Class 7
Templatic morphology: problems for cyclic concatenation, and parallel alternatives

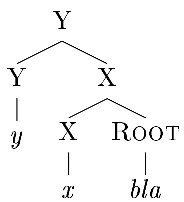
11/9/23

1 Recap: Cyclic concatenation and the Mirror Principle

- **Last time:** We can derive basic cases of Mirror Principle behavior from **cyclic concatenation**.
  - Terminal nodes are spelled-out one at time, from the bottom up.
  - This accords with the basic DM approach to spell-out and the interrelationship between syntax and morphology we saw for allomorphy. This architecture is partly inspired by the Mirror Principle.

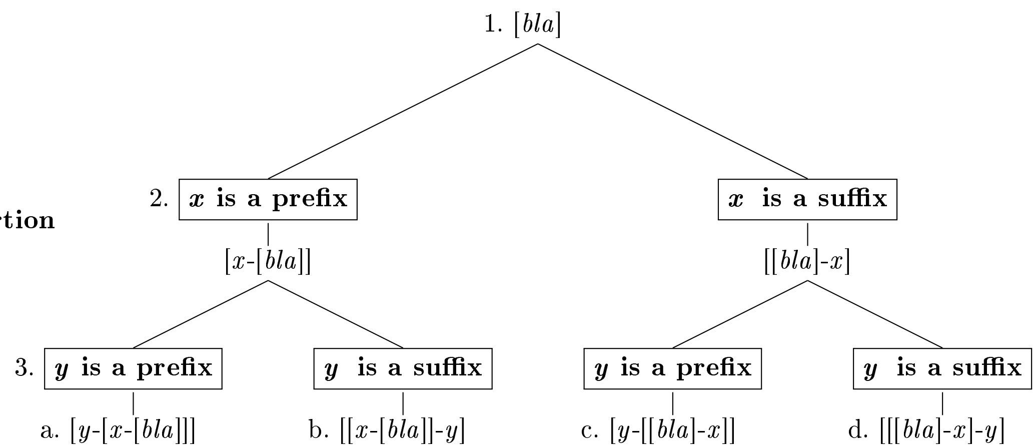
- Cyclic concatenation results in the decision tree in (3), which derives all and only the Mirror Principle-compliant orders; i.e., it correctly rules out *[*x*-*y*-*bla*] and *[*bla*-*y*-*x*].

(1) **Schematic tree**



(2) **Vocabulary insertion**
1. Root ⇔ *bla*
2. X ⇔ *x*
3. Y ⇔ *y*

(3) **Cyclic spellout**



- If the spellout process, or the VI rules themselves, come equipped with a means of determining whether a given affix is a prefix or a suffix, then the decision tree in (3) reduces to a single output.

⋆ **The problem:** There are various kinds of morphology that don't appear to work this way.

→ I have some ideas about how to do things differently, in a way that fixes a lot of these problems:
  - Delay concatenation until the phonology (i.e. no linear in the morphology)
  - Various kinds of morphosyntactic information is visible by / retained into the phonology
  - Order is determined via constraint competition between different candidate orders

- I'll start by going through the original proposal, the "Mirror Alignment Principle" (MAP) (Zukoff 2023), and show how it can account for "asymmetric compositionality" in the Bantu CARP template (Hyman 2003).
- Then I'm going to show you how additional related facts might be better explained with a similar (and compatible) model using Base-Derivative faithfulness constraints (Benua 1997).
- Then we'll look at Arabic "root-and-pattern" morphology with these proposals in mind.

2 The Mirror Alignment Principle (MAP)

2.1 Generalized Alignment

- McCarthy & Prince (1993) propose a type of OT constraint called an **alignment constraint**.
  - Alignment constraints require that specified edges of phonological and/or morphological constituents coincide in the phonological output.

(4) **Generalized Alignment** [GA] (McCarthy & Prince 1993:80)
Align (Cat[egory]1, Edge1, Cat[egory]2, Edge2) $=_{\text{def}}$
$\forall$ Cat1 $\exists$ Cat2 such that Edge1 of Cat1 and Edge2 of Cat2 coincide.
Where
Cat1, Cat2 $\in$ P[rosodic]Cat $\cup$ G[rammatical]Cat
Edge1, Edge2 $\in$ Right, Left

A GA requirement demands that a designated edge of each prosodic or morphological constituent of type Cat1 coincide with a designated edge of some other prosodic or morphological constituent Cat2.

- Alignment constraints have been used for all sorts of purposes, with all sorts of combinations of constituents and edges. The main two:
  - Stress & footing
  - **Morpheme placement**

- Let's think about what happens if the only kind of alignment constraint is one that relates a specific morpheme to a specific edge of the word.

2.2 Competing alignment constraints

- Given a word that consists of a Root plus three affixes X, Y, and Z, let's make the following assumptions:

(5) **Assumptions**
i. The phonological UR is a *linearly unordered* set of exponents: /Root, X, Y, Z/
→ In other words, the morphology has not established any ordering between exponents.

ii. Each affix is referenced by a single alignment constraint; for this example, they're all going to be right-word-edge-oriented (6).
- As a general matter, I assume that opposite edge alignment is not permitted.
- Violations of the alignment constraints are assigned gradiently.

(6) **Alignment constraints**
a. ALIGN(X, R; PWD, R) [ALIGN-X-R]
Assign one violation for each segment intervening between the right edge of morpheme X and the right edge of the prosodic word.
b. ALIGN(Y, R; PWD, R) [ALIGN-Y-R]
Assign one violation for each segment intervening between the right edge of morpheme Y and the right edge of the prosodic word.
c. ALIGN(Z, R; PWD, R) [ALIGN-Z-R]
Assign one violation for each segment intervening between the right edge of morpheme Z and the right edge of the prosodic word.

- Each alignment constraint is maximally satisfied when it is the rightmost morpheme within the word.
  - ∘ In any candidate output, only one morpheme can be rightmost (full alignment satisfaction).
  - ∘ Satisfaction of one of alignment constraint therefore entails increased violation of the others.[1]
  - → These constraints are in *direct competition* for final position.

(7) **Violation profiles** (no rankings yet)

/Root, X, Y, Z/		Align-X-R	Align-Y-R	Align-Z-R
a.	Root-X-Y-Z	**	*	
b.	Root-Y-X-Z	*	**	
c.	Root-X-Z-Y	**		*
d.	Root-Z-X-Y	*		**
e.	Root-Y-Z-X		**	*
f.	Root-Z-Y-X		*	**

- Each candidate has three alignment violations,[2] distributed across the different constraints:
  - ∘ 0 violations for the rightmost morpheme.
  - ∘ 1 violation for the morpheme that is second from the right.
  - ∘ 2 violations for the morpheme that is third from the right.

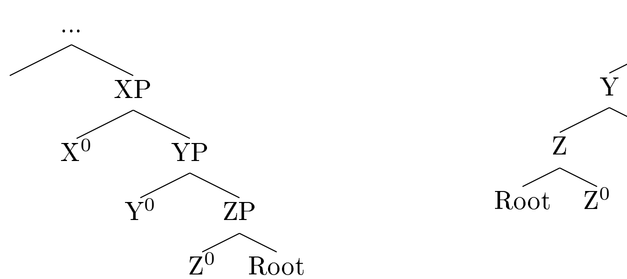
→ The six possible rankings each correspond to the selection of one of the six candidate orders.

2.3 Alignment and the Mirror Principle

- Now let's assume that the word we're building is associated with the syntactic structure in (8):

(8) **Syntax of /Root, X, Y, Z/**

a. Base-generated structure → b. Complex head



- If the affixes are all suffixes, we'd expect the Mirror Principle-compliant order to be [Root-Z-Y-X].
  - ∘ This is candidate order (7f). We can generate this order with the alignment ranking in (9):

(9) **Generating the Mirror Principle order for (8)**

i. *Ranking*: Align-X-R ≫ Align-Y-R ≫ Align-Z-R

ii. *Tableau*:

/Root, X, Y, Z/			Align-X-R	Align-Y-R	Align-Z-R
a.		Root-X-Y-Z	*!*	*	
b.		Root-Y-X-Z	*!	**	
c.		Root-X-Z-Y	*!*		*
d.		Root-Z-X-Y	*!		**
e.		Root-Y-Z-X		**!	*
f.	☞	Root-Z-Y-X		*	**

[1] With one exception: non-realization (i.e. deletion) leads to vacuous satisfaction...

