Excluding SVO in Ergative Languages: A New View on Mahajan's Generalisation

Antje Lahne

Goal of this talk:
Derive Mahajan's Generalisation.

(1) *Mahajan's Generalisation (Mahajan 1994, 1997):*
Ergative languages exhibit SOV and VSO order, but not SVO.

1 Previous Analyses

1.1 Mahajan (1994, 1997)

Basic property of the analysis:
SVO, therefore not ergative.

Goals of the analysis:
I. Derive that verb-peripheral languages exhibit ergative case marking pattern, whereas SVO languages are never ergative.
II. Derive that the auxiliary *have* usually occurs in SVO languages only.

Table 1
Mahajan's Generalisation

	ERGATIVITY	AUX 'HAVE'
VERB-PERIPHERAL LANGUAGES	√	rare
VERB-MEDIAL LANGUAGES	—	√

Background Assumption:

(2) *Extended UTAH (Mahajan 1994, 1997):*
Identical (thematic) relationships between items not ONLY have identical structural relationships but they are also represented by identical categorial structures. That is, the categories that are involved in representing these structural relationships are also the same.

Observations:
1. In Hindi, the subject is a PP marked with an ergative postposition[1]; in French, on the other hand, the subject is an unmarked NP.
2. The auxiliary is *have* in French, and *be* in Hindi.

(3) a. Rām-ne vah kitā-bẽ paṛʰ-ĩī hɛ̃
Ram-ERG.MASC those book-FEM.PL read-PERF.F.PL be:PRES.PL
'Ram has read those books' (Hindi)

1. Evidence for the ergative marker being a postposition: (1) can be separated from the NP by an emphatic marker; (2) does not appear twice on a coordinated NP.

b. Jean a cuit les tomates
Jean has cooked the tomatoes
'Jean has cooked the tomatoes' (French)

Analysis:

– Consequence of the Extended UTAH: The category of the subject must be the same in (3a) and (3b).

– The subject in French is originally a PP (of which the head correponds to the Hindi P head *ne* 'ERG').

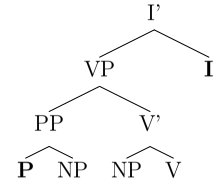
– *have* = BE + incorporated empty adposition (P originates as the sister of the external argument; see Freeze 1992; Kayne 1993).

– In French, P is incorporated into I, which yields the *have*-auxiliary. In Hindi, on the other hand, P cannot incorporate into I – it spells out as the ergative marker.

Question:
Why can P not incorporate into I in Hindi?

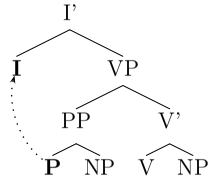
Answer:
This is due to a condition on incorporation which is not fulfilled in verb-final languages like Hindi: Incorporation requires adjacency.

(4) Structure in SOV languages:



⇝ P and I are not adjacent, so that P-I incorporation is impossible. Thus, I surfaces as *be*, and P surfaces as the ergative marker on NP_{ext}.

(5) Structure in SVO languages:



⇝ P and I are adjacent, so that P-I incorporation takes place, P+I then surface as the auxiliary *have*, and the external argument does not bear an oblique case marker; it surfaces with nominative case.

Results:
1. SVO languages never yield ergative morphology, because P, which otherwise surfaces as the ergative marker, is always incorporated into I in the relevant context.
2. Perfective constructions like (3b) are always underlyingly ergative.

1.2 Bittner and Hale (1996)

Basic property of the analysis:
Ergative, therefore not SVO.

Recap:
The system works in such a way that the case of primary arguments is determined by

two heads, K_1 (=[erg]=I) and K_2 (=[acc]=v). Case "assignment" is by case binding; if a K cannot case-bind an argument, then the derivation does not crash, but the argument is assigned the default case (=nom). There are two conditions on case binding: K must c-command the argument, and there must be a case competitor (caseless coargument) in the local m-command domain of K.

In ergative languages, K_1 case-binds the external argument, and K_2 does not determine a structural case. Thus, NP_{int} does not receive structural case, but it is a case competitor for NP_{ext}. The case competitor can become visible to I in two ways:
1. by being remerged as SpecI;
2. by V-to-I movement (head movement opens barriers).

