A TSL Analysis of Japanese Case

Kenneth Hanson
Department of Linguistics
Stony Brook University
Stony Brook, NY 11794, USA
kenneth.hanson@stonybrook.edu

Abstract

Recent work in subregular syntax has revealed deep parallels among syntactic phenomena, many of which fall under the computational class TSL (Graf, 2018, 2022). Vu et al. (2019) argue that case dependencies are yet another member of this class. But their analysis focuses mainly on English, which is famously case-poor. In this paper I present a TSL analysis of Japanese, which features a much wider range of case-marking patterns, adding support to the claim that case dependencies, and by extension syntactic dependencies, are TSL.

1 Introduction

Work on the computational complexity of strings has identified a rich hierarchy of subregular classes and shown that phonological patterns are among the simplest possible (Heinz, 2018). Local dependencies fall under the class of *strictly local* (SL) languages while most long-distance dependencies fall within a superclass of SL known as *tier-based strictly local* (TSL). These findings are of interest to both computational and general linguistics as they make strong typological predictions and inform development of learning algorithms (Lambert et al., 2021). A tantalizing possibility is that the tree-based equivalents of the string classes might reveal the same result in syntax. Graf (2018) generalizes SL and TSL to trees, and subsequent work (Graf and Shafiei, 2019; Vu et al., 2019; Graf, 2022, a.o.) presents preliminary evidence that many disparate syntactic phenomena are indeed TSL. But confirming this hypothesis requires much additional work, because the abstractness of syntactic representations makes it difficult to claim with certainty what structures are possible.

This paper focuses on the syntactic distribution of morphological case, which I define to be those heads or features realized by case morphology. In other words, we are not interested in the raw surface forms (which may exhibit accidental syncretism), but in the systematic distinctions among nominals made on the basis on their syntactic context. Vu et al. (2019) provides a proof of concept for a TSL analysis of case, focusing primarily on English. This work provides an analysis of Japanese, which features a much richer range of case patterns, including: (1) case marking conditioned by temporal properties of verbs, (2) lexical and structural dative case, (3) long-distance case marking in embedded clauses, and (4) case alternations in complex predicates. In addition to strengthening the claim that the syntactic distribution of case is TSL, the analysis also shows that case patterns that might otherwise be considered complex or surprising are in fact quite simple from a computational perspective.

The remainder of this paper is structured as follows. Section 2 introduces the computational background for establishing the TSL nature of syntactic dependencies. Section 3 provides an overview of the basic case patterns in Japanese, and proposes a set of descriptive generalizations. In Section 4, I define TSL grammars which encode these generalizations, then show how the analysis extends easily to more complex structures. Section 5 concludes.

2 Computational background

2.1 SL and TSL string languages

A strictly local (SL) language is characterized by a set of *forbidden substrings* of a fixed length k. For example, an SL grammar enforcing strict CV syllable structure consists of the set {V, CC, VV, C}, where $ stands for beginning/end of string. Words in this language include CV and CVCV but not CVCCV (which contains CC) or CVC (C$). Each forbidden substring is of length 2, making this a *strictly 2-local* (SL-2) language.[1]

TSL is a generalization of SL in which certain

[1] An equivalent definition of SL utilizes sets of *permissible substrings* of fixed length k, {C, CV, VC, V} in the case of the present example. Under this definition, a word is well-formed iff all of its length k substrings are well-formed.

Proceedings of the Society for Computation in Linguistics (SCiL) 2023, pages 15-24.
Amherst, Massachusetts, June 15-17, 2023

symbols are ignored. The remaining symbols are projected onto a *tier* in which elements that were formerly separated become adjacent, allowing a restricted type of long-distance dependency: a string is well-formed iff its tier conforms to a given SL grammar. A simple example from phonology is (symmetric) consonant harmony. Assuming an alphabet {a, m, s, ʃ}, we project only {s, ʃ}, and ban the substrings {sʃ, ʃs}. Words like 'samaas' (tier: 'ss') and 'ʃaʃaʃa' (ʃʃʃ) are part of this language but 'ʃamas' (ʃs) and 'saʃaʃa' (sʃʃ) are not. Since the forbidden substrings on the tier are of length 2, this language is TSL-2.

2.2 TSL in syntax

Graf (2018) generalizes TSL from strings to trees as follows. First, we project a *tree tier* which retains a subset of the original nodes, preserving dominance and precedence relations. The daughters of each node on the tier are then regulated by a TSL string language. This means that there are two opportunities to project a tier; we will take advantage this in of our treatment of adjuncts in Section 4.8.

Somewhat more formally, a TSL tree language is defined in terms of two functions: a *tier projection function* specifying which nodes to retain on the tree tier, and a *daughter string language function* which determines the constraints on the daughter string of each node.[2] Each function considers a finite local context of the argument node. I will use only the label of the node itself—a context of size 1, with one exception as discussed in Section 4.7.

For the syntactic formalism, I follow recent work (Graf, 2022; Graf and Shafiei, 2019) by adopting Minimalist Grammar (MG, Stabler 1997) dependency trees. These trees record the order of Merge steps in a Minimalist derivation: the rightmost child of a head is its complement, and other children are specifiers. Dependency trees are more compact than other representations while containing all necessary information about the derivation.

