Case assignment in TSL syntax: a case study

Mai Ha Vu
University of Delaware
maiha@udel.edu

Nazila Shafiei
Stony Brook University
nazila.shafiei@stonybrook.edu

Thomas Graf
Stony Brook University
mail@thomasgraf.net

Abstract

Recent work suggests that the subregular complexity of syntax might be comparable to that of phonology and morphology. More specifically, whereas phonological and morphological dependencies are tier-based strictly local over strings, syntactic dependencies are tier-based strictly local over derivation trees. However, a broader range of empirical phenomena must be considered in order to solidify this claim. This paper investigates various phenomena related to morphological case, and we argue that they, too, are tier-based strictly local. Not only do our findings provide empirical support for a kind of computational parallelism across language modules, they also offer a new, computationally unified perspective of structural and lexical case. We hope that this paper will enable other researchers to fruitfully study syntactic phenomena from a subregular perspective.

1 Introduction

After significant success in phonology and morphology (Heinz, 2018 and references therein), the subregular approach has recently been extended to syntax. Graf (2018) shows that the basic Minimalist operations Merge and Move — which form the core of syntax — belong to the formal class *tier-based strictly local* (TSL; Heinz et al., 2011). More precisely, Graf shows that Minimalist grammars have TSL derivation tree languages even though their string languages are mildly context-sensitive (Joshi, 1985; Harkema, 2001; Michaelis, 2001). It has been known for a long time that many mildly context-sensitive formalisms have regular derivation tree languages (see Morawietz 2003, Kobele et al. 2007, and references therein). Graf's result is noteworthy because it identifies the very limited subclass TSL as sufficient for Minimalist grammars. Since TSL also plays a central role in phonology (McMullin and Hansson, 2015; McMullin, 2016) and morphology (Aksënova et al., 2016), this suggests a kind of cognitive parallelism: linguistic dependencies have comparable subregular complexity across language modules.

While the findings in Graf (2018) are promising, they are far from conclusive as even the most dedicated Minimalist will readily admit that syntax consists of a lot more than just Merge and Move. Many phenomena remain to be explored, including those at the interface to morphology or semantics. Vu (2018) has already started this enterprise with an investigation of NPI licensing. This paper continues along these lines with a TSL-based analysis of morphosyntactic case.

Case is a fruitful area to explore for several reasons. First, the data is very robust compared to, say, binding or quantifier scope. Second, case exhibits few interactions with movement; such interactions still pose major challenges for TSL-accounts. At the same time, case dependencies are still sufficiently intricate that it is not immediately obvious how they could be handled by TSL mechanisms. Existing treatments of case in Minimalist grammars (Laszakovits, 2018) or agreement in general (Ermolaeva, 2018) imply that case dependencies are definable in first-order logic, but TSL is much more restricted than that. Case also lacks a unified linguistic theory. Our TSL approach combines ideas from standard case theory (Chomsky, 1981) and Dependent Case Theory (DCT; Marantz, 1991; Baker and Vinokurova, 2010) into a computationally uniform solution for both structural case (nominative and accusative) and lexical case.

The paper proceeds as follows. All necessary preliminaries are put in place in §2. This includes Minimalist grammars as a formal model of syntax, Minimalist derivation trees as the central data structure, and finally TSL itself. TSL uses a sim-

Proceedings of the Society for Computation in Linguistics (SCiL) 2019, pages 267-276.
New York City, New York, January 3-6, 2019

ple projection mechanism to construct tree tiers over which distributional constraints can be enforced in a local fashion. We then apply this idea to morphological case in §3. The central idea is that there is a single *case tier* that contains every case-carrying head. In addition, each head requires a licensor on the tier, and the licensor must be the sister of the licensee's mother. This holds for both structural and lexical case. Conceptual and methodological aspects of this subregular analysis are discussed in §4.

We deliberately keep all notation and formal machinery to a minimum in order to accommodate readers without much exposure to computational linguistics. More formally inclined readers should be able to reconstruct the formal machinery from Graf (2018) and Tab. 1.

2 TSL syntax

2.1 Minimalist grammars

Subregular syntax operates over tree structures rather than strings. Hence it requires both a linguistic theory of what those tree structures are, and a formally rigorous model for implementing these ideas. The latter is provided by Minimalist grammars (MGs; Stabler, 1997, 2011), which implement ideas of Chomsky's Minimalist syntax (Chomsky, 1995).

