

One analysis of three Mandarin ‘two’s

Marcin Wągiel & Pavel Caha

GLOW in Asia 13, August 4–7, 2022

Introduction

Data

Structures

Spellout

Conclusions

Numerals

Two functions of numerals

Bultinck (2005), Rothstein (2013, 2017)

Numerals

Two functions of numerals

Bultinck (2005), Rothstein (2013, 2017)

- ▶ **abstract counting** ⇒ arithmetical use
 - ▶ reference to a number concept
- (1) a. Ten divided by **five** equals two.
 b. **Five** is a Fibonacci number.

Numerals

Two functions of numerals

Bultinck (2005), Rothstein (2013, 2017)

- ▶ **abstract counting** ⇒ arithmetical use
 - ▶ reference to a number concept
- (1) a. Ten divided by **five** equals two.
 b. **Five** is a Fibonacci number.
-
- ▶ **object counting** ⇒ quantifying use
 - ▶ quantification over entities
- (2) a. **five** cats
 b. the **five** girls

The relationship between the functions

Object counting → abstract counting

Ionin & Matushansky (2006, 2018), Rothstein (2013, 2017), Kennedy (2015)

- ▶ the object-counting meaning is basic
- ▶ the abstract-counting meaning is derived

$$(3) \quad [[\text{three}_{\text{Ac}}]] = [[[\alpha [\text{three}_{\text{oc}}]]]]$$

The relationship between the functions

Object counting → abstract counting

Ionin & Matushansky (2006, 2018), Rothstein (2013, 2017), Kennedy (2015)

- ▶ the object-counting meaning is basic
- ▶ the abstract-counting meaning is derived

$$(3) \quad \llbracket \text{three}_{\text{Ac}} \rrbracket = \llbracket [\alpha [\text{three}_{\text{oc}}]] \rrbracket$$

Abstract counting → object counting

Scha (1981), Krifka (1995), Hackl (2000)

- ▶ the abstract-counting meaning is basic
- ▶ the object-counting meaning is derived

$$(4) \quad \llbracket \text{three}_{\text{oc}} \rrbracket = \llbracket [\beta [\text{three}_{\text{Ac}}]] \rrbracket$$

The relationship between the functions

Morphological marking patterns: abstract ~ object counting

cf. Greenberg (1978), Hurford (1998, 2001), Fassi Fehri (2018), Wągiel & Caha (2020, 2021)

- ▶ distinct morphological forms across languages

The relationship between the functions

Morphological marking patterns: abstract ~ object counting

cf. Greenberg (1978), Hurford (1998, 2001), Fassi Fehri (2018), Wągiel & Caha (2020, 2021)

- distinct morphological forms across languages

FUNCTION	PATTERNS					
ABSTRACT COUNTING	A	A+B	A	A+B	A	A+B
OBJECT COUNTING	A	A+B	A+C	A+B+C	C	A+C

The relationship between the functions

Morphological marking patterns: abstract ~ object counting

cf. Greenberg (1978), Hurford (1998, 2001), Fassi Fehri (2018), Wągiel & Caha (2020, 2021)

- distinct morphological forms across languages

FUNCTION	PATTERNS					
ABSTRACT COUNTING	A	A+B	A	A+B	A	A+B
OBJECT COUNTING	A	A+B	A+C	A+B+C	C	A+C

- (5) GENERALIZATION: When they differ, object-counting numerals are typically morphologically more complex than abstract-counting numerals.

(Wągiel & Caha 2020, 2021; pace Greenberg 1978)

The relationship between the functions

Morphological marking patterns: abstract ~ object counting

cf. Greenberg (1978), Hurford (1998, 2001), Fassi Fehri (2018), Wągiel & Caha (2020, 2021)

- distinct morphological forms across languages

FUNCTION	PATTERNS					
ABSTRACT COUNTING	A	A+B	A	A+B	A	A+B
OBJECT COUNTING	A	A+B	A+C	A+B+C	C	A+C

- (5) GENERALIZATION: When they differ, object-counting numerals are typically morphologically more complex than abstract-counting numerals.

(Wągiel & Caha 2020, 2021; pace Greenberg 1978)

The relationship between the functions

Morphological marking patterns: abstract ~ object counting

Zhang (2013), He (2015), Po-Ching & Rimmington (2015), Qi & He (2019)

- ▶ stacking: object counting > abstract counting

The relationship between the functions

Morphological marking patterns: abstract ~ object counting

Zhang (2013), He (2015), Po-Ching & Rimmington (2015), Qi & He (2019)

- ▶ stacking: object counting > abstract counting

- (6) a. ***wǔ** qiú
 five ball
 Intended: 'five balls'
 b. **wǔ gè** qiú
 five CLF ball
 'five balls'

Mandarin

The relationship between the functions

Morphological marking patterns: abstract ~ object counting

Zhang (2013), He (2015), Po-Ching & Rimmington (2015), Qi & He (2019)

- ▶ stacking: object counting > abstract counting

- (6) a. ***wǔ** qiú
five ball
Intended: 'five balls'

- b. **wǔ gè** qiú
five CLF ball
'five balls'

Mandarin

- (7) a. shí chúyǐ **wǔ** shì èr.
ten divide.by five CLF COP two₁
'Ten divided by five is two.'

- b. #shí gè chúyǐ **wǔ gè** shì {èr/liǎng} gè.
ten CLF divide.by five CLF COP two₁/two₂ CLF
Intended: 'Ten divided by five is two.'

Mandarin

The relationship between the functions

Morphological marking patterns: abstract ~ object counting

A. Borg (1974), A. J. Borg (1987), Hurford (1998, 2001)

- ▶ suppletion: object counting \neq abstract counting

The relationship between the functions

Morphological marking patterns: abstract ~ object counting

A. Borg (1974), A. J. Borg (1987), Hurford (1998, 2001)

- ▶ suppletion: object counting \neq abstract counting

- (8) a. ***tnejn** nisa
 two women
 Intended: 'two women'
 b. **żewġ** nisa
 two women
 'two women'

Maltese

The relationship between the functions

Morphological marking patterns: abstract ~ object counting

A. Borg (1974), A. J. Borg (1987), Hurford (1998, 2001)

- ▶ suppletion: object counting ≠ abstract counting

- (8) a. ***tnejn** nisa
two women
Intended: 'two women'
- b. **żewġ** nisa
two women
'two women'

Maltese

- (9) a. **Tnejn** u **tnejn** jagħmlu erbgħa.
two and two they-make four
'Two and two make four.'
- b. ***Żewġ** u **żewġ** jagħmlu erbgħa.
two and two they-make four
Intended: 'Two and two make four.'

Maltese

Puzzle

Mandarin Chinese forms for ‘two’

Zhang (2013), He (2015), Po-Ching & Rimmington (2015), Qi & He (2019)

- ▶ suppletion + stacking

Puzzle

Mandarin Chinese forms for ‘two’

Zhang (2013), He (2015), Po-Ching & Rimmington (2015), Qi & He (2019)

- ▶ suppletion + stacking

(10) liǎng gè qiú
 two₂ CLF ball
 ‘two balls’

Mandarin

Puzzle

Mandarin Chinese forms for 'two'

Zhang (2013), He (2015), Po-Ching & Rimmington (2015), Qi & He (2019)

- ▶ suppletion + stacking

(10) liǎng gè qiú

two₂ CLF ball

'two balls'

Mandarin

(11) shí chúyǐ èr shì wǔ.

ten divide.by two₁ COP five

'Ten divided by two is five.'

Mandarin

The main claim

Research question

- ▶ what can we learn about functions and compositionality of numerals by looking at their shape?

The main claim

Research question

- ▶ what can we learn about functions and compositionality of numerals by looking at their shape?

Generalization

- ▶ object-counting numerals are semantically and syntactically derived from abstract-counting numerals

The main claim

Research question

- ▶ what can we learn about functions and compositionality of numerals by looking at their shape?

Generalization

- ▶ object-counting numerals are semantically and syntactically derived from abstract-counting numerals

Puzzle

- ▶ Mandarin Chinese: èr ‘two₁’ ~ liǎng gè ‘two₂ CLF’
- ▶ suppletion + stacking ⇒ no clear relationship

The main claim

Proposal

- ▶ object-counting numerals both syntactically and semantically contain abstract-counting numerals

The main claim

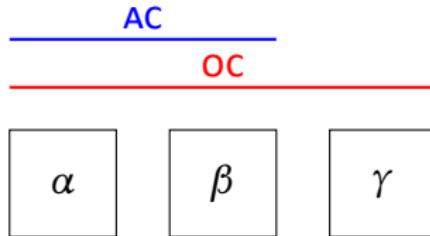
Proposal

- ▶ object-counting numerals both syntactically and semantically contain abstract-counting numerals
- ▶ abstract-counting numerals are already complex

The main claim

Proposal

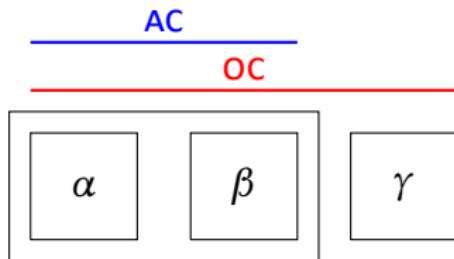
- ▶ object-counting numerals both syntactically and semantically contain abstract-counting numerals
- ▶ abstract-counting numerals are already complex



The main claim

Proposal

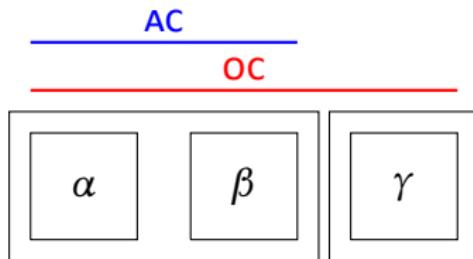
- ▶ object-counting numerals both syntactically and semantically contain abstract-counting numerals
- ▶ abstract-counting numerals are already complex



The main claim

Proposal

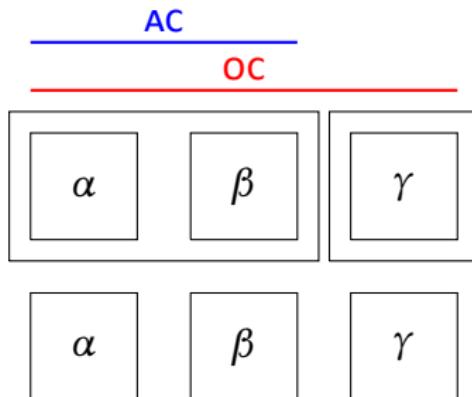
- ▶ object-counting numerals both syntactically and semantically contain abstract-counting numerals
- ▶ abstract-counting numerals are already complex



The main claim

Proposal

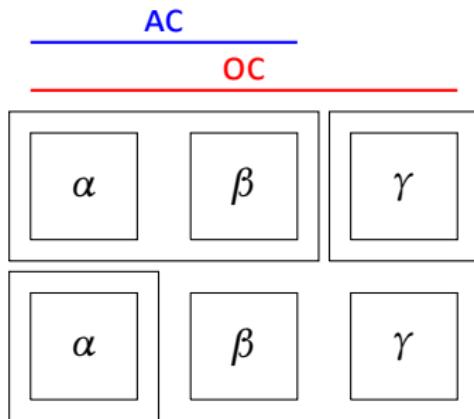
- ▶ object-counting numerals both syntactically and semantically contain abstract-counting numerals
- ▶ abstract-counting numerals are already complex



The main claim

Proposal

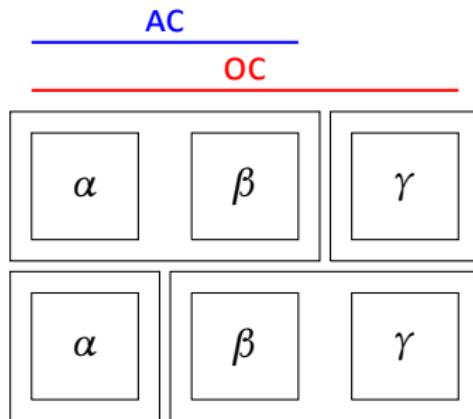
- ▶ object-counting numerals both syntactically and semantically contain abstract-counting numerals
- ▶ abstract-counting numerals are already complex



The main claim

Proposal

- ▶ object-counting numerals both syntactically and semantically contain abstract-counting numerals
- ▶ abstract-counting numerals are already complex



Introduction

Data

Structures

Spellout

Conclusions

Distinguishing abstract and object counting

Abstract number concepts ~ pluralities of individuals

Rothstein (2013, 2017), Wągiel & Caha (2020, 2021)

- ▶ different properties

Distinguishing abstract and object counting

Abstract number concepts ~ pluralities of individuals

Rothstein (2013, 2017), Wągiel & Caha (2020, 2021)

- ▶ different properties

- (12) a. Five times two equals ten.
 b. #Five things times two things equals ten things.

Distinguishing abstract and object counting

Abstract number concepts ~ pluralities of individuals

Rothstein (2013, 2017), Wągiel & Caha (2020, 2021)

- ▶ different properties

- (12) a. Five times two equals ten.
 b. #Five things times two things equals ten things.
- (13) a. Five is a natural number.
 b. #Five things are a natural number.

Distinguishing abstract and object counting

Abstract number concepts ~ pluralities of individuals

Rothstein (2013, 2017), Wągiel & Caha (2020, 2021)

- ▶ different properties

- (12) a. Five times two equals ten.
 b. #Five things times two things equals ten things.
- (13) a. Five is a natural number.
 b. #Five things are a natural number.
- (14) a. Five is the third prime number.
 b. #Five things are the third prime number.

Distinguishing abstract and object counting

Abstract number concepts ~ pluralities of individuals

Rothstein (2013, 2017), Wągiel & Caha (2020, 2021)

- ▶ different properties

- (12) a. Five times two equals ten.
 b. #Five things times two things equals ten things.
- (13) a. Five is a natural number.
 b. #Five things are a natural number.
- (14) a. Five is the third prime number.
 b. #Five things are the third prime number.
- (15) a. Kim can count up to five.
 b. #Kim can count up to five things.

Distinguishing abstract and object counting

Abstract number concepts ~ pluralities of individuals

Rothstein (2013, 2017), Wągiel & Caha (2020, 2021)

- ▶ different properties

Distinguishing abstract and object counting

Abstract number concepts ~ pluralities of individuals

Rothstein (2013, 2017), Wągiel & Caha (2020, 2021)

- ▶ different properties

- (16) a. Five is odd.
 b. Five things are odd.

Distinguishing abstract and object counting

Abstract number concepts ~ pluralities of individuals

Rothstein (2013, 2017), Wągiel & Caha (2020, 2021)

- ▶ different properties

- (16) a. Five is odd.
 b. Five things are odd.
- (17) a. Five is smaller than six.
 b. Five things are smaller than six things.

Distinguishing abstract and object counting

Abstract number concepts ~ pluralities of individuals

Rothstein (2013, 2017), Wągiel & Caha (2020, 2021)

- ▶ modification by numeral modifiers

Distinguishing abstract and object counting

Abstract number concepts ~ pluralities of individuals

Rothstein (2013, 2017), Wągiel & Caha (2020, 2021)

- ▶ modification by numeral modifiers

- (18) a. All **five** cats who live in the barn are crazy.
 b. #All **five** is an odd number.

Distinguishing abstract and object counting

Abstract number concepts ~ pluralities of individuals

Rothstein (2013, 2017), Wągiel & Caha (2020, 2021)

- ▶ modification by numeral modifiers

- (18) a. All **five** cats who live in the barn are crazy.
 b. #All **five** is an odd number.

- (19) a. More than **five** cities were destroyed.
 b. #More than **five** is a natural number.

Distinguishing abstract and object counting

Abstract number concepts ~ pluralities of individuals

Rothstein (2013, 2017), Wągiel & Caha (2020, 2021)

- ▶ scalar implicatures

Distinguishing abstract and object counting

Abstract number concepts ~ pluralities of individuals

Rothstein (2013, 2017), Wągiel & Caha (2020, 2021)

► scalar implicatures

- (20) a. You must take **five** cards to win. ✓ AT LEAST
 b. You must multiply two by **five** to get ten. #AT LEAST

Distinguishing abstract and object counting

Abstract number concepts ~ pluralities of individuals

Rothstein (2013, 2017), Wągiel & Caha (2020, 2021)

► scalar implicatures

- | | | | |
|------|----|---|------------|
| (20) | a. | You must take five cards to win. | ✓ AT LEAST |
| | b. | You must multiply two by five to get ten. | #AT LEAST |
| (21) | a. | Kim took five cards, if not more. | |
| | b. | Two multiplied by five equals ten, #if not more. | |

Mandarin numerals

Regular non-suppletive numerals

Zhang (2013), He (2015), Po-Ching & Rimmington (2015), Qi & He (2019)

- ▶ stacking: object counting > abstract counting

Mandarin numerals

Regular non-suppletive numerals

Zhang (2013), He (2015), Po-Ching & Rimmington (2015), Qi & He (2019)

- ▶ stacking: object counting > abstract counting

- (22) a. ***wǔ** qiú
 five ball
 Intended: ‘five balls’
- b. **wǔ gè** qiú
 five CLF ball
 ‘five balls’

Mandarin

Mandarin numerals

Regular non-suppletive numerals

Zhang (2013), He (2015), Po-Ching & Rimmington (2015), Qi & He (2019)

- ▶ stacking: object counting > abstract counting

- (22) a. ***wǔ** qiú
five ball
Intended: ‘five balls’
- b. **wǔ gè** qiú
five CLF ball
‘five balls’

Mandarin

- (23) a. shí chúyǐ **wǔ** shì èr.
ten divide.by five CLF COP two₁
‘Ten divided by five is two.’
- b. #shí gè chúyǐ **wǔ gè** shì {èr/liǎng} gè.
ten CLF divide.by five CLF COP two₁/two₂ CLF
Intended: ‘Ten divided by five is two.’

Mandarin

Data

Regular non-suppletive numerals

Zhang (2013), He (2015), Po-Ching & Rimmington (2015), Qi & He (2019)

- ▶ distinction not due to the syntactic position of the numeral
- ▶ predicate position, fragment answers

Data

Regular non-suppletive numerals

Zhang (2013), He (2015), Po-Ching & Rimmington (2015), Qi & He (2019)

- ▶ distinction not due to the syntactic position of the numeral
- ▶ predicate position, fragment answers

(24) a. wǒ yǒu de qiú shì wǔ gè

I have DE ball be five CLF

'The balls I have are five.'

b. *wǒ yǒu de qiú shì wǔ

I have DE ball be five

Intended: 'The balls I have are five.'

Mandarin

Data

Regular non-suppletive numerals

Zhang (2013), He (2015), Po-Ching & Rimmington (2015), Qi & He (2019)

- ▶ distinction not due to the syntactic position of the numeral
- ▶ predicate position, fragment answers

- (24) a. wǒ yǒu de qiú shì wǔ gè

I have DE ball be five CLF

'The balls I have are five.'

- b. *wǒ yǒu de qiú shì wǔ

I have DE ball be five

Intended: 'The balls I have are five.'

Mandarin

- (25) Q: How many balls do you have?

- a. wǔ gè

five CLF

'Five.'

- b. *wǔ

five

Intended: 'Five.'

Mandarin

Mandarin numerals

Three Mandarin 'two's: èr

Zhang (2013), He (2015), Po-Ching & Rimmington (2015), Qi & He (2019)

- ▶ incompatible with classifiers
- ▶ infelicitous in object-counting contexts
- ▶ felicitous in abstract-counting contexts

Mandarin numerals

Three Mandarin 'two's: èr

Zhang (2013), He (2015), Po-Ching & Rimmington (2015), Qi & He (2019)

- ▶ incompatible with classifiers
- ▶ infelicitous in object-counting contexts
- ▶ felicitous in abstract-counting contexts

- (26) a. *èr qiú
 two₁ ball
 Intended: 'two balls'
 b. *èr gè qiú
 two₁ CLF ball
 Intended: 'two balls'

Mandarin

Mandarin numerals

Three Mandarin ‘two’s: èr

Zhang (2013), He (2015), Po-Ching & Rimmington (2015), Qi & He (2019)

- ▶ incompatible with classifiers
- ▶ infelicitous in object-counting contexts
- ▶ felicitous in abstract-counting contexts

- (26) a. *èr qiú
two₁ ball
Intended: ‘two balls’
- b. *èr gè qiú
two₁ CLF ball
Intended: ‘two balls’

Mandarin

- (27) a. shí chúyǐ èr shì wǔ.
ten divide.by two₁ COP five
‘Ten divided by two is five.’
- b. *shí gè chúyǐ èr gè shì wǔ gè.
ten CLF divide.by two₁ CLF COP five CLF
Intended: ‘Ten divided by two is five.’

Mandarin

Mandarin numerals

Three Mandarin ‘two’s: èr

Zhang (2013), He (2015), Po-Ching & Rimmington (2015), Qi & He (2019)

- ▶ infelicitous in predicate position
- ▶ infelicitous in fragment answers

Mandarin numerals

Three Mandarin 'two's: èr

Zhang (2013), He (2015), Po-Ching & Rimmington (2015), Qi & He (2019)

- ▶ infelicitous in predicate position
- ▶ infelicitous in fragment answers

(28) a. *wǒ yǒu de qiú shì èr gè

I have DE ball be two₁ CLF

Intended: 'The balls I have are two.'

b. #wǒ yǒu de qiú shì èr

I have DE ball be two₁

Intended: 'The balls I have are two.'

Mandarin

Mandarin numerals

Three Mandarin ‘two’s: èr

Zhang (2013), He (2015), Po-Ching & Rimmington (2015), Qi & He (2019)

- ▶ infelicitous in predicate position
- ▶ infelicitous in fragment answers

- (28) a. *wǒ yǒu de qiú shì èr gè

I have DE ball be two₁ CLF

Intended: ‘The balls I have are two.’

- b. #wǒ yǒu de qiú shì èr

I have DE ball be two₁

Intended: ‘The balls I have are two.’

Mandarin

- (29) Q: How many balls do you have?

- a. *èr gè

two₁ CLF

Intended: ‘Two.’

- b. #èr

two₁

Intended: ‘Two.’

Mandarin

Mandarin numerals

Three Mandarin ‘two’s: *liǎng*

Zhang (2013), He (2015), Po-Ching & Rimmington (2015), Qi & He (2019)

- ▶ compatible with classifiers
- ▶ infelicitous in abstract-counting contexts
- ▶ felicitous in object-counting contexts with a classifier

Mandarin numerals

Three Mandarin ‘two’s: *liǎng*

Zhang (2013), He (2015), Po-Ching & Rimmington (2015), Qi & He (2019)

- ▶ compatible with classifiers
- ▶ infelicitous in abstract-counting contexts
- ▶ felicitous in object-counting contexts with a classifier

- (30) a. ***liǎng** qiú
 two₂ ball
 Intended: ‘two balls’
- b. **liǎng gè** qiú
 two₂ CLF ball
 ‘two balls’

Mandarin

Mandarin numerals

Three Mandarin ‘two’s: *liǎng*

Zhang (2013), He (2015), Po-Ching & Rimmington (2015), Qi & He (2019)

- ▶ compatible with classifiers
- ▶ infelicitous in abstract-counting contexts
- ▶ felicitous in object-counting contexts with a classifier

- (30) a. ***liǎng** qiú
two₂ ball
Intended: ‘two balls’
- b. **liǎng gè** qiú
two₂ CLF ball
‘two balls’

Mandarin

- (31) a. *shí chúyǐ **liǎng** shì wǔ.
ten divide.by two₂ COP five
Intended: ‘Ten divided by two is five.’
- b. #shí gè chúyǐ gè **liǎng gè** shì wǔ gè.
ten CLF divide.by two₂ CLF COP five CLF
Intended: ‘Ten divided by two is five.’