Consider the Japanese example in (1). This is a simple transitive sentence, in which the subject *Taroo* is followed by the nominative case marker *ga* and the object *piano* is followed by the accusative marker *o*. An X′-style phrase structure tree for this sentence is shown on the left of Figure 1. Details of the syntactic analysis will be introduced in Section 4. For now, it suffices to note that the case marker is the head of a K(ase) phrase, and that the subject KP asymmetrically c-commands the object KP.

(1) Taroo ga/*o piano o/*ga hiita.
Taroo NOM/ACC piano ACC/NOM played
'Taroo played the piano.'

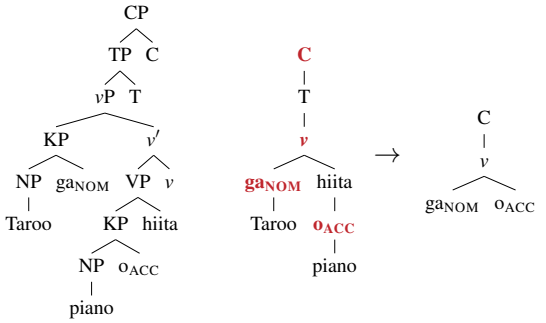


Figure 1: Left: phrase structure tree. Right: dependency tree and tier projection enforcing accusative constraint. Nodes of category K, C, and *v* are projected. It is required that every *o* has a *ga* among its left sisters.

On the right of Figure 1 is the dependency tree corresponding to the phrase structure tree, along with the case tier projection. Each node in the dependency tree is a lexical item, taking the place of a head and all of its projections in the phrase structure tree. *v* has two daughters, corresponding to its specifier (the subject, headed by the case marker) and its complement (VP, headed by the verb). Other nodes have only a single child, corresponding to their complements. A full dependency tree would display the features of each node; for brevity I omit everything but the node label and relevant features such as case, using the category as the label of empty elements such as C/T/*v*.

This brings us to the tier projection. The general approach taken in this paper will be to construct a tier such that all nominals in some case licensing domain become daughters of the domain node, and to state the constraints on case configurations over the daughter strings of the domain nodes. In the present example, the relevant constraint (simplified) is that the accusative marker *o* must be c-commanded by nominative *ga* in the same clause. To enforce this constraint, we project a tier which includes all nodes of category *v*, K, and C. Since dominance and precedence are preserved, *ga* and *o* become daughters of *v*. We then require that *ga* be a left sister of *o*.[3] The TSL grammar for the daughter string language of *v* will thus ban substrings such as *o ga*. The full analysis, which contains

[2] See Graf and Kostyszyn (2021) for a full definition.

[3] In principle it is possible for a left sister on the tier not to be c-commander in the dependency tree. In practice this turns out not to be an issue. See Section 4.8 for an example.

many additional constraints over several tiers, will be fleshed out in Section 4.

It is worth noting that in an MG dependency tree all elements appear in the position of first merge—when a nominal has moved, such as by passivization or scrambling, only its base position is considered for purposes of case licensing. This assumption has been adequate for all phenomena examined in this framework to date, and it also seems to be appropriate for case in Japanese. Scrambling, for example, is widely understood to preserve case marking. Even when a certain case correlates with movement (as in some analyses of differential object marking), it is usually possible to predict the case of nominal based on its context and its other features (e.g. definiteness). Since this issue does not arise in the Japanese data, I say no more here.

3 Basic case patterns

Japanese has four core cases, marked by the suffixes *ga* (nominative), *o* (accusative), *ni* (dative), and *no* (genitive). Their prototypical functions are similar to German and other Indo-European languages: subjects receive nominative case, direct objects receive accusative, and indirect objects receive dative, while complements and possessors of nouns are genitive. Examples of simple intransitive (2a), transitive (2b), and ditransitive sentences (2c) are given below, along with examples of a nominal complement (2d) and possessor (2e).[4] All examples are presented in topic-less sentences since topic marking masks the underlying case.[5]

(2) a. Taroo ga hasitta.
Taroo NOM ran
'Taroo ran.'

b. Taroo ga piano o hiita. (=1)
Taroo NOM piano ACC played
'Taroo played the piano.'

c. Jin ga Yumi ni hon o ageta.
Jin NOM Yumi DAT book ACC gave
'Jin gave Yumi a book.'

d. Taroo ga [yama no e] o mita.
Taroo NOM mountain GEN picture ACC saw
'Taroo saw a picture of a mountain.'

e. Taroo no hon
Taroo GEN book
'Taroo's book'

[4]Data is adapted from (Miyagawa, 1989) unless noted otherwise.

[5]Abbreviations: NOM = nominative, ACC = accusative, DAT = dative, LD = lexical dative, GEN = genitive, APPL = applicative, NPST = non-past, PASS = passive, IPASS = indirect passive, CAUS = causative.

