Non-Vacuous Duke of York Derivations in Hebrew - OT with Candidate Chains Analysis

Idan Drori

Introduction

In this paper I'm going to assume the reader has a basic familiarity of the workings, nomenclature and notation of both classical Optimality Theory and rule-based phonology. The question I will attempt to answer, is whether McCarthy's Optimality Theory with Candidate Chains is capable of adequately explaining the non-vacuous Duke of York derivations in Hebrew.

Duke of York Derivations

Duke of York derivations are interactions of two phonological processes such that the second undoes the first, i.e $A \to B \to A$ (Pullum, 1976).
A distinction is made between a vacuous Duke of York derivation and a non-vacuous one (McCarthy, 2003). A non-vacuous Duke of York derivation is characterized by some third process, that occurs (or doesn't occur) precisely because A turned to B. And as such provides evidence that A did in fact become B and back again. Vacuous Duke of York derivations on the other hand don't have some third process that happens somewhere in between $A \to B \to A$.
I am going to be discussing non-vacuous Duke of York derivations because they're the ones that OT and its variants generally struggle to explain.

Modern Hebrew

Modern Hebrew has four processes that I'm going to look at, that interact to form a bleeding Duke of York interaction. Analysis from Rasin, 2022
The first two are deletion processes:

1. **ɑ-Deletion** - ɑ gets deleted in the stem-penultimate syllable, this always happens in adjectives and participles, and occasionally in nouns.
   - ɑ $\to \emptyset / \#C\_\sigma\ ˈ\sigma$
2. **e-Deletion** - Deletes unstressed e in a two-sided open-syllable environments. This happens in adjectives, participles and verbs.
   - ĕ $\to \emptyset / VC\_CV$

The last two are epenthesis processes, These generally happen in order to break up impermissible consonant clusters involving sonorants and glottal consonants in a syllable.

- **e-Epenthesis** - e is inserted to break complex onset consisting of either a sonorant (l, m, n, r, j) followed by another consonant, or any consonant followed by one of the glottals h and ʔ. It should be noted that in casual speech the glottals are often omitted, but the epenthesis is triggered nontheless.

  Examples:

  1. me̲sirá 'passing'
  2. je̲niká 'suckling'
  3. pe̲(ʔ)imá 'pulse'
  4. ʃe̲(h)ijá 'stay'
  5. psiká 'ruling'
  6. ʃriká 'whistle'

- **ɑ-Epenthesis** - ɑ is inserted between a glottal consonant or x and another consonant. In addition ɑ is inserted between a non-low vowel and either a word-final x or an unpronounced word-final ʔ

  Examples:

  1. (ʔ)ɑ̲fijá 'baking'
  2. (ʔ)ɑ̲kirá 'uprooting'
  3. (h)ɑ̲lixá 'walking'
  4. xɑ̲simá 'blocking'
  5. ʃimúɑ̲ 'hearing'
  6. pitúɑ̲x 'development'

Written as rules:

- e-Epenthesis 1: $\emptyset \rightarrow e/\#\underset{[+son]}{C}\_C$
- e-Epenthesis 2: $\emptyset \rightarrow e/\#C\_$\{h, ʔ\}
- ɑ-Epenthesis 1: $\emptyset \rightarrow A/$\{h, ʔ\}$\_C$
- ɑ-Epenthesis 2: $\emptyset \rightarrow A/\#x\_C$
- ɑ-Epenthesis 3: $\emptyset \rightarrow A/\underset{[-low]}{V}\_$\{h, ʔ, x\}#

The traditional analysis would order vowel deletion before vowel epenthesis:

1. ɑ-Deletion

2. e-Deletion
3. ɑ-Epenthesis
4. e-Epenthesis

To see the interactions between the rules I'll present the derivation of several adjectives (Rasin, 2022):

UR	zɑken-ím	xiver-ím	me-xɑser-ím	jɑveʃ-ím	**xɑser-ím**
ɑ-Deletion	zkením	-	-	jveʃím	**xserím**
e-Deletion	-	xivrím	mexɑsrím	-	**-**
ɑ-Epenthesis	-	-	-	-	**xɑserím**
e-Epenthesis	-	-	-	jeveʃím	**-**
SR	[zkením]	[xivrím]	[mexɑsrím]	[jeveʃím]	**[xɑserím]**

Note the final column, xɑserím. There we can see that the underlying ɑ is deleted, removing the environment neccessary for e-Deletion, ɑ-Epenthesis applies afterwards and reinstates e-Deletion's environment. And so we get an opaque interaction, e-Deletion underapplies, where the underapplication opacity comes from the counterfeeding interaction between e-Deletion and ɑ-Epenthesis. This is the aforementioned Bleeding Duke of York interaction, the bleeding relation between ɑ-Deletion and e-Deletion is what provides independent evidence for the fact that ɑ is deleted and then reinstated. We also see this same bleeding interaction in the first and fourth columns.

