Grammatik verbaler Argumente
Argumentkodierung 3: Minimalistische Ansätze

Gereon Müller

Institut für Linguistik
Universität Leipzig

WiSe 2006/2007

www.uni-leipzig.de/∼muellerg

Overview

Three minimalist analyses:

1. Bobaljik (1993): 1 = ERG, NOM, 2 = ACC, ABS
2. Murasugi (1992): 1 = NOM,ABS, 2 = ACC,ERG
3. Bittner & Hale (1996): 1 = ERG, 2 = ACC, 3 = NOM,ABS

Ergativity in Yup′ik

(1) Yup′ik (canonical ergative pattern):

a. Angute-m qusngiq ner-aa
man-ERG reindeer-ABS eat-+TRANS.3s/3s
'The man is eating the reindeer.'

b. Qusngiq ner′-uq
reindeer-ABS eta-–TRANS.3s
'The reindeer is eating.'

Focus of Bobaljik (1993):

The three basic argument encoding patterns (ergative, accusative, active); not: argument-type based, clause-type based, aspect/tense based splits.

Bobaljik (1993): Basic Assumptions

The analysis follows Levin & Massam (1985), and particularly Chomsky (1993):
The cases of primary arguments are determined by two different syntactic heads K_1, K_2 (e.g.: $K_1 = Agr_s$, $K_2 = Agr_o$). The two language types are identical with respect to V_t contexts; in V_i contexts, there are differences. Only K_2 is "activated" in ergative languages, and only K_1 is "activated" in accusative languages.

1. ERG, NOM $\rightarrow K_1$
2. ABS, ACC $\rightarrow K_2$

Bobaljik (1993): Basic Assumptions

The analysis follows Levin & Massam (1985), and particularly Chomsky (1993):
The cases of primary arguments are determined by two different syntactic heads K_1, K_2 (e.g.: $K_1 = Agr_s$, $K_2 = Agr_o$). The two language types are identical with respect to V_t contexts; in V_i contexts, there are differences. Only K_2 is "activated" in ergative languages, and only K_1 is "activated" in accusative languages.

1. ERG, NOM $\rightarrow$ K_1
2. ABS, ACC $\rightarrow$ K_2

The morphological marking problem (Chomsky (1993)):
"The "active" element typically assigns a less-marked Case to its Spec."
(Chomsky (1993))

The Obligatory Case Parameter

(2) Obligatory Case:
Case X is obligatorily assigned/checked.

(3) Obligatory Case Parameter (OCP):

a. In nominative/accusative languages, CASE X is nominative (= ERG).

b. In ergative/absolutive languages, CASE X is absolutive (= ACC).

The Obligatory Case Parameter

(2) Obligatory Case:
Case X is obligatorily assigned/checked.

(3) Obligatory Case Parameter (OCP):

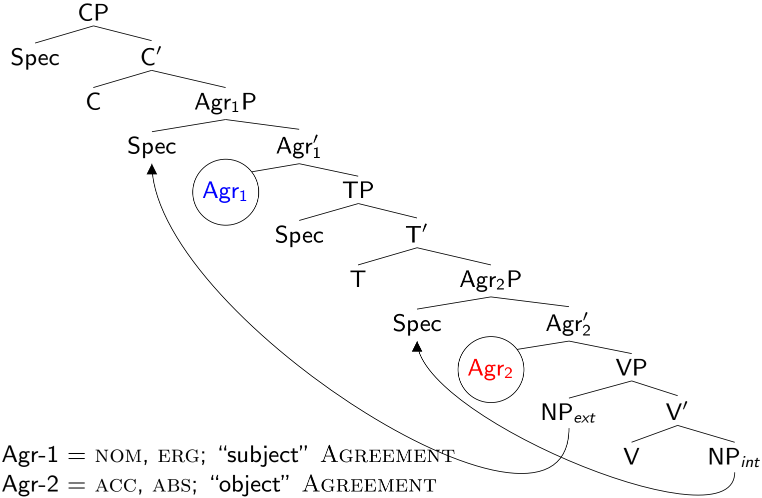
a. In nominative/accusative languages, CASE X is nominative (= ERG).

b. In ergative/absolutive languages, CASE X is absolutive (= ACC).

"Presumably, the observed morphological tendency towards null morphology for these Cases is a reflection of this obligatory status." (Bobaljik (1993, 51))

Clause Structure

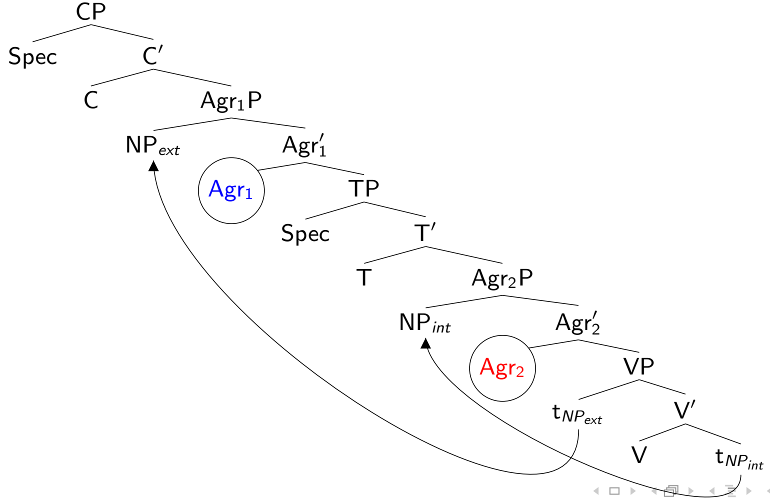
(4) Clause Structure (assumed by Bobaljik (1993)):



Agr-1 = NOM, ERG; "subject" AGREEMENT
Agr-2 = ACC, ABS; "object" AGREEMENT

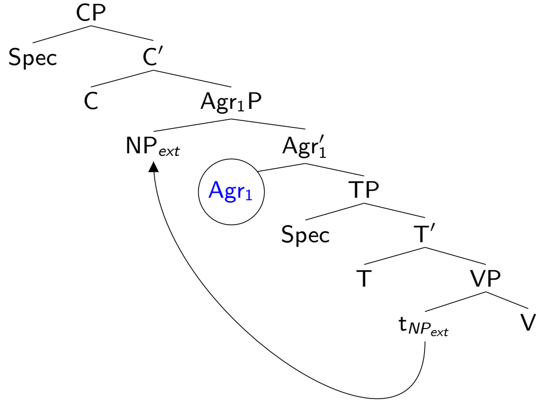
Transitive Clauses: Ergative and Accusative Patterns

(5) NP_{ext}-V_t and NP_{int}-V_t move to case positions in transitive causes:



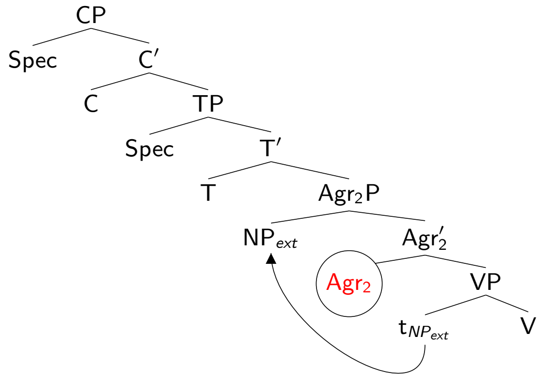
Intransitive Clauses: Accusative Patterns

(6) NP_{ext}-V_i (or NP_{int}-V_i) moves to SpecAgr$_1$ in transitive causes:



Intransitive Clauses: Ergative Patterns

(7) NP_{ext}-V_i (or NP_{int}-V_i) moves to SpecAgr$_2$ in transitive causes:



Argument Realization

Prediction:
Unless further assumptions are made, syntactic ergativity is not expected (argument realization is uniform).

(8) *Reflexivization*:

a. Mary$_1$ saw herself$_{1/*2}$ (in the mirror)
b. *Herself$_1$ saw Mary$_1$ (in the mirror)

(9) *Principles A and B of Chomsky's (1981) Binding Theory*:

a. A: An anaphor must be A-bound in its governing category.
b. B: A pronoun must be A-free in its governing category.

