

Optimalitätstheorie

- Was ist die Optimalitätstheorie?
- Universalien

OT-Grammatik als Input-Output Vorrichtung (Funktion)

Eine Grammatik kann als Funktion betrachtet werden, die einem Input eindeutig einen Output zuordnet: $G(\text{Input}) = \text{Output}$.

Unter der Annahme, dass die zugrunde liegende Form vom Weissrussischen „*kalá*“ /kola/ lautet (bzw. „*shept*“ /ʃept/), erhalten wir

- ◆ $G (/kola/) = /ka'la/$
- ◆ $G (/ʃept/) = /ʃept/$

Dies gilt gleichermaßen für die generative Phonologie wie für die Optimalitätstheorie.

Diese unterscheiden sich in der Art und Weise, wie die Funktion "berechnet" wird. In der klassischen Generativen Phonologie wird der Output durch Anwendung einer geordneten Menge von Regeln aus dem Input abgeleitet (Derivation).

Ableitung in der Generativen Phonologie

Input	/kɔla/	/ʃept/
Betonung	/kɔ ^l la/	/ ^l ʃept/
Vokalreduktion	/ka ^l la/	—
Output	/ka ^l la/	/ʃept/

OT-Grammatik als Input-Output Vorrichtung (Funktion)

Auch OT-Grammatik ist Input-Output-Funktion, die einem Input eindeutig einen Output zuordnet: $G_{OT}(\text{Input}) = \text{Output}$.

Diese Abbildung verteilt sich in der OT jedoch auf verschiedene Komponenten:

Lexikon (bzw. Input):

Das Lexikon enthält die lexikalischen Repräsentationen (zugrunde liegenden Formen) der Morpheme und liefert den Input für den Generator.

Die phonologische Form dieser Morpheme ist sprachspezifisch.

„...well-formed objects in the sense that the input does not contain non-linguistic objects ...“

Komponenten einer OT-Grammatik

Generator:

Der Generator erzeugt aus einem Input eine potentiell unbegrenzte Menge von Output-Kandidaten : $\text{Gen}(\text{Input}) \Rightarrow \{K_1, K_2, K_3, \dots, K_n\}$ und übergibt sie an den Evaluator.

Evaluator:

Der Evaluator besteht aus einer Menge von geordneten Beschränkungen $\{B_1 \gg B_2 \gg \dots B_n\}$ und bewertet die Output-Kandidaten hinsichtlich ihrer "Harmonier-Werte" und wählt danach den optimalen Kandidaten aus. Diese Auswahl ist eindeutig, d.h. es gibt für eine Kandidaten-Menge jeweils nur einen optimalen Kandidaten als Output: $\text{Eval}(\{K_1, K_2, K_3, \dots, K_n\}) \Rightarrow \text{Output}$