Mandarin

Mandarin numerals

Three Mandarin 'two's: *liǎng*

Zhang (2013), He (2015), Po-Ching & Rimmington (2015), Qi & He (2019)

- ▶ felicitous in predicate position with a classifier
- ▶ felicitous in fragment answers with a classifier

Mandarin numerals

Three Mandarin 'two's: *liǎng*

Zhang (2013), He (2015), Po-Ching & Rimmington (2015), Qi & He (2019)

- ▶ felicitous in predicate position with a classifier
- ▶ felicitous in fragment answers with a classifier

(32) a. wǒ yǒu de qiú shì **liǎng** gè

I have DE ball be two₂ CLF

'The balls I have are two.'

b. *wǒ yǒu de qiú shì **liǎng**

I have DE ball be two₂

Intended: 'The balls I have are two.'

Mandarin

Mandarin numerals

Three Mandarin 'two's: *liǎng*

Zhang (2013), He (2015), Po-Ching & Rimmington (2015), Qi & He (2019)

- ▶ felicitous in predicate position with a classifier
- ▶ felicitous in fragment answers with a classifier

- (32) a. wǒ yǒu de qiú shì **liǎng gè**
 I have DE ball be two₂ CLF
 'The balls I have are two.'
 b. *wǒ yǒu de qiú shì **liǎng**
 I have DE ball be two₂
 Intended: 'The balls I have are two.'

Mandarin

- (33) Q: How many balls do you have?
- a. **liǎng gè**
 two₂ CLF
 'Two.'
b. ***liǎng**
 two₂
 Intended: 'Two.'

Mandarin

Mandarin numerals

Three Mandarin 'two's: *liǎ*

Zhang (2013), He (2015), Po-Ching & Rimmington (2015), Qi & He (2019)

- ▶ infelicitous in abstract-counting contexts
- ▶ felicitous in object-counting contexts
- ▶ incompatible with classifiers

Mandarin numerals

Three Mandarin 'two's: *liǎ*

Zhang (2013), He (2015), Po-Ching & Rimmington (2015), Qi & He (2019)

- ▶ infelicitous in abstract-counting contexts
- ▶ felicitous in object-counting contexts
- ▶ incompatible with classifiers

(34) a. **liǎ xuésheng**

two₃ student

'two students'

b. ***liǎ gè xuésheng**

two₃ CLF student

Intended: 'two students'

Mandarin

Mandarin numerals

Three Mandarin 'two's: *liǎ*

Zhang (2013), He (2015), Po-Ching & Rimmington (2015), Qi & He (2019)

- ▶ infelicitous in abstract-counting contexts
- ▶ felicitous in object-counting contexts
- ▶ incompatible with classifiers

(34) a. **liǎ xuésheng**

two₃ student

'two students'

b. ***liǎ gè xuésheng**

two₃ CLF student

Intended: 'two students'

Mandarin

(35) a. #shí chúyǐ **liǎ shì wǔ.**

ten divide.by two₃ COP five

Intended: 'Ten divided by two is five.'

b. *shí gè chúyǐ **liǎ gè shì wǔ gè.**

ten CLF divide.by two₃ CLF COP five CLF

Intended: 'Ten divided by two is five.'

Mandarin

Mandarin numerals

Three Mandarin 'two's: *liǎ*

Zhang (2013), He (2015), Po-Ching & Rimmington (2015), Qi & He (2019)

- ▶ interspeaker variation
- ▶ some speakers accept *liǎ* only with human nouns
- ▶ for others it is only a matter of register

Mandarin numerals

Three Mandarin 'two's: *liǎ*

Zhang (2013), He (2015), Po-Ching & Rimmington (2015), Qi & He (2019)

- ▶ interspeaker variation
- ▶ some speakers accept *liǎ* only with human nouns
- ▶ for others it is only a matter of register

- (36) a. **liǎ** xuésheng
 two₃ student
 'two students'
 b. **liǎng wèi** xuésheng
 two₃ CLF.HUM student
 'two students'

Mandarin

Mandarin numerals

Three Mandarin 'two's: *liǎ*

Zhang (2013), He (2015), Po-Ching & Rimmington (2015), Qi & He (2019)

- ▶ interspeaker variation
- ▶ some speakers accept *liǎ* only with human nouns
- ▶ for others it is only a matter of register

- (36) a. **liǎ** xuésheng
two₃ student
'two students'
- b. **liǎng wèi** xuésheng
two₃ CLF.HUM student
'two students'

Mandarin

- (37) a. %**liǎ** qiú
two₃ ball
'two balls'
- b. **liǎng gè** qiú
two₃ CLF ball
'two balls'

Mandarin

Mandarin numerals

Three Mandarin 'two's: *liǎ*

Zhang (2013), He (2015), Po-Ching & Rimmington (2015), Qi & He (2019)

- ▶ felicitous in predicate position without a classifier
- ▶ felicitous in fragment answers without a classifier

Mandarin numerals

Three Mandarin 'two's: *liǎ*

Zhang (2013), He (2015), Po-Ching & Rimmington (2015), Qi & He (2019)

- ▶ felicitous in predicate position without a classifier
- ▶ felicitous in fragment answers without a classifier

(38) a. *wǒ yǒu de qiú shì liǎ gè

I have DE ball be two₃ CLF

Intended: 'The balls I have are two.'

b. wǒ yǒu de qiú shì liǎ

I have DE ball be two₃

'The balls I have are two.'

Mandarin

Mandarin numerals

Three Mandarin 'two's: *liǎ*

Zhang (2013), He (2015), Po-Ching & Rimmington (2015), Qi & He (2019)

- ▶ felicitous in predicate position without a classifier
- ▶ felicitous in fragment answers without a classifier

- (38) a. *wǒ yǒu de qiú shì **liǎ** **gè**
 I have DE ball be two₃ CLF
 Intended: 'The balls I have are two.'
 b. wǒ yǒu de qiú shì **liǎ**
 I have DE ball be two₃
 'The balls I have are two.'

Mandarin

- (39) Q: How many balls do you have?
- a. ***liǎ** **gè**
 two₃ CLF
 Intended: 'Two.'
 b. **liǎ**
 two₃
 'Two.'

Mandarin

Mandarin numerals

Data summary

- ▶ non-suppletive numerals
- ▶ three ‘two’s: suppletion + stacking

Mandarin numerals

Data summary

- ▶ non-suppletive numerals
- ▶ three ‘two’s: suppletion + stacking

		ABSTRACT COUNTING	OBJECT COUNTING
wǔ	‘five’	✓	*
èr	‘two ₁ ’	✓	*
èr gè	‘two ₁ CLF’	*	*
liǎng	‘two ₂ ’	*	*
liǎ gè	‘two ₃ CLF’	*	*
wǔ gè	‘five CLF’	*	✓
liǎng gè	‘two ₂ CLF’	*	✓
liǎ	‘two ₃ ’	*	✓

Introduction

Data

Structures

Spellout

Conclusions

Universal semantic features

Key intuition concerning numerals

- ▶ numerals are at their core scalar expressions

Universal semantic features

Key intuition concerning numerals

- ▶ numerals are at their core scalar expressions
- ▶ ordering \Rightarrow primitive notion

Universal semantic features

Key intuition concerning numerals

- ▶ numerals are at their core scalar expressions
- ▶ ordering \Rightarrow primitive notion
- ▶ numbers \Rightarrow constructed objects

Universal semantic features

Key intuition concerning numerals

- ▶ numerals are at their core scalar expressions
- ▶ ordering \Rightarrow primitive notion
- ▶ numbers \Rightarrow constructed objects
- ▶ each numeral \Rightarrow interval on the number scale

Universal semantic features

Key intuition concerning numerals

- ▶ numerals are at their core scalar expressions
- ▶ ordering \Rightarrow primitive notion
- ▶ numbers \Rightarrow constructed objects
- ▶ each numeral \Rightarrow interval on the number scale
- ▶ non-arbitrary starting point $\Rightarrow 0$

Universal semantic features

Key intuition concerning numerals

- ▶ numerals are at their core scalar expressions
- ▶ ordering \Rightarrow primitive notion
- ▶ numbers \Rightarrow constructed objects
- ▶ each numeral \Rightarrow interval on the number scale
- ▶ non-arbitrary starting point $\Rightarrow 0$

Motivation

Corver & Zwarts (2006), Nouwen (2008, 2016); Seuren (1984), Kennedy (2001)

Universal semantic features

Key intuition concerning numerals

- ▶ numerals are at their core scalar expressions
- ▶ ordering ⇒ primitive notion
- ▶ numbers ⇒ constructed objects
- ▶ each numeral ⇒ interval on the number scale
- ▶ non-arbitrary starting point ⇒ 0

Motivation

Corver & Zwarts (2006), Nouwen (2008, 2016); Seuren (1984), Kennedy (2001)

- ▶ research on spatial/directional numeral modifiers
- (40) a. above five
 b. between five and eight

Universal semantic features

Key intuition concerning numerals

- ▶ numerals are at their core scalar expressions
- ▶ ordering ⇒ primitive notion
- ▶ numbers ⇒ constructed objects
- ▶ each numeral ⇒ interval on the number scale
- ▶ non-arbitrary starting point ⇒ 0

Motivation

Corver & Zwarts (2006), Nouwen (2008, 2016); Seuren (1984), Kennedy (2001)

- ▶ research on spatial/directional numeral modifiers
- (40) a. above five
 b. between five and eight

- ▶ interval-based semantics of degree

Universal semantic features

Standard approach to classifiers

e.g., Borer (2005), Chierchia (1998, 2010), Rothstein (2010), Li (2011), Scontras (2013)

- ▶ mass-like semantics of nouns in classifier languages
- ▶ classifiers compensate semantic deficits of nouns

Universal semantic features

Standard approach to classifiers

e.g., Borer (2005), Chierchia (1998, 2010), Rothstein (2010), Li (2011), Scontras (2013)

- ▶ mass-like semantics of nouns in classifier languages
- ▶ classifiers compensate semantic deficits of nouns

Alternative view

Krifka (1995), Bale & Coon (2014), Sudo (2016)

- ▶ different semantics of numerals in classifier languages
- ▶ classifiers compensate semantic deficits of numerals

Universal semantic features

Standard approach to classifiers

e.g., Borer (2005), Chierchia (1998, 2010), Rothstein (2010), Li (2011), Scontras (2013)

- ▶ mass-like semantics of nouns in classifier languages
- ▶ classifiers compensate semantic deficits of nouns

Alternative view

Krifka (1995), Bale & Coon (2014), Sudo (2016)

- ▶ different semantics of numerals in classifier languages
- ▶ classifiers compensate semantic deficits of numerals

Adjectival theory of numerals

Landman (2003, 2004)

- ▶ numerals are intersective modifiers
- ▶ combine with NPs via Predicate Modification

Universal semantic features

Meaning components

- ▶ closed interval \Rightarrow set of numbers

- (41) a. $\llbracket \text{SCALE}_m \rrbracket_{\langle n, t \rangle} = \lambda n_n [0 \leq n \leq m]$
 b. $\llbracket \text{SCALE}_5 \rrbracket = [0, 5]$

Universal semantic features

Meaning components

- ▶ closed interval \Rightarrow set of numbers

(41) a. $\llbracket \text{SCALE}_m \rrbracket_{\langle n, t \rangle} = \lambda n_n [0 \leq n \leq m]$
 b. $\llbracket \text{SCALE}_5 \rrbracket = [0, 5]$

- ▶ maximization operator \Rightarrow name of a number concept

(42) a. $\llbracket \text{NUM} \rrbracket_{\langle \langle n, t \rangle, n \rangle} = \lambda P_{\langle n, t \rangle} [\text{MAX}(P)]$
 b. $\llbracket \text{NUM} \rrbracket(\llbracket \text{SCALE}_5 \rrbracket) = 5$

Universal semantic features

Meaning components

- ▶ closed interval \Rightarrow set of numbers

(41) a. $\llbracket \text{SCALE}_m \rrbracket_{\langle n, t \rangle} = \lambda n_n [0 \leq n \leq m]$
 b. $\llbracket \text{SCALE}_5 \rrbracket = [0, 5]$

- ▶ maximization operator \Rightarrow name of a **number concept**

(42) a. $\llbracket \text{NUM} \rrbracket_{\langle \langle n, t \rangle, n \rangle} = \lambda P_{\langle n, t \rangle} [\text{MAX}(P)]$
 b. $\llbracket \text{NUM} \rrbracket(\llbracket \text{SCALE}_5 \rrbracket) = 5$

- ▶ classifier semantics \Rightarrow shift to a **counting device**

(43) a. $\llbracket \text{CL} \rrbracket_{\langle n, \langle e, t \rangle \rangle} = \lambda n_n \lambda x_e [\#(x) = n]$
 b. $\llbracket \text{CL} \rrbracket(\llbracket \text{NUM} \rrbracket(\llbracket \text{SCALE}_5 \rrbracket)) = \lambda x_e [\#(x) = 5]$

Universal semantic features

Structures

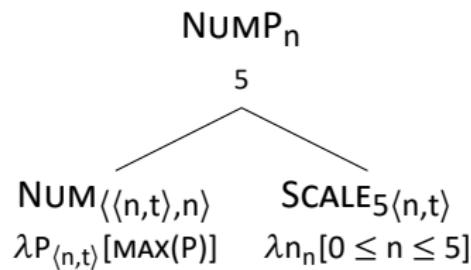
- ▶ abstract-counting numerals

Universal semantic features

Structures

- abstract-counting numerals

(44)



Universal semantic features

Structures

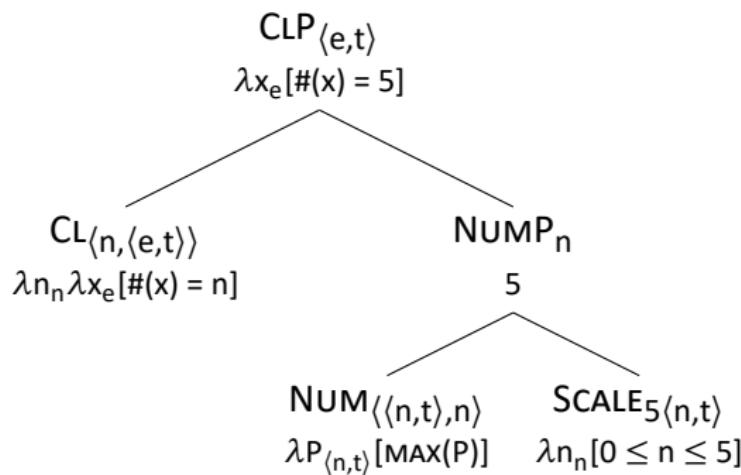
- ▶ object-counting numerals

Universal semantic features

Structures

- ▶ object-counting numerals

(45)



Empirical corroboration

Numerals in Shuhi (Qiangic)

Qi & He (2019)

- ▶ abstract-counting forms: morphologically complex

Empirical corroboration

Numerals in Shuhi (Qiangic)

Qi & He (2019)

- ▶ abstract-counting forms: morphologically complex

- (46)
- a. rɔ?³⁵ **dzi³³-ko³⁵**
horse one-CLF
'one horse'
 - b. nw⁵⁵gu³¹ **dzi³³-ly⁵⁵**
cloth one-CLF
'one cloth'
 - c. la³³re⁵⁵ **dzi³³-tshu⁵⁵**
towel one-CLF
'one towel'

Shuhi

Empirical corroboration

Numerals in Shuhi (Qiangic)

Qi & He (2019)

► abstract-counting forms: morphologically complex

- (46) a. rɔ?³⁵ **dzi³³-ko³⁵**
horse one-CLF
'one horse'
b. nw⁵⁵ gu³¹ **dzi³³-ly⁵⁵**
cloth one-CLF
'one cloth'
c. la³³ re⁵⁵ **dzi³³-tshu⁵⁵**
towel one-CLF
'one towel'

Shuhi

- (47) **dzi³³-ko³⁵-re³³ dzi³³-ko³⁵-ho~³³ me³³-ba³³-le⁵⁵ nɛ³³-ko³⁵**
one-CLF-ABL one-CLF-LOC DIR-add-AUX two-CLF
le³³-zi?³³-dzo~³³.
DIR-become-DUR
'One plus one is two.'

Shuhi

Empirical corroboration

Numerals 1–5 in Vera'a (Vanuatu)

Schnell (2011)

- ▶ full agglutination: each head pronounced

Empirical corroboration

Numerals 1–5 in Vera'a (Vanuatu)

Schnell (2011)

- ▶ full agglutination: each head pronounced

NUMBER	CARDINAL	MULTIPLICATIVE
1	vō-wal	vag-wal
2	vō-ruō	vag-ruō
3	vō-'ōl	vag-'ōl
4	vō-ve'	vag-ve'

Empirical corroboration

Numerals 1–5 in Vera'a (Vanuatu)

Schnell (2011)

- ▶ full agglutination: each head pronounced

	NUMBER	CARDINAL	MULTIPLICATIVE
1	vō-wal	vag-wal	
2	vō-ruō	vag-ruō	
3	vō-'ōl	vag-'ōl	
4	vō-ve'	vag-ve'	

- (48) vēvē-gi ne lukun ēn naw, din ēn vō-'ōl...
mother-3SG TAM count ART wave reach ART NBR-three
'Then his mother counted the waves reaching (the number)
three...' Vera'a

Empirical corroboration

Numerals 1–5 in Vera'a (Vanuatu)

Schnell (2011)

- ▶ full agglutination: each head pronounced

	NUMBER	CARDINAL	MULTIPLICATIVE
1	vō-wal	vag-wal	
2	vō-ruō	vag-ruō	
3	vō-'ōl	vag-'ōl	
4	vō-ve'	vag-ve'	

- (48) vēvē-gi ne lukun ēn naw, din ēn vō-'ōl...
mother-3SG TAM count ART wave reach ART NBR-three
'Then his mother counted the waves reaching (the number)
three...' Vera'a

- (49) ēn woqe'enge ne vō-ru
ART tree LIG NBR-two
'two trees' Vera'a

Universal semantic features

Specifying heads

Sudo (2016)

- ▶ specific classification: two types of information

Universal semantic features

Specifying heads

Sudo (2016)

- ▶ specific classification: two types of information
- ▶ grammatical: animacy, size,...
- ▶ conceptual: edibility, artefacts,...

Universal semantic features

Specifying heads

Sudo (2016)

- ▶ specific classification: two types of information
- ▶ grammatical: animacy, size,...
- ▶ conceptual: edibility, artefacts,...
- ▶ presuppositions regarding the referents of the NP

Universal semantic features

Specifying heads

Sudo (2016)

- ▶ specific classification: two types of information
- ▶ grammatical: animacy, size,...
- ▶ conceptual: edibility, artefacts,...
- ▶ presuppositions regarding the referents of the NP

$$(50) \quad \llbracket \text{ANIM} \rrbracket_{\langle \langle e, t \rangle, \langle e, t \rangle \rangle} = \lambda P_{\langle e, t \rangle} \lambda x_e : \text{ANIMATE}(x)[P(x)] \quad (\text{zhī})$$

$$(51) \quad \llbracket \text{HUM} \rrbracket_{\langle \langle e, t \rangle, \langle e, t \rangle \rangle} = \lambda P_{\langle e, t \rangle} \lambda x_e : \text{HUMAN}(x)[P(x)] \quad (\text{wèi})$$

$$(52) \quad \llbracket \text{HON} \rrbracket_{\langle \langle e, t \rangle, \langle e, t \rangle \rangle} = \lambda P_{\langle e, t \rangle} \lambda x_e : \text{HIGH-STATUS}(x)[P(x)] \quad (\text{míng})$$

Universal semantic features

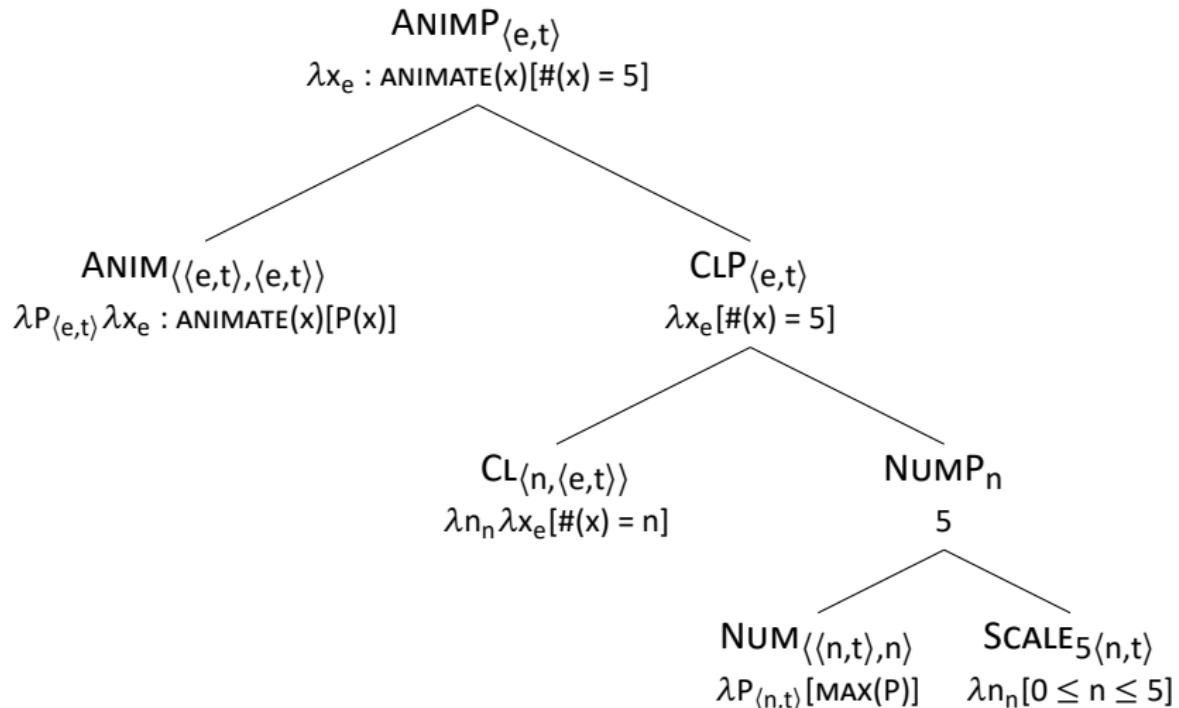
Structures

- ▶ specific object-counting numerals

Universal semantic features

Structures

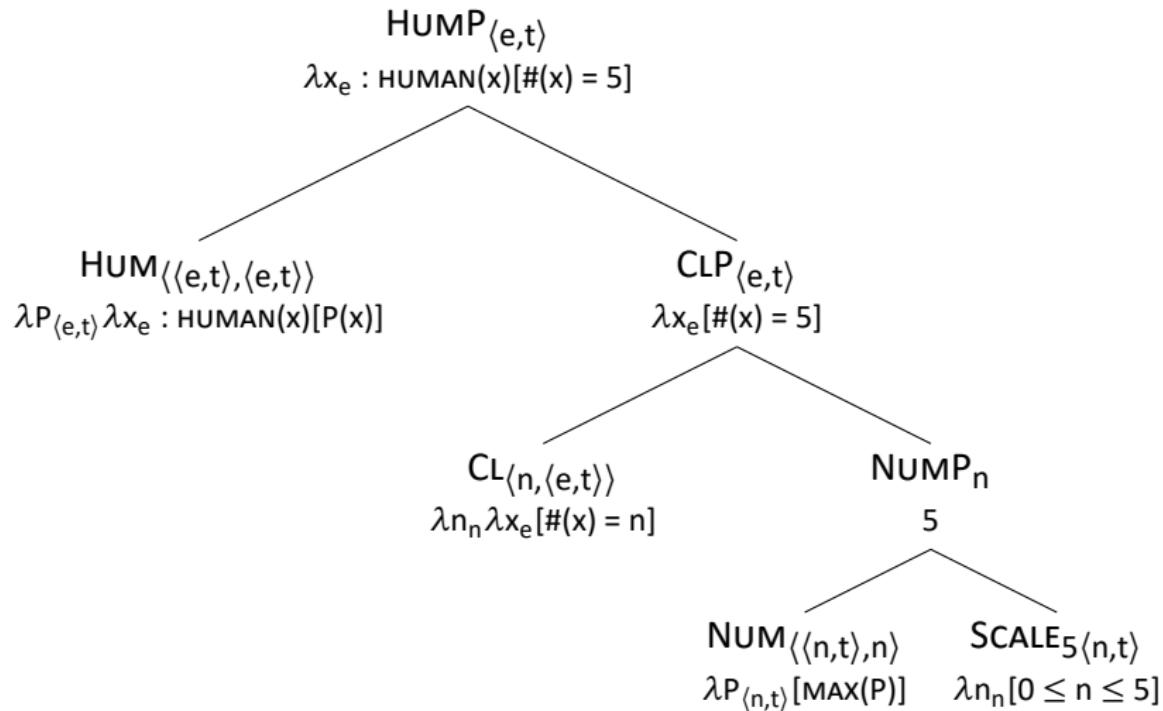
- ▶ specific object-counting numerals



Universal semantic features

Structures

► specific object-counting numerals



Introduction

Data

Structures

Spellout

Conclusions

Nanosyntax

In Wągiel & Caha (2021), we show that it is possible to maintain a universal structure for numerals and still account for crosslinguistic variation in morphology — using Nanosyntax

for Nanosyntax, see Starke (2009, 2018), Caha (2009), Baunaz & Lander (2018), Caha et al. (2019)

