

When Phonology is Algebra: Solving Problems with Interacting Processes

Chelsea Sanker

Rutgers University
chelsea.sanker@rutgers.edu

NAPhC9, Concordia University

7 May 2016

Solve for X?

- Usually in phonology problems you know the outputs, and are determining (1) the inputs and (2) the function that is producing the observed outputs
- Often students are more accustomed to math problems where they know two of these things and are solving for the other:
 - Output and function, e.g. Solve for x : $6x + 13 = 28$
 - Input and output, e.g. Identify the function:

| input | output |
|-------|--------|
| 2 | 9 |
| 3 | 14 |
| 4 | 21 |
| 5 | 30 |
| 6 | 41 |

Multiple Processes

- When there are several interacting functions, this task is even more complicated, because there is the additional question of what order the functions occur in
- Students often get lost at the very first stage of such problems, because they write out the environments based on the surface forms and then use that as a basis for rules

Types of Interactions

- Because many students rely on the surface forms for their environments, they are good at identifying processes for which the full environment is still present, so some types of interactions are easier than others
- Students also seem to have an easier time keeping track of rules producing the contexts for each other than eliminating them

Types of Interactions

- **Feeding** (one rule creates the conditions for a subsequent one): High accuracy – both rules' environments can be present in the surface forms
- **Counter-feeding** (one rule creates the conditions for a previous one): Moderate accuracy – looks like the first rule should apply more than it does
- **Bleeding** (one rule eliminates the conditions for a subsequent one): Moderate accuracy – both rules' environments can be present in the surface forms
- **Counter-bleeding** (one rule eliminates the conditions for a previous one): Low accuracy – looks like the first rule should apply less than it does

Break it down

- Give students three of the four pieces of information (input, output, rules, rule ordering) and have them work through those to find the fourth
- In particular, make them practice with deriving step-by-step to test the predicted output of the two possible orders of two known rules and a known input
- Without interactions, when students are given input and just one rule they do very well
- When there are interactions, even when they are told the input, the rules, and their order, they still make mistakes, often ignoring some aspect of the ordering

In-class practice

- A lot of solving phonology problems is getting accustomed to different strategies and ways of thinking
- My favorite way to work with this is to have students work on problems during class, so I can come around and help them work through the process; often they get stuck in a particular assumption (e.g. what the underlying form is)
- Just being reminded of their options (rethinking morphological boundaries, breaking changes into multiple interacting rules) will often allow them to figure the problem
- Practice with these strategies makes them more likely to use them on their own

Good practices

In-class work can also allow us to see what strategies students are using and allow us to encourage them to include some important elements in their analytical process:

- Remembering that conditioning environments might be part of what was present in the underlying form that has been changed in the surface form
- Using step-by-step derivations to test rule orderings
- Remembering that in fact usually all of the elements of your analysis are hypotheses, so not to make assumptions and moreover to test your answers (look for counterexamples)

What else can we do to help students with the concept of interacting processes?

| | | | |
|---------------|-----------------|--------------|----------------|
| klup 'club' | klubi 'clubs' | zwup 'crib' | zwobi 'cribs' |
| trup 'corpse' | trupi 'corpses' | lut 'ice' | lodi 'ices' |
| fum 'noise' | fumi 'noises' | vus 'cart' | vozi 'carts' |
| snop 'sheaf' | snopi 'sheaves' | nuf 'knife' | noze 'knife' |
| trut 'labor' | trudi 'labors' | ruk 'horn' | rogi 'horns' |
| kot 'cat' | koti 'cats' | bur 'forest' | bori 'forests' |
| grus 'rabble' | gruzi 'rabbls' | vuw 'ox' | vowi 'oxen' |
| nos 'nose' | nosi 'noses' | sul 'salt' | sole 'salts' |
| koj 'basket' | koje 'baskets' | buj 'fight' | buje 'fights' |
| wuk 'lye' | wugi 'lyes' | mul 'moth' | mole 'moths' |
| wuk 'arch' | wuki 'arches' | bup 'bean' | bobi 'beans' |
| ul 'beehive' | ule 'beehives' | brut 'beard' | brodi 'beards' |

Feeding

e.g.

1. $V > \emptyset / _ \#$
2. $C > [-\text{voice}] / _ \#$

/podo/

pod

pot

[pot]

Counterfeeding

e.g.

1. C > [-voice]/__#
2. V > ∅/__#

/podo/

—

pod

[pod]

Bleeding

e.g.

1. $V > \emptyset / _ \#$
2. $C > [+voice] / V _ V$

/poto/

pot

pot

[pot]

Counterbleeding

e.g.

1. $C > [+voice]/V_V$
2. $V > \emptyset/_ \#$

/poto/
podo
pod
[pod]