Explanation for absence of SVO order in ergative languages (Bittner and Hale 1996:14):
In ergative languages, I already case-binds the external argument in NP_{ext}'s base position (=edge of V)[2]. Hence, there is no motivation for NP_{ext} to additionally move to the edge of I.

(6) *Last Resort (Chomsky 1991, 1995):*
Movement must result in feature checking.

Result: if NP_{ext} does not need to move, then it must not move. NP_{ext} is therefore never remerged as SpecI; which means that SVO cannot be derived in ergative languages.

2 New Analysis

2.1 Background

Basic property of the analysis:
Both ergativity and *SVO result from independent properties of a certain class of languages.

Starting point:
SO is a derived word order. It can be derived from both SOV and SVO orders.

(7) Deriving VSO from SVO
S V O → V S __ O

Question:
How is VSO derived in SOV languages? Two possible ways:

(8) Deriving VSO from SOV: one step
S O V → V S O __

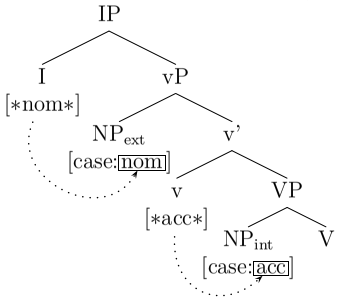
(9) Deriving VSO from SOV: two steps (via SVO)
S O V → S V O __ → V S __ O __

Proposal:
In ergative languages, VSO order is not derived via the intermediate SVO step (i.e., derivation (9) does not arise in ergative languages). Reason: In ergative languages, NP_{ext} and NP_{int} both end up as specifiers of the same head; consequently, there is no position between them that the verb could possibly fill. In accusative languages, on the other hand, NP_{ext} and NP_{int} end up as specifiers of different heads, so that there is a head between them that the verb can fill. The underlying reason for this different behaviour is that in ergative languages, in contrast to accusative languages, case assignment dependencies are nested, as shown in (10).

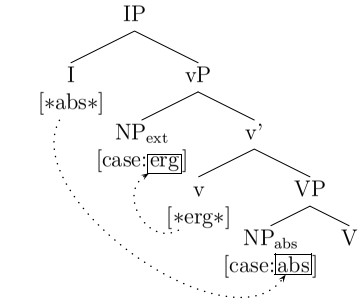
2. That is possible because the case competitor NP_{int} becomes visible to I if V-to-I movement takes place, which opens the VP barrier.

(10) Case assignment patterns

a. Case assignment in accusative languages (pretheoretical, without phases)



b. Case assignment in ergative languages (pretheoretical, without phases)



2.2 Constraints

(11) *Earliness Principle/ Cyclicity:*
An uninterpretable feature must be marked for deletion as early in the derivation as possible (Pesetsky 1989; Pesetsky and Torrego 2001).

(12) *Argument structure:*
v assigns internal case, I assigns external case; the internal argument is merged as the sister of V, the external argument as Spec,v (e.g. Murasugi 1992).

(13) There is only one structural argument encoding feature [case], which has two values: ext(ernal) and int(ernal) (determined with respect to vP, the predicate domain). The feature specification [CASE:ext] replaces [abs] and [nom]; and [CASE:int] replaces [erg], [acc] (e.g. Müller 2008).

(14) *Agree:*
α can agree with β with respect to a feature bundle Γ iff (a), (b) and (c) hold:
(a) α bears a probe feature [∗F∗] in Γ, β bears a matching goal feature [F] in Γ.
(b) α m-commands β.
(c) There is no δ such that (i) and (ii) hold:
(i) δ is closer to α than β (= the path from δ to α is shorter than the path from β to α).
(ii) δ has a feature [F].

(15) *Path:*
The path from A to B is the set of categories C such that (a) and (b) hold:
(a) C dominates A or B.
(b) The minimal XP that dominates C is the minimal XP that dominates both A and B.