Nanosyntax

In Wągiel & Caha (2021), we show that it is possible to maintain a universal structure for numerals and still account for crosslinguistic variation in morphology — using Nanosyntax

for Nanosyntax, see Starke (2009, 2018), Caha (2009), Baunaz & Lander (2018), Caha et al. (2019)

- ▶ realizational model of morphology

Nanosyntax

In Wągiel & Caha (2021), we show that it is possible to maintain a universal structure for numerals and still account for crosslinguistic variation in morphology — using Nanosyntax

for Nanosyntax, see Starke (2009, 2018), Caha (2009), Baunaz & Lander (2018), Caha et al. (2019)

- ▶ realizational model of morphology
- ▶ maps structures to their pronunciation using

Nanosyntax

In Wągiel & Caha (2021), we show that it is possible to maintain a universal structure for numerals and still account for crosslinguistic variation in morphology — using Nanosyntax

for Nanosyntax, see Starke (2009, 2018), Caha (2009), Baunaz & Lander (2018), Caha et al. (2019)

- ▶ realizational model of morphology
- ▶ maps structures to their pronunciation using
 - ▶ language-specific lexical entries

Nanosyntax

In Wągiel & Caha (2021), we show that it is possible to maintain a universal structure for numerals and still account for crosslinguistic variation in morphology — using Nanosyntax

for Nanosyntax, see Starke (2009, 2018), Caha (2009), Baunaz & Lander (2018), Caha et al. (2019)

- ▶ realizational model of morphology
- ▶ maps structures to their pronunciation using
 - ▶ language-specific lexical entries
 - ▶ a language-invariant spellout procedure

Nanosyntax

In Wągiel & Caha (2021), we show that it is possible to maintain a universal structure for numerals and still account for crosslinguistic variation in morphology — using Nanosyntax

for Nanosyntax, see Starke (2009, 2018), Caha (2009), Baunaz & Lander (2018), Caha et al. (2019)

- ▶ realizational model of morphology
- ▶ maps structures to their pronunciation using
 - ▶ language-specific lexical entries
 - ▶ a language-invariant spellout procedure
- ▶ phrasal spellout

Nanosyntax

In Wągiel & Caha (2021), we show that it is possible to maintain a universal structure for numerals and still account for crosslinguistic variation in morphology — using Nanosyntax

for Nanosyntax, see Starke (2009, 2018), Caha (2009), Baunaz & Lander (2018), Caha et al. (2019)

- ▶ realizational model of morphology
- ▶ maps structures to their pronunciation using
 - ▶ language-specific lexical entries
 - ▶ a language-invariant spellout procedure
- ▶ phrasal spellout
- ▶ cyclic

Nanosyntax

In Wągiel & Caha (2021), we show that it is possible to maintain a universal structure for numerals and still account for crosslinguistic variation in morphology — using Nanosyntax

for Nanosyntax, see Starke (2009, 2018), Caha (2009), Baunaz & Lander (2018), Caha et al. (2019)

- ▶ realizational model of morphology
- ▶ maps structures to their pronunciation using
 - ▶ language-specific lexical entries
 - ▶ a language-invariant spellout procedure
- ▶ phrasal spellout
- ▶ cyclic
- ▶ spellout driven movement

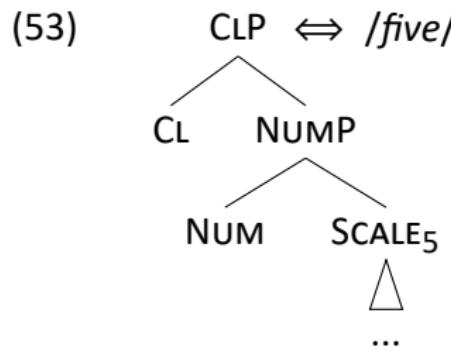
Nanosyntax

In Wągiel & Caha (2021), we show that it is possible to maintain a universal structure for numerals and still account for crosslinguistic variation in morphology — using Nanosyntax

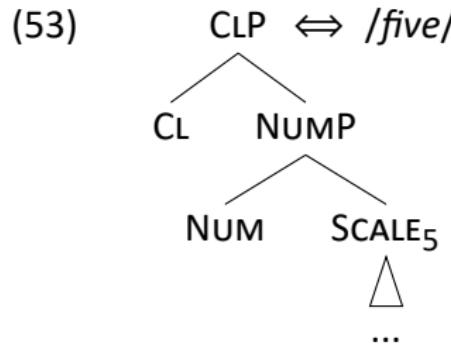
for Nanosyntax, see Starke (2009, 2018), Caha (2009), Baunaz & Lander (2018), Caha et al. (2019)

- ▶ realizational model of morphology
- ▶ maps structures to their pronunciation using
 - ▶ language-specific lexical entries
 - ▶ a language-invariant spellout procedure
- ▶ phrasal spellout
- ▶ cyclic
- ▶ spellout driven movement

Simplex numerals: Syncretism



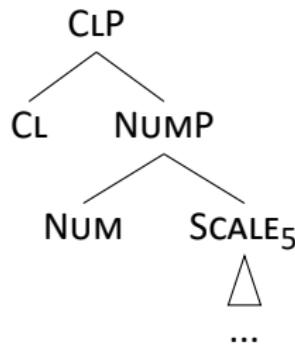
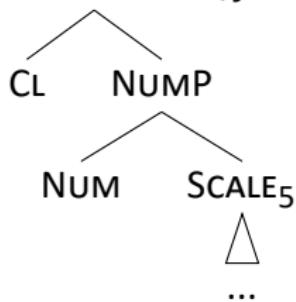
Simplex numerals: Syncretism



- (54) THE SUPERSET PRINCIPLE (Starke 2009):
A lexically stored tree L matches a syntactic node S iff L contains the syntactic tree dominated by S as a subtree.

Simplex numerals: Syncretism

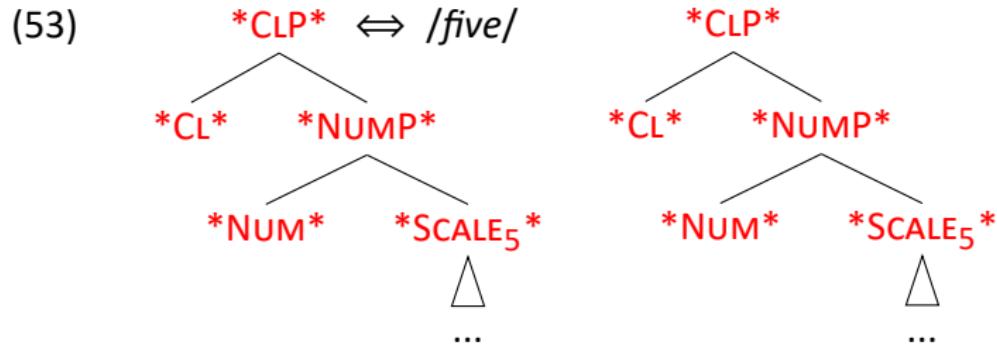
(53) CLP \Leftrightarrow /five/



(54) THE SUPERSET PRINCIPLE (Starke 2009):

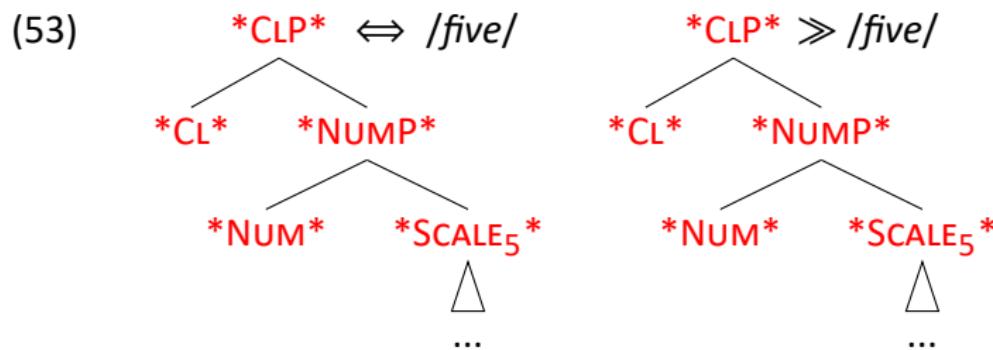
A lexically stored tree L matches a syntactic node S iff L contains the syntactic tree dominated by S as a subtree.

Simplex numerals: Syncretism



- (54) THE SUPERSET PRINCIPLE (Starke 2009):
A lexically stored tree L matches a syntactic node S iff L contains the syntactic tree dominated by S as a subtree.

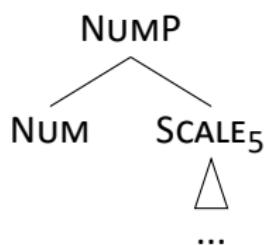
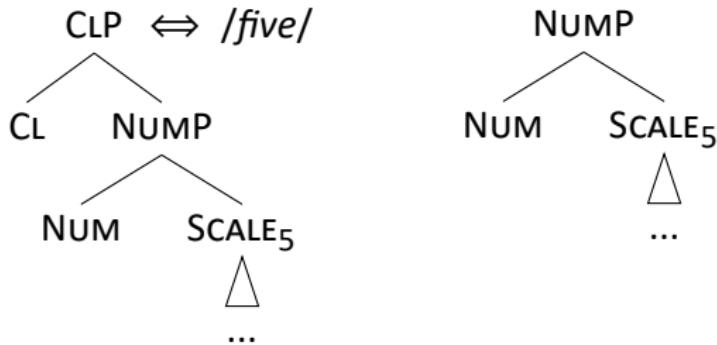
Simplex numerals: Syncretism



- (54) THE SUPERSET PRINCIPLE (Starke 2009):
A lexically stored tree L matches a syntactic node S iff L contains the syntactic tree dominated by S as a subtree.

Simplex numerals: Syncretism

(53) CLP \Leftrightarrow /five/

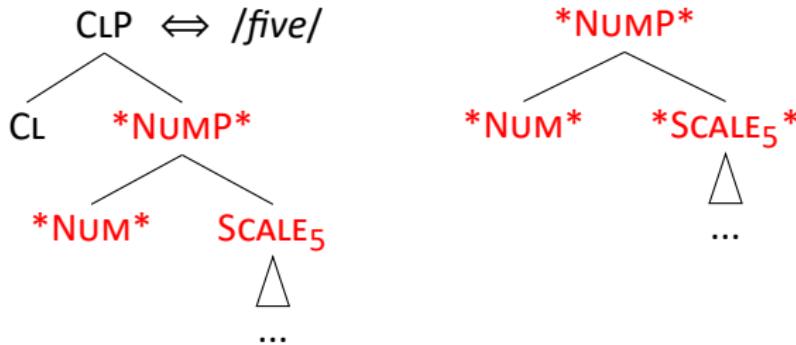


(54) THE SUPERSET PRINCIPLE (Starke 2009):

A lexically stored tree L matches a syntactic node S iff L contains the syntactic tree dominated by S as a subtree.

Simplex numerals: Syncretism

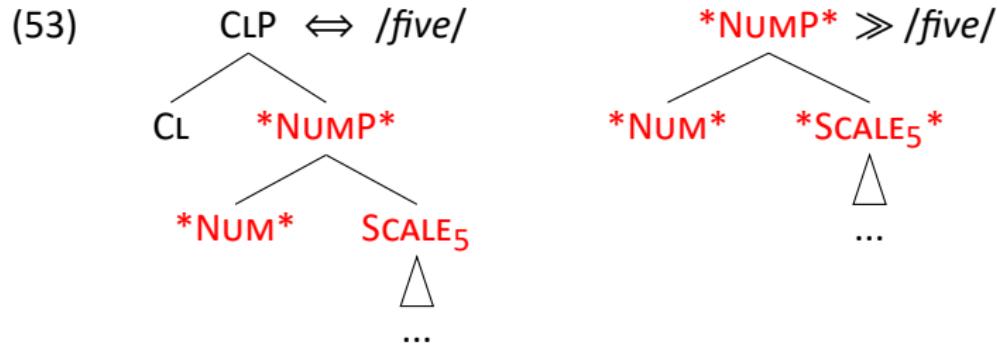
(53) CLP \Leftrightarrow /five/



(54) THE SUPERSET PRINCIPLE (Starke 2009):

A lexically stored tree L matches a syntactic node S iff L contains the syntactic tree dominated by S as a subtree.

Simplex numerals: Syncretism



- (54) THE SUPERSET PRINCIPLE (Starke 2009):
A lexically stored tree L matches a syntactic node S iff L contains the syntactic tree dominated by S as a subtree.

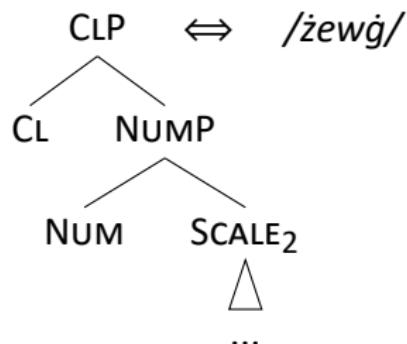
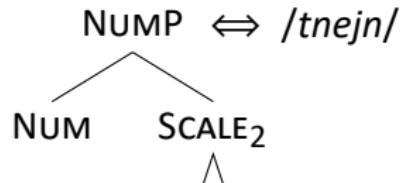
Simplex numerals: Suppletion

	SCALE	NUM	CL
(55)	ENG 5, abstract	<i>five</i>	
	ENG 5, object	<i>five</i>	

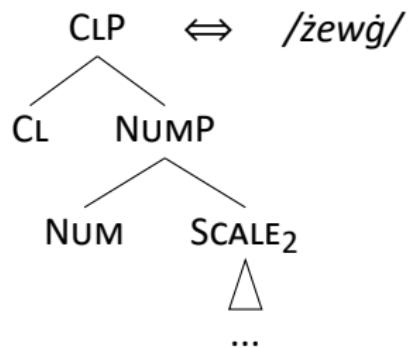
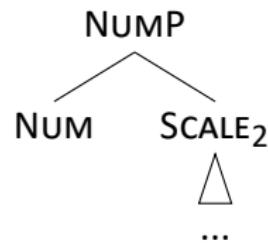
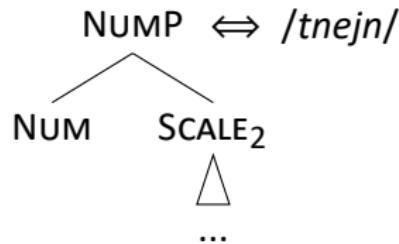
Simplex numerals: Suppletion

		SCALE	NUM	CL
(55)	ENG 5, abstract	<i>five</i>		
	ENG 5, object		<i>five</i>	
	MALT 2, abstract	<i>tnejn</i>		
	MALT 2, object		<i>żewg̊</i>	

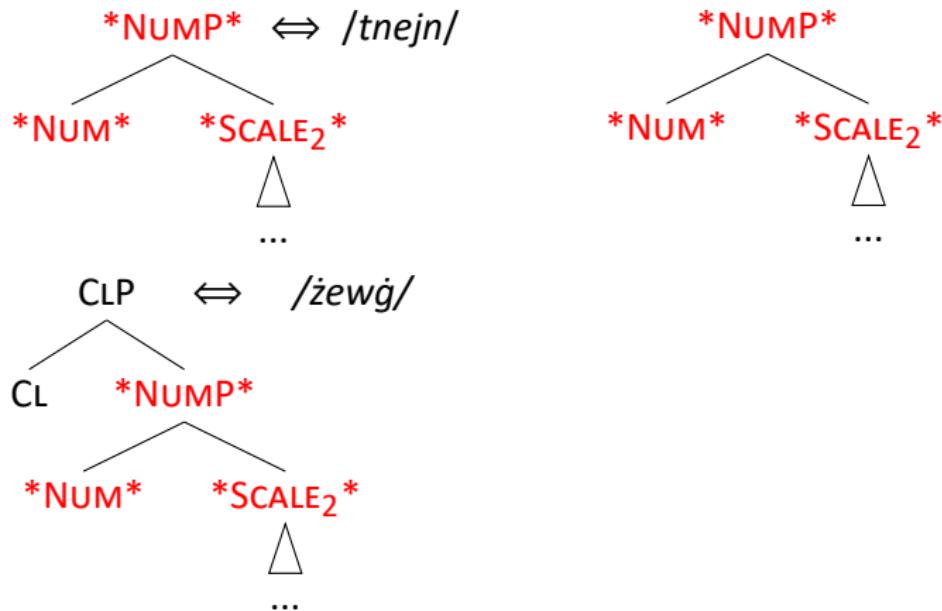
Simplex numerals: Suppletion



Simplex numerals: Suppletion

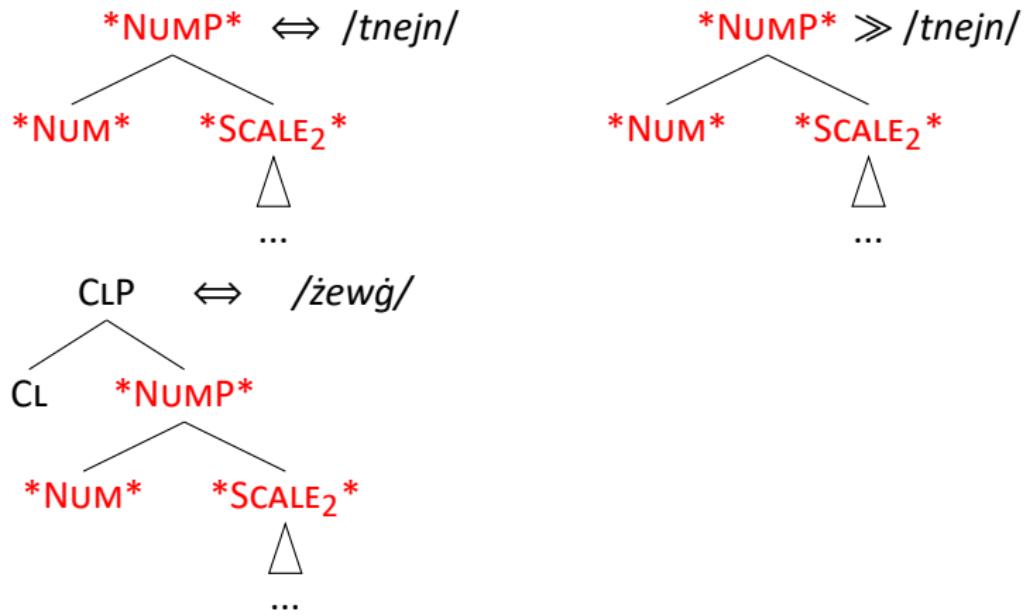


Simplex numerals: Suppletion



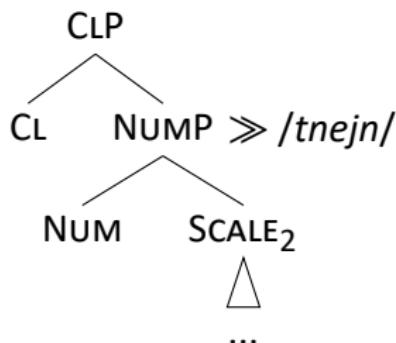
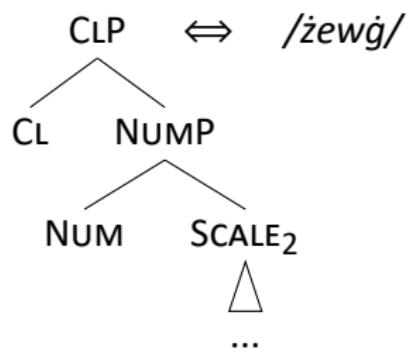
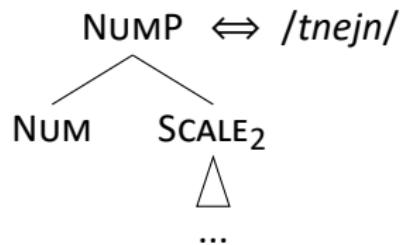
- (56) THE ELSEWHERE CONDITION (Kiparsky 1973):
When multiple items match, chose the more specific one (it has fewer superfluous features).

Simplex numerals: Suppletion



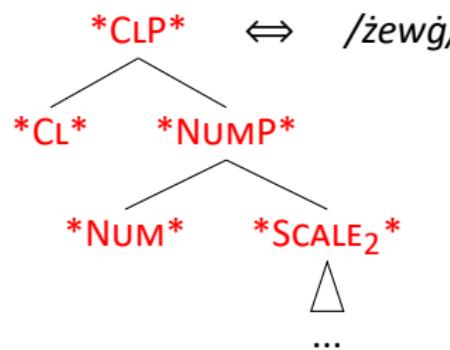
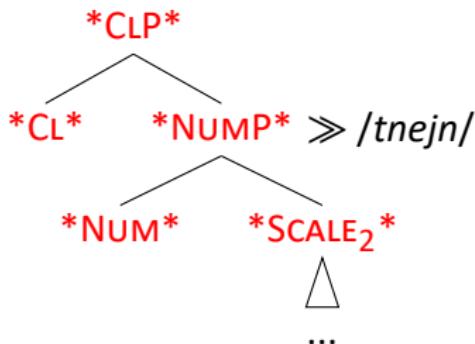
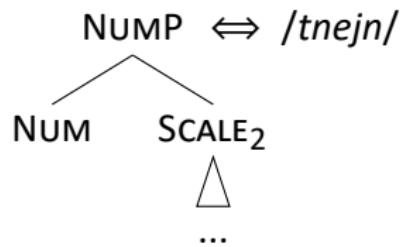
- (56) THE ELSEWHERE CONDITION (Kiparsky 1973):
When multiple items match, chose the more specific one (it has fewer superfluous features).

Simplex numerals: Suppletion



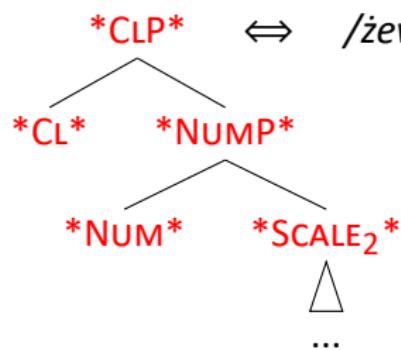
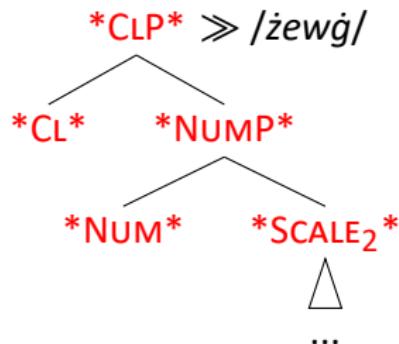
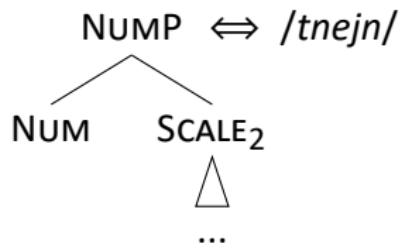
- (56) THE ELSEWHERE CONDITION (Kiparsky 1973):
When multiple items match, chose the more specific one (it has fewer superfluous features).