(10) *Binding*:
α A-binds β iff (a) and (b) hold:

a. α and β are coindexed.
b. α c-commands β.

Note: Reciprocals are anaphors in the sense of (9).

Basque Reciprocals

The evidence shows that the external argument binds the internal argument in transitive contexts, not vice versa.

(11) Reciprocals in Basque:

a. mutil-ek elkar ikusi dute
   boys-ERG each other-ABS see AUX.3sA/3pE
   'The boys saw each other.'

b. *elkar-rek mutil-ak ikusi ditu(zte)
   each other-ERG boys-ABS see AUX.3pA/3sE(3pE)
   'The boys saw each other.'

Abkhaz Reflexive Agreement

Abkhaz (North East Caucasian; Georgia) ensures argument encoding via head marking. Still, there is evidence that the NP_{ext} asymmetrically c-commands NP_{int}.

(12) Reflexive Agreement in Abkhaz:

a. l-xe y-l-ba-yt′
3sf-head(n) 3snA-3sfE-see-PRES
'She sees herself.'

b. s-xe y-z-ba-yt′
1s-head 3snA-1sE-see-PRES
'I see myself.'

Inuit Reflexive Possessives

(13) *Inuit Reflexive Possessives*:

a.

Piita-up	anaana-ni	nagligi-ja∼a
Piita-ERG	mother-POSS.3s/refl/ABS	love-3s/3s

'Piita loves his mother.' (his = Piita)

b.

Piita-up	anaana-∼a	nagligi-ja∼a
Piita-ERG	mother-POSS.3s/ABS	love-3s/3s

'Piita loves his mother.' (*his = Piita)

(14) *Impossible anaphoric binding into external arguments*:

a.

*Anaana-mi	Piita	nagligi-ja∼a
mother.3s/REFL/ERG	Piita-ABS	love-3s/3s

'*His$_1$ mother loves Piita$_1$.'

b.

Anaana-∼ata	Piita	nagligi-ja∼a
mother.3s/ERG	Piita-ABS	love-3s/3s

'His$_{1/2}$ mother loves Piita$_1$.'

This is exactly the same pattern as in the accusative language Russian.

Russian Reflexive Possessives

(15) *Russian Reflexive Possessives*:

a. Ol′ga ljubit svoju mamu
Ol′ga-NOM loves her.REFL-ACC mother-ACC
'Ol′ga loves her mother.' (her = Ol′ga)

b. Ol′ga ljubit eë mamu
Ol′ga-NOM loves her-ACC mother-ACC
'Ol′ga loves her mother.' (*her = Ol′ga)

(16) *Impossible anaphoric binding into external arguments*:

a. *Svoja mama ljubit Ol′gu
her.REFL-NOM mother-NOM loves Ol′ga-ACC
'*Her$_1$ mother loves Ol′ga.'

b. Eë mama ljubit Ol′gu
her-NOM mother-NOM loves Ol′ga-ACC
'Her$_{1/2}$ mother loves Ol′ga$_1$.'

Weak Crossover

Weak crossover is a further diagnostic to determine argument hierarchies in syntax (via asymmetric c-command).

(17) Weak crossover in English:

a. Who$_1$ t$_1$ loves his$_1$ mother ?

b. *Who$_2$ did his$_2$ mother love t$_2$?

Weak Crossover

Weak crossover is a further diagnostic to determine argument hierarchies in syntax (via asymmetric c-command).

(17) Weak crossover in English:

a. Who$_1$ t$_1$ loves his$_1$ mother ?
b. *Who$_2$ did his$_2$ mother love t$_2$?

The same effect occurs in the ergative language Nisgha.

(18) Weak crossover in Nisgha (Tsimshian, Western Canada):

a. næt ʔæn-sip′ən-s nɔx^w-t
who-3E REL-love-DM mother-3s
'Who$_1$ loves his$_1$ mother?'

b. næ-gat ɬ ti-sip′ən-s nɔx^w-t
who-one ND FOC-love-DM mother-3s
'*Who$_1$ does his$_1$ mother love?'
'Who$_1$ does his$_2$ mother love?'

Active Argument Encoding Patterns 1

A consequence of the analysis:
Ergative case can only be assigned in transitive environments. Therefore, active patterns (as in Basque, Guaraní, Hindi, Georgian) should not exist.

Active Argument Encoding Patterns 1

A consequence of the analysis:
Ergative case can only be assigned in transitive environments. Therefore, active patterns (as in Basque, Guaraní, Hindi, Georgian) should not exist.

Bobaljik's (1993) assumption (also see Laka (1993), Nash (1996), Bittner & Hale (1996), and many others):
Unergatives are transitive!
In languages with an active ergative argument encoding pattern, what looks like an intransitive verb is in fact a transitive verb with a hidden object (which can, e.g., be overtly realized as a cognate object ('dream a dream')); sometimes the presence of the internal argument is indicated by overt agreement morphology (Basque).

Note:
This is in line with certain theories of argument structure, e.g., the approach taken in Hale & Keyser (2002).

Active Argument Encoding Patterns 1

A consequence of the analysis:
Ergative case can only be assigned in transitive environments. Therefore, active patterns (as in Basque, Guaraní, Hindi, Georgian) should not exist.

Bobaljik's (1993) assumption (also see Laka (1993), Nash (1996), Bittner & Hale (1996), and many others):
Unergatives are transitive!
In languages with an active ergative argument encoding pattern, what looks like an intransitive verb is in fact a transitive verb with a hidden object (which can, e.g., be overtly realized as a cognate object ('dream a dream')); sometimes the presence of the internal argument is indicated by overt agreement morphology (Basque).

Note:
This is in line with certain theories of argument structure, e.g., the approach taken in Hale & Keyser (2002).

Question: What about pure ergative encoding patterns, as in Yup'ik, Archi, Sierra Popoluca? In these languages, the internal argument of the relevant verbs is incorporated into V; hence, V becomes intransitive.

Active Argument Encoding Patterns 2

A minimal pair: Yup'ik vs. Basque. (Both languages are pro-drop languages; from the absence of an overt argument, one cannot tell whether it is present in syntax or not.)

(19) Yup'ik:

a. John-am ner-aa
   John-ERG eat-3s/3s
   'John ate it.' — not: 'John ate.'

b. John ner'-uq
   John-ABS eat-3s
   'John ate.' — not: 'John ate it.'

(20) Basque:

a. Jon-ek jaten du
   Jon-ERG eat AUX
   'Jon ate it.'

b. Jon-ek jaten du
   Jon-ERG eat AUX
   'Jon ate.'

Potential Problems

1. The correlation with morphological marking (ERG,ACC vs. NOM,ABS) is not straightforward.
2. Does the approach to active patterns based on Basque generalize to all languages that instantiate this pattern?
3. How can person-based, aspect-based, or clause-type based split ergativity be integrated into the analysis?
4. What about the well-established cases of syntactic ergativity?
5. The ergative/accusative parameter is closely tied to movement of NP arguments.

Potential Problems

1. The correlation with morphological marking (ERG,ACC vs. NOM,ABS) is not straightforward.
2. Does the approach to active patterns based on Basque generalize to all languages that instantiate this pattern?
3. How can person-based, aspect-based, or clause-type based split ergativity be integrated into the analysis?
4. What about the well-established cases of syntactic ergativity?
5. The ergative/accusative parameter is closely tied to movement of NP arguments.
6. It is likely that NPs can check (or assign, or valuate) case without undergoing movement; see the concept of Agree in Chomsky (2001; 2005).

Murasugi (1992): Basic Assumptions 1

- The cases of primary arguments are determined by two different syntactic heads K_1, K_2 (e.g.: $K_1 = Agr_s$, $K_2 = Agr_o$). In V_i contexts, the two language types are identical (only K_1 can determine case). In V_t contexts, K_2 is "strong" in ergative languages; and K_1 is "strong" in accusative languages.