The technical details of MGs are largely irrelevant for the purposes of this paper. It suffices to know that phrase structure trees are assembled from feature-annotated lexical items via the structure building operations Merge and Move. The sequence of Merge and Move steps can be represented as a derivation tree, the central data structure for MGs. Derivation trees differ only minimally from phrase structure trees: I) interior nodes are labeled Merge (•) or Move, and II) moving phrases remain in their base position, and the specifier that would be occupied by the mover remains empty. To improve readability, all derivation trees in this paper are shown with the usual X′-labels for interior nodes (see e.g. Fig. 2).

Due to our reliance on MGs, the derivations we posit are closely modeled after standard accounts in Minimalist syntax: I) subjects start in a low position (Spec,VP or alternatively Spec,vP), from which they move to Spec,TP, and II) finite clauses are CPs, whereas infinitival clauses may be TPs or CPs. Most importantly, we make the less standard assumption that the head of a DP enters the derivation with a preassigned case. Consequently, the problem of case assignment amounts to regulating the distribution of case-carrying D-heads in MG derivation trees.

Figure 1: Long-distance sibilant harmony is tier-based strictly local over strings as it can be expressed as a local ban against [sʃ] and [ʃs] on a tier of sibilants.

2.2 TSL over strings

Graf (2018) analyzes MGs from a subregular perspective by adapting the phonologically motivated class of TSL string sets for syntax. The idea behind string-based TSL is simple: if a dependency is non-local, it can be made local by masking out irrelevant material. This masking out can be understood in linguistic terms as the construction of a tier. Constraints on this tier take the form of forbidding a finite number of substrings, the length of which must be finitely bounded. For example, long-distance sibilant harmony with respect to feature f can be modeled as a TSL-dependency by first projecting all sibilants and then requiring adjacent sibilants on the tier to always agree on their value for f. If the only sibilants are [s] and [ʃ], for instance, then long-distance anteriority harmony amounts to banning the substrings [sʃ] and [ʃs] on the tier (see Fig. 1). Even though there is no limit on the length of the tier, the constraint never has to consider more than two adjacent segments at a time and thus satisfies the bounded-size requirement.

2.3 Projection of tree tiers

TSL over trees as defined by Graf (2018) is more general than string-based TSL. First, and perhaps most importantly, the tier projection function is allowed to consider the local context of a node n in the derivation tree in order to determine whether n projects. So rather than constructing a tier that contains, say, all nodes that are verbs, one may opt for more complex tiers such as "every verb that selects a DP headed by *the*". Formally, a local context for projection to tier T is specified by a subtree in which exactly one node is superscripted by T to indicate that it should be put on tier T. If a node n in the derivation tree occurs as part of a subtree that matches at least one specified context for tier

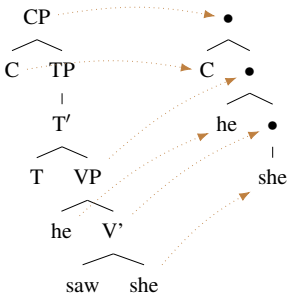

Figure 2: Nodes of the derivation tree (with X′-labels for clarity) are projected onto a tree tier based on their label and their local context. The dominance relations in the tier are inherited from the original tree.

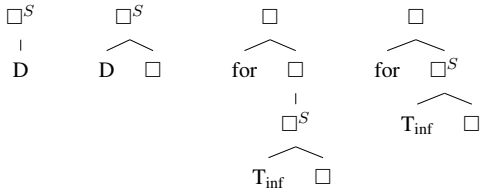

Figure 3: Four contexts for projection to a structural case tier S; the first one projects a node whose only daughter is a D-head, whereas the second one projects a node with a D-head as the left daughter and some other node to the right (□ is used as a placeholder that matches a node irrespective of its label). The third and fourth handle projection of the mother of an infinitival T-head selected by *for* in cases with and without subject movement, respectively.

T, n is projected to T. Note that the depth of the subtrees in all context specifications (i.e. the maximum distance between the root and any leaves) must be less than some fixed finite bound k, otherwise the context specifications would not be locally bounded. This is the case for our example here as it suffices for every node to look at its label and the labels of its daughters, if they exist.

Figure 2 shows the construction of a tier that contains all C-heads and their mothers as well as all D-heads with nominative case and their mothers. Crucially, the size of the context that is considered by the tier projection is finitely bounded for each node — for C-heads and D-heads only the label needs to be considered, and for all other nodes it suffices to check whether their daughter is a C-head or D-head.

Figure 3 displays two contexts. The first two ensure that every mother of a D-head is projected. The remaining two handle projection of the mother of an infinitival T-head that is selected by the C-head *for*. This is one of the most complex instances of projection used in this paper (§3.6, §3.7).