Simplex numerals: Suppletion



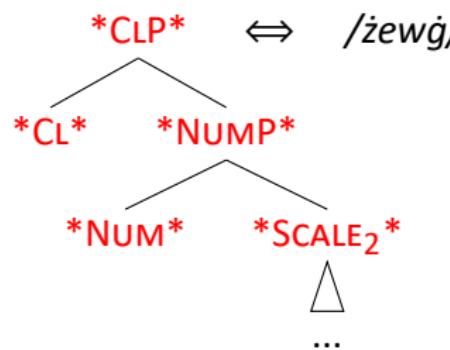
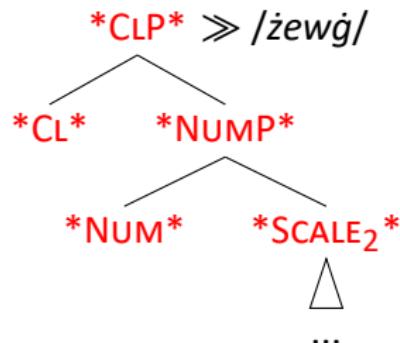
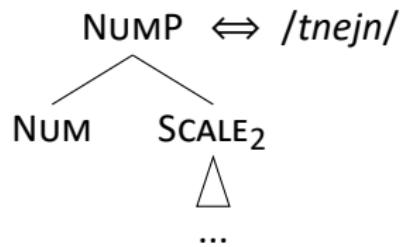
- (56) THE ELSEWHERE CONDITION (Kiparsky 1973):
When multiple items match, chose the more specific one (it has fewer superfluous features).

Simplex numerals: Suppletion



- (56) THE ELSEWHERE CONDITION (Kiparsky 1973):
When multiple items match, chose the more specific one (it has fewer superfluous features).

Simplex numerals: Suppletion

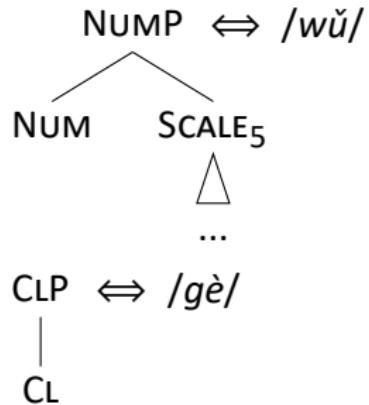


- (56) THE ELSEWHERE CONDITION (Kiparsky 1973):
When multiple items match, chose the more specific one (it has fewer superfluous features).

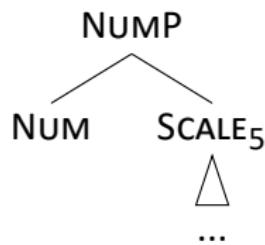
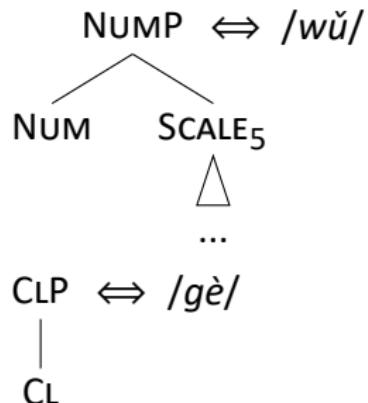
	SCALE	NUM	CL
ENG 5, abs	<i>five</i>		
ENG 5, obj	<i>five</i>		
MALT 2, abs	<i>tnejn</i>		
MALT 2, obj	<i>żewġ</i>		

	SCALE	NUM	CL
ENG 5, abs	<i>five</i>		
ENG 5, obj	<i>five</i>		
MALT 2, abs	<i>tnejn</i>		
MALT 2, obj	<i>żewġ</i>		
MAND 5, abs	<i>wǔ</i>		
MAND 5, obj	<i>wǔ</i>	<i>gè</i>	

Simplex numerals: Stacking

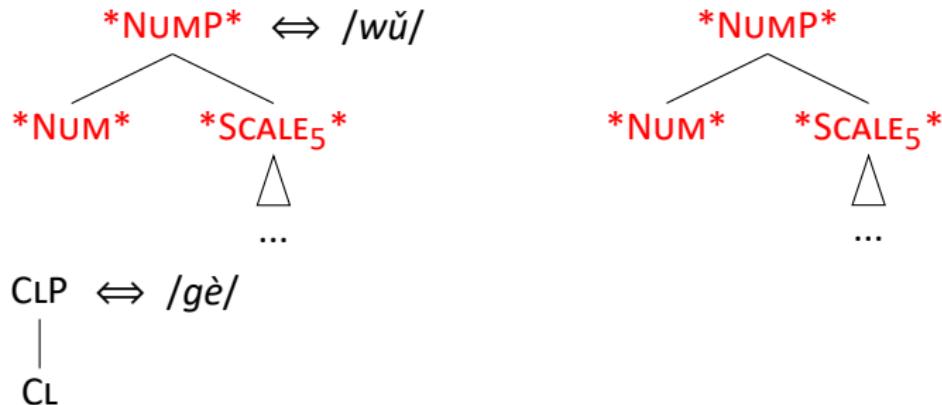


Simplex numerals: Stacking



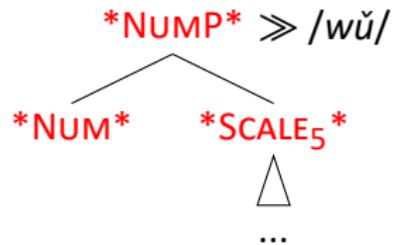
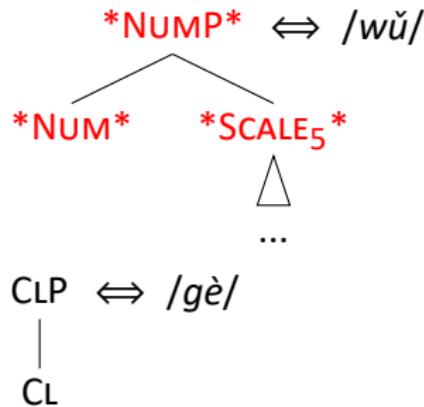
- (57) Spellout algorithm (Starke 2018)
- Merge F and Spell out FP
 - If (a) fails, move the complement of F, and retry (a)

Simplex numerals: Stacking



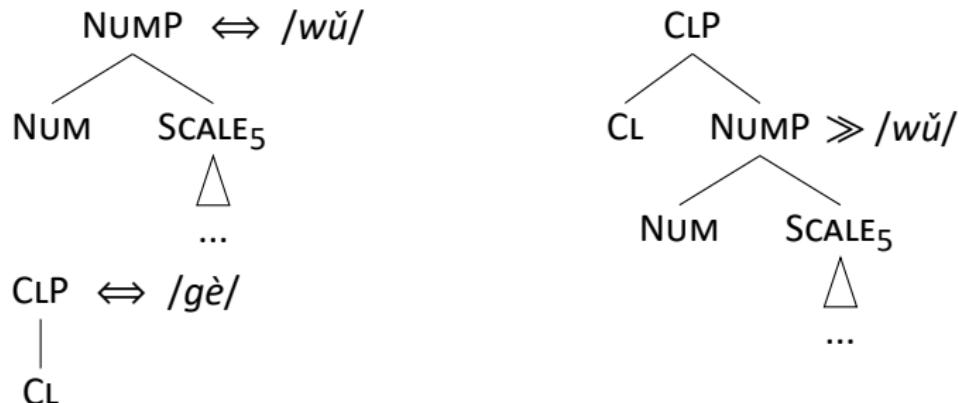
- (57) Spellout algorithm (Starke 2018)
- Merge F and Spell out FP
 - If (a) fails, move the complement of F, and retry (a)

Simplex numerals: Stacking



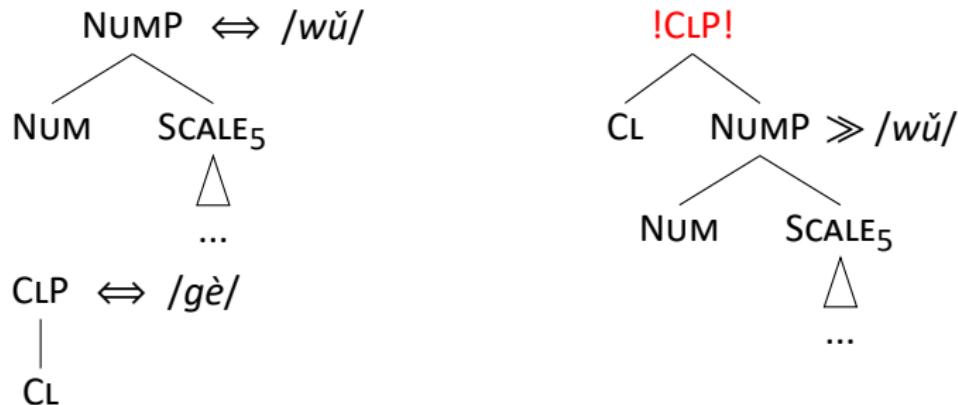
- (57) Spellout algorithm (Starke 2018)
- Merge F and Spell out FP
 - If (a) fails, move the complement of F, and retry (a)

Simplex numerals: Stacking



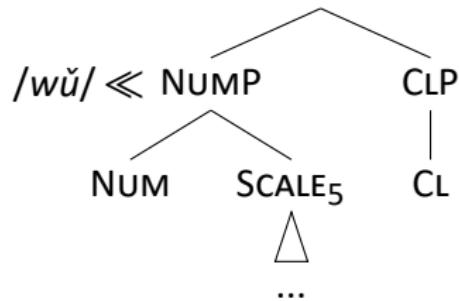
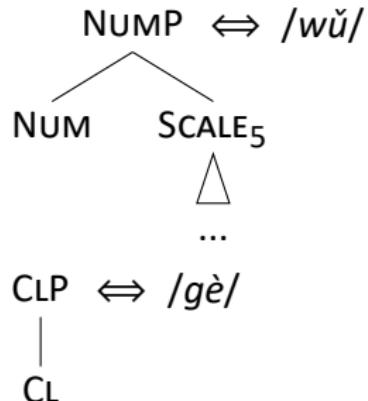
- (57) Spellout algorithm (Starke 2018)
- Merge F and Spell out FP
 - If (a) fails, move the complement of F, and retry (a)

Simplex numerals: Stacking



- (57) Spellout algorithm (Starke 2018)
- Merge F and Spell out FP
 - If (a) fails, move the complement of F, and retry (a)

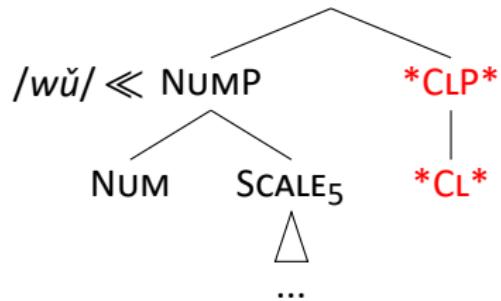
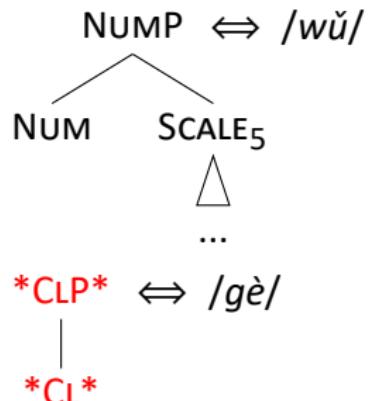
Simplex numerals: Stacking



(57) Spellout algorithm (Starke 2018)

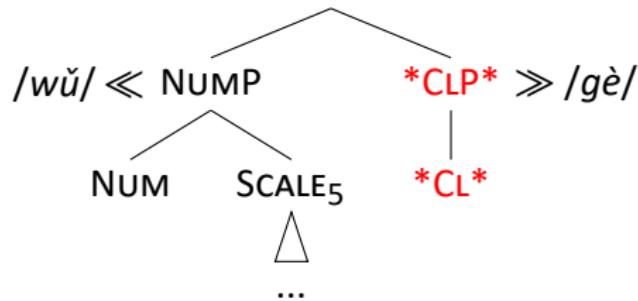
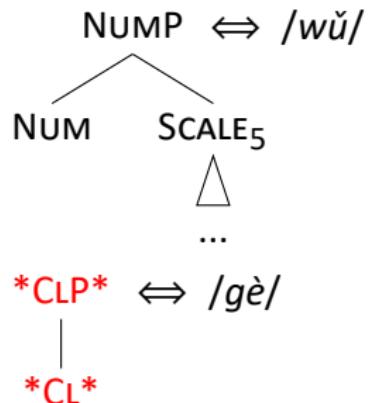
- Merge F and Spell out FP
- If (a) fails, move the complement of F, and retry (a)

Simplex numerals: Stacking



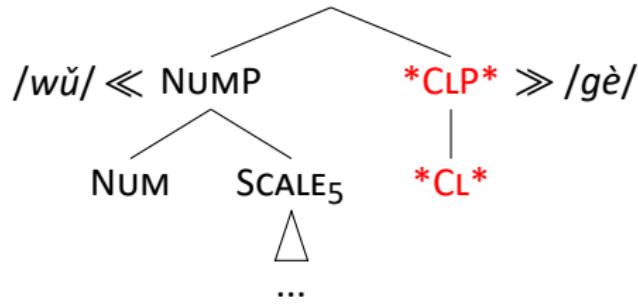
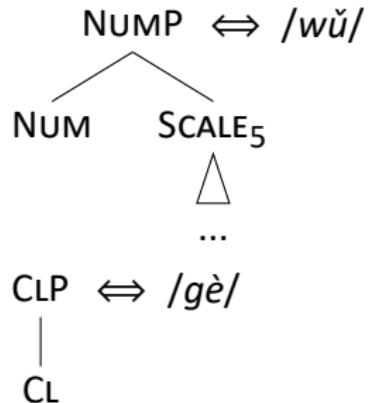
- (57) Spellout algorithm (Starke 2018)
- Merge F and Spell out FP
 - If (a) fails, move the complement of F, and retry (a)

Simplex numerals: Stacking



- (57) Spellout algorithm (Starke 2018)
- Merge F and Spell out FP
 - If (a) fails, move the complement of F, and retry (a)

Simplex numerals: Stacking



- (57) Spellout algorithm (Starke 2018)
- Merge F and Spell out FP
 - If (a) fails, move the complement of F, and retry (a)

	SCALE	NUM	CL
ENG 5, abs	<i>five</i>		
ENG 5, obj	<i>five</i>		
MALT 2, abs	<i>tnejn</i>		
MALT 2, obj	<i>żewǵ</i>		
MAND 5, abs	<i>wě</i>		
MAND 5, obj	<i>wě</i>	<i>gě</i>	

	SCALE	NUM	CL
ENG 5, abs	<i>five</i>		
ENG 5, obj	<i>five</i>		
MALT 2, abs	<i>tnejn</i>		
MALT 2, obj	<i>żewǵ</i>		
MAND 5, abs	<i>wě</i>		
MAND 5, obj	<i>wě</i>	<i>gě</i>	
MAND 2, abs	<i>èr</i>		

	SCALE	NUM	CL
ENG 5, abs	<i>five</i>		
ENG 5, obj	<i>five</i>		
MALT 2, abs	<i>tnejn</i>		
MALT 2, obj	<i>żewǵ</i>		
MAND 5, abs	<i>wě</i>		
MAND 5, obj	<i>wě</i>	<i>gě</i>	
MAND 2, abs	<i>èr</i>		
MAND 2, obj	<i>èr</i>	<i>gě</i>	

	SCALE	NUM	CL
ENG 5, abs	<i>five</i>		
ENG 5, obj	<i>five</i>		
MALT 2, abs	<i>tnejn</i>		
MALT 2, obj	<i>żewǵ</i>		
MAND 5, abs	<i>wě</i>		
MAND 5, obj	<i>wě</i>	<i>gě</i>	
MAND 2, abs	<i>èr</i>		
MAND 2, obj	<i>èr</i>	<i>gě</i>	
MAND 2, abs	<i>èr</i>		

	SCALE	NUM	CL
ENG 5, abs	<i>five</i>		
ENG 5, obj	<i>five</i>		
MALT 2, abs	<i>tnejn</i>		
MALT 2, obj	<i>żewǵ</i>		
MAND 5, abs	<i>wě</i>		
MAND 5, obj	<i>wě</i>	<i>gě</i>	
MAND 2, abs	<i>èr</i>		
MAND 2, obj	<i>èr</i>	<i>gě</i>	
MAND 2, abs	<i>èr</i>		
MAND 2, obj	<i>liǎng</i>	<i>gě</i>	

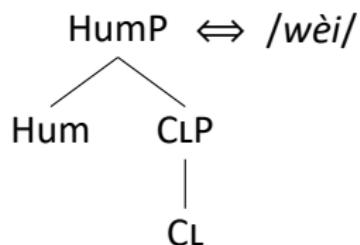
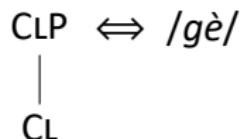
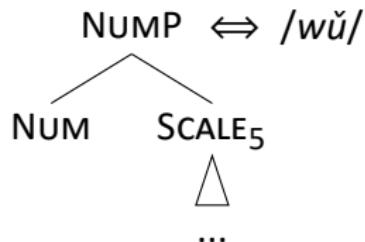
- (58) a. Merge F, Spell out FP
- b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
- c. If (b) fails, move the complement of F, and retry (a)

- (58) a. Merge F, Spell out FP
 b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 c. If (b) fails, move the complement of F, and retry (a)

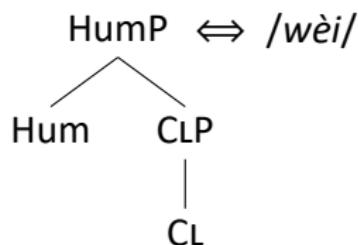
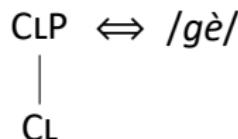
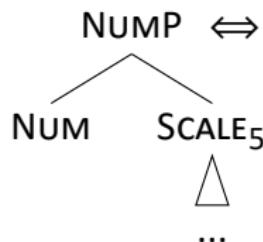
	SCALE	NUM	CL	HUM
--	-------	-----	----	-----

- (58) a. Merge F, Spell out FP
b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
c. If (b) fails, move the complement of F, and retry (a)

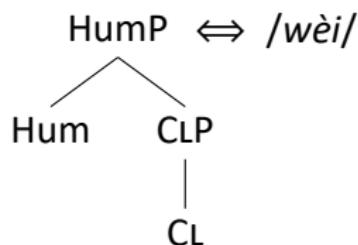
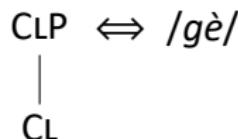
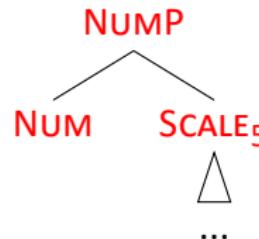
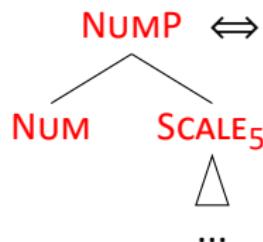
	SCALE	NUM	CL	HUM
MAND 5, abs	<i>wǔ</i>			
MAND 5, obj	<i>wǔ</i>		<i>gè</i>	
MAND 5, hum	<i>wǔ</i>			<i>wèi</i>



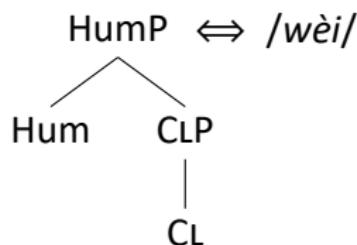
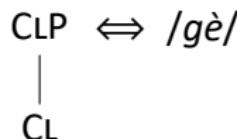
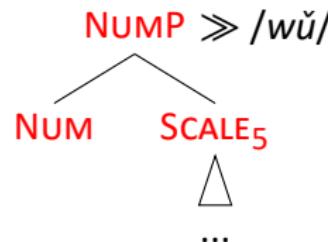
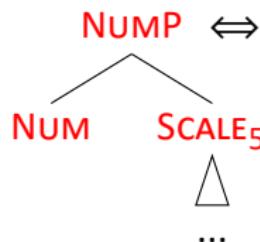
- (59)
- Merge F, Spell out FP
 - If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 - If (b) fails, move the complement of F, and retry (a)



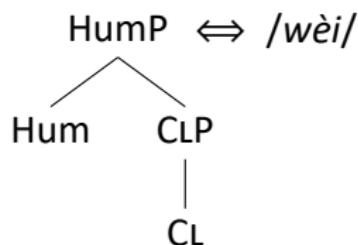
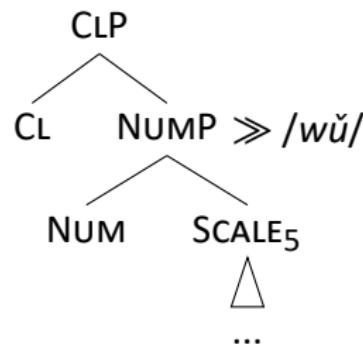
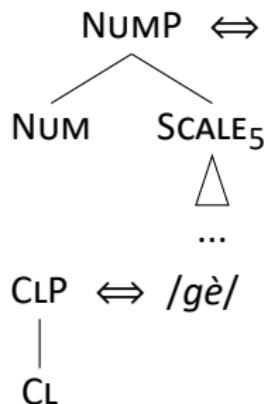
- (60) a. Merge F, Spell out FP
 b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 c. If (b) fails, move the complement of F, and retry (a)



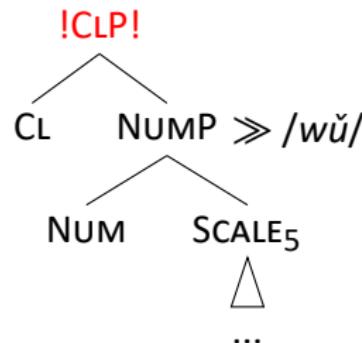
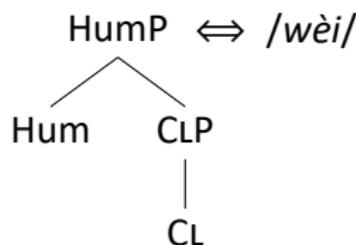
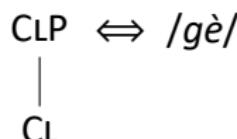
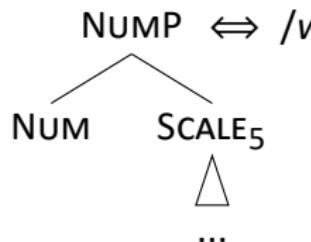
- (61)
- a. Merge F, Spell out FP
 - b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 - c. If (b) fails, move the complement of F, and retry (a)



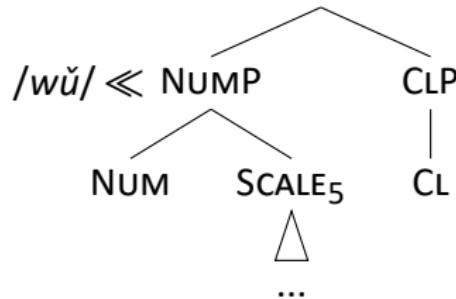
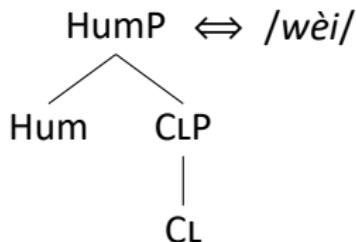
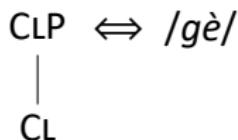
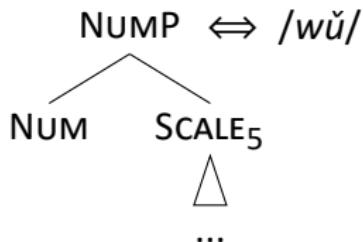
- (62)
- a. Merge F, Spell out FP
 - b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 - c. If (b) fails, move the complement of F, and retry (a)