Murasugi (1992): Basic Assumptions 1

- The cases of primary arguments are determined by two different syntactic heads K_1, K_2 (e.g.: $K_1 = Agr_s$, $K_2 = Agr_o$). In V_i contexts, the two language types are identical (only K_1 can determine case). In V_t contexts, K_2 is "strong" in ergative languages; and K_1 is "strong" in accusative languages.
  Assumption: Strong K attracts the highest NP argument.

Murasugi (1992): Basic Assumptions 1

- The cases of primary arguments are determined by two different syntactic heads K_1, K_2 (e.g.: $K_1 = Agr_s$, $K_2 = Agr_o$). In V_i contexts, the two language types are identical (only K_1 can determine case). In V_t contexts, K_2 is "strong" in ergative languages; and K_1 is "strong" in accusative languages.
  Assumption: Strong K attracts the highest NP argument.
  Consequence: Embedded vs. nesting paths in ergative vs. accusative languages.
  1. ERG, ACC $\rightarrow K_2$
  2. NOM, ABS $\rightarrow K_1$

(Murasugi (1992), Jelinek (1993))

Murasugi (1992): Basic Assumptions 2

Murasugi's (1992) main idea:

- The assumptions about syntactic phrase structure are similar to those made in Chomsky (1993), Bobaljik (1993).

Murasugi (1992): Basic Assumptions 2

Murasugi's (1992) main idea:

- The assumptions about syntactic phrase structure are similar to those made in Chomsky (1993), Bobaljik (1993).
- However, case assignment is reversed.

Murasugi (1992): Basic Assumptions 2

Murasugi's (1992) main idea:

- The assumptions about syntactic phrase structure are similar to those made in Chomsky (1993), Bobaljik (1993).
- However, case assignment is reversed.
- ergative = accusative, nominative = absolutive.

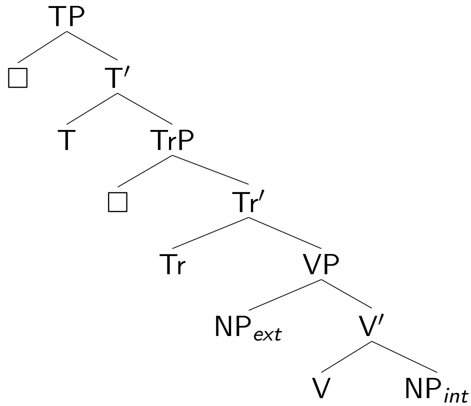
Murasugi (1992): Basic Assumptions 2

Murasugi's (1992) main idea:

- The assumptions about syntactic phrase structure are similar to those made in Chomsky (1993), Bobaljik (1993).
- However, case assignment is reversed.
- ergative = accusative, nominative = absolutive.
- The distinction between crossing paths and nesting paths is crucial.

Clause Structure

(21)



Remarks on Clause Structure

Murasugi's phrase structure is modern; e.g., it anticipates the analysis in Chomsky (1995; 2001):

Remarks on Clause Structure

Murasugi's phrase structure is modern; e.g., it anticipates the analysis in Chomsky (1995; 2001):

- Tr is nowadays usually called v.

Remarks on Clause Structure

Murasugi's phrase structure is modern; e.g., it anticipates the analysis in Chomsky (1995; 2001):

- Tr is nowadays usually called v.
- Agr_O and Agr_S are gone (cf. the meta-gramamtical tenet that there can be no semantically uninterpretabale functional projections; see Chomsky (1995)).

Remarks on Clause Structure

Murasugi's phrase structure is modern; e.g., it anticipates the analysis in Chomsky (1995; 2001):

- Tr is nowadays usually called v.
- Agr_O and Agr_S are gone (cf. the meta-gramamtical tenet that there can be no semantically uninterpretabale functional projections; see Chomsky (1995)).

Difference between TrP and vP:

Remarks on Clause Structure

Murasugi's phrase structure is modern; e.g., it anticipates the analysis in Chomsky (1995; 2001):

- Tr is nowadays usually called v.
- Agr_O and Agr_S are gone (cf. the meta-gramamtical tenet that there can be no semantically uninterpretabale functional projections; see Chomsky (1995)).

Difference between TrP and vP:

- Tr checks/assigns structural case (so does v).
- Tr does not introduce an external argument NP_{ext} (in contrast to v).

Assumptions about Case Assignment 1

Accusative pattern:

1. T checks nominative (case and agreement).
2. Tr checks accusative (case and agreement).

Assumptions about Case Assignment 1

Accusative pattern:

1. T checks nominative (case and agreement).
2. Tr checks accusative (case and agreement).

Ergative pattern:

1. T checks absolutive (case and agreement).
2. Tr checks ergative (case and agreement).

Assumptions about Case Assignment 1

Accusative pattern:

1. T checks nominative (case and agreement).
2. Tr checks accusative (case and agreement).

Ergative pattern:

1. T checks absolutive (case and agreement).
2. Tr checks ergative (case and agreement).

Markedenss:

1. The case that is checked by T is an unmarked case (morphologically less marked, or not marked at all; citation form).
2. The case that is checked by Tr is a marked case (morphologically more marked, not a citation form)

Assumptions about Case Assignment 2

- NPs bear case (including morphological markers); however, (structural) case must be checked.

Assumptions about Case Assignment 2

- NPs bear case (including morphological markers); however, (structural) case must be checked.
- Case is checked by movement of an NP to the specifier of T/Tr.

Assumptions about Case Assignment 2

- NPs bear case (including morphological markers); however, (structural) case must be checked.
- Case is checked by movement of an NP to the specifier of T/Tr.
- Case-driven movement takes place in the syntax (overtly) or at LF (= an abstract level of representation, i.e., covertly).

Assumptions about Case Assignment 2

- NPs bear case (including morphological markers); however, (structural) case must be checked.
- Case is checked by movement of an NP to the specifier of T/Tr.
- Case-driven movement takes place in the syntax (overtly) or at LF (= an abstract level of representation, i.e., covertly).
- Syntactic movement is triggered by strong features. LF movement is triggered by weak features.

Further Prerequisites

Φ-features:

1. Φ-features are located on NPs.

Further Prerequisites

Φ-features:

1. Φ-features are located on NPs.
2. Φ-features are located on V, for all primary arguments (NP_{ext}-V_t, NP_{ext}-V_i, NP_{int}-V_t, NP_{int}-V_i).

Further Prerequisites

Φ-features:

1. Φ-features are located on NPs.
2. Φ-features are located on V, for all primary arguments (NP_{ext}-V_t, NP_{ext}-V_i, NP_{int}-V_t, NP_{int}-V_i).
3. Φ-features are not located on T or Tr.

Further Prerequisites

Φ-features:

1. Φ-features are located on NPs.
2. Φ-features are located on V, for all primary arguments (NP_{ext}-V_t, NP_{ext}-V_i, NP_{int}-V_t, NP_{int}-V_i).
3. Φ-features are not located on T or Tr.
4. In order to check Φ-features of V and NP, V must undergo movement to F, and NP must undergo movement to SpecF (where F is a functional head).

Further Prerequisites

Φ-features:

1. Φ-features are located on NPs.
2. Φ-features are located on V, for all primary arguments (NP_{ext}-V_t, NP_{ext}-V_i, NP_{int}-V_t, NP_{int}-V_i).
3. Φ-features are not located on T or Tr.
4. In order to check Φ-features of V and NP, V must undergo movement to F, and NP must undergo movement to SpecF (where F is a functional head).

Tense features:

1. T:[+tense] $\longrightarrow$ finite clause

Further Prerequisites

Φ-features:

1. Φ-features are located on NPs.
2. Φ-features are located on V, for all primary arguments (NP_{ext}-V_t, NP_{ext}-V_i, NP_{int}-V_t, NP_{int}-V_i).
3. Φ-features are not located on T or Tr.
4. In order to check Φ-features of V and NP, V must undergo movement to F, and NP must undergo movement to SpecF (where F is a functional head).