In general, nominative, accusative, and dative case are available for arguments of verbs, and the number of arguments predicts what their cases should be: if there is one argument then it is nominative; if there are two then the latter is accusative, and if there are more than two then the middle nominals are dative. This is also true for complex verbal predicates, with some complications (discussed in Sections 4.6 and 4.7). Conversely, arguments of nouns are usually genitive no matter how many there are. An example of a noun phrase multiple genitive arguments is given in (3) below.

(3) Taroo no yama no e
Taroo GEN mountain GEN picture
'Taroo's picture of a mountain'

While these are the canonical patterns, several others are possible. Some transitive verbs take a dative object rather than the usual accusative (4a). Additionally, stative verbs such as *dekiru* 'can do' take a nominative object, and allow dative and/or nominative for the subject (this varies depending on the exact verb), yielding dative-nominative (4b) and double nominative (4c) structures. Transitive adjectives and complex verbs formed with a stative suffix also allow nominative objects.

(4) a. Taroo ga Yumi ni atta.
Taroo NOM Yumi DAT met
'Taroo met Yumi.'

b. Yumi ni tenisu ga dekiru.
Yumi DAT tennis NOM can.do
'Yumi can play tennis.'

c. Yumi ga tenisu ga dekiru.
Yumi NOM tennis NOM can.do
'Yumi can play tennis.'

Of the four cases, nominative has the widest distribution. As we will see later (Sections 4.5 and 4.7), it can also be replaced with another case when a verb or adjective and its arguments are embedded in a larger structure. Thus, it makes sense to treat nominative as the *default* case, appearing when no other condition applies.

To briefly summarize, the case that marks a nominal in some domain depends primarily on (1) the category of the domain and (2) the position of that nominal relative to others in the domain. Additionally, certain predicates specify that one of their arguments must be dative rather than the case that would otherwise be expected. Specifically, we could say that accusative and genitive are *structural* cases (i.e. licensed by the structural context); some instances of dative are structural while oth-

ers are *lexical* (licensed by specific lexical items); finally, nominative is the *default*.

I am not aware of any work in the syntactic literature that analyzes the entire case system of Japanese in this manner. However, the individual patterns are well-known, and the analysis is a direct application of ideas from dependent case theory (Marantz, 2000; Baker and Vinokurova, 2010). The primary purpose of this paper is to show that the generalizations outlined above are easily implemented using a TSL grammar, and that they can be extended to more complex constructions with little or no modification. As a computational analysis, it essentially descriptive in nature, and in principle compatible with a variety of theories of case licensing. At the same time, most of the patterns discussed here (or close analogues) can also be found in other rich case-marking languages, so there is good reason to believe that the approach should generalize beyond Japanese.

4 Analysis

4.1 Preliminaries

In this section, I will formalize the generalizations made in the previous section. To begin, I lay out a few syntactic assumptions. First, clauses are assumed to have the following functional hierarchy:

$$\text{C} > \text{T} > (\textsc{pass}) > (\textsc{caus}) > v > (\textsc{appl}) > \text{V}$$

In essence, this is a modern version of the "biclausal" analysis for the passive and causative constructions. These heads may be considered subtypes of *v*, labeled separately for convenience. Next, goals of ditransitive verbs may appear in two positions: low goals are PP daughters of VP, while high goals are KP daughters of an applicative head (Miyagawa and Tsujioka, 2004). This fact will be relevant to the analysis of passivization in Section 4.6. Finally, nominals are treated as NPs, with case markers occupying a higher KP.[6]

The remainder of this section is structured as follows. First, I introduce three tree tiers corresponding to structural cases licensed in the verbal domain, the nominal domain, and lexical case. Next, I consider more complex constructions, including embedded clauses, passives, and causatives, making several small revisions. From there, I refine the analysis to handle adjuncts, and address a potential problem involving coordination.

[6]PPs take an NP complement instead of a KP. PPs may alternatively be analyzed as KPs bearing semantic case. Such cases do not need to be licensed syntactically.

4.2 Accusative and structural dative case

First, we define a tier to license structural cases in the verbal domain: accusative and dative. On this tier we project non-stative *v* and all K heads. We also project C heads in order to limit the case licensing domain to a single clause; while *v* in the embedded clause will normally do this, it will not always be present on the tier.

The constraints on the tier are, roughly: (1) the rightmost of two or more K children of *v* must bear accusative, (2) the middle of three or more K heads must bear dative, and (3) no other K heads may bear accusative or dative. Other K heads are underspecified; if not specified as genitive or lexical dative on the relevant tiers they become nominative by default. This includes all subjects as well as objects of stative verbs (since stative *v* is not projected).

An example for the simple transitive sentence in (1) is given in Figure 2a. Since non-stative *v* is projected, the tier is unchanged from the example in Section 2.2. The daughter string of *v* satisfies the constraints just mentioned: the accusative K is the rightmost of two K children of *v* and the nominative K meets the elsewhere criterion. Further examples of well/ill-formed tiers are given in Figures 2b and 2c, respectively.