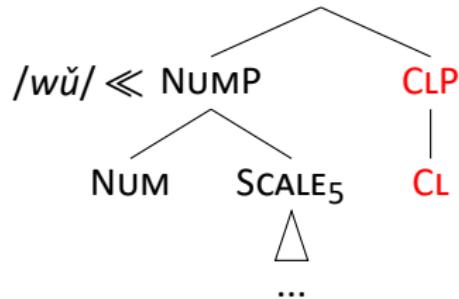
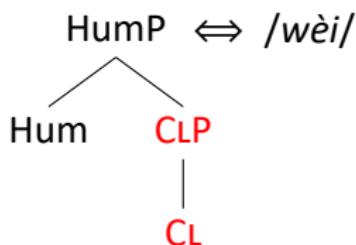
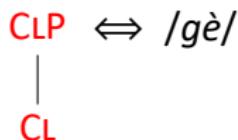
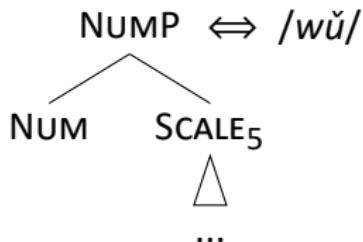
- (63)
- a. Merge F, Spell out FP
 - b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 - c. If (b) fails, move the complement of F, and retry (a)



- (64)
- Merge F, Spell out FP
 - If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 - If (b) fails, move the complement of F, and retry (a)

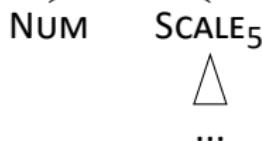


- (65)
- a. Merge F, Spell out FP
 - b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 - c. If (b) fails, move the complement of F, and retry (a)



- (66)
- a. Merge F, Spell out FP
 - b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 - c. If (b) fails, move the complement of F, and retry (a)

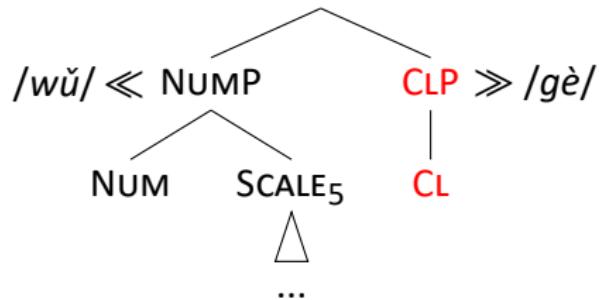
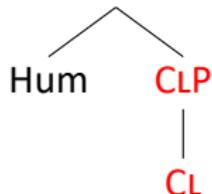
NUMP $\Leftrightarrow /wǔ/$



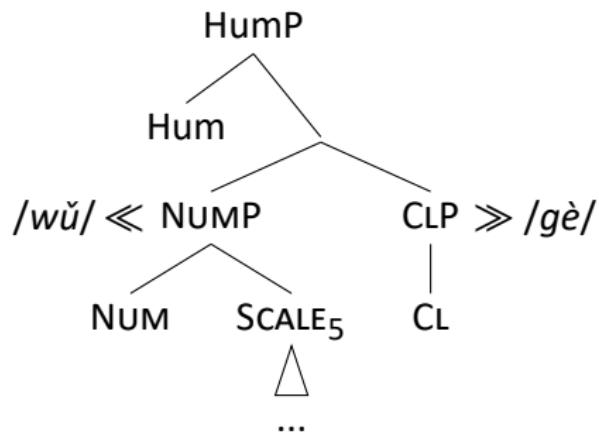
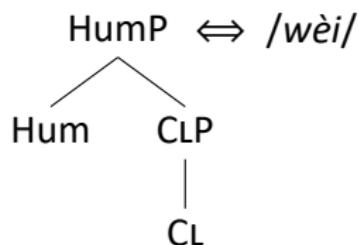
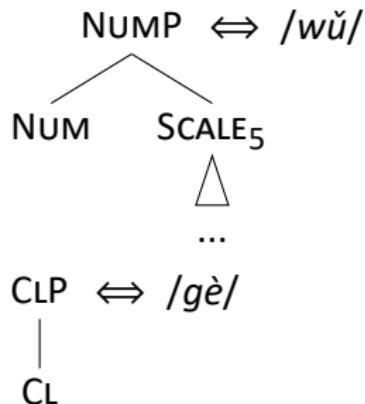
CLP $\Leftrightarrow /gè/$



HumP $\Leftrightarrow /wèi/$

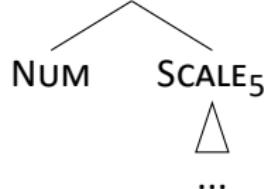


- (67)
- Merge F, Spell out FP
 - If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 - If (b) fails, move the complement of F, and retry (a)



- (68)
- a. Merge F, Spell out FP
 - b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 - c. If (b) fails, move the complement of F, and retry (a)

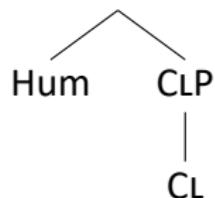
NUMP $\Leftrightarrow /wǔ/$



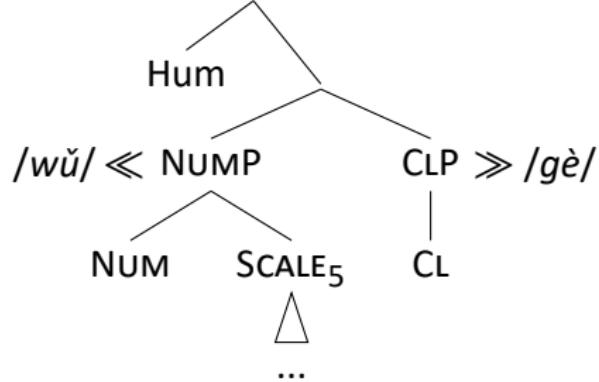
CLP $\Leftrightarrow /gè/$



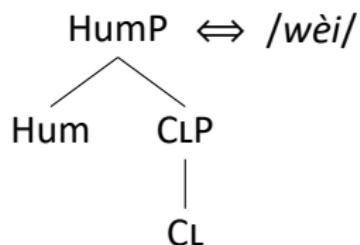
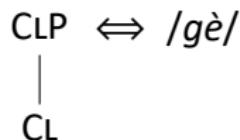
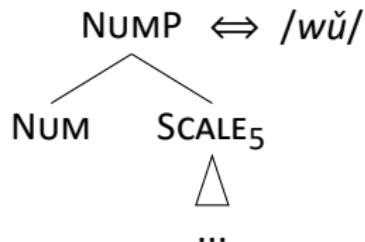
HumP $\Leftrightarrow /wèi/$



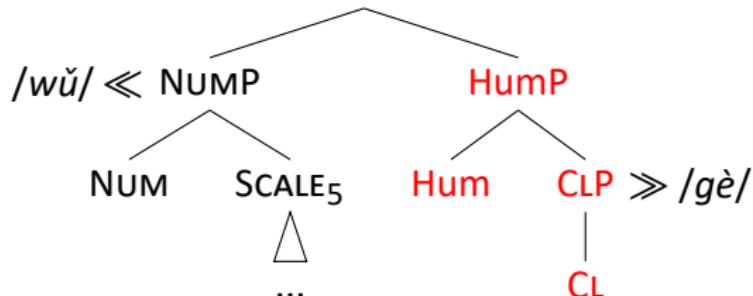
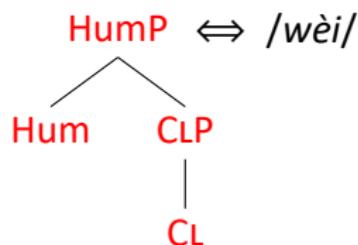
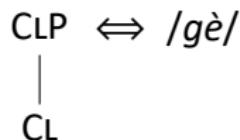
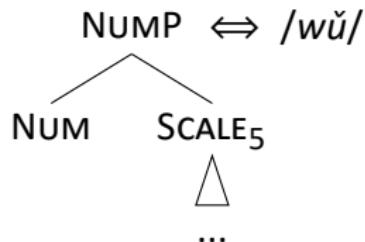
!HumP!



- (69)
- Merge F, Spell out FP
 - If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 - If (b) fails, move the complement of F, and retry (a)

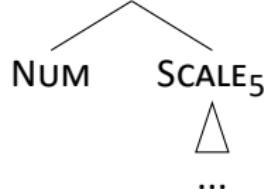


- (70)
- Merge F, Spell out FP
 - If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 - If (b) fails, move the complement of F, and retry (a)



- (71)
- a. Merge F, Spell out FP
 - b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 - c. If (b) fails, move the complement of F, and retry (a)

NUMP $\Leftrightarrow /wǔ/$



CLP $\Leftrightarrow /gè/$



/wǔ/ ≪ NumP

NUM

SCALE₅



...

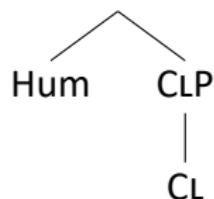
HumP ≫ /wèi/

Hum

CLP



HumP $\Leftrightarrow /wèi/$



- (72)
- Merge F, Spell out FP
 - If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 - If (b) fails, move the complement of F, and retry (a)

	SCALE	NUM	CL	HUM
MAND 5, abs	<i>wǔ</i>			
MAND 5, obj	<i>wǔ</i>		<i>gè</i>	
MAND 5, hum	<i>wǔ</i>			<i>wèi</i>

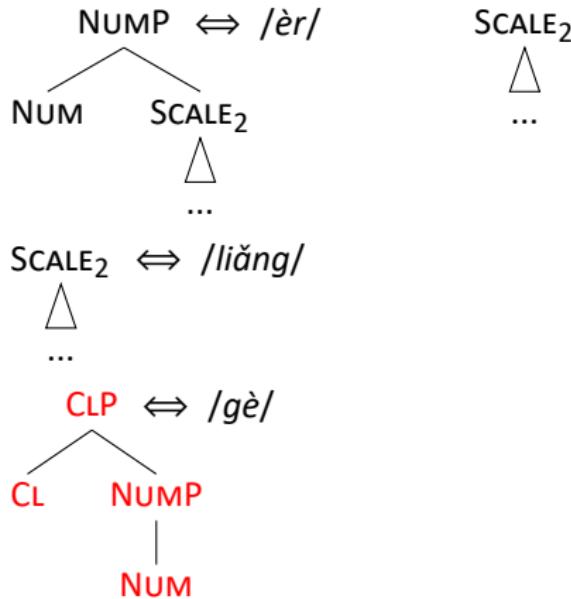
	SCALE	NUM	CL
MAND 2, abs	<i>er</i>		

	SCALE	NUM	CL
MAND 2, abs	<i>èr</i>		
MAND 2, obj	<i>liǎng</i>		<i>gè</i>

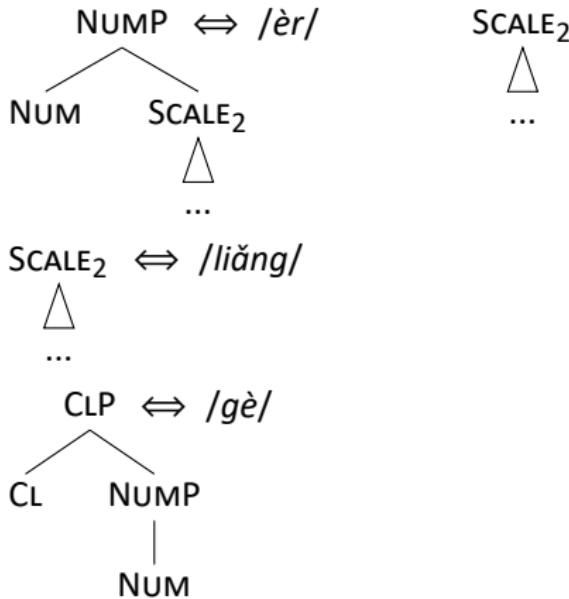
	SCALE	NUM	CL
MAND 2, abs		<i>èr</i>	
MAND 2, obj	<i>liǎng</i>		<i>gè</i>

(73) Spellout algorithm (Starke 2018)

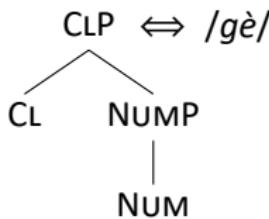
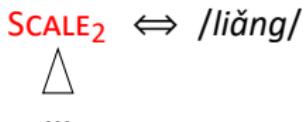
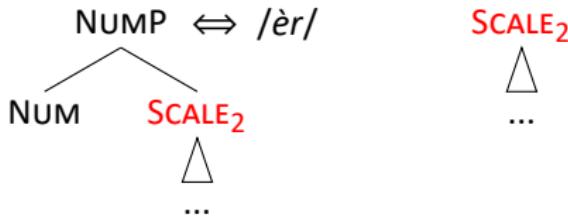
- a. Merge F, Spell out FP
- b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
- c. If (b) fails, move the complement of F, and retry (a)
- d. If (c) fails, go back to the previous cycle and try the next option for that cycle



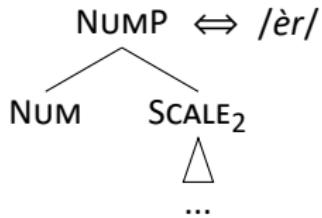
- (74) a. Merge F, Spell out FP
 b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 c. If (b) fails, move the complement of F, and retry (a)
 d. If (c) fails, go back to the previous cycle and try the next option for that cycle



- (75) a. Merge F, Spell out FP
 b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 c. If (b) fails, move the complement of F, and retry (a)
 d. If (c) fails, go back to the previous cycle and try the next option for that cycle



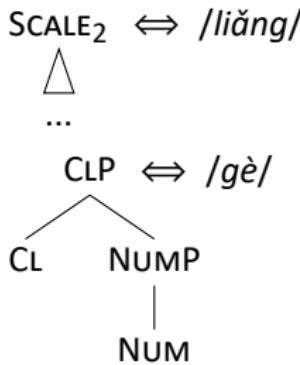
- (76) a. Merge F, Spell out FP
 b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 c. If (b) fails, move the complement of F, and retry (a)
 d. If (c) fails, go back to the previous cycle and try the next option for that cycle



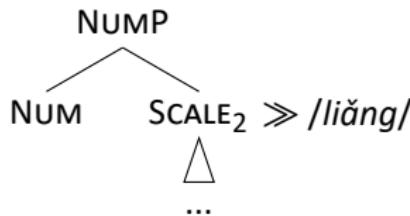
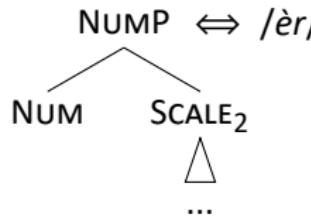
$\text{SCALE}_2 \gg /liǎng/$

```

graph TD
    SCALE2[ $\text{SCALE}_2 \gg /liǎng/$ ] --- triangle3(( ))
    triangle3 --- triangle4(( ))
    triangle4 --- dots3[...]
  
```



- (77) a. Merge F, Spell out FP
 b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 c. If (b) fails, move the complement of F, and retry (a)
 d. If (c) fails, go back to the previous cycle and try the next option for that cycle



SCALE2 $\Leftrightarrow /liǎng/$

```

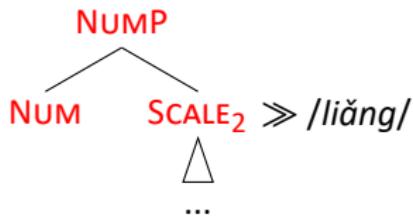
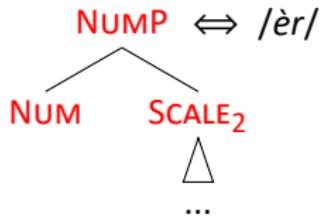
graph TD
    SCALE2[SCALE2] --- triangle1(( ))
    triangle1 --- triangle2(( ))
    ...
  
```

CLP $\Leftrightarrow /gè/$

```

graph TD
    CLP[CLP] --- CL[CL]
    CLP --- NUMP[NUMP]
    NUMP --- NUM[NUM]
  
```

- (78) a. Merge F, Spell out FP
 b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 c. If (b) fails, move the complement of F, and retry (a)
 d. If (c) fails, go back to the previous cycle and try the next option for that cycle



SCALE₂ $\Leftrightarrow /liǎng/$

```

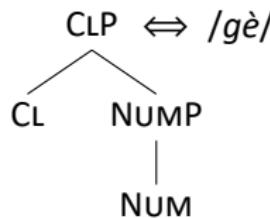
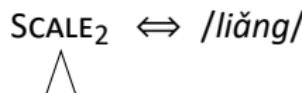
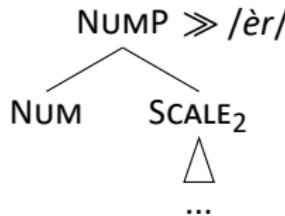
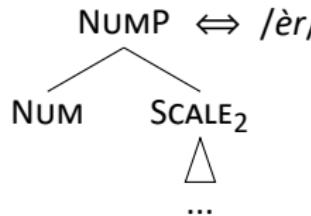
graph TD
    SCALE2[SCALE2] --- triangle3(( ))
    SCALE2 --- dots3[...]
    
```

CLP $\Leftrightarrow /gè/$

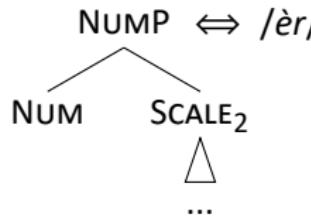
```

graph TD
    CLP[CLP] --- CL[CL]
    CLP --- NUMP[NUMP]
    NUMP --- NUM[NUM]
    CL --- dots4[...]
    
```

- (79) a. Merge F, Spell out FP
 b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 c. If (b) fails, move the complement of F, and retry (a)
 d. If (c) fails, go back to the previous cycle and try the next option for that cycle



- (80) a. Merge F, Spell out FP
 b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 c. If (b) fails, move the complement of F, and retry (a)
 d. If (c) fails, go back to the previous cycle and try the next option for that cycle



SCALE₂ $\Leftrightarrow /liǎng/$

```

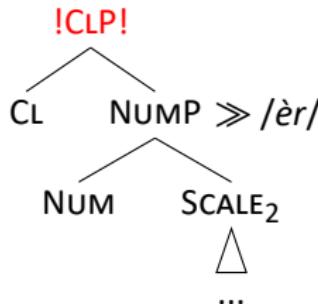
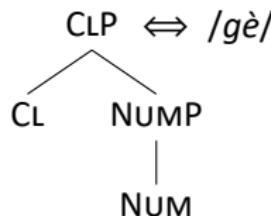
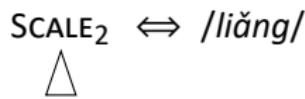
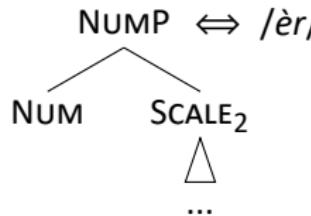
graph TD
    SCALE2[SCALE2] --- NUM[NUM]
    SCALE2 --- SCALE2_2[SCALE2]
    NUM --- triangle1(( ))
    SCALE2_2 --- triangle2(( ))
    ...
  
```

CLP $\Leftrightarrow /gè/$

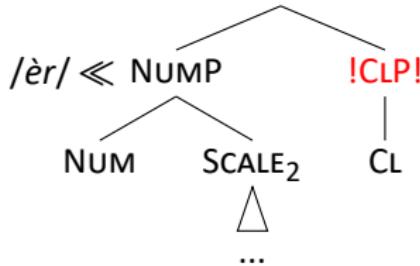
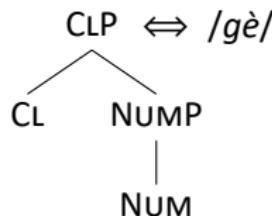
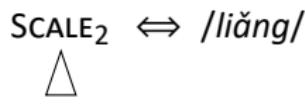
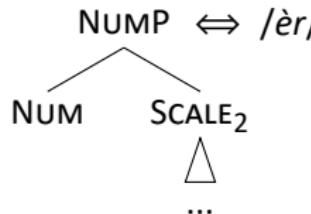
```

graph TD
    CLP[CLP] --- CL[CL]
    CLP --- NUMP[NUMP]
    NUMP --- NUM[NUM]
    CL --- triangle1(( ))
    NUMP --- triangle2(( ))
    ...
  
```

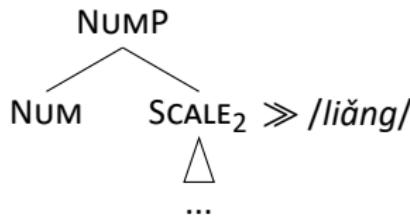
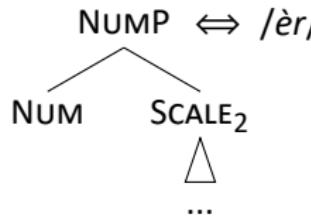
- (81) a. Merge F, Spell out FP
 b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 c. If (b) fails, move the complement of F, and retry (a)
 d. If (c) fails, go back to the previous cycle and try the next option for that cycle



- (82) a. Merge F, Spell out FP
 b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 c. If (b) fails, move the complement of F, and retry (a)
 d. If (c) fails, go back to the previous cycle and try the next option for that cycle



- (83)
- Merge F, Spell out FP
 - If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 - If (b) fails, move the complement of F, and retry (a)
 - If (c) fails, go back to the previous cycle and try the next option for that cycle



SCALE2 $\Leftrightarrow /liǎng/$

```

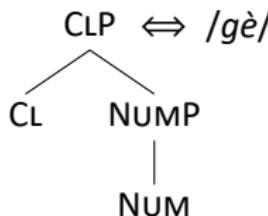
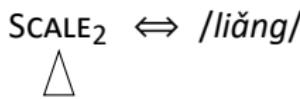
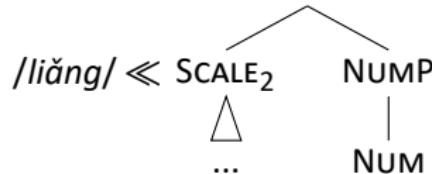
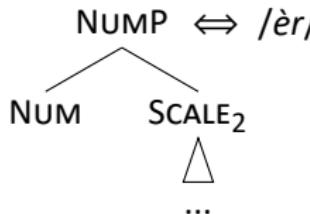
graph TD
    SCALE2[SCALE2] --- triangle1(( ))
    triangle2(( ))
    ...
  
```

CLP $\Leftrightarrow /gè/$

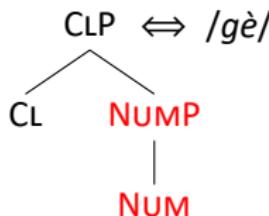
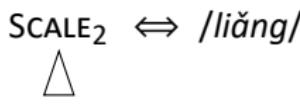
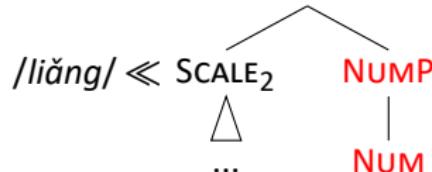
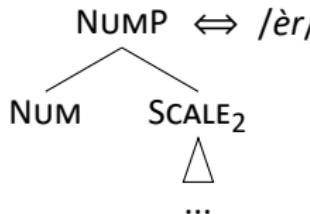
```

graph TD
    CLP[CLP] --- CL[CL]
    CLP --- NUMP[NUMP]
    NUMP --- triangle1(( ))
    NUMP --- triangle2(( ))
    ...
  
