to-Spec movement. This structure is now a match for *le*.



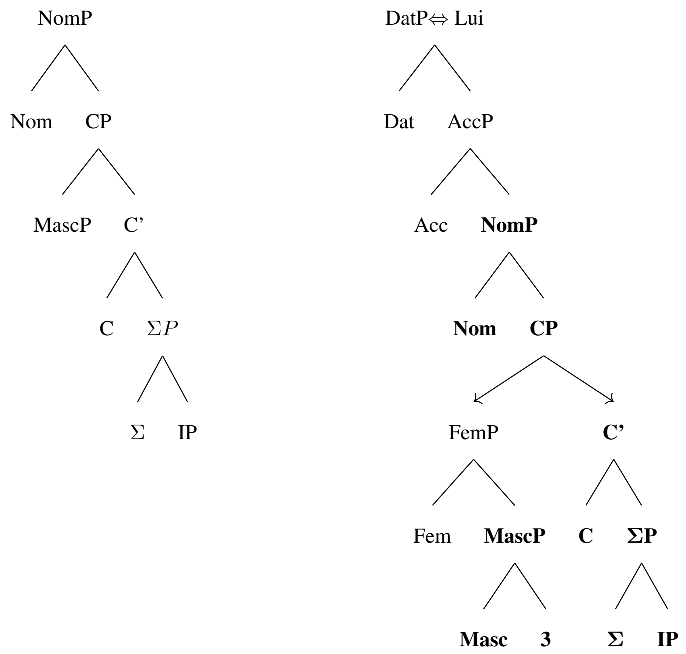
3. ΣP is merged next in the derivation. At this stage, the resulting structure cannot yet be lexicalized. To resolve this, Spec-to-Spec movement applies, yielding a structure that can now be lexicalized by il.



4. CP is merged next in the derivation, but the resulting structure does not yield a lexical match. To remedy this, Spec-to-Spec movement applies, creating a structure that can be lexicalized by *lui*.



5. Finally, the feature Nom is merged to the structure. No movement is required and this is a match with *lui* once again.



8.2 Discussion

One aspect of this analysis, which is not fully novel but still not widely accepted, is the idea that a pointer can refer to a structure that is not itself a lexical item realized anywhere in the language. For *lui* and *elle*, I propose a pointer to CP>ΣP<IP, even though this configuration never surfaces independently: it must co-occur with case and φ-features, and therefore has no phonological realization. This type of "non-lexical target" pointer has precedent in Blix (2021), but, to my knowledge, there are no other analyses that extend this strategy in a comparable way.

Secondly, A second criticism that may be raised concerns the use of more than one pointer within a single lexical item. Multiple pointers are not unprecedented in the literature, but when combined with the fact that these pointers target a structure that is never independently spelled out, the analysis could be viewed as overly powerful. Allowing both (i) multiple pointers and (ii) pointers to non-realized structures risks granting too much flexibility to the system.

Finally, one issue I have yet to address, concerns the strong datives in both the masculine and feminine forms. Unlike the nominative and accusative pronouns, the strong dative occurs with a preposition. In the examples below, I demonstrate this by coordinating the pronouns, a construction possible only with strong pronouns. While *lui* in (71) can be either masculine or feminine, this is not the case for *à lui* and *à elle* which are only masculine and only feminine respectively. The feminine *à elle* exhibits the same distribution as *à lui*, as shown in examples (74) to (79).

(71) Je **lui** donne le livre / *Je **lui et Jean** donne le livre.

(72) *Je donne **lui** le livre / *Je donne **lui et Jean** le livre

(73) *Je donne le livre **lui/** *Je donne le livre **lui et Jean**

(74) *Je **à lui** donne le livre / *Je **à lui et Jean** donne le livre.