Optimality Theory with Candidate Chains

Optimality Theory (OT) as a general rule does not deal with opaque processes particularly well. In rule-based phonology, opacity is a consequence of derivations, where rules affect intermediate representations between the underlying and the surface forms. The more successful variations on OT that attempt to deal with opacity incorporate some intermediate form, and in so doing emulate derivations in a way.

Possibly the most successful variation on OT to deal with opacity is Stratal OT. However McCarthy shows several examples of opaque interactions that Stratal OT fails to capture. From this McCarthy concludes that a different variation on OT is needed, and he develops Optimality Theory with Candidate Chains, referred from now on as OT-CC (McCarthy, 2007).

The main difference between classical OT and OT-CC is how candidates are defined, instead of a single output form, a candidate is a chain of forms that differ minimally starting from the input form and ending with an output form. Candidate chains approximate a derivation in rule-based phonology.
Chains are ordered n-tuples, and will be represented as a vector, for example: <pat,pati,pat͡ʃi> is a chain representing a process of [i]-epenthesis and palatalization triggered by said [*i*].

Chains must also follow three conditions in order to be considered valid chains:

1. **Faithful First Member** The first member of every candidate chain based on input */ABC/* is fully faithful to that input, meaning no faithfulness constraints are violated. If there are more than one fully faithful candidate, the one that's chosen is determined by the principle of Local Optimality in (3).

2. **Gradualness** A single violation of a faithfulness constraint will be called a *localized unfaithful mapping* or LUM. Gradualness means that each successive form adds exactly one LUM to its predecessor. So a chain of the form <pap,pa.pə,pa.bə> is a valid chain. But a chain of the form <pap,pa.bə> is not, because going from pap to pa.bə is not possible with a single LUM.

3. **Local Optimality** This has two aspects:

   (a) The initial form of the chain is the fully faithful candidate that is most harmonic according to the constraint hierarchy of that particular language. It is the locally optimal candidate out of all the fully faithful candidates.

   (b) every (non-initial) form in a chain is more harmonic than its predecessor, meaning it violates fewer markedness constraints.

Precedence Constraints and Candidate Convergence -

Candidate chains can often contain too much information, for example when both chains have the same input and output but differ in the order of their LUM's. In this case we'll say that the candidates converge and simply treat both of them as the same candidate chain.
However, we must make a distinction between irrelevant LUM orderings and those that are linguistically relevant. For this purpose we'll define a new type of constraint, a PREC (precedence) constraint.
Given faithfulness constraints A and B:

- PREC(A, B) assigns a violation if B is violated before A

A note on chain validity and PREC constraints: Because PREC constraints refer to the set of all LUM's and the order they're applied in the chain, they can't be evaluated until its been determined that the chain converges. Meaning PREC constraints are irrelevant when determining chain validity.

Candidates in OT-CC

McCarthy defines a candidate as an ordered 4-tuple <in, out, $\mathcal{L}$-set, rLUM-Seq>, where
in is a linguistic form, the input.
out is a linguistic form, the output.
$\mathcal{L}$-set is a set of the LUM's that are violated in the chain
rLUMSeq (reduced LUM Sequence) is the partial ordering of a subset of $\mathcal{L}$-set,

i.e, the ordering of relevant crucially ordered constraints.

Notation of rLUMSeq - McCarthy writes rLUMSeq as an n-tuple of the form <ID(back)@3>, where ID(back)@3 means a violation of ID(back) at index 3, assuming that the faithful form is indexed. This sort of notation would be rather confusing when there are deletion and epenthesis processes. And because the relevant interaction I intend to look at involves those kinds of processes, I will not be using this notation. Instead I will just write the rLUMSeq as <A, B, C> where A, B, C are faithfulness constraints, and if it's unclear where exactly a particular faithfulness constraint was violated, I will explain where it happens.

I intend to further simplify the way candidates are represented by simply omitting $\mathcal{L}$-set, as I think figuring out which faithfulness constraints were violated between one form and the next is relatively straightforward and doesn't need to be stated outright. What this means is that in practice, I'll first show the relevant candidates as chains of forms, and beneath them there will be rLUMSeqs. And afterwards in the tableu, in order to save space I will write the final form only and the rLUMSeq beneath it, while the UR would be at the top left as is standard with OT tableaux.