```

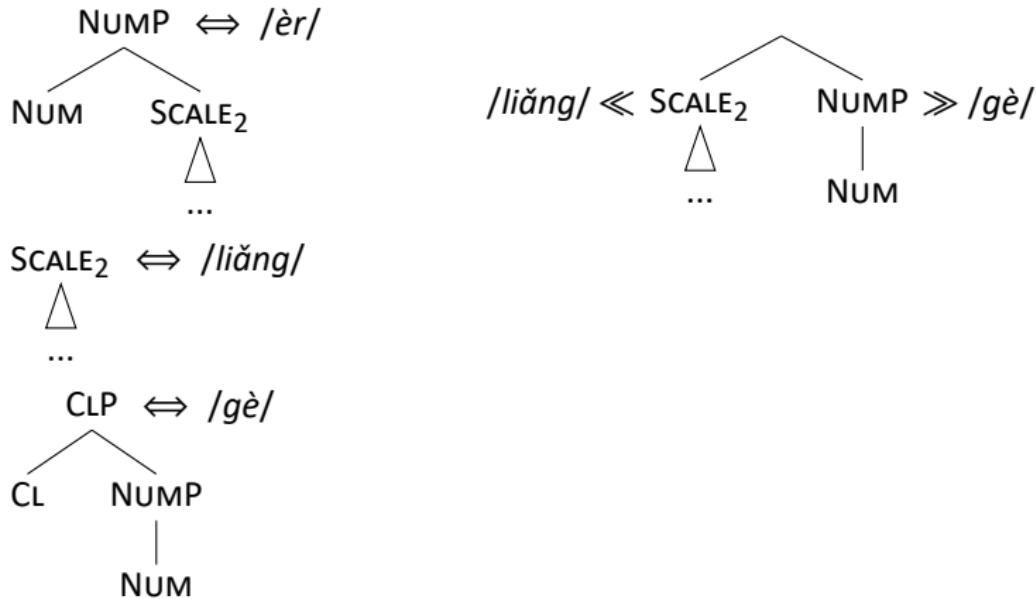
- (84) a. Merge F, Spell out FP
 b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 c. If (b) fails, move the complement of F, and retry (a)
 d. If (c) fails, go back to the previous cycle and try the next option for that cycle



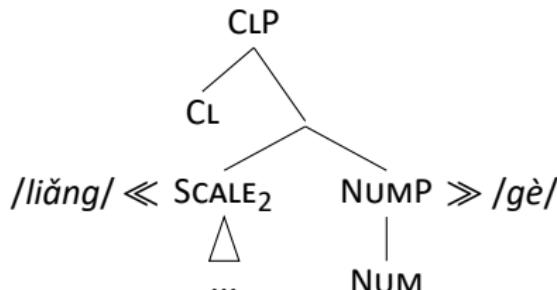
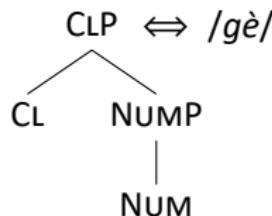
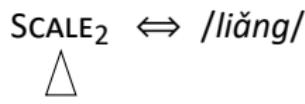
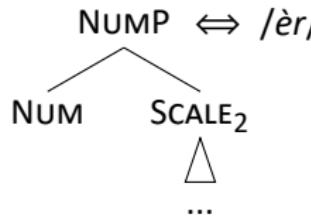
- (85) a. Merge F, Spell out FP
 b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 c. If (b) fails, move the complement of F, and retry (a)
 d. If (c) fails, go back to the previous cycle and try the next option for that cycle



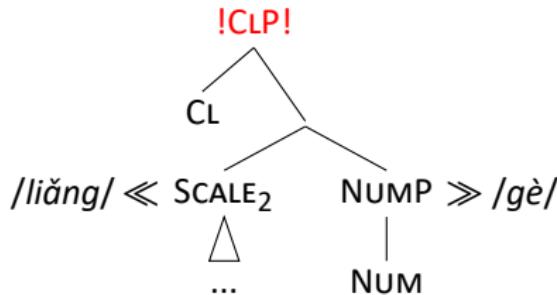
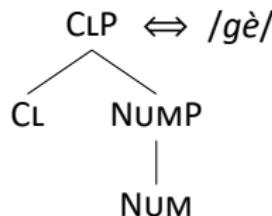
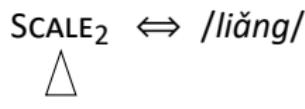
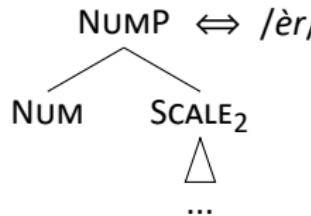
- (86) a. Merge F, Spell out FP
 b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 c. If (b) fails, move the complement of F, and retry (a)
 d. If (c) fails, go back to the previous cycle and try the next option for that cycle



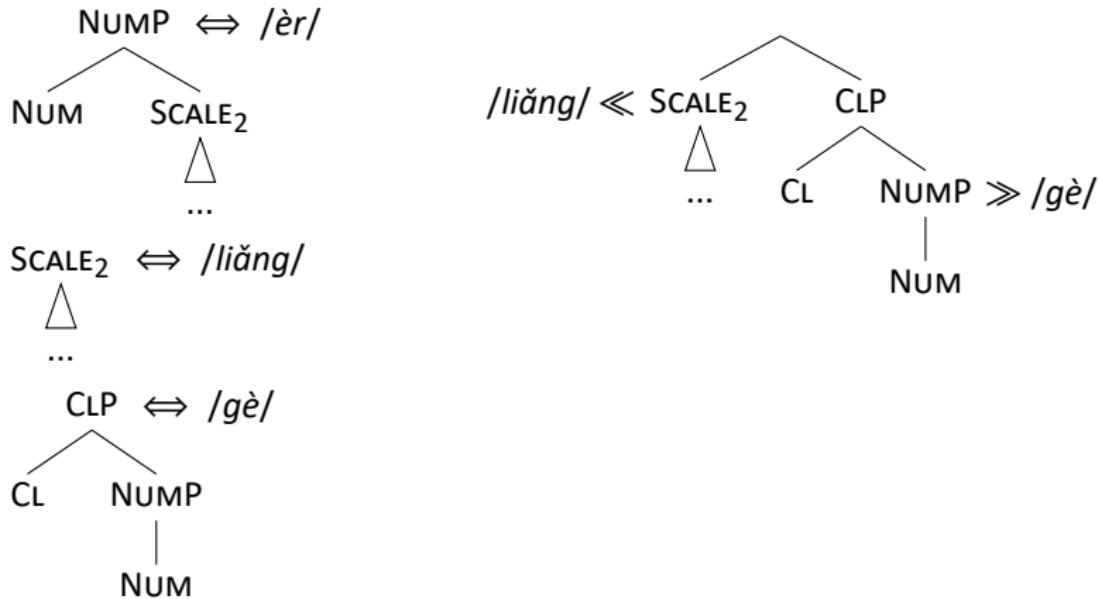
- (87) a. Merge F, Spell out FP
 b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 c. If (b) fails, move the complement of F, and retry (a)
 d. If (c) fails, go back to the previous cycle and try the next option for that cycle



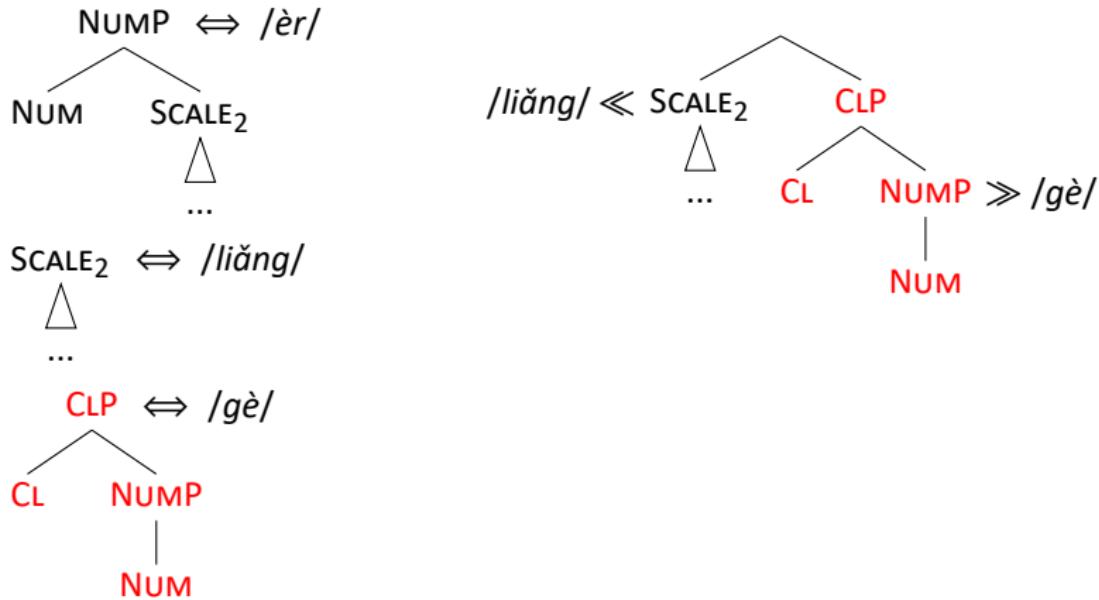
- (88)
- Merge F, Spell out FP
 - If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 - If (b) fails, move the complement of F, and retry (a)
 - If (c) fails, go back to the previous cycle and try the next option for that cycle



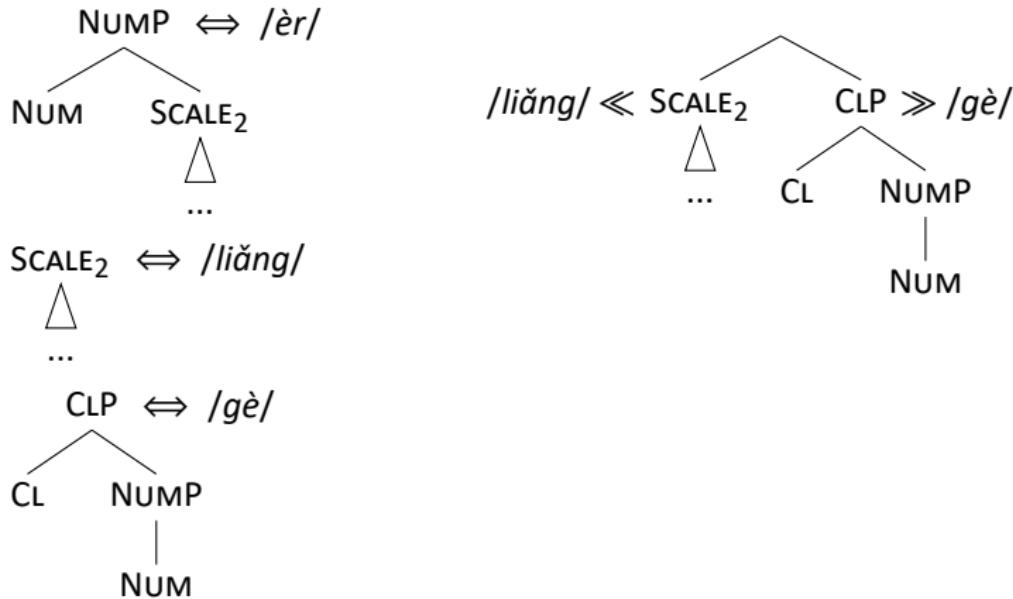
- (89) a. Merge F, Spell out FP
 b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 c. If (b) fails, move the complement of F, and retry (a)
 d. If (c) fails, go back to the previous cycle and try the next option for that cycle



- (90) a. Merge F, Spell out FP
 b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 c. If (b) fails, move the complement of F, and retry (a)
 d. If (c) fails, go back to the previous cycle and try the next option for that cycle



- (91)
- a. Merge F, Spell out FP
 - b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 - c. If (b) fails, move the complement of F, and retry (a)
 - d. If (c) fails, go back to the previous cycle and try the next option for that cycle



- (92) a. Merge F, Spell out FP
 b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 c. If (b) fails, move the complement of F, and retry (a)
 d. If (c) fails, go back to the previous cycle and try the next option for that cycle

	SCALE	NUM	CL
MAND 2, abs		<i>èr</i>	

	SCALE	NUM	CL
MAND 2, abs		èr	
MAND 2, obj	liǎng		gè

	SCALE	NUM	CL
MAND 2, abs	<i>èr</i>		
MAND 2, obj	<i>liǎng</i>	<i>gè</i>	
MAND 5, abs	<i>wǔ</i>		
MAND 5, obj	<i>wǔ</i>	<i>gè</i>	

	SCALE	NUM	CL	HUM

	SCALE	NUM	CL	HUM
MAND 5, abs		<i>wǔ</i>		

	SCALE	NUM	CL	HUM
MAND 5, abs		wǔ		
MAND 5, obj	wǔ		gè	

	SCALE	NUM	CL	HUM
MAND 5, abs	<i>wǔ</i>			
MAND 5, obj	<i>wǔ</i>	<i>gè</i>		
MAND 5, hum	<i>wǔ</i>		<i>wèi</i>	

	SCALE	NUM	CL	HUM
MAND 5, abs		wǔ		
MAND 5, obj	wǔ		gè	
MAND 5, hum	wǔ			wèi
MAND 2, abs		èr		

	SCALE	NUM	CL	HUM
MAND 5, abs		wǔ		
MAND 5, obj	wǔ		gè	
MAND 5, hum	wǔ		wèi	
MAND 2, abs		èr		
MAND 2, obj	liǎng		gè	

	SCALE	NUM	CL	HUM
MAND 5, abs		wǔ		
MAND 5, obj	wǔ		gè	
MAND 5, hum	wǔ			wèi
MAND 2, abs		èr		
MAND 2, obj	liǎng		gè	
MAND 2, hum	liǎng			wèi

	SCALE	NUM	CL	HUM
MAND 5, abs	<i>wǔ</i>			
MAND 5, obj	<i>wǔ</i>	<i>gè</i>		
MAND 5, hum	<i>wǔ</i>		<i>wèi</i>	
MAND 2, abs	<i>èr</i>			
MAND 2, obj	<i>liǎng</i>	<i>gè</i>		
MAND 2, hum	<i>liǎng</i>		<i>wèi</i>	
REG1 2, abs	<i>èr</i>			

	SCALE	NUM	CL	HUM
MAND 5, abs		wǔ		
MAND 5, obj	wǔ		gè	
MAND 5, hum	wǔ			wèi
MAND 2, abs		èr		
MAND 2, obj	liǎng		gè	
MAND 2, hum	liǎng			wèi
REG1 2, abs		èr		
REG1 2, obj		liǎ		

	SCALE	NUM	CL	HUM
MAND 5, abs		wǔ		
MAND 5, obj	wǔ		gè	
MAND 5, hum	wǔ			wèi
MAND 2, abs		èr		
MAND 2, obj	liǎng		gè	
MAND 2, hum	liǎng			wèi
REG1 2, abs		èr		
REG1 2, obj		liǎ		
REG1 2, hum			liǎ	

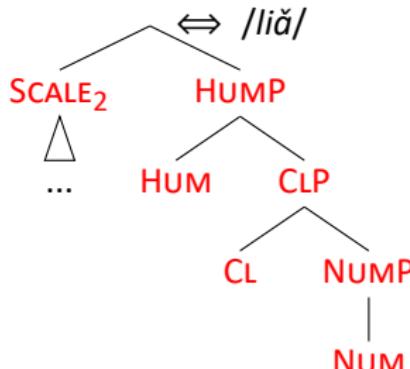
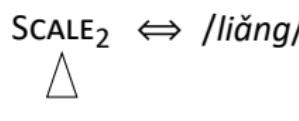
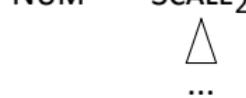
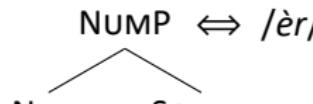
	SCALE	NUM	CL	HUM
MAND 5, abs		wǔ		
MAND 5, obj	wǔ		gè	
MAND 5, hum	wǔ			wèi
MAND 2, abs		èr		
MAND 2, obj	liǎng		gè	
MAND 2, hum	liǎng			wèi
REG1 2, abs		èr		
REG1 2, obj		liǎ		
REG1 2, hum			liǎ	

	SCALE	NUM	CL	HUM
MAND 5, abs	<i>wǔ</i>			
MAND 5, obj	<i>wǔ</i>	<i>gè</i>		
MAND 5, hum	<i>wǔ</i>		<i>wèi</i>	
MAND 2, abs	<i>èr</i>			
MAND 2, obj	<i>liǎng</i>	<i>gè</i>		
MAND 2, hum	<i>liǎng</i>		<i>wèi</i>	
REG1 2, abs	<i>èr</i>			
REG1 2, obj	<i>liǎ</i>			
REG1 2, hum		<i>liǎ</i>		
REG2 2, abs	<i>èr</i>			

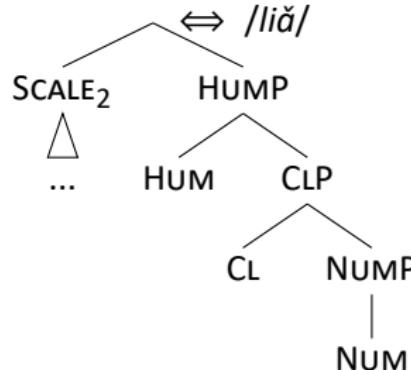
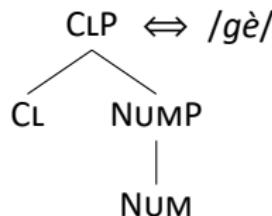
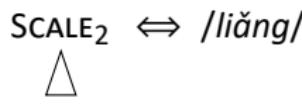
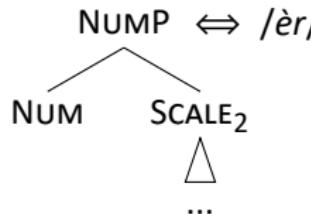
	SCALE	NUM	CL	HUM
MAND 5, abs		wǔ		
MAND 5, obj	wǔ		gè	
MAND 5, hum	wǔ			wèi
MAND 2, abs		èr		
MAND 2, obj	liǎng		gè	
MAND 2, hum	liǎng			wèi
REG1 2, abs		èr		
REG1 2, obj		liǎ		
REG1 2, hum			liǎ	
REG2 2, abs		èr		
REG2 2, obj	liǎng		gè	

	SCALE	NUM	CL	HUM
MAND 5, abs	<i>wǔ</i>			
MAND 5, obj	<i>wǔ</i>	<i>gè</i>		
MAND 5, hum	<i>wǔ</i>		<i>wèi</i>	
MAND 2, abs	<i>èr</i>			
MAND 2, obj	<i>liǎng</i>	<i>gè</i>		
MAND 2, hum	<i>liǎng</i>		<i>wèi</i>	
REG1 2, abs	<i>èr</i>			
REG1 2, obj	<i>liǎ</i>			
REG1 2, hum		<i>liǎ</i>		
REG2 2, abs	<i>èr</i>			
REG2 2, obj	<i>liǎng</i>	<i>gè</i>		
REG2 2, hum		<i>liǎ</i>		

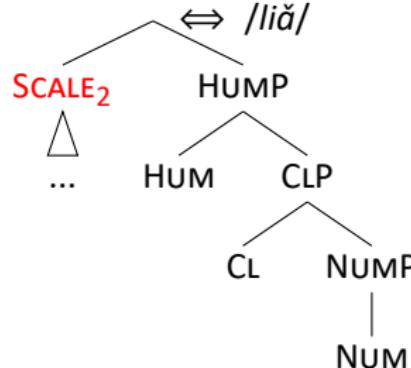
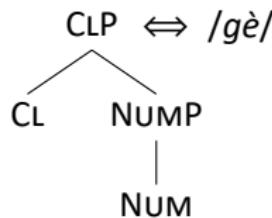
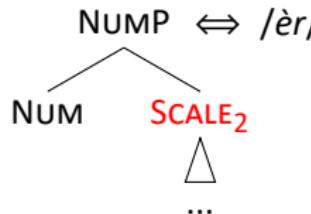
Movement-containing trees (Blix 2021)



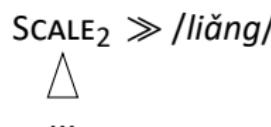
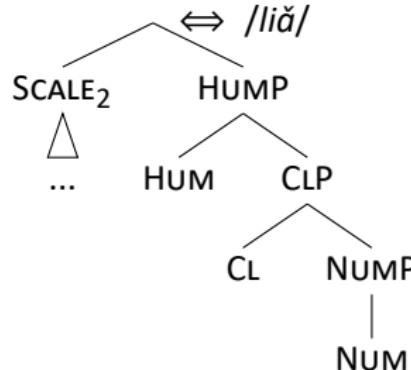
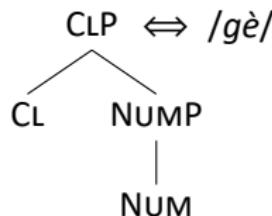
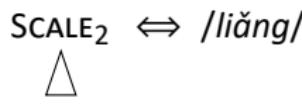
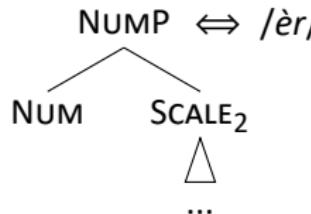
- (93) a. Merge F, Spell out FP
b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
c. If (b) fails, move the complement of F, and retry (a)
d. If (c) fails, go back to the previous cycle and try the next option for that cycle



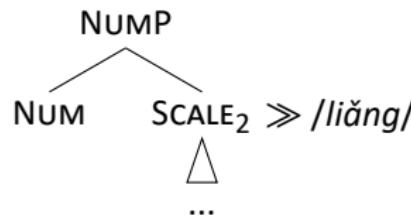
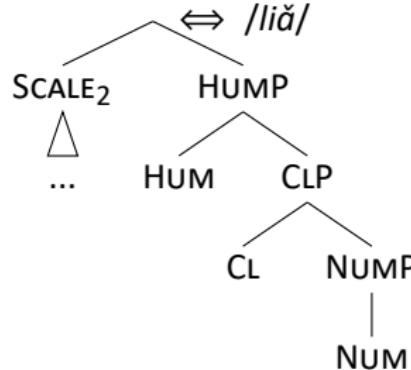
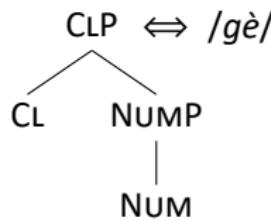
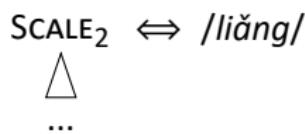
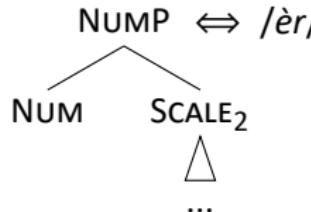
- (94) a. Merge F, Spell out FP
 b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 c. If (b) fails, move the complement of F, and retry (a)
 d. If (c) fails, go back to the previous cycle and try the next option for that cycle



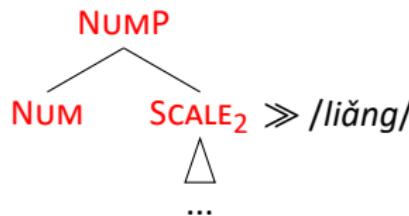
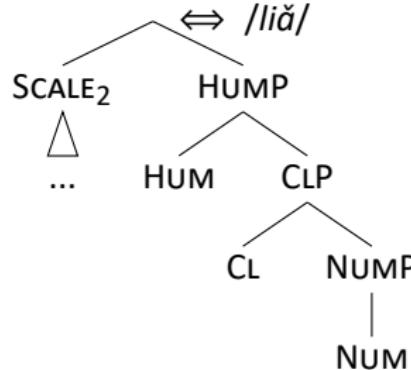
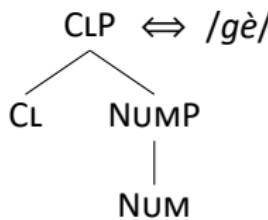
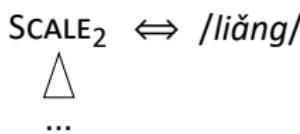
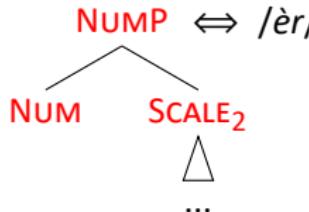
- (95) a. Merge F, Spell out FP
 b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 c. If (b) fails, move the complement of F, and retry (a)
 d. If (c) fails, go back to the previous cycle and try the next option for that cycle