(75) Je donne **à lui** le livre / Je donne **à lui et Jean** le livre

(76) Je donne le livre **à lui** / Je donne le livre **à lui et Jean**

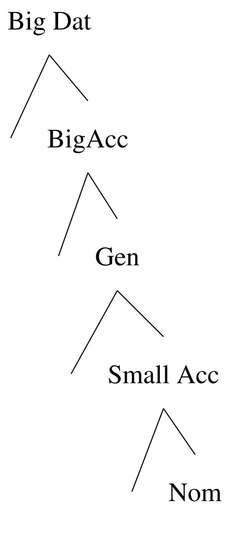
(77) *Je **à elle** donne le livre / *Je **à elle et Jean** donne le livre.

(78) Je donne **à elle** le livre / Je donne **à elle et Jean** le livre

(79) Je donne le livre **à elle** / Je donne le livre **à elle et Jean**

The issue with *à lui* concerns why the features associated with *à lui/elle* are not simply lexicalized as *lui*. The L-tree proposed in (66) contains all the relevant features (e.g., dative, CP, Masc) and satisfies the Superset Principle. However, as discussed in 2.3, the presence of any pre-element (e.g., a preposition) triggers the creation of a secondary workspace—the final-resort operation of the lexicalization algorithm. Without this additional workspace, the L-tree proposed in 8 for *lui* would overgenerate, since it would also be capable of lexicalizing the structure of strong datives.

One possible explanation, whcih is beyond the scope of this paper, is that the dative splits into two, a "big DAT" and a "small DAT" (Starke, 2017). (Parallel distinctions for accusative are also argued for, though they are not directly relevant for the French facts.)

(80)


Starke proposes that this is necessary because the relative ordering of the Dative and Genitive features is difficult to establish: both positions have been argued for in the literature, supported either by syncretism patterns (which sometimes align Dative with lower cases and sometimes with higher ones) or by morphological containment facts in languages where one case form properly contains another. This ambiguity makes the Dative–Genitive portion of the f-sequence one of the least understood cross-linguistically, and motivates the need for additional more features to capture the full range of empirical patterns without forcing a single universal ordering.

Chapter 9

Conclusion

This thesis set out to explain the syncretism patterns found in the French third-person pronominal paradigm, a domain where multiple morphosyntactic dimensions (case, gender, and pronoun class) interact in ways that challenge standard assumptions about lexical insertion. The empirical problem was that forms such as *lui* and *elle* participate in syncretisms that appear to require "shrinking in the middle," something that the classical formulation of the Superset Principle cannot derive.

Building on a nanosyntactic architecture that assumes a highly articulated functional sequence for pronouns, I argued that two independently motivated tools—pointers and branching L-trees—can account for the data. Pointers allow a lexical entry to omit non-edge material by referring to another stored lexical tree; branching structures (complex left branches) ensure that movement-driven lexicalization steps can target the correct subtrees without invoking backtracking. Together, these mechanisms derive all instances of syncretism in the French paradigm, including the most problematic ones involving *lui* and *elle*.

The analysis follows and supports the idea that systematic syncretism is not accidental but follows from the internal organization of lexical entries and the structure of the functional sequence itself. Finally, the French data support a view in which lexical items may contain internal structure richer than the syntactic span that they spell out in any particular context.

By deriving the paradigm without modifying the core architecture of nanosyntax, this thesis provides further evidence that the combination of pointers, complex left branches, and the original Superset Principle offers a powerful and predictive model for multidimensional syncretism. This approach therefore contributes not only to our understanding of French pronouns but also to broader questions about the limits of lexicalization, the nature of syncretism, and the architecture of the syntax–morphology interface.

Bibliography

Baker, M. (2008). The macroparameter in a microparametric world. In Biberauer, T., editor, *The Limits of Syntactic Variation*, pages 351–374. John Benjamins, Amsterdam.

Baunaz, L. (2015). On the various sizes of complementizers. *Probus*, 27(2):193–236.

Baunaz, L. (2016). Deconstructing complementizers in serbo-croatian, modern greek and bulgarian. In Hammerly, C. and Prickett, B., editors, *Proceedings of NELS 46 (1)*, pages 69–77, Amherst, MA. Graduate Linguistics Student Association.

Baunaz, L., Lander, E., De Clercq, K., and Haegeman, L. (2018). Nanosyntax: the basics. *Oxford Studies in Comparative Syntax*, pages 3–56.