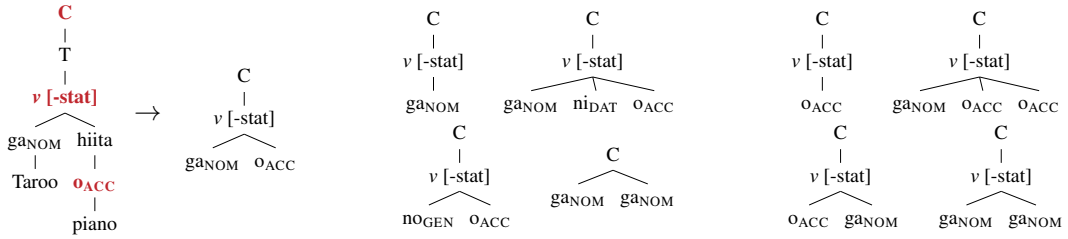
We must also take into account the fact that lexical datives are allowed as direct objects. It turns out that many verbs are compatible with either an accusative or dative object, with a difference in temporal properties (cf. Fukuda, 2007); I assume such verbs to be optional licensors of lexical dative case. The full tier definition is given below.[7]

(5) **Verbal case tier (initial version)**
Project categories: {*v*[-stat], K, C}
Daughter string languages:
v: $\{\textsc{nom}, \textsc{gen}, \textsc{ld}\} \cdot (\textsc{dat}^* \cdot \{\textsc{acc}, \textsc{ld}\})$
K/C: $\overline{\{\textsc{acc}, \textsc{dat}\}}^*$

For clarity, all daughter string languages are defined using regular expressions. Since it may not be immediately obviously that these languages are TSL (or SL), I also provide grammars for the verbal tier:

- The daughter string language of *v* is SL-2. The grammar (set of forbidden substrings) is {$ DAT, $ ACC, NOM NOM, NOM GEN, GEN

[7]String languages are notated using regular expressions. NOM/ACC/etc. stand for a K head bearing said case. A dot (·) represents concatenation. Set braces represent a choice among alternatives. An overbar represents set complement.



(a) Dependency tree and verbal case tier for a transitive sentence. The subject licenses the accusative object to its right.

(b) Other licit verbal case tiers. From left to right: intransitive verb, ditransitive verb, genitive subject, stative verb.

(c) Examples of illicit verbal case tiers. In order: lone accusative, double accusative, accusative before nominative, non-stative double nominative.

Figure 2: Examples of licit and illicit verbal case tiers.

NOM, GEN GEN, DAT $, DAT NOM, DAT GEN, ACC NOM, ACC GEN, ACC DAT, ACC ACC, ACC LD, LD NOM, LD GEN, LD LD}.

- The daughter string language of K/C is SL-1. The grammar for this language is {ACC, DAT}.

Small modifications to the verbal case tier will be required; the revised tier definition is given in Section 4.7. Also, while the current daughter string languages are SL, they will later be converted to TSL to accommodate adjuncts (Section 4.8).

4.3 Genitives

Next, we turn to genitives, which have the simplest distribution: as a first approximation, all KPs in the domain of a nominal are genitive, and no others. We construct the genitive tier as follows:

(6) **Genitive case tier (initial version)**
Project categories: {N, K, C}
Daughter string languages:
N: $\{\text{GEN}, \text{N}, \text{C}\}^*$
K/C: $\overline{\{\text{GEN}\}}^*$

The tier projection for (2d), in which the object nominal contains a genitive complement, is shown in Figure 3a. There is only a single K child the noun *e* 'picture', and it bears GEN as required, so the tier is well-formed. There are no restrictions on the other KPs, though they could of course be ruled out on other tiers.

Subjects of certain embedded clauses appear in genitive case, an apparent exception to the current tier definition; this will require modification as discussed in Section 4.5. Another issue worth noting is that the particle *no* can appear between PPs and their head nouns, as in example (7) below. This *no* is traditionally considered to be a marker of adnominal modification rather than a case particle. Fortunately, we can abstract away from this issue. If these instances of *no* are case particles, then they still adhere to the constraint as stated; if not, then the constraint simply does not apply.

(7) otera e no michi
temple to NO road
'the road to the temple'

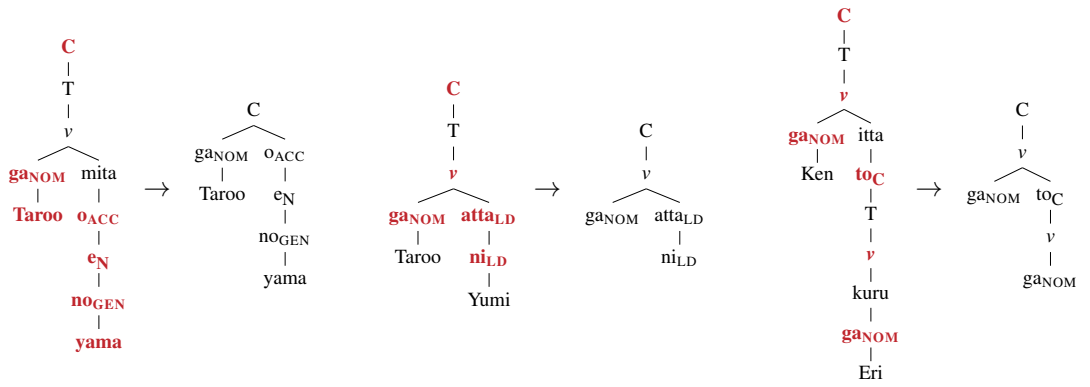
4.4 Lexical datives

The third case tier controls the distribution of lexically dative-marked nominals. While we could reasonably leave lexical case to be handled by the selection (i.e. subcategorization) mechanism, it is worth demonstrating that both structural and lexical case can be regulated in a unified manner if desired. Lexical dative may be assigned to an argument of either *v* or V depending on the verb, but only *v* appears on the verbal case tier, so a new tier is required which projects both. On this tier, verbal heads licensing a lexical dative KP must have exactly one such daughter; lexical dative KPs may appear nowhere else. The tier is defined as follows:

