

Opazität
Turbid Output Representations and the Unity of
Opacity
(Matthew Goldrick 2000)

Anke Assmann
anke.assmann@uni-leipzig.de

Universität Leipzig, Institut für Linguistik

15.05.2012

Was ist das?



(Quelle: <http://www.nationalgeographic.de/reportagen/fotostrecke-alles-aus-einer-hand>)

Inhalt

- 1 Idee der trüben Outputrepräsentationen
- 2 Fallstudien
 - Fallstudie 1: Vokallängung im Luganda
 - Fallstudie 2: Ortsassimilation dorsaler Frikative im Deutschen
- 3 Trübheit vs. Sympathie und OO-Treue
 - Trübheit vs. Sympathie (McCarthy 1999)
 - Trübheit vs. OO-Treue (Burzio 1994, Benua 1997)
- 4 Appendix: Das (trübe) verrückte Labyrinth

Inhalt

- 1 Idee der trüben Outputrepräsentationen
- 2 Fallstudien
 - Fallstudie 1: Vokallängung im Luganda
 - Fallstudie 2: Ortsassimilation dorsaler Frikative im Deutschen
- 3 Trübheit vs. Sympathie und OO-Treue
 - Trübheit vs. Sympathie (McCarthy 1999)
 - Trübheit vs. OO-Treue (Burzio 1994, Benua 1997)
- 4 Appendix: Das (trübe) verrückte Labyrinth

Opazität und OT

Problem

Bestimme Fälle von Opazität lassen sich nicht einfach in Optimalitätstheorie analysieren, da Treue-Constraints von den Outputrepräsentationen verletzt werden.

Beispiel für Opazität

	/tue/	/tio/	/tou/	/tei/
Palatalisierung: $t \rightarrow tʃ / __ [-\text{hint}]$	tue	tʃio	tou	tʃei
V1-Tilgung: $V \rightarrow \emptyset / __ V$	t e	tʃ o	t u	tʃ i
	[te]	[tʃo]	[tu]	[tʃi]

Counter-Bleeding: /tio/ \rightarrow [tʃo]

Man würde erwarten das Vokaltilgung des /i/ im Falle von /tio/ die Regel der Palatalisierung ausblutet. Palatalisierung findet aber statt.





Counter-Feeding: /tue/ \rightarrow [te]

Man würde erwarten das Vokaltilgung des /u/ im Falle von /tue/ die Regel der Palatalisierung füttert. Palatalisierung findet aber nicht statt.

Opazität in OT

Input: /tue/	*VV	*t[-hint]	Max	Ident
1a. [tue]				
1b. [te]				
1c. [tʃe]				
Input: /tio/	*VV	*t[-hint]	Max	Ident
2a. [tio]				
2b. [to]				
2c. [tʃo]				

Opazität in OT

Input:	/tue/	*VV	*t[-hint]	Max	Ident
a.	[tue]	*!			
 b.	[te]		*!		
 c.	[tʃe]			*	*
Input:	/tio/	*VV	*t[-hint]	Max	Ident
a.	[tio]	*!	*		
 b.	[to]			*	
 c.	[tʃo]			*	*!

Opazität doch in OT möglich

Lösung: Trübe Outputrepräsentationen

Outputrepräsentationen werden angereichert.

Konkret: Outputkandidaten sind Repräsentationen komplexer Derivationen. Auch Zwischenstufen der Derivation sind so Teil des Outputs.

Inhalt

- 1 Idee der trüben Outputrepräsentationen
- 2 Fallstudien
 - Fallstudie 1: Vokallängung im Luganda
 - Fallstudie 2: Ortsassimilation dorsaler Frikative im Deutschen
- 3 Trübheit vs. Sympathie und OO-Treue
 - Trübheit vs. Sympathie (McCarthy 1999)
 - Trübheit vs. OO-Treue (Burzio 1994, Benua 1997)
- 4 Appendix: Das (trübe) verrückte Labyrinth

Kompensatorische Längung nach gelöschten Vokalen

Daten:

/ka + tiko/	→	[katiko]	'mushroom'
/ka + oto/	→	[ko:to]	'fireplace (dim.)'
/ka + ezi/	→	[ke:zi]	'moon (dim.)'