2.4 Constraints on tree tiers

Constraints on tree tiers are also allowed to be more general than in the string case. Each node on the tier is mapped to a string language that its string of daughters must belong to. Graf (2018) only discusses the simple case where this mapping considers just the label of the node. But it is easy to amend the definition so that local context information is taken into account here, too.

Continuing the example in Fig. 2, we may put in place a simple constraint to capture the fact that two distinct arguments in the same clause cannot both be nominative: if a node on the tier is labeled • and has a sister carrying nominative case, then it must not have a daughter carrying nominative case. Formally, a node n that is labeled • and appears in the context [□ [NOM, n]] is mapped to the string language $(\Sigma - \{\text{NOM}\})^*$ (where □ is a placeholder matching every node label and Σ is a fixed set of node labels). This constraint is violated in Fig. 2 as one node is labeled • and has both a NOM-carrying sister *he* and a NOM-carrying daughter *she*. If *she* were replaced by *her*, then neither *she* nor its mother would be projected on the tier, and consequently the constraint would no longer be violated.

In this paper, only Merge nodes put restrictions on their daughter strings. For each Merge node, its set of daughter strings forms a finite language of strings of length 1. Such string languages are maximally simple. The precise shape of the daughter string language for any given Merge node depends only on the siblings of the Merge node. This, too, is a very simple context specification.

The complexity of a TSL-dependency over trees is an aggregate of three different parameters: the complexity of the tier projection, the complexity of the function that associates each node on the tier with a set S of permissible daughter strings, and the complexity of S. For our treatment of morphological case, all these components will turn out to be exceedingly simple.

3 Analysis

While the preliminaries section has been kept fairly informal, it nonetheless provides all the nec-

essary background to develop a TSL analysis of morphological case. In this section, we present our TSL account for the distribution of nominative (NOM) and accusative (ACC) in various constructions of English. We cover transitive, intransitive, and unaccusative verbs (§3.1), infinitival clauses (§3.2), ditransitives (§3.3), raising to subject (§3.4), control (§3.5), raising to object (§3.6), *for*-clauses (§3.7), and the effects of clausal subjects on case assignment (§3.8). None of the data is new, but our TSL approach represents a novel synthesis of traditional case theory and DCT that relies on a single, unified mechanism of sister-daughter dependencies on a tier.

3.1 Transitive, intransitive, and unaccusative

The core generalization of structural case assignment is that NOM is assigned to the structurally highest DP x in a case domain (modulo movement), and ACC to the DPs c-commanded by x (Marantz, 1991; Baker and Vinokurova, 2010). What constitutes a case domain is a contentious issue, but largely irrelevant for our purposes. For the sake of simplicity, we will assume that each CP is a case domain, and nothing else is.

Examples (1a-1c) show simple sentences with transitive, intransitive, and unaccusative verbs. No matter the type of verb, we observe that the subject is always NOM, and the object DP, c-commanded by the subject, carries ACC.

(1) a. He saw her.
b. He slept.
c. He arrived.

Under many generative analyses, both the subject and the object enter the derivation as arguments of the verb or one of its functional projections (usually vP). They are thus part of a structural configuration of bounded size. For example, if the subject starts out in Spec,VP and the object as the complement of VP, then they are part of a subtree of size 3. Strictly speaking, then, case assignment in these configurations does not require tiers at all. Instead, one can simply list all possible subtrees of size 3 — there are only finitely many, after all — and mark a subtree as illicit if the subject does not carry NOM or the object does not carry ACC. This would be the subregular counterpart to a fully lexicalized analysis of case assignment.

But this simple account is inadequate. Merely listing all forbidden subtrees allows for arbitrary case variations. We may require two nominatives with *like*, two accusatives with *see*, and any random combination with *want*. A list cannot capture any relevant generalizations, just like listing all attested words of English is not a description of its phonology. The lack of generalizations also entails a lack of succinctness — the list of forbidden subtrees would quickly reach hundreds of thousands for even a fairly small lexicon. Most importantly, apparent cases of long-distance case assignment, e.g. ECM, cannot be handled this way. We contend that a TSL analysis that takes inspiration from DCT is empirically more adequate, more insightful, and more succinct.

Let us return to the generalization that NOM is assigned to the highest DP in a case domain, and ACC to the argument c-commanded by this DP. We have to ban the following configurations in simple sentences: two NOM DPs in a c-command relation (2a), and ACC DPs that are not c-commanded by a "licensing" NOM DP (2b).

