

Bare Singulars and Pseudo-Incorporation in Western Armenian

Alexandros Kalomoiros

University of Pennsylvania

SALT 31, 7th May 2021

- ▶ **Main Idea:** Bare singulars in Western Armenian (WA) denote unambiguously properties of (sub)kinds.

- ▶ **Main Idea:** Bare singulars in Western Armenian (WA) denote unambiguously properties of (sub)kinds.
- **Previous literature (Dayal 2004):** The singular (at least in English) is ambiguous between denoting a property of kinds and a property of objects.

- **Our argument:**
 - In WA, certain Pseudo-Incorporating (PI) nominal elements can only be modified by kind-level adjectives.

- **Our argument:**
 - In WA, certain Pseudo-Incorporating (PI) nominal elements can only be modified by kind-level adjectives.
 - This cannot be captured by restricting what can undergo PI to kind-level denotations (cf. Saĝ 2019), as object-level denoting nouns can also PI.

- **Our argument:**

- In WA, certain Pseudo-Incorporating (PI) nominal elements can only be modified by kind-level adjectives.
- This cannot be captured by restricting what can undergo PI to kind-level denotations (cf. Saĝ 2019), as object-level denoting nouns can also PI.
- The modification restriction follows naturally if we assume that the bare singular always denotes a (property) of kinds in WA.

- WA allows 'Num Noun' constructions of the form 'Num N_{sg}' (Bale et al. 2011, Bale & Khanjian 2014):

(1) jerek (had) afagerd
three (CLF) student

(2) jerek (had) afagerd-ner
three (CLF) student-**PL**

Covert plurals

- WA allows 'Num Noun' constructions of the form 'Num N_{sg}' (Bale et al. 2011, Bale & Khanjian 2014):

(1) jerek (had) afagerd (2) jerek (had) afagerd-ner
three (CLF) student three (CLF) student-**PL**

- 'Num Noun' constructions like (1) (covert plurals) can trigger either singular, (3), or plural, (4), verbal agreement (Sigler 1997):

(3) jerek afagerd inga-v (4) jerek afagerd inga-n
three student fall-PST.**3SG** three student fall-PST-**3PL**
'Three students fell' 'Three students fell'

- **Focus:** Covert plurals that show singular agreement (non-agreeing)

Bare Sg Pseudo-Incorporate

- We argue that (non-agreeing) covert plurals undergo PI. To do this we show that they pattern like other PI-ed elements in the language, namely bare singulars.
- So, we first show that bare sg undergo PI:

Bare Sg Pseudo-Incorporate

- We argue that (non-agreeing) covert plurals undergo PI. To do this we show that they pattern like other PI-ed elements in the language, namely bare singulars.
- So, we first show that bare sg undergo PI:
- Bare sg are number-neutral:

(5) John- \emptyset manug tasdiajarage-ts
John-DEF child educate-PST.3SG
'John educate child(ren)'

Bare Sg Pseudo-Incorporate

- We argue that (non-agreeing) covert plurals undergo PI. To do this we show that they pattern like other PI-ed elements in the language, namely bare singulars.
- So, we first show that bare sg undergo PI:
- Bare sg are number-neutral:

(5) John-ə manug tasdiajarage-ts
John-DEF child educate-PST.3SG
'John educate child(ren)'

- Bare sg take low scope:

(6) John-ə manug tʃə tasdiajarage-ts
John-DEF child NEG educate-PST.3SG
'John did not educate any children' ($\neg > \exists$, $*\exists > \neg$)

- Animate full DPs in WA are marked dative in object position (DOM):

(7) John-ə manug-i-n tasdiajarage-ts
John-DEF child-DAT-DEF educate-PST.3SG
'John educated the (unique) child'

(8) ??John-ə manug-ə tasdiajarage-ts
John-DEF child-DEF educate-PST.3SG
'John educated the child'

- Bare singulars resist the dative, even if animate:

(9) John- \emptyset manug tasdiajarage-ts
John-DEF child educate-PST.3SG
'John educate child(ren)'

(10) ?*John- \emptyset manug-i tasdiajarage-ts
John-DEF child-DAT educate-PST.3SG
'John loves a child'

- Bare singulars resist the dative, even if animate:

(9) John- \emptyset manug tasdiajarage-ts
John-DEF child educate-PST.3SG
'John educate child(ren)'

(10) ?*John- \emptyset manug-i tasdiajarage-ts
John-DEF child-DAT educate-PST.3SG
'John loves a child'

- So bare sg do not behave as full arguments. We can understand these patterns if we take bare sg to PI (Massam 2001).