[2] I'm treating each morpheme as if it's a single segment. This actually doesn't matter.

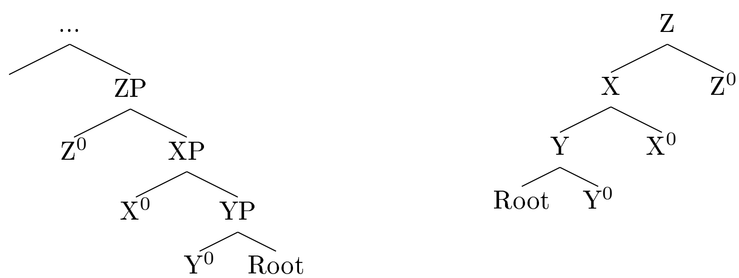
⋆ Notice that there's a relationship between the hierarchical structure in (8) and the ranking in (9):
  - The highest terminal in the syntactic tree is X; the highest-ranked constraint in the constraint ranking is Align-X.
  - The next highest terminal in the syntactic tree is Y; the next highest-ranked constraint is Align-Y.
  - The lowest terminal in the syntactic tree is Z; the lowest-ranked constraint is Align-Z.

⇒ A formal connection between hierarchical structure (in the form of asymmetric c-command) and alignment ranking can therefore *derive* the Mirror Principle. I call this the **Mirror Alignment Principle**:

(10) **The Mirror Alignment Principle (The MAP)**
  a. If a terminal node α asymmetrically c-commands a terminal node β, then the alignment constraint referencing α dominates the alignment constraint referencing β.
  b. *Shorthand*: If α c-commands β → Align-α ≫ Align-β

- The MAP lets us dynamically derive the Mirror Principle for different structures.
  - If the syntax gives us a different structure for the same combination of morphemes (11), then the MAP will provide a different alignment ranking, which will select a different morpheme order (12).

(11) **A different syntactic structure (Z is highest)**
  a. Base-generated structure → b. Complex head



(12) **Generating the Mirror Principle order for (11)**
  i. *Ranking*: Align-Z-R ≫ Align-X-R ≫ Align-Y-R
  ii. *Tableau*:

/Root, X, Y, Z/		Align-Z-R	Align-X-R	Align-Y-R
a.	Root-X-Y-Z		**!	*
b. ☞	Root-Y-X-Z		*	**
c.	Root-X-Z-Y	*!	**	
d.	Root-Z-X-Y	*!*	*	
e.	Root-Y-Z-X	*!		**
f.	Root-Z-Y-X	*!*		*

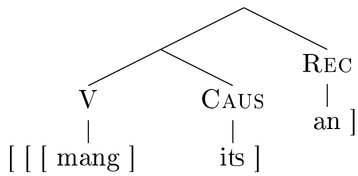
- This approach is thus sensitive to alternations in syntactic structure within a language, of the sort that motivated the Mirror Principle in the first place.
  - It does entail that a language will have different constraint rankings for different syntactic derivations.
  - This is a little unusual in standard OT, but has clear parallels in models like Cophonology Theory (Inkelas & Zoll 2007, Sande, Jenks, & Inkelas 2020).

3 The MAP and templatic morphology in Bantu

- Recall the interaction between Causative and Reciprocal in Chichewa (13).
  - When Reciprocal /an/ is further to the right than Causative /its/ (13a), the Reciprocal is interpeted as taking scope over the Causative.
  - When Causative /its/ is further to the right than Reciprocal /an/ (13b), the Causative is interpeted as taking scope over the Reciprocal.

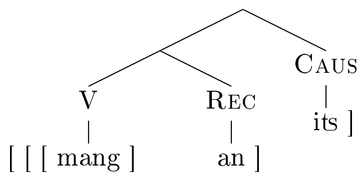
(13) Orders of Causative and Reciprocal in Chichewa (Hyman & Mchombo 1992:350, Hyman 2003:247)

a. **Reciprocalized Causative:** *mang-its-an-* 'cause each other to tie'



[REC: [Z_i cause [*each other*$_i$ tie Y]]]
[Z cause [X tie Y]]
[X tie Y]

b. **Causativized Reciprocal:** *mang-an-its-* 'cause to tie each other'



[Z cause [REC: [X_i tie *each other*$_i$]]]
[REC: [X_i tie *each other*$_i$]]
[X tie Y]

3.1 Deriving Chichewa's Mirror Principle behavior

- As we talked about last time, it would be easy to derive these facts with cyclic concatenation.
  - ⋆ It is also easy to derive this with the MAP!

- The constraints we need are the following:

(14) **Alignment constraints for Chichewa verbal extensions**

a. Align(Reciprocal, R; PWd, R) [Align-Rec-R]
Assign one violation for each segment intervening between the right edge of the exponent of Reciprocal and the right edge of the word.

b. Align(Causative, R; PWd, R) [Align-Caus-R]
Assign one violation for each segment intervening between the right edge of the exponent of Causative and the right edge of the word.

∗ **Step 1:** Use the MAP to generate the rankings for the two syntactic structures from (13) in (15):

(15) **Mirror Alignment Principle rankings for the structures in (13)**

a. **Reciprocalized Causative (13a):**

C-command relations		MAP ranking
Rec c-commands Caus	→	Align-Rec-R ≫ Align-Caus-R

b. **Causativized Reciprocal (13b):**

C-command relations		MAP ranking
Caus c-commands Rec	→	Align-Caus-R ≫ Align-Rec-R

∗ **Step 2:** Use these rankings to derive the right output orders for the respective syntactic structures:

(16) **Reciprocalized Causative** *mang-its-an-* (13a)

/mang$_{\text{ROOT}}$, its$_{\text{CAUS}}$, an$_{\text{REC}}$/	Align-Rec-R	Align-Caus-R
a. ☞ mang-its-an-		** (an)
b. mang-an-its-	*!* (its)	
c. its-mang-an-		**, *!*** (an, mang)
d. an-mang-its-	*!*, **** (its, mang)	

(17) **Causativized Reciprocal** *mang-an-its-* (13b)

/mang$_{\text{ROOT}}$, its$_{\text{CAUS}}$, an$_{\text{REC}}$/	Align-Caus-R	Align-Rec-R
a. mang-its-an-	*!* (an)	
b. ☞ mang-an-its-		** (its)
c. its-mang-an-	*!*, **** (an, mang)	
d. an-mang-its-		**, *!*** (its, mang)

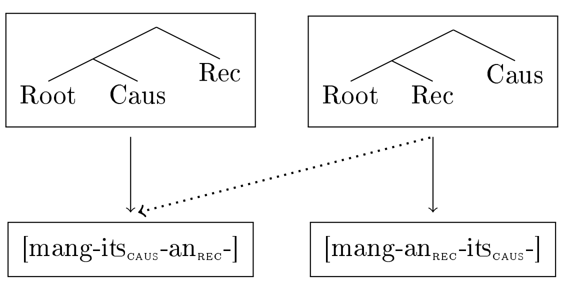
3.2 The CARP template and asymmetric compositionality

- This data can be handled equally well with either approach. But actually, it's not quite so simple:

→ The **order** *mang-its-an-* (13a)/(16a) can also **mean** 'cause to tie each other', the meaning in (13b)/(16b).
  - Put another way:
    - The order *mang-its-an-* can have either meaning, **and**
    - The meaning 'cause to tie each other' can have either order.
  - **But**, the order *mang-an-its-* (13b)/(16b). can only mean 'cause to tie each other' (13b)/(16b).

- This situation can be represented as follows:

(18) **Permissible mappings between structure and order**



- This is one component of what Hyman (2003) (following Hyman & Mchombo 1992) calls the **"CARP template"**:
  - In most Bantu languages, combinations of Causative [C], Applicative [A], Reciprocal [R], and Passive [P], can always appear in that order [C-A-R-P], regardless of their syntax / semantic interpretation.
  - In some languages, these affixes can *only* appear in this order, regardless of interpretation.
  - In some languages, like Chichewa, some pairs of affixes, like Caus and Rec, can appear in either order, but with the interpretative asymmetry in (18), which Hyman calls "**asymmetric compositionality**".
  - Even in such languages, certain other pairs of affixes, like Caus and Appl, only surface in the CARP order ([CA]).

⋆ **What does this tell us about the architecture of the grammar and the Mirror Principle?**

3.3 Proving the syntax

- The dotted line mapping in (18) presupposes that we in fact have distinct syntactic structures in the cases where order is neutralized.
  - This is inescapable in our feed-forward architecture DM-style architecture, but is only so because of lots of other assumptions we're making.
  - It would be nice to have empirical evidence to make the argument.