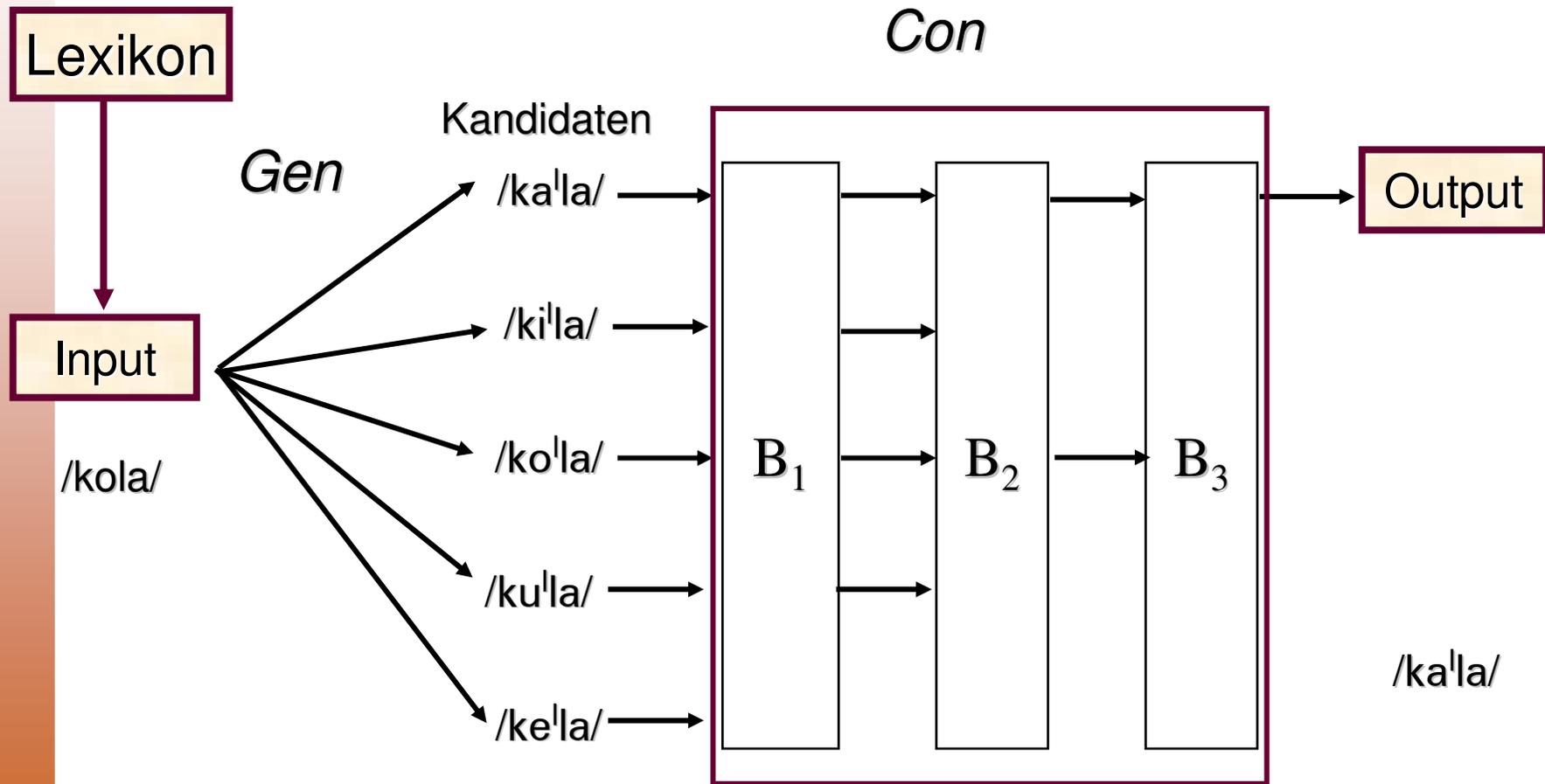
Komponenten einer OT-Grammatik

Output:

Wenn zwei Kandidaten mehrere Beschränkungen nicht verletzen, wird es auf niedriger Ebene eine Beschränkung geben, die genau einen Kandidaten übrig lässt.

Sind zwei Kandidaten gar nicht von einander zu unterscheiden, sind sie identisch.

Architektur der Optimalitätstheorie



OT-Darstellungsform (Tableaus)

/Y/	A	B	C
x	*!		*
y		*!	
 z			*

- in der oberen linken Ecke - UR;
- von Gen produzierten Kandidaten (x, y, z) - einer pro Spalte;
- links-nach-rechts Spalten geben die Dominanzreihenfolge der Beschränkungen (A, B, C) an;
- durchgezogene Linien bedeuten Hierarchisierung;
- gestrichelte Linien bedeuten Gleichrangigkeit;
- Erfüllung wird durch eine leere Zelle gezeigt;
- Asterix bedeutet Verletzung der Beschränkung;
- Ausrufezeichen ! Nicht-Optimalität, fatale Verletzung;
- graue Schattierung - irrelevant;
-  zeigt auf den optimalen Kandidaten.