(2) a. * He saw she.
b. * Him slept.

We have already seen in §2 how a TSL account can handle some of these facts. But some modifications are needed.

We still project every C-head and its mother, as well as every D-head h carrying NOM. In contrast to what was said in §2, we do not project the mother of this head, but rather the Merge node that checks its category feature. For a simple DP like *she*, this Merge node will indeed be the mother. But consider a larger DP such as *John's father*, which is commonly analyzed as [$_{\text{DP}}$ [$_{\text{DP}}$ John] [$_{\text{D}'}$'s [$_{\text{NP}}$ father]]]. Here the mother of *'s* would indicate the point where *'s* is merged with *father*, not the point where the whole DP is selected as an argument. Said point corresponds to when the Merge node checks the category feature of the DP's head *'s*. By projecting the Merge node that checks the category feature of a head h, we effectively project the mother of the whole phrase headed by h. Fortunately, it holds for every MG that there is an upper bound k such that no head is more than k steps away from the Merge node that selects its category feature (cf. Graf, 2012). Which node that is can be inferred from the feature specification of the head. This guarantees that the projection only needs to consider contexts of finitely bounded size in order to project both a case

carrying D-head and the Merge node checking its category feature.

The projection function just described produces tiers that resemble the one in Fig. 2. If *he* were replaced by *John's mother* in this derivation, the tier would be almost the same except that *he* would have to be replaced by *'s*. Recall from §2 that we assume that D-heads are annotated with the morphological case of the DP, so in both instances the tier clearly indicates the presence of NOM. The constraint from §2 does not allow Merge nodes to have both a NOM sister and a NOM daughter. This rules out *he saw she* (Fig. 2), whereas *he thinks that she has arrived* is allowed thanks to the intervening C-head starting a new case domain (Fig. 4).

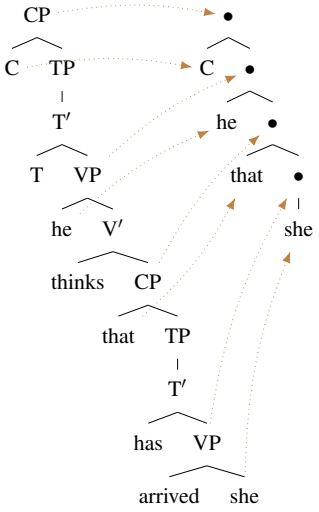

Figure 4: This case assignment pattern is permitted because no Merge node on the tier has both a NOM daughter and a NOM sister.

This takes care of double nominatives, leaving only ACC. Although ACC licensing can be enforced on a separate tier, it is easier to use a single tier for licensing of both NOM and ACC. To this end, we extend the tier projection function so that D-heads carrying ACC also end up on the tier, as do the Merge nodes checking their category features. In other words, (3b) is amended so that h is a D-head carrying NOM or ACC.

We then enforce that no Merge node without a NOM D-head as a sister may have an ACC D-head as a daughter — or equivalently, every Merge node with an ACC daughter must have a NOM sister. This correctly rules out *him slept* while allowing for *he saw her*.

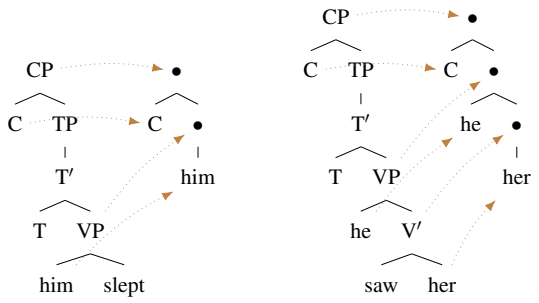

Figure 5: The left derivation is illicit because ACC is not licensed on the tier; the right one is well-formed as the required NOM sister is present on the tier.

(3) **Case tier (version 1)**
Construct a tier that contains all of the following, and only those:

a. all C-heads and their mothers, and

b. every D-head h carrying NOM or ACC, and

c. the unique Merge node that checks the category feature of h.

The following constraints hold for every Merge node m:

i. if a daughter of m is NOM, then no sister of m carries NOM,

ii. if a daughter of m is ACC, the same sister of m carries NOM.

NOM and ACC thus are regulated by almost exactly the same machinery, the only difference is what daughter/sister-combinations are allowed for Merge node. In order to determine the well-formedness of a derivation with respect to case assignment, one constructs the case tier and checks each Merge node for potential constraint violations. Like DCT, the TSL approach can immediately account for the fact that subjects of passives and unaccusatives receive NOM even though they start out in object position (cf. Fig. 4), an important fact that would have to be stipulated under a purely lexicalist account.