Derivation (Clements 1986):

	/V ₁ V ₂ C/		
	μ_1	μ_2	
Project- μ	V ₁	V ₂	C
	μ_1	μ_2	
Hiatus Resolution	V ₁	V ₂	C
	μ_1	μ_2	
	↘		
Re-association	V ₁	V ₂	C
	[V ₂ :C]		

Counter-Bleeding

Man würde erwarten, dass die Vokallängung das Projizieren von μ_1 verhindert. Es gibt aber zwei Moren.


Analyse: Anreichern der Repräsentation

Derivation	Repräsentation	Symbolik
μ_1 μ_2 V_1 V_2 C	μ_1 μ_2 ↑ ↑ V_1 V_2 C	Y Y ↑ ↗ X X X projiziert Y
μ_1 μ_2 V_1 V_2 C	μ_1 μ_2 ↑ ↓ V_1 V_2 C	Y Y ↓ ↙ X X X realisiert Y
μ_1 μ_2 ↘ V_1 V_2 C	μ_1 μ_2 ↑ ↓ V_1 V_2 C	

Analyse: Definition von Constraints

Max	Alle Input segmente sollen korrespondierenden Outputsegmente haben.
V-Wt	Alle Vokale müssen ihre eigene Mora projizieren.
Pronounce-μ	Alle Moren müssen realisiert werden.
*VV	Zwei adjazent realisierte Vokale sind verboten.
Reciprocity$_{RT}^{\mu}$	Wenn ein Segment X eine Mora M projiziert, muss M X realisieren.
Pronounce-RT	Jedes Segment muss realisiert werden.

Analyse: Tableau

Input:	$/V_1V_2/$	*VV	Max	V-Wt	Pr- μ	R_{RT}^μ	Pr-RT
a.	μ_2 \updownarrow V_2		*!				
b.	V_1 μ_2 \updownarrow V_2			*!			*
c.	μ_1 μ_2 \uparrow \updownarrow V_1 V_2				*!	*	*
d.	μ_1 μ_2 \updownarrow \updownarrow V_1 V_2	*!					
e.	μ_1 μ_2 \uparrow \updownarrow V_1 V_2 					*	*

Längung vor pränasalisierten Plosiven

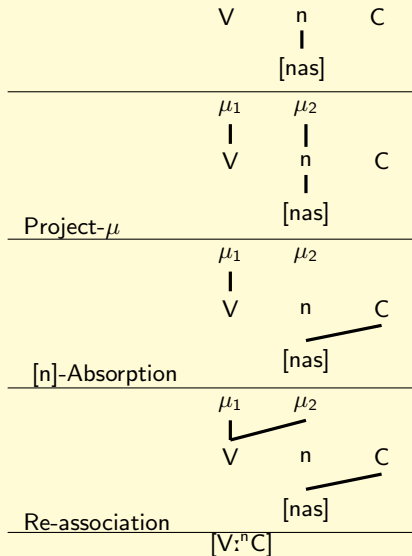
Daten:

/ku	+ linda/	→	[kuli: ⁿ da]	'to wait'
/mu	+ ntu/	→	[mu: ⁿ tu]	'person'
/ba	+ ntu/	→	[ba: ⁿ tu]	'people'

Counter-Bleeding

Man würde erwarten, dass die Absorption des Nasals das Projizieren einer More verhindert. Es gibt aber zwei Moren.

Derivation (Clements 1986)



Analyse: Definition Constraints

CodaCond	Außer der ersten Hälfte einer Geminate darf nichts in der Coda realisiert werden.
Max	Alle Input segmente sollen korrespondierende Outputsegmente haben.
Nas-Wt	Ein Segment, was von einem Merkmal [nas] projiziert wurde, muss seine eigene Mora projizieren.
Pronounce-μ	Alle Moren müssen realisiert werden.
Pronounce-F	Alle Merkmale müssen realisiert werden.
Reciprocity$_{RT}^{\mu}$	Wenn ein Segment X eine Mora M projiziert, muss X M realisieren.
Reciprocity$_F^{RT}$	Wenn ein Merkmal F eine Segment S projiziert, muss S F realisieren.
Pronounce-RT	Jedes Segment muss realisiert werden.