Grundbegriffe der OT

- Beschränkungen (constraints)
- Konflikt
- "Herrschaft" (domination)
- Optimalität

Beschränkungen - constraints

- Eine **Beschränkung** ist eine **Strukturbedingung**, die von einer Output-Form entweder **erfüllt** (engl. *satisfy*) oder **verletzt** (engl. *violate*) werden kann.

- Es können drei **Typen** von Beschränkungen unterschieden werden:
 - ◆ **Treue-Beschränkungen** (*faithfulness constraints*)
 - ◆ **Markiertheits-Beschränkungen** (*markedness constraints*)
 - ◆ **Zuordnungs-beschränkungen** (*alignment constraints*)

Markiertheits-Beschränkungen

- **Markiertheits-Beschränkung** verlangen, dass Output-Formen ein bestimmtes strukturelles Wohlgeformtheits-Kriterium erfüllen. Diese können sowohl positiv als auch negativ formuliert sein, so daß man unterscheiden kann zwischen:
 - ◆ **Geboten** und
 - ◆ **Verboten**
- **Verbote**
 - ◆ Vokale dürfen nicht nasaliert sein
 - ◆ Silben dürfen keine Koda haben (NoCoda bzw. *Coda)
 - ◆ Obstruenten dürfen in der Kodaposition nicht stimmhaft sein
- **Gebote**
 - ◆ Sonoranten müssen stimmhaft sein
 - ◆ Silben müssen einen Anlaut haben (Onset)
 - ◆ Silben müssen einen Silbengipfel haben (Peak)

Treue-Beschränkungen

- Während Markiertheits-Beschränkungen sich nur auf Output-Formen beziehen und den Input nicht berücksichtigen, verlangen **Treue-Beschränkungen** (*faithfulness*), daß in den Output-Formen die Eigenschaften ihrer zugrundeliegenden (lexikalischen) Formen erhalten bleiben. Im Idealfall sollte der Output identisch mit dem Input sein. Beispiele:
 - ◆ Im Output müssen alle Segmente des Inputs erhalten bleiben (keine Tilgung, Max bzw. Parse)
 - ◆ Der Output muß die lineare Abfolge aller Segmente des Inputs beibehalten (keine Metathesen)
 - ◆ Output-Segmente müssen Entsprechungen im Input haben (Epenthese-Verbot; Dep bzw. Fill)
 - ◆ Output-Segmente und Input-Segmente müssen identische Merkmalswerte aufweisen (Ident-IO_{Merkmal} bzw. Preserve-IO_{Merkmal}).

Zuordnungs-Beschränkungen

- Zuordnungs-Beschränkungen stellen Verknüpfungen her:
 - ◆ z.B. A-R_{stem-σ} : Jeder Stamm endet am rechten Rand einer Silbe.
vgl. Liäson im Französischen: „on est au salon“ [õ.nɛ.to.salõ]

Universalität und Verletzbarkeit (violability)

- Beschränkungen sind universell
- Beschänkungen sind verletzbar

Optimalität: Herrschaft und Konflikt

- **Optimalität:**
ein Output ist optimal, wenn er die hierarchisch geordnete Menge der Beschränkungen am besten erfüllt, d.h. die am wenigsten gravierenden Verletzungen aufweist
- **Konflikt:**
Beschränkungen stehen in einem Konkurrenzverhältnis zueinander. Insbesondere besteht zwischen Strukturbeschränkungen (Markiertheits-Beschränkungen) und Treue-Beschränkungen ein grundlegender Konflikt.
- **"Herrschaft" (dominance):**
Die höherrangige von zwei konfligierenden Beschränkungen hat Vorrang vor (dominiert) die niederrangige Beschränkung.

Beschränkungs-Interaktion – constraint interaction am Bsp. des Wessrussischen

- Ziel: ***Keine mittleren Vokale in unbetonten Silben!***

- Markiertheits-Beschränkung:
 - ◆ * **Lic-Mid/Stress** :
Mittlere Vokale sind erlaubt nur, wenn sie betont sind.