→ I'm going to show you some evidence that Hyman adduces from the Causative and Applicative.
  - I think it does make the correct point, but I think it may not be as straightforward as he (or I) thought.

3.3.1 Causative and Applicative: the view from 30,000 feet

- In Chichewa, Causative and Applicative always surface as [CA].
  - When this order corresponds to an Applicativized Causative interpretation (C < A), and gets passivized, only the Applicative argument can be promoted to subject, as shown in (19).
  - On the other hand, when this order corresponds to a Causativized Applicative interpretation (C > A), and gets passivized, only the Causee can be promoted to subject, as shown in (20).

(19) **Applicativized Causatives in Chichewa** (Hyman 2003:260, ex. 22)
(Caus *-its*, Appl *-il*, Pass *-idw*, 'cry' *líl*, 'children' *aná*, 'stick' *ndodo*)

a. Mchómbó a-ná-líl-**its-il**-a [CAUSEE aná] [APPL ndodo]
'Mchombo made the children cry with a stick'

b. [APPL ndodo] i-ná-líl-**its-il**-idw-á [CAUSEE aná]
'a stick was used to make the children cry'

c. ?*[CAUSEE aná] a-ná-líl-**its-il**-idw-á [APPL ndodo]
'the children were made to cry with a stick'

(20) **Causativized Applicatives in Chichewa** (Hyman 2003:260, ex. 23)
('cultivate' *lím*, 'hoes' *makásu*)

a. Mchómbó a-ná-lím-**its-il**-a [CAUSEE aná] [APPL makásu]
'Mchombo made the children cultivate with hoes'

b. [CAUSEE aná] á-ná-lím-**its-il**-idw-á [APPL makásu]
'the children were made to cultivate with hoes'

c. ?*[APPL makásu] a-ná-lím-**its-il**-idw-á [CAUSEE aná]
'hoes were used to make the children cultivate'

- Here's the logic:
  - Only the argument that is syntactically highest is available for movement to subject.
  - This requires that the arguments, and, correspondingly, the heads that introduce them, be merged in different syntactic orders for the two different scopal interpretations.
  - → There must be distinct syntactic structures underlying the ambiguous surface form of the verb word.

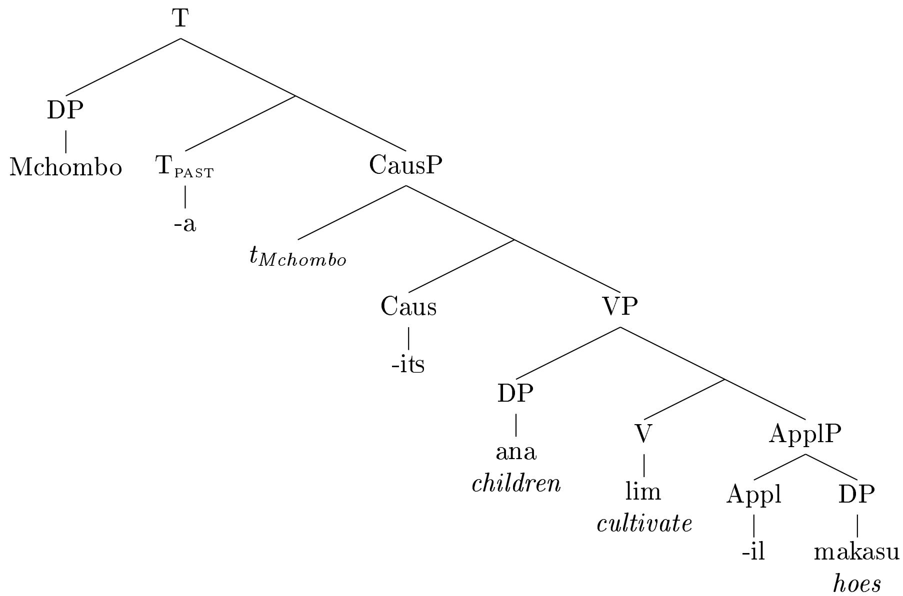
⋆ **Take-away:** The CARP template neutralizes underlying syntactic contrasts on the way to linear order.

3.3.2 Zooming in: Applicatives are tricky

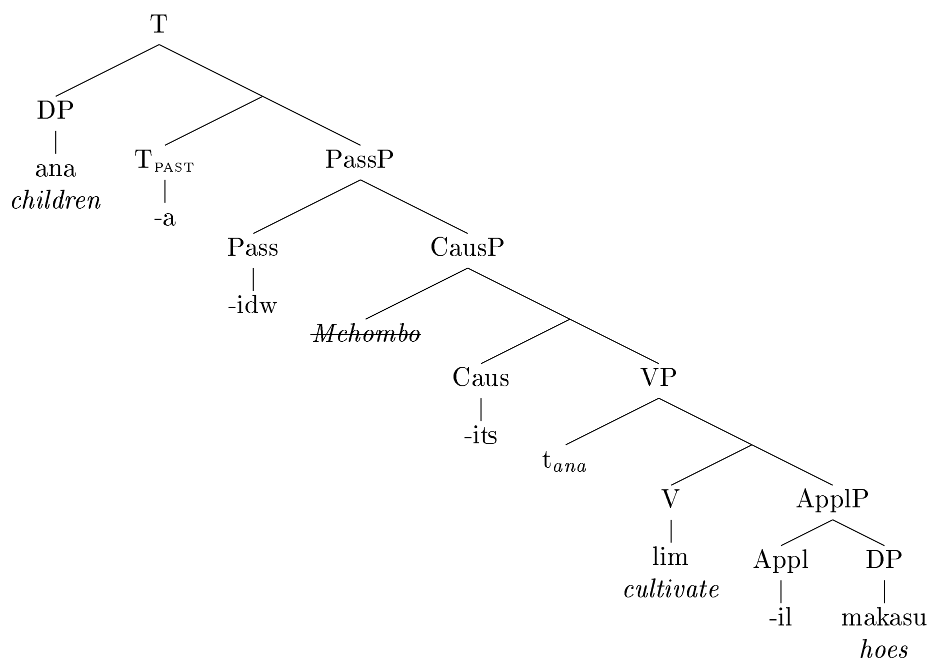
- Here's the problem:
  - The Causee *ana* 'children' is the argument of the Verb/Root in both cases.
  - If the Causative head and the Applicative head always merge *above* the Verb/Root, then the Applicative argument should always be higher than the Causee.
  - → If this is true, then the distinction about raising to subject in the two cases couldn't be about argument height.

- But I think we can save the argument by positing that the Applicative in the second case is a *low applicative* (Pylkkänen 2002), merged below the Verb/Root.

(21) **Tree of (20a) Causativized (Low) Applicative**



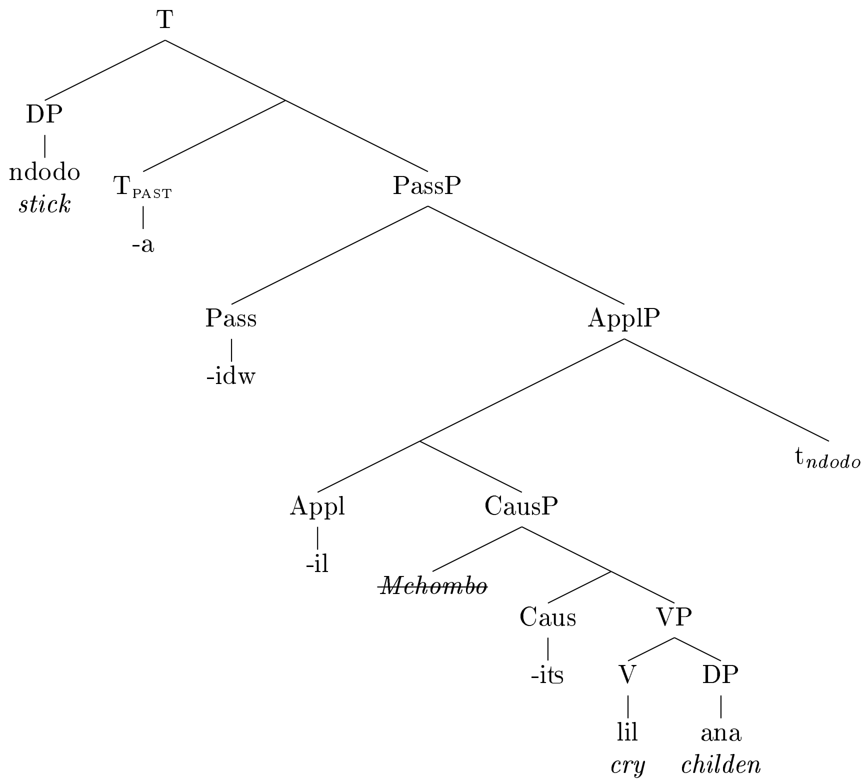
(22) **Tree of (20b) Passivized Causativized (Low) Applicative**



- This gets the raising-to-subject facts right for the Causativized Applicative.