3.2 Infinitival clauses

The current account puts no restrictions on NOM in subject position, missing the fact that subjects of infinitival clauses cannot carry NOM.

(4) * He/Him to leave early is surprising.

While the ungrammaticality of *him* is unsurprising under our account, nothing should be wrong with

he to leave early is surprising. One might argue that this sentence is indeed well-formed with respect to case assignment but violates some other constraint. However, we prefer to explicitly ban NOM subjects in infinitival clauses as this will simplify things later on in §3.6 and §3.7.

(5) **Case tier (version 2)**
Construct a tier that contains all of the following, and only those:
   a. all C-heads and their mothers, and
   b. all finite T-heads and their mothers, and
   c. every D-head h carrying NOM or ACC, and
   d. the unique Merge node that checks the category feature of h.

   The following constraints hold for every Merge node m:
   i. if a daughter of m is NOM, then some sister of m is T_{fin},
   ii. if a daughter of m is ACC, the some sister of m carries NOM.

Note that (5i) renders the ban against multiple NOM DPs in (3i) obsolete as NOM is now much more restricted in its distribution.

3.3 Ditransitives

Ditransitive verbs in English seem to take two ACC objects.

(6) a. ?? I showed her him.
   b. I showed him to her.

According to Larson (1990), the indirect object (IO) in double object constructions starts out as the complement of the verb, and the direct object (DO) as a specifier. The subject enters the derivation as the specifier of some higher functional head. This is problematic for our treatment of ACC as this means that the tier would contain a Merge node with IO ACC as a daughter and the ACC DO as a sister. The requirement that every Merge node with an ACC daughter must have a NOM sister explicitly forbids this.

There are at least three ways to address the ditransitive challenge. Rather than picking one among them, we briefly sketch all of them because I) they illustrate the flexibility of the TSL approach to syntax, and II) each strategy may be useful in cases that do not involve ditransitives.

Chain-licensing The simplest solution is to weaken (5ii) such that an ACC daughter is also acceptable if the Merge node has an ACC sister. This effectively allows for a kind of *chain-licensing* such that each ACC is licensed by the next higher ACC. The highest ACC still needs to be licensed by NOM, though.

Dative Alternatively, one may reanalyze ACC on IO as a dative (DAT) that merely happens to be syncretic with ACC. English already displays morphological syncretism for NOM and ACC on all DPs except pronouns, so a total syncretism of ACC and DAT is conceivable. The main advantage of this solution is that it readily extends to languages that display a morphological distinction between ACC and DAT. In German, for example, IO is explicitly marked as DAT in most cases.

(7) a. Ich zeige ihr den Weg.
   1SG.NOM show 3SG.F.DAT the.M.ACC way
   b. * Ich zeige sie den Weg.
   1SG.NOM show 3SG.F.ACC the.M.ACC way
   'I show her the way.'

Dependent DAT would be handled like ACC, except that now it is ACC that functions as a licensor instead of NOM. In other words, no Merge node on the case tier may have a DAT daughter unless it also has an ACC sister. The reader is invited to verify that this blocks (7b) but not (7a), assuming that the base DO position c-commands the base IO position.

Lexical case One may also contend that (6a) is in fact ill-formed because the IO carries an unlicensed ACC, and that (6b) escapes this fate only because the preposition *to* acts as a lexical case assigner. Lexical case is handled by constructing a separate tier for lexical case. If a case-carrying D-head occurs in the local scope of a lexical case assigner, both the head and the Merge node checking its category feature are projected to the lexical case tier instead of the usual case tier. This tier also contains every lexical case assigner and its mother. On this tier, it holds for every Merge node m that its sibling assigns case c iff exactly one of the daughters of m carries case c.

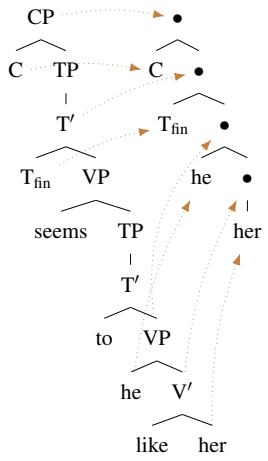

Figure 6: Case assignment in raising works exactly the same as in simple clauses.

3.4 Raising to subject

Our TSL analysis has the advantage that it readily extends to seemingly more complicated structures, e.g. subject raising. Consider the example in (8), where the subject must carry nominative case.