Belletti, A., editor (2004). *Structures and Beyond: The Cartography of Syntactic Structures, Vol. 3*. Oxford University Press, New York.

Benincà, P. (1988). L'ordine degli elementi della frase e le costruzioni marcate. In Renzi, L., editor, *Grande Grammatica Italiana di Consultazione*, pages 129–194. Il Mulino, Bologna.

Blix, H. (2021). Phrasal spellout and partial overwrite: On an alternative to backtracking. *Glossa: a journal of general linguistics*, 6(1).

Bobaljik, J. (2007). On comparative suppletion. Manuscript of an encyclopedia chapter.

Bobaljik, J. (2012). *Universals in Comparative Morphology: Suppletion, Superlatives, and the Structure of Words*. MIT Press, Cambridge, MA.

Bonet, M. E. (1991). *Morphology after Syntax: Pronominal Clitics in Romance*. Ph.d. dissertation, Massachusetts Institute of Technology, Cambridge, MA.

Borer, H. (1984). *Parametric Syntax*. Foris, Dordrecht.

Caha, P. (2009). The nanosyntax of case. *Linguistic Inquiry*, 40:275–320.

Caha, P. (2010). The parameters of case marking and spell out driven movement. In van Craenenbroeck, J., editor, *Linguistic Variation Yearbook 2010*, pages 33–77. John Benjamins, Amsterdam/Philadelphia.

Caha, P. (2013). Explaining the structure of case paradigms by the mechanisms of nanosyntax: The classical armenian nominal declension. *Natural Language and Linguistic Theory*, 31(4):1015–1066.

Caha, P. (2021). Comparatives in czech. Lectures at Masaryk University, Fall 2021.

Caha, P. (2023). Nanosyntax. Lectures at Masaryk University, Spring 2023.

Caha, P., De Clercq, K., Starke, M., and Vanden Wyngaerd, G. (2024). Nanosyntax: state of the art and recent developments. *NanoDays, Date: 2024/02/07-2024/02/08, Location: Brno*.

Cardinaletti, A. and Starke, M. (1994). The typology of structural deficiency. on the three grammatical classes. *Working papers in linguistics*, 4(2):41–109.

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. MIT Press, Cambridge.

Chomsky, N. (1981). *Lectures on Government and Binding*. Foris Publications, Dordrecht.

Chomsky, N. (1986). *Barriers*. MIT Press, Cambridge, MA.

Chomsky, N. (1995). *The Minimalist Program*. MIT Press, Cambridge, MA.

Chomsky, N. (2001). Derivation by phase. In Kenstowicz, M., editor, *Ken Hale: A Life in Language*, pages 1–50. MIT Press, Cambridge, MA.

Cinque, G. (1999). *Adverbs and functional heads: A cross-linguistic perspective*. Oxford University Press.

Cinque, G., editor (2002). *Functional Structure in DP and IP: The Cartography of Syntactic Structures, Vol. 1*. Oxford University Press, New York.

Cinque, G. et al. (1990). *Types of A'-dependencies*. MIT Press (United States).

Cinque, G. and Rizzi, L. (2008). The cartography of syntactic structures. In Moscati, V., editor, *CISCL Working Papers on Language and Cognition*, volume 2, pages 43–59. University of Siena, Siena, Italy. Studies in Linguistics.

Clercq, K. D. (2013). *A Unified Syntax of Negation*. Ph.d. dissertation, Ghent University.

Cortiula, M. (2023). The nanosyntax of friulian verbs. .

Giusti, G. (1993). *La sintassi dei determinanti*. Unipress, Padova.

Hein, J., Weisser, P., and Leipzig, U. (2021). Syncretism–recurring patterns.

Hjelmslev, L. (1935). *La catégorie des cas: étude de grammaire générale*, volume VII/1 of *Acta Jutlandica*. Universitetsforlaget, Aarhus. Published 1935–1937.

Jakobson, R. (1971). Beitrag zur allgemeinen kasuslehre. gesamtbedeutungen der russischen kasus. In *Selected Writings. Vol. 2: Word and Language*, pages 23–71. Mouton, The Hague/Paris. Originally published in 1936.