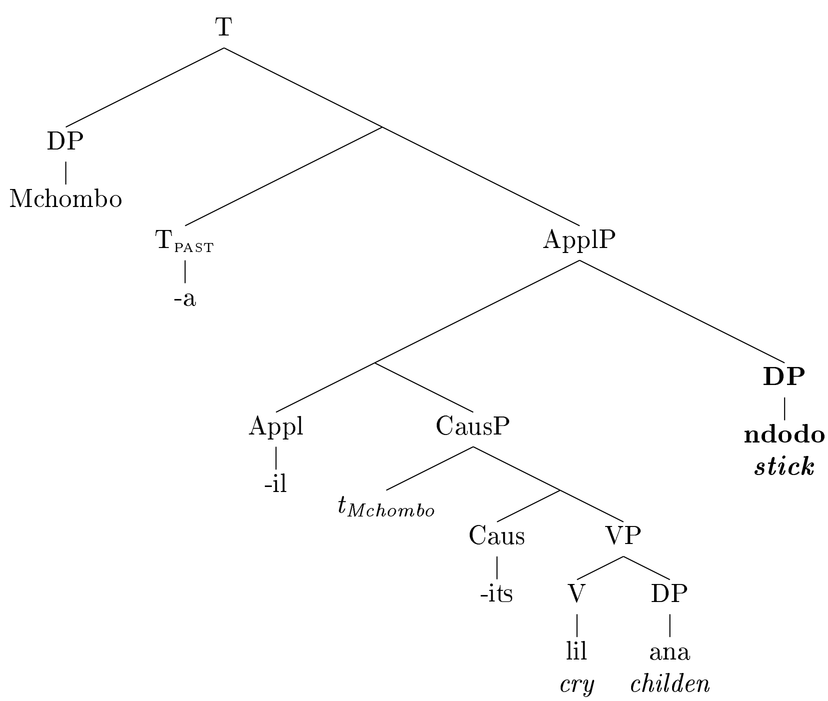
- We then need to say that the cases of Applicativized Causatives represent high(er) Applicatives.
  - This works for the raising-to-subject facts in the Passivized Applicativized Causative:

(23) **Tree of (19b) Passivized Applicativized Causative (Appl above Caus)**



- But this structure then raises a question about how to get the Causer argument *Mchombo* to raise to subject in the non-passivized case, where the Applicative argument seems to be higher:

(24) **Tree of (19a) Applicativized Causative (Appl above Caus)**



- This problem would be alleviated if we merged the Appl head below the Caus head but above V, as this would make the Causer the highest argument.
  - But then the semantics (maybe) don't work out quite right.

⋆ So maybe this doesn't turn out to be the cleanest argument for having [[[Root]Caus]Appl] vs. [[[Root]Appl]Caus].
  → But it does show that there must be distinct structures underlying different meanings that map to the same order.
  - And furthermore, the Mirror order of one (or maybe both) of these structures would generate *Root-Appl-Caus-*, since Appl is lower than Caus, which is the exact opposite of the templatic/fixed order of *Root-Caus-Appl-*. So cyclic concatenation is still not going to do the trick.

∗ Despite the complications here, I'll continue to assume that the answer to the original question is ***yes***: distinct syntactic structures underlie cases where distinct meanings have been neutralized to a single order.

3.4 Capturing CARP (fixed ordering) with bigram constraints

- Once we take this position, then we need something later in the derivation that can prefer mapping to the templatic order, even (and especially) when this is not the mirror/cyclic order.

- I'm going to go in Ryan's (2010) "bigram morphotactic constraints", which are a general way of accounting for templatic morphology with constraints.
  - These are constraints that operate in the phonology (or maybe a constraint-based morphological component) that (arbitrarily) penalize certain orders of morphemes.
  - For example, the fact that Applicative can't precede Causative in Chichewa would follow from the constraint in (25):

(25) **Caus-Appl:** Assign one violation for each exponent of Causative which is not immediately followed by an exponent of Applicative.

- If Caus-Appl outranks all the alignment constraints, it can prevent the mirror order from emerging.

- When the Applicative scopes over the Causative (26), Align-Appl-R will be the highest ranked alignment constraint via the MAP.
  - This advocates for Appl /il/ to surface further to the right than Caus /its/.
  - This is the same order that's preferred by Caus-Appl, so we happen to get a Mirror Principle-compliant order.

(26) **Applicativized Causative** *mang-its-il-* (MP-obeying):
MAP ranking: Align-Appl-R ≫ Align-Caus-R

[[[Root]Caus]Appl]	**Bigram 1**	**MAP constraints**		**Bigram 2**
/mang$_{\text{ROOT}}$, its$_{\text{CAUS}}$, il$_{\text{APPL}}$/	Caus-Appl	Align-Appl-R	Align-Caus-R	Appl-Caus
a. ☞ mang-its-il- [CA]			** (il)	*
b. mang-il-its- [AC]	*!	** (its)		

- But when the Causative scopes over the Applicative (27), Align-Caus-R will be the highest ranked alignment constraint via the MAP.
  - This advocates for Caus /its/ to surface further to the right than Appl /il/.
  - This contradicts the order preferred by Caus-Appl.
  → As long as Caus-Appl outranks all the alignment constraints, we'll still select the [CA] order, which now violates the Mirror Principle.

(27) **Causativized Applicative** *mang-its-il-* (MP-violating):
MAP ranking: ALIGN-CAUS-R ≫ ALIGN-APPL-R

[[[Root]Appl]Caus] /mang$_{ROOT}$, its$_{CAUS}$, il$_{APPL}$/	**Bigram 1**	**MAP constraints**		**Bigram 2**
	CAUS-APPL	ALIGN-CAUS-R	ALIGN-APPL-R	APPL-CAUS
a. ☞ mang-its-il- [CA]		** (il)		*
b. mang-il-its- [AC]	*!		** (its)	

- To capture the full CARP template, we'd need to multiply the set of high-ranked bigram constraints to include all pairwise combinations of CARP elements in that order (to the extent that all combinations are attested independently).

3.5 Capturing asymmetric compositionality with bigrams + MAP

- How can bigram constraints help us capture asymmetric compositionality? *Variable ranking.*
  → We can characterize cases of asymmetric compositionality as rankings where the CARP bigram constraint stands in an "underlyingly" *variable* ranking with the MAP constraints.
  ◦ Alternatively, similar constraint weighting in Harmonic Grammar.

- Just as in the fixed ordering case, when the MAP ranking prefers the same output as the CARP bigram constraint CAUS-REC (28) — i.e. when Reciprocal scopes over Causative — we always get the Mirror Principle-compliant order (29).
  ◦ Since the MAP and the bigram constraint pull in the same direction, it doesn't matter which constraint ranks higher — you'll always get the CARP output.

(28) **CAUS-REC:** Assign one violation for each exponent of Causative which is not immediately followed by an exponent of Reciprocal.