(8) He/*him seems to *t* like her.

Assuming a transformational analysis, the embedded clause is a TP containing the surface subject, which subsequently moves into the matrix clause. This is indicated by a trace in (8).

But if the transformational analysis is indeed correct, then raising constructions do not need any special treatment in our account. As movement does not play a role in our tier-projection, the respective case tiers will mirror those for *he likes her* and *him likes her* (see Fig. 6).

3.5 Control

Control constructions as in (9) are superficially similar to subject raising, but involve a very different structure.

(9) He persuaded her [$_{CP}$ PRO to leave *she/her].

The standard assumption in generative syntax is that the embedded clause in these constructions is a CP whose subject is an unpronounced PRO. In order to capture the contrast above, we expand (5) so that PRO can license ACC.

(10) **Case tier (addendum to (5))**

a. Also project PRO and the merge node that checks its category feature.

b. If a daughter of Merge node m is ACC, then some sister of m is PRO or carries NOM.

We would also like to point out an alternative analysis that might merit further exploration by syntacticians. If one allows for chain-licensing of ACC (§3.3), then one can do away with PRO and treat the embedded clause as a TP (or perhaps even just a VP). The lower ACC would then be licensed by either NOM (*He was persuaded to leave her*) or a higher ACC (*John persuaded him to leave her*). While intriguing, this view seems incompatible with the idea that other cases such as DAT in German are (at least sometimes) structurally licensed by ACC.

3.6 Exceptional Case Marking (ECM)

Another important construction is raising to object, also known as ECM.

(11) He believes [$_{TP}$ *she/her to like *he/him].

Given (5i), the ungrammaticality of NOM on the embedded subject in (11) already follows from the lack of a licensing T$_{fin}$. This also explains why the embedded object cannot carry NOM. The only open issue, then, is the well-formedness of *he believes her to like him*.

In contrast to control, ECM does not involve an underlying PRO. Hence the PRO-less analysis of control sketched above would work exactly the same for ECM constructions: NOM on the main clause subject licenses ACC on the embedded subject, which in turn licenses ACC on the object. But this analysis depends on chain-licensing, which may run into various problems in other languages. There is, however, an alternative proposal that eschews chain-licensing, is compatible with a PRO-based analysis of control, and builds on the implementation of lexical case at the end of §3.3.

Said analysis posits that the infinitival T-head (T$_{inf}$) acts as a lexical licensor for both the ACC subject and the ACC object, but does so in two very different ways. First, T$_{inf}$ and its mother are projected to the tier instead of the ACC subject, exempting the latter from any licensing requirements. The T-head then acts a licensor for the ACC object on the tier (Fig. 7). The embedded subject must carry ACC for this to work, otherwise it will project on the tier and prevent T$_{inf}$ from licensing ACC on the object. Since the subject cannot be NOM, it cannot license the object's ACC instead

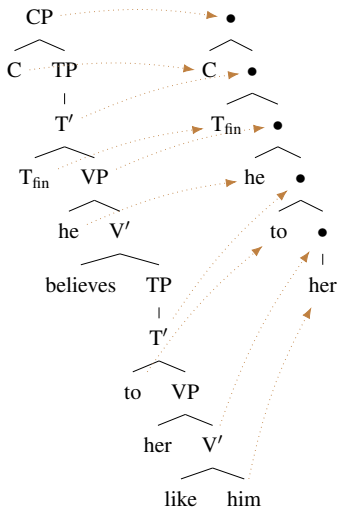

Figure 7: ECM verbs allow infinitival T to project instead of the subject to license ACC.

of the T-head, and the whole derivation is rendered illicit.

In order for this solution to work as desired, it may only apply to T-heads selected by an ECM verb, and only an ACC subject under which such a T-head may be exempt from projection. While the former is a local property, the latter is less clear-cut because a clause may contain an unbounded number of VP-adjuncts. If the subject enters in a derivation below these adjuncts, then the distance to the T-head is unbounded. However, if one posits that subjects reside in the higher Spec,vP position, and that VP-adjuncts must adjoin lower than that, the projection context is still local. It is also true that the subject stands in a local configuration to the T-head by virtue of moving there, so a slightly more powerful tier projection mechanism that is at least partially aware of movement can construct the desired tier irrespective of the locus of VP-adjuncts relative to the subject.

3.7 Clauses headed by *for*

Note that the two ECM analyses presented in the previous section make slightly different predictions. With chain-linking, there has to be at least one NOM in the derivation. With lexical licensing, no NOM is needed. This predicts that there should be ECM verbs that do not take any subject, e.g. an ECM-counterpart to *seem*.