Covert plurals PI

- Non-agreeing Covert plurals behave just like bare sg with respect to the PI diagnostics:
- (11) shows low scope:

(11) jerek aʃagerd tʃ-inga-v
three student NEG-fall-PST.3SG
'Three students did not fall' ($\neg > \exists$, $*\exists > \neg$)

Covert plurals PI

- Non-agreeing Covert plurals behave just like bare sg with respect to the PI diagnostics:

- (11) shows low scope:

(11) jerek aʃagerd tʃ-inga-v
three student NEG-fall-PST.3SG
'Three students did not fall' ($\neg > \exists$, $*\exists > \neg$)

- (12) shows inability to be marked Dative:

(12) John- \emptyset harujr had zinvor-(*i) mert-uts
John-DEF 100 CLF soldier-(*DAT) killed.PST.3SG
'John killed 100 soldiers'

Covert plurals PI

- Non-agreeing Covert plurals behave just like bare sg with respect to the PI diagnostics:

- (11) shows low scope:

(11) jerek aʃagerd tʃ-inga-v
three student NEG-fall-PST.3SG
'Three students did not fall' ($\neg > \exists$, $*\exists > \neg$)

- (12) shows inability to be marked Dative:

(12) John- \emptyset harujr had zinvor-(*i) mert-uts
John-DEF 100 CLF soldier-(*DAT) killed.PST.3SG
'John killed 100 soldiers'

- **Conclusion:** Non-agreeing covert plurals undergo PI.
- **NB:** Non-agreeing covert plurals denote object-level properties.

Modification 1

- Bare Sg in WA allow modification only by kind-level adjectives (this observation is also made in Sağ 2019).

(13) jereg, John-ə fantasi/ # hin kirk garta-ts
yesterday, john-DEF fantasy/ # old book read-PST.3SG
'Yesterday, John read fantasy/ old book(s)'

- Turkish shows the same pattern (Sağ 2019). Bare Sg in Turkish also PI.

Modification 1

- Bare Sg in WA allow modification only by kind-level adjectives (this observation is also made in Sağ 2019).

(13) jereg, John-ə fantasi/ # hin kirk garta-ts
yesterday, john-DEF fantasy/ # old book read-PST.3SG
'Yesterday, John read fantasy/ old book(s)'

- Turkish shows the same pattern (Sağ 2019). Bare Sg in Turkish also PI.
- **Analysis of Sağ 2019:**
 - The bare sg is ambiguous between object-level and kind-level properties.
 - The PI mechanism is restricted to apply to kind-level properties only ⇒ bare sg will never allow object-level mod in PI environments.

Modification 1

- Bare Sg in WA allow modification only by kind-level adjectives (this observation is also made in Sağ 2019).

(13) jereg, John-ə fantasi/ # hin kirk garta-ts
yesterday, john-DEF fantasy/ # old book read-PST.3SG
'Yesterday, John read fantasy/ old book(s)'

- Turkish shows the same pattern (Sağ 2019). Bare Sg in Turkish also PI.
- **Analysis of Sağ 2019:**
 - The bare sg is ambiguous between object-level and kind-level properties.
 - The PI mechanism is restricted to apply to kind-level properties only ⇒ bare sg will never allow object-level mod in PI environments.
- **However, WA allows PI of object-level properties.** Therefore, PI in WA cannot be restricted to just kinds.

Modification 2

- Covert plurals show the same restrictions with regards to modification as bare singulars.

(14) hink (had) jevropagan zinvor mertsə-ve-ts-av
5 CLF European soldier kill-PASS-AOR-PST.3SG
'Five European soldiers were killed'

(15) hink #(had) anoti zinvor mertsə-ve-ts-av
5 CLF hungry soldier kill-PASS-AOR-PST.3SG
'Five hungry soldiers were killed'

- (15) becomes fine either when the classifier is overt, or when the context supports 'hungry soldiers' as a type (i.e. we have a roster of hungry soldiers).