Tense features:

1. T:[+tense] $\longrightarrow$ finite clause
2. T:[–tense] $\longrightarrow$ non-finite clause

Further Prerequisites

Φ-features:

1. Φ-features are located on NPs.
2. Φ-features are located on V, for all primary arguments (NP_{ext}-V_t, NP_{ext}-V_i, NP_{int}-V_t, NP_{int}-V_i).
3. Φ-features are not located on T or Tr.
4. In order to check Φ-features of V and NP, V must undergo movement to F, and NP must undergo movement to SpecF (where F is a functional head).

Tense features:

1. T:[+tense] $\longrightarrow$ finite clause
2. T:[–tense] $\longrightarrow$ non-finite clause

Transitivity features:

1. Tr: [+trans] $\longrightarrow$ V takes two primary arguments.

Further Prerequisites

Φ-features:

1. Φ-features are located on NPs.
2. Φ-features are located on V, for all primary arguments (NP_{ext}-V_t, NP_{ext}-V_i, NP_{int}-V_t, NP_{int}-V_i).
3. Φ-features are not located on T or Tr.
4. In order to check Φ-features of V and NP, V must undergo movement to F, and NP must undergo movement to SpecF (where F is a functional head).

Tense features:

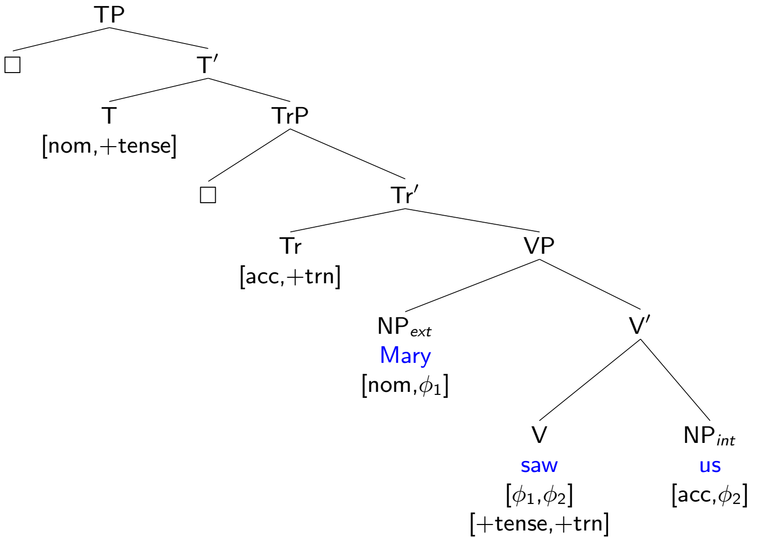
1. T:[+tense] $\longrightarrow$ finite clause
2. T:[–tense] $\longrightarrow$ non-finite clause

Transitivity features:

1. Tr: [+trans] $\longrightarrow$ V takes two primary arguments.
2. Tr: [–trans] $\longrightarrow$ V takes one primary argument.

Example

(22)



TP
□
T′
T
[nom,+tense]
TrP
□
Tr′
Tr
[acc,+trn]
VP
NP$_{ext}$
Mary
[nom,ϕ_1]
V′
V
saw
[ϕ_1,ϕ_2]
[+tense,+trn]
NP$_{int}$
us
[acc,ϕ_2]

Example

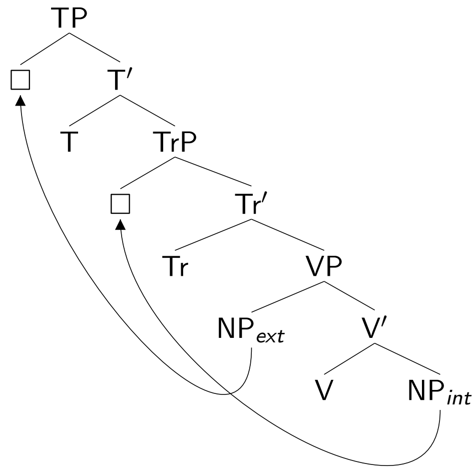
(22)

- TP
  - □
  - T′
    - T [nom,+tense]
    - TrP
      - □
      - Tr′
        - Tr [acc,+trn]
        - VP
          - NP_{ext} Mary [nom,ϕ_1]
          - V′
            - V saw [ϕ_1,ϕ_2] [+tense,+trn]
            - NP_{int} us [acc,ϕ_2]

Note:
There is no alternation of the two structure-building operations Merge and Move yet (Chomsky (2001; 2005)). Rather:
All movement operations (Move) follow all basic structure-building operations (Merge).

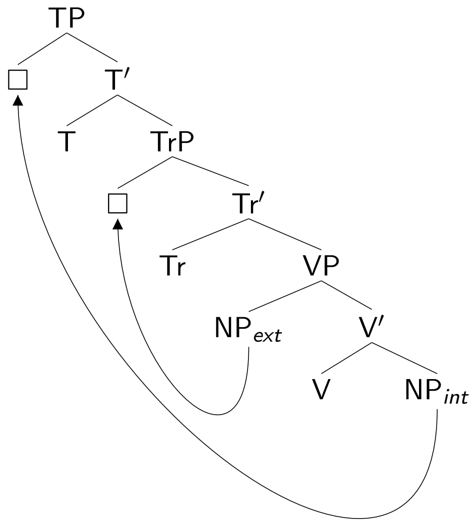
Accusative pattern: crossing paths

(23)



Ergative pattern: nesting paths

(24)



The System

(25) The ergative/accusative parameter:

a. The case feature of T is strong in an accusative language.
b. The case feature of Tr is strong in an ergative language.

The System

(25) The ergative/accusative parameter:

a. The case feature of T is strong in an accusative language.
b. The case feature of Tr is strong in an ergative language.

(26) Economy principles (moderatley updated terminology):

a. Minimal Goal ('Closest Available Source'):
At all levels of the derivation, a probe attracts the closest available NP.

The System

(25) The ergative/accusative parameter:

a. The case feature of T is strong in an accusative language.
b. The case feature of Tr is strong in an ergative language.

(26) Economy principles (moderatley updated terminology):

a. Minimal Goal ('Closest Available Source'):
At all levels of the derivation, a probe attracts the closest available NP.
b. Minimal Probe ('Closest Featured Target'):
At all levels of the derivation, a goal NP must be moved to the closest available probe.

The System

(25) The ergative/accusative parameter:

a. The case feature of T is strong in an accusative language.
b. The case feature of Tr is strong in an ergative language.

(26) Economy principles (moderatley updated terminology):

a. Minimal Goal ('Closest Available Source'):
At all levels of the derivation, a probe attracts the closest available NP.
b. Minimal Probe ('Closest Featured Target'):
At all levels of the derivation, a goal NP must be moved to the closest available probe.
c. Procrastinate:
A syntactic operation applies as late as possible.
("Covert (LF) movement is cheaper than overt movement.)

Note: "goal" here stands for the target of the operation, it does not refer to the Θ-role of the same name.

Consequences 1

Assumptions about case-driven movement of NPs:

- At a given level of representation, a goal NP has to be the NP that is closest to the minimal probe before any movement takes place in order to be eligible for movement.
- A goal NP has to be available for movement; i.e., it must not have checked its case features yet.

Consequences 1

Assumptions about case-driven movement of NPs:

- At a given level of representation, a goal NP has to be the NP that is closest to the minimal probe before any movement takes place in order to be eligible for movement.
- A goal NP has to be available for movement; i.e., it must not have checked its case features yet.

Consquence:

- Only one NP can undergo case-driven movement in overt syntax: NP_{ext} ('subject').
- Case-driven movement of NP_{int} ('object') takes place at LF.

Consequences 2

"Suppose that both T and Tr have strong Case features [...], requiring movement to their Specs at S-structure [...] The closest NP to both T and Tr is [...] the subject. However, this NP cannot satisfy the feature requirements of both functional heads simultaneously. Therefore, unless something else is inserted in SpecT to satisfy T, the derivation will crash." (p.25-26)

"At any one level, then, there will be neither Crossing nor Nested Paths (i.e., the result of both subject and object raising), but only independent movements of subjects to functional specs."

Consequences 3

Comment:
This last consequence is potentially empirically problematic. However, closer inspection reveals that it is probably not essential.