(29) **Possible rankings and outputs for [[[Root]Caus]Rec]**

i. CAUS-REC ≫ {ALIGN-REC-R ≫ ALIGN-CAUS-R} ⇒ *Output*: CR

[[[Root]Caus]Rec] /mang$_{ROOT}$, its$_{CAUS}$, an$_{REC}$/	**Bigram 1**	**MAP 1**	**MAP 2**	**Bigram 2**
	CAUS-REC	ALIGN-REC-R	ALIGN-CAUS-R	REC-CAUS
a. ☞ mang-its-an- [CR]			** (an)	*
b. mang-an-its- [RC]	*!	** (its)		

ii. {ALIGN-REC-R ≫ ALIGN-CAUS-R} ≫ CAUS-REC ⇒ *Output*: CR

[[[Root]Caus]Rec] /mang$_{ROOT}$, its$_{CAUS}$, an$_{REC}$/	**MAP 1**	**MAP 2**	**Bigram 1**	**Bigram 2**
	ALIGN-REC-R	ALIGN-CAUS-R	CAUS-REC	REC-CAUS
a. ☞ mang-its-an- [CR]		** (an)		*
b. mang-an-its- [RC]	*!* (its)		*	

- But when the MAP ranking prefers the *non*-CARP order (i.e. the mirror order would violate CARP), as it does when Causative scopes over Reciprocal (30), the MAP and the bigram conflict.
  - ◦ If a derivation selects the ranking where the bigram constraint outranks the MAP constraints (30.i), it will select the CARP-obeying order, even though this order violates the Mirror Principle.
  - ◦ If a derivation selects the ranking where the MAP constraints outrank the bigram constraint (30.ii), it will not select the Mirror Principle-compliant order, which is CARP-violating.
  - ∗ The only constraints that actually play any role in the evaluation are MAP 1 and Bigram 1.

(30) **Possible rankings and outputs for [[[Root]Rec]Caus]**

i. CAUS-REC ≫ {ALIGN-CAUS-R ≫ ALIGN-REC-R} ⇒ *Output*: CR

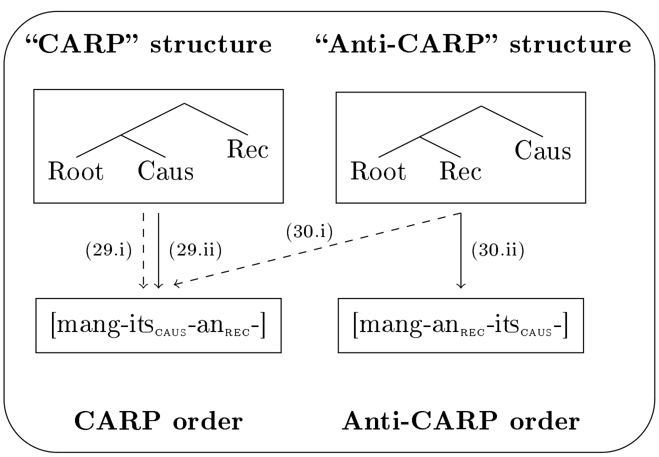
[[[Root]Rec]Caus]	**Bigram 1**	**MAP 1**	**MAP 2**	**Bigram 2**
/mang$_{\text{ROOT}}$, its$_{\text{CAUS}}$, an$_{\text{REC}}$/	CAUS-REC	ALIGN-CAUS-R	ALIGN-REC-R	REC-CAUS
a. ☞ mang-its-an- [CR]		** (an)		*
b. mang-an-its- [RC]	*!		** (its)	

ii. {ALIGN-CAUS-R ≫ ALIGN-REC-R} ≫ CAUS-REC ⇒ *Output*: RC

[[[Root]Rec]Caus]	**MAP 1**	**MAP 2**	**Bigram 1**	**Bigram 2**
/mang$_{\text{ROOT}}$, its$_{\text{CAUS}}$, an$_{\text{REC}}$/	ALIGN-CAUS-R	ALIGN-REC-R	CAUS-REC	REC-CAUS
a. mang-its-an- [CR]	*!* (an)			*
b. ☞ mang-an-its- [RC]		** (its)	*	

- We can summarize the tableaux in (29) and (30) by returning to the possible mappings between structure and order from (18).
  - ◦ The dashed lines represent the cases where the CARP bigram constraint (CAUS-REC) is selected as the higher-ranked constraint. Regardless of syntactic input, this maps to the CARP order.
  - ◦ The solid lines represent the cases where the MAP constraint is selected as the higher-ranked constraint. This selects the order that complies with the Mirror Principle.
- → This is the only way to get the CARP-violating order (30.ii).

(31) **Permissible mappings between structure and order**



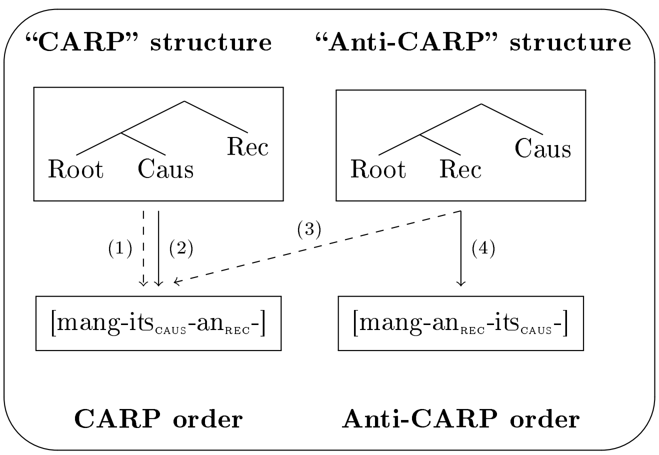
3.6 Local conclusions

- "Morphological templates" like CARP are a problem for the Mirror Principle, because they allow (or, in some instances, require) violating it.
- Trying to capture these kinds of morphological templates with a cyclic concatenation approach to the Mirror Principle is inelegant at best.
  - You'd have to posit that the morphology puts together the wrong order by cyclic concatenation.
  - Then something comes in later to clean things up — maybe Local Dislocation (Embick & Noyer 2001), or some other affix movement operation.
- Using a constraint-based analysis at a late stage of the derivation allows us to derive both fixed ordering and asymmetric compositionality in a direct way.

4 Templatic morphology with Base-Derivative faithfulness

- Here's our mappings again:

(32) **Permissible mappings between structure and order**



- The MAP system works nicely for asymmetric compositionality and fixed ordering. But there's other non-cyclic stuff associated with the CARP template too.
  → To handle this, we might want to adjust our model a bit.

4.1 Suffix doubling in Chichewa

- In Chichewa, Applicative and Reciprocal combinations are obligatorily CARP obeying (essentially, fixed ordering). That is, only the equivalents of the dotted line mappings in (32) are licit.

(33) a. *mang-il-an-*
tie-APPL-REC-
✓'tie each other for'
✓'tie for each other'

b. **mang-an-il-*
tie-REC-APPL-
✗'tie each other for'
✗'tie for each other'

- As in fixed ordering generally, this means the bigram constraint — APPL-REC — is undominated and thus unviolated.

- But actually, one more output is permitted:

(34) a. *mang-an-il-an-*
tie-REC-APPL-REC-
✓'tie each other for'
✗'tie for each other'

b. **mang-il-an-il-*
tie-REC-APPL-REC-
✗'tie each other for'
✗'tie for each other'

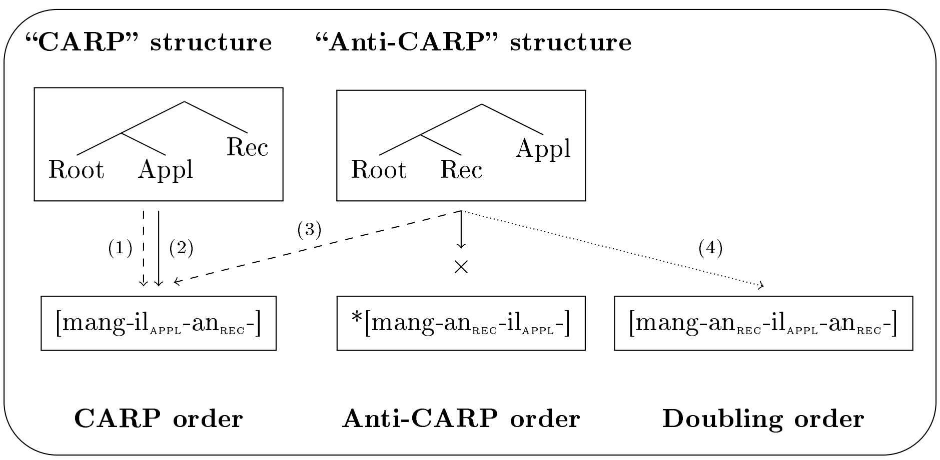
- Here's the whole distribution summed up:

(35) Permitted orderings of Applicative /il/ + Reciprocal /an/ in Chichewa
(Hyman & Mchombo 1992:351ff., Hyman 2003:253ff.)