Modification 2

- Covert plurals show the same restrictions with regards to modification as bare singulars.

(14) hink (had) jevropagan zinvor mertsə-ve-ts-av
5 CLF European soldier kill-PASS-AOR-PST.3SG
'Five European soldiers were killed'

(15) hink #(had) anoti zinvor mertsə-ve-ts-av
5 CLF hungry soldier kill-PASS-AOR-PST.3SG
'Five hungry soldiers were killed'

- (15) becomes fine either when the classifier is overt, or when the context supports 'hungry soldiers' as a type (i.e. we have a roster of hungry soldiers).
- **Claim:**
 - We **cannot** account for these patterns **by restricting PI** to just kinds.
 - But we **can** account for them **by restricting the bare sg** to just kinds (leaving PI unrestricted).

- Bare sg unambiguously denote properties of kinds.

Analysis in a nutshell

- Bare sg unambiguously denote properties of kinds.
- When bare sg/covert plural without a CLF PI \rightsquigarrow the kind needs to be instantiated. This is costly for non-well-established kinds.

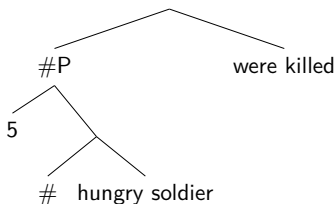
- Bare sg unambiguously denote properties of kinds.
- When bare sg/covert plural without a CLF PI \rightsquigarrow the kind needs to be instantiated. This is costly for non-well-established kinds.
- Covert plurals with a CLF \rightsquigarrow the classifier is responsible for instantiating the kind directly. So no cost incurred during PI.
 - **Come to the breakout room for more details!** –

Thank you!

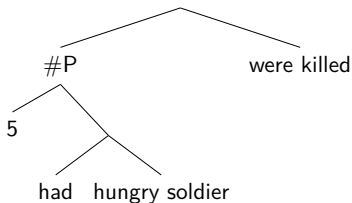
Some syntax

- We will assume the following syntax for covert plurals (evidence for this also comes from agreement, see Kalomoiros (forthcoming)):

(16)



(17)



- Recall the patterns we are trying to capture:
 - **Generalization 1:** Bare singulars, and covert plurals + kind level adjectives \rightsquigarrow felicitous in an out-of-the-blue context.
 - **Generalization 2:**
 - Bare singulars, and covert plurals + object level adjective \rightsquigarrow felicitous only if the context establishes the relevant subkind.
 - Covert plurals with an overt classifier are fine in an out-of-the-blue context regardless of adjective type.

- **From Dayal 2004:** Bare sg denote properties of singular kinds. These are taxonomic individuals (technically groups in the sense of Landman 1989).

Analysis: Preliminaries

- **From Dayal 2004:** Bare sg denote properties of singular kinds. These are taxonomic individuals (technically groups in the sense of Landman 1989).
- **Diverging from Dayal:** At least in WA, bare sg only denote properties of kinds.

- **From Dayal 2004:** Bare sg denote properties of singular kinds. These are taxonomic individuals (technically groups in the sense of Landman 1989).
- **Diverging from Dayal:** At least in WA, bare sg only denote properties of kinds.
- PI is broken down into three components: **(1)** Restriction (Chung & Ladusaw 2004), **(2)** Sort Adjustment (if necessary), **(3)** Existential Closure.

Restriction: If α is branching node, and $\{\beta, \gamma\}$ the set of its daughters, where $[[\beta]] = \lambda x.P(x)$ and $[[\gamma]] = \lambda x_1 \dots \lambda x_n.Q(x_1, \dots, x_n)$, then $[[\alpha]] = \lambda x_2 \dots \lambda x_n \lambda x_1. Q(x_1, \dots, x_n) \wedge P(x_1)$.

Analysis: Some tools

- We will make use of the following tools to talk about the relationship between kinds and their instantiations:

Analysis: Some tools

- We will make use of the following tools to talk about the relationship between kinds and their instantiations:
 - (18) *belong – to*(x, y_k) is true iff x is an object-level entity and y_k is a kind-level entity that has x as its part/instantiation.

Analysis: Some tools

- We will make use of the following tools to talk about the relationship between kinds and their instantiations:

(18) *belong – to*(x, y_k) is true iff x is an object-level entity and y_k is a kind-level entity that has x as its part/instantiation.