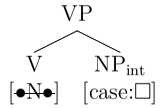
Consequence of the path definition:
The length of a path is determined by its cardinality. Specifier and complement of a head α are equally close to α; the specifier of α is closer to α than any category that is further embedded in the complement of α (Pesetsky 1982; Collins 1994). Elements at the edge of the complement of α are equidistant to α.

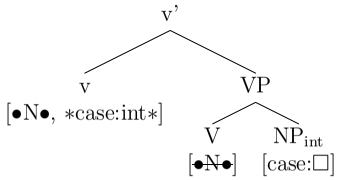
2.3 Analysis

Note:
The structures given in the analysis are head-initial. This is in order to show that SVO cannot be derived in ergative languages even if the language is underlyingly head-initial (which in fact advantages verb-medial structures).

2.3.1 VP/vP

Context: simple transitive predicate with NP_{int}, NP_{ext}. Common derivational steps for both accusative and ergative language types:

(16)


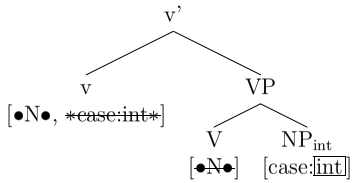
(17)


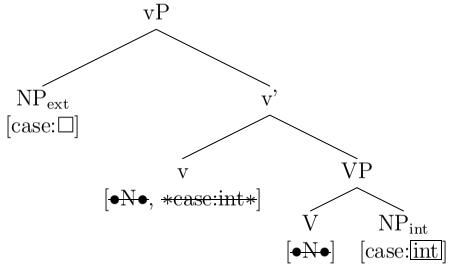
Observation (Müller 2008):
At this point, an indeterminacy in rule application arises: The next operation could either be Value(v, NP_{int}), or Merge (v, NP_{ext}). The Earliness Principle has the effect that if Merge and/or Agree are possible at a stage of the derivation, then they must apply at once.

Consequence:

> [T]here is a dilemma: Only one operation can apply first, as required by Earliness. [...] I would like to contend that conflicts of this type are real, and must be resolved in a language by giving one Earliness requirement priority over the other in the case of conflict – in other words, in ranking the two requirements. [This] is in fact all that needs to be assumed to derive the core difference between accusative and ergative encoding patterns. (Müller 2008)

Scenario in accusative languages:
Some languages resolve the indeterminacy in rule application by giving Agree priority over Merge. Order of feature handling: [∗case:int∗] ≫ [•N•]. This yields an accusative pattern.
Derivation: At first, v agrees with NP_{int} in [∗case:int∗], then NP_{ext} is merged.

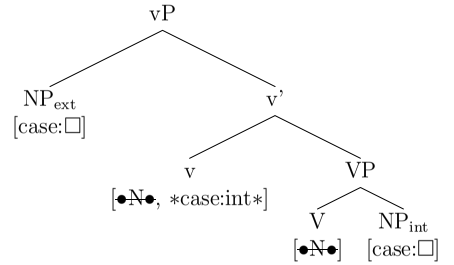
(18)


(19)


Result: NP_{ext} does not receive case at this point. NP_{ext} is at the edge of v.

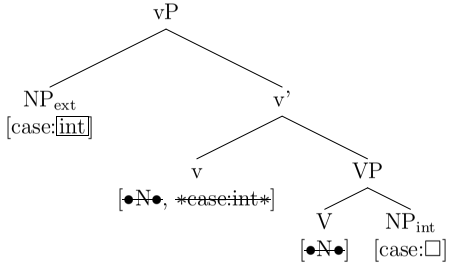
Scenario in ergative languages:
Some languages resolve the indeterminacy in rule application by giving Merge priority over Agree. Order of feature handling: [•N•] ≫ [∗case:int∗]. This gives rise to an ergative pattern.

Derivation: At first, NP_{ext} is merged.

(20)


Now [∗case:int∗] must be assigned. But: it cannot be assigned to NP_{int}. Reason: NP_{ext} is closer to v.