(8) **Lexical case tier**
Project categories: {*v*, V, K, C}
Daughter string languages:
v/V (LD licensor): $\overline{\{\text{LD}\}}^* \cdot \text{LD} \cdot \overline{\{\text{LD}\}}^*$
v/V (non-LD licensor): $\overline{\{\text{LD}\}}^*$
K/C: $\overline{\{\text{LD}\}}^*$

The tier projection for (4a), in which the verb requires a lexical dative object, is given in Figure 3b. The only KP child of V is a lexical dative, so the tier is well-formed. If the dative licensor was *v* then the subject would need to be dative instead.

While it might be desirable to use the same feature for both structural and lexical datives, this would prevent structural datives from being ruled out in subject and direct object position. Since lexical datives differ in behavior from other nominals (they cannot be passivized in Japanese, for example), such a distinction seems appropriate. In effect, we are treating structural and lexical dative



(a) Genitive tier for nominal complement. (b) Lexical case tier for dative object verb. (c) Verbal case tier for embedded CP.

Figure 3: Example tier projections for genitive and lexical dative, and an embedded clause.

as different cases.

One question that this analysis raises is what should happen if there are two KP children of the same lexical dative case licensor, in particular V. As far as I am aware, this situation never arises in Japanese. If it does in other languages, then the grammar must specify which KP should be dative.

Now, having defined three tiers modeling the canonical uses of the four core cases, we will consider more complex structures, and see that the system can handle them with minimal adjustment.[8]

4.5 Embedded clauses

There are several types of finite embedded clauses in Japanese. By default, these show the same case marking as matrix clauses, but under certain circumstances the embedded subject may be marked accusative or genitive.

We will first confirm that the analysis works correctly for the basic pattern. Examples of two types of finite embedded clauses are given in (9) below. Here, *to* is analyzed as a complementizer, while *koto* is a noun taking a CP complement.

[8] As noted by a reviewer, it has been suggested for phonology that when a dependency involves multiple tiers, the tier alphabets are either nested or disjoint, but never incomparable (Aksënova and Deshmukh, 2018). Since lexical dative is always assigned locally the tier projection could be expanded to all categories (in effect, an SL tree grammar), making it a superset of the others. It may also be possible to combine the verbal and genitive case tiers into a single tier, in which case the generalization would be upheld. But generally speaking we when we look at the whole system (not just case licensing) we expect overlapping tiers (Thomas Graf, p.c.).

(9) a. Ken ga [Eri ga kuru to] itta.
Ken NOM Eri NOM come C said
'Ken said that Eri will come.'

b. Eri ga [Ken ga tegami o okutta ∅]
Eri NOM Ken NOM tegami ACC sent C
koto o sitteiru.
thing ACC know
'Eri knows that Ken sent the letter.'

Since we project C on the tier, a new case domain is created for each embedded clause, resulting in the same case configuration as in a matrix clause. As an example, the tier projection for sentence (9a) is shown in Figure 3c. While projecting C may seem redundant, it is necessary because *v* is not projected on the verbal tier when it is stative. It also provides the basis for the analysis of the alternative case marking patterns, which we now turn to.

In the Japanese ECM construction, the embedded subject appears to take accusative case (10). If this nominal was a matrix object binding a *pro* subject in the embedded clause (a prolepsis analysis) then there would be nothing to explain, but Kishimoto (2018) argues that at least some ECM subjects are genuine. Similarly, in *ga-no conversion* the subject of a nominal clause takes genitive case (11). Both structures are also grammatical with a nominative embedded subject.

(10) Finite ECM (Kishimoto 2018)
Ken ga [Eri {ga/o} kawaii to] omotteiru.
Ken NOM Eri {NOM/ACC} be.cute C think
'Ken thinks that Eri is cute.'

(11) Ga-no conversion (Maki and Uchibori 2008)
Eri ga [Ken {ga/no} kita ∅] riyuu
Eri NOM Ken {NOM/GEN} came C reason
o sitteiru.
ACC know
'Eri knows the reason that Ken came.'