(19) **DSKP**: Let F be a function of the form $\lambda x_1. \dots \lambda x_n. \alpha$, defined on object-level entities and where α is some formula. Let k be a singular kind. Then, restricting x_i ($1 \leq i \leq n$) to k , i.e. $\lambda x_1. \dots \lambda x_i. \dots \lambda x_n. \alpha \wedge x_i = k$, is equivalent to restricting x_i to the instantiations of k , i.e.
 $\lambda x_1. \dots \lambda x_i. \dots \lambda x_n. \alpha \wedge \textit{belong – to}(x_i, k)$

Analysis: Some tools

- We will make use of the following tools to talk about the relationship between kinds and their instantiations:

(18) *belong – to*(x, y_k) is true iff x is an object-level entity and y_k is a kind-level entity that has x as its part/instantiation.

(19) **DSKP:** Let F be a function of the form $\lambda x_1. \dots \lambda x_n. \alpha$, defined on object-level entities and where α is some formula. Let k be a singular kind. Then, restricting x_i ($1 \leq i \leq n$) to k , i.e. $\lambda x_1. \dots \lambda x_i. \dots \lambda x_n. \alpha \wedge x_i = k$, is equivalent to restricting x_i to the instantiations of k , i.e.
 $\lambda x_1. \dots \lambda x_i. \dots \lambda x_n. \alpha \wedge \textit{belong – to}(x_i, k)$

(20) **Instantiation Principle:** When DSKP is used to instantiate a non-well-established kind in a context that does not support that kind, it leads to decreased contextual acceptability.

Analysis: Some tools

- We will make use of the following tools to talk about the relationship between kinds and their instantiations:

(18) *belong – to*(x, y_k) is true iff x is an object-level entity and y_k is a kind-level entity that has x as its part/instantiation.

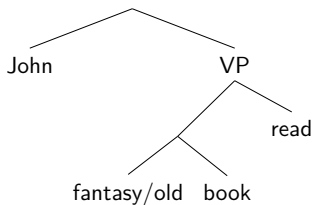
(19) **DSKP:** Let F be a function of the form $\lambda x_1 \dots \lambda x_n. \alpha$, defined on object-level entities and where α is some formula. Let k be a singular kind. Then, restricting x_i ($1 \leq i \leq n$) to k , i.e. $\lambda x_1 \dots \lambda x_i \dots \lambda x_n. \alpha \wedge x_i = k$, is equivalent to restricting x_i to the instantiations of k , i.e.
 $\lambda x_1 \dots \lambda x_i \dots \lambda x_n. \alpha \wedge \textit{belong – to}(x_i, k)$

(20) **Instantiation Principle:** When DSKP is used to instantiate a non-well-established kind in a context that does not support that kind, it leads to decreased contextual acceptability.

- The key is whether the *belong – to* relation that instantiates the kind is established via DSKP or not.

Bare Singulars

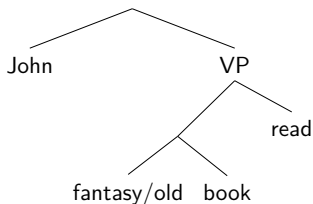
(21)



(22) $[[book]] = \lambda x.x = BOOK$

Bare Singulars

(21)

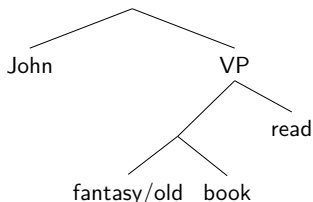


(22) $[[book]] = \lambda x.x = BOOK$

$$\begin{aligned} & [[read]](\lambda x.x = OLD\ BOOK) \xrightarrow{Restrict} \lambda y.\lambda x.read(y)(x) \wedge x = \\ & OLD\ BOOK \xrightarrow{DSKP} \lambda y.\lambda x.read(y)(x) \wedge belong - to(x, OLD\ BOOK) \\ & \xrightarrow{\exists-closure} \lambda y.\exists x[read(y)(x) \wedge belong - to(x, OLD\ BOOK)]. \end{aligned}$$

Bare Singulars

(21)