Consequences 3

Comment:
This last consequence is potentially empirically problematic. However, closer inspection reveals that it is probably not essential.

- Given a minimally revised notion of availability in the definition of the economy principles in (26), both crossing and nesting paths are permitted on a single level or representation.

Consequences 3

Comment:
This last consequence is potentially empirically problematic. However, closer inspection reveals that it is probably not essential.

- Given a minimally revised notion of availability in the definition of the economy principles in (26), both crossing and nesting paths are permitted on a single level or representation.

Procrastinate

Question:
Why is the constraint Procrastinate needed?

Procrastinate

Question:
Why is the constraint Procrastinate needed?

Answer:
Procrastinate ensures that movement that is triggered by weak features is confined to LF.
(It is not so clear whether this assumption is actually needed in the present context.)

Procrastinate

Question:
Why is the constraint Procrastinate needed?

Answer:
Procrastinate ensures that movement that is triggered by weak features is confined to LF.
(It is not so clear whether this assumption is actually needed in the present context.)

Note:
Movement of an internal argument NP_{int} at LF may violate the Strict Cycle Condition (see Chomsky (1973)) (or the Extension Condition of Chomsky (1993)) verletzen.

Transitive Clauses, Accusative Pattern

(27) a. Base structure before movement:
$\mathrm{T}_{[+\mathit{tense}],[\mathit{nom}]}$ $\mathrm{Tr}_{[+\mathit{trn},\mathit{acc}]}$ [John saw Mary]

Transitive Clauses, Accusative Pattern

(27) a. Base structure before movement:
$T_{[+\mathit{tense}],[\mathit{nom}]}$ $Tr_{[+\mathit{trn},\mathit{acc}]}$ [John saw Mary]
b. Overt syntactic movement:
$John_1$ $T_{[+\mathit{tense}],[\mathit{nom}]}$ $Tr_{[+\mathit{trn},\mathit{acc}]}$ [t_1 saw Mary]
c. *$Mary_2$ $T_{[+\mathit{tense}],[\mathit{nom}]}$ $Tr_{[+\mathit{trn},\mathit{acc}]}$ [John saw t_2]

Transitive Clauses, Accusative Pattern

(27)
a. Base structure before movement:
$T_{[+tense],[nom]}$ $Tr_{[+trn,acc]}$ [John saw Mary]
b. Overt syntactic movement:
$John_1$ $T_{[+tense],[nom]}$ $Tr_{[+trn,acc]}$ [t_1 saw Mary]
c. *$Mary_2$ $T_{[+tense],[nom]}$ $Tr_{[+trn,acc]}$ [John saw t_2]
d. Covert LF movement:
$John_1$ $T_{[+tense],[nom]}$ $Mary_2$ $Tr_{[+trn,acc]}$ [t_1 saw t_2]

Transitive Clauses, Ergative Pattern

(28) Inuktitut (Inuit, SOV):

Jaani$_1$-up	[t_1	tuktu-Ø	malik-p-a-a]	$\mathrm{Tr}_{[+\mathit{trn},\mathit{erg}]}$
John-ERG		Karibou-NOM	follow-Ind-Tr-3sE.3sN	

'John followed the Karibou.'

(29) Mam (Maya, VSO):

ma	Ø-jaw	t-tx'eeʔma-n$_1$	Cheep$_2$	$\mathrm{Tr}_{[+\mathit{trn},\mathit{erg}]}$	[t_2	t_1	tzeeʔ	]
REC	3sN-DIR	3sE-cut-DS	José				tree	

'José cut the tree.'

Intransitive Clauses

Prediction:

- There is movement of the sole NP argument (in need of case checking) to SpecT in both language types. The reason for this uniform behaviour is that Tr does not have a case feature in this context.

Intransitive Clauses

Prediction:

- There is movement of the sole NP argument (in need of case checking) to SpecT in both language types. The reason for this uniform behaviour is that Tr does not have a case feature in this context.
- In accusative systems, this movement operation takes place overtly. (Reason: the case feature of T is strong.)

Intransitive Clauses

Prediction:

- There is movement of the sole NP argument (in need of case checking) to SpecT in both language types. The reason for this uniform behaviour is that Tr does not have a case feature in this context.
- In accusative systems, this movement operation takes place overtly. (Reason: the case feature of T is strong.)
- In ergative systems, this movement operation takes place covertly (at LF). (Reason: The case feature of T is weak.)
  (Note: Murasugi acknowledges that there might be a problem lurking here; see her footnote 21, p.40.)

Consequences: Chomsky/Bobaljik vs. Murasugi

Consequences: Chomsky/Bobaljik vs. Murasugi

Consequence:

Chomsky (1993), Bobaljik (1993):

1. Transitive constructions are identical in ergative and accusative argument encoding systems.
2. Intransitive constructions are different in ergative and accusative argument encoding systems.

Murasugi (1992):

1. Transitive constructions are different in ergative and accusative argument encoding systems.
2. Intransitive constructions are identical in ergative and accusative argument encoding systems.

Intransitive Clauses: Unergative Verbs

Assumption: NP_{ext} is merged in SpecV.

Intransitive Clauses: Unergative Verbs

Assumption: NP_{ext} is merged in SpecV.

(30) English:

a. $T_{[+tense,nom]}$ [John sang]
b. $John_1$ $T_{[+tense,nom]}$ [t_1 sang]

Intransitive Clauses: Unergative Verbs

Assumption: NP_{ext} is merged in SpecV.

(30) English:

a. $T_{[+tense,nom]}$ [John sang]
b. $John_1$ $T_{[+tense,nom]}$ [t_1 sang]

(31) Inuktitut:

[Jaani pisuk-p-u-q] $T_{[+tense,nom]}$
John-NOM go-IND-INTR-3sN

'John went.'

Intransitive Clauses: Unergative Verbs

Assumption: NP$_{ext}$ is merged in SpecV.

(30) English:

a. T$_{[+tense,nom]}$ [John sang]

b. John$_1$ T$_{[+tense,nom]}$ [t$_1$ sang]

(31) Inuktitut:

[Jaani	pisuk-p-u-q]	T$_{[+tense,nom]}$
John-NOM	go-IND-INTR-3sN	

'John went.'

(32) Mam:

ma	Ø-beet$_1$-T$_{[+tense,nom]}$	[xuʔj	t$_1$]
REC	3sN-go	woman	

'The woman went.'

Intransitive Clauses: Unaccusative Verbs

Assumption: NP_{int} is merged in CompV.

Intransitive Clauses: Unaccusative Verbs

Assumption: NP_{int} is merged in CompV.

(33) English:

a. $T_{[+tense,nom]}$ [arrived the man]
b. the man_1 $T_{[+tense,nom]}$ [t_1 arrived]

Intransitive Clauses: Unaccusative Verbs

Assumption: NP_{int} is merged in CompV.

(33) English:

a. $T_{[+tense,nom]}$ [arrived the man]
b. the man_1 $T_{[+tense,nom]}$ [t_1 arrived]

(34) Jacaltec (Maya):

a. x-∅-'ich-i munil
ASP-3sN-begin-INTR work
'Work began.'

b. ch-∅-aw-ich-e munil
ASP-3sN-2sE-begin-Tr work
'You begin the work.'

Active Patterns

Note:
As with Bobaljik (1993), ergative case for truly intransitive verbs is unexpected. The solution of this problem will have to be similar.

Potential Problems

1. The ergative/accusative parameter is closely tied to movement of NP arguments.

Potential Problems

1. The ergative/accusative parameter is closely tied to movement of NP arguments.
2. It is likely that NPs can check (or assign, or valuate) case without undergoing movement; see the concept of Agree in Chomsky (2001; 2005).

Potential Problems

1. The ergative/accusative parameter is closely tied to movement of NP arguments.
2. It is likely that NPs can check (or assign, or valuate) case without undergoing movement; see the concept of Agree in Chomsky (2001; 2005).
3. In contrast to Move, Agree does not dependent on strength of features; cf. uninterpretability, probe features.