PREC constraints, like all OT constraints can be ranked, but this ranking is limited somewhat. A violation of PREC(A, B) clearly depends on whether B is violated, but B must be ranked higher than PREC(A, B), for all faithfulness constraints A and B, so that PREC(A, B) wont affect B. As an example, lets say we have the faithfulness constraints MAX >> PREC(ID(back), MAX) >> ID(back). So we have a process of palatalization and syncope and we want palatalization to happen before the syncope. If we ranked PREC(ID(back), MAX) above MAX, then the result would be that expected syncope is blocked in the cases where it's not accompanied by palatalization. This is completely unattested and quite weird, essentially syncope is blocked unless palatalization applies in an opaque way. There's no reason to think that any phonological system works or should work this way. For the full example refer to (McCarthy, 2007 99-102)

Modern Hebrew OT-CC Analysis

Note: I am going to ignore the constraints related to stress because they do not affect any of the relevant interactions.

We'll first look at transparent interactions in order to find the basic constraint ordering.

For ɑ-Deletion lets look at /kɑtɑn-ɑ/ → [ktɑnɑ́]. The relevant constraints are: MAX(V), *#Cɑσˈσ.

kɑtɑn-ɑ			*#Cɑσˈσ	MAX(ɑ)
a.		kɑtɑnɑ	*!	
b.	☞	ktɑnɑ		*

From this we conclude that *#Cɑσˈσ >> MAX(ɑ).

For e-Deletion lets look at xiver-im → [xivrím]. The relevant constraints are: MAX(V), *VCĕCV.

xiver-im			*VCĕCV	MAX(e)
a.		xiverím	*!	
b.	☞	xivrím		*

We'll conclude that *VCĕCV >> MAX(e).

For ɑ-Epenthesis we'll look at /xsimɑ/ → [xɑsimɑ́], /ʃimu(ʔ)/ → [ʃimúɑ(ʔ)] and /(ʔ)fijɑ/ → [(ʔ)ɑfijɑ́].
The relevant constraints are: DEP(V), *#xC, *$\underset{[-low]}{V}${h,ʔ,x}, *{h,ʔ}C.

xsima			*#xC	DEP(V)
a.		xsimɑ́	*!	
b.	☞	xɑsimɑ́		*

ʃimu(ʔ)			*$\underset{[-low]}{V}${h,ʔ,x}	DEP(V)
a.		ʃimú(ʔ)	*!	
b.	☞	ʃimúɑ(ʔ)		*

(ʔ)fijɑ			*{h,ʔ}C	DEP(V)
a.		(ʔ)fijɑ́	*!	
b.	☞	(ʔ)ɑfijɑ́		*

From this we conclude that *#xC, *$\underset{[-low]}{V}${h,ʔ,x}, *{h,ʔ}C >> DEP(V). For the sake of simplicity, from here on out I'm only going to look at one of these three markedness constraints at a time, depending on its relevance.

for e-Epenthesis we'll look at /msirɑ/ → [mesirɑ́] and /ʃ(h)ijɑ/ → [ʃe(h)ijɑ́].
The relevant constraints are: DEP(V), *#$\underset{[+son]}{C}$ C, *#C{h,ʔ}.

msirɑ			*#$\underset{[+son]}{C}$ C	DEP(V)
a.		msirɑ́	*!	
b.	☞	mesirɑ́		*

ʃ(h)ijɑ			*#C{h,ʔ}	DEP(V)
a.		ʃ(h)ijá	*!	
b.	☞	ʃe(h)ijá		*

From this we get that $*\#\underset{[+son]}{C}C$, *#C{h,ʔ} >> DEP(V). Like the ɑ-Epenthesis, I'm going to treat these two markedness constraints one at a time, depending on which is relevant.

For the interaction between the deletion processes we'll look at /zɑken-ot/ → [zkenót].
The relevant constraints are: *VCĕCV, *#Cɑσˈσ, MAX(e), MAX(A).

zɑken-ót			*VCĕCV	*#Cɑσˈσ	MAX(e)	MAX(ɑ)
a.		zɑkenót	*!	*		
b.	☞	zkenót				*
c.		zɑknót			*!	

