Redundant errors in child language

Johannes Hein, Imke Driemel, Yining Nie, Fabienne Martin and Artemis Alexiadou

IGRA Colloquium, Universität Leipzig, 13 July 2022

Humboldt-Universität zu Berlin

Introduction

During language acquisition, children produce errors of omission and **commission**:

(1) English past tense
eat > **eat-ed** > **ate-d** > ate (Kuczaj 1977, 1978)

Commission error: Production of overt material not realized in the standard adult language (see e.g. Alexiadou et al. 2021)

- **Distributive** commission error: *eat-ed* (eat-PAST)
- **Redundant** commission error: *ate-d* (eat.PAST-PAST)

See Stemberger (1982) on full vs. partial regularizations

Introduction

Target form	**Commission errors**		Typology of errors from Martin et al. (2021)
	Distributive	**Redundant**	
ate	*eat-ed*	*ate-d*	Kuczaj (1977, 1978),
	EAT PAST	EAT.PAST PAST	Menn and MacWhinney (1984)
donner	*faire avoir*	*faire donner*	Lord (1979),
'to give'	'make have'	'make give'	Bezinska et al. (2008)
mieux	*plus bon*	*plus mieux*	Moline (1971)
better	'more good'	'more better'	Corver (2005)
kein NP	*nicht...ein NP*	*nicht...kein NP*	Nicolae and Yatsushiro (2020),
'no NP'	'not...a NP'	'not...no NP'	Hein et al. (2022)
ohne	*mit nicht/kein*	*mit ohne*	Cohen (1925), Sauerland (2019),
'without'	'with not'	'with without'	Meyer et al. (2021)

Introduction

3 morphological approaches to redundant exponence:

- *ate-d*: [... EAT PAST ...]
- **Allomorphy:** A single feature is shared between two Vocabulary Items as a primary and secondary feature specification
  - *ate* ⇔ [EAT] / __ [PAST], -*ed* ⇔ [PAST]
- **Multiple insertion:** A single feature is realized by different Vocabulary Items
  - *ate* ⇔ [EAT, PAST], -*ed* ⇔ [PAST]
- **Doubling**: A feature is doubled and realized by different Vocabulary Items
  - Doubling rule: [EAT PAST PAST]
  - *ate* ⇔ [EAT, PAST], -*ed* ⇔ [PAST]

Introduction

1. Redundant error case studies from child corpus data
    - French causative
    - English past tense
2. Deriving redundant errors
    - Allomorphy in Distributed Morphology: Insertion of a less specific Vocabulary Item
    - Multiple insertion in Nanosyntax: Overlapping application of phrasal and spanning spellout
3. Non-local redundant error: Negative indefinites
    - Doubling approach

Redundant error case studies

Redundant error case studies

French causative

French causative

- Lexical causative verbs encode a causative meaning component CAUSE, e.g. French *fermer* 'close' or *montrer* 'show'
- Periphrastic causatives can be formed in French using the verb *faire* 'make', which encodes an additional CAUSE component

(2) a. *Montre le camion de pompiers.*
'Show the firetruck.'

b. *J'ai fait montrer le camion au client par un de nos meilleurs vendeurs.*
'I made one of our best salesmen show the truck to the client.'

- French children may produce lexical causatives with a redundant *faire* (Bezinska et al. 2008)

French causative

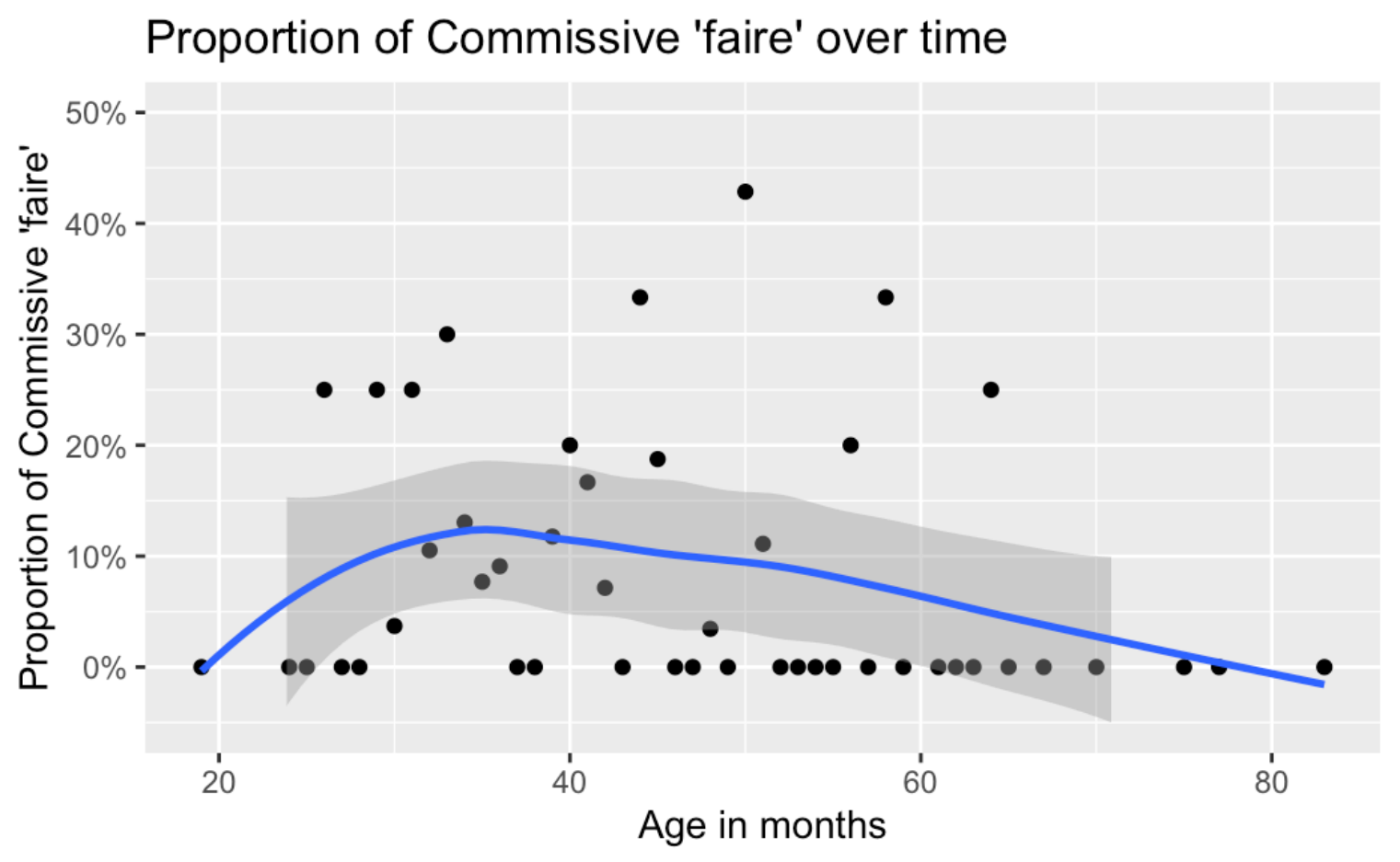
French CHILDES corpus study (Martin et al. 2021)

- We collected all *faire* + infinitive verb occurrences and their conversational contexts in 10 French CHILDES corpora
- *N*=419 occurrences from 83 typically-developing children ages 1;7 to 6;11

Use of *faire*	*N*	%
COMMISSIVE (redundant)	35	8%
NON-COMMISSIVE	335	80%
UNRESOLVED	49	12%

French causative

- Redundant errors: 10% of *faire* causatives up to about 50 months



French causative

- Redundant use of *faire* limited to lexical causatives, suggesting that children are spelling out CAUSE twice

(3) a. *faire fermer les yeux.*
Intended: 'Close the eyes.' (LSN 4;02, Palasis 2009)

b. *va le faire couper.*
Intended: 'Going to cut it.'
(Marilyn 2;09, Demuth and Tremblay 2008)

c. *du bon feu ici pour les faire réchauffer.*
Intended: 'A nice fire here for reheating them.'
(Camille 3,09, Le Normand 1986)

French causative

- Matteo (Palasis 2009) and Madeleine (Morgenstern et al. 2009) use the portmanteau lexical causative form before/alongside the redundant form

(4) a. *Elle a fait tomber ma petite cabane.*
'She made my little shed fall.' (Matteo 2;11)
b. *J'ai montré ça.* 'I showed that.' (Matteo 3;02)
c. *Eh fais montrer le camion de pompiers!*
Intended: 'Hey show the firetruck!' (Matteo 3;03)

(5) *(a)près on va le cacher … on va le cacher … va le faire cacher.*
Lit.: 'Then we'll hide it … we'll hide it … we'll make hide it.'
(Madeleine 2;02)

Causatives

- Redundant exponence of CAUSE is attested in several child languages
  - French (Bezinska et al. 2008, Martin et al. 2021)
  - Turkish (Aksu-Koç and Slobin 1985)
  - Persian (Family and Allen 2015)
  - Japanese (Yamakoshi et al. 2018)
  - English (Lord 1979, Nie et al. in progress)

Redundant error case studies

English past tense

English past tense

- Commission errors involving irregular forms are also known as overregularization (Kuczaj 1977, 1978, Menn and MacWhinney 1984, Marcus et al. 1992)
- Marcus et al. (1992): 2.5% rate of commission errors for past tense irregulars in a subset of English CHILDES corpora (N =11,521 occurrences)
- Stemberger (1982): More distributive errors than redundant errors in adult English (26 distributive vs. 5 redundant errors)

English past tense

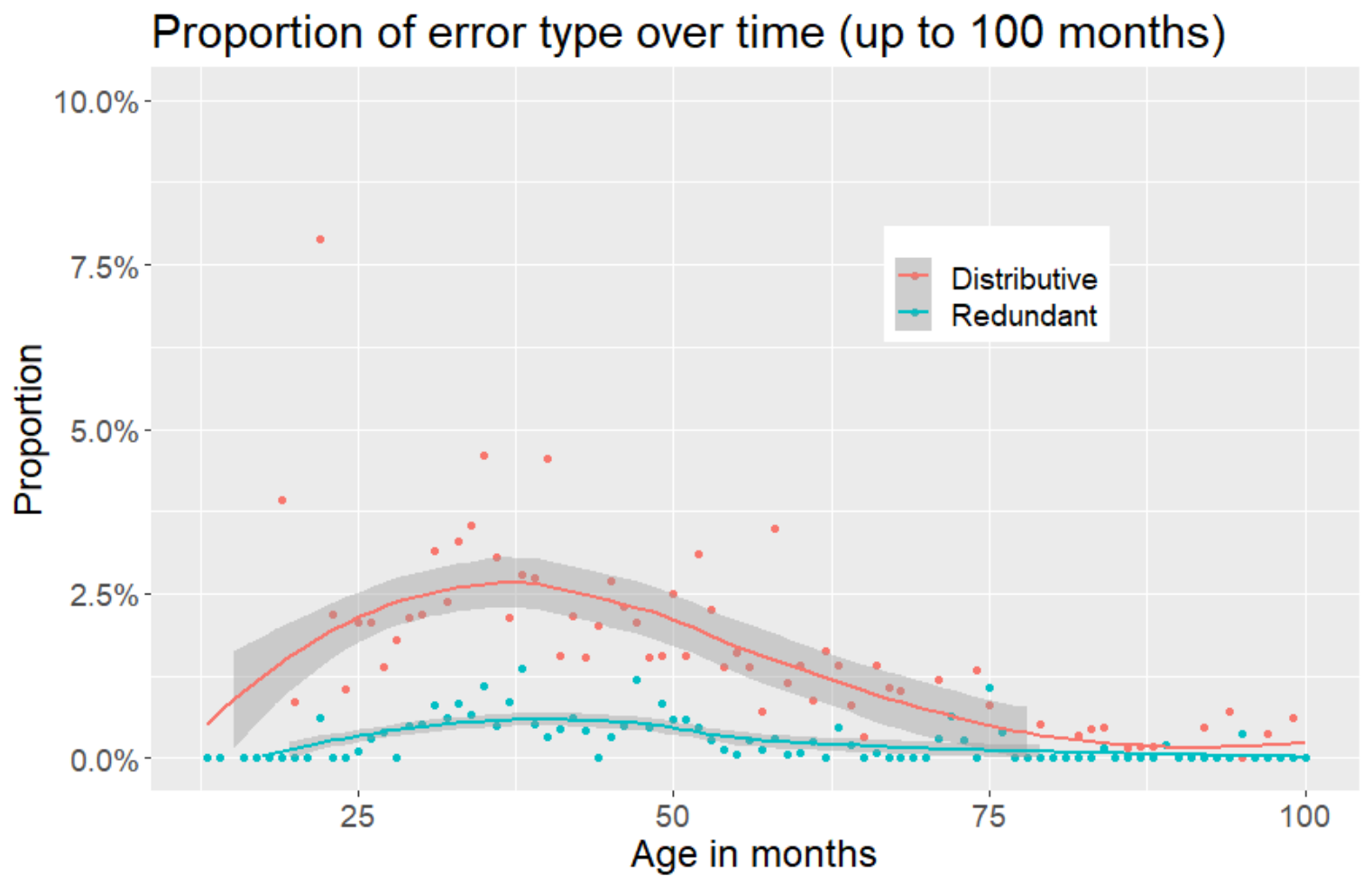
English CHILDES corpus study:

- We collected all past tense occurrences of 38 of the 44 most frequent irregular verbs in 18 UK-English and 40 NA-English corpora
  - Excluded 6 homographs: *cut*, *read*, *let*, *put*, *fit*, *hit*
- $N = 103{,}589$ occurrences from typically-developing children ages 1;5 to 15;11

Error type	*N*	%
DISTRIBUTIVE	1752	1.7%
REDUNDANT	364	0.4%
OTHER	412	0.4%

English past tense

- Distributive errors: 2.5% of past tense irregulars up to 50 months
- Redundant errors: 0.5% of past tense irregulars up to 50 months



English past tense

- Children use the correct irregular past tense before/alongside both the redundant and distributive forms.

(6)
a. Then she **comed** out. (Sarah 4;01, Brown 1973)
b. Then she **came** out again. (Sarah 4;07, Brown 1973)
c. What color **camed** out? (Sarah 4;09, Brown 1973)
d. I **ate** raisins. (Abe 2;11, Kuczaj 1977)
e. I **eated** it with a spoon, Daddy. (Abe 3;05, Kuczaj 1977)
f. She **ated** it, Mom. (Abe 3;11, Kuczaj 1977)

(7) I saw his cousin she had she had she had she had toasted the eggs and they couldn't make none cause cause they **ran** they **ranned** out of eggs. (Lef 4;09, Hall et al. 1984)

Generalizations

- Redundant commission errors are attested in many domains in child language
- The element that is redundantly realized tends to be
  - A higher element in the projection
  - A functional element, rather than the root
- Redundant exponence of lower elements is rare

	Redundant commission error	
Target form	**well attested**	**unattested/rare**
donner	*faire donner*	**donner avoir*
CAUSE HAVE	CAUSE CAUSE.HAVE	CAUSE.HAVE HAVE
ate	*ate-d*	**eat ate*
EAT.PST	EAT.PST-PST	EAT EAT.PST

Deriving redundant errors

Question

- What do children do wrong?
- Which part of their grammar is not adult-like (yet)?
- Distributed Morphology (Halle and Marantz 1993, 1994):
  Answer: They don't fully respect Specificity.
- Nanosyntax (Starke 2009, Caha 2009, et seq.):
  Answer: They don't apply the two modes of lexicalization disjunctively.

Deriving redundant errors

Distributed Morphology

Relevant DM tenets

- Vocabulary items are inserted into terminal nodes following the Subset Principle and Specificity (e.g. Halle 1997).
- Exponents may be specified for two types of features (Carstairs 1987, Noyer 1997)
  - **primary features** must be present on the terminal node targeted for insertion
  - **secondary features** must be present on a terminal node in the local environment of the terminal node targeted for insertion.[1]
- Those secondary (or contextual) features count for calculation of specificity since they further narrow an exponents distribution.

[1] See Stump (2001), Müller (2020) for problems of secondary features.

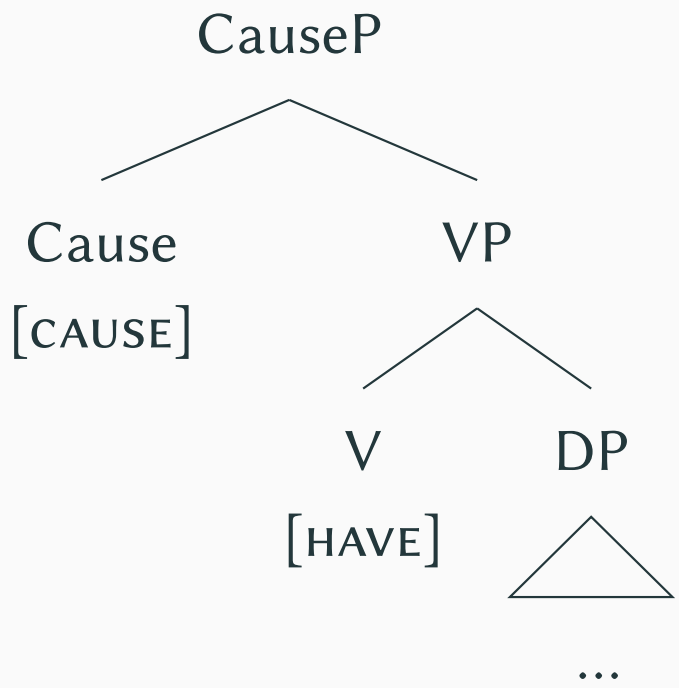
Commission errors in DM

Claim

Children's commission errors result from **disregarding specificity**, in particular when secondary features are involved.

