Irreducible parallelism in process interactions

Jeffrey Adler (UCSC) & Jesse Zymet (UCLA)

1. Introduction

Regarding HS (Harmonic Serialism; McCarthy 2000, a.o.), McCarthy (2013) asks: for two particular operations op_1 and op_2 and input x, need GEN generate candidate $op_2(op_1(x))$ in a single step? This would constitute *irreducible parallelism* in the sense that op_1 and op_2 must apply to the input in parallel. Four cases have been argued to necessitate $op_2(op_1(x))$ and require parallel OT (Optimality Theory; Prince & Smolensky 1993/2004): assimilation-epenthesis in Lithuanian (Baković 2005, Albright & Flemming 2013), reduplication-glide formation in Maragoli (Zymet 2015), footing-lengthening in Mohawk (Adler 2016), and harmony-deletion in Gurindji (Stanton 2016). Little, however, has been said about what formal properties these systems share (*cf.* McCarthy 2010). In this paper, we elucidate the abstract conditions under which generation of $op_2(op_1(x))$ is necessary.

Assuming the validity of our predecessors' arguments, we show that the cases above are only superficially different. They receive a uniform formal description: the same set of violation profiles and ranking arguments characterizes their successful derivation in OT, and failed derivation in HS. These cases are all instances of a general process interaction: do Operation 1 followed by Operation 2 unless the result is a marked structure, in which case start over and do Operation 3, which is different from Operation 1. To express this interaction, we argue, generation of $op_2(op_1(x))$ is necessary. We summarize Lithuanian assimilation-epenthesis below, then give the abstract characterization of the general process interaction Lithuanian represents, and then schemata of how the four cases all meet this characterization.

2. Lithuanian assimilation-epenthesis

Verbal prefixes (1) undergo voicing and palatal assimilation to following obstruents (2) unless it would result in a geminate, in which case epenthesis (with subsequent assimilation to [i]) takes place instead (3).

(1) <u>Faithful forms</u>
ap-raʃʲiːtʲi,'describe'
ap-tarʲtʲi,'describe'

(2) <u>Assimilated forms</u>
a**b-g**autʲi,'deceive'
a**pʲ-tʲ**emʲdʲiːtʲi,'obscure'
a**bʲ-dʲ**egʲtʲi,'get'

(3) <u>Epenthetic forms</u>
apʲ**i**-barʲtʲi (*a**b-b**arʲtʲi),'spill on'
apʲ**i**-bʲerʲtʲi (*a**bʲ-bʲ**erʲtʲi),'strew'

Parallel OT can express this interaction. DEP ≫ IDENT broadly favors full assimilation to epenthesis (4b), but NOGEM ≫ DEP favors epenthesis where assimilation would yield a geminate (4d).[1]

(4)

	Input	Winner	Loser	AGR	NOGEM	DEP	IDENT
a.	p+dʲ	bʲ-dʲ	p-dʲ	W			L
b.	p+dʲ	bʲ-dʲ	pʲi-dʲ			W	L
c.	p+bʲ	pʲi-bʲ	p-bʲ	W		L	
d.	p+bʲ	pʲi-bʲ	bʲ-bʲ		W	L	W

However, HS fails (Albright & Flemming 2013).[2] Assimilation requires two steps to fully satisfy AGR while epenthesis only one, so DEP must rank higher than AGR for assimilation to ever apply (5a). This incorrectly predicts that epenthesis never applies. In particular, NOGEM cannot block assimilation since the geminate candidate cannot be compared against the epenthetic candidate (5c). With the ranking below, the derivation converges on the pathological form *[b-bʲ] (5d). The desired forms can only be derived if multiply assimilated candidates are generated in Step 1.

(5)

	Step	Input	Winner	Loser	NOGEM	DEP	AGR	IDENT
a.	1	p+dʲ	b-dʲ	pi-dʲ		W	L	L
b.	2	b+dʲ	bʲ-dʲ	b-dʲ			W	L
c.	1	p+bʲ	💣 b-bʲ	☹ pi-dʲ		W	L	L
d.	2	b+bʲ	💣 b-bʲ	bʲ-bʲ	W		L	W

---

[1] We collapse AGR[VOI] and AGR[PAL] to highlight the essential rankings.