Input: /VnC/	CodaCond	Max	Nas-Wt	Pr- μ	Pr-F	R_{RT}^{μ}	R_F^{RT}	Pr-RT
a. $\begin{array}{c} V \\ \sigma \\ \downarrow \\ \mu \\ \downarrow \\ V \end{array} \quad \begin{array}{c} C \\ \sigma \\ \uparrow \\ \downarrow \\ C \end{array}$		*						
b. $\begin{array}{c} \sigma \\ \downarrow \\ \mu \\ \downarrow \\ V \end{array} \quad \begin{array}{c} \mu \\ \uparrow \\ n \\ \uparrow \\ [nas] \end{array} \quad \begin{array}{c} \sigma \\ \uparrow \\ \downarrow \\ C \end{array}$			*		*		*	*
c. $\begin{array}{c} \sigma \\ \downarrow \\ \mu \\ \downarrow \\ V \end{array} \quad \begin{array}{c} \mu \\ \uparrow \\ n \\ \uparrow \\ [nas] \end{array} \quad \begin{array}{c} \sigma \\ \uparrow \\ \downarrow \\ C \end{array}$				*	*	*	*	*
d. $\begin{array}{c} \sigma \\ \downarrow \\ \mu \\ \downarrow \\ V \end{array} \quad \begin{array}{c} \mu \\ \uparrow \\ n \\ \uparrow \\ [nas] \end{array} \quad \begin{array}{c} \sigma \\ \uparrow \\ \downarrow \\ C \end{array}$	*							
e. $\begin{array}{c} \sigma \\ \downarrow \\ \mu \\ \downarrow \\ V \end{array} \quad \begin{array}{c} \mu \\ \uparrow \\ n \\ \uparrow \\ [nas] \end{array} \quad \begin{array}{c} \sigma \\ \uparrow \\ \downarrow \\ C \end{array}$						*	*	*

Längung nach Konsonant-Gleitlaut-Clustern

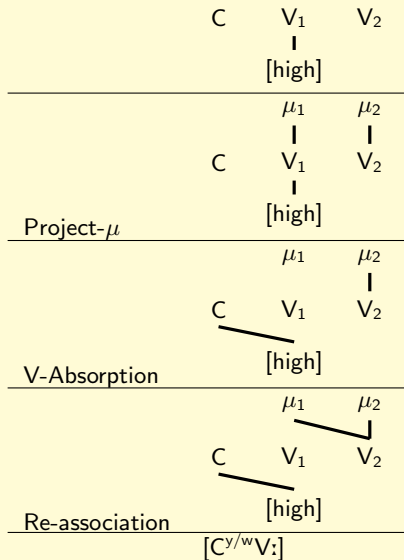
Daten:

/ki	+ buga/	→	[kibuga]	'town'
/ki	+ uma/	→	[k ^y u:ma]	'metal object'
/mu	+ kazi/	→	[mukazi]	'woman'
/mu	+ oyo/	→	[m ^w o:yo]	'soul'

Counter-Bleeding

Man würde erwarten, dass die Änderung des Vokals zum Gleitlaut das Projizieren von einer More verhindert. Es gibt aber zwei Moren.

Derivation (Clements 1986)



Analyse: Definition Constraints

Übung siehe Tafel

Analyse: Definition Tableau

Übung siehe Tafel

Vorlesen

	Deutsch	HMM-Deutsch
Buch	[bu:x]	[bu:x]
Pech	[pɛç]	[pɛç]
Masochist	[ma.so.xist]	[ma.so.çist]
Charisma	[ka.ris.ma]	[ça.ris.ma]

Beobachtung

Goldrick zitiert Hall (1989), Moltmann (1990) und Merchant (1995, 1996) für die deutschen Daten. Die darin beschriebenen Aussprachevarianten (HMM-Deutsch) sind scheinbar für einige Sprecher möglich, allerdings existiert eine Aussprachealternative (Deutsch).

HMM-Deutsch vs. Deutsch: Daten

	HMM-Deutsch		Deutsch	
kein Suffix	bu[x] na[x]	pe[ç] ge.sprä[ç]	bu[x] na[x]	pe[ç] ge.sprä[ç]
Level-1-Suffix	ma.so[x] a.nar[ç]	ma.so.[ç]-ist a.nar.[ç]-ist	ma.so[x] a.nar.[ç]	ma.so.[x]-ist a.nar.[ç]-ist
Level-2-Suffix	rau[x] frau	rau.[x]-en frau.-[ç]en	rau.[x] frau	rau.[x]-en frau.-[ç]en

Beobachtung

HMM-Deutsch hat einen Unterschied zwischen Level-1- und Level-2-Suffixen: Ein Level-1-Suffix kann Assimilation von [ç] im Stamm an einen vorangehenden hinteren Vokal verhindern, ein Level-2-Suffix nicht.