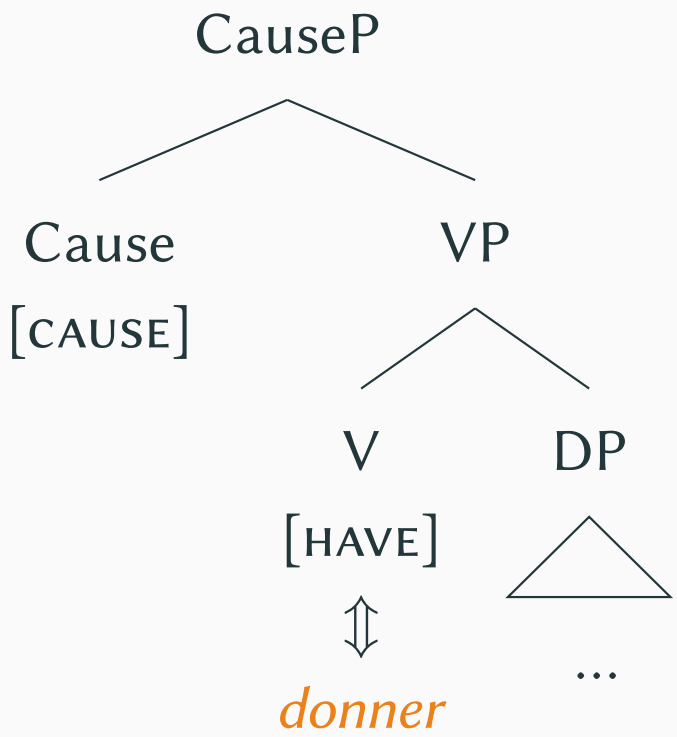
Causatives: Target *donner* ‘give’

(8) a. /avoir/ ⇔ [HAVE]
b. /faire/ ⇔ [CAUSE]
c. /donner/ ⇔ [HAVE] / __ CAUSE
d. /∅/ ⇔ [CAUSE] / __ {HAVE, ...}



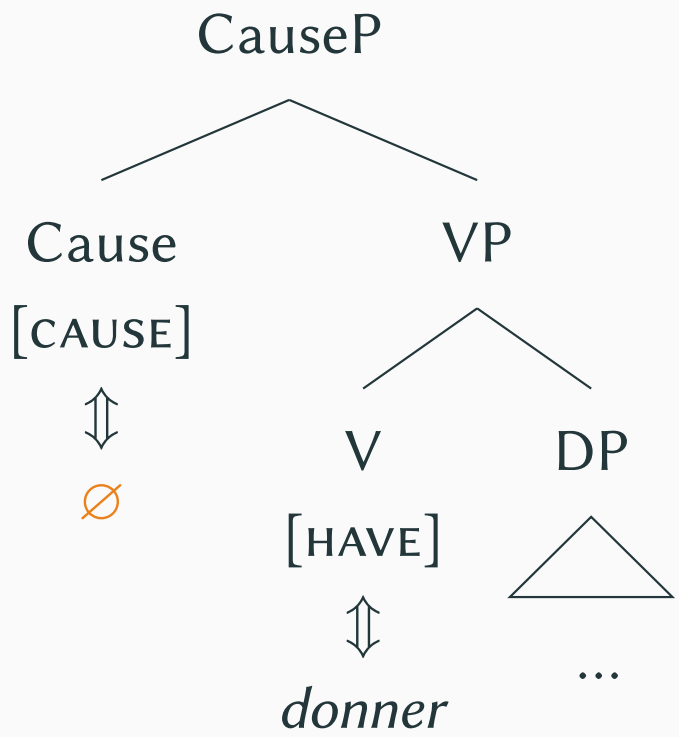
Causatives: Target *donner* 'give'

(8) a. /avoir/ ⇔ [HAVE]
b. /faire/ ⇔ [CAUSE]
c. /donner/ ⇔ [HAVE] / __ CAUSE
d. /∅/ ⇔ [CAUSE] / __ {HAVE, ...}



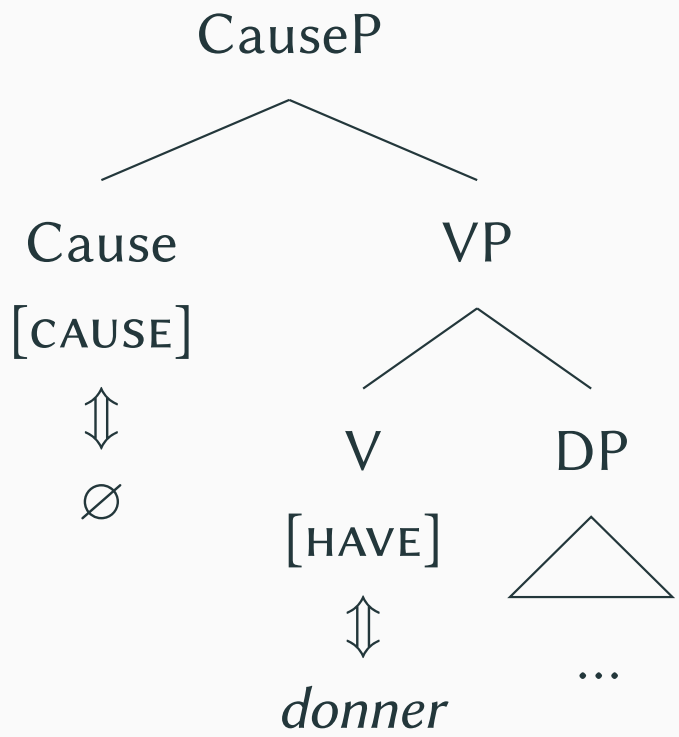
Causatives: Target *donner* 'give'

(8) a. /avoir/ ⇔ [HAVE]
b. /faire/ ⇔ [CAUSE]
c. /donner/ ⇔ [HAVE] / __ CAUSE
d. /∅/ ⇔ [CAUSE] / __ {HAVE, ...}



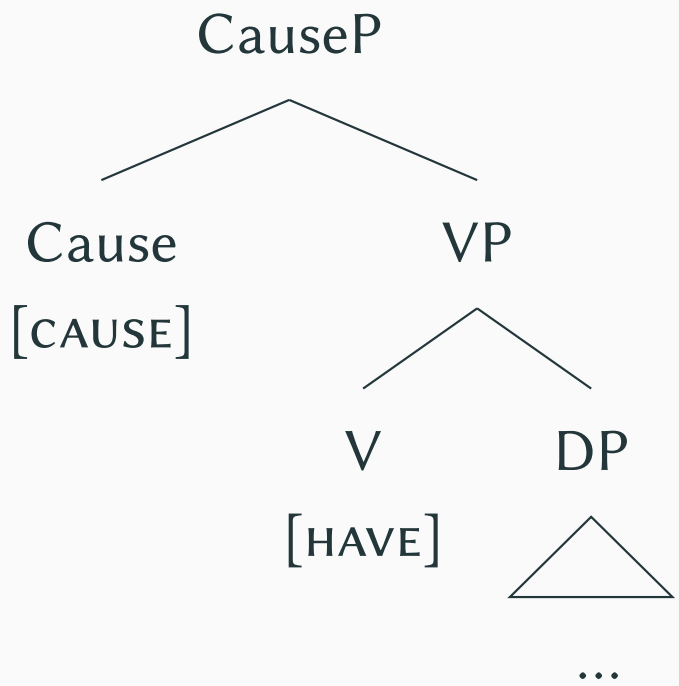
Causatives: Target *donner* 'give'

(8) a. /avoir/ ⇔ [HAVE]
b. /faire/ ⇔ [CAUSE]
c. /donner/ ⇔ [HAVE] / __ CAUSE
d. /∅/ ⇔ [CAUSE] / __ {HAVE, ...}



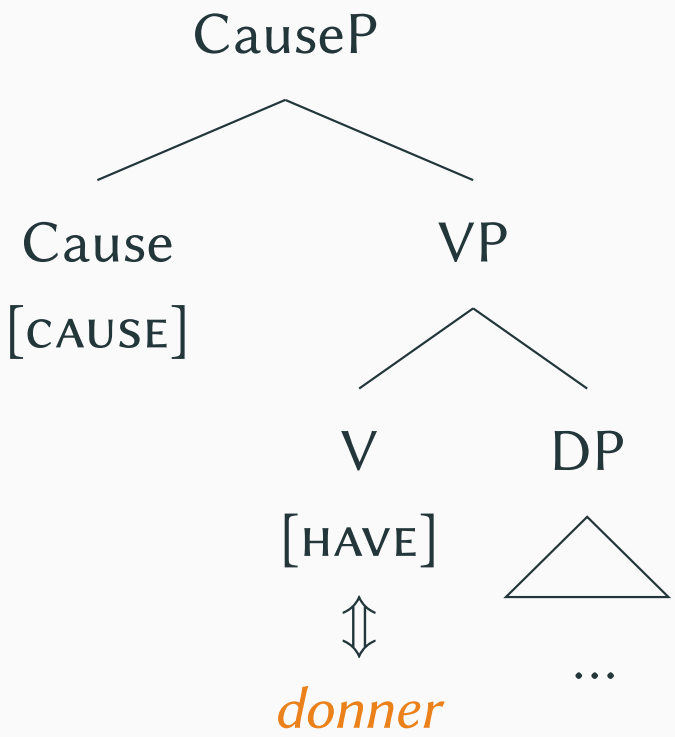
Causatives: Redundant *faire donner* ‘make give’

(9) a. /avoir/ ⇔ [HAVE]
b. /faire/ ⇔ [CAUSE]
c. /donner/ ⇔ [HAVE] / __ CAUSE
d. /∅/ ⇔ [CAUSE] / __ {HAVE, ...}



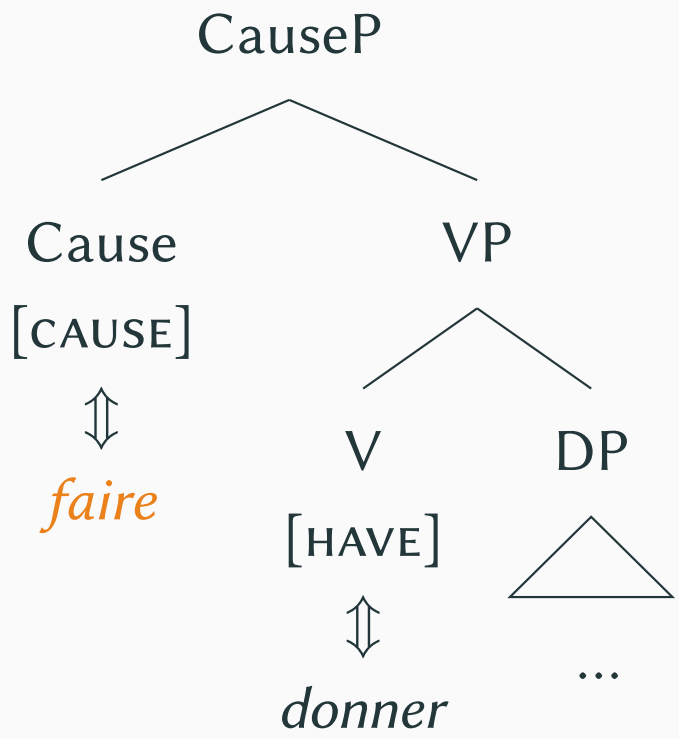
Causatives: Redundant *faire donner* ‘make give’

(9) a. /avoir/ ⇔ [HAVE]
b. /faire/ ⇔ [CAUSE]
c. /donner/ ⇔ [HAVE] / __ CAUSE
d. /∅/ ⇔ [CAUSE] / __ {HAVE, ...}



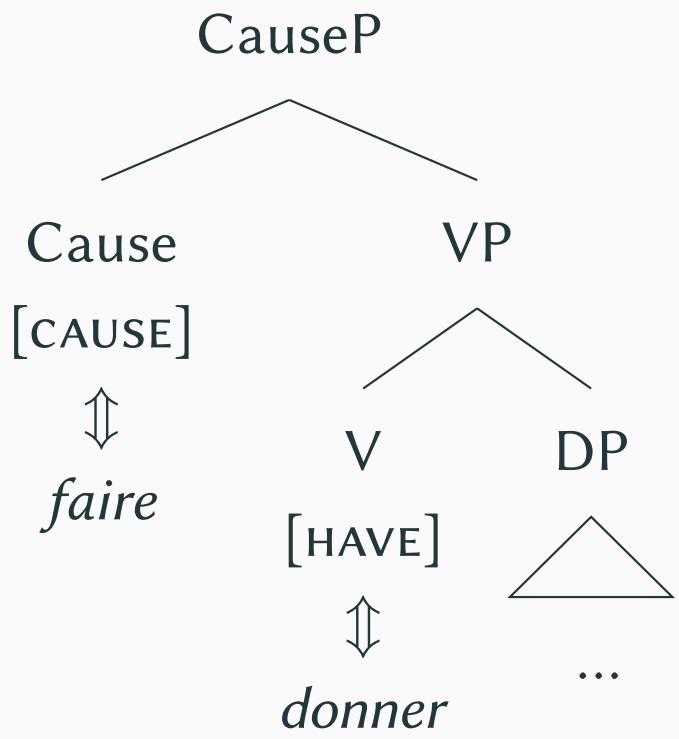
Causatives: Redundant *faire donner* ‘make give’

(9) a. /avoir/ ⇔ [HAVE]
b. /faire/ ⇔ [CAUSE]
c. /donner/ ⇔ [HAVE] / __ CAUSE
d. /∅/ ⇔ [CAUSE] / __ {HAVE, ...}



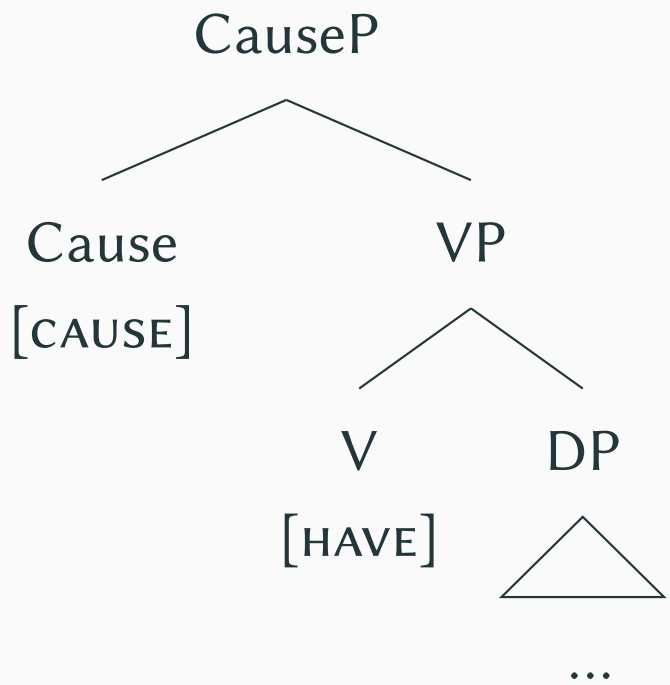
Causatives: Redundant *faire donner* ‘make give’

(9) a. /avoir/ ⇔ [HAVE]
b. /faire/ ⇔ [CAUSE]
c. /donner/ ⇔ [HAVE] / __ CAUSE
d. /∅/ ⇔ [CAUSE] / __ {HAVE, ...}



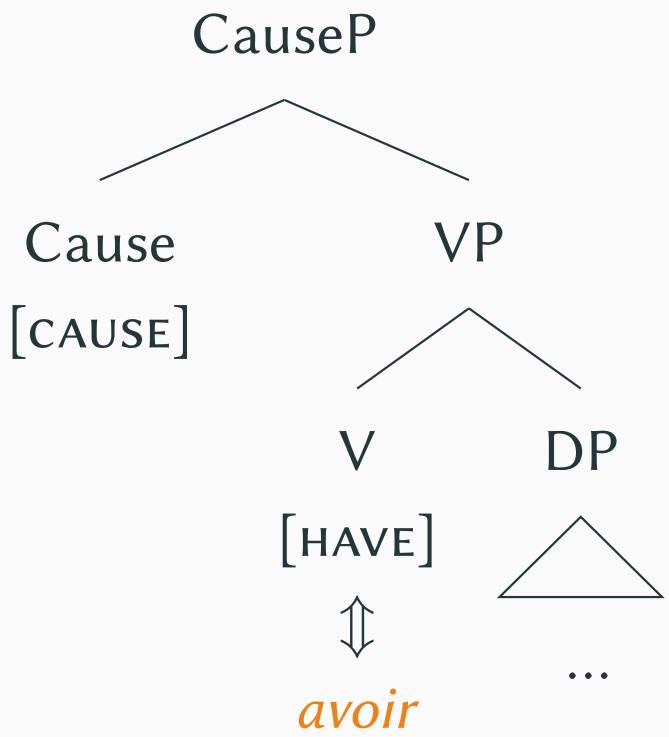
Causatives: Distributive *faire avoir* 'make have'

(10) a. /avoir/ ⇔ [HAVE]
b. /faire/ ⇔ [CAUSE]
c. /donner/ ⇔ [HAVE] / __ CAUSE
d. /∅/ ⇔ [CAUSE] / __ {HAVE, ...}



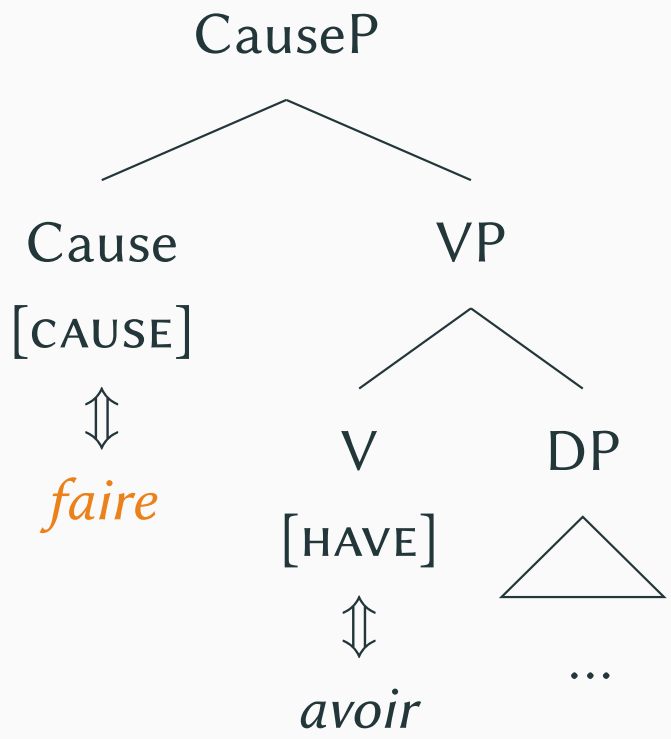
Causatives: Distributive *faire avoir* 'make have'

(10) a. /avoir/ ⇔ [HAVE]
b. /faire/ ⇔ [CAUSE]
c. /donner/ ⇔ [HAVE] / __ CAUSE
d. /∅/ ⇔ [CAUSE] / __ {HAVE, ...}



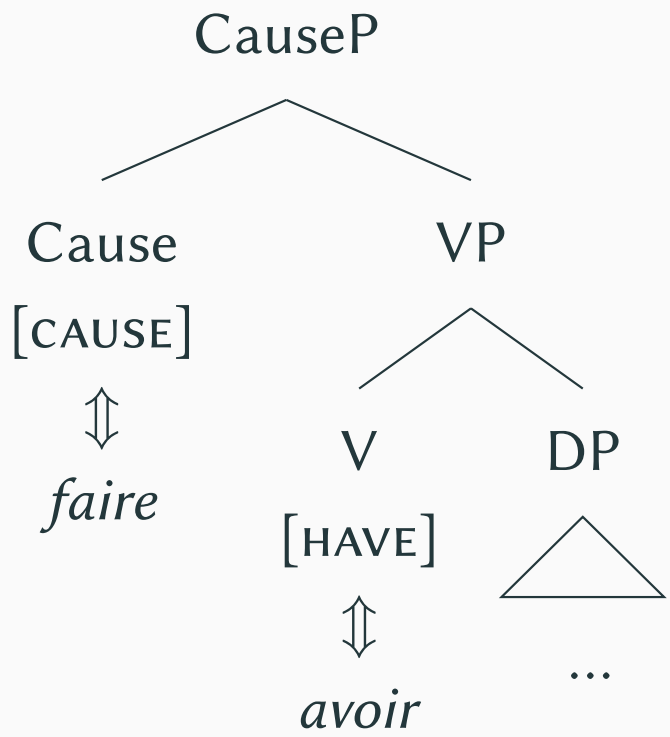
Causatives: Distributive *faire avoir* 'make have'

(10) a. /avoir/ ⇔ [HAVE]
b. /faire/ ⇔ [CAUSE]
c. /donner/ ⇔ [HAVE] / __ CAUSE
d. /∅/ ⇔ [CAUSE] / __ {HAVE, ...}