(12) * There seems him to like her.

As far as we know, no language has such ECM-verbs. But something akin to this configuration arises with *for*-clauses.

(13) For *he/him to leave early is surprising.

If one treats ECMs as a case of special licensing by T_{inf}, then (13) follows immediately if the special behavior of T_{inf} can also be prompted by *for* rather than an ECM verb. The chain-licensing account, on the other hand, has to project *for* as a special licensor of ACC. So the intriguing typological prediction of the chain-licensing account comes at the cost of a less unified treatment of ECM and *for*-clauses.

3.8 Clausal subjects

One final complication arises from the fact that clausal subjects are ACC licensors even though they arguably do not carry NOM.

(14) That John left early surprised *she/her.

While this may be problematic for syntactic theories of case, it is entirely unsurprising from the TSL perspective. There is no intrinsic property of nodes that qualifies them as potential licensors. Without linguistic stipulations, case-carrying D-heads and functional heads are on equal footing regarding licensing, they are all just nodes of a tree. Suppose, then, that we not only project the mother of a C-head, but also the Merge node checking its category feature (if it exists). This would mean that C-heads are unique in that they cause two Merge nodes to project, the mother m and the selector s. Since all other elements on the tier are always projected together with their selector, clausal subjects cause a unique configuration where the Merge node s has exactly two daughters, one being the Merge node m and the other being some other Merge node dominating a case carrier or licensor. So if we expand the set of potential ACC licensors to also include •, this captures the fact that clausal subjects license ACC on their object (Fig. 8).

4 Discussion

Our treatment of case licensing is far from exhaustive. We focused almost exclusively on English, omitting many important issues such as ergative-absolutive case systems and quirky case in Icelandic. Even for English we had to put aside the complicated and little understood behavior of case assignment in coordination (Progovac, 1998).

Project …	if …
C + mother + selecting •	always
T_{fin} + mother	always
T_{inf} + mother	selected by ECM-verb or *for*
PRO + selecting •	always
NOM + selecting •	always
ACC + selecting •	not subject under projecting T_{inf}
DAT+ selecting •	treated as dependent case

Daughter	Licensing sibling
NOM	T_{fin}
ACC	•, *for*-C, T_{inf}, PRO, NOM
DAT	ACC

Table 1: Projection rules for structural case tier (left) and what sibling a Merge node must have on the tier based on its daughter(s); if case-licensing is assumed for ACC, T_{inf} need not be projected.

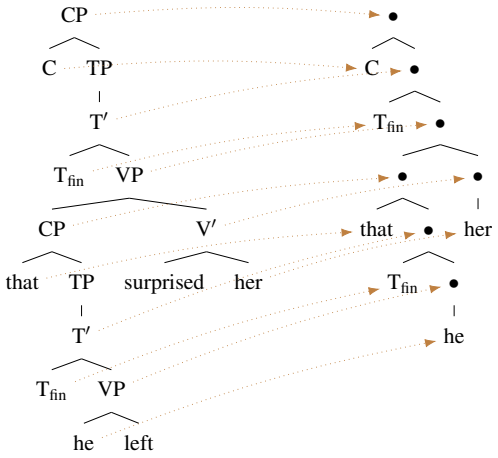

Figure 8: Licensing of an accusative object by the Merge node of a clausal subject

Some readers may also object that the case licensors we identified do not form a natural class. But this presumes a specific notion of naturalness that is based on linguistic substance. This view groups together the heads and phrases involved in case licensing according to their function or meaning. But besides uniformity of substance there is also uniformity of mechanisms, and the TSL view reveals that case is very principled from this perspective.

As can be seen in Tab. 1, the mechanisms of case licensing are highly uniform. The projection function singles out specific heads and their mother and/or selector, and all constraints on the case tier make the daughter of a Merge node m dependent on the sister of m. A single abstract pattern underlies all instances of case licensing. In addition, all parts of the formal machinery fit into TSL (with the possible exception of T_{inf} nonlocally blocking projection of the subject). From a computational perspective, then, morphological case marking does not appear to be an oddity that requires special machinery, but a formally uniform phenomenon that can be handled by the computational mechanisms that are already needed for Merge and Move (Graf, 2018).

Putting substantive naturalness over formal naturalness means missing this insight. We expect that our approach will require various modifications as its empirical scope is widened, but the central role of TSL and the uniformity of mechanisms across phenomena should be preserved.