This variable cross-clausal case marking may

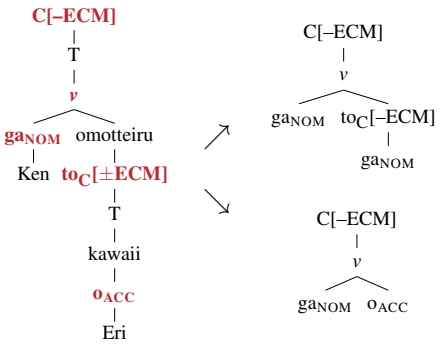


Figure 4: Verb case tier for embedded clause with and without ECM. C[+ECM] is not projected, so the embedded subject becomes part of the higher case domain.

seem mysterious, but in fact all that is needed to derive the patterns is to selectively ignore the embedded C head. The easiest way to do this is to posit that the relevant lexical predicates may select a C head with a special feature, call it [ECM]. Such C heads are not projected, similar to our treatment of stative verbs. This approach also has precedent in work which attributes variation in cross-clausal dependencies to the feature composition of the complementizer (cf. Lohninger et al., 2022). Indeed, Hiraiwa (2001) claims that ga-no conversion involves a special complementizer.

Example tier projections for (10) are shown in Figure 4 (the treatment of (11) is exactly parallel). Revised tier definitions are provided in Section 4.7. For simplicity, I treat the subject of an adjective as its complement, and ignore the aspectual morphology of *omotteiru*. All said, the facts about embedded clauses are handled quite well under the TSL perspective.

4.6 Passives

Next, we examine complex predicates within a single clause, formed with the passive suffix *-rare* and the causative suffix *-sase*. The passive suffix itself has at least two functions: the *direct passive*, which decreases the valency of a transitive verb by eliminating the agent, and the *indirect passive*, which increases valency. The literature disagrees on exactly how many distinct lexical items exist (see Ishizuka 2017 for an overview). I assume two homophonous passive suffixes corresponding to the two major functions. Recall also that I assume these suffixes to realize distinct functional heads, though the analysis could also work with verbs bearing 'passive' and 'causative' features.

The direct passive will be the focus on this section; the indirect passive will be discussed together with causatives. An example is given in (12).

(12) Active/passive transitive verb
  a. Sensei ga gakusei o hometa.
  teacher NOM student ACC praised
  'The teacher praised the student.'
  b. Gakusei ga (sensei ni) homerareta.
  student NOM teacher by praised.PASS
  'The student was praised (by the teacher).'

The object of a passivized transitive verb is promoted to the subject, and receives nominative case. These facts are straightforwardly understood under the common assumption that agent is not projected in Spec-*v*P in passives, and that the optional *by*-phrase is an adjunct PP. Miyagawa (1989) argues that this is indeed the case in Japanese.

For ditransitive verbs, there are two possibilities: either the (higher) goal is promoted, or the (lower) theme is promoted. Example (13) shows an active ditransitive verb along with the corresponding goal (13b) and theme (13c) passives (optional *by*-phrases are omitted).

(13) Active/passive ditransitive verb
  a. Mari ga kodomo ni okasi o ataeta.
  Mari NOM child DAT candy ACC gave
  'Mari gave the child candy.'
  b. Kodoma ga okasi o ataerareta.
  child NOM candy ACC gave.PASS
  'The child was given candy.'
  c. Okasi ga kodomo ni ataerareta.
  candy NOM child DAT gave.PASS
  'The candy was given to the child.'

The goal passive (13b) requires no special explanation assuming that dative case here is structural. Once the agent is eliminated, there are only two arguments, effectively creating a transitive verb. There is an elegant solution for the theme passive as well. Recall that the goal of a ditransitive verb may occupy one of two positions, and that the higher position is a KP while the lower position is a PP. Thus, it should be possible to target the direct object for promotion by selecting the low goal structure. So we see that the facts about passives fall out naturally in this analysis. Dependency trees and verbal case tiers for the ditransitive goal passive and theme passive are shown in Figure 5.

4.7 Causatives

As our final case study, we consider the causative construction. The causative morpheme *sase* is compatible with verbs of any valency. Causative equivalents of the examples in (2) are given in (14) be-

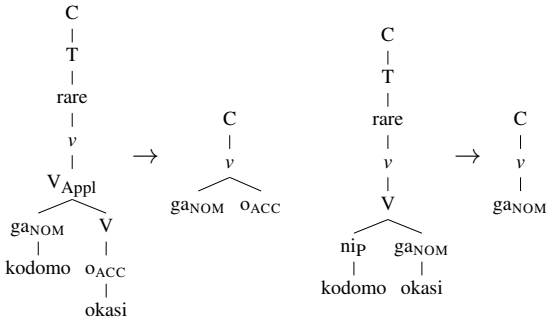


Figure 5: Verbal case tiers for passives of ditransitive. Left: goal passive (goal is a KP). Right: theme passive (goal is a PP).

low. The causee of an intransitive verb may be accusative or dative, corresponding to the *make* or *let* interpretations, respectively. For other verbs the causee must be dative, and either interpretation is possible. I set aside these semantic details.

(14) a. Ken ga Taroo {ni/o} hasiraseta.
Ken NOM Taroo DAT/ACC ran.CAUS
'Ken made/let Taroo run.'

b. Ken ga Taroo ni piano o hikaseta.
Ken NOM Taroo DAT piano ACC played.CAUS
'Ken made/let Taroo play the piano.'

c. Ken ga Jin ni Yumi ni hon o
Ken NOM Jin DAT Yumi DAT book ACC
agesaseta.
gave.CAUS
'Ken made/let Jin give Yumi a book.'