Potential Problems

1. The ergative/accusative parameter is closely tied to movement of NP arguments.
2. It is likely that NPs can check (or assign, or valuate) case without undergoing movement; see the concept of Agree in Chomsky (2001; 2005).
3. In contrast to Move, Agree does not dependent on strength of features; cf. uninterpretability, probe features.
4. Is it possible to come up with a similar model of parametrization under an Agree-based (rather than Move-based) approach?

Potential Problems

1. The ergative/accusative parameter is closely tied to movement of NP arguments.
2. It is likely that NPs can check (or assign, or valuate) case without undergoing movement; see the concept of Agree in Chomsky (2001; 2005).
3. In contrast to Move, Agree does not dependent on strength of features; cf. uninterpretability, probe features.
4. Is it possible to come up with a similar model of parametrization under an Agree-based (rather than Move-based) approach?
5. Probably not: A violation of the Strict Cycle Condition will otherwise invariably occur in accusative languages.
6. Murasugi's analysis can avoid this general problem only by assuming that case is checked on two separate levels of representation (S-structure, LF). If all case checking takes place on a single level of representation, there is a problem (compare the concept of multiple spell-out of phases).

Bittner & Hale (1996): Background

- The cases of primary arguments are determined by two different syntactic heads K_1, K_2 (K_1 = I, K_2 = V). In ergative languages, K_1 determines ergative case, and K_2 does not determine a structural case. In accusative languages, K_1 does not determine a structural case, and K_2 determines accusative case. The remaining (or single) argument receives C(omp)-related default case ('K-Filter').

  1. ERG $\rightarrow$ K_1
  2. ACC $\rightarrow$ K_2
  3. NOM, ABS $\rightarrow$ Default

(Bittner & Hale (1996))

Bittner & Hale (1996): Basic Assumptions

Nominal arguments can be KPs ('Case phrases'), or DPs, or even bare NPs:

(35) $[_{\text{KP}}$ K $[_{\text{DP}}$ D $[_{\text{NP}}$ N ...]]]

Bittner & Hale (1996): Basic Assumptions

Nominal arguments can be KPs ('Case phrases'), or DPs, or even bare NPs:

(35) $[_{\mathrm{KP}}$ K $[_{\mathrm{DP}}$ D $[_{\mathrm{NP}}$ N ...]]]

Clause Structure:
The external argument is merged by adjunction to VP; this produces a small clause. (Order is irrelevant here.)

(36) $[_{\mathrm{CP}}$ C $[_{\mathrm{IP}}$ I $[_{\mathrm{VP}}$ $\{\mathrm{KP/DP}\}_{ext}$ $[_{\mathrm{VP}}$ V $\{\mathrm{KP/DP}\}_{int}$]]]]

Bittner & Hale (1996): Basic Assumptions

Nominal arguments can be KPs ('Case phrases'), or DPs, or even bare NPs:

(35) $[_{\mathrm{KP}}$ K $[_{\mathrm{DP}}$ D $[_{\mathrm{NP}}$ N ...]]]

Clause Structure:
The external argument is merged by adjunction to VP; this produces a small clause. (Order is irrelevant here.)

(36) $[_{\mathrm{CP}}$ C $[_{\mathrm{IP}}$ I $[_{\mathrm{VP}}$ $\{$KP/DP$\}_{ext}$ $[_{\mathrm{VP}}$ V $\{$KP/DP$\}_{int}$]]]]

- In ergative systems, I Case-binds KP_{ext}: ERG.
- In accusative systems, V Case-binds KP_{int}: ACC.
- The remaining argument in a transitive context is a DP (rather than KP), which gets default Case from C.

Definitions 1

(37) K Filter (NOM):
An argument chain headed by a K-less nominal (DP or NP) contains a position that is c-commanded and governed by K or C, and does not contain any Case-bound position.

(38) Oblique Case Realizations (DAT, INS, ABL; for Inuit):
If α Case-binds an overt empty-headed KP β and does not meet the conditions of (39-ab), then the empty K of β is realized as

a. DAT, if α is V and is not c-commanded by β.
b. INS, if α is V and is c-commanded by β.
c. ABL, if α is N and is not c-commanded by β.

(39) Direct Case Realizations (ERG, ACC):
If α Case-binds an overt empty-headed KP β, then the empty K of β is realized as

a. ERG, if α is I;
b. ACC, if α is V and has an adjoined D.

Definitions 2

(40) Case-Binding:
Let α be a head that delimits a clause, and let β be an argument. Then α Case-binds β, and β's head, iff

a. α locally c-commands β.
b. α governs a Case competitor for β.

(41) Delimiting heads:
A small clause is delimited by its lexical head, from below, and by any governing functional head, from above.

(42) Local C-Command:
Let α be a head that delimits a small clause, and let β be an argument. Then α locally c-commands β, iff:

a. α c-commands β, and
b. no other argument, or head that delimits a small clause, both c-commands β and is c-commanded by α.

(43) Case Competitor:
γ is a Case competitor for an argument β, iff γ is a K-less nominal that is (in a chain with) a coargument of β, or a pseudo coargument.

Definitions 3

(44) Coargument:
Let β and γ be arguments. Then γ is a coargument of β, iff (a) and (b) hold:

a. Locality: Some head that governs or A-projects γ also governs or A-projects β.

b. Independence: γ excludes β and is not in a chain with β.

(45) Government:
α governs β, iff:

a. α m-commands β.

b. There is no barrier between α and β.

(46) M-Command:
α m-commands β, iff α does not include β, and every maximal projection that includes α also includes β.

(47) C-Command:
α c-commands β, iff α excludes β, every projection that includes α also includes β, and at most one projection segment dominates α but not β.

(48) Barrier:
A barrier between α and β is an XP, γ, with the X^0 head, γ^0, such that

a. γ excludes α, includes β, and is not an extended projection of β;

b. γ^0 c-commands β, and neither α nor any adjunct of α binds γ^0.

Deriving an Ergative/Absolutive Pattern in Transitive Contexts 1

(49) $[_{\text{CP}}$ C $[_{\text{IP}}$ I $[_{\text{VP}}$ Arg$_{ext}$ $[_{\text{VP}}$ V Arg$_{int}$]]]]

What we want to derive:

1. Arg$_{ext}$ is a KP Case-bound by I (then it is assigned ergative).
2. Arg$_{int}$ is a DP that obeys the K Filter (then it has no case: nominative/absolutive).

Deriving an Ergative/Absolutive Pattern in Transitive Contexts 2

Case-Binding of KP_{ext} by I and K Filter for DP_{int}:

1. If I is to Case-bind Arg_{ext} as a KP, then I must be a head that delimits a clause. It is such a head (it delimits the VP small clause from above because it is a governing functional head).

Deriving an Ergative/Absolutive Pattern in Transitive Contexts 2

Case-Binding of KP_{ext} by I and K Filter for DP_{int}:

1. If I is to Case-bind Arg_{ext} as a KP, then I must be a head that delimits a clause. It is such a head (it delimits the VP small clause from above because it is a governing functional head).
2. If I is to Case-bind Arg_{ext} as a KP, then I must locally c-command Arg_{ext}. It does: There is no other argument (or small-clause-delimiting head) that intervenes between I and Arg_{ext}. (In particular, Arg_{int} does not intervene: it is lower in the structure.)

Deriving an Ergative/Absolutive Pattern in Transitive Contexts 3

3 If I is to Case-bind Arg_{ext} as a KP, then I must govern a Case competitor for Art_{ext}. I does not govern such a Case competitor for Arg_{ext} in the structure in (49). The reason is that Arg_{int} is protected by government by I through a barrier, viz., VP. However, there are two ways to make I govern Arg_{int} after all: First, Arg_{int} can move to SpecI (movement may cross a barrier as defined here). Second, Arg_{int} may be governed by I because head movement of V to I opens up the barrier and makes government of I into the VP possible (V is then an adjunct of I that binds its trace γ^0). Thus, for I to govern Arg_{int} as a Case competitor for Arg_{ext}, either V or Arg_{int} has to move out of the VP. Furthermore, if I is to Case-bind Arg_{ext}, Arg_{int} must be a K-less nominal: a DP. Finally, Arg_{int} must be a Case competitor for Arg_{ext}. It is because they are co-arguments. (They are co-arguments because they are A-projected by the same head – V –, and because they are not in a dominance or chain relation.)