5 Conclusions

This subregular case study has shown morphological case assignment to be TSL, with ECM as the only problematic case. Even this, though, can be addressed by a tier projection function that is partially sensitive to movement. Since movement interacts with numerous syntactic dependencies, e.g. pronominal binding, a movement-aware tier projection is needed anyways and should be a high priority for future work in subregular syntax.

We also note that the TSL perspective reveals case licensing to be very uniform at a computational level. This is striking considering the lack of a unified theory of case in generative syntax. In fact, our TSL account of case exhibits many abstract parallels to the analysis of Merge and Move in Graf (2018). We are confident that the TSL perspective will prove insightful for many other syntactic phenomena, and we hope that the paper will inspire other researchers to reevaluate syntax through this lens.

References

Alëna Aksënova, Thomas Graf, and Sedigheh Moradi. 2016. Morphotactics as tier-based strictly local dependencies. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 121–130.

Mark C. Baker and Nadya Vinokurova. 2010. Two modalities of case assignment: Case in Sakha. *Natural Language and Linguistic Theory*, 28(3):593–642.

Noam Chomsky. 1981. *Lectures on Government and Binding: The Pisa Lectures*. Foris, Dordrecht.

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

Marina Ermolaeva. 2018. Morphological agreement in Minimalist grammars. In *Proceedings of Formal Grammar 2017*, pages 20–36, Berlin. Springer.

Thomas Graf. 2012. Locality and the complexity of Minimalist derivation tree languages. In *Formal Grammar 2010/2011*, volume 7395 of *Lecture Notes in Computer Science*, pages 208–227, Heidelberg. Springer.

Thomas Graf. 2018. Why movement comes for free once you have adjunction. To appear in *Proceedings of CLS 53*.

Henk Harkema. 2001. A characterization of Minimalist languages. In Philippe de Groote, Glyn Morrill, and Christian Retoré, editors, *Logical Aspects of Computational Linguistics (LACL'01)*, volume 2099 of *Lecture Notes in Artificial Intelligence*, pages 193–211. Springer, Berlin.

Jeffrey Heinz. 2018. The computational nature of phonological generalizations. In Larry Hyman and Frank Plank, editors, *Phonological Typology*, Phonetics and Phonology, chapter 5, pages 126–195. Mouton De Gruyter.

Jeffrey Heinz, Chetan Rawal, and Herbert G. Tanner. 2011. Tier-based strictly local constraints in phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 58–64.

Aravind Joshi. 1985. Tree-adjoining grammars: How much context sensitivity is required to provide reasonable structural descriptions? In David Dowty, Lauri Karttunen, and Arnold Zwicky, editors, *Natural Language Parsing*, pages 206–250. Cambridge University Press, Cambridge.

Gregory M. Kobele, Christian Retoré, and Sylvain Salvati. 2007. An automata-theoretic approach to Minimalism. In *Model Theoretic Syntax at 10*, pages 71–80.

Richard K. Larson. 1990. Double objects revisited: Reply to Jackendoff. *Linguistic Inquiry*, 21(4):589–632.

Sabine Laszakovits. 2018. Case theory in Minimalist grammars. In *Proceedings of Formal Grammar 2018*, pages 37–61, Berlin. Springer.

Alec Marantz. 1991. Case and Licensing. In *ESCOL '91: Proceedings of the Eighth Eastern states conference on linguistics*, pages 234–253.

Kevin McMullin. 2016. *Tier-Based Locality in Long-Distance Phonotactics: Learnability and Typology*. Ph.D. thesis, University of British Columbia.

Kevin McMullin and Gunnar Ólafur Hansson. 2015. Long-distance phonotactics as tier-based strictly 2-local languages. In *Proceedings of AMP 2014*.

Jens Michaelis. 2001. Transforming linear context-free rewriting systems into Minimalist grammars. *Lecture Notes in Artificial Intelligence*, 2099:228–244.

Frank Morawietz. 2003. *Two-Step Approaches to Natural Language Formalisms*. Walter de Gruyter, Berlin.

Ljiljana Progovac. 1998. Structure for coordination. *Glot International*, 3(7):3–9.

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, Berlin.

Edward P. Stabler. 2011. Computational perspectives on Minimalism. In Cedric Boeckx, editor, *Oxford Handbook of Linguistic Minimalism*, pages 617–643. Oxford University Press, Oxford.

Mai Ha Vu. 2018. Towards a formal description of NPI-licensing patterns. In *Proceedings of the Society for Computation in Linguistics*, volume 1, pages 154–163.