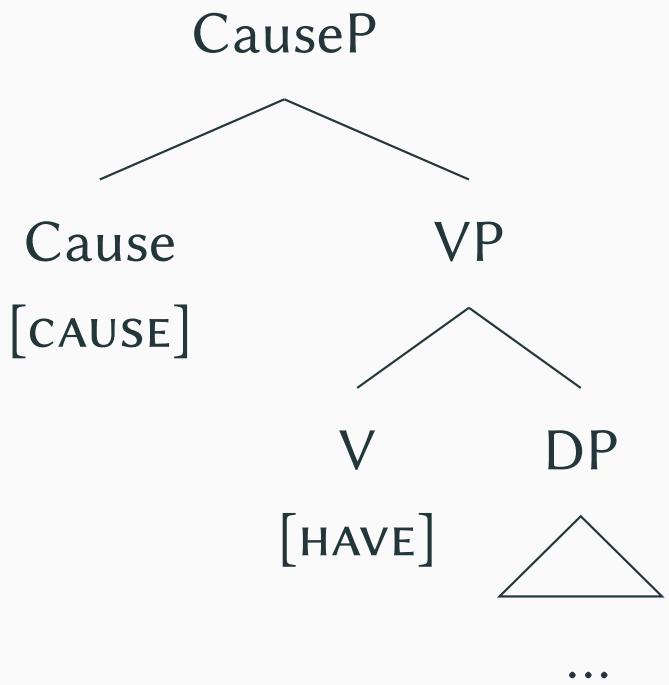
Causatives: Distributive *faire avoir* 'make have'

(10) a. /avoir/ ⇔ [HAVE]
b. /faire/ ⇔ [CAUSE]
c. /donner/ ⇔ [HAVE] / __ CAUSE
d. /∅/ ⇔ [CAUSE] / __ {HAVE, ...}



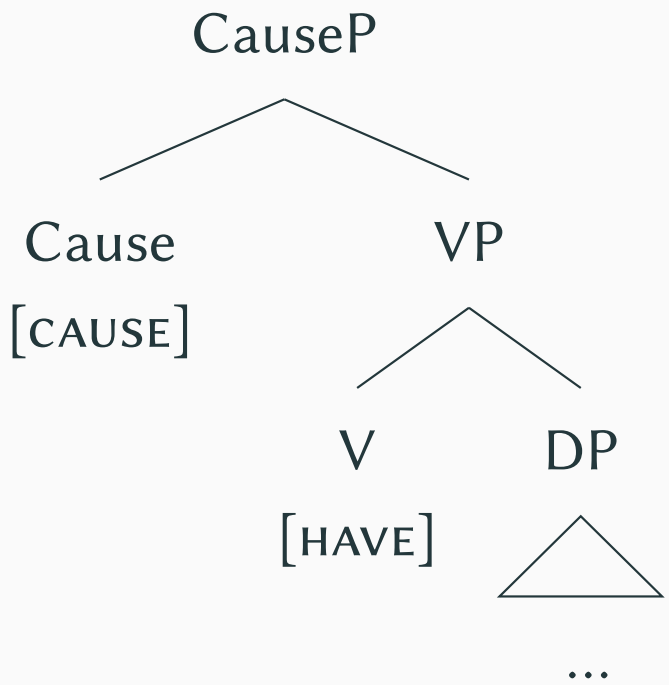
*donne avoir

(11) a. /avoir/ ⇔ [HAVE]
b. /faire/ ⇔ [CAUSE]
c. /donner/ ⇔ [HAVE] / __ CAUSE
d. /∅/ ⇔ [CAUSE] / __ {HAVE, ...}



*donne avoir

(11) a. /avoir/ ⇔ [HAVE]
b. /faire/ ⇔ [CAUSE]
c. /donner/ ⇔ [HAVE] / __ CAUSE
d. /∅/ ⇔ [CAUSE] / __ {HAVE, ...}



*donne avoir

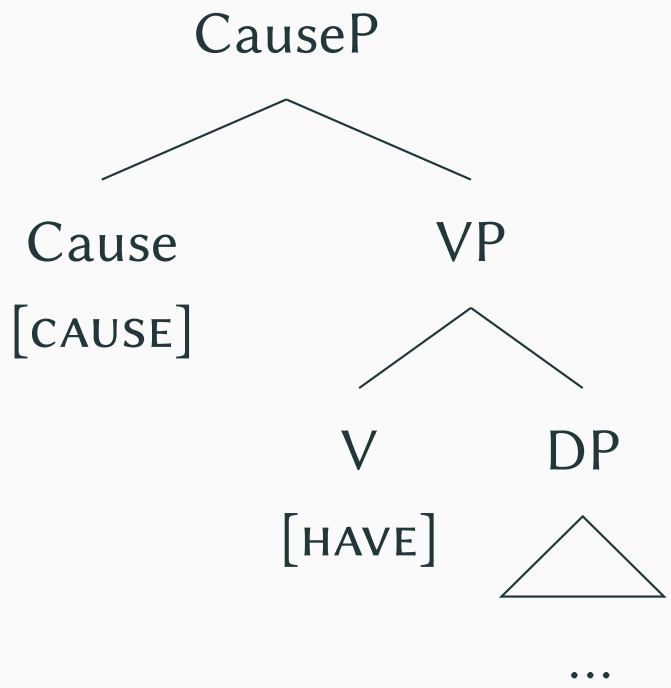
(11) *Implausible VIs*

a. /avoir/ ↔ [HAVE]

b. /faire/ ↔ [CAUSE]

c. /donner/ ↔ [CAUSE] / __ HAVE

d. /∅/ ↔ [{HAVE, ...}] / __ CAUSE



*donne avoir

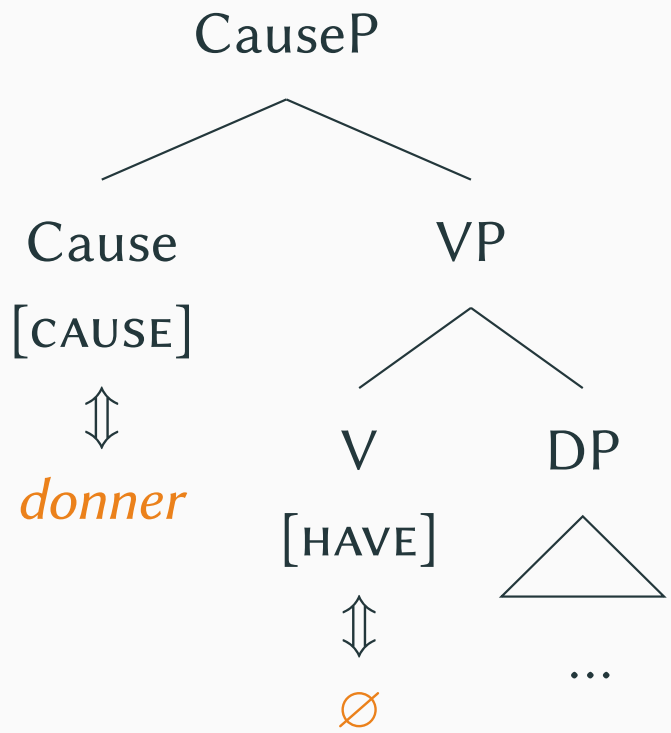
(11) *Implausible VIs*

a. /avoir/ ↔ [HAVE]

b. /faire/ ↔ [CAUSE]

c. /donner/ ↔ [CAUSE] / __ HAVE

d. /∅/ ↔ [{HAVE, ...}] / __ CAUSE



*donne avoir

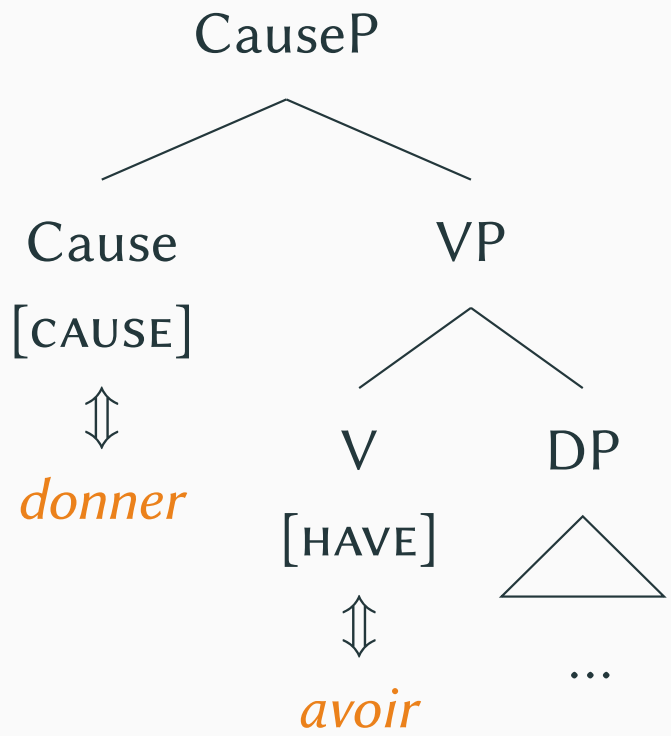
(11) *Implausible VIs*

a. /avoir/ ↔ [HAVE]

b. /faire/ ↔ [CAUSE]

c. /donner/ ↔ [CAUSE] / __ HAVE

d. /∅/ ↔ [{HAVE, ...}] / __ CAUSE



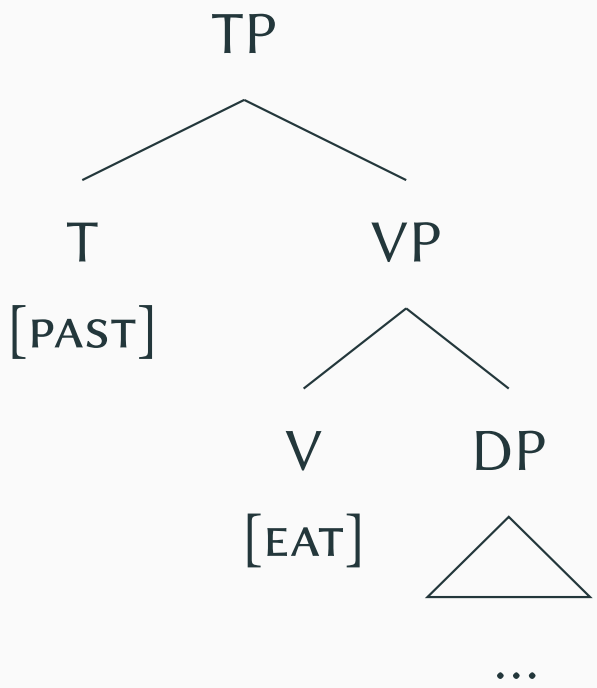
DM: A typology of causative errors

(12) a. /avoir/ ↔ [HAVE]
b. /faire/ ↔ [CAUS]
c. /donner/ ↔ [HAVE] / __ CAUS
d. /∅/ ↔ [CAUS] / __ {HAVE, DRY, ...}

[CAUS]	[$\sqrt{\text{HAVE}}$]	error location	error type
∅	*donner*	none	target
∅	*avoir*	ROOT	omissive
fais	*donner*	CAUSE	redundant
fais	*avoir*	ROOT & CAUSE	distributive

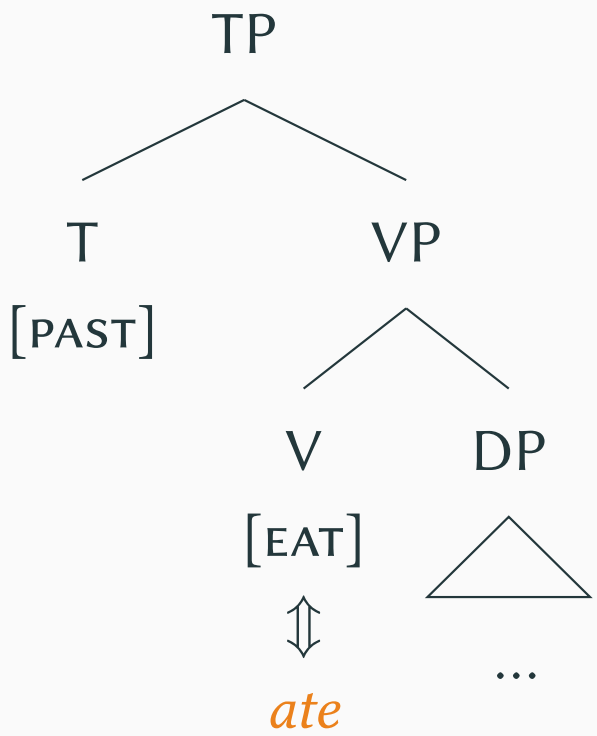
Past tense: Target *ate*

(13) a. /eat/ ⇔ [EAT]
b. /-ed/ ⇔ [PAST]
c. /ate/ ⇔ [EAT] / ___ PAST
d. /∅/ ⇔ [PAST] / ___ {EAT, BREAK, …}



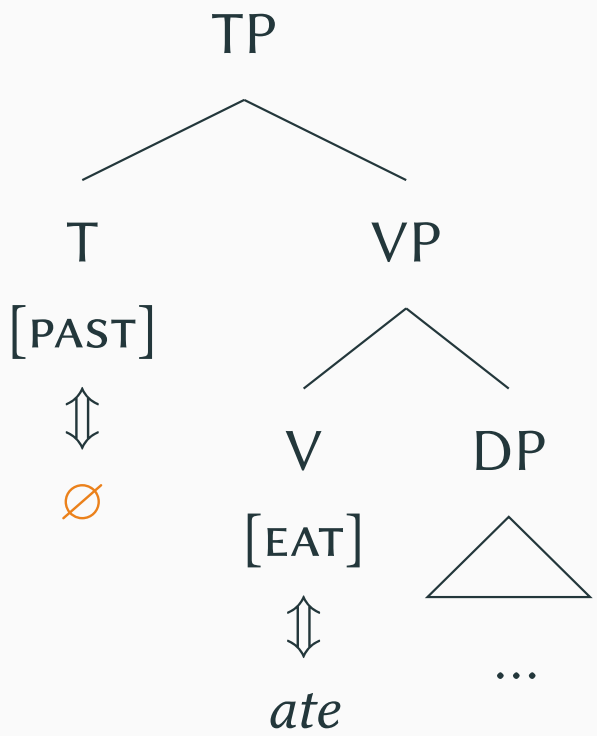
Past tense: Target *ate*

(13) a. /eat/ ⇔ [EAT]
b. /-ed/ ⇔ [PAST]
c. /ate/ ⇔ [EAT] / __ PAST
d. /∅/ ⇔ [PAST] / __ {EAT, BREAK, …}



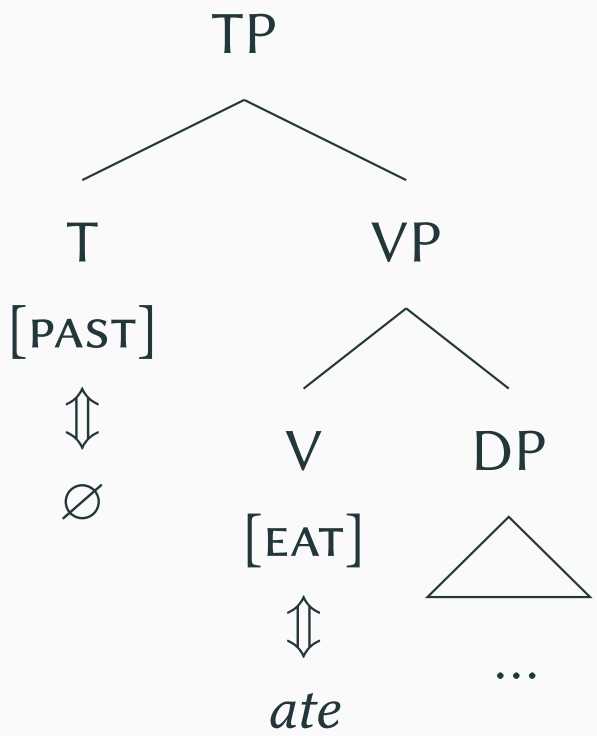
Past tense: Target *ate*

(13) a. /eat/ ⇔ [EAT]
b. /-ed/ ⇔ [PAST]
c. /ate/ ⇔ [EAT] / ___ PAST
d. /∅/ ⇔ [PAST] / ___ {EAT, BREAK, …}



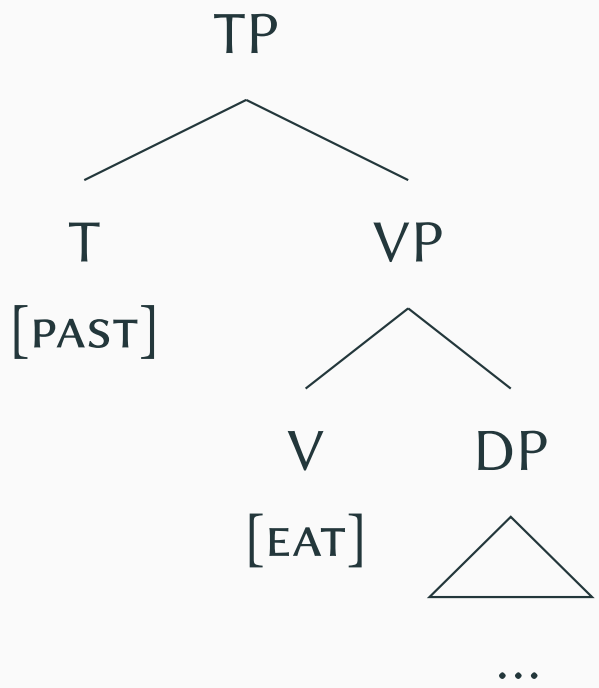
Past tense: Target *ate*

(13) a. /eat/ ⇔ [EAT]
b. /-ed/ ⇔ [PAST]
c. /ate/ ⇔ [EAT] / __ PAST
d. /∅/ ⇔ [PAST] / __ {EAT, BREAK, …}



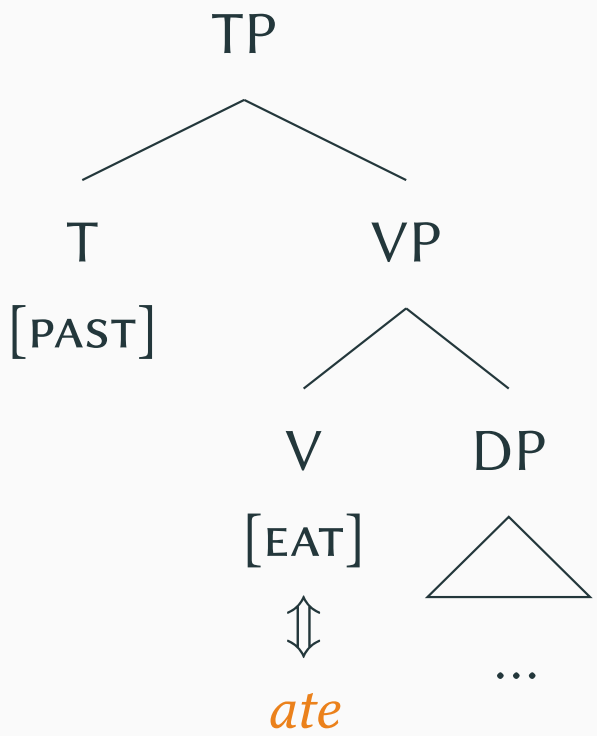
Past tense: Redundant *ate-d*

(14) a. /eat/ ⇔ [EAT]
b. /-ed/ ⇔ [PAST]
c. /ate/ ⇔ [EAT] / __ PAST
d. /∅/ ⇔ [PAST] / __ {EAT, BREAK, …}