Deutsch unterscheidet diesbezüglich nicht zwischen Level-1- und Level-2-Suffixen. Hier entscheidet die Morphemgrenze.

HMM-Deutsch vs. Deutsch: Opazität

HMM-Deutsch

Man erwartet, dass Assimilation nicht über Silbengrenzen hinweg appliziert.

rau.[x]en ist unerwartet.

Es gibt Überapplikation der Assimilation.

Counter-Bleeding

Wir würden erwarten, dass die Silbifizierung die Assimilation ausblutet.

Deutsch

Man erwartet, dass Assimilation über Silbengrenzen aber nicht über Morphemgrenzen hinweg appliziert.

Alle Daten sind erwartet.

Es gibt nur normale Applikation.

keine Opazität

Da Morphemgrenzen die Assimilation blockieren, gibt es keine Opazität.

HMM-Deutsch: Analyse

	Level I Affix [ç] im Stamm	Level II Affix [ç] im Stamm	Level II Affix [ç] im Affix
INPUT	maso[ç]	rau[ç]	frau
L1-Affigierung	maso[ç]+ist	—	—
Silbifizierung	ma.so.[ç]ist	rau[ç]	frau
Assimilation	—	rau[x]	—
L2-Affigierung	—	rau[x]+en	frau+[ç]en
Resilbifizierung	—	rau.[x]en	frau.[ç]en
Assimilation	—	—	—
OUTPUT	ma.so.[ç]ist	rau.[x]en	frau.[ç]en
	Normale Appl.	Überapplikation Counter-Bleeding	Normale Appl.

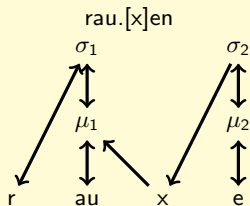
Deutsch: Analyse

	Level I Affix [ç] im Stamm	Level II Affix [ç] im Stamm	Level II Affix [ç] im Affix
INPUT	maso[ç]	rau[ç]	frau
Affigierung	maso[ç]+ist	rau[ç]+en	frau+[ç]en
Assimilation	maso[x]+ist	rau[x]+en	—
Silbifizierung	ma.so.[x]ist	rau.[x]en	frau.[ç]en
OUTPUT	ma.so.[x]ist	rau.[x]en	frau.[ç]en
	Normale Appl.	Normale Appl.	Normale Appl.

HMM-Deutsch in trüber OT: Idee

Idee

Frikative werden als Coda projiziert, aber als Onset realisiert.



HMM-Deutsch in trüber OT: Constraints

Onset	Alle Silben müssen ein Segment vor ihrem Nukleus realisieren.
Align-II (Morph, σ)	Segmentale Ränder von Level-II-Morphemen müssen mit den Silbenrändern desselben Morphems hinsichtlich der Projektion übereinstimmen.
Assimilate(back)	Ein Segment, das als Coda einer Silbe σ_i projiziert, muss sich das Merkmal [hinten] mit dem Vokal teilen, der in σ_i projiziert.
Reciprocity^{μ}_{RT}	Wenn ein Segment X eine Mora M projiziert, muss M X realisieren.
Faith(back)	Segmente sollten von den Merkmalen projiziert werden, mit denen sie zugrundeliegend assoziiert waren.