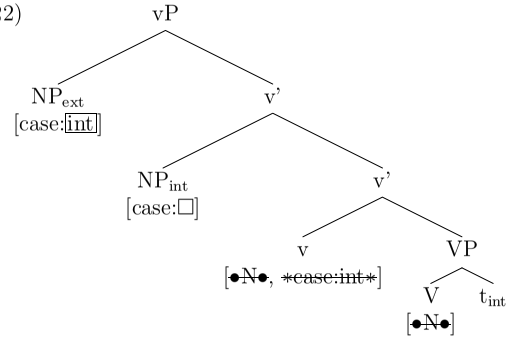
Result: NP_{ext} is assigned internal case.

(21)


Now NP_{int} must be remerged at the edge of v. Reason: The complement of the phase head v is spelled out as soon as the vP is complete; however, the case feature of NP_{int} has not been valued yet.

Assumption:
There is a shape preservation movement at this stage of the derivation: NP_{ext} is remerged again, yielding SOV. Or: Due to shape preservation, NP_{int} is not merged as the highest specifier, but tucked in (Richards 1997) lower than NP_{ext}.

(22)



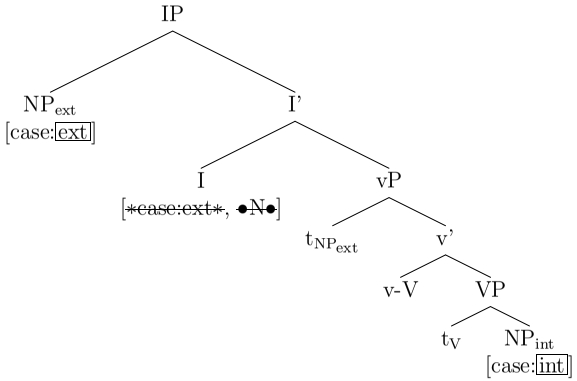
2.3.2 IP

I is merged. The feature set of I contains [∗case:ext∗], and can contain am EPP feature [•N•]. Again, an indeterminacy in rule application arises: The next operation could either be Agree(I, NP_{int}), or Merge (I, NP_{ext}).

Scenario in accusative languages:
In languages that finally show an accusative pattern, the order of operation application is Agree ≫ Merge. Thus, the order of feature handling of I is [∗case:ext∗] ≫ [•N•].

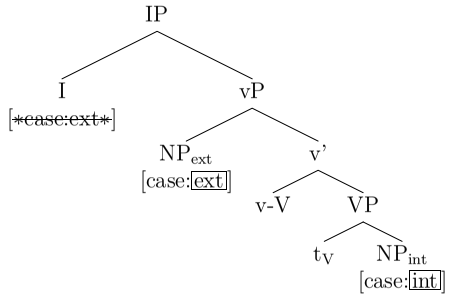
▶ Case 1: I has an EPP feature.
Derivation: First, I values the case feature of NP_{ext}. Then NP_{ext} is remerged as SpecI.
Result: No matter if V-to-I movement takes place or not, this yields **SVO**.

(23)



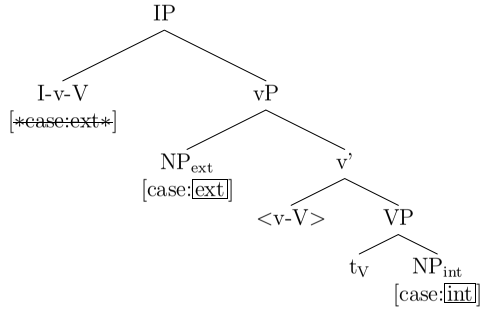
▶ Case 2: I has no EPP feature. There is no V-to-I movement.
Derivation: No relevant indeterminacy in rule application arises in the first place. I values the case feature of NP_{ext}.
Result: This yields **SVO**.

(24)



▶ Case 3: I has no EPP feature. There is V-to-I movement.
Derivation: V moves to I. No relevant indeterminacy in rule application arises. I values the case feature of NP_{ext}.
Result: This yields **VSO**.