Past tense: Redundant *ate-d*

(14) a. /eat/ ⇔ [EAT]
b. /-ed/ ⇔ [PAST]
c. /ate/ ⇔ [EAT] / __ PAST
d. /∅/ ⇔ [PAST] / __ {EAT, BREAK, …}



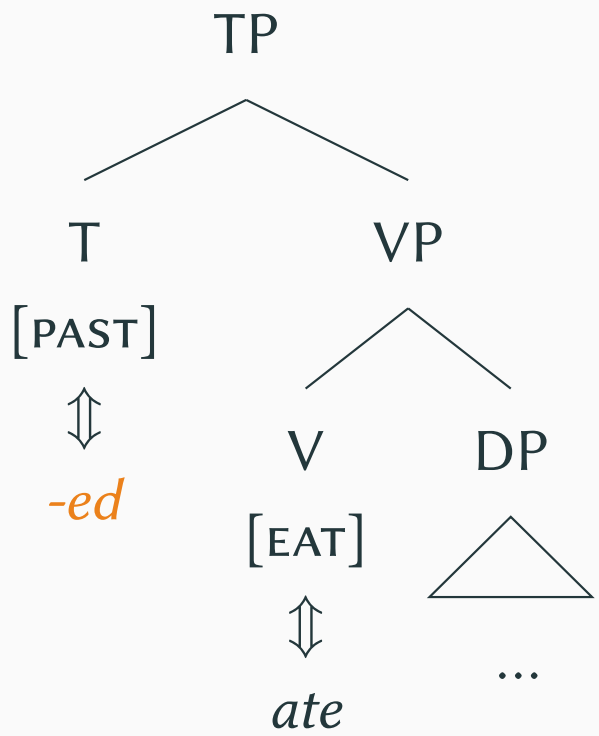
Past tense: Redundant *ate-d*

(14) a. /eat/ ⇔ [EAT]

b. /-ed/ ⇔ [PAST]

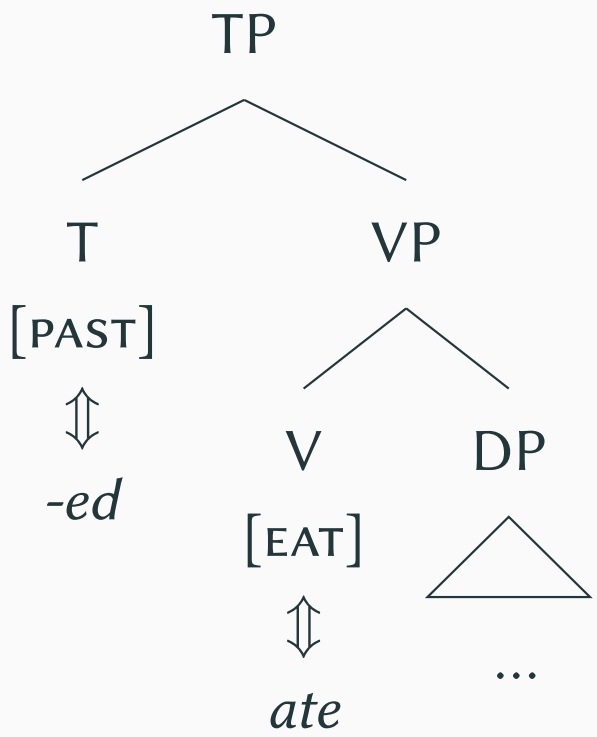
c. /ate/ ⇔ [EAT] / ___ PAST

d. /∅/ ⇔ [PAST] / ___ {EAT, BREAK, …}



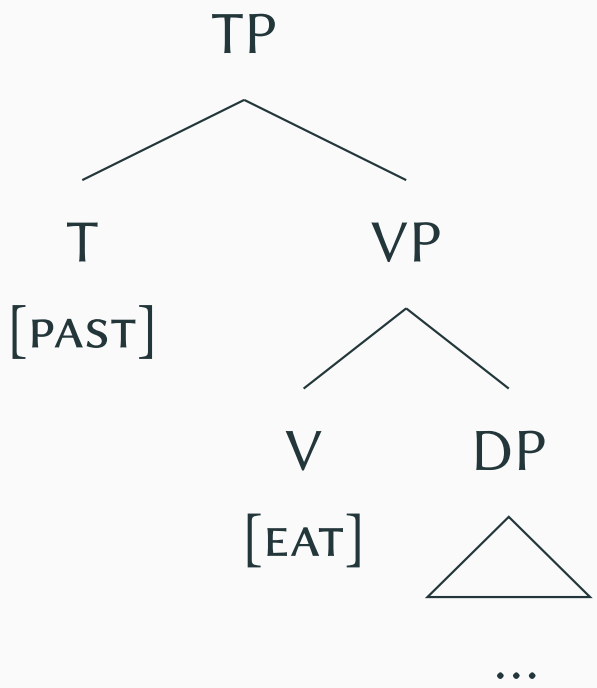
Past tense: Redundant *ate-d*

(14) a. /eat/ ⇔ [EAT]
b. /-ed/ ⇔ [PAST]
c. /ate/ ⇔ [EAT] / __ PAST
d. /∅/ ⇔ [PAST] / __ {EAT, BREAK, …}



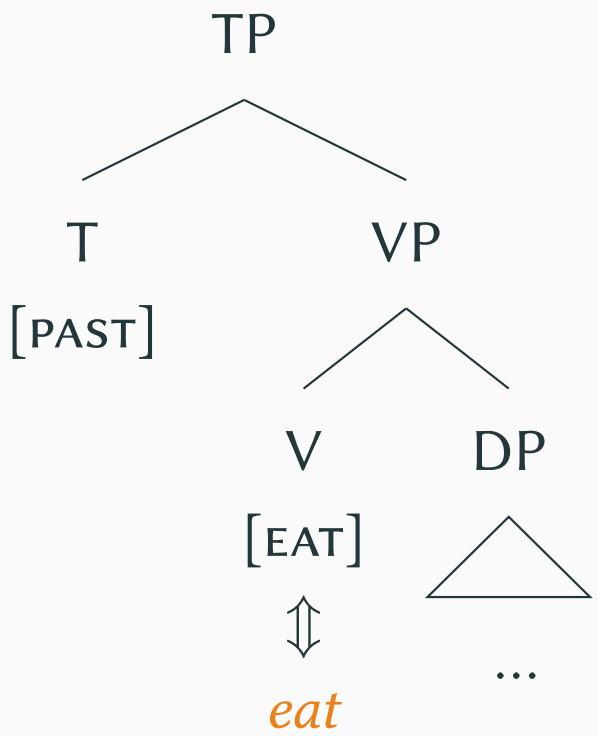
Past tense: Distributive *eat-ed*

(15) a. /eat/ ⇔ [EAT]
b. /-ed/ ⇔ [PAST]
c. /ate/ ⇔ [EAT] / __ PAST
d. /∅/ ⇔ [PAST] / __ {EAT, BREAK, …}



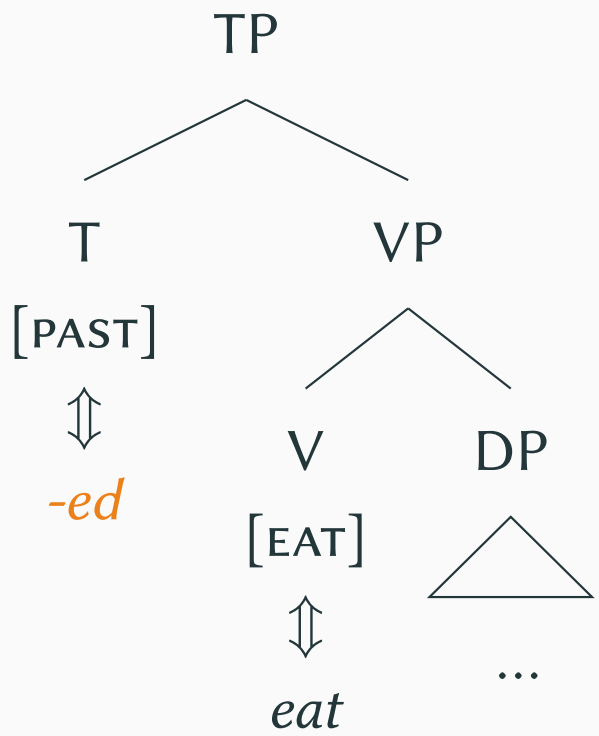
Past tense: Distributive *eat-ed*

(15) a. /eat/ ⇔ [EAT]
b. /-ed/ ⇔ [PAST]
c. /ate/ ⇔ [EAT] / ___ PAST
d. /∅/ ⇔ [PAST] / ___ {EAT, BREAK, …}



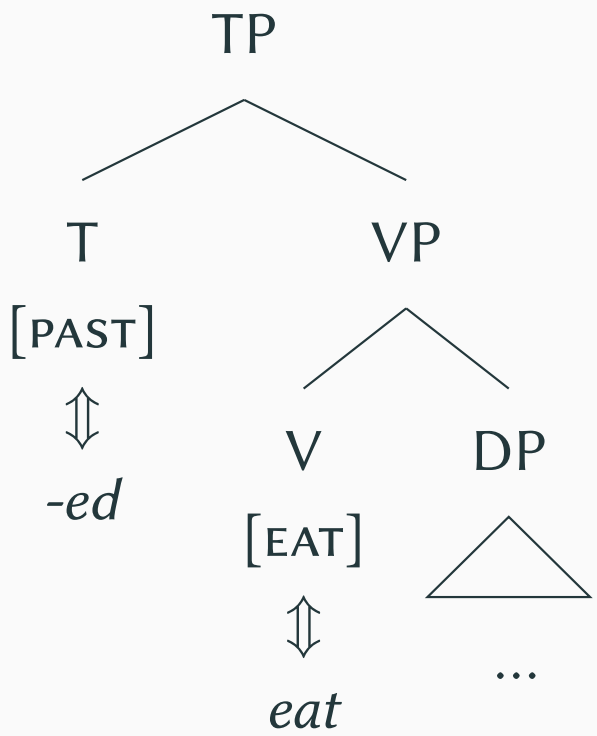
Past tense: Distributive *eat-ed*

(15) a. /eat/ ⇔ [EAT]
b. /-ed/ ⇔ [PAST]
c. /ate/ ⇔ [EAT] / ___ PAST
d. /∅/ ⇔ [PAST] / ___ {EAT, BREAK, …}



Past tense: Distributive *eat-ed*

(15) a. /eat/ ⇔ [EAT]
b. /-ed/ ⇔ [PAST]
c. /ate/ ⇔ [EAT] / ___ PAST
d. /∅/ ⇔ [PAST] / ___ {EAT, BREAK, …}



DM: Typology of past tense errors

(16) a. /eat/ ↔ [EAT]
b. /-ed/ ↔ [PAST]
c. /ate/ ↔ [EAT] / ___ PAST
d. /∅/ ↔ [PAST] / ___ {EAT, BREAK, ...}

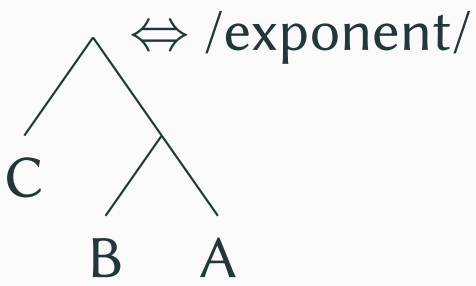
[$\sqrt{\text{EAT}}$]	[PAST]	error location	error type
ate	∅	none	target
eat	∅	ROOT	omissive
eat	*-ed*	ROOT & PAST	distributive
ate	*-ed*	PAST	redundant

Deriving redundant errors

Nanosyntax

Relevant tenets of Nanosyntax: Constituent spellout

- Nanosyntax (Starke 2009, Caha 2009, et seq.) allows non-terminal spellout, i.e. spellout out of several terminal nodes that form a constituent at once.

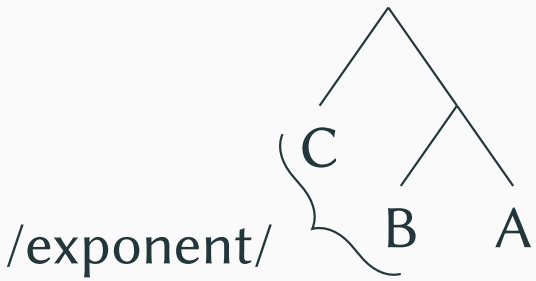
(17) 

- Lexicalization follows the Superset Principle. Previous lexicalizations may be overridden by subsequent lexicalizations.

Relevant tenets of Nanosyntax: Spanning spellout

- Spanning (Williams 2003, Abels and Muriungi 2008, Taraldsen 2010, Svenonius 2012, a.o.) allows lexical items to spell out non-constituents (span = "a contiguous sequence of heads in a head-complement relation", Svenonius 2016: 205).

(18)



Nanosyntax: Commission errors

Claim

Children's commission errors result from **erroneous overlapping application** of spanning lexicalization (S-lexicalization) and constituent lexicalization (C-lexicalization).

Causatives: Target *donner* ‘give’

(19) *Lexical items*

a. avoir ⇔ [HAVE]

b. donner ⇔ [CAUSE [HAVE]]

c. faire ⇔ [CAUSE

(20) *C-lexicalization overrides previous C-lexicalization*

Causatives: Target *donner* ‘give’

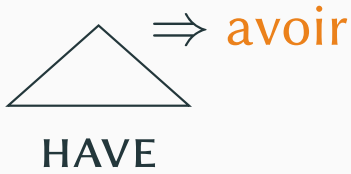
(19) *Lexical items*

a. avoir ⇔ [HAVE]

b. donner ⇔ [CAUSE [HAVE]]

c. faire ⇔ [CAUSE

(20) *C-lexicalization overrides previous C-lexicalization*



Causatives: Target *donner* ‘give’

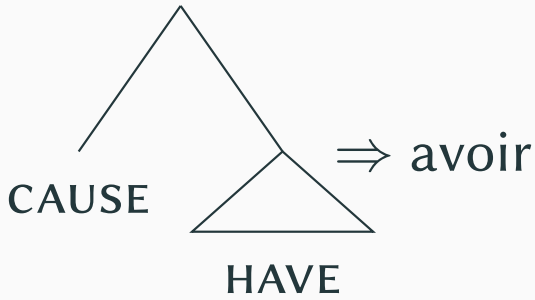
(19) *Lexical items*

a. avoir $\Leftrightarrow$ [HAVE]

b. donner $\Leftrightarrow$ [CAUSE [HAVE]]

c. faire $\Leftrightarrow$ [CAUSE

(20) *C-lexicalization overrides previous C-lexicalization*



Causatives: Target *donner* ‘give’

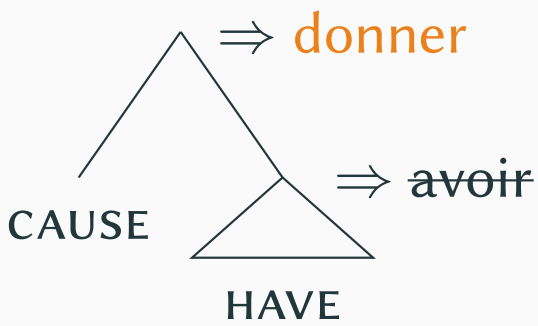
(19) *Lexical items*

a. avoir ⇔ [HAVE]

b. donner ⇔ [CAUSE [HAVE]]

c. faire ⇔ [CAUSE

(20) *C-lexicalization overrides previous C-lexicalization*



Causatives: Redundant *faire donner* 'make give'

(21) *Lexical items*

a. avoir ⇔ [HAVE]

b. donner ⇔ [CAUSE [HAVE]]

c. faire ⇔ [CAUSE

(22) *Simultaneous C-lexicalization and S-lexicalization*

Causatives: Redundant *faire donner* ‘make give’

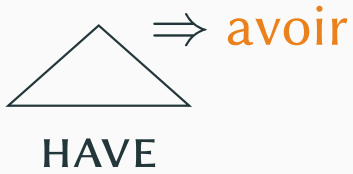
(21) *Lexical items*

a. avoir $\Leftrightarrow$ [HAVE]

b. donner $\Leftrightarrow$ [CAUSE [HAVE]]

c. faire $\Leftrightarrow$ [CAUSE

(22) *Simultaneous C-lexicalization and S-lexicalization*



Causatives: Redundant *faire donner* ‘make give’

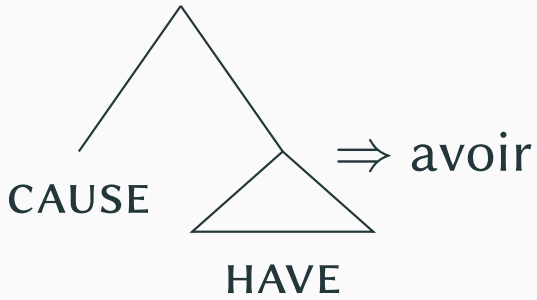
(21) *Lexical items*

a. avoir $\Leftrightarrow$ [HAVE]

b. donner $\Leftrightarrow$ [CAUSE [HAVE]]

c. faire $\Leftrightarrow$ [CAUSE

(22) *Simultaneous C-lexicalization and S-lexicalization*



Causatives: Redundant *faire donner* ‘make give’

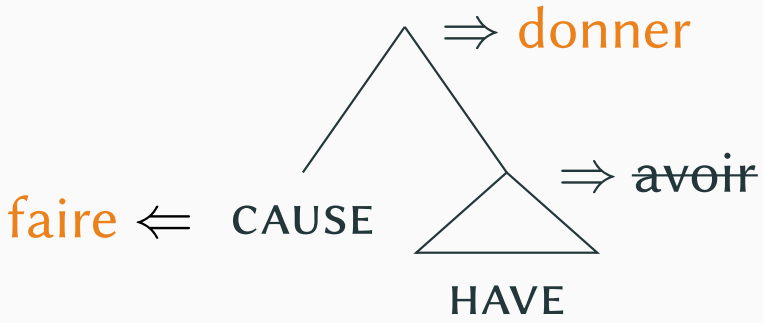
(21) *Lexical items*

a. avoir $\Leftrightarrow$ [HAVE]

b. donner $\Leftrightarrow$ [CAUSE [HAVE]]

c. faire $\Leftrightarrow$ [CAUSE

(22) *Simultaneous C-lexicalization and S-lexicalization*



Causatives: Distributive *faire avoir* ‘make have’

(23) *Lexical items*

a. avoir ⇔ [HAVE]

b. donner ⇔ [CAUSE [HAVE]]

c. faire ⇔ [CAUSE

(24) *Neglecting C-lexicalization*

Causatives: Distributive *faire avoir* 'make have'

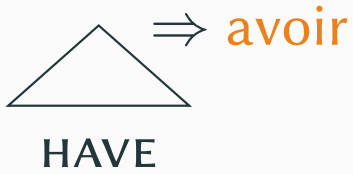
(23) *Lexical items*

a. avoir ⇔ [HAVE]

b. donner ⇔ [CAUSE [HAVE]]

c. faire ⇔ [CAUSE

(24) *Neglecting C-lexicalization*



Causatives: Distributive *faire avoir* 'make have'

