

Replicative Processes: Morphology

Master: 04-046-2011 (Morphologie: Flexion)
IGRA: 07, Topics in Morphology (seminar)

Tuesdays, 17:00–18:30, S102, NSG
SoSe 2015, Universität Leipzig

Institut für Linguistik
Gereon Müller
gereon.mueller@uni-leipzig.de
http://www.uni-leipzig.de/~muellerg

Reduplication: Frampton (2009, Kap. 3-4)

1. The Morphology of Reduplication

Claim:

Juncture insertion is *readjustment* in the sense of Distributed Morphology. The readjustment rule can be triggered by lexical insertion of an overt vocabulary item, or by zero exponence for an f-morpheme.

Transcription:

Timing slots generated by transcription are not associated with morphemes; they are not represented in hierarchical structure.

- (1) *Default choices:*
- Rule application is leftmost, unless specified otherwise.
 - The domain of juncture insertion rules is taken to be the stem, unless specified otherwise.
 - Default closure:*
Unpaired [and] junctures which remain after juncture insertion are closed by inserting] or [at the edges of the domain.
 $\emptyset \rightarrow]/V_$

2. Truncated Reduplication

(2) *String copy as a general process:*

- k a t
- Δ (k a t)
- Δ (<>k a t)
- k_{Δ} (<k> a t)
- ka_{Δ} (<k a> t)

- kat_{Δ} (<k a t>)
- kat_{Δ} k a t

(3) Generation of *kat-kat* from *kat*:

- Base

$$\begin{array}{c} [\quad \times \quad \times \quad \times \quad] \\ | \quad | \quad | \\ k \quad a \quad t \end{array}$$
- Init

$$\begin{array}{c} [\langle \rangle \quad \times \quad \times \quad \times \quad] \\ | \quad | \quad | \\ k \quad a \quad t \end{array}$$
- Copy Shift

$$\begin{array}{c} x \quad [\langle \rangle \quad \times \quad \times \quad \times \quad] \\ \quad \quad \quad | \quad | \quad | \\ \quad \quad \quad k \quad a \quad t \end{array}$$
- Copy Shift

$$\begin{array}{c} \times \quad \times \quad [\langle \rangle \quad \times \quad \times \quad \times \quad] \\ \quad \quad \quad \quad \quad | \quad | \quad | \\ \quad \quad \quad \quad \quad k \quad a \quad t \end{array}$$
- Copy Shift

$$\begin{array}{c} \times \quad \times \quad \times \quad [\langle \rangle \quad \times \quad \times \quad \times \quad] \\ \quad \quad \quad \quad \quad \quad \quad | \quad | \quad | \\ \quad \quad \quad \quad \quad \quad \quad k \quad a \quad t \end{array}$$
- Cleanup

$$\begin{array}{c} \times \quad \times \quad \times \quad \quad \times \quad \times \quad \times \\ \quad \quad \quad \quad \quad \quad | \quad | \quad | \\ \quad \quad \quad \quad \quad \quad k \quad a \quad t \end{array}$$
- NCC-Repair

$$\begin{array}{c} \times \quad \times \quad \times \quad \quad \times \quad \times \quad \times \\ | \quad | \quad | \quad \quad | \quad | \quad | \\ k \quad a \quad t \quad \quad k \quad a \quad t \end{array}$$

(Note: This step is counter-cyclic.)

2.1. Truncation at the Leading Edge (Reduplicant Truncation)

Assumption:

Truncation junctures (<, >) are genuine phonological objects. They can be used for

truncation if they are inserted into timing slots by readjustment, in the same way that transcription junctures are.

- a. Base
- $$\begin{array}{c} [\quad \times \quad \times \quad \times \quad] \\ | \quad | \quad | \\ k \quad a \quad t \end{array}$$
- b. Readjustment
- $$\begin{array}{c} [\quad < \times > \quad \times \quad \times \quad] \\ | \quad | \quad | \\ k \quad a \quad t \end{array}$$
- c. Copy Shift
- $$\begin{array}{c} x \quad [\quad < \times \quad \times > \quad \times \quad] \\ \diagdown \quad | \quad | \\ \quad \quad k \quad a \quad t \end{array}$$
- d. Copy Shift
- $$\begin{array}{c} \times \quad \times \quad [\quad < \times \quad \times \quad \times > \quad] \\ \diagdown \quad | \quad | \quad | \\ \quad \quad k \quad a \quad t \end{array}$$
- e. ... → atkat

(4) *Tasty-shmasty reduplication:*

- a. Base, readjustment:
- $$\begin{array}{ccccccccccc} \times & \times & \times & \times & \times & \rightarrow & \times & [& \times & \times & \times & \times & < \# & \times & \times & >] \\ | & | & | & | & | & & | & & | & | & | & | & & | & | \\ t & a & s & t & i & & t & & a & s & t & i & & sh & m \end{array}$$
- b. Multiple copy shift:
- $$\begin{array}{cccccccccccc} \times & [& \times & \times & \times & \times & < \# & \times & \times & >] & \times & \times & \times & \times \\ | & & | & | & | & | & | & | & | & | & | & | & | & | \\ t & & a & s & t & i & & sh & m & & & & & & \end{array}$$
- c. Cleanup and NCC-repair:
- $$\begin{array}{cccccccccccc} \times & \times & \times & \times & \times & \# & \times & \times & \times & \times & \times & \times \\ | & | & | & | & | & & | & | & | & | & | & | \\ t & a & s & t & i & & sh & m & a & s & t & i \end{array}$$

2.2. Trailing Truncation

(5) *Transcription rules* (left)

- a. Init:
 $\emptyset \rightarrow <>/[_ \times$
- b. Copy Shift:
 $\Delta \dots > \times \rightarrow \times \Delta \dots \times >$
- c. Cleanup:
 $[< \times * >] \rightarrow \times *$

Problem:

Truncation junctures that do not show up at the leading edge are not compatible with the algorithm yet.

- (6) a. $\Delta[k<at>]$
b. $\Delta[<>k<at>]$
c. $\Delta[<k><at>]$
d. no continuation possible

Suggestion:

There is a further copy operation that *deletes* items, and that is applicable in (6-c).

(7) *Two further transcription rules:*

- a. Copy Delete:
 $\Delta \dots > < \times \rightarrow \times \Delta \dots > <$
- b. $<>$ -Cleanup:
 $<> \rightarrow \emptyset / > _$

Consequences:

Infixation, metathesis