In an intransitive sentence the causee may be dative without an accompanying accusative object, suggesting lexical dative case. Additional arguments appear in the expected cases, suggesting that these are the usual structural cases. But in the present system it is the causer that would receive dative case from *sase*, not the causee. Furthermore, it is possible to passivize a causative, in which case the causee becomes nominative like any other structurally case-marked nominal, as shown in (15).

(15) Taroo ga (Ken ni) hasiraserareta.
Taroo NOM Ken by run.CAUS.PASS.PAST
'Taroo was made to run (by Ken).'

There are several possible solutions. One is to add additional case tiers corresponding to each functional head, allowing each to restrict the case of the first K child of that head. So, a new *causative tier* would determine the case of causee, leaving the case of the causer up to the next higher tier. While this is technically possible, a more elegant solution makes use of the context-sensitive nature of Graf's (2018) tier projection and daughter string functions in the definition of the verbal case tier. We select the *highest v* head in each clause, that is, the one selected by T, increasing the context to a window of height 2. Then, we let the identity of the *v* head determine its daughter string language.

The indirect passive (adversative passive) can be handled in the same manner. Examples of this construction are given in (16) below.

(16) a. Ken ga Taroo ni hasirareta.
Ken NOM Taroo DAT run.PASS.PAST
'Ken was annoyed by Taroo running.'

b. Ken ga Taroo ni piano o
Ken NOM Taroo DAT piano ACC
hikareta.
play.PASS.PAST
'Ken was annoyed by Taroo playing the piano.'

Unlike in the causative construction, the embedded subject always receives dative case. We define the daughter string language of the indirect passive head accordingly. The revised definitions for both the verbal and genitive tiers are given in (17) below. A comparison of the old and new verbal case tier is shown in Figure 6.

(17) **Verbal case tier (revised)**
Project categories:
{*v*[-stat]/CAUS/IPASS daughter of T, K, C[–ECM]}
Daughter string languages:

v: $\{\text{NOM}, \text{GEN}, \text{LD}\} \cdot (\text{DAT}^* \cdot \{\text{ACC}, \text{LD}\})$

CAUS: $\{\text{NOM}, \text{GEN}\} \cdot \left\{ \begin{matrix} \{\text{ACC}, \text{DAT}\} \\ \text{DAT}^* \cdot \{\text{ACC}, \text{LD}\} \end{matrix} \right\}$

IPASS: $\{\text{NOM}, \text{GEN}\} \cdot \left\{ \begin{matrix} \text{DAT} \\ \text{DAT}^* \cdot \{\text{ACC}, \text{LD}\} \end{matrix} \right\}$

K/C: $\overline{\{\text{ACC}, \text{DAT}\}}^*$

(18) **Genitive case tier (revised)**
Project categories: {N, K, C[–ECM]}
Daughter string languages:
N: $\{\text{GEN}, \text{N}, \text{C}\}^*$
K/C: $\overline{\{\text{GEN}\}}^*$

4.8 Adjuncts

Adjuncts such as adverbs and PPs interfere with the tier constraints as currently defined, since there is in principle no bound to the number that may appear in any given case domain. We would like to ignore them since their presence does not affect case licensing. However, we cannot omit them on the tree tiers because any K heads they dominate would be interspersed among the daughters of a higher head. Instead, we must modify the daughter string languages, converting them from SL to TSL languages and ignoring adjuncts at this stage.

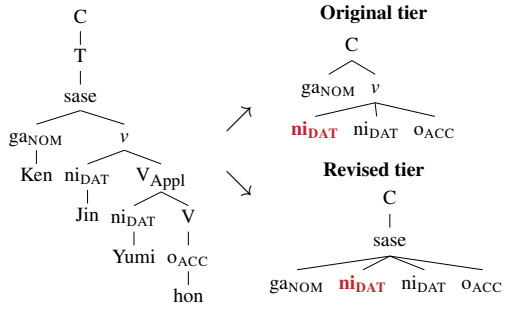


Figure 6: Original verbal case tier (top right) and revised version (bottom right) for causative ditransitive. In the original analysis, the causee (Jin) is the first K child of *v*, where it cannot be dative. In the revised version, all verbal arguments are daughters of *sase*, allowing the causee to be assigned dative in the usual manner.

To see why this works, recall again the form of the daughter string language of *v* on the verbal case tier, which (simplified) has the form $a(b^*c)$. Representing adjuncts by the symbol x and allowing them to occur anywhere, our string language is now $x^*a((x^*b)^*x^*c)x^*$. If we project a tier omitting x then the tier language is once again $a(b^*c)$.

While I cannot go into detail for reasons of space, the approach I have in mind makes use of *category-preserving selection* by means of adjunctizer heads. For example, the adjunctizer head for adjectives selects for categories A and N and itself is an N. The number of such heads in any MG lexicon is finite. We add all such heads to all *tree* tier projection functions, but omit them from them the daughter string tier projection function as just described.