- Dem steht folgende Treue-Beschränkungen gegenüber:
 - ◆ **Ident-IO[low] bzw. Preserve[low]:**
Die Spezifikation für das Merkmal [low] eines Input-Segments muß im korrespondierenden Output-Segment erhalten bleiben.

 - ◆ **Ident-IO[high] bzw. Preserve[high]:**
Die Spezifikation für das Merkmal [high] eines Input-Segments muß im korrespondierenden Output-Segment erhalten bleiben.

- Diese beiden Beschränkungen stehen im Konflikt zueinander.

Beschränkungs-Interaktion – constraint interaction

- Die zugrundeliegende lexikalische Form (Input) ist /kola/
- Der Generator erzeugt daraus die Kandidaten [kola], [kala], [kila], [kula], [kela].
- Rangordnung der Beschränkungen:
Im Weissrussischen ist das Merkmal [low] so modifiziert, dass unbetonte mittlere Vokale vermieden werden. Folglich wird das Weissrussische die Verletzung von Preserve[low] tolerieren. Umgekehrt Lic-Mid/Stress und Preserve[high] werden nie verletzt.
- *Lic-Mid/Stress >> Preserve[high] >> Preserve[low]*

Vokalreduktion - Weissrussisch

The Constraint Tableau

/kola/	Lic-Mid/Stress	Preserve[High]	Preserve[Low]
 [kalá]			*
[kulá]		!*	
[kilá]		!*	
[kolá]	!*		
[kelá]	!*		

Faktoriale Typologie - ein blödes Beispiel

Universale Beschränkungen:

CAT = "Lass die Katze nicht raus."
WINDOW = "Öffne das Fenster."
DOOR = "Öffne die Tür."

Mögliche Rangordnungen: Ergebnisse jeder Rangordnung:

1. CAT » WINDOW » DOOR Die Katze bleibt drinnen; Fenster und Tür sind zu.
2. CAT » DOOR » WINDOW Die Katze bleibt drinnen; Tür und Fenster sind zu.
3. WINDOW » CAT » DOOR Die Katze ist draussen; Fenster zu, Tür auf.
4. WINDOW » DOOR » CAT Die Katze ist draussen; Fenster und Tür sind auf.
5. DOOR » CAT » WINDOW Die Katze ist draussen; Tür auf, Fenster zu.
6. DOOR » WINDOW » CAT Die Katze ist draussen; Tür und Fenster sind auf.

Faktoriale Typologie - ein blödes Beispiel

*Grammatiken 1 and 2 generieren das gleiche Ergebnis;
auch Grammatiken 4 and 6.*

3! = 6 Also es gibt 6 mögliche Grammatiken.

Jedoch in diesem Fall korrespondieren sie zu nur 3 Output_Mustern. Die Rangordnung von DOOR hinsichtlich WINDOW ist irrelevant, da diese Beschränkungen nicht im Konflikt sind.

Faktoriale Typologie - Hausaufgabe

Erstellt jetzt eine faktoriale Typologie.

Es wird eine 5-Vokalreduktion-System angenommen.

3 Beschränkungen = 6 mögliche Rangordnungen.

Lic-Mid/Stress
Preserve[low]
Preserve[high]

Faktoriale Typologie - Hausaufgabe

Lic-Mid/Stress » Preserve[Low] » Preserve[High]

Lic-Mid/Stress » Preserve[High] » Preserve[Low] =

Belarusian

Preserve[Low] » Preserve[High] » Lic-Mid/Stress

Preserve[High] » Preserve[Low] » Lic-Mid/Stress

Preserve[Low] » Lic-Mid/Stress » Preserve[High]

Preserve[High] » Lic-Mid/Stress » Preserve[Low] =

Belarusian

The Constraint Tableau

/kola/	Lic-Mid/Stress	Preserve[High]	Preserve[Low]
[kalá]			*
[kulá]		*!	
[kilá]		*!	
[kolá]	*!		
[kelá]	*!		

Faktoriale Typologie - Hausaufgabe

/kola/	Lic-Mid/Stress	Preserve[Low]	Preserve[High]
☞ [kulá]			*
[kilá]			*
[kalá]		*!	
[kolá]	*!		
[kelá]	*!		