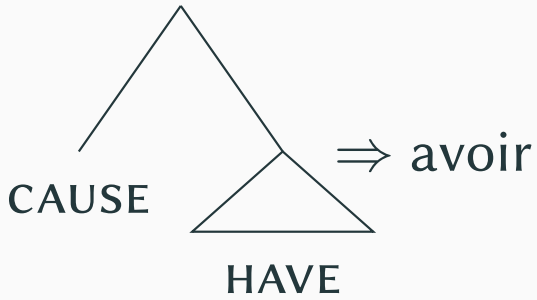
(23) *Lexical items*

a. avoir $\Leftrightarrow$ [HAVE]

b. donner $\Leftrightarrow$ [CAUSE [HAVE]]

c. faire $\Leftrightarrow$ [CAUSE

(24) *Neglecting C-lexicalization*



Causatives: Distributive *faire avoir* 'make have'

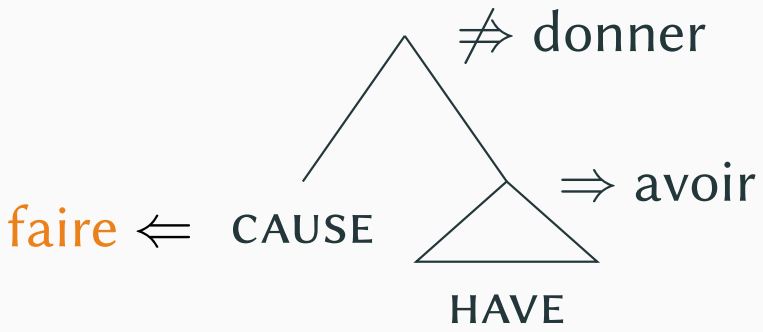
(23) *Lexical items*

a. avoir $\Leftrightarrow$ [HAVE]

b. donner $\Leftrightarrow$ [CAUSE [HAVE]]

c. faire $\Leftrightarrow$ [CAUSE

(24) *Neglecting C-lexicalization*



*donne avoir

(25) *Lexical items*

a. avoir ⇔ [HAVE]

b. donner ⇔ [CAUSE [HAVE]]

c. faire ⇔ [CAUSE

(26) *Failure to override?*

*donne avoir

(25) *Lexical items*

a. avoir ⇔ [HAVE]

b. donner ⇔ [CAUSE [HAVE]]

c. faire ⇔ [CAUSE

(26) *Failure to override?*

*donne avoir

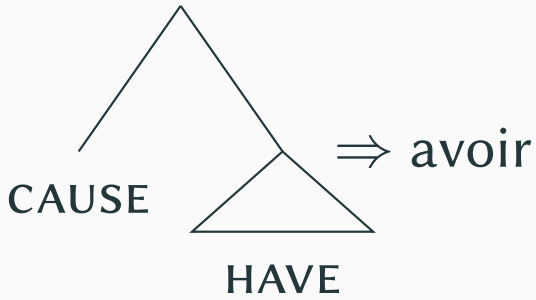
(25) *Lexical items*

a. avoir ⇔ [HAVE]

b. donner ⇔ [CAUSE [HAVE]]

c. faire ⇔ [CAUSE

(26) *Failure to override?*



*donne avoir

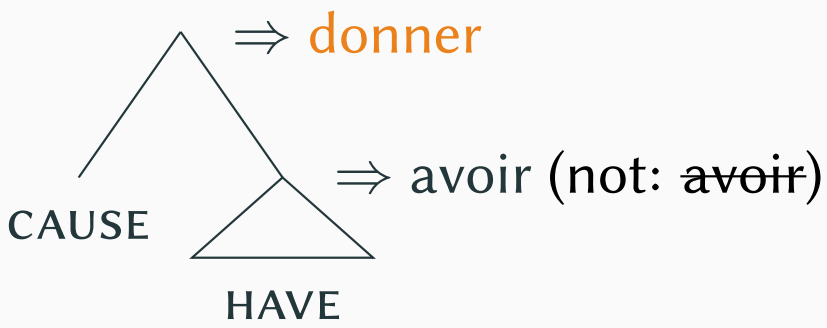
(25) *Lexical items*

a. avoir ⇔ [HAVE]

b. donner ⇔ [CAUSE [HAVE]]

c. faire ⇔ [CAUSE

(26) *Failure to override?*



Nanosyntax: Summary causatives

	failure	**location**	**error type**
donner	none	none	target
faire avoir	no C	2nd cycle	distributive
faire donner	simultaneous C & S	2nd cycle	redundant

Past tense: Target *ate*

(27) *Lexical items*

a. eat ⇔ [EAT]

b. ate ⇔ [PAST [EAT]]

c. -ed ⇔ [PAST

(28) *C-lexicalization overrides previous C-lexicalization*

Past tense: Target *ate*

(27) *Lexical items*

a. eat ⇔ [EAT]

b. ate ⇔ [PAST [EAT]]

c. -ed ⇔ [PAST

(28) *C-lexicalization overrides previous C-lexicalization*

Past tense: Target *ate*

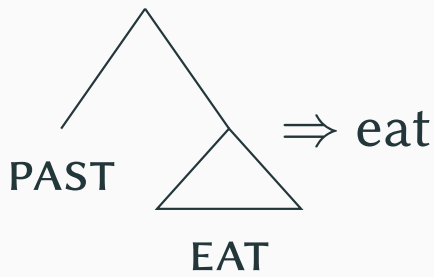
(27) *Lexical items*

a. eat ⇔ [EAT]

b. ate ⇔ [PAST [EAT]]

c. -ed ⇔ [PAST

(28) *C-lexicalization overrides previous C-lexicalization*



Past tense: Target *ate*

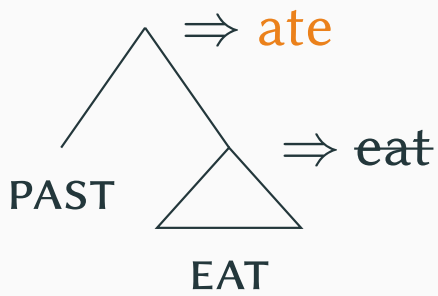
(27) *Lexical items*

a. eat ⇔ [EAT]

b. ate ⇔ [PAST [EAT]]

c. -ed ⇔ [PAST

(28) *C-lexicalization overrides previous C-lexicalization*



Past tense: Redundant *ate-d*

(29) *Lexical items*

a. eat ⇔ [EAT]

b. ate ⇔ [PAST [EAT]]

c. -ed ⇔ [PAST

(30) *Simultaneous C-lexicalization and S-lexicalization*

Past tense: Redundant *ate-d*

(29) *Lexical items*

a. eat ⇔ [EAT]

b. ate ⇔ [PAST [EAT]]

c. -ed ⇔ [PAST

(30) *Simultaneous C-lexicalization and S-lexicalization*

Past tense: Redundant *ate-d*

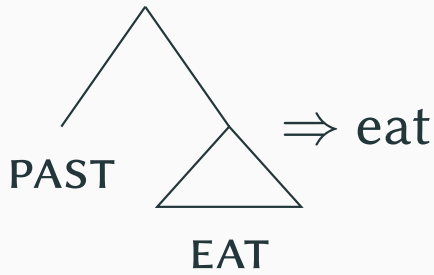
(29) *Lexical items*

a. eat ⇔ [EAT]

b. ate ⇔ [PAST [EAT]]

c. -ed ⇔ [PAST

(30) *Simultaneous C-lexicalization and S-lexicalization*



Past tense: Redundant *ate-d*

(29) *Lexical items*

a. eat ⇔ [EAT]

b. ate ⇔ [PAST [EAT]]

c. -ed ⇔ [PAST

(30) *Simultaneous C-lexicalization and S-lexicalization*

Past tense: Distributive *eat-ed*

(31) *Lexical items*

a. eat ⇔ [EAT]

b. ate ⇔ [PAST [EAT]]

c. -ed ⇔ [PAST

(32) *Neglecting C-lexicalization*

Past tense: Distributive *eat-ed*

(31) *Lexical items*

a. eat ⇔ [EAT]

b. ate ⇔ [PAST [EAT]]

c. -ed ⇔ [PAST

(32) *Neglecting C-lexicalization*

Past tense: Distributive *eat-ed*

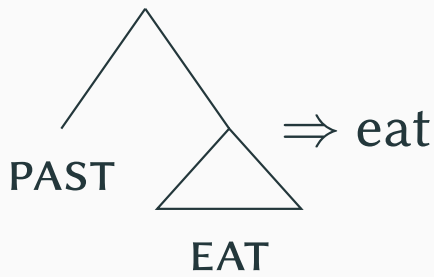
(31) *Lexical items*

a. eat ⇔ [EAT]

b. ate ⇔ [PAST [EAT]]

c. -ed ⇔ [PAST

(32) *Neglecting C-lexicalization*



Past tense: Distributive *eat-ed*

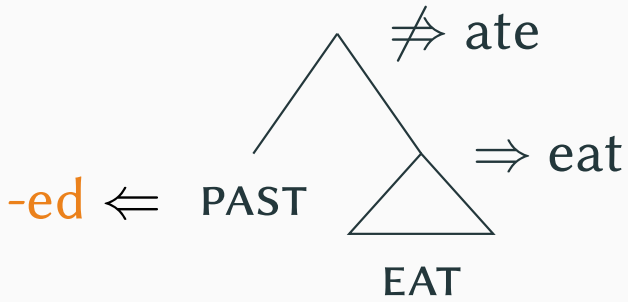
(31) *Lexical items*

a. eat $\Leftrightarrow$ [EAT]

b. ate $\Leftrightarrow$ [PAST [EAT]]

c. -ed $\Leftrightarrow$ [PAST

(32) *Neglecting C-lexicalization*



Nanosyntax: Summary past tense

	failure	**location**	**error type**
ate	none	none	target
eat-ed	no C	2nd cycle	distributive
ate-d	simultaneous C & S	2nd cycle	redundant

What do children do wrong?

- They insert a less specific, i.e. more general exponent
  - if the specificity difference is due to secondary features (DM).
  - if both exponents can be inserted via different modes of lexicalization (Nanosyntax).
- In the domains at hand, they choose an exponent whose insertion is conditioned by only a single feature over/simultaneously with one where it is conditioned by a primary and an additional secondary feature.

Negative concord[a]

[a]Joint work with Cory Bill, Aurore Gonzalez, Ivona Ilić, Paloma Jeretič

Negated indefinites across languages

In the majority of languages, negated indefinites are expressed with a positive indefinite and sentence negation (Kahrel 1996, Miestamo 2007, van der Auwera and Alsenoy 2016, 2018).

(33) *Evenki (Miestamo 2007: 564)*

a. ekun-da ō-ra-n.
something-CLT become-NFUT-3SG
'Something happened.'

b. ekun-da **e-** che o-ra.
something-CLT NEG-PST become-PTCP
'Nothing happened.'

Negated indefinites across languages

In NC languages, negated indefinites are expressed via sentence negation and a morphologically marked negative indefinite, a neg-word/NCI.

(34) Milan **ne** vidi **ništa**. *BCS, (Progovac 1994: 40)*
Milan not sees nothing
'Milan cannot see anything.'

Non-NC languages also use neg-words to express negated indefinites, but without the presence of sentence negation. Adding sentence negation would lead to a double negation reading.

(35) Milan sieht **nichts**. *German*
Milan sees nothing
'Milan cannot see anything.'

Negated indefinites

(roughly) three grammars:

type 1 NEG … positive indefinite (e.g., Evenki)

type 2 NEG … negative indefinite (e.g., BCS)

type 3 ∅ … negative indefinite (e.g., German)

Negative concord errors: Some examples

(36) a. We do n't want **no** gas. (Adam 3;11, Brown 1973)

b. **No** tigers do n't bit you? (Mark 2;08, MacWhinney 1991)

c. I do n't care about **nothing**. (Ross 5;04, MacWhinney 1991)

d. He wo n't hurt his head **never**. (Eleanor 2;11, Lieven et al. 2009)

e. **No** one's not drying him, mum. (Fraser 3;00, Lieven et al. 2009)

(37) a. **Kein** Gewitter kommt nicht heute. (Leo 2;03, Behrens 2006)
no thunderstorm comes not today
'There's no thunderstorm coming today.'

b. Wir haben noch **keine** Zudecke nich. (Simone 3;07, Miller 1979)
we have yet no duvet not
'We don't have a duvet yet.'

c. **Kein** Teller kann s net sein. (Sebastian 5;04, Lieven and Stoll 2013)
no plate can it not be
'It can't be a plate.'

Goals

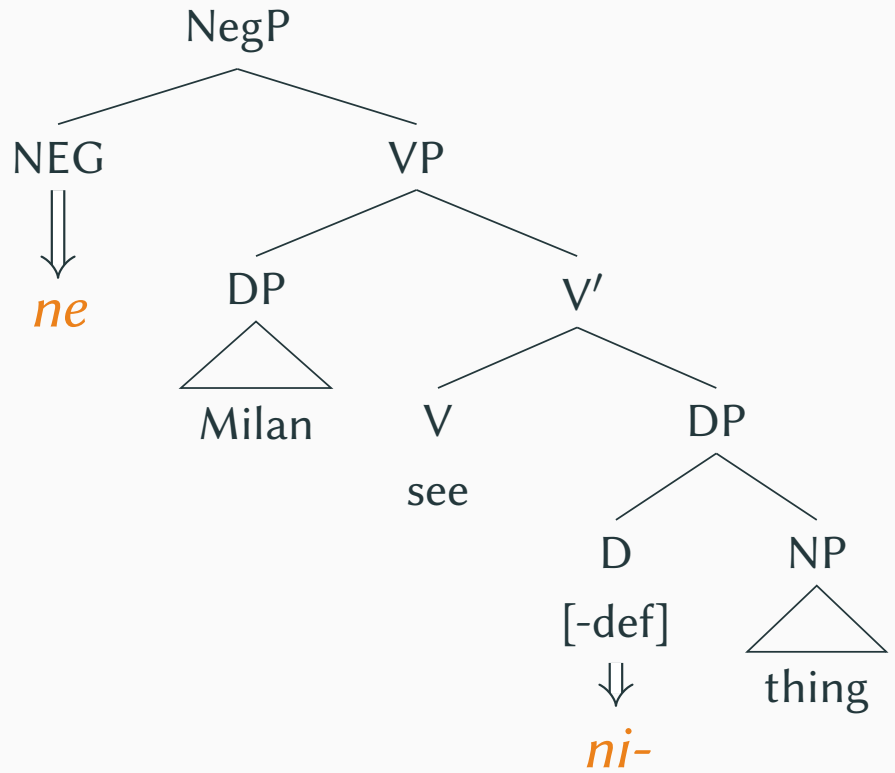
1. Derive adult typology, i.e. three language types.
2. Account for negative concord errors in the acquisition of non-negative concord languages like English and German.
3. Account for errors of the form: NEG ... positive indefinite.
4. Account for omission errors in the acquisition of negative concord languages like BCS, Italian etc.

An allomorphy account of negative concord?

(38) Milan **ne** vidi **ništa**.
Milan not sees nothing
'Milan cannot see anything.'

(39)
a. /∅/ ⇔ [NEG]
b. /nešto/ ⇔ [-def]
c. /ništa/ ⇔ [-def] / NEG ... __
d. /ne/ ⇔ [NEG] / __ ... [-def]

(40)



Locality issue!

Meaning First architecture (Sauerland and Alexiadou 2020)

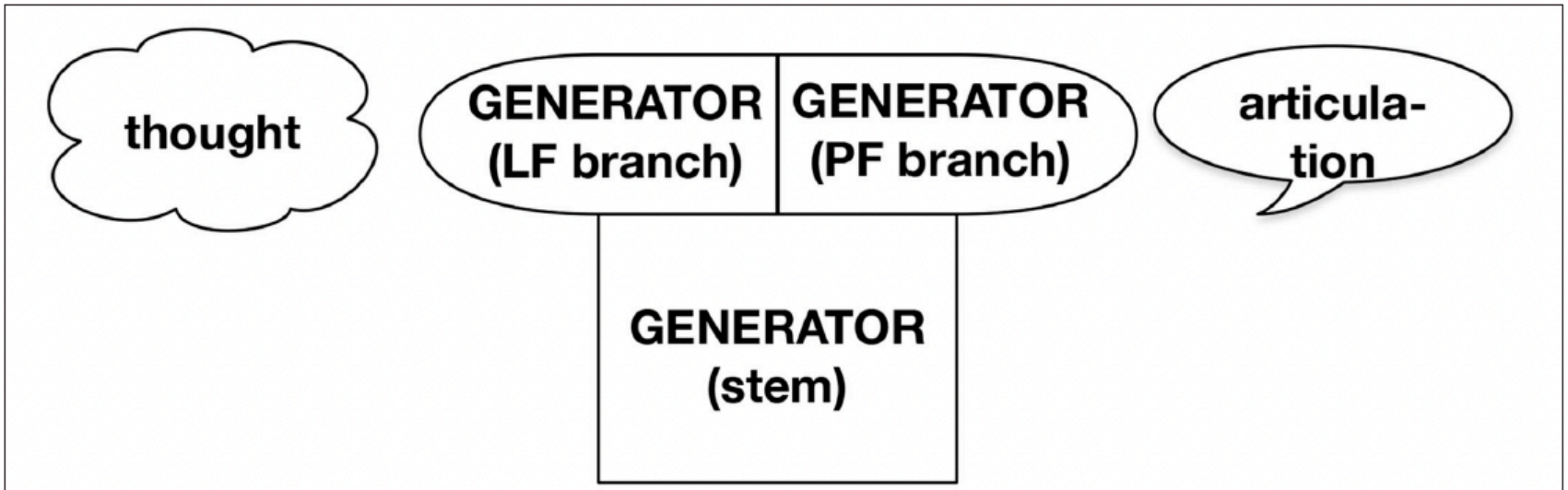


FIGURE 2 | The T-model of language assumes that structural representations are generated within language and produce a thought-articulation pair.