Single exponents		| **Structure:**	i. [[[Root] Appl] Rec]	ii. [[[Root] Rec] Appl]
a.	Appl-Rec (CARP)	*mang-il-an-*	✓ (MP)	✓
b.	Rec-Appl	*mang-an-il-*	✗	✗ (MP)
Doubled exponents				
c.	Appl-Rec-Appl	*mang-il-an-il-*	✗	✗
d.	Rec-Appl-Rec	*mang-an-il-an-*	✗	☑ ✓

⋆ What is (34a)? It's basically the equivalent of mapping #4 in (32), but to something other than the anti-CARP (mirror) ordering:

(36) **Permissible mappings between structure and order: Applicative and Reciprocal**

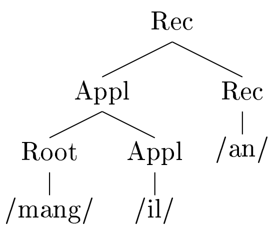


- The MAP on its own can't get this result — it can really only give you asymmetric compositionality, fixed ordering, or free ordering (this is when two opposing bigrams are variably ranked with each other, but both ranked above the MAP alignment constraints; cf. Ryan 2010).

4.2 The solution: Contiguity-BD

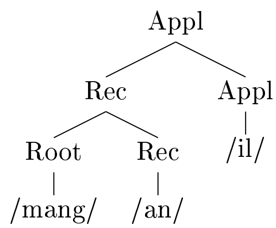
- What would happen if we tried to build these forms up with cyclic concatenation:

(37) **Reciprocalized Applicative:**



[[[mang]APPL]REC] → [[mang-il]REC] → [mang-il-an-]

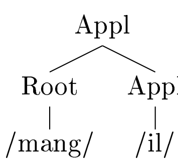
(38) **Applicativized Reciprocal:**



[[[mang]REC]APPL] → [[mang-an]APPL] → *[mang-an-il-]

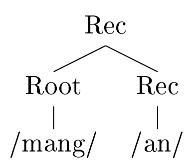
- The first step of cyclic concatenation in both cases actually represents an actual derivation — the simple Applicative and the simple Reciprocal, respectively:

(39) **Applicative:**



[[mang]APPL] → [mang-il-]

(40) **Reciprocal:**



[[mang]REC] → [mang-an-]

- Now, what would happen if we just mapped everything straight onto the CARP template?

(41) **CARP template:** [mang-il-an-]

  - Obviously no conflict for the Reciprocalized Applicative.
  - Obvious conflict for the Applicativized Reciprocal.

- Now let's take another look at the doubling output for the Applicativized Reciprocal:

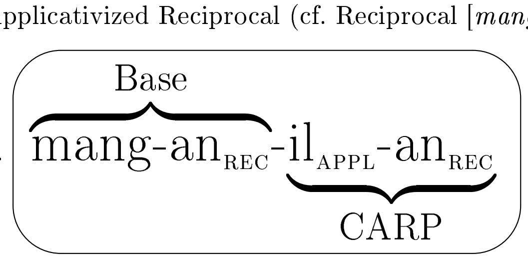
(42) **Doubling output:** [mang-an-il-an-]

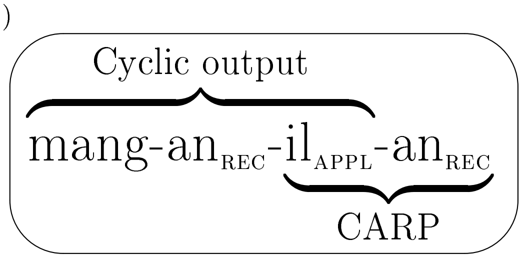
⋆ We now have all the pieces we need in order to map the Applicativized Reciprocal (38) onto the Doubling output (42).

1. The sequence *mang-an-* ([[Root]Rec]) is present in the simple Reciprocal (40).
2. The sequence *-il-an-* (*Appl-Rec*) is what is called for by the CARP template (41).

→ If we stick these two things together (43a), we get our Doubling output:

(43) Applicativized Reciprocal (cf. Reciprocal [*mang-an*$_{\text{REC}}$])

a.


b.


- How do we get one output form to indirectly influence another without cyclic concatenation?
  → Base-derivative correspondence and faithfulness (Benua 1997)

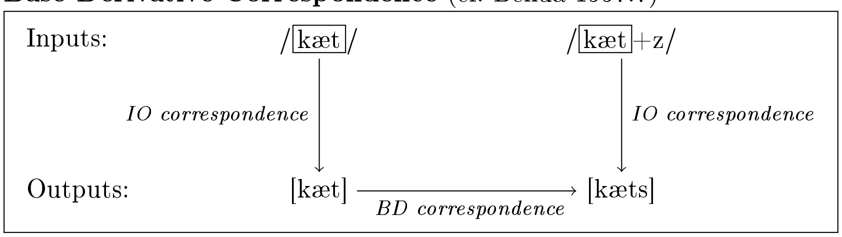
∗ Clearly, (43b) shows that the would-be output of cyclic concatenation is also contained in the Doubling output. So we could alternatively consider trying to use that fact directly.

4.3 BD-Correspondence Theory

⋆ Output-Output / Base-Derivative Correspondence Theory (Benua 1995, 1997, Burzio 1996, Kenstowicz 1996, Kager 1999, *et seq.*) is a way to generate "cyclic" effects without a literal cycle.

- In this theory, in addition to Input-Output (IO) correspondence/faithfulness, there is a separate dimension of correspondence/faithfulness that holds between a derivative and a lexically-related "base":

(44) **Base-Derivative Correspondence** (cf. Benua 1997:7)



- When a property that is present in the output base (in this example, the word [kæt]) would not otherwise be present in the output of the derivative, a high-ranked BD-faithfulness constraint can trigger application (or non-application) of a process in the derivative, if even the right phonotactic context is not met in the derivative.

- Cyclic theories of phonology propogate that property into the derivative by assuming that it automatically inherits it from the cyclic base.
- BD-correspondence holds that this is controlled by constraint ranking, and thus is not automatic.

⋆ In our suffix doubling case, if we treat the output of the **immediate morphosyntactic subconstituent** as the "**base**" for BD-correspondence, then properties of that base (in this case, the adjacency between Root and Rec) can be called upon to account for the extra *-an* adjacent to the Root.

4.4 A BD-correspondence analysis of suffix doubling

4.4.1 Constraints

- The analysis requires only three constraints:

  1. The bigram constraint:

(45) **Appl-Rec:** When exponents of Applicative and Reciprocal are both present in the output, assign a violation if an exponent of Applicative is not followed by an exponent of Reciprocal.

  ▪ Notice that this is a slightly different kind of definition than I used before for the bigram constraints above (25, 28). The existential quantification in this definition is going to make our lives easier.

  ◦ This constraint is undominated, as reflected in the fact that all licit outputs have the sequence Appl-Rec somewhere in the suffix string.

- The other two constraints will be in a variable ranking relationship (equivalent to what we saw earlier in the MAP analysis). (See McCarthy & Prince 1995 for constraint definitions.)

2. The BD-faithfulness constraint, enforcing retention of the base sequence, where what the "base" is depends on the morphosyntactic structure:

(46) **Contiguity-BD:** Assign one violation for each pair of segments which are adjacent in the base but not adjacent in the derivative.

3. The constraint that penalizes suffix doubling, which I will treat as splitting of input segments, meaning that the relevant constraint is:

(47) **Integrity-IO:** Assign one violation for each segment in the input with multiple correspondents in the output.

- We can represent the ranking as:

(48) **Ranking:** Appl-Rec ≫ Contig-BD ∼ Integ-IO

4.4.2 Tableaux

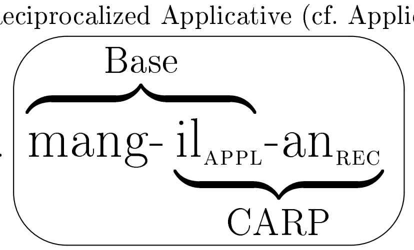
- When this ranking is applied to the Reciprocalized Applicative ([[[Root]Appl]Rec]) (37), we get a unique output (49).

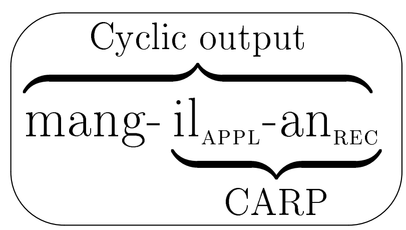
(49) Realizations of [[[Root]Appl]Rec]

Base: [[Root]Appl] (*Root-Appl*) Input: [[[Root]Appl]Rec]			Appl-Rec	Integ-IO	Contig-BD
a. ☞	*mang-il-an-*	(Root-Appl-Rec)			
b.	*mang-an-il-*	(Root-Rec-Appl)	*!		*
c.	*mang-il-an-il-*	(Root-Appl-Rec-Appl)		*!	
d.	*mang-an-il-an-*	(Root-Rec-Appl-Rec)		*!	*!