(22) $[[book]] = \lambda x.x = BOOK$

$[[read]](\lambda x.x = OLD\ BOOK) \xrightarrow{Restrict} \lambda y.\lambda x.read(y)(x) \wedge x =$

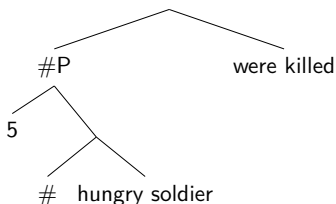
$OLD\ BOOK \xrightarrow{DSKP} \lambda y.\lambda x.read(y)(x) \wedge belong - to(x, OLD\ BOOK)$

$\xrightarrow{\exists-closure} \lambda y.\exists x[read(y)(x) \wedge belong - to(x, OLD\ BOOK)].$

- Because DSKP was used to instantiate a non-well-established kind that has no contextual support, a cost is incurred; thus 'old book' is infelicitous.

Covert plurals: covert classifier

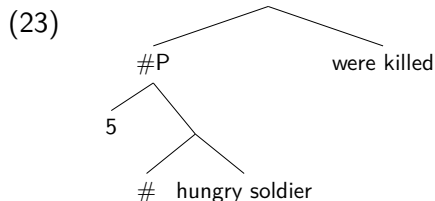
(23)



(24) $[[\#]] = \lambda P_{et}.\lambda n_d.\lambda x_e.P(x) \wedge |x| = n$

(25) $[[soldier]] = \lambda x.x = SOLDIER$

Covert plurals: covert classifier



(24) $[[\#]] = \lambda P_{et}.\lambda n_d.\lambda x_e.P(x) \wedge |x| = n$

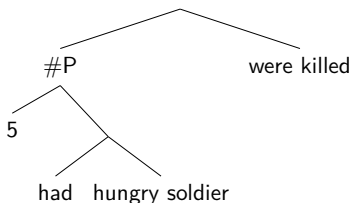
(25) $[[soldier]] = \lambda x.x = SOLDIER$

$$[[\#]](\lambda x.x = HUNGRY SOLDIER) \Rightarrow \lambda x_e.x = HS \wedge |x| = 5 \xrightarrow{DSKP}$$

$$\lambda x_e.belong - to(x, HS) \wedge |x| = 5 \xrightarrow{Restrict, \exists} \exists x[belong - to(x, HS) \wedge |x| = 5 \wedge were - killed(x)]$$

Covert plurals: overt classifier

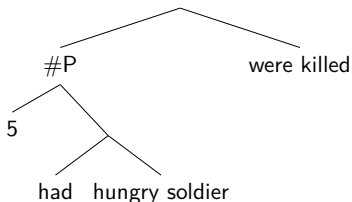
(26)



(27) $[[had]] = \lambda P_{et}.\lambda n_d.\lambda x_e.belong - to(x, \iota yP(y)) \wedge |x| = n$

Covert plurals: overt classifier

(26)



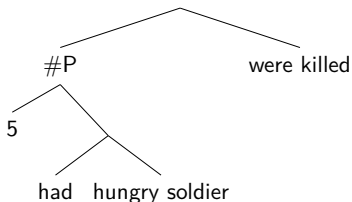
(27) $[[had]] = \lambda P_{et}.\lambda n_d.\lambda x_e.belong - to(x, \iota yP(y)) \wedge |x| = n$

$[[had]](\lambda x.x = HS) = \lambda n_d.\lambda x_e.belong - to(x, HS) \wedge |x| = n \Rightarrow$
 $\lambda x_e.belong - to(x, HS) \wedge |x| = 5 \xrightarrow{\text{Restrict, closure}}$

$\exists x[belong - to(x, HS) \wedge |x| = 5 \wedge were - killed(x)]$

Covert plurals: overt classifier

(26)



(27) $[[had]] = \lambda P_{et}.\lambda n_d.\lambda x_e.belong - to(x, \iota yP(y)) \wedge |x| = n$

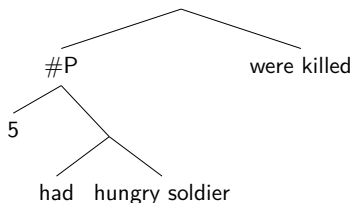
$[[had]](\lambda x.x = HS) = \lambda n_d.\lambda x_e.belong - to(x, HS) \wedge |x| = n \Rightarrow$
 $\lambda x_e.belong - to(x, HS) \wedge |x| = 5 \xrightarrow{\text{Restrict, closure}}$

$\exists x[belong - to(x, HS) \wedge |x| = 5 \wedge were - killed(x)]$

- DSKP is not used here, so no contextual cost is incurred.