Deriving an Ergative/Absolutive Pattern in Transitive Contexts 3

3. If I is to Case-bind Arg_{ext} as a KP, then I must govern a Case competitor for Art_{ext}. I does not govern such a Case competitor for Arg_{ext} in the structure in (49). The reason is that Arg_{int} is protected by government by I through a barrier, viz., VP. However, there are two ways to make I govern Arg_{int} after all: First, Arg_{int} can move to SpecI (movement may cross a barrier as defined here). Second, Arg_{int} may be governed by I because head movement of V to I opens up the barrier and makes government of I into the VP possible (V is then an adjunct of I that binds its trace γ^0). Thus, for I to govern Arg_{int} as a Case competitor for Arg_{ext}, either V or Arg_{int} has to move out of the VP. Furthermore, if I is to Case-bind Arg_{ext}, Arg_{int} must be a K-less nominal: a DP. Finally, Arg_{int} must be a Case competitor for Arg_{ext}. It is because they are co-arguments. (They are co-arguments because they are A-projected by the same head – V –, and because they are not in a dominance or chain relation.)
4. As a result, we derive that I Case-binds KP_{ext} if the internal argument is a K-less nominal DP_{int} that either moves out of VP or shows up in a VP out of which the verb has moved to I.

Deriving an Ergative/Absolutive Pattern in Transitive Contexts 4

5. If Arg_{int} is a DP, it obeys the K Filter. This means that it must be governed by C, and is not Case-bound itself. It cannot be Case-bound since it is not locally c-commanded by a clause-delimiting head; and we can assume that C governs Arg_{int} (IP is transparent, e.g., because of I-to-C movement).

Deriving an Ergative/Absolutive Pattern in Transitive Contexts 4

5. If Arg_{int} is a DP, it obeys the K Filter. This means that it must be governed by C, and is not Case-bound itself. It cannot be Case-bound since it is not locally c-commanded by a clause-delimiting head; and we can assume that C governs Arg_{int} (IP is transparent, e.g., because of I-to-C movement).
6. This then means that given a structure like (49), the external argument is a KP that is assigned ergative, and the internal argument is a DP that has default case (nominative/absolutive).

Deriving an Ergative/Absolutive Pattern in Transitive Contexts 4

5. If Arg_{int} is a DP, it obeys the K Filter. This means that it must be governed by C, and is not Case-bound itself. It cannot be Case-bound since it is not locally c-commanded by a clause-delimiting head; and we can assume that C governs Arg_{int} (IP is transparent, e.g., because of I-to-C movement).
6. This then means that given a structure like (49), the external argument is a KP that is assigned ergative, and the internal argument is a DP that has default case (nominative/absolutive).

(50) Two ways to get an ergative encoding pattern:

a. Movement of NP_{int} to SpecI:
$[_{CP}\ [_{IP}\ DP_2\ [_{I'}\ [_{VP}\ KP_1\ [_{VP}\ V\ t_2\ ]]\ I\ ]]\ C\ ]$

Deriving an Ergative/Absolutive Pattern in Transitive Contexts 4

5. If Arg_{int} is a DP, it obeys the K Filter. This means that it must be governed by C, and is not Case-bound itself. It cannot be Case-bound since it is not locally c-commanded by a clause-delimiting head; and we can assume that C governs Arg_{int} (IP is transparent, e.g., because of I-to-C movement).
6. This then means that given a structure like (49), the external argument is a KP that is assigned ergative, and the internal argument is a DP that has default case (nominative/absolutive).

(50) Two ways to get an ergative encoding pattern:

a. Movement of NP_{int} to SpecI:
$[_{\text{CP}}\ [_{\text{IP}}\ \text{DP}_2\ [_{\text{I}'}\ [_{\text{VP}}\ \text{KP}_1\ [_{\text{VP}}\ \text{V}\ t_2\ ]]\ \text{I}\ ]]\ \text{C}\ ]$

b. Movement of V to I:
$[_{\text{CP}}\ [_{\text{IP}}\ –\ [_{\text{I}'}\ [_{\text{VP}}\ \text{KP}_1\ [_{\text{VP}}\ t_{\text{V}}\ \text{DP}_2\ ]]\ \text{V-I}\ ]]\ \text{C}\ ]$

Conclusion:
Ergative case shows up on an external argument, but only in the presence of a lower coargument.

Syntactic vs. Morphological Ergativity

The two options in (50) cover syntactic vs. morphological ergativity:

- Syntactic Ergativity:
  If DP_{int} moves to SpecI, it becomes the highest argument. Syntactic operations referring to the notion of highest argument ('subject') will now treat Arg_{int}-V_t in the same way as Arg_{ext}-V_i and Arg_{int}-V_i (and will treat Arg_{ext}-V_t differently).
- Morphological Ergativity:
  If DP_{int} stays in situ, within VP (and V moves to I), it maintains 'object properties'. Syntactic operations referring to the notion of highest argument ('subject') will now treat Arg_{ext}-V_t in the same way as Arg_{ext}-V_i and Arg_{int}-V_i (and will treat Arg_{int}-V_t differently).

Assumption:
Dyirbal, Inuit: syntactic ergativity (but recall Bobaljik (1993) on reflexives in Inuit; see (13))
Samoan, Warlpiri: morphological ergativity

Active Patterns

There is no obvious way to account for an ergative case on an external argument of a transitive verb, as in Basque, Hindi, Guaraní, and Georgian. Strategy (well-known by now): Unergative verbs are hidden transitive verbs. In Basque, the evidence for this may not be poor: "Unergatives regularly take the form of light verb constructions," as in hitz egin ('word do', 'speak'). However, things are not so clear in Georgian, where the verbs that are involved do not look like light verb constructions ('Funktionsverbgefüge'); also see Nash (1996).

(51) Active patterns in Georgian (past-tense, perfective aspect only):

a. Vano-m gamozarda dzma
Vano-ERG$_1$ 3.SG$_2$.raised.3.SG$_1$ brother-NOM$_2$
'Vano raised his brother.' (transitive)

b. Bavšv-ma itira
child-ERG$_1$ cried.3.SG$_1$
'The child cried.' (unergative intransitive)

c. Rezo gamoizarda
Rezo-NOM$_2$ grew.3.SG$_2$
'Rezo grew up.' (unaccusative intransitive)

Why Ergative Patterns are Simpler

We have seen that I Case-binds Arg$_{ext}$ in (52). Can Arg$_{int}$ also be Case-bound?

(52) $[_{\text{CP}}$ C $[_{\text{IP}}$ I $[_{\text{VP}}$ Arg$_{ext}$ $[_{\text{VP}}$ V Arg$_{int}$]]]]

Two candidates: I and V.

1. I cannot Case-bind Arg$_{int}$ in (52) because I does not locally c-command Arg$_{int}$ (Arg$_{ext}$ intervenes).
2. V cannot Case-bind Arg$_{int}$ either because V does not govern a Case competitor for Arg$_{int}$ (Arg$_{ext}$ is not governed by V because V does not m-command it: VP includes V but not Arg$_{ext}$ in the VP-Adj position).

The latter consequence follows in an even simpler way (without invoking the inclusion/exclusion distinction) if external arguments are base-generated in the specifier of vP (rather than in a VP-adjoined position).

Consequence: Accusative patterns are more marked than ergative patterns; something extra needs to be said about the former!

Towards Accusative Encoding Patterns

Recall the notion of Case competitor in (53); pseudo coargument still needs to be defined:

(53) Case Competitor:
γ is a Case competitor for an argument β, iff γ is a K-less nominal that is (in a chain with) a coargument of β, or a pseudo coargument.

(54) Pseudo Coargument:
Let β be an argument; δ, a head that delimits a small clause; and γ; a head adjoined to δ. Then γ is a pseudo coargument of β, iff (a) and (b) hold:

a. Locality: δ governs β, and γ c-commands β.
b. Independence: γ is not in a chain with the X^0 head of β, and β is not in a chain with the subject of the small clause delimited by δ.