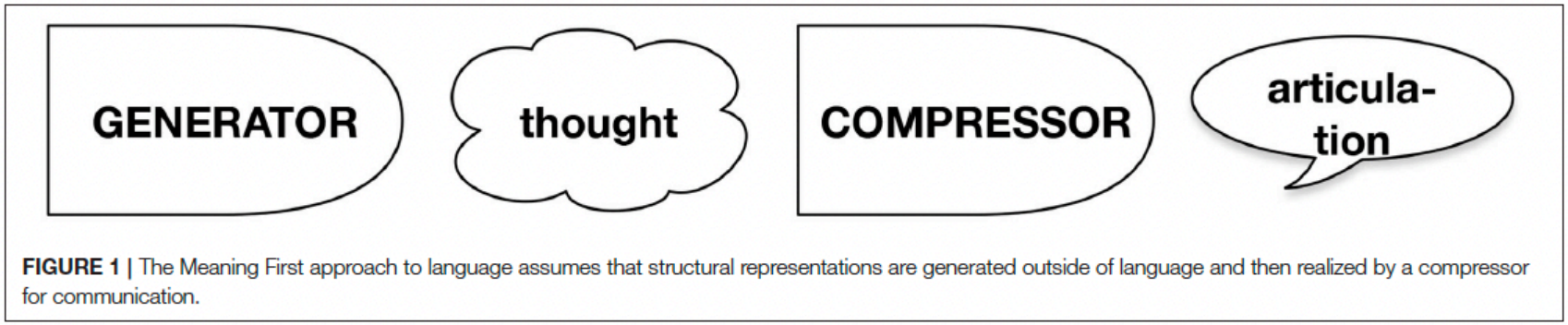


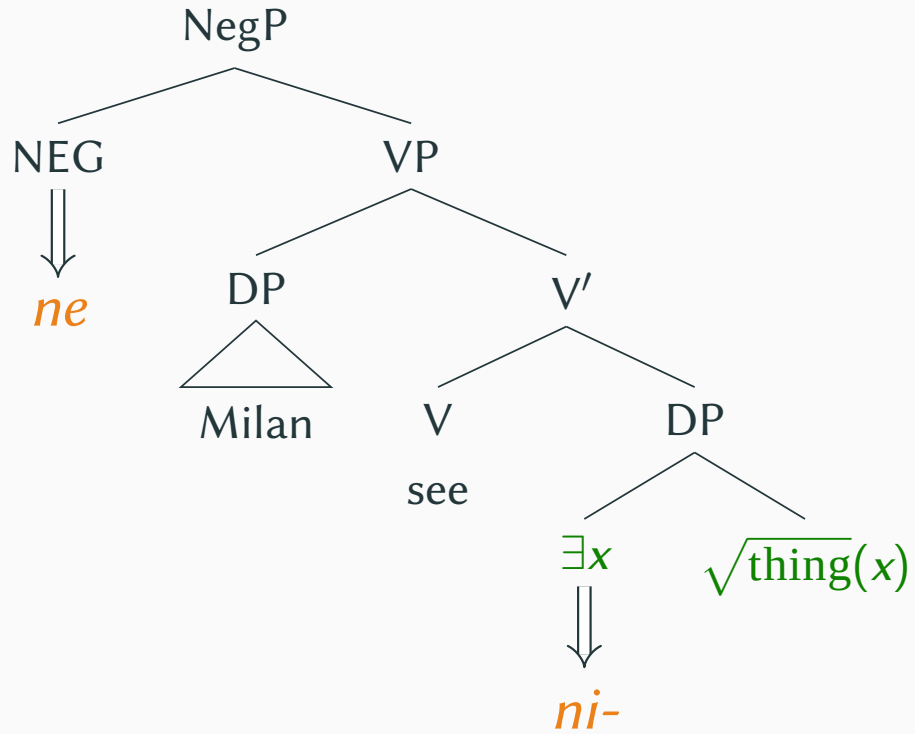
FIGURE 1 | The Meaning First approach to language assumes that structural representations are generated outside of language and then realized by a compressor for communication.

An allomorphy account of negative concord?

(41) Milan **ne** vidi **ništa**.
Milan not sees nothing
'Milan cannot see anything.'

(42)
a. /∅/ ⇔ [NEG]
b. /nešto/ ⇔ [∃]
c. /ništa/ ⇔ [∃] / NEG ... ___
d. /ne/ ⇔ [NEG] / ___ ... ∃

(43)



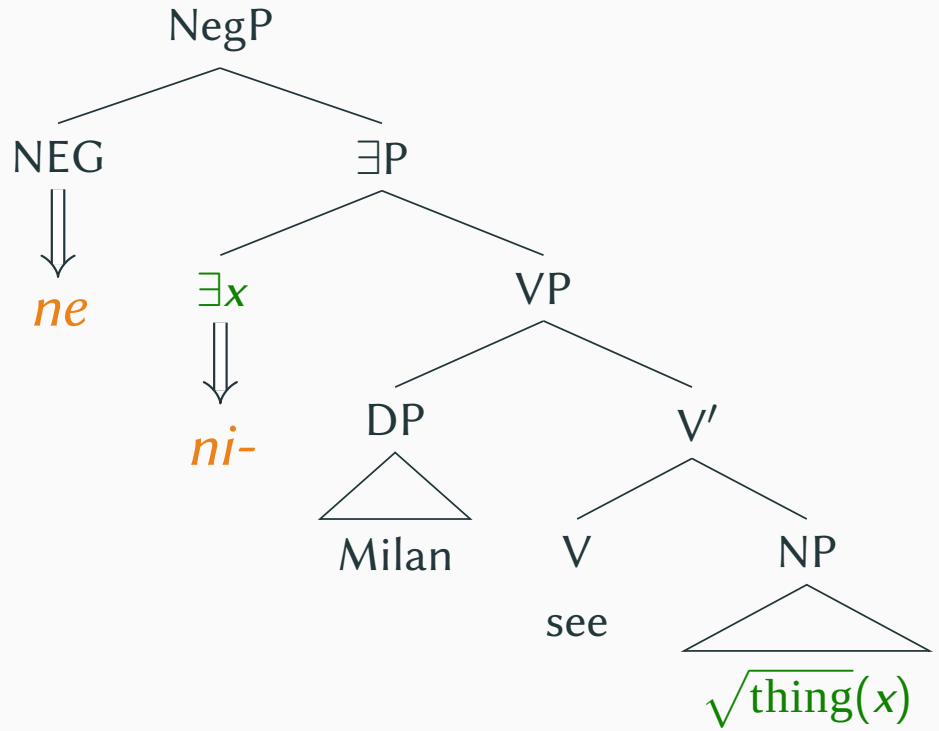
Locality issue!

An allomorphy account of negative concord?

(44) Milan **ne** vidi **ništa**.
Milan not sees nothing
'Milan cannot see anything.'

(45)
a. /∅/ ⇔ [NEG]
b. /nešto/ ⇔ [∃]
c. /ništa/ ⇔ [∃] / NEG [__
d. /ne/ ⇔ [NEG] / __ [∃

(46)



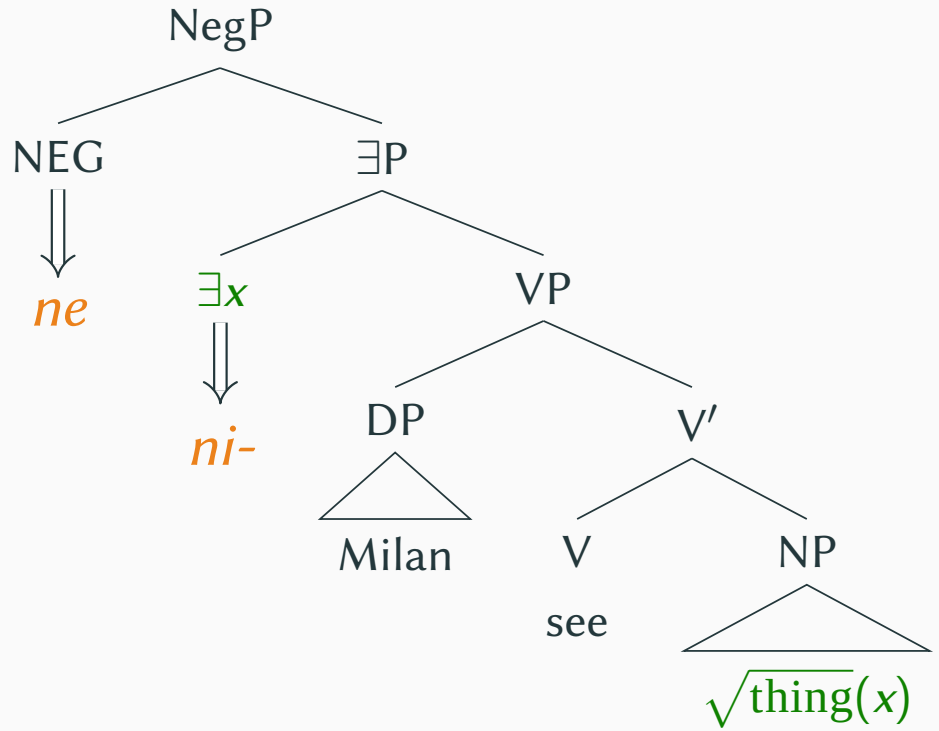
Following Heim (1982), Reinhart (1997), Winter (1997), Kratzer (1998) etc.

An allomorphy account of negative concord?

(47) Milan **ne** vidi **ništa**.
Milan not sees nothing
'Milan cannot see anything.'

(48)
a. /∅/ ⇔ [NEG]
b. /nešto/ ⇔ [∃]
c. /ništa/ ⇔ [∃] / NEG [__
d. /ne/ ⇔ [NEG] / __ [∃

(49)



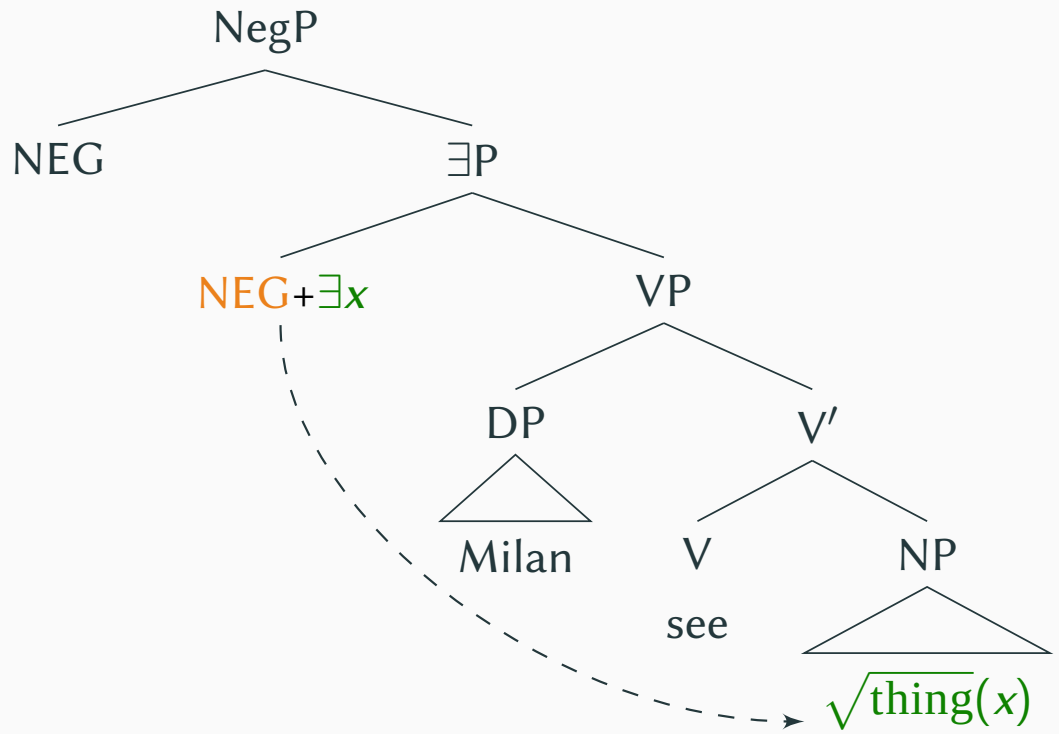
New issue: Linearization!

Alternative: Redundancy via a duplication rule

(50) Milan **ne** vidi **ništa**.
Milan not sees nothing
'Milan cannot see anything.'

(51) Duplication rule:
$\varnothing \rightarrow$ NEG / NEG [__ $\exists$
(cf. Müller 2007)

(52)

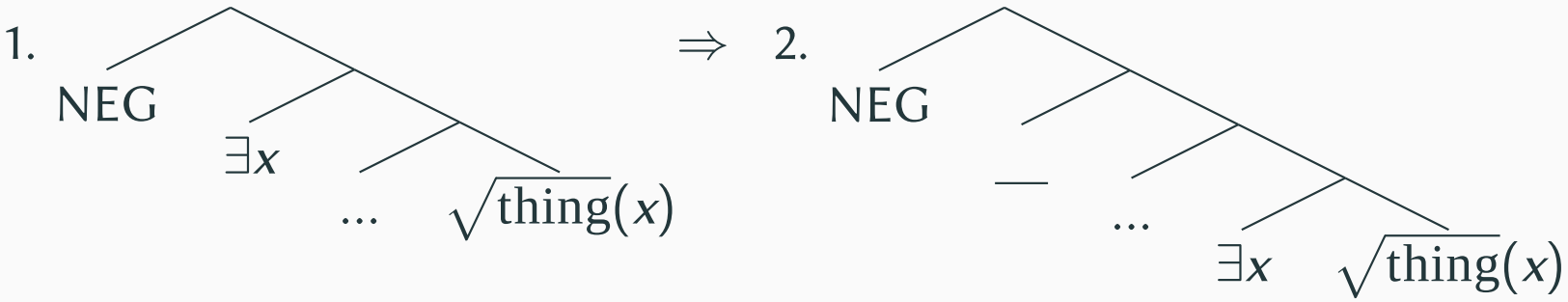


Assumption: Linearization can make reference to semantic dependencies (compatible with Meaning First framework).

Type 1 grammar

(53) a. ekun-da ō-ra-n. *Evenki*
something-CLT become-NFUT-3SG
'Something happened.'

b. ekun-da **e-** che o-ra.
something-CLT NEG-PST become-PTCP
'Nothing happened.'

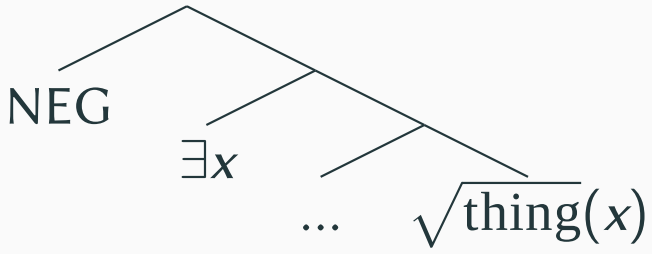


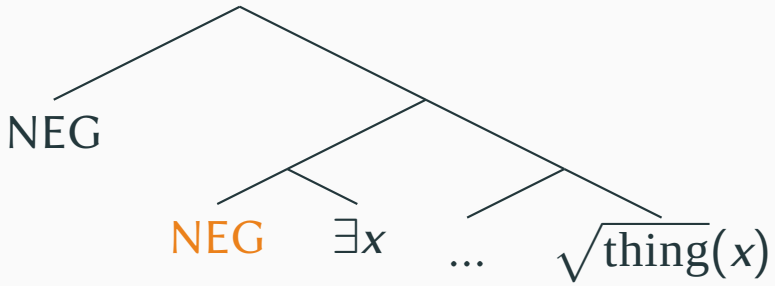
1. Base structure
2. Linearization
3. Vocabulary insertion

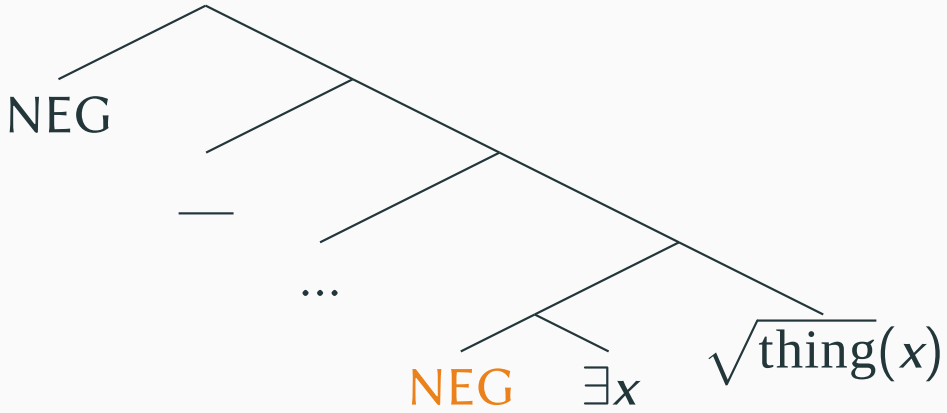
Type 2: A negative concord grammar

(54) Milan **ne** vidi **ništa**.
Milan not sees nothing
'Milan cannot see anything.'

(55) Dupl: $\varnothing \rightarrow$ NEG / NEG [__ $\exists$

1.


2.


3.


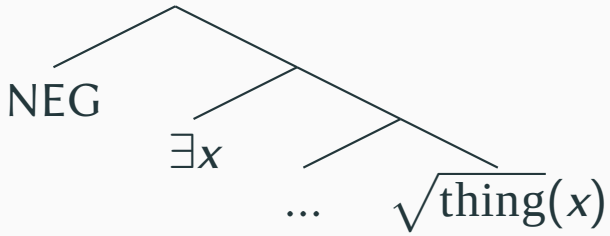
1. Base structure
2. Duplication
3. Linearization
4. Vocabulary insertion

Type 3: A non-negative concord grammar

(56) Milan sieht **nichts**.
Milan sees nothing
'Milan cannot see anything.'

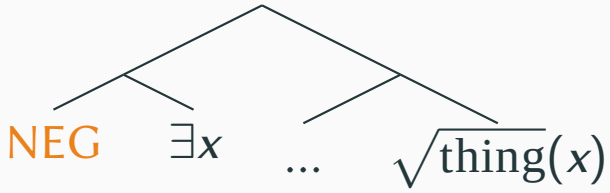
(57) Dupl: $\varnothing \rightarrow$ NEG / NEG [__ $\exists$
Del: NEG $\rightarrow \varnothing$ / __ [NEG $\exists$

1.

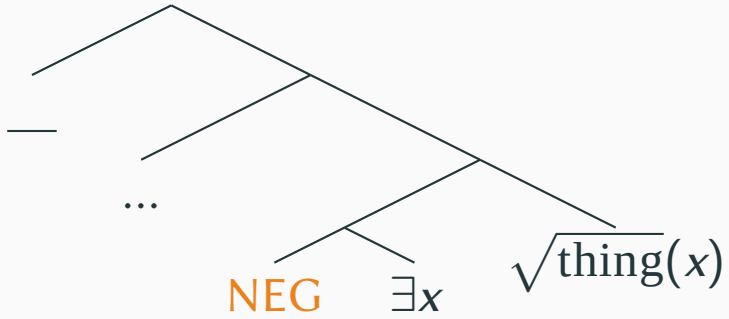


2.

3.



4.



1. Base structure
2. Duplication
3. Deletion
4. Linearization
5. Vocabulary insertion

What do children do wrong?

1. ~~Derive adult typology, i.e. three language types.~~
2. Account for negative concord errors in the acquisition of non-negative concord languages like English and German.
   → **Children have not acquired the deletion rule.**
3. Account for errors of the form: NEG ... positive indefinite.
   → **Neither a duplication nor a deletion rule is acquired yet.**
4. Account for omission errors in the acquisition of negative concord languages like BCS, Italian etc.
   → **Children wrongfully postulate a deletion rule?**

Summary

Duplication: $\varnothing \longrightarrow$ NEG / NEG [__ $\exists$

Deletion: NEG $\longrightarrow \varnothing$ / __ [NEG $\exists$

Grammars:

1. NEG ... positive indefinite
    - Low
2. NEG ... negative indefinite
    - Dupl$\prec$Low
3. $\varnothing$ ... negative indefinite
    - Dupl$\prec$Del$\prec$Low

Errors children make:

1. NEG ... positive indefinite
    - no errors predicted
2. NEG ... negative indefinite
    - type 1: Low
    - type 3: Dupl$\prec$Del$\prec$Low
3. $\varnothing$ ... negative indefinite
    - type 1: Low
    - type 2: Dupl$\prec$Low

Conclusion

- Redundant commission errors are attested in many domains in child language.
- They can be modelled in a variety of ways
    - Allomorphy
    - Multiple insertion
    - Doubling

Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 856421).

References i

Abels, K. and Muriungi, P. K. (2008). The focus particle in Kîîtharaka: Syntax and semantics. *Lingua*, 118:687–731.