Covert plurals: overt classifier

(26)



(27) $[[had]] = \lambda P_{et}.\lambda n_d.\lambda x_e.belong - to(x, \iota yP(y)) \wedge |x| = n$

$[[had]](\lambda x.x = HS) = \lambda n_d.\lambda x_e.belong - to(x, HS) \wedge |x| = n \Rightarrow$
 $\lambda x_e.belong - to(x, HS) \wedge |x| = 5 \xrightarrow{\text{Restrict, closure}}$

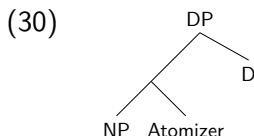
$\exists x[belong - to(x, HS) \wedge |x| = 5 \wedge were - killed(x)]$

- DSKP is not used here, so no contextual cost is incurred.
- So, we capture the patterns: Modification with object-level adjectives is costly, unless there is contextual support or an overt classifier.

- A definite in WA, [N-DEF], can mean either 'the N kind', (28), or 'the unique N':

- (28) John-ə fun-ə pənatʃəntʃe-ts
John-DEF dog-DEF make.extinct-PST.3SG
'John made the dog kind extinct'
- (29) John-ə manug-i-n tasdiajarage-ts
John-DEF child-DAT-DEF educate-PST.3SG
'John educated the (unique) child'

- To capture this in the absence of ambiguity of the bare singular, we need a head to instantiate the kind:



(31) $\lambda P_{et}.\lambda x_e.belong -$
 $to(x, \iota yP(y)) \wedge Atom(x)$

- $[[child\ Atomizer]] = \lambda x_e.belong - to(x, CHILD) \wedge Atom(x)$
- This is the set of instantiations of the child-kind that are atoms.
- The definite article then will return the unique such instantiation if there is one; it will be undefined otherwise.

- Given that DSKP is a general operation, one might expect that it has no positional restrictions.

(32) meyu-??(mə) Marjam-i-n gə-xajte gor
bee-(INDEF) Mariam-DAT-DEF INDC-sting PROG
'A bee is stinging Mariam'

- (32) would appear to go against this idea.
- But it's plausible that bare singulars just cannot move to [Spec, TP] because they lack a D layer. Full DPs on the other hand can, but they do not require DSKP to compose with the verb.

- Covert plurals can be in [Spec, TP]. In that case, they show full agreement and take obligatorily high scope. In that case, DSKP seems to apply:

(33) ??hink anoti zinvor merts-ve-ts-an
five hungry soldier kill-PASS-AOR-PST.3PL
'5 hungry soldiers were killed'

(34) hink had anoti zinvor merts-ve-ts-an
five CLF hungry soldier kill-PASS-AOR-PST.3PL
'5 hungry soldiers were killed'

- One wrinkle is that (33) does not seem to improve in a context that supports 'hungry soldier' as a type.

- Bale, A. & Hrayr Khanjian. 2014. Syntactic complexity and competition: The singular plural distinction in Western Armenian. *Linguistic Inquiry* 45(1). 1–26.
- Chierchia, G. 1998. Reference to Kinds across Language. *Natural Language Semantics* 6(4), 339–405.
- Chung, S. & W. A. Ladusaw. 2004. *Restriction and Saturation*. Cambridge, Massachusetts: MIT Press.
- Dayal, V. 2004. Number marking and indefiniteness in kind terms. *Linguistics and Philosophy* (27), 393–450.

- Landman, F. 1989. Groups, I. *Linguistics and Philosophy* 12(5), 559–605.
- Massam, D. 2001. Pseudo Noun Incorporation In Niuean. *Natural Language & Linguistic Theory* 19(1), 153–197.
- Sağ, Y. 2019. The Semantics of Number Marking: Reference to Kinds, Counting, and Optional Classifiers. PhD Thesis, Rutgers.
- Sigler, M. 1997. Specificity and Agreement in standard Western Armenian. PhD Thesis, MIT.