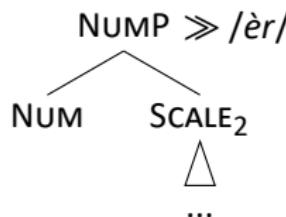
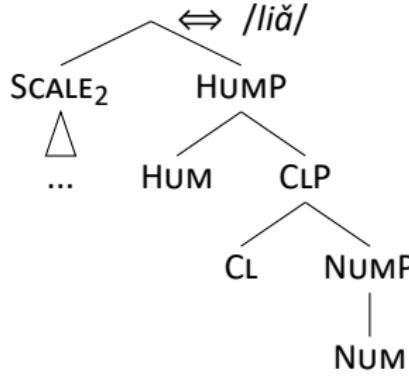
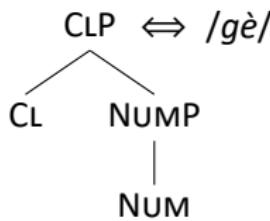
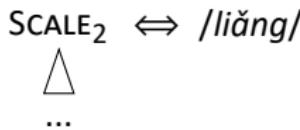
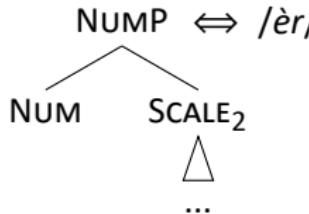
- (96) a. Merge F, Spell out FP
 b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 c. If (b) fails, move the complement of F, and retry (a)
 d. If (c) fails, go back to the previous cycle and try the next option for that cycle



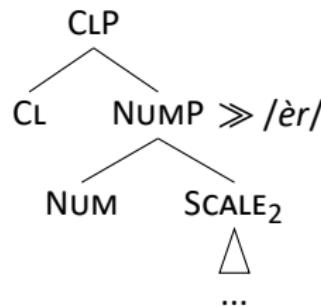
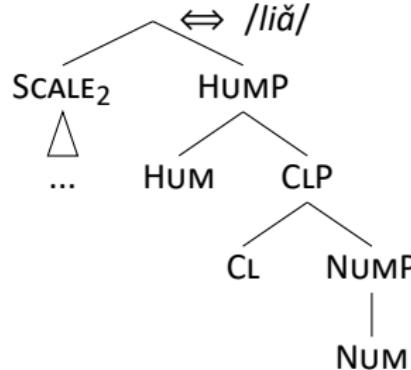
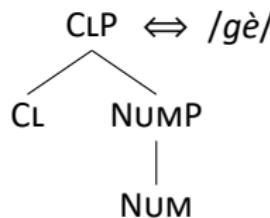
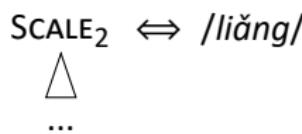
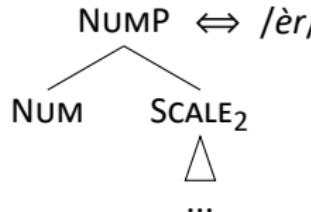
- (97)
- Merge F, Spell out FP
 - If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 - If (b) fails, move the complement of F, and retry (a)
 - If (c) fails, go back to the previous cycle and try the next option for that cycle



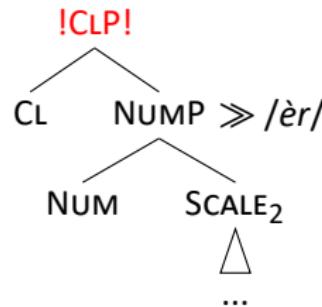
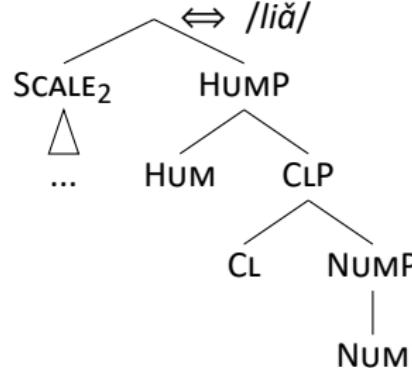
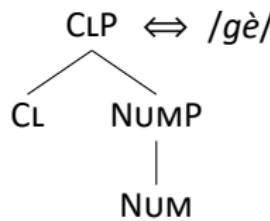
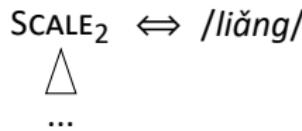
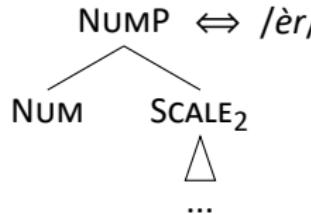
- (98)
- Merge F, Spell out FP
 - If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 - If (b) fails, move the complement of F, and retry (a)
 - If (c) fails, go back to the previous cycle and try the next option for that cycle



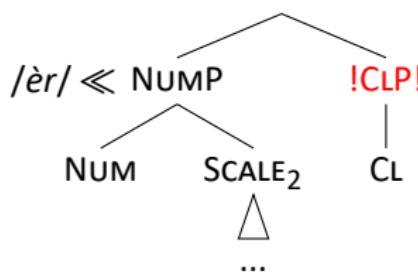
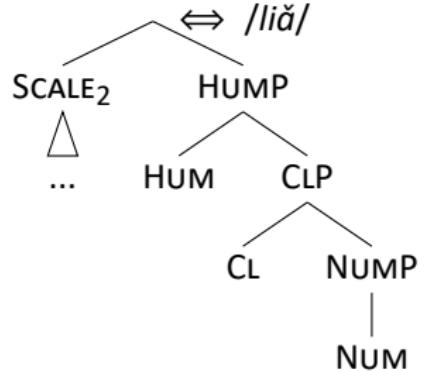
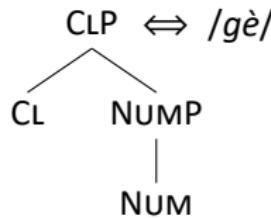
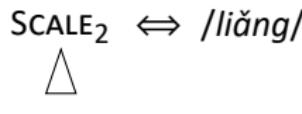
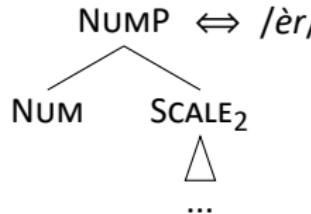
- (99)
- Merge F, Spell out FP
 - If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 - If (b) fails, move the complement of F, and retry (a)
 - If (c) fails, go back to the previous cycle and try the next option for that cycle



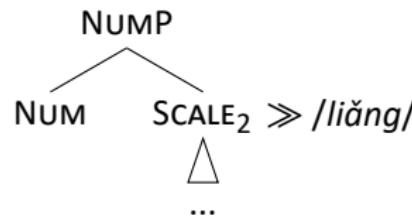
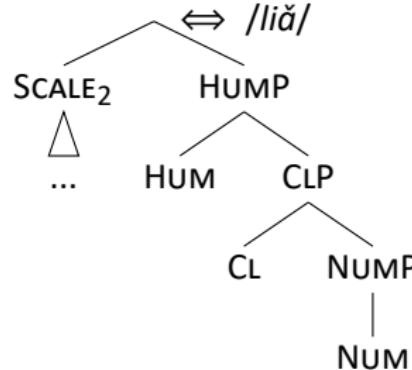
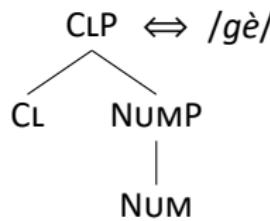
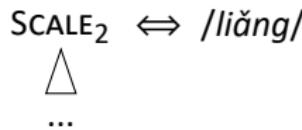
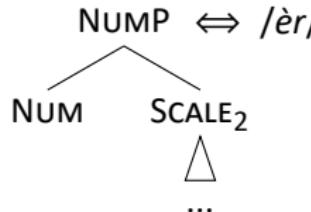
- (100)
- Merge F, Spell out FP
 - If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 - If (b) fails, move the complement of F, and retry (a)
 - If (c) fails, go back to the previous cycle and try the next



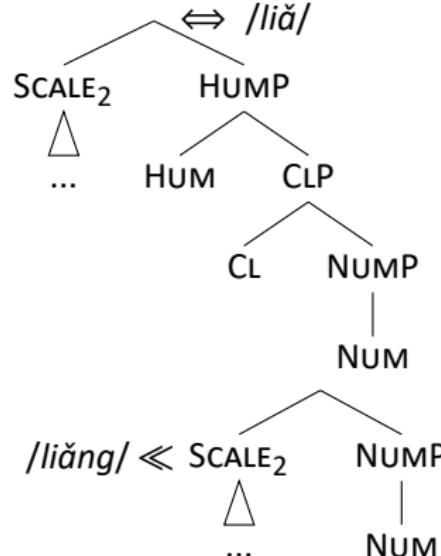
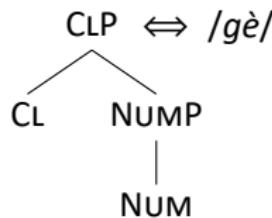
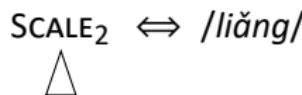
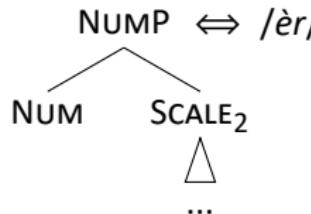
- (101)
- Merge F, Spell out FP
 - If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 - If (b) fails, move the complement of F, and retry (a)
 - If (c) fails, go back to the previous cycle and try the next



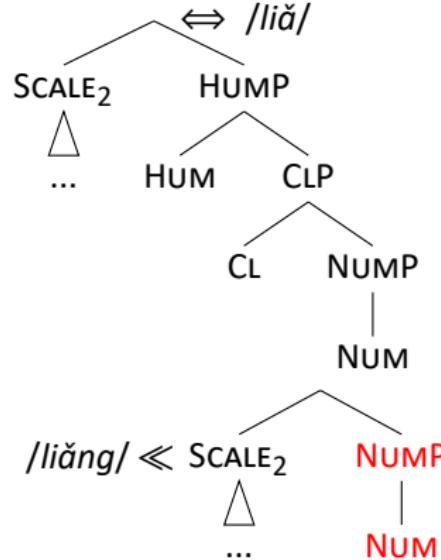
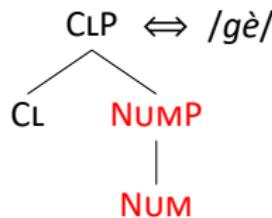
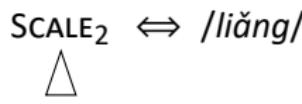
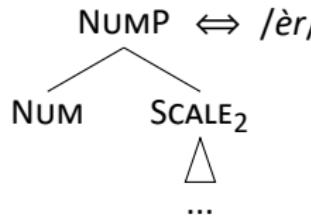
- (102)
- a. Merge F, Spell out FP
 - b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 - c. If (b) fails, move the complement of F, and retry (a)
 - d. If (c) fails, go back to the previous cycle and try the next option for that cycle



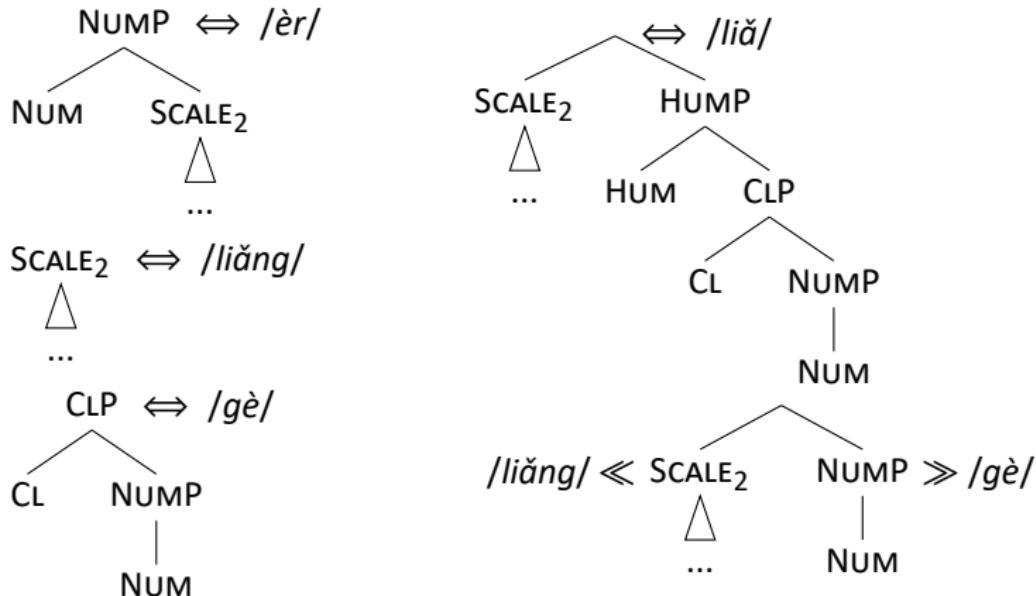
- (103)
- Merge F, Spell out FP
 - If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 - If (b) fails, move the complement of F, and retry (a)
 - If (c) fails, go back to the previous cycle and try the next option for that cycle



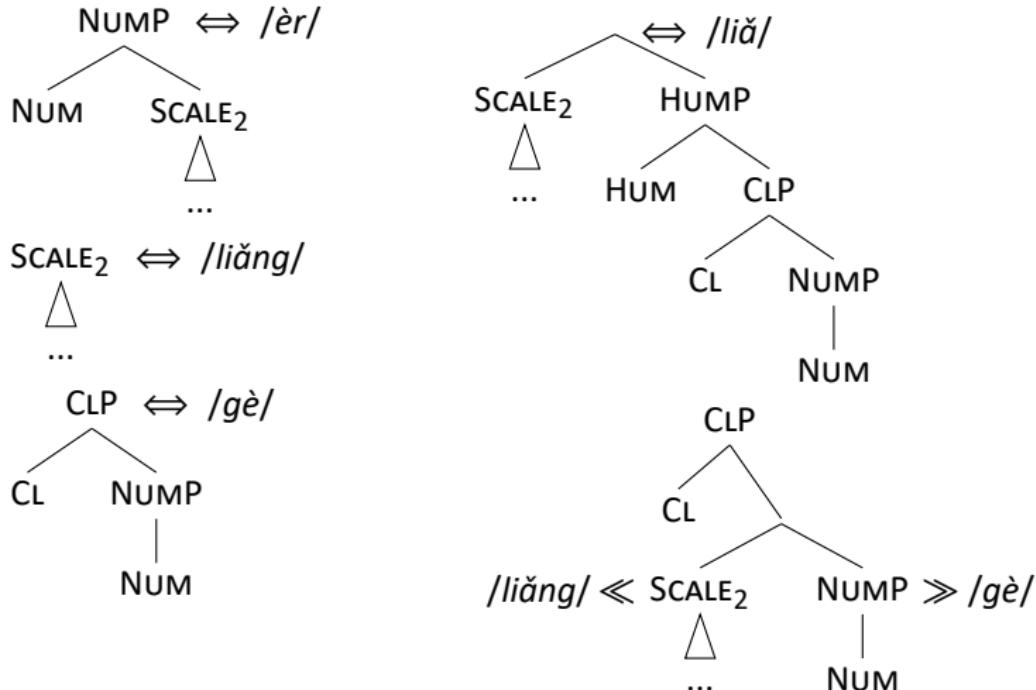
- (104)
- a. Merge F, Spell out FP
 - b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 - c. If (b) fails, move the complement of F, and retry (a)
 - d. If (c) fails, go back to the previous cycle and try the next option for that cycle



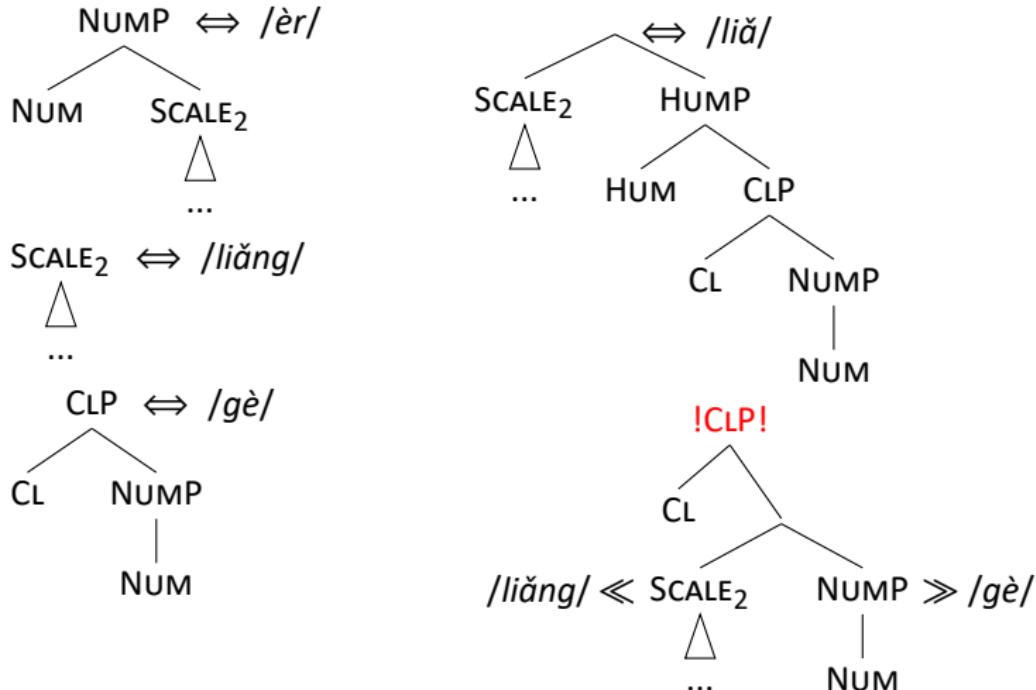
- (105)
- Merge F, Spell out FP
 - If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 - If (b) fails, move the complement of F, and retry (a)
 - If (c) fails, go back to the previous cycle and try the next option for that cycle



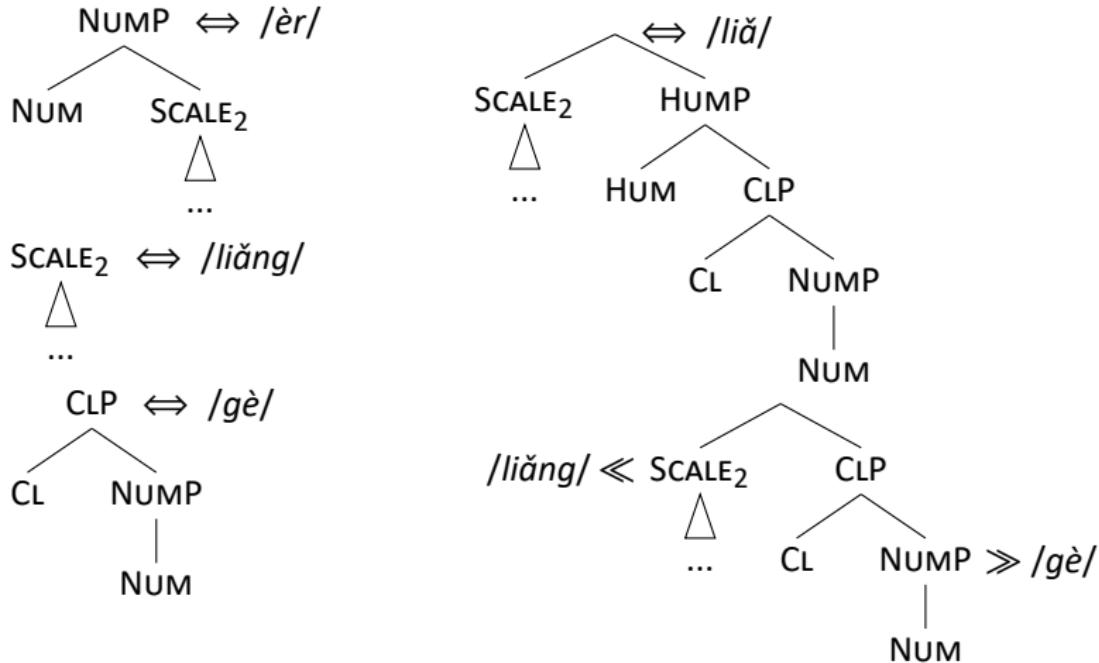
- (106) a. Merge F, Spell out FP
 b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 c. If (b) fails, move the complement of F, and retry (a)
 d. If (c) fails, go back to the previous cycle and try the next option for that cycle



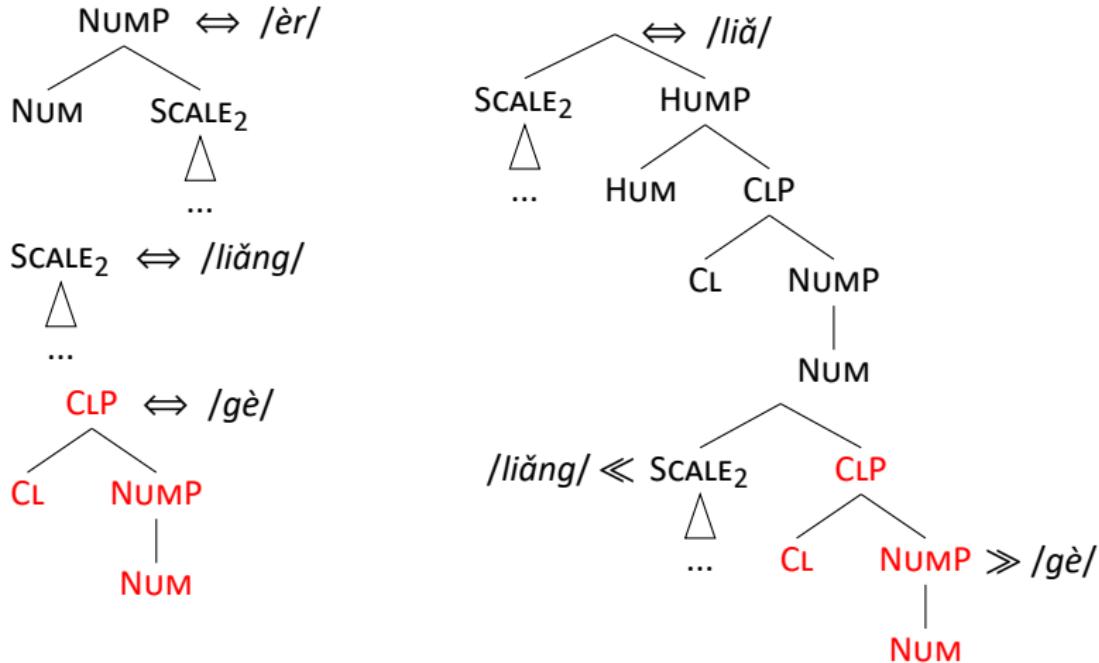
- (107)
- a. Merge F, Spell out FP
 - b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 - c. If (b) fails, move the complement of F, and retry (a)
 - d. If (c) fails, go back to the previous cycle and try the next option for that cycle