HMM-Deutsch in trüber OT: Tableau

Input: /rau[ç]+en/	Onset	Align-II	Assim(back)	R_{RT}^{μ}	Faith(back)
a. $\begin{array}{cc} \sigma_1 & \sigma_2 \\ \updownarrow & \updownarrow \\ \mu_1 & \mu_2 \\ \swarrow & \searrow \\ r & \text{au} \quad \text{ç} \\ \downarrow & \downarrow \\ & e \end{array}$		*!			
b. $\begin{array}{cc} \sigma_1 & \sigma_2 \\ \updownarrow & \downarrow \\ \mu_1 & \mu_2 \\ \swarrow & \searrow \\ r & \text{au} \quad x \\ \downarrow & \downarrow \\ & e \end{array}$	*!				*
c. $\begin{array}{cc} \sigma_1 & \sigma_2 \\ \updownarrow & \updownarrow \\ \mu_1 & \mu_2 \\ \swarrow & \searrow \\ r & \text{au} \quad \text{ç} \\ \downarrow & \downarrow \\ & e \end{array}$			*!	*	
d. $\begin{array}{cc} \sigma_1 & \sigma_2 \\ \updownarrow & \updownarrow \\ \mu_1 & \mu_2 \\ \swarrow & \searrow \\ r & \text{au} \quad x \\ \downarrow & \downarrow \\ & e \end{array}$				*	*

Umlaute und Assimilation: Daten

Stamm	Umlaut+Affix
ba[x]	bä[ç].+lein
bru[x]	brü.[ç]+ig

HMM-Deutsch: Counter-Feeding

Umlaut tritt nur zusammen mit dem Affix auf. Da es sich um Level-II-Affixe handelt, werden sie nach Ortsassimilation und Silbifizierung affigiert. Damit sollte man ein ähnliches Muster wie bei "rau.[x]en" erwarten.

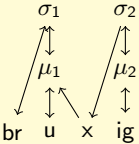
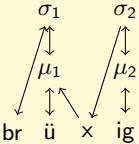
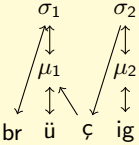
HMM-Deutsch: Umlaute und Assimilation: Derivation

INPUT	Level II Affix [ç] im Stamm ba[ç]	Level II Affix [ç] im Stamm bru[ç]
L1-Affigierung	—	—
Silbifizierung	ba[ç]	bru[ç]
Assimilation	ba[x]	bru[x]
L2-Affigierung	bä[x]+lein	brü[x]+ig
Resilbifizierung	bä[x].lein	brü.[x]ig
Assimilation	bä[ç].lein	—
OUTPUT	bä[ç].lein Normale Appl.	*brü.[x]ig Unterapplikation Counter-Feeding

Deutsch: Derivation

INPUT	Level II Affix [ç] im Stamm ba[ç]	Level II Affix [ç] im Stamm bru[ç]
Affigierung	bä[ç]+lein	brü[ç]+ig
Assimilation	—	—
Silbifizierung	bä[ç].lein	brü.[ç]ig
OUTPUT	bä[ç].lein Normale Appl.	brü.[ç]ig Normale Appl.

HMM-Deutsch: OT-Analyse

Input: /bru[ç]+ig/	Onset	Umlaut	Align-II	Assim(back)	R_{RT}^{μ}	Faith(back)
a. 		*!			*	*
b. 				*!	*	**
c. 					*	*

Inhalt

1 Idee der trüben Outputrepräsentationen

2 Fallstudien

- Fallstudie 1: Vokallängung im Luganda
- Fallstudie 2: Ortsassimilation dorsaler Frikative im Deutschen

3 Trübheit vs. Sympathie und OO-Treue

- Trübheit vs. Sympathie (McCarthy 1999)
- Trübheit vs. OO-Treue (Burzio 1994, Benua 1997)

4 Appendix: Das (trübe) verrückte Labyrinth



Beispiel für Opazität

		/tio/
Palatalisierung:	$t \rightarrow tʃ / _ [-\text{hint}]$	tʃio
V1-Tilgung:	$V \rightarrow \emptyset / _ V$	tʃ o
		[tʃo]

Counter-Bleeding: /tio/ \rightarrow [tʃo]

Man würde erwarten das Vokaltilgung des /i/ im Falle von /tio/ die Regel der Palatalisierung ausblutet. Palatalisierung findet aber statt.

Opazität in OT

Input:	/tio/	*VV	*t[-hint]	Max	Ident
a.	[tio]	*!	*		
b.	[tʃio]	*!			*
 c.	[to]			*	
 d.	[tʃo]			*	*!

Opazität und Sympathie

Grundidee

Die Zwischenrepräsentation ist Teil der Kandidatenmenge und wird mithilfe eines Auswahl-Constraints (☆) als besonderer Kandidat (⊛) ausgewählt.