[2] An HS analysis with constraints banning similar but non-identical adjacent obstruents might succeed, but see Baković (2005), Pająk & Baković (2010) for arguments against the existence of such constraints.

3. Conspiracy of procedures

The Lithuanian case is an example of a *conspiracy of procedures*, in which the same driving constraint (AGR above) is satisfied by two distinct sequences of operations (multiple assimilation vs. epenthesis). The driving constraints are ordinarily satisfied by Procedure A, which consists of Operation 1, followed by Operation 2. But, in environments where Procedure A would result in a marked structure, the driving constraints are satisfied by Procedure B, which consists of Operation 3, different from Operation 1. The formal expression of conspiracy of procedures involves the following constraints:

- DRIVER: constraint driving the application of Procedures A & B (eg. AGR in Lithuanian)
- *PROCA: constraint violated in the application of Procedure A repair (IDENT)
- *PROCB: constraint violated in the application of Procedure B repair (DEP)
- BLOCKER: constraint preferring the output of Procedure 2 over that of Procedure 1 (NOGEM)

Let x and y be inputs whose faithful candidates violate the DRIVER. Schematically:

x undergoes Procedure A: /x/ → $op_1(x)$ → $op_2(op_1(x))$ → ..., where op_1, op_2, op_3 are maps of single
y undergoes Procedure B: /y/ → $op_3(y)$ → ..., changes (*cf.* McCarthy 2016), $op_1 \neq op_3$

Parallel OT can express these mappings, since it can compare whole outputs. Procedure A outputs are generally preferred to Procedure B outputs (*PROCB ≫ *PROCA; 6b) unless the former output violates BLOCKER, in which case Procedure B is preferred (BLOCKER ≫ *PROCB; 6d).

	Input	*Winner*	*Loser*	DRIVER	BLOCKER	*PROCB	*PROCA
(6) a.	x	$op_2(op_1(x))$	x	W			L
b.	x	$op_2(op_1(x))$	$op_3(x)$			W	L
c.	y	$op_3(y)$	y	W		L	
d.	y	$op_3(y)$	$op_2(op_1(y))$		W	L	W

HS, on the other hand, cannot express the conspiracy. At the first step of the derivation of $op_2(op_1(x))$, $op_1(x)$ must win over $op_3(x)$. $op_1(x) \sim op_3(x)$ entails a ranking that demotes the constraint that favors op_3, DRIVER, below the constraint that disfavors it, *PROCB (7a). This predicts that y maps to $op_1(y)$ in Stage 1 rather than to $op_3(y)$ as desired (7c), thereby missing the generalization that op_3 applies when op_1 followed by op_2 results in a violation of BLOCKER. GEN therefore must be able to generate candidates that undergo op_1 and op_2 in the same stage so that they can be compared against those that undergo op_3.

	Step	*Input*	*Winner*	*Loser*	BLOCKER	*PROCB	DRIVER	*PROCA
(7) a.	1	x	$op_1(x)$	$op_3(x)$		W	L	L
b.	2	$op_1(x)$	$op_2(op_1(x))$	$op_1(x)$			W	L
c.	1	y	💣 $op_1(y)$	☹ $op_3(y)$		W	L	L
d.	2	$op_1(y)$	💣 $op_1(y)$	$op_2(op_1(y))$	W		L	W

4. Attested conspiracies

In this paper, we show that each of the cases below fits the formal characterization above.

Language	**Driver constraint(s):**	***do* Procedure A…**	***unless result is…***	***else do* Procedure B:**
Lithuanian	Adjacent obstruents agree on [pal] and [voi]	Palat. assim. & voic. assim.	Geminate	[i] epenthesis
Maragoli	Reduplicants are realized; no hiatuses	Gliding → reduplication	Complex reduplicant onset	Reduplication → gliding
Mohawk	Feet are bimoraic	Monosyl. footing → V lengthening	Long epenthetic vowel	Disyllabic footing
Gurindji	Pre-nasal segments are nasal	Iterative [nasal] spreading	$NC_0\tilde{V}$ sequence	[nasal] deletion