From this we get *VCĕCV, *#Cɑσˈσ, MAX(e) >> MAX(ɑ)

In summary, we get this tentative ranking:

$$
\begin{tikzcd}
 & & & \text{*VCĕCV} \arrow[d] \\
\#\underset{[+son]}{C}C \arrow[dr] & \text{\#xC} \arrow[d] & \text{*\#C}\alpha\sigma\text{ˈ}\sigma \arrow[dr] & MAX(e) \arrow[d] \\
 & DEP(V) & & MAX(A)
\end{tikzcd}
$$

For interactions between epenthesis and deletion we'll look at /po(ʔ)el-ím/ → [po(ʔ)ɑlím], a transparent interaction. This is an alteration that can't be reached with a single LUM, so we must determine valid chains.
Potentially valid chains for input /po(ʔ)el-ím/ and their rLUMSeqs:

	Chain	rLUMSeq	Description
1.	<po(ʔ)elím>	<>	Fully faithful
2.	<po(ʔ)elím, po(ʔ)lím>	<MAX(e)>	Only e-Deletion
3.	<po(ʔ)elím, po(ʔ)lím, po(ʔ)Alím>	<MAX(e), DEP(V)>	e-Deletion followed by ɑ-epenthesis

We want candidate number 3 to be the most harmonic candidate. In order for that to happen, both chain 2 and chain 3 need to be valid chains. In order for chain 2 to be considered valid, it must maintain local optimality, meaning *VCĕCV must be ranked higher than *{h,ʔ}C. And likewise for chain 3 to be valid, *{h,ʔ}C must be ranked higher than DEP(V).

A tableau to illustrate this point:

po(ʔ)el-ím			*VCĕCV	*{h,ʔ}C	MAX(e)	DEP(V)
a.		po(ʔ)elím	∗!			
b.	☞	po(ʔ)ɑlím			∗	∗
c.		po(ʔ)lím		∗!	∗	

So we get that *VCĕCV >> *{h,ʔ}C >> DEP(V).
As a reminder, I'm treating all the constraints regarding ɑ-Epenthesis as equivalent, and so *{h,ʔ}C is equivalent to *#xC in the ranking.

Updated constraint ranking:

$$\begin{tikzcd}
 & & \text{*VCĕCV} \arrow[dl] \arrow[dr] & \\
\text{*\#}\underset{[+son]}{C}C \arrow[dr] & \text{*\#xC} \arrow[d] & \text{*\#Cɑ}\sigma\text{'}\sigma \arrow[dl] \arrow[dr] & MAX(e) \arrow[d] \\
 & DEP(V) & & \text{MAX(ɑ)}
\end{tikzcd}$$

Opaque Interactions

With a tentative ranking at hand, it's time to turn our attention to determining valid chains for opaque interactions. First we'll look at jɑveʃ-ím → jeveʃím. This is a very similar interaction to the bleeding Duke of York, where e-deletion underapplies. Potentially valid chains for the input /jɑveʃ-ím/ and their rLUMSeqs:

	Chain	rLUMSeq	Description
1.	<jɑveʃím>	<>	Fully Faithful
2.	<jɑveʃím, jveʃím>	<MAX(ɑ)>	ɑ-Deletion only
3.	<jɑveʃím, jveʃím, jeveʃím>	<MAX(ɑ), DEP(V)>	ɑ-Deletion followed by e-Epenthesis
4.	<jɑveʃím, jɑvʃím>	<MAX(e)>	e-Deletion

Because the preferred candidate is 3, we need both 2 and 3 to be valid chains.
Candidate 1 violates *#Cɑσ'σ and *VCĕCV.
The second form in candidate 2 violates *#$\underset{[+son]}{C}C$, so that constraint needs to be ranked lower than *#Cɑσ'σ and *VCĕCV in order for candidate 2 to maintain local optimality.
The third form in candidate 3 violates *VCĕCV, so that constraint needs to be ranked lower than *#$\underset{[+son]}{C}\_C$.
In conclusion we need the following to hold *#Cɑσ'σ >> *#$\underset{[+son]}{C}C$ >> *VCĕCV.
But if we fill out a tableau we can see that candidate 4 causes problems:

jɑveʃ-ím	*#Cɑσˈσ	$*\#\underset{[+son]}{C}C$	*VCĕCV	MAX(e)	MAX(ɑ)	DEP(V)
a. jɑveʃím, <>	*!		*			
b. jveʃím, <MAX(ɑ)>		*!			*	
c. jeveʃím, <MAX(ɑ), DEP(V)>,			*!		*	*
d. ☞ jɑvʃím, <MAX(e)>				*		

We can't rank MAX(e) higher than *VCĕCV because this would be a ranking paradox.
Splitting DEP(V) into DEP(e) and DEP(ɑ) and trying with some PREC constraint from there wouldn't work either. From /po(ʔ)el-ím/ → [po(ʔ)ɑlím] we'll get that *VCĕCV >> DEP(ɑ). And so using a constraint like PREC(MAX(e), DEP(ɑ)) won't help because *VCĕCV >> DEP(ɑ) >> PREC(MAX(e), DEP(ɑ)).

