

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 von engl. *sing* /sing/ lautet (analog zu *sink* /sink/), erhalten wir

- ▶ $G(/sink/) = /sɪŋk/$
- ▶ $G(/sing/) = /sɪŋ/$

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	/sink/	/sing/
Nasalassimilation	/siŋk/	/siŋg/
"Nasalharmonie"		/siŋ:/
Degemination		/siŋ/
Output	/siŋk/	/siŋ/

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.

Zunächst gibt es eine Funktion namens Generator (Gen), die den Input auf eine potentiell unbegrenzte Menge von Output-Kandidaten abbildet: $\text{Gen}(\text{Input}) \Rightarrow \{K_1, K_2, K_3, \dots, K_n\}$

Eine zweite Funktion, der Evaluator (Eval) erhält diese Kandidaten-Menge als Input und wählt daraus 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

Eine OT-Grammatik besteht aus folgenden Komponenten:

Lexikon:

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

Generator:

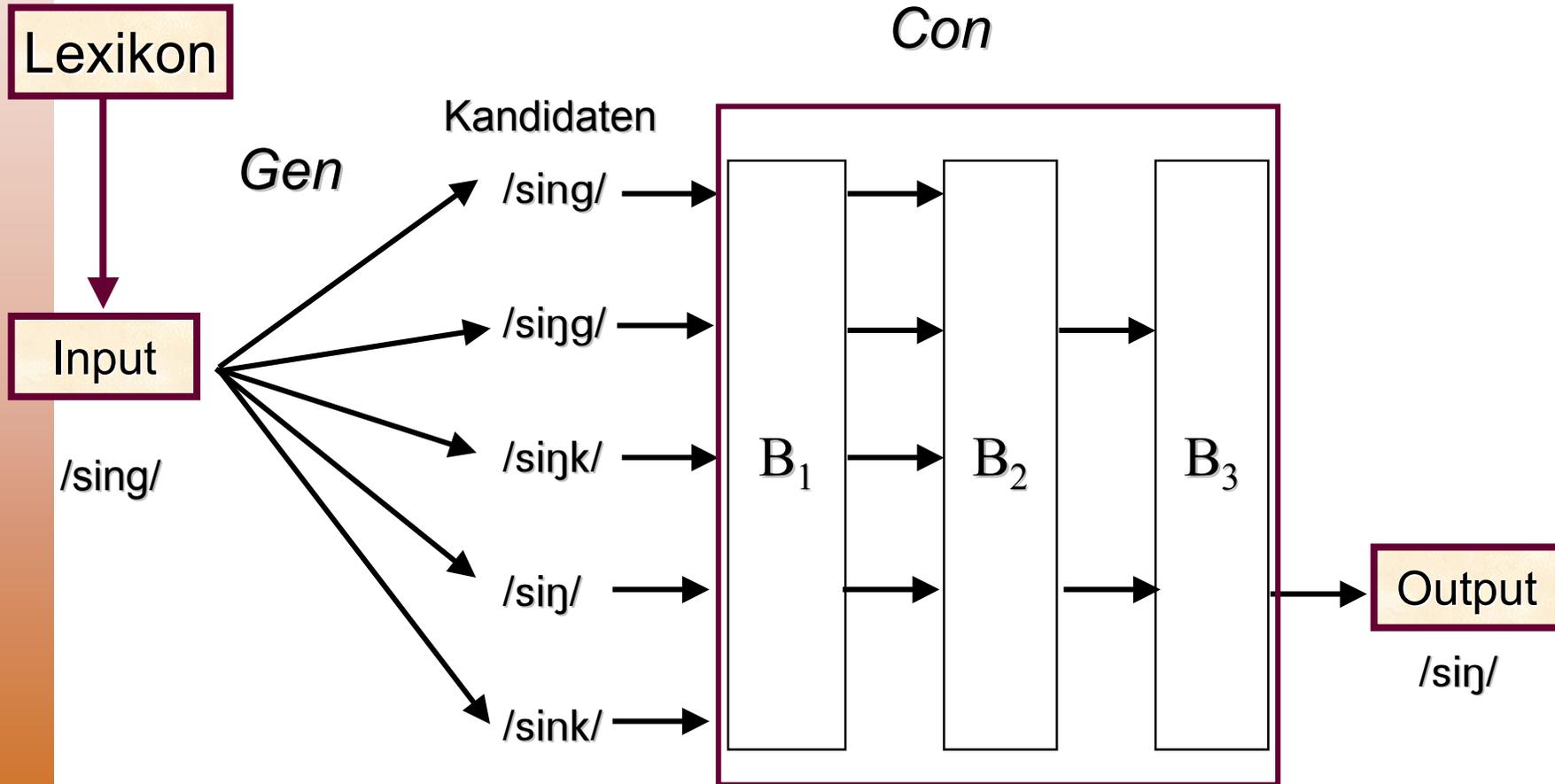
Der Generator erzeugt aus einem Input eine potentiell unendliche Menge von Output-Kandidaten K_i ($1 \leq i \leq \infty$) und übergibt sie an den Evaluator.

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.



Architektur der Optimalitätstheorie



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 zwei **Typen** von Beschränkungen unterschieden werden:
 - ▶ **Treue-Beschränkungen** (*faithfulness constraints*)
 - ▶ **Markiertheits-Beschränkungen** (**markedness 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)
 - ▶ Obstruenten dürfen in der Kodaposition nicht stimmhaft sein
- **Gebote**
 - ▶ Sonoranten müssen stimmhaft sein
 - ▶ Silben müssen einen Anlaut haben (Onset)



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)
 - ▶ Der Output muß die lineare Abfolge aller Segmente des Inputs beibehalten (keine Metathesen)
 - ▶ Output-Segmente müssen Entsprechungen im Input haben (Epenthese-Verbot; Dep)
 - ▶ Output-Segmente und Input-Segmente müssen identische Merkmalswerte aufweisen (Ident-IO_{Merkmal}).



Treue-Beschränkungen

- Unter einem funktionalen Aspekt kann man sagen, daß Treue-Beschränkungen die Aufgabe haben, die lexikalischen Einheiten einer Sprache vor den "korrodierenden" Kräften der Markiertheits-Beschränkungen zu schützen. Das erfüllt zweierlei Zwecke:
 - ▶ es wird sichergestellt, daß eine Sprache ausreichend distinktive Oppositionen zur Verfügung hat, um formal verschiedene lexikalische Einheiten mit unterschiedlicher Bedeutung auszudrücken. (Vgl. im Gegensatz dazu das hohe Maß an Homophonie im Chinesischen)
 - ▶ Treue-Beschränkungen restringieren das Ausmaß der Gestalt-Variation lexikalischer Einheiten (In geschriebenen Sprachen wird diese Aufgabe hauptsächlich durch die Orthographie wahrgenommen; der konservative Charakter der engl. Orthographie bewahrt die Identität des Wortes (vgl. /'deməkræt ,demə'rkætɪc dɪmɔkrəsi/ vs. *democrat*)

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

- Beispiel: Auslautverhärtung im Deutschen
- Obstruenten in Kodaposition sind stimmlos: /hʊnt/ 'Hund' vs. /hʊndə/ 'Hunde'.
- Die zugrundeliegende lexikalische Form ist /hund/
- 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

Optimalitätstheorie (OT)

Auslautverhärtung – Deutsch

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



Optimalitätstheorie (OT)

Beispiel Englisch

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