(25)



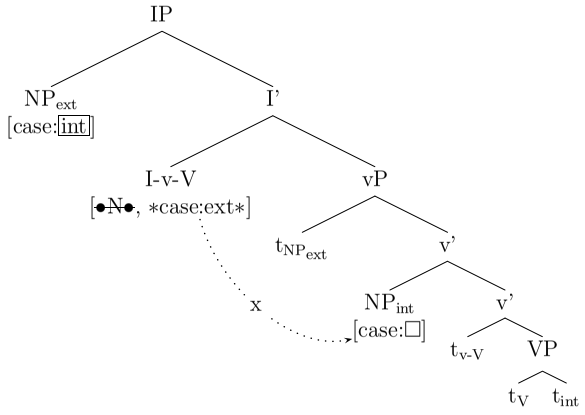
Scenario in ergative languages:
In languages that finally show an ergative pattern, the order of operation application is Merge ≫ Agree. Thus the order of feature handling of I is [•N•] ≫ [∗case:ext∗].

▶ Case 1: I has an EPP feature. There is V-to-I movement.
Derivation: V moves to I. Next, I remerges NP_{ext} as its specifier. This would yield SVO!

But: [∗case:ext∗] must be handled next. The derivation encounters the same constellation as within the vP: NP_{ext} is closer to I than NP_{int}. The case feature of NP_{ext} is already valued.

Problem: I cannot value the case feature of NP_{int}, because NP_{ext} is an defective intervener for Agree. This follows from the definition of intervener in (14c-ii): NP_{ext} is closer to I than NP_{int} is, and NP_{ext} has a case feature.

(26)



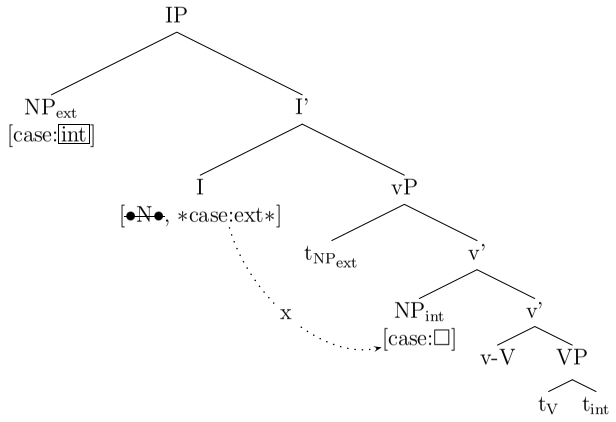
Result: The derivation crashes.

Note:
The same constellation would not lead to an intervention effect in an accusative language, as Agree features are handled before structure-building features in this language type.

▶ Case 2: I has an EPP feature. There is no V-to-I movement.
Derivation: First, I remerges NP_{ext} as its specifier. This would yield SOV.

But: The same reasoning as above applies: [∗case:ext∗] must be handled next, but cannot be handled by agreeing with NP_{int} (due to defective intervention).

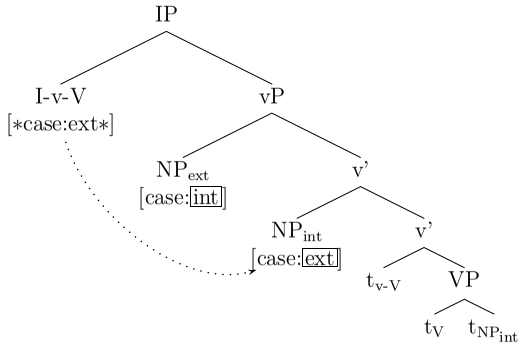
(27)



Result: The derivation crashes.

▶ Case 3: I has no EPP feature. There is V-to-I movement.
Derivation: No indeterminacy in rule application arises. V-to-I movement takes place, and I values the case feature of NP_{int}.
Result: This yields **VSO**.

(28)

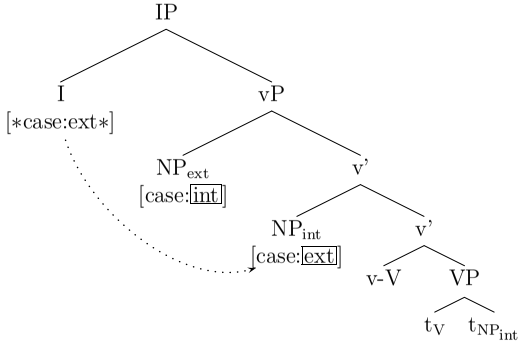


Question:
Why is NP_{ext} not a defective intervener for agree in (28)?