Durch ein spezielles Treueconstraint (⊛) wird verlangt, dass der optimale Output dem ⊛-Kandidaten (Zwischenrepräsentation) in bestimmten Merkmalen gleicht.

Analyse Ausgangspunkt: Beschränkungsprofile

Input:	/tio/	*VV	*t[-hint]	Max	Ident
a.	[tio]	*!	*		
b.	[tʃio]	*!			*
c.	[to]			*	
d.	[tʃo]			*	*!

Analyse 1. Schritt: Auswahl Selektorconstraint ☆

Input:	/tio/	*VV	*t[-hint]	(☆) Max	Ident
a.	[tio]	*!	*	✓	
b.	[tʃio]	*!		✓	*
c.	[to]			*	
d.	[tʃo]			*	*!

Analyse 2. Schritt: Bestimmung des Blümchenkandidats (✿) durch Vergleich der Beschränkungsprofile

Input:	/tio/	*VV	*t[-hint]	(☆) Max	Ident
a.	[tio]	*!	*	✓	
✿ b.	[tʃio]	*!		✓	*
c.	[to]			*	
d.	[tʃo]			*	*!

Analyse 3. Schritt: Treue-Constraint (✿) einfügen

Input: /tio/	*VV	*t[-hint]	(✿) Ident-tʃ	(☆) Max	Ident
a. [tio]	*!	*		✓	
✿ b. [tʃio]	*!			✓	*
c. [to]				*	
d. [tʃo]				*	*!

Analyse 4. Schritt: Optimalen Output bestimmen

Input: /tio/	*VV	*t[-hint]	(☼) Ident-tʃ	(☆) Max	Ident
a. [tio]	*!	*	*	✓	
☼ b. [tʃio]	*!			✓	*
c. [to]			*!	*	
☞ d. [tʃo]				*	*

Möglicher Nachteil der Sympathietheorie

Behauptung:

Für die Analyse von opaken Daten braucht es für jeden Fall ein besonderes ☆-Constraint und ✿-Constraint, was den optimalen Output in Hinblick auf den ✿-Kandidaten bestimmt.

Trübheit kann verschiedene Daten mit den gleichen Rankings ableiten.

Luganda: Vokaltilgung vs. Pränasalisierung

	Pränasalisierung	Vokaltilgung
Trübheit	$\begin{array}{ccc} \mu_1 & & \mu_2 \\ \downarrow & \swarrow & \uparrow \\ V & & n & C \end{array}$	$\begin{array}{ccc} \mu_1 & & \mu_2 \\ \downarrow & \swarrow & \uparrow \\ V & & V & C \end{array}$
	Pronounce- $\mu \gg R_{Rt}^\mu$	Pronounce- $\mu \gg R_{Rt}^\mu$
Sympathie	$\begin{array}{ccc} \mu_1 & & \mu_2 \\ & \swarrow & \\ V & & n & C \end{array}$	$\begin{array}{ccc} \mu_1 & & \mu_2 \\ & \swarrow & \\ V & & V & C \end{array}$
	✿Faith(☆Uni) $\gg * \mu$	✿Faith(☆Max) $\gg * \mu$

Luganda: Vokaltilgung vs. Pränasalisierung

Angeblicher Nachteil von Sympathie

Sympathie braucht zwei verschiedene “Mechanismen”, um die Daten abzuleiten, d.h. sie braucht zwei verschiedene Constraints (Uniformity und Max).

Trübheit braucht aber die gleichen Constraints, um die Daten abzuleiten.

Tatsache

Der eigentliche Mechanismus ist auch bei Sympathie der gleiche: Das Treueconstraint, das im jeweiligen Fall den stattfindenden Prozess blockiert (Tilgung: Max; Pränasalisierung: Uniformity), wählt den * ϕ -Kandidaten, zu dem Treue gehalten werden muss. Die Unterschiede kommen nur zustande, da es sich um verschiedene Prozesse handelt.

Auch Trübheit braucht Constraints, die die Prozesse unterscheiden:

Vowel-Weight vs. **Nasal-Weight**, ***VV** vs. **CodaCond**.