- We get the unique output because there is no constraint conflict. We see why when we look at the pieces:

(50) Reciprocalized Applicative (cf. Applicative [mang-il$_{\text{APPL}}$])

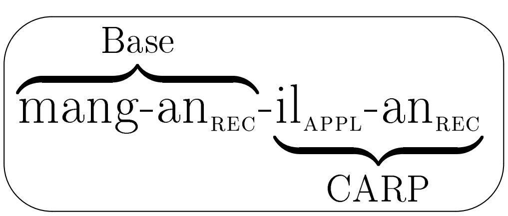
a. 

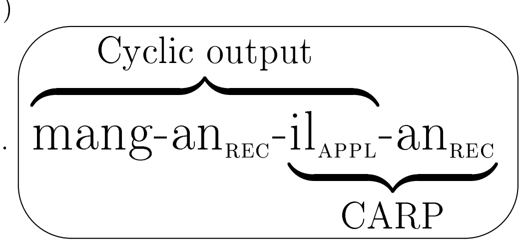
b. 

→ The same string is able to realize the base/cyclic order and the CARP order at the same time. Therefore, doubling (49c) would be superfluous at best (unmotivated Integ-IO violation).

- The same is not true for the Applicativized Reciprocal [[[Root]Rec]Appl] (38). As seen before, the base/cyclic order and the CARP order can't be effectuated with the same string:

(51) Applicativized Reciprocal (cf. Reciprocal [*mang-an*$_{\text{REC}}$])

a. 

b. 

- This can be seen when we look at the constraint violations in the tableaux.
  - Since we are now looking at a different syntactic structure, there is a different base:
    - Now [[Root]Rec] (*Root-Rec*) as opposed to [[Root]Appl] (*Root-Appl*).
  - This means that there will now be a CONTIG-BD violation in the (a) candidate, which is the simple CARP output, rather than the (b) candidate, the simple anti-CARP output.
    - In other words, the (non-doubling) candidate which is faithful to the base is now the anti-CARP output.
  - CONTIG-BD violations also switch for the two doubling candidates, because they differ in what the first suffix is.

- As long as APPL-REC is undominated, it will eliminate the anti-CARP candidate (b) right off the bat.

→ Importantly, this is the cyclic output.

- Among the remaining candidates, the CARP output (a) and the Reciprocal-doubling output (d) trade off single violations of the lower-ranked constraints.

⋆ If those two constraints are variably ranked, the different resolutions of that ranking will produce exactly the two desired outputs:
  - The CARP output (a) if INTEG-IO is ranked higher (52), because it's worse to double than to be unfaithful to the base.
  - the Reciprocal-doubling output (d) if CONTIG-BD is ranked higher (53), because it's worse to be unfaithful to the base than to double.

(52) Realization of [[[Root]Rec]Appl] when INTEG-IO ≫ CONTIG-BD

BASE: [[Root]Rec] (*Root-Rec*) INPUT: [[[Root]Rec]Appl]			APPL-REC	INTEG-IO	CONTIG-BD
a. ☞	*mang-il-an-*	(Root-Appl-Rec)			*
b.	*mang-an-il-*	(Root-Rec-Appl)	*!		
c.	*mang-il-an-il-*	(Root-Appl-Rec-Appl)		*!	*
d.	*mang-an-il-an-*	(Root-Rec-Appl-Rec)		*!	

(53) Realization of [[[Root]Rec]Appl] when CONTIG-BD ≫ INTEG-IO

BASE: [[Root]Rec] (*Root-Rec*) INPUT: [[[Root]Rec]Appl]			APPL-REC	CONTIG-BD	INTEG-IO
a.	*mang-il-an-*	(Root-Appl-Rec)		*!	
b.	*mang-an-il-*	(Root-Rec-Appl)	*!		
c.	*mang-il-an-il-*	(Root-Appl-Rec-Appl)		*!	*
d. ☞	*mang-an-il-an-*	(Root-Rec-Appl-Rec)			*

- This gets us the distribution perfectly.
  - We don't need the MAP to play any role here.
    - Though if the MAP alignment constraints all ranked below all of these constraints, nothing bad would happen. This is what I did by accident in the Zukoff (2023).
    - So MAP and CONTIG-BD are compatible theories.
  - So, could we have actually done asymmetric compositionality with CONTIG-BD and without the MAP?

4.5 Asymmetric compositionality with CONTIG-BD

- Yep. If we replace the MAP alignment constraints directly with CONTIG-BD, placing it in variable ranking with CAUS-REC, we derive:
  - Variation with the Causativized Reciprocal (54–55)
  - A unique output with the Reciprocalized Causative (56)

(54) Realization of [[[Root]Rec]Caus] when CAUS-REC ≫ CONTIG-BD

BASE: [[Root]Rec] (*Root-Rec*) INPUT: [[[Root]Rec]Caus]	CAUS-REC	CONTIG-BD
a. ☞ *mang-its-an-* (Root-Caus-Rec)		*
b. *mang-an-its-* (Root-Rec-Caus)	*!	

(55) Realization of [[[Root]Rec]Caus] when CONTIG-BD ≫ CAUS-REC

BASE: [[Root]Rec] (*Root-Rec*) INPUT: [[[Root]Rec]Caus]	CONTIG-BD	CAUS-REC
a. *mang-its-an-* (Root-Caus-Rec)	*!	
b. ☞ *mang-an-its-* (Root-Rec-Caus)		*

(56) Realization of [[[Root]Caus]Rec]

BASE: [[Root]Caus] (*Root-Caus*) INPUT: [[[Root]Rec]Caus]	CAUS-REC	CONTIG-BD
a. ☞ *mang-its-an-* (Root-Caus-Rec)		
b. *mang-an-its-* (Root-Rec-Caus)	*!	*!

∗ Lingering problem with ranking transitivity:
- If CAUS-REC ∼ CONTIG-BD, and CONTIG-BD ∼ INTEG-IO, then CAUS-REC ∼ INTEG-IO
- This *incorrectly* predicts that suffix doubling should be available for the Causativized Reciprocal.

→ MaxEnt is the answer (we can talk about it if there's time).

- This is clearly a much simpler analysis than the MAP, so it is to be preferred if it can capture all the relevant data.
  - ⋆ BD-faithfulness, independent of the MAP, also captures cyclic opacity in Nyakyusa (see my recent posters, we can also go over this if there's time).
- One reason CONTIG-BD works well here is because the language is pretty much concatenative, even when it's not straightforwardly cyclic.

→ In the next two weeks, we'll look at nonconcatenative systems (root-and-pattern morphology in Arabic, mobile affixation in Huave) to see whether it can do the job there, or whether we need the MAP after all.

5 Cyclic opacity in Nyakyusa templatic morphology

- We've just seen that a Base-Derivative faithfulness model derives asymmetric compositionality and suffix doubling in Chichewa relating to the CARP template.
  → This model also explains a curious case of opacity in Nyakyusa (Bantu; Persohn 2017) relating to templatic morphology.

5.1 Data

- Nyakyusa has another verbal extension: "Transitive" /i̧/ (a.k.a. "short causative").
  ◦ Nyakusa templatically orders Reciprocal before Transitive (cf. (57)), regardless of scope (58c,d).

(57) **Nyakyusa's "CARTP" template:** CAUSATIVE-APPLICATIVE-RECIPROCAL-TRANSITIVE-PASSIVE

- This suffix triggers spirantization on immediately preceding segments: e.g. simple Transitive (58b).

⋆ Only in the Reciprocalized Transitive (58d), spirantization **overapplies** to the root *even though the target and trigger are not adjacent* in the output.

(58) Transitive and reciprocal (Myler 2017:105, citing Hyman 2000:9)

a.	[so**b**-]	'get lost (intr.)'		
b.	[so**f**-i̧-]	'lose' (tr.)'	Transitive	[[Root]Trans]
c.	[so**b**-an-i̧-]	'get each other lost'	Transitivized Reciprocal	[[[Root]Rec]Trans]
d.	[so**f**-an-i̧-]	'lose each other'	Reciprocalized Transitive	[[[Root]Trans]Rec]

- This is a cyclic effect (as suggested by Hyman 2003, *a.o.*), which we can explain using the exact same BD-faithfulness model as before.