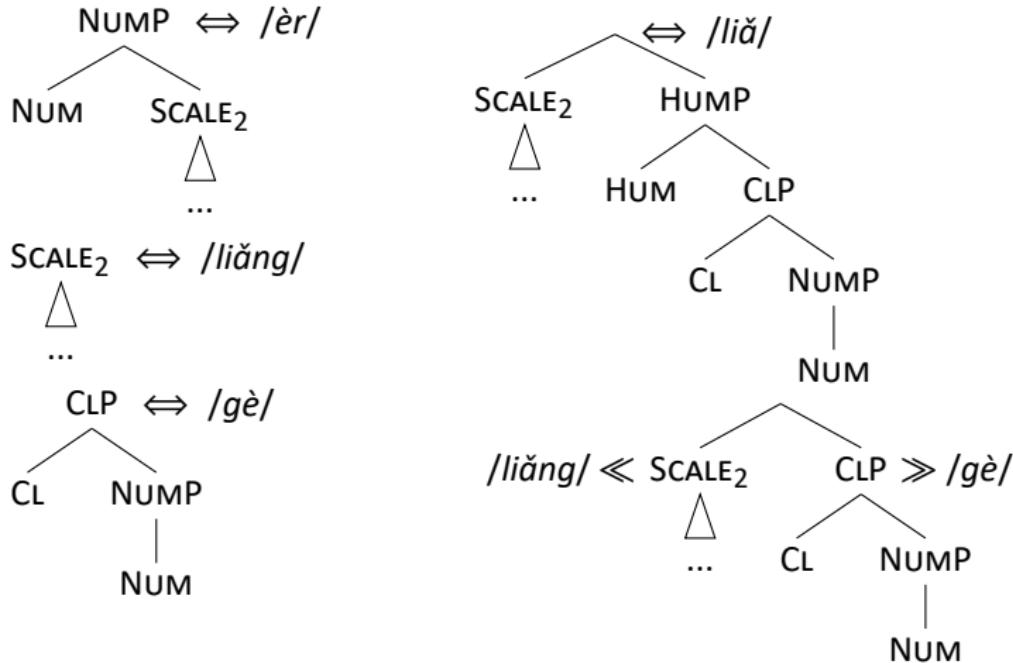
- (108)
- Merge F, Spell out FP
 - If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 - If (b) fails, move the complement of F, and retry (a)
 - If (c) fails, go back to the previous cycle and try the next option for that cycle



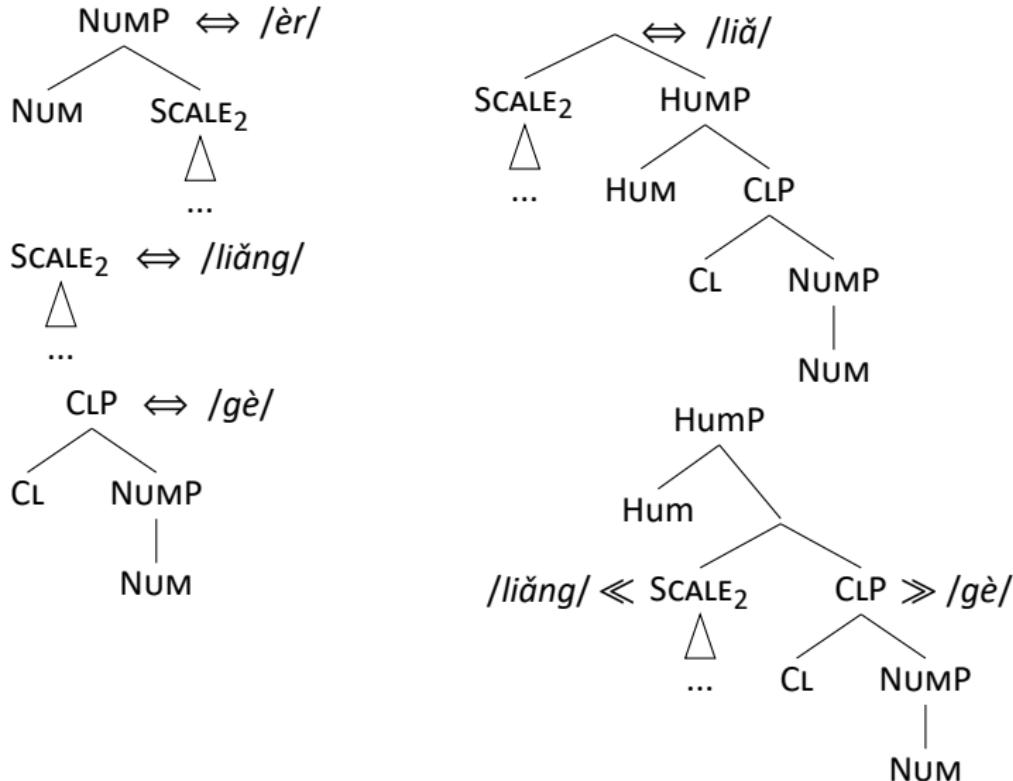
- (109) a. Merge F, Spell out FP
 b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 c. If (b) fails, move the complement of F, and retry (a)
 d. If (c) fails, go back to the previous cycle and try the next option for that cycle



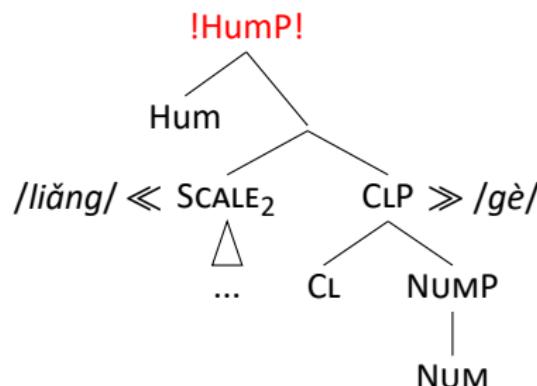
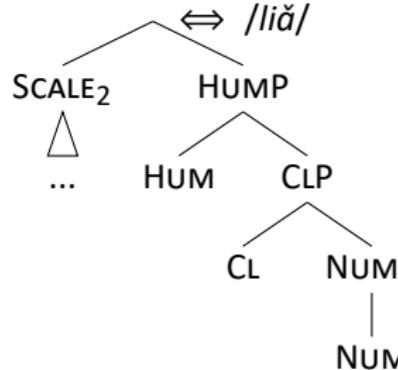
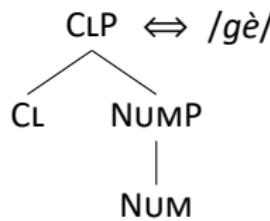
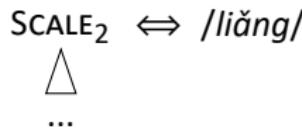
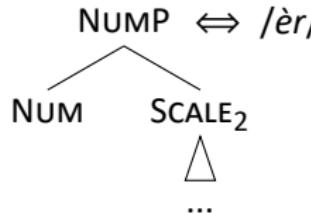
- (110) a. Merge F, Spell out FP
 b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 c. If (b) fails, move the complement of F, and retry (a)
 d. If (c) fails, go back to the previous cycle and try the next option for that cycle



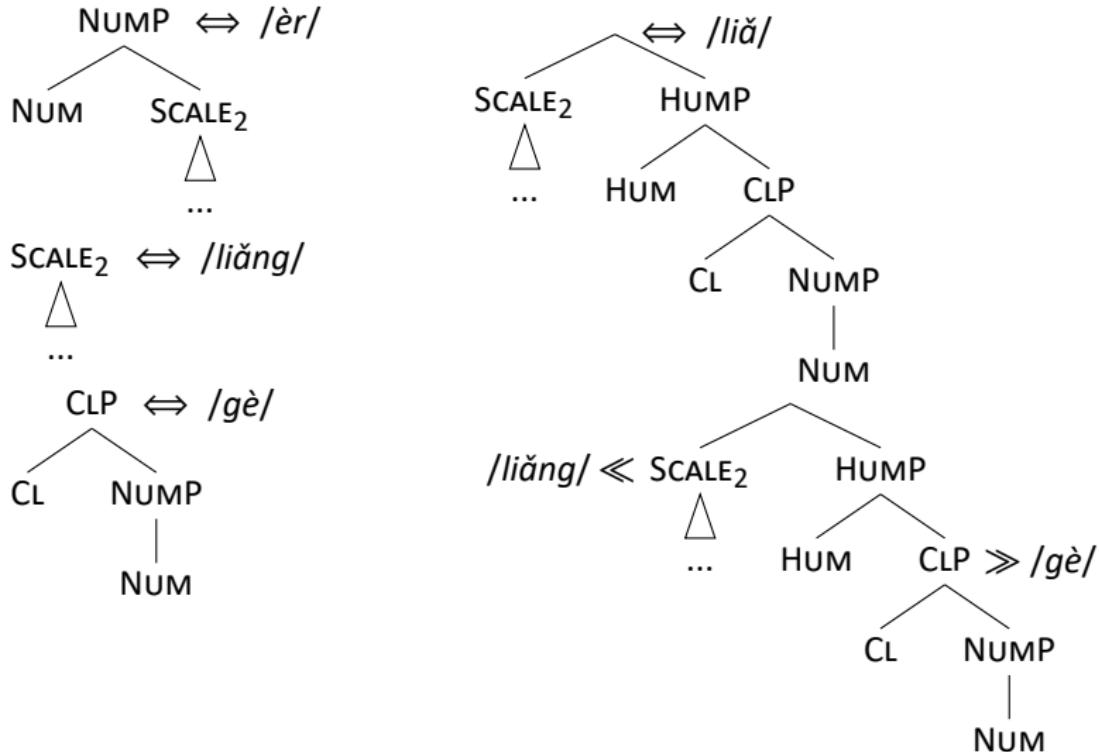
- (111) a. Merge F, Spell out FP
 b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 c. If (b) fails, move the complement of F, and retry (a)
 d. If (c) fails, go back to the previous cycle and try the next option for that cycle



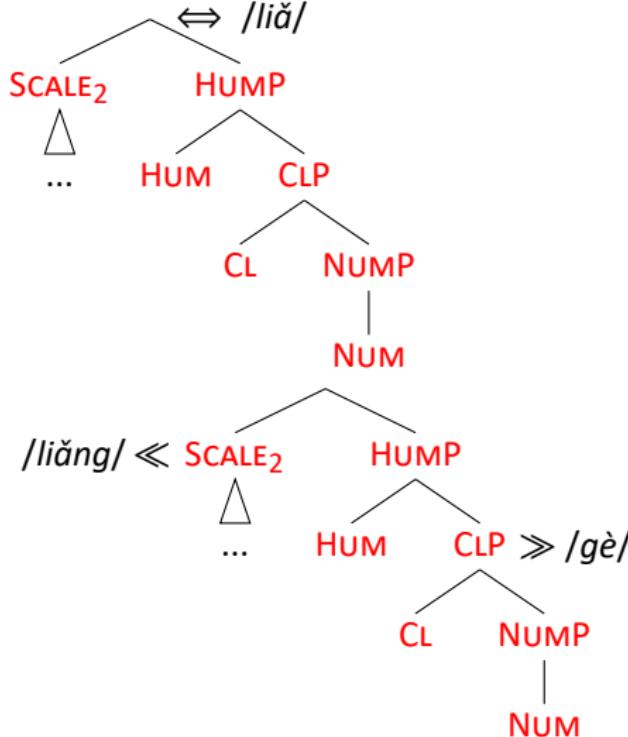
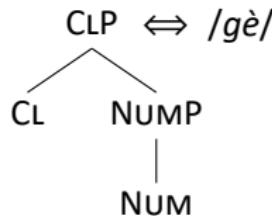
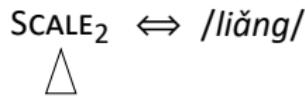
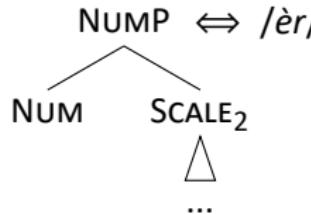
- (112) a. Merge F, Spell out FP
 b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 c. If (b) fails, move the complement of F, and retry (a)



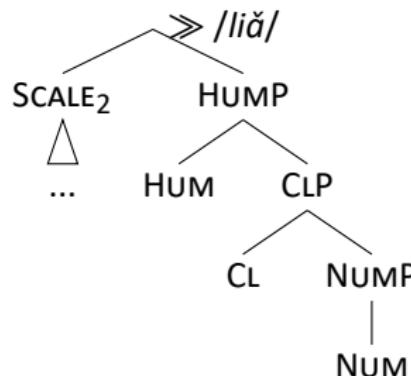
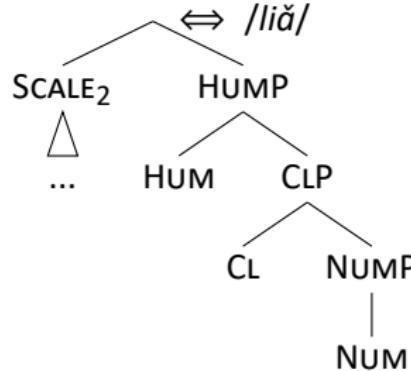
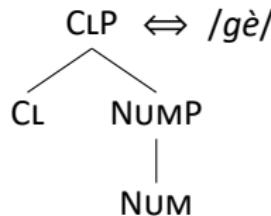
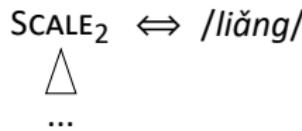
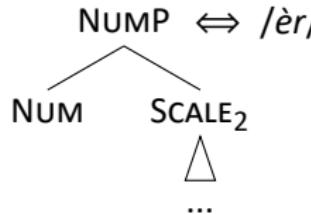
- (113) a. Merge F, Spell out FP
 b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 c. If (b) fails, move the complement of F, and retry (a)



- (114)
- Merge F, Spell out FP
 - If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 - If (b) fails, move the complement of F, and retry (a)



- (115) a. Merge F, Spell out FP
 b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 c. If (b) fails, move the complement of F, and retry (a)



- (116) a. Merge F, Spell out FP
 b. If (a) fails, attempt movement of the spec of the complement of F, and retry (a)
 c. If (b) fails, move the complement of F, and retry (a)

	SCALE	NUM	CL	HUM
--	-------	-----	----	-----

	SCALE	NUM	CL	HUM
REG2 2, abs		èr		
REG2 2, abs	liǎng		gè	
REG2 2, abs			liă	

Introduction

Data

Structures

Spellout

Conclusions

Conclusion

Question

- ▶ what is the **relationship** between object-counting and abstract-counting numerals?

Conclusion

Question

- ▶ what is the **relationship** between object-counting and abstract-counting numerals?

Answer

- ▶ object-counting numerals both syntactically and semantically **contain** abstract-counting numerals!

Conclusion

Question

- ▶ what is the **relationship** between object-counting and abstract-counting numerals?

Answer

- ▶ object-counting numerals both syntactically and semantically **contain** abstract-counting numerals!

Proposal

- ▶ universal underlying structure: [**CL** [**NUM SCALE**]]

Conclusion

Question

- ▶ what is the **relationship** between object-counting and abstract-counting numerals?

Answer

- ▶ object-counting numerals both syntactically and semantically **contain** abstract-counting numerals!

Proposal

- ▶ universal underlying structure: [**CL** [**NUM SCALE**]]
- ▶ lexical items pronounce structures built out of semantic atoms

Conclusion

Question

- ▶ what is the **relationship** between object-counting and abstract-counting numerals?

Answer

- ▶ object-counting numerals both syntactically and semantically **contain** abstract-counting numerals!

Proposal

- ▶ universal underlying structure: [**CL** [**NUM SCALE**]]
- ▶ lexical items pronounce structures built out of semantic atoms

The three Mandarin ‘two’-s

Conclusion

Question

- ▶ what is the **relationship** between object-counting and abstract-counting numerals?

Answer

- ▶ object-counting numerals both syntactically and semantically **contain** abstract-counting numerals!

Proposal

- ▶ universal underlying structure: [**CL** [**NUM SCALE**]]
- ▶ lexical items pronounce structures built out of semantic atoms

The three Mandarin 'two'-s

- ▶ root suppletion of èr ~ *liǎng* can be handled without abandoning this idea

Conclusion

Question

- ▶ what is the **relationship** between object-counting and abstract-counting numerals?

Answer

- ▶ object-counting numerals both syntactically and semantically **contain** abstract-counting numerals!

Proposal

- ▶ universal underlying structure: [**CL** [**NUM SCALE**]]
- ▶ lexical items pronounce structures built out of semantic atoms

The three Mandarin ‘two’-s

- ▶ root suppletion of èr ~ *liǎng* can be handled without abandoning this idea
- ▶ the system is also compatible with a mono-morphemic human-counting *liǎ*

Thanks to

Informants and consultants

- ▶ Mary Chimaobi Amaechi, Alan Bale, Albert J. Borg, Lisa Bylinina, Viacheslav Chirikba, Jessica Coon, Flóra Lili Donáti, Maia Duguine, Abdelkader Fassi Fehri, Andreas Haida, Nina Haslinger, Dominika Kuruncziová, Chang Liu, Catriona Malau, Roumyana Pancheva, Marina Pantcheva, Pittayawat Pittayaporn, Dolf Rami, Bridget Schvarcz, Yasu Sudo, Peter Sutton, Tue Trinh, Kazuko Yatsushiro

References I

- Bale, A. and J. Coon (2014). Classifiers are for numerals, not for nouns: Consequences for the mass/count distinction. *Linguistic Inquiry* 45(4), 695–707.
- Baunaz, L. and E. Lander (2018). Nanosyntax: The Basics. In L. Baunaz, K. De Clercq, L. Haegeman, and E. Lander (Eds.), *Exploring Nanosyntax*, pp. 3–56. Oxford: Oxford University Press.
- Blix, H. (2021). Phrasal spellout and partial overwrite: On an alternative to backtracking. *Glossa* 6(1), 62.1–17.
- Borer, H. (2005). *Structuring Sense I: In Name Only*. Oxford: Oxford University Press.
- Borg, A. (1974). Maltese numerals. *Zeitschrift der Deutschen Morgenländischen Gesellschaft* 124(2), 291–305.
- Borg, A. J. (1987). To be or not to be a copula in Maltese? *Journal of Maltese Linguistics* 17(18), 54–71.
- Bultinck, B. (2005). *Numerous Meanings: The Meaning of English Cardinals and the Legacy of Paul Grice*. Amsterdam: Elsevier.
- Caha, P. (2009). *The Nanosyntax of Case*. Ph. D. thesis, University of Tromsø.
- Caha, P., K. De Clercq, and G. Vanden Wyngaerd (2019). The fine structure of the comparative. *Studia Linguistica* 73(3), 470–521.
- Chierchia, G. (1998). Plurality of mass nouns and the notion of ‘semantic parameter’. In S. Rothstein (Ed.), *Events and Grammar*, pp. 53–103. Dordrecht: Kluwer.
- Chierchia, G. (2010). Mass nouns, vagueness and semantic variation. *Synthese* 174(1), 99–149.

References II

- Corver, N. and J. Zwarts (2006). Prepositional numerals. *Lingua* 116(6), 811–836.
- Fassi Fehri, A. (2018). *Constructing Feminine to Mean: Gender, Number, Numeral, and Quantifier Extensions in Arabic*. Lanham: Rowman & Littlefield.
- Greenberg, J. H. (1978). Generalizations about numeral systems. In J. H. Greenberg (Ed.), *Universals of Human Language*, Volume 3, pp. 249–295. Stanford, CA: Stanford University Press.
- Hackl, M. (2000). *Comparative Quantifiers*. Ph. D. thesis, Massachusetts Institute of Technology.
- He, C. (2015). Complex numerals in Mandarin Chinese are constituents. *Lingua* 164, 189–214.
- Hurford, J. R. (1998). The interaction between numerals and nouns. In F. Plank (Ed.), *Noun Phrase Structure in the Languages of Europe*, pp. 561–620. Berlin: Mouton de Gruyter.
- Hurford, J. R. (2001). Languages treat 1-4 specially. *Mind & Language* 16(1), 69–75.
- Ionin, T. and O. Matushansky (2006). The composition of complex cardinals. *Journal of Semantics* 23(4), 315–360.
- Ionin, T. and O. Matushansky (2018). *Cardinals: The Syntax and Semantics of Cardinal-Containing Expressions*. Cambridge, MA: MIT Press.
- Kennedy, C. (2001). Polar opposition and the ontology of ‘degrees’. *Linguistics and Philosophy* 24(1), 33–70.
- Kennedy, C. (2015). A ‘de-Fregean’ semantics (and neo-Gricean pragmatics) for modified and unmodified numerals. *Semantics & Pragmatics* 8, 1–44.

References III

- Kiparsky, P. (1973). 'Elsewhere' in phonology. In S. Anderson and P. Kiparsky (Eds.), *A Festschrift for Morris Halle*, pp. 93–106. New York: Holt, Rinehart & Winston.
- Krifka, M. (1995). Common nouns: A contrastive analysis of Chinese and English. In G. N. Carlson and F. J. Pelletier (Eds.), *The Generic Book*, pp. 398–411. Chicago: University of Chicago Press.
- Landman, F. (2003). Predicate-argument mismatches and the adjectival theory of indefinites. In *From NP to DP: Volume 1: The Syntax and Semantics of Noun Phrases*, pp. 211–237. Amsterdam: John Benjamins.
- Landman, F. (2004). *Indefinites and the Type of Sets*. Oxford: Blackwell.
- Li, X. (2011). *On the Semantics of Classifiers in Chinese*. Ph. D. thesis, Bar-Ilan University, Ramat Gan.
- Nouwen, R. (2008). Directionality in numeral quantifiers: The case of 'up to'. In T. Friedman and S. Ito (Eds.), *Proceedings from Semantics and Linguistic Theory 18*, pp. 569–582. Ithaca: CLC Publications.
- Nouwen, R. (2016). Making sense of the spatial metaphor for number in natural language. Unpublished manuscript. Available online on LingBuzz.
- Po-Ching, Y. and D. Rimmington (2015). *Chinese: A comprehensive grammar*. London: Routledge.
- Qi, J. and C. He (2019). The morphosyntax of numerals dzi^{33}/dzi^{35} 'one' in Shuhi and implications for the semantics of numerals. *Lingua* 225, 63–80.

References IV

- Rothstein, S. (2010). Counting and the mass/count distinction. *Journal of Semantics* 27(3), 343–397.
- Rothstein, S. (2013). A Fregean semantics for number words. In M. Aloni, M. Franke, and F. Roelofsen (Eds.), *Proceedings of the 19th Amsterdam Colloquium*, pp. 179–186. Amsterdam: ILLC.
- Rothstein, S. (2017). *Semantics for Counting and Measuring*. Cambridge: Cambridge University Press.
- Scha, R. (1981). Distributive, collective and cumulative quantification. In J. A. Groenendijk, T. M. Janssen, and M. B. Stokhof (Eds.), *Formal Methods in the Study of Language*, pp. 483–512. Amsterdam: Mathematisch Centrum.
- Schnell, S. (2011). *A Grammar of Vera'a, an Oceanic Language of North Vanuatu*. Kiel: Kiel University.
- Scontras, G. (2013). Accounting for counting: A unified semantics for measure terms and classifiers. In T. Snider (Ed.), *Proceedings of Semantics and Linguistic Theory* 23, pp. 549–569. Ithaca, NY: CLC Publications.
- Seuren, P. A. (1984). The comparative revisited. *Journal of Semantics* 3(1), 109–141.
- Starke, M. (2009). Nanosyntax: A short primer to a new approach to language. *Nordlyd* 36(1), 1–6.

References V

- Starke, M. (2018). Complex left branches, spellout, and prefixes. In L. Baunaz, K. De Clercq, L. Haegeman, and E. Lander (Eds.), *Exploring Nanosyntax*, pp. 239–249. Oxford: Oxford University Press.
- Sudo, Y. (2016). The semantic role of classifiers in Japanese. *Baltic International Yearbook of Cognition, Logic and Communication* 11(1), 1–15.
- Wągiel, M. and P. Caha (2020). Universal semantic features and the typology of cardinal numerals. *Catalan Journal of Linguistics* 19, 199–229.
- Wągiel, M. and P. Caha (2021). Complex simplex numerals. *Acta Linguistica Academica* 68(4), 470–515.
- Zhang, N. N. (2013). *Classifier Structures in Mandarin Chinese*. Berlin: De Gruyter Mouton.