Beispiel für Opazität

		/tio/
Palatalisierung:	$t \rightarrow tʃ / _ [-\text{hint}]$	tʃio
V1-Tilgung:	$V \rightarrow \emptyset / _ V$	tʃ o
		[tʃo]
<hr/>		
		/tio/
Infigierung:	-t-	ti-t-o
Palatalisierung:	$t \rightarrow tʃ / _ [-\text{hint}]$	tʃito
V1-Tilgung:	$V \rightarrow \emptyset / _ V$	—
		[tʃito]

Opazität und OO-Treue

Grundidee


Es gibt einen anderen Wettbewerb mit gleichem Input, bei dem der optimale Output transparent ist.

Durch ein Output-Output-Treueconstraint wird verlangt, dass der optimale Output des eigentlichen Wettbewerbs, dem transparenten Outputkandidaten eines anderen Wettbewerbs gleicht.

Analyse Ausgangspunkt: Beschränkungsprofile

Input:	/tio/	*VV	*t[-hint]	Max	Ident
a.	[tio]	*!	*		
b.	[tʃio]	*!			*
c.	[to]			*	
d.	[tʃo]			*	*!



Analyse 1. Schritt: Anderen Wettbewerb einsehen

Input:	/tio/	*VV	*t[-hint]	Max	Ident
a.	[tio]	*!	*		
b.	[tʃio]	*!			*
c.	[to]			*	
d.	[tʃo]			*	*!
Input:	/tio/ + -t-	*VV	*t[-hint]	Max	Ident
a.	[tito]		*!		
 b.	[tʃito]				*
c.	[tto]			*!	
d.	[tʃto]			*!	

Analyse 2. Schritt: OO-Constraint involvieren

Input:	/tio/	*VV	*t[-hint]	OO-Treue	Max	Ident
a.	[tio]	*!	*			
b.	[tʃio]	*!				*
c.	[to]				*	
d.	[tʃo]				*	*!
Input:	/tio/ + -t-	*VV	*t[-hint]		Max	Ident
a.	[tito]		*!			
☞ b.	[tʃito]					*
c.	[tto]				*!	
d.	[tʃto]				*!	

Analyse 3. Schritt: Optimalen Output bestimmen

Input: /tio/	*VV	*t[-hint]	OO-Treue	Max	Ident
a. [tio]	*!	*	*		
b. [tʃio]	*!				*
c. [to]			*!	*	
 d. [tʃo]				*	*
Input: /tio/ + -t-	*VV	*t[-hint]		Max	Ident
a. [tito]		*!			
 b. [tʃito]					*
c. [tto]				*!	
d. [tʃto]				*!	

Möglicher Nachteil von OO-Treue

Die Daten des HMM-Deutsch können nicht ohne weiteres abgeleitet werden:

Underivierte Form	→	Derivierte Form
rau[x]		rau.[x]ig
bru[x]		*brü.[x]ig

Um die Assimilationsdaten ableiten zu können, muss OO-Treue sich auf die underivierte Form beziehen. Dann können aber die Umlautdaten nicht ohne zusätzliche Constraints abgeleitet werden.

Konklusion

- Trübheit erklärt Opazität, indem die Repräsentation der Derivation Bestandteil des Outputs ist.
- Damit kann sie die hier vorgestellten Daten elegant erklären.
- Sympathie und OO-Treue benötigen evtl. zusätzliche Mittel, um die gleichen Daten ableiten zu können.
- Damit scheint Trübheit aus Perspektive dieser bestimmten Daten die elegantere Theorie zu sein.

Inhalt

- 1 Idee der trüben Outputrepräsentationen
- 2 Fallstudien
 - Fallstudie 1: Vokallängung im Luganda
 - Fallstudie 2: Ortsassimilation dorsaler Frikative im Deutschen
- 3 Trübheit vs. Sympathie und OO-Treue
 - Trübheit vs. Sympathie (McCarthy 1999)
 - Trübheit vs. OO-Treue (Burzio 1994, Benua 1997)
- 4 Appendix: Das (trübe) verrückte Labyrinth

Das verrückte Labyrinth

(http://kids.t-online.de/onlinespiel-das-verrueckte-labyrinth-von-ravensburger/id_44396420/index)

Pro Spieler gibt es drei Aktionen:

- 1 Verschiebe Wegfelder auf einem Brett (so dass sich möglicherweise Pfade innerhalb des Labyrinths ändern) – **obligatorisch**.
- 2 Bewege dich innerhalb der Pfade – **optional**.
- 3 Sammle Gegenstände ein (indem du dich auf das Feld des Gegenstandes stellst) – **optional**.