4.9 Coordination

Due to the complexity of the data and the number of theories of coordination, it is beyond the scope of this paper to consider these in any depth, but I wish to at least outline the general problem and what it means for the analysis. Essentially, coordination is a problem when it splits a case domain, such as when a *v*P contains a coordinated VP. When we project the verbal case tier, the children of both Vs all end up as daughters of *v*; this predicts a change in case marking, which is contrary to fact.

Does this issue actually arise in Japanese? Perhaps not. Japanese allows coordination of TP and *v*P but not VP, and subjects may optionally remain in situ (cf. Hirata, 2006, and references therein). This means that whether each verb phrase has its own subject (remaining in situ), or both share a single subject (raised via across-the-board movement), there is no conflict. In a language similar where VP coordination is possible, we would need to restructure the analysis to include additional nested case domains (we avoided this earlier for the passive and causative constructions by using structure-sensitive tier projection). Should this prove unfeasible, this seems to be the most likely way in which the tree tier-based analysis could be invalidated.

It is at this point that I should note an alternative generalization of TSL to trees based on so-called *c[ommand]-strings* which, roughly speaking, encode chains of c-commanding elements (Graf and Shafiei, 2019). Because the present analysis already operates by collecting nominals in the daughter string of the case licensing domain node, it should be straightforward to recast it in terms of c-strings. This new version would also be robust against the domain-splitting problem presented by coordination. I leave the investigation of this possibility to future work.

5 Conclusion

In this paper, I developed a TSL analysis of Japanese case, and showed that the descriptive generalizations are captured neatly with a system of three tiers and a small number of constraints, and that the analysis extends with minimal modification to a wide range of constructions. The analysis is simple in computational terms and concise as a description of the case patterns of Japanese. These results support the proposal that the syntactic distribution of morphological case is TSL.

As mentioned earlier, the case patterns discussed in this paper also have close parallels in other languages. In particular, ergative case as analyzed in dependent case theory fits neatly into the current system, as does variation in case marking according to tense or aspect. It seems likely that this type of computational analysis can bring insight into our understanding of case marking across languages.

Acknowledgments

This work was supported by the National Science Foundation under Grant No. BCS-1845344. I thank Thomas Graf for his feedback throughout this project. I also thank three anonymous reviewers for their comments. Reviewer 3 in particular provided numerous suggestions that improved the readability of the final paper.

References

Alëna Aksënova and Sanket Deshmukh. 2018. Formal restrictions on multiple tiers. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2018*, pages 64–73.

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

Shin Fukuda. 2007. Object case and event type: Accusative-dative object case alternation in Japanese. In *Annual Meeting of the Berkeley Linguistics Society*, volume 33, pages 165–176.

Thomas Graf. 2018. Why movement comes for free once you have adjunction. *Proceedings of CLS*, 53:117–136.

Thomas Graf. 2022. Typological implications of tier-based strictly local movement. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2022*, pages 184–193.

Thomas Graf and Kalina Kostyszyn. 2021. Multiple wh-movement is not special: The subregular complexity of persistent features in Minimalist Grammars. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2021*, pages 275–285.

Thomas Graf and Nazila Shafiei. 2019. C-command dependencies as TSL string constraints. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, pages 205–215.

Jeffrey Heinz. 2018. The computational nature of phonological generalizations. *Phonological Typology, Phonetics and Phonology*, pages 126–195.

Ken Hiraiwa. 2001. On nominative-genitive conversion. *MIT Working Papers in Linguistics*, 39:66–125.

Ichiro Hirata. 2006. Predicate coordination and clause structure in Japanese. *Linguistic Review*, 23(1).

Tomoko Ishizuka. 2017. The passive voice. In *Handbook of Japanese Syntax*, pages 403–446. De Gruyter Mouton.

Hideki Kishimoto. 2018. On exceptional case marking phenomena in Japanese. *Kobe Papers in Linguistics*, 11:31–49.

Dakotah Lambert, Jonathan Rawski, and Jeffrey Heinz. 2021. Typology emerges from simplicity in representations and learning. *Journal of Language Modelling*, 9(1):151–194.

Magdalena Lohninger, Iva Kovač, and Susanne Wurmbrand. 2022. From Prolepsis to Hyperraising. *Philosophies*, 7(2).

Hideki Maki and Asako Uchibori. 2008. Ga/no conversion. In *The Oxford handbook of Japanese linguistics*, pages 192–216. Oxford University Press.

Alec Marantz. 2000. Case and licensing. In *Arguments and case: Explaining Burzio's generalization*, pages 11–30. John Benjamins.

Shigeru Miyagawa. 1989. *Structure and Case Marking in Japanese*. Academic Press.

Shigeru Miyagawa and Takae Tsujioka. 2004. Argument structure and ditransitive verbs in Japanese. *Journal of East Asian Linguistics*, 13(1):1–38.

Edward P. Stabler. 1997. Derivational minimalism. In Christian Retore, editor, *Logical Aspects of Computational Linguistics*. Springer.

Mai Ha Vu, Nazila Shafiei, and Thomas Graf. 2019. Case assignment in TSL syntax: A case study. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, pages 267–276.