Towards Accusative Encoding Patterns

Recall the notion of Case competitor in (53); pseudo coargument still needs to be defined:

(53) Case Competitor:
γ is a Case competitor for an argument β, iff γ is a K-less nominal that is (in a chain with) a coargument of β, or a pseudo coargument.

(54) Pseudo Coargument:
Let β be an argument; δ, a head that delimits a small clause; and γ; a head adjoined to δ. Then γ is a pseudo coargument of β, iff (a) and (b) hold:

a. Locality: δ governs β, and γ c-commands β.
b. Independence: γ is not in a chain with the X^0 head of β, and β is not in a chain with the subject of the small clause delimited by δ.

Consequences:

- Locality: γ can never be a pseudo coargument of Arg_{ext}.
- Independence: Pseudo coarguments only come into being if there is more than one argument in the clause.

Antipassive

(55) Antipassive Alternation in Chukchee (Paleosibirian; Comrie (1979)):

a. Yemronə-na qərir-ərkən-in ekək
Yemron-ERG$_1$ search-PRS-3.SG$_1$.3.SG$_2$ son-NOM$_2$
'Yemron is searching for his son.'

b. Yemron ine-lqərir-ərkən (akka-gtə)
Yemron-NOM$_1$ APASS-search-PRS.3SG$_1$ (son-DAT)
'Yemron is searching (for his son).'

Analysis:

1. The APASS morpheme is an N head adjoined to V; it is a pseudo coargument for Arg$_{int}$, which accordingly is a Case-bound KP.
2. KP$_{int}$ is Case-bound by V and gets DAT rather than ACC case for the simple reason that the pseudo coargument is an N rather than a D (see (38), (39)).
3. Arg$_{ext}$ cannot be Case-bound anymore in this configuration because there is no coargument (or close pseudo coargument) that might act as a Case competitor (i.e., be K-less – the Case-bound Arg$_{int}$ certainly is not).
4. Arg$_{ext}$ therefore must be DP and gets default case from C.

Accusative Encoding Patterns

Assumptions:

1. Accusative patterns can be traced back to a V-adjoined pseudo-argument, as in antipassives. The only difference: The adjoined item is D rather than N; hence, the Case realized for a KP Case-bound by V-D is ACC rather than, say, DAT.
2. DP_{ext} must be governed by C, which can take place either via raising to SpecI, or via head movement of I to C (just as in ergative encoding systems). Result: English-type vs. Japanese/German-type accusative languages.

(56) Two accusative patterns:

a. Movement of DP_{ext} to SpecI:
$[_{CP}$ C $[_{IP}$ DP_1 $[_{I'}$ $[_{VP}$ t_1 $[_{VP}$ V KP_2]] I]]]

b. Movement of I to C:
$[_{CP}$ C-I $[_{IP}$ – $[_{I'}$ $[_{VP}$ DP_1 $[_{VP}$ V KP_2]] t_I]]]

Three-Way Systems 1

Like optimality-theoretic analyses, but in contrast to the other two minimalist types of approach, the present analysis can in principle capture a co-occurrence of ERG and ACC. Languages like Antekerrepenhe (Arandic; Central Australia) and Nez Perce (Penutian; Oregon) seem to instantiate this rare pattern.

(57) Antekerrepenhe:

a. Arengke-le aye-nhe ke-ke
   dog-ERG me-ACC bite-PST
   'The dog bit me.'

b. Apwerte-le athe arengke-nhe we-ke
   stones-INS I-ERG dog-ACC pelt-PST
   'I pelted the dog with stones.'

c. Arengke nterre-ke
   dog-NOM run-PST
   'The dog ran.'

Three-Way Systems 2

Analysis:

1. There must be an additional pseudo coargument for Arg_{ext} somewhere in the structure, so that the latter can be Case-bound by I even though Arg_{int} is Case-bound by V.
2. There is an additional NP shell on top of the Arg_{int} KP. The N head of the NP shell undergoes incorporation to D in V. Now Arg_{ext} and Arg_{int} both have Case competitors, and two structural cases can be realized.

(58) VP Structure in Three-Way Systems:
$[_{\text{VP}}\ \text{KP}_{ext}\ [_{\text{VP}}\ [_{\text{V}'}\ [_{\text{NP}}\ [_{\text{N}'}\ [_{\text{KP}_{int}}\ \text{K}\ [_{\text{DP}}\ \text{D}\ [_{\text{NP}}\ \text{N}\ ]]]\ t_{\text{N}}\ ]]\ [_{\text{V}}\ [_{\text{D}}\ \text{N D}\ ]\ \text{V}\ ]]]]$

Three-Way Systems 2

Analysis:

1. There must be an additional pseudo coargument for Arg_{ext} somewhere in the structure, so that the latter can be Case-bound by I even though Arg_{int} is Case-bound by V.
2. There is an additional NP shell on top of the Arg_{int} KP. The N head of the NP shell undergoes incorporation to D in V. Now Arg_{ext} and Arg_{int} both have Case competitors, and two structural cases can be realized.

(58) VP Structure in Three-Way Systems:
$[_{VP}\ KP_{ext}\ [_{VP}\ [_{V'}\ [_{NP}\ [_{N'}\ [_{KP_{int}}\ K\ [_{DP}\ D\ [_{NP}\ N\ ]]]\ t_N\ ]]\ [_{V}\ [_{D}\ N\ D\ ]\ V\ ]]]]$

Claim:
This is indicative of a more general feature of Bittner & Hale's analysis: The system is quite flexible (more so than the analyses developed by Bobaljik and Murasugi, e.g.), but this is mainly due to the fact that highly articulate structures, and subtle structural differences (e.g., N vs. D) between languages, are postulated.

Literatur

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

Bobaljik, Jonathan (1993): Ergativity and Ergative Unergatives. In: C. Phillips, ed., *Papers on Case and Agreement II*. Vol. 19 of *MIT Working Papers in Linguistics*, MITWPL, MIT: Cambridge, Mass., pp. 45–88.

Chomsky, Noam (1973): Conditions on Transformations. In: S. Anderson & P. Kiparsky, eds., *A Festschrift for Morris Halle*. Academic Press, New York, pp. 232–286.

Chomsky, Noam (1981): *Lectures on Government and Binding*. Foris, Dordrecht.

Chomsky, Noam (1993): A Minimalist Program for Syntactic Theory. In: K. Hale & S. J. Keyser, eds., *The View from Building 20*. MIT Press, Cambridge, Mass., pp. 1–52.

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

Chomsky, Noam (2001): Derivation by Phase. In: M. Kenstowicz, ed., *Ken Hale. A Life in Language*. MIT Press, Cambridge, Mass., pp. 1–52.

Chomsky, Noam (2005): On Phases. Ms., MIT, Cambridge, Mass.

Hale, Ken & Samuel Jay Keyser (2002): *Prolegomenon to a Theory of Argument Structure*. MIT Press, Cambridge, Mass.

Jelinek, Eloise (1993): Ergative 'Splits' and Argument Type. In: J. Bobaljik & C. Phillips, eds., *Papers on Case and Agreement I*. Vol. 18 of *MIT Working Papers in Linguistics*, MITWPL, MIT: Cambridge, Mass., pp. 15–42.

Laka, Itziar (1993): Unergatives that Assign Ergative, Unaccusatives that Assign Accusative. In: J. Bobaljik & C. Phillips, eds., *Papers on Case and Agreement I*. Vol. 18 of *MIT Working Papers in Linguistics*, MITWPL, MIT: Cambridge, Mass., pp. 149–172.

Levin, Juliette & Diane Massam (1985): Surface Ergativity: Case/Theta Relations Reexamined. In: S. Berman, ed., *Proceedings of NELS 15*. BLS, Amherst, Mass.

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

Nash, Léa (1996): The Internal Ergative Subject Hypothesis. In: K. Kusumoto, ed., *Proceedings of NELS 26*. GLSA, Amherst, Mass, pp. 195–209.