Vor dem Schieben



Nach dem Schieben



Das verrückte Labyrinth: Opazität

Situation 1:

Input:

Spieler A möchte von Feld 1 auf Feld 2. Es gibt keinen durchgehenden Pfad.

Derivation:

Spieler A verschiebt die Wegfelder so, dass sich ein Weg zu Feld 2 ergibt.

Spieler A bewegt sich auf Feld 2.

Spieler B verschiebt die Wegfelder so, dass die Verbindung zwischen Feld 1 und Feld 2 getrennt wird.

Spieler B bewegt sich (nicht).

Output (nach einer Runde):

Die fehlende Verbindung zwischen Feld 1 und Feld 2 sollte die Bewegung von Spieler A unmöglich machen. Trotzdem hat sich Spieler A bewegt.

Opazität:

Counter-Bleeding

Das verrückte Labyrinth: Opazität

Situation 2:

Input:

Spieler A möchte von Feld 1 auf Feld 2. Es gibt einen durchgehenden Pfad.

Derivation:

Spieler A muss die Wegfelder so verschieben, dass die Verbindung zwischen Feld 1 und Feld 2 getrennt wird. (Angenommen, jede Schiebemöglichkeit trennt die Verbindung.)

Spieler A kann sich nicht auf Feld 2 bewegen.

Spieler B verschiebt die Wegfelder so, dass sich wieder ein Weg zwischen Feld 1 und Feld 2 ergibt.

Spieler B bewegt sich (nicht).



Output (nach einer Runde):

Die vorhandene Verbindung zwischen Feld 1 und Feld 2 sollte die Bewegung von Spieler A möglich machen. Der Spieler hat sich aber nicht bewegt.

Opazität:

Counter-Feeding

Das verrückte Labyrinth in OT

Input: /F1(S) \neq F2/	MovePath	*F1(S)	NoMove	NoChange
a. [F1(S) \neq F2]		*!		
b. [F1(S) = F2]		*!		*
 c. [F1 = F2(S)]			*	*
 d. [F1 \neq F2(S)]	*!		*	

Das verrückte Labyrinth in trüber OT

Input: $/F1(S) \neq F2/$	MovePath	*F1(S)	NoMove	NoChange	R_F^{Path}
a. $\begin{array}{cc} = & \neq \\ & \swarrow \uparrow \\ F1(S) & F2 \end{array}$		*!			
b. $\begin{array}{cc} = & \neq \\ & \swarrow \uparrow \\ F1 & F2(S) \end{array}$	*!		*		
c. $\begin{array}{cc} = & \neq \\ \uparrow \downarrow & \swarrow \uparrow \\ F1 & F2(S) \end{array}$			*	*!	
d. $\begin{array}{cc} = & \neq \\ \uparrow \downarrow & \swarrow \uparrow \\ F1 & F2(S) \end{array}$			*		*

Literatur

- Benua, L. (1997): *Transderivational identity: Phonological relations between words*. Doctoral dissertation, University of Massachusetts, Amherst.
- Burzio, L. (1994): *Principles of English stress*. Cambridge: Cambridge University Press.
- Clements, N. (1986): Compensatory lengthening and consonant gemination in Luganda. In: Wetzels, L. & Sezer, E. (eds.) *Studies in compensatory lengthening*, 37-77. Dordrecht: Foris.
- Goldrick, M. (2000): Turbid Output Representations and the Unity of Opacity. In: *Proceedings of NELS 30*.
- Hall, T. A. (1989): Lexical phonology and the distribution of German [ç] and [x]. *Phonology* 6, 1-17.
- McCarthy, J. (1999): Sympathy and Phonological Opacity, *Phonology* 16, 331-399.
- Merchant, J. (1995): Deriving cyclic syllabification effects: Fricative assimilation and final devoicing in German. Ms., University of California, Santa Cruz.
- Merchant, J. (1996): Alignment and fricative assimilation in German. *Linguistic Inquiry* 27, 709-719.
- Moltmann, F. (1990): Syllabification and lexical phonology in German. Ms., Massachusetts Institute of Technology.