5.2 Normal spirantization

- In Nyakyusa, spirantization applies to various consonants, as follows:

(59) **Spirantization** (Hyman 2003:269, Persohn 2017:85)

a. Coronals/dorsals: /t,l,j,k,g/ → [s] / _ i̧

b. Labials: /p,b/ → [f] / _ i̧

  ◦ Notably, it does not apply to nasals.

- We don't need to care too much about the details of the process, only that it involves a $\mathbb{M} \gg \mathbb{F}$-IO ranking:

(60) **Ranking:** $*C_{[-\text{strident}]}\text{i̧} \gg$ IDENT[±strident]-IO

  ◦ Assume that [s,f] are [+strident], all the targets are [−strident], and /n/ is [0strident].

(61) **Spirantization in the Transitive** (trigger is present)

/sob, i̯$_{\text{TRANS}}$/		*$C_{[-\text{strident}]}$i̯	Ident[±strident]-IO
a.	sob-i̯-	*!	
b. ☞	sof-i̯-		*

(62) **No spirantization in the Reciprocal** (trigger is not present)

/sob, an$_{\text{REC}}$/		*$C_{[-\text{strident}]}$i̯	Ident[±strident]-IO
a. ☞	sob-an-		
b.	sof-an-		*!

5.3 Opaque spirantization

- We'll implement fixed templatic ordering in the same way as before: an undominated bigram constraint Rec-Trans (cf. Ryan 2010). (Any definition will do here.)

- The generalization driving the analysis:

(63) Spirantization overapplies iff spirantization occurred normally in the base.

- This only happens if the base is the simple Transitive ((58b)/(61)).
  - As long as the base continues to be morphosyntactically determined, the simple Transitive will only serve as the base for the Reciprocalized Transitive ((58d)/(64)).
  - The Transitivized Reciprocal ((58c)/(65)) will take as its base the simple Reciprocal (62), which does not undergo spirantization.

⋆ Once these base-derivative relations are established, we can use a BD-faithfulness constraint — something like Ident[±strident]-BD — to (over-)apply the process just in case it applied in the base, i.e. when the base is [[Root]Trans].

(64) **Opaque spirantization in Reciprocalized Transitive (58d)**

Base: [sof$_{\text{RT}}$-i̯$_{\text{TRANS}}$-] [[Rt]Trans] Input: /sob$_{\text{RT}}$, i̯$_{\text{TRANS}}$, an$_{\text{REC}}$/ [[[Rt]Trans]Rec]		Rec-Trans	Id[str]-BD	*$C_{[-\text{str}]}$i̯	Id[str]-IO
a.	sob-an-i̯- (Root-Rec-Trans)		*!		
b. ☞	sof-an-i̯- (Root-Rec-Trans)				*
c.	sob-i̯-an- (Root-Trans-Rec)	*!	*!	*!	
d.	sof-i̯-an- (Root-Trans-Rec)	*!			*

(65) **Regular non-spirantization in Transitivized Reciprocal (58c)**

Base: [sob$_{\text{RT}}$-an$_{\text{REC}}$-] [[Rt]Rec] Input: /sob$_{\text{RT}}$, i̯$_{\text{TRANS}}$, an$_{\text{REC}}$/ [[[Rt]Rec]Trans]		Rec-Trans	Id[str]-BD	*$C_{[-\text{str}]}$i̯	Id[str]-IO
a. ☞	sob-an-i̯- (Root-Rec-Trans)				
b.	sof-an-i̯- (Root-Rec-Trans)		*!		*
c.	sob-i̯-an- (Root-Trans-Rec)	*!	*!	*!	
d.	sof-i̯-an- (Root-Trans-Rec)	*!			*

5.4 Local conclusions

- This analysis is not available for a simple cyclic model, because the templatic order in the crucial case is Mirror Principle-violating.
  - A cyclic model would have to involve something like "interfixation" of the Reciprocal in (64):

(66) [[[sob]TRANS]REC] → [[sob-i̧]REC] $\xrightarrow{\text{spirantization}}$ [[so**f**-i̧]REC] $\xrightarrow{\text{"interfixation"}}$ [so**f**-<u>an</u>-i̧-~~an~~]

  - Not only does this disallow bracket erasure (*how else would it know to target a morpheme boundary?*), it introduces an unconstrained morphological movement operation that must follow phonological rule application.

- The MAP also has nothing to say about this pattern — it's not incompatible, but doesn't add anything beyond what the bigram and BD-faithfulness constraint provide.

References

Benua, Laura. 1995. Identity Effects in Morphological Truncation. In Jill Beckman, Suzanne Urbanczyk & Laura Walsh Dickey (eds.), *Papers in Optimality Theory* (University of Massachusetts Occasional Papers in Linguistics 18), 77–136. Amherst, MA: Graduate Linguistics Student Association.

———. 1997. Transderivational Identity: Phonological Relations Between Words. PhD Dissertation, University of Massachusetts, Amherst.

Burzio, Luigi. 1996. Surface constraints versus Underlying Representation. In Jacques Durand & Bernard Laks (eds.), *Current Trends in Phonology: Models and Methods*, 97–122. Salford: University of Salford Publications.

Embick, David & Rolf Noyer. 2001. Movement Operations after Syntax. *Linguistic Inquiry* 32(4):555–595.

Hyman, Larry M. 2000. Bantu Suffix Ordering and its Phonological Consequences. Talk Presented at University of California, Berkeley.

———. 2003. Suffix Ordering in Bantu: A Morphocentric Account. In Geert Booij & Jaap van Marle (eds.), *Yearbook of Morphology 2002*, 245–281. Kluwer.

Hyman, Larry M. & Sam Mchombo. 1992. Morphotactic Constraints in the Chichewa Verb Stem. In *BLS 18: Proceedings of the Eighteenth Annual Meeting of the Berkeley Linguistics Society: General Session and Parasession on The Place of Morphology in a Grammar (1992)*, 350–364.

Inkelas, Sharon & Cheryl Zoll. 2007. Is Grammar Dependence Real? A Comparison Between Cophonological and Indexed Constraint Approaches to Morphologically Conditioned Phonology. *Linguistics* 45(1):133–171.

Kager, René. 1999. *Optimality Theory*. Cambridge: Cambridge University Press.

Kenstowicz, Michael. 1996. Base-Identity and Uniform Exponence: Alternatives to Cyclicity. In Jacques Durand & Bernard Laks (eds.), *Current Trends in Phonology: Models and Methods*, 363–393. Salford: University of Salford Publications.

McCarthy, John J. & Alan Prince. 1993. Generalized Alignment. In Geert Booij & Jaap van Marle (eds.), *Yearbook of Morphology 1993*, 79–153. Kluwer. doi:10.1007/978-94-017-3712-8_4.

———. 1995. Faithfulness and Reduplicative Identity. In Jill Beckman, Suzanne Urbanczyk & Laura Walsh Dickey (eds.), *Papers in Optimality Theory* (University of Massachusetts Occasional Papers in Linguistics 18), 249–384. Amherst, MA: Graduate Linguistics Student Association.

Myler, Neil. 2017. Exceptions to the Mirror Principle and Morphophonological 'Action at a Distance': The Role of 'Word'-Internal Phrasal Movement and Spell-Out. In Heather Newell, Máire Noonan, Glyne Piggott & Lisa Travis (eds.), *The Structure of Words at the Interfaces*, 100–125. Oxford: Oxford University Press.

Persohn, Bastian. 2017. *The Verb in Nyakyusa: A Focus on Tense, Aspect and Modality* (Contemporary African Linguistics 2). Berlin: Language Science Press. doi:10.5281/ZENODO.926408.

Pylkkänen, Liina. 2002. Introducing Arguments. PhD Dissertation, Cambridge, MA MIT.

Ryan, Kevin M. 2010. Variable Affix Order: Grammar and Learning. *Language* 86(4):758–791.

Sande, Hannah, Peter Jenks & Sharon Inkelas. 2020. Cophonologies by Ph(r)Ase. *Natural Language & Linguistic Theory* 38(4):1211–1261. doi:10.1007/s11049-020-09467-x.

Zukoff, Sam. 2023. The Mirror Alignment Principle: Morpheme Ordering at the Morphosyntax-Phonology Interface. *Natural Language & Linguistic Theory*. 41(1):399–458. doi:10.1007/s11049-022-09537-2.