Janků, L. (2022). *The Nanosyntax of Czech Nominal Declension*. Phd dissertation, Masaryk University, Brno, Czech Republic.

Kayne, R. S. (1975). *French Syntax: The Transformational Cycle*. Phd dissertation, Massachusetts Institute of Technology.

Keine, S. and Müller, G. (2019). Impoverishment. In Alexiadou, A., Kramer, R., Marantz, A., and Oltra-Massuet, I., editors, *The Cambridge Handbook of Distributed Morphology*, pages xx–yy. Cambridge University Press.

Kiparsky, P. (1973). 'elsewhere' in phonology. In Kiparsky, P. and Anderson, S., editors, *A Festschrift for Morris Halle*. Holt, Rinehart and Winston, New York.

Pantcheva, M. (2011). *Decomposing Path: The Nanosyntax of Directional Expressions*. Ph.d. dissertation, University of Tromsø.

Pott, A. F. (1836). *Etymologische Forschungen auf dem Gebiete der Indo-Germanischen Sprachen: Grammatischer Lautwechsel und Wortbildung. 2*, volume 2. Meyer.

Ritter, E. (1988). A head-movement approach to construct-state noun phrases. *Linguistics*, 26:909–929.

Ritter, E. (1991). Two functional categories in noun phrases: evidence from modern hebrew. In Rothstein, S. D., editor, *Perspectives on Phrase Structure: Heads and Licensing*, volume 25 of *Syntax and Semantics*, pages 37–62. Academic Press, San Diego.

Rizzi, L. (1997). The fine structure of the left periphery. In Haegeman, L., editor, *Elements of Grammar*, pages 281–337. Kluwer, Dordrecht.

Rizzi, L., editor (2004). *The Structure of CP and IP: The Cartography of Syntactic Structures, Vol. 2*. Oxford University Press, New York.

Rizzi, L. (2013). Syntactic cartography and the syntacticisation of scope-discourse semantics. In Reboul, A., editor, *Mind, Values and Metaphysics – Philosophical Papers Dedicated to Kevin Mulligan*, pages 517–533. Springer, Dordrecht.

Rocquet, A. (2013). *Splitting Objects: A Nanosyntactic Account of Direct Object Marking*. Ph.d. dissertation, Ghent University.

Starke, M. (2002). The day syntax ate morphology. *Class taught at the EGG summer school, Novi Sad*.

Starke, M. (2009). Nanosyntax: A short primer to a new approach to language. *Nordlyd*, 36(1):1–6.

Starke, M. (2011). Issues in nanosyntax. Research seminar, CASTL, University of Tromsø.

Starke, M. (2013). Auxiliaries and structural gaps: Current issues in nanosyntax. Lecture series presented at CRISSP, Hogeschool-Universiteit Brussel, 18, 20, 22 March.

Starke, M. (2014). Towards elegant parameters: Language variation reduces to the size of lexically stored trees. *Linguistic variation in the minimalist framework*, pages 140–154.

Starke, M. (2017). Resolving (dat = acc) gen. *Glossa*, 2(1):104.1–8.

Taraldsen, K. T. (2009). The nanosyntax of nguni noun class prefixes and concords. Manuscript, CASTL. Available at: `https://lingbuzz.net/lingbuzz/000876`.

Taraldsen, K. T. (2019). Nanosyntax and syncretism in multidimensional paradigms. *Linguistics Vanguard*, 5(1):20180058.

Vanden Wyngaerd, G., De Clercq, K., and Caha, P. (2021). Late insertion and root suppletion. *Revista Virtual de Estudos da Linguagem-ReVEL*, 19(18):81–123.

Wyngaerd, G. V. (2018). The feature structure of pronouns: A probe into multidimensional paradigms. *Exploring nanosyntax*, pages 277–304.

Øystein A. Vangsnes (2014). Indexicals by nanosyntax: Wh and d items apart. Paper presented at the SLE Workshop on Nanosyntax, Poznań, Poland. September 12.