*(in real life, [kula]
is better than [kila]
in this case due to
Preserve[Round])*

1. Diese Grammatik sieht vor, dass unbetonte mittlere Vokale durch Anhebung eliminiert werden.
2. Ein unbetontes /e/ wird zu [i] reduziert ([kila]), Preserve[Front].
3. Dieses Reduktionsmuster wird im Luiseño benutzt: /e/ > [i], /o/ > [u].
4. Dasselbe Muster ergibt sich aus der Hierarchie: Preserve[Low] » Lic-Mid/Stress » Preserve[High].

Faktoriale Typologie - Hausaufgabe

/kola/	Preserve[Low]	Preserve[High]	Lic-Mid/Stress
☞ [kolá]			*
[kelá]			*
[kulá]		*!	
[kilá]		*!	
[kalá]	*!		

1. Diese Grammatik lässt *keine* Reduktion von unbetonten mittleren Vokalen zu.
2. Dieses Reduktionsmuster beobachtet man in vielen Sprachen, z.B. Spanish, Polish.
3. Dasselbe Muster ergibt sich aus der Hierarchie:
Preserve[Low] » Preserve[High] » Lic-Mid/Stress.

Predicting Possible Language in a Process-Based Approach

- $[-\text{high}, -\text{low}] > [+high] / [C^0 \text{ __ } C^0]_{\sigma}$ = Reduces *stressed* vowels.
- $[-\text{high}, -\text{low}] > [+high, +\text{round}] / [C^0 \text{ __ } C^0]_{\sigma}$ = Reduces unstressed mid vowels to [u].
- $[-\text{high}, -\text{low}, \alpha\text{front}] > [+high, -\alpha\text{front}] / [C^0 \text{ __ } C^0]_{\sigma}$
= Reduces unstressed /e/ to [u] and
unstressed /o/ to [i].

Beschränkungs-Interaktion – constraint interaction

- Beispiel: Auslautverhärtung im Deutschen
- Obstruenten in Kodaposition sind stimmlos: /hʊnt/ 'Hund' vs. /hʊndə/ 'Hunde'.
- Die zugrundeliegende lexikalische Form ist /hʊnd/
- Zur Erklärung gibt es folgende Markiertheits-Beschränkung:
 - ▶ ***Voiced-Coda:**
Obstruenten in Kodaposition dürfen nicht stimmhaft sein
- Dem steht folgende Treue-Beschränkung gegenüber
 - ▶ Ident-IO(sth):
Die Spezifikation für das Merkmal [stimmhaft] eines Input-Segments muß im korrespondierenden Output-Segment erhalten bleiben
- Diese beiden Beschränkungen stehen im Konflikt zueinander.

Beschränkungs-Interaktion – constraint interaction

- Die zugrundeliegende lexikalische Form (Input) ist /hund/
- Der Generator erzeugt daraus die Kandidaten [hund] und [hʊnt] (neben vielen anderen wie [hʊn], [hʊnəd], [hʊnət], [hund]). Wir wollen uns auf die ersten beiden beschränken:
 - ▶ [hund] erfüllt Ident-IO(sth), aber verletzt *Voiced-Coda
 - ▶ [hʊnt] verletzt Ident-IO(sth), aber erfüllt *Voiced-Coda
- Wir erhalten die optimale Form [hʊnt], wenn wir von folgender Rangordnung der Beschränkungen ausgehen:
 - ▶ *Voiced-Coda >> Ident-IO(sth)
- Für das Englische würde umgekehrt gelten:
 - ▶ Ident-IO(sth) >> *Voiced-Coda

Auslautverhärtung – Deutsch

Kandidaten		*Voiced-Coda	Ident-IO(sth)
a.	[hund]	* !	
b.	[hʊnt]		*

Beispiel Englisch

Kandidaten		Ident-IO(sth)	*Voiced-Coda
a.	[haʊnd]		*
b.	*[haʊnt]	* !	