Aksu-Koç, A. and Slobin, D. (1985). The acquisition of Turkish. In *The Cross-linguistic Studies of Language Acquisition. Vol. 1: The Data*, pages 839–876. Lawrence Erlbaum Associates, Hillsdale.

Alexiadou, A., Guasti, M.-T., and Sauerland, U. (2021). A meaning first approach through a kids first view. Talk given at the University Milano-Bicocca, 17 March 2021.

Behrens, H. (2006). The input-output relationship in first language acquisition. *Language and Cognitive Processes*, 21:2–24.

Bezinska, Y., Chevrot, J.-P., and Novakova, I. (2008). Le prédicat complexe *faire+ Vinf* dans le langage enfantin entre 4 et 6 ans. In *Congrès Mondial de Linguistique Française*, page 157. EDP Sciences.

Brown, R. (1973). *A first language: The early stages*. Harvard University Press, Cambridge, MA.

Caha, P. (2009). *The Nanosyntax of Case*. PhD thesis, University of Tromsø.

Carstairs, A. (1987). *Allomorphy in inflexion*. Croom Helm, London.

Cohen, M. (1925). Sur les langages successifs de l'enfant. In *Mélanges linguistiques offerts à M.J. Vendryès par ses amis et ses élèves*, pages 109–127. Champion, Paris.

Corver, N. (2005). Double comparatives and the comparative criterion. *Recherches linguistiques de Vincennes*, 34:165–190.

Demuth, K. and Tremblay, A. (2008). Prosodically-conditioned variability in children's production of french determiners. *Journal of child language*, 35(1):99–127.

Family, N. and Allen, S. E. M. (2015). The development of the causative construction in Persian child language. *Journal of Child Language*, 42(6):1337–1378.

Hall, W. S., Nagy, W. E., and Linn, R. T. T. (1984). *Spoken words: Effects of situation and social group on oral word usage and frequency*. Erlbaum, Hillsdale, NJ.

References ii

Halle, M. (1997). Distributed morphology: Impoverishment and fission. In Bruening, B., Kang, Y., and McGinnis, M., editors, *Papers at the Interface*, volume 30 of *MIT Working Papers in Linguistics*, pages 425–449. MITWPL, Cambridge, Mass.

Halle, M. and Marantz, A. (1993). Distributed Morphology and the Pieces of Inflection. In Hale, K. and Keyser, S. J., editors, *The View from Building 20*, pages 111–176. MIT Press, Cambridge Massachusetts.

Halle, M. and Marantz, A. (1994). Some key features of distributed morphology. In Carnie, A., Harley, H., and Bures, T., editors, *Papers on Phonology and Morphology*, volume 21 of *MIT Working Papers in Linguistics*, pages 275–288. MITWPL, Cambridge, Mass.

Heim, I. (1982). *The Semantics of Definite and Indefinite Noun Phrases*. PhD thesis, University of Massachusetts.

Hein, J., Bill, C., Driemel, I., Gonzalez, A., Ilić, I., Jeretič, P., and Guasti, M. T. (2022). Negative concord in the acquisition of non-negative concord languages. Talk presented at *CLS 58*, The University of Chicago, April 2022.

Kahrel, P. (1996). *Aspects of negation*. PhD thesis, University of Amsterdam.

Kratzer, A. (1998). Scope or Pseudoscope? Are there Wide Scope Indefinites? In Rothstein, S., editor, *Events and Grammar*, pages 163–196. Kluwer, Dordrecht.

Kuczaj, S. A. (1977). The acquisition of regular and irregular past tense forms. *Journal of Verbal Learning and Verbal Behavior*, 16(5):589–600.

Kuczaj, S. A. (1978). Children's judgments of grammatical and ungrammatical irregular past-tense verbs. *Child Development*, 49(2):319–326.

Le Normand, M.-T. (1986). A developmental exploration of language used to accompany symbolic play in young, normal children (2–4 years old). *Child: care, health and development*, 12(2):121–134.

Lieven, E., Salomo, D., and Tomasello, M. (2009). Two-year-old children's production of multiword utterances: A usage-based analysis. *Cognitive Linguistics*, 20(3):481–508.

References iii

Lieven, E. and Stoll, S. (2013). Early communicative development in two cultures. *Human Development*, 56:178–206.

Lord, C. (1979). "don't you fall me down": children's generalizations regarding cause and transitivity. In *Papers and Report on Child Language Development*, volume 17, pages 81–89. Stanford University.

MacWhinney, B. (1991). *The CHILDES project: Tools for analyzing talk*. Erlbaum, Hillsdale, NJ.

Marcus, G. F., Pinker, S., Ullman, M., Hollander, M., Rosen, T. J., Xu, F., and Clahsen, H. (1992). Overregularization in language acquisition. *Monographs of the society for research in child development*, pages i–178.

Martin, F., Nie, Y., Alexiadou, A., and Guasti, M.-T. (2021). Wearing causation on its sleeve: Overt CAUSE in child french causatives. Talk presented at *BUCLD 46*, Boston University (online), November 2021.

Menn, L. and MacWhinney, B. (1984). The repeated morph constraint: Towards an explanation. *Language*, 3:519–641.

Meyer, M.-C., Yatsushiro, K., and Sauerland, U. (2021). Cum-sine patterns. Talk presented at the LeibnizDream Retreat, September 20, 2021, Humboldt-Universität zu Berlin/ZAS.

Miestamo, M. (2007). Negation: an overview of typological research. *Language and Linguistics Compass*, 1/5:552–570.

Miller, M. (1979). *The Logic of Languages Development in Early Childhood*. Springer-Verlag, Berlin.

Moline, R. (1971). Recensement des tournures incorrectes. *Repères pour la rénovation de l'enseignement du français à l'école élémentaire*, 10–11:7–29.

Morgenstern, A., Benazzo, S., Leroy, M., Mathiot, E., C., P., and Sekali, M. (2009). *L'enfant dans la langue. De l'observation du naturaliste à l'analyse du linguiste*. Presses de la Sorbonne Nouvelle, Paris.

Müller, G. (2007). Extended Exponence by Enrichment. Argument Encoding in German, Archi, and Timucua. In Scheffler, T., Tauberer, J., Eilam, A., and Mayol, L., editors, *Proceedings of the 30th Annual Penn Linguistics Colloquium*, pages 253–266. University of Pennsylvania, Philadelphia.

References iv

Müller, G. (2020). *Inflectional Morphology in Harmonic Serialism*. Advances in Optimality Theory. Equinox, Sheffield.

Nicolae, A. C. and Yatsushiro, K. (2020). Not eating kein veggies: negative concord in child German. Linguistic Evidence 2020, Proceedings.

Noyer, R. (1997). *Features, Position and Affixes in Autonomous Morphological Structure*. Garland, New York.

Palasis, K. (2009). *Syntaxe générative et acquisition: le sujet dans le développement du système linguistique du jeune enfant*. PhD thesis, Nice.

Progovac, L. (1994). *Negative and positive polarity: a binding approach*. CUP.

Reinhart, T. (1997). Quantifier scope: How labor is divided between QR and choice functions. *Linguistics and Philosophy*, 20:335–397.

Sauerland, U. (2019). Compression within a generative view of thought and language. Talk given at Harvard University, 27 September 2021.

Sauerland, U. and Alexiadou, A. (2020). Generative grammar: A meaning first approach. *Frontiers in Psychology*, 11:3104.

Starke, M. (2009). Nanosyntax: A short primer to a new approach to language. *Nordlyd*, 36(1):1–6.

Stemberger, J. (1982). *The lexicon in a model of language production*. PhD thesis, University of California, San Diego.

Stump, G. (2001). *Inflectional Morphology*. Cambridge University Press, Cambridge.

Svenonius, P. (2012). Spanning. ling.auf.net/lingBuzz/001501.

Svenonius, P. (2016). Spans and words. In Siddiqi, D. and Harley, H., editors, *Morphological Metatheory*, pages 201–222. John Benjamins, Amsterdam/Philadelphia.

Taraldsen, K. T. (2010). The nanosyntax of Nguni noun class prefixes and concord. *Lingua*, 120:1522–1548.

van der Auwera, J. and Alsenoy, L. V. (2016). On the typology of negative concord. *Studies in Language*, 40:473–512.

References v

van der Auwera, J. and Alsenoy, L. V. (2018). More ado about nothing: On the typology of negative indefinites. In und Laurence Horn, K. P. T., editor, *Pragmatics, Truth and Underspecification*, pages 107–146. Brill, Leiden.

Williams, E. (2003). *Representation Theory*. MIT Press, Cambridge, MA.

Winter, Y. (1997). Choice functions and the scopal semantics of indefinites. *Linguistics and Philosophy*, 20:399–467.

Yamakoshi, K., Miura, K., Jorinbo, H., Angata, K., and Yamasaki, K. (2018). An experimental study of children's comprehension of lexical and productive causatives in japanese. In *Topics in Theoretical Asian Linguistics: Studies in honor of John B. Whitman*, pages 229–250. John Benjamins, Amsterdam & Philadelphia.

Appendix

Type 2: Single negation with two indefinites — Dupl≺Low

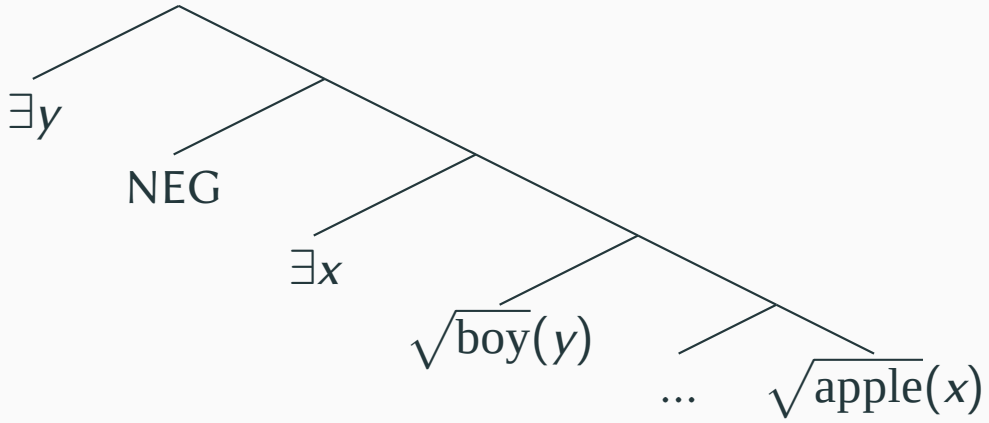
Semantic structure:

$\exists$ NEG $\exists$ … $\sqrt{\text{apple}}$ $\sqrt{\text{boy}}$

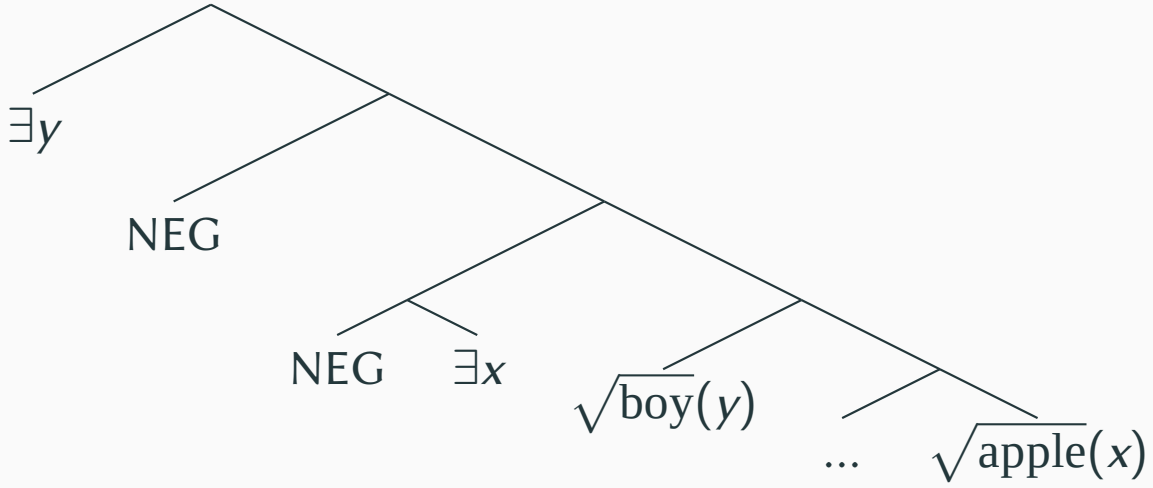
(58) Dupl: $\varnothing \rightarrow$ NEG / NEG [__ $\exists$

(59) A boy didn't eat no apple.

1.



2.

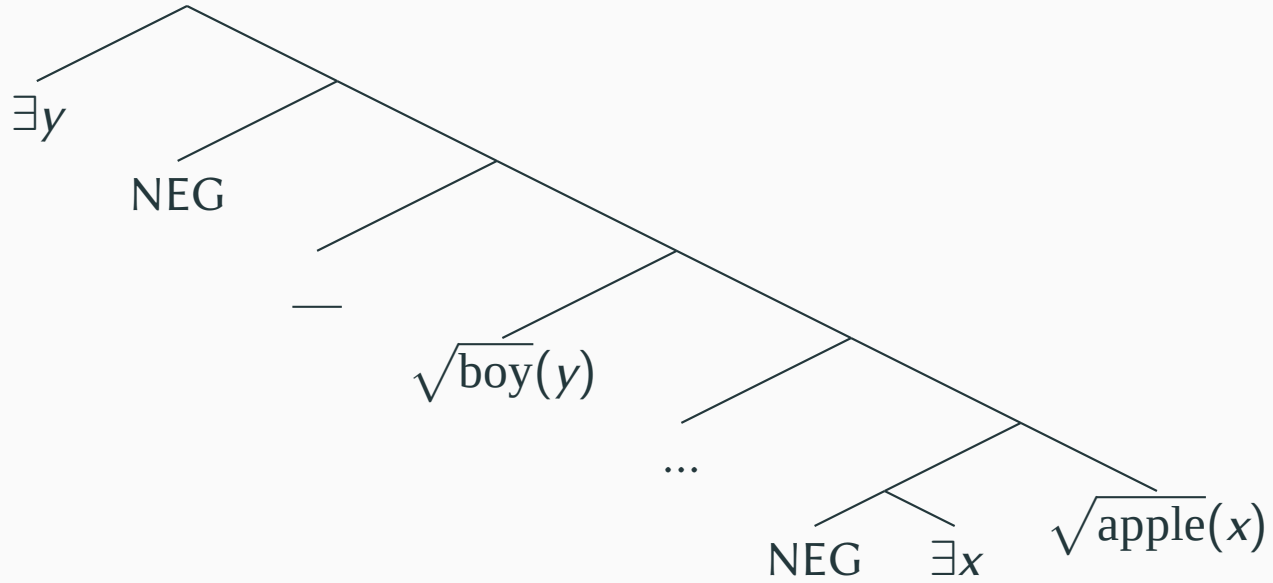


Type 2: Single negation with two indefinites

Semantic structure:
$\exists$ NEG $\exists$ … $\sqrt{\text{apple}}$ $\sqrt{\text{boy}}$

(60) Dupl: $\varnothing \rightarrow$ NEG / NEG [__ $\exists$
(61) A boy didn't eat no apple.

3.



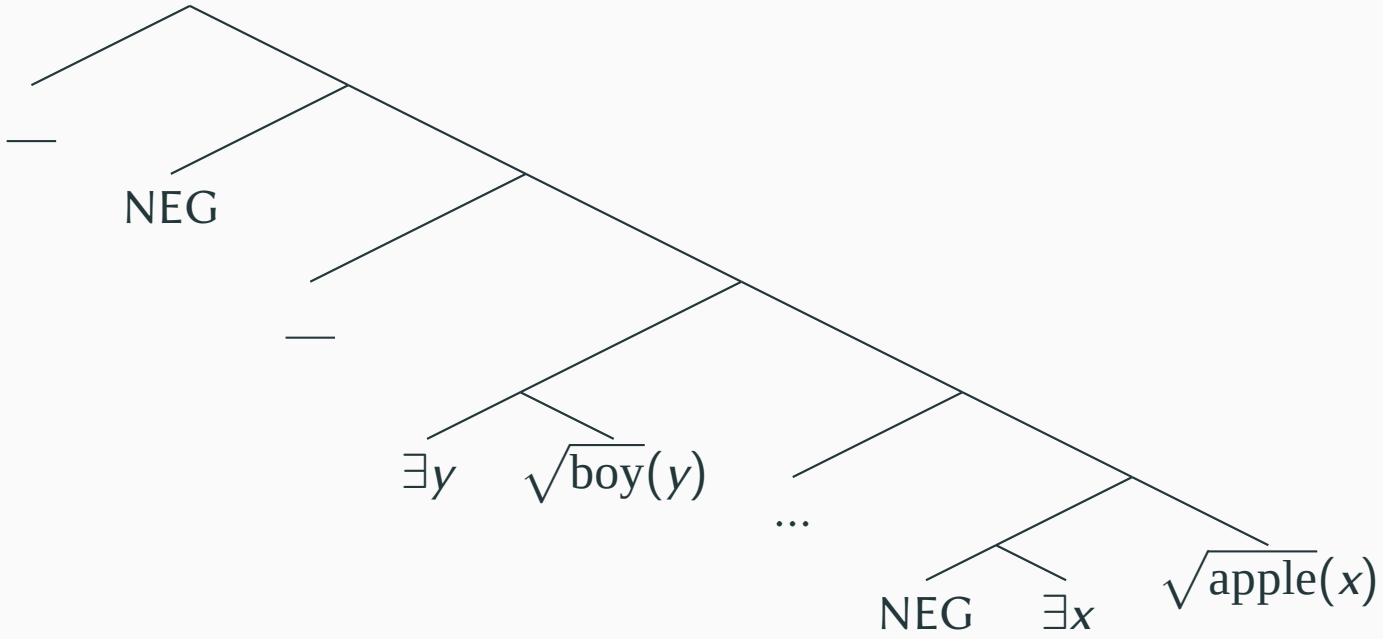
Type 2: Single negation with two indefinites — Dupl≺Low

Semantic structure:
$\exists$ NEG $\exists$ ... $\sqrt{\text{apple}}$ $\sqrt{\text{boy}}$

(62) Dupl: $\varnothing \rightarrow$ NEG / NEG [__ $\exists$
(63) A boy didn't eat no apple.

4. Dupl does not apply

5.

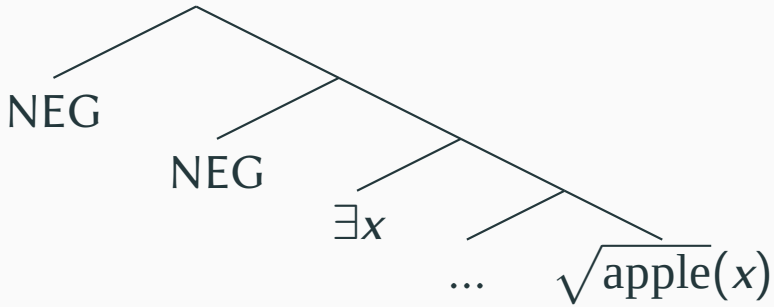


Output:
sentence negation + 1 NCI (single negation reading)

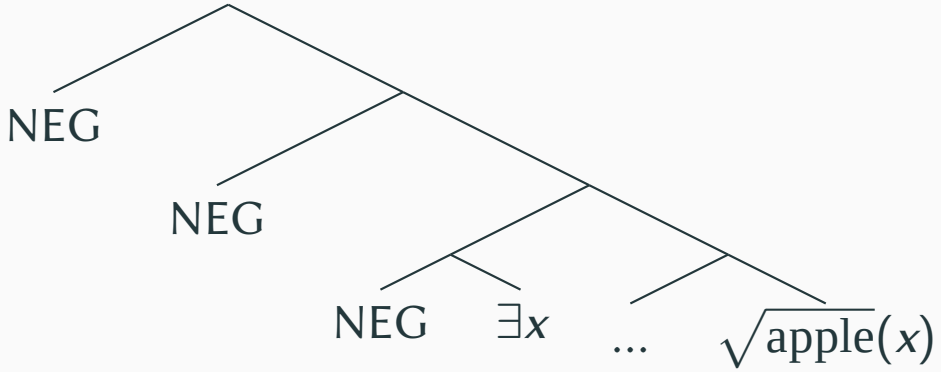
Type 2: Double negation

Semantic structure:
NEG NEG ∃ … $\sqrt{\text{apple}}$

1.