Answer:
NP_{ext} and NP_{int}, both specifiers of v, are *equidistant*. This follows from the definition of *Path* in (15b): "The minimal XP that dominates C is the minimal XP that dominates both A and B." According to this definition, the node v' in (28) does not count as a step on the path between I and NP_{int}, because the minimal XP that dominates it is vP, which does not dominate I. NP_{ext} and NP_{int} are therefore equidistant from I.

▶ Case 4: I has no EPP feature. There is no V-to-I movement.
Derivation: No indeterminacy in rule application arises. I values the case feature of NP_{int}.
Result: This yields **SOV**.

(29)



A Problem:[3]
There is evidence that in Hindi (SOV, split-ergative), I has an EPP feature, and the ergative argument is a specifier of I (Anand and Nevins 2006).

Solution:
There *is* a way of "switching off" defective intervention: NP_{int} is last-resort-moved to the edge of I and tucked in below the internal argument (for word order reasons). Then NP_{ext} and NP_{int} are equidistant, and I can assign case to NP_{int}. This yields SOV.

3. Marc Richards, p.c.

2.4 Summary

– In ergative languages, S and O are both specifiers of v ⇝ there is no position between them that the V could possibly fill. If S is extracted, then it blocks case assignment to O, so that the derivation crashes.
In accusative languages, S is the specifier of I, and O is the complement of V ⇝ there are head positions between them that V can fill. Extraction of S does not make the derivation crash as case assignment happens before S is extracted.

– Reason for this different behaviour: Order of Merge and Agree, which leads to nested case assignment dependencies in ergative languages.

References

Anand, Pranav and Andrew Nevins (2006): The Locus of Ergative Case Assignment: Evidence from Scope. *In:* A. Johns, D. Massam and J. Ndayiragije, eds, *Ergativity: Emerging Issues*. Kluwer, Dordrecht, pp. 3–25.

Bittner, Maria and Ken Hale (1996): 'The Structural Determination of Case and Agreement', *Linguistic Inquiry* **27**(1), 1–68.

Chomsky, Noam (1991): Some Notes on Economy of Derivation and Representation. *In:* R. Freidin, ed., *Principles and Parameters in Comparative Grammar*. MIT Press, Cambridge, Mass., pp. 417–454.

Chomsky, Noam (1995): *The Minimalist Program*. MIT Press, Cambridge, Mass.

Collins, Chris (1994): 'Economy of Derivation and the Generalized Proper Binding Condition', *Linguistic Inquiry* **25**(1), 45–61.

Freeze, R. (1992): 'Existentials and Other Locatives', *Language* **68**, 553–595.

Kayne, Richard S. (1993): 'Toward a Modular Theory of Auxiliary Selection', *Studia Linguistica* **47**(1), 3–31.

Mahajan, Anoop (1994): The Ergativity Parameter: *Have-be* Alternation, Word Order and Split Ergativity. In: *Proceedings of NELS 24*. GLSA, Amherst, pp. 317–331.

Mahajan, Anoop (1997): Universal Grammar and the Typology of Ergative Languages. *In:* A. Alexiadou and T. A. Hall, eds, *Studies on Universal Grammar and Typological Variation*. John Benjamins, Amsterdam, pp. 35–57.

Müller, Gereon (2008): 'Ergativity, Accusativity, and the Order of Merge and Agree'. Manuscript, Universität Leipzig.

Murasugi, Kumiko (1992): Crossing and Nested Paths. PhD thesis, MIT, Cambridge, Mass.

Pesetsky, David (1982): Paths and Categories. PhD thesis, Massachusetts Institute of Technology, Cambridge, Mass.

Pesetsky, David (1989): 'Language-particular Processes and the Earliness Principle'. Manuscript, MIT, Cambridge, Mass.

Pesetsky, David and Esther Torrego (2001): T to C movement: Causes and Consequences. *In:* M. Kenstowicz, ed., *Ken Hale: A Life in Language*. MIT Press, MIT, Cambridge, Mass., pp. 355–426.

Richards, Norvin (1997): What Moves Where When in Which Language?. PhD thesis, Massachusetts Institute of Technology, Cambridge, Mass.