It seems that the underapplication of e-deletion poses a problem for OT-CC, the candidate where e-deletion applies transparently is the optimal candidate. The failure likely comes from the fact that without the involvement of PREC constraints, OT-CC is essentially just OT, and so works in parallel as opposed to serially. And as has been mentioned before, parallel theories generally struggle to explain opaque interactions. This is potentially an ill omen for the upcoming Duke of York derivation, as it has a similar underapplication of e-Deletion, but it's a different interaction and so warrants its own discussion.

Current Constraint Ranking:

$$
\begin{tikzcd}
 & *\#\text{C}ɑ\sigma\text{ˈ}\sigma \arrow[d] & \\
 & *\#\underset{[+son]}{C}C \arrow[d] & \\
 & *\text{VCĕCV} \arrow[dl] \arrow[dr] & \\
*\#xC \arrow[d] & & MAX(e) \arrow[d] \\
DEP(V) & & \text{MAX(ɑ)}
\end{tikzcd}
$$

Bleeding Duke of York

With a constraint ranking in hand, let's look at the alternation with the bleeding Duke of York: /xɑser-ím/ → [xɑserím].
Potentially valid chains for the input /xɑser-ím/ and their rLUMSeqs:

Candidate 3 is the chain we want, but a question emerges, is this even a valid chain?

	Chain	rLUMSeq	Description
1.	<xɑserím>	<>	Fully faithful
2.	<xɑserím, xserím>	<MAX(ɑ)>	Only ɑ-Deletion
3.	<xɑserím, xserím, xɑserím>	<MAX(ɑ), DEP(V)>	ɑ-Deletion followed by ɑ-Epenthesis
4.	<xɑserím, xɑsrím>	<MAX(e)>	Only e-Deletion

The first member of the chain is faithful and gradualness is maintained throughout, so only local optimality is in question.
The first form (the input) violates *VCĕCV and *#Cɑσˈσ.
The second form doesn't violate either of those but does violate *#xC, this is still harmonic improvement though because *#Cɑσˈσ >> *#xC.
The third form is the same as the first form, so it violates *#Cɑσˈσ again, so local optimality isn't maintained. The desired chain isn't a valid chain.
If we tweak the definitions a bit and allow PREC constraints to be counted when determining valid chains will this help? No, We'll still have the fundamental problem that you can't go from the second form in the the chain back to the first. The second form in the chain violates MAX(ɑ) and doesn't violate DEP(V) so it won't violate PREC(MAX(ɑ), DEP(V)). So even if we could add that PREC constraint into consideration, it simply would not rule out candidate 2, no matter how high it's placed.

Putting the potential candidates in a tableau will show that, similarly to /ʝaveʃ-ím/, candidate 4 where only e is deleted, is optimal:

		xɑserím	*#Cɑσˈσ	*VCĕCV	*#xC	MAX(e)	MAX(ɑ)	DEP(V)
a.		xɑserím, <>	*!	*				
b.		xserím,<MAX(ɑ)>			*!		*	
c.		xɑserím,<MAX(ɑ),DEP(V)>	*!	*			*	*
d.	☞	xɑsrím,<MAX(e)>				*		

Even if we ignore the fact that candidate 4 is the optimal candidate, we can see that candidate 2 still beats out the preferred candidate (candidate 3).

Conclusion

OT-CC could not produce the bleeding Duke of York derivation found in Modern Hebrew. However I think the problem is more general than this. It seems that OT-CC cannot produce alternations that are in any way circular. Its requirement of local optimality in a candidate means that two forms in a chain can't be identical, in this particular case the input and output can't be identical. This means it wouldn't be able to produce Duke of York derivations (vacuous or not) in any language.

In other words, OT-CC under-generates with respect to Modern Hebrew. In fact it seems like OT-CC would under-generate with respect to any language with a Duke of York alternation.

Bibliography

McCarthy, J. J. (2003). Sympathy, Cumulativity, and the Duke-of-York Gambit. In *The Syllable In Optimality Theory, Fery, Caroline, & van de Vijver, Ruben (eds), New York: Cambridge U Press, 2003, pp 23-76* (pp. 23–76). Cambridge University Press.

McCarthy, J. J. (2007). *Hidden Generalizations : Phonological Opacity in Optimality Theory*. Equinox.

Pullum, G. K. (1976). The Duke of York gambit. *Journal of linguistics*, *12*(1), 83–102.

Rasin, E. (2022). Challenges for the size-based parallel analysis of Modern Hebrew vowel deletion. *Radical: A Journal of Phonology*, *4*, 9–43.