2.



(64) Dupl: ∅ → NEG / NEG [__ ∃

(65) Mary didn't not eat no apple.

Type 2: Double negation

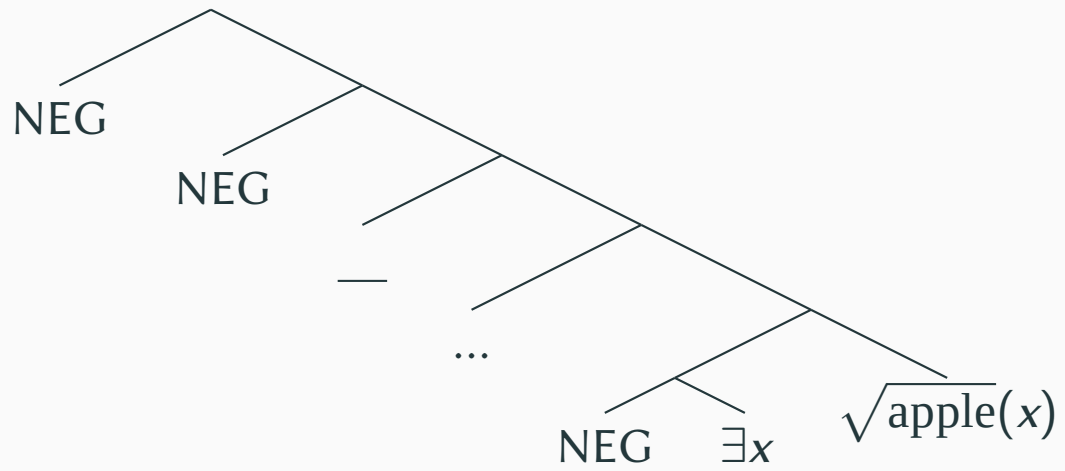
Semantic structure:

NEG NEG ∃ … $\sqrt{\text{apple}}$

(66) Dupl: ∅ → NEG / NEG [__ ∃

(67) Mary didn't not eat no apple.

3.



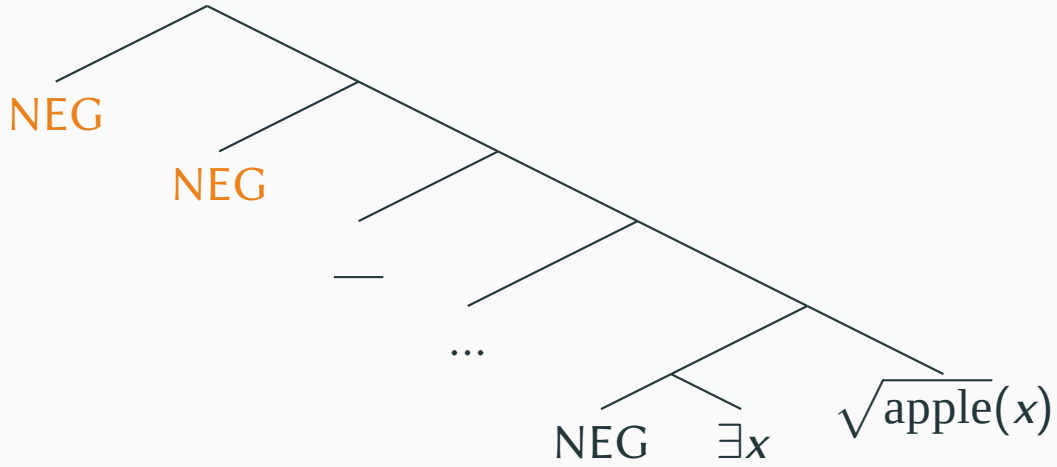
Output:

2 x sentence negation + 1 NCI (double negation reading)

Semantic structure:
NEG NEG ∃ … $\sqrt{\text{apple}}$

(68) Dupl: ∅ → NEG / NEG [__ ∃
(69) Mary didn't not eat no apple.

Output:



Idea:
The output triggers an OCP effect (prohibition of adjacent identical elements).

Ways out of the OCP:
fragments answers (no pronounciation of NEG NEG); ineffability (leads to alternative bi-clausal structures); pitch contour resulting from haplology (very speculative); morphologically distinct exponents of NEG

Type 3: Single negation with two indefinites Dupl≺Del≺Low

Semantic structure:

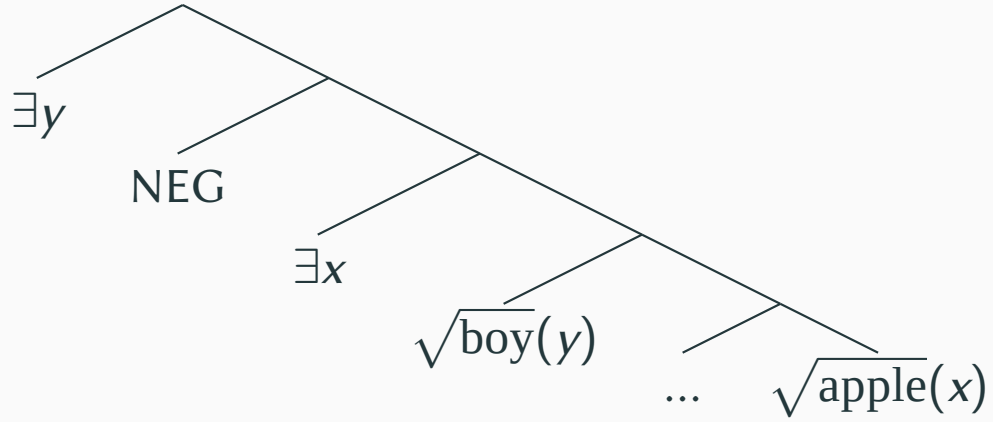
$\exists$ NEG $\exists$ … $\sqrt{\text{apple}}$ $\sqrt{\text{boy}}$

(70) Dupl: $\varnothing \rightarrow$ NEG / NEG [__ $\exists$

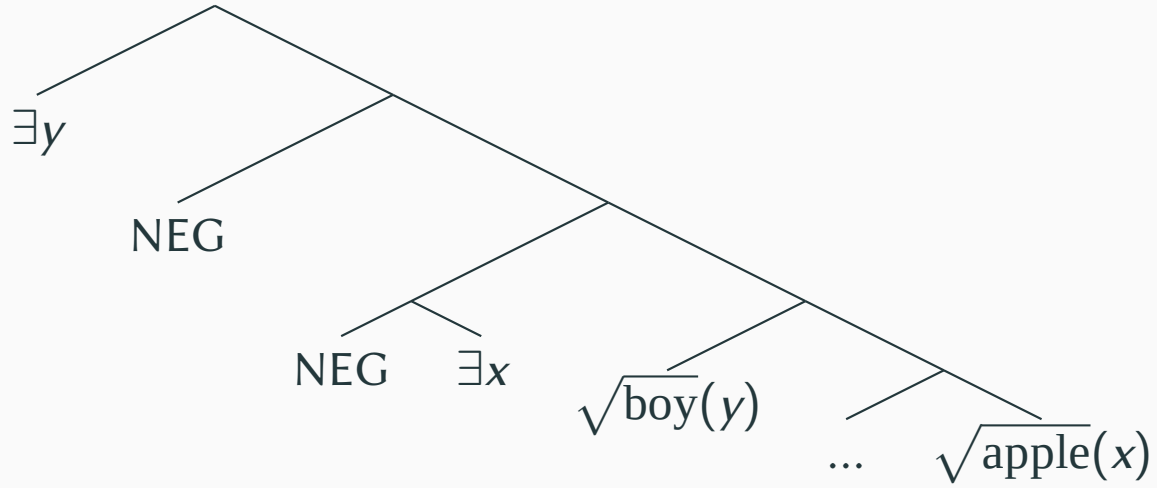
(71) Del: NEG $\rightarrow \varnothing$ / __ [NEG $\exists$

(72) A boy ate no apple.

1.



2.

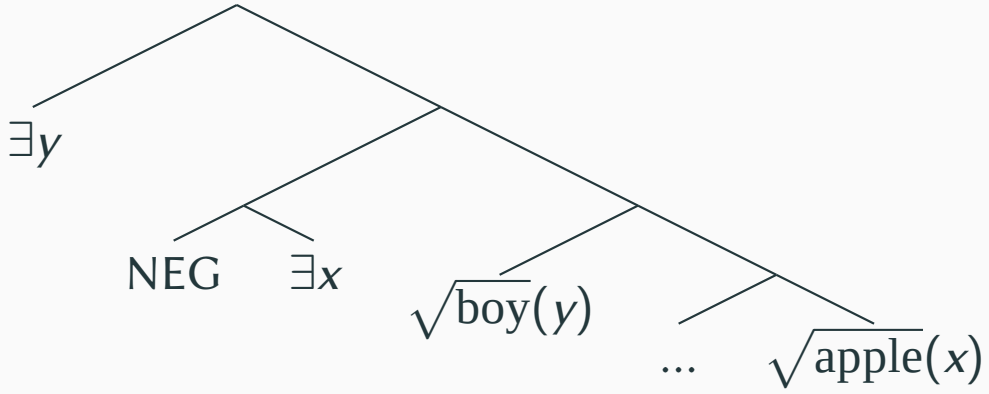


Type 3: Single negation with two indefinites Dupl≺Del≺Low

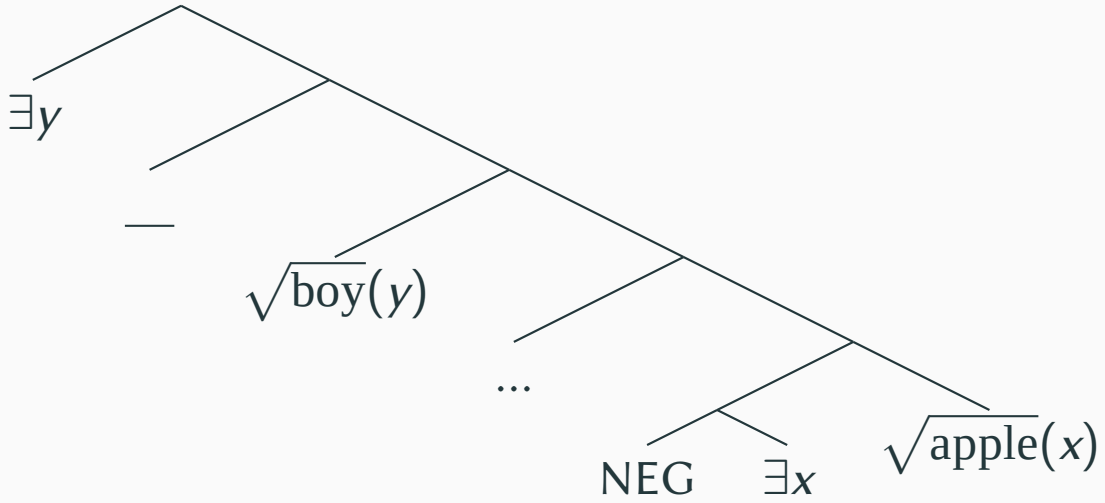
Semantic structure:
∃ NEG ∃ … $\sqrt{\text{apple}}$ $\sqrt{\text{boy}}$

(73) Dupl: ∅ → NEG / NEG [__ ∃
(74) Del: NEG → ∅ / __ [NEG ∃
(75) A boy ate no apple.

3.



4.



Type 3: Single negation with two indefinites Dupl≺Del≺Low

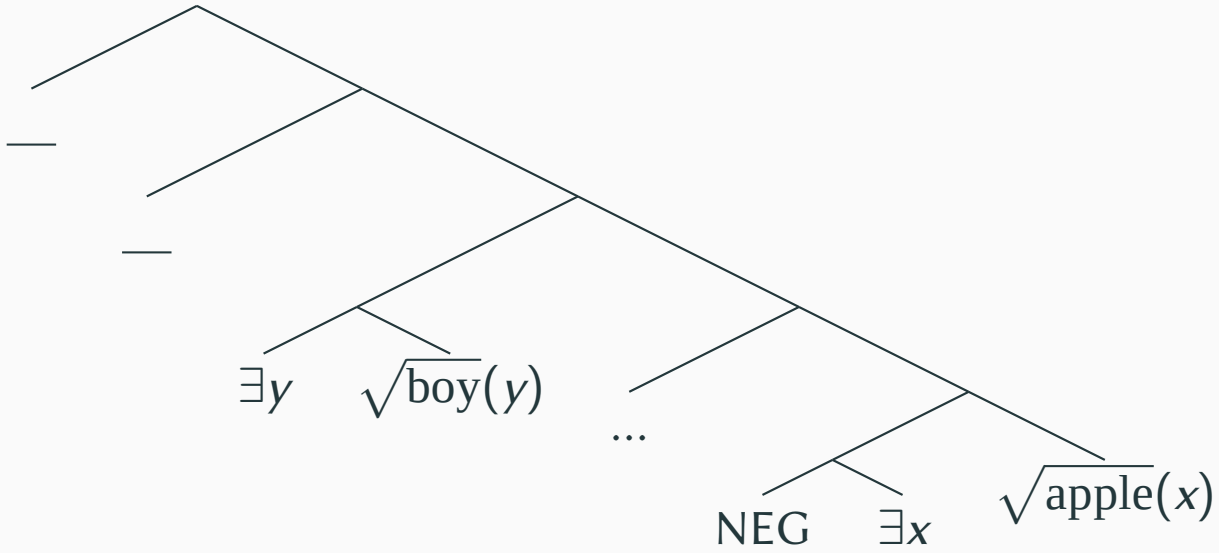
Semantic structure:
$\exists$ NEG $\exists$ ... $\sqrt{\text{apple}}$ $\sqrt{\text{boy}}$

(76) Dupl: $\varnothing \rightarrow$ NEG / NEG [__ $\exists$
(77) Del: NEG $\rightarrow \varnothing$ / __ [NEG $\exists$
(78) A boy ate no apple.

5. Dupl does not apply

6. Del does not apply

7.



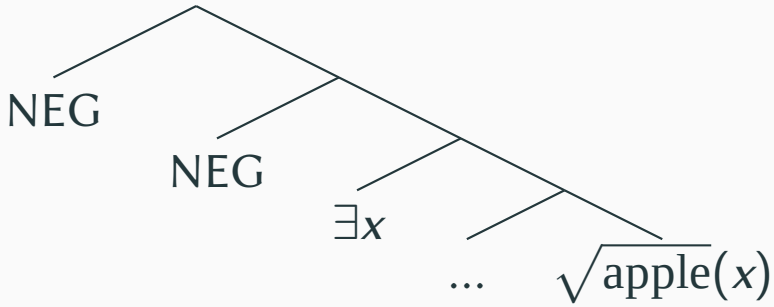
Output:
1 NCI, 1 indefinite (single negation reading)

Type 3: Double negation

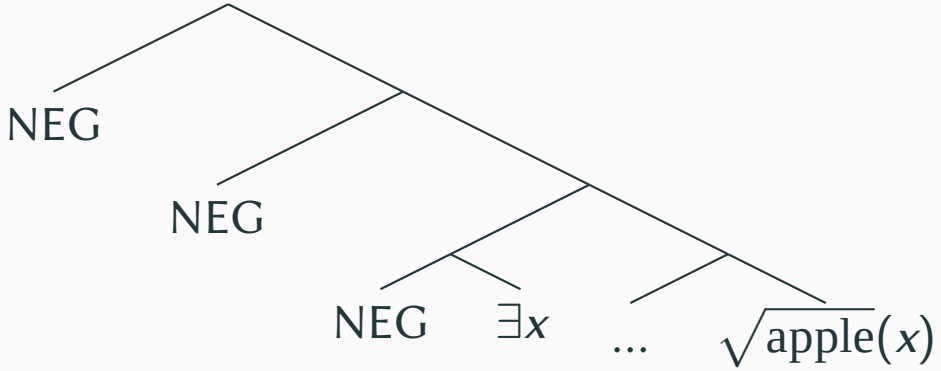
Semantic structure:
NEG NEG ∃ ... $\sqrt{\text{apple}}$

(79) Dupl: ∅ → NEG / NEG [__ ∃
(80) Del: NEG → ∅ / __ [NEG ∃
(81) Mary didn't eat no apple.

1.



2.



Type 3: Double negation

Dupl≺Del≺Low

Semantic structure:

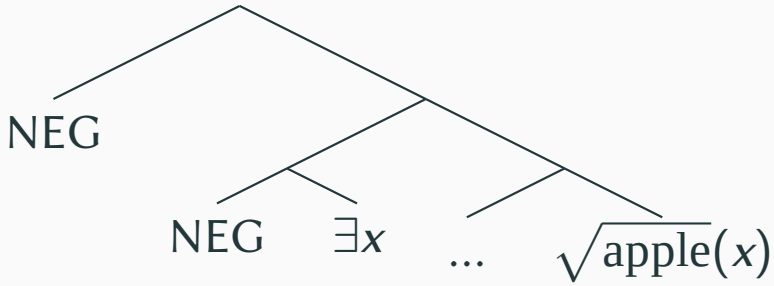
NEG NEG ∃ … $\sqrt{\text{apple}}$

(82) Dupl: ∅ → NEG / NEG [__ ∃

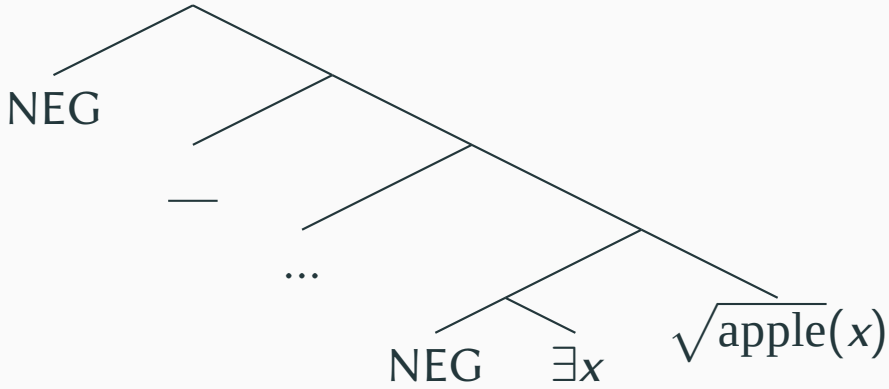
(83) Del: NEG → ∅ / __ [NEG ∃

(84) Mary didn't eat no apple.

3.



4.



Output:

sentence negation + 1 NCI (double negation reading)