

Optimalitätstheorie



Entstehung

1991

Alan Prince & Paul Smolensky: *Optimality*.
University of Arizona Phonology Conference in Tucson,
April 1991

Notes on connectionism and harmony theory

1993

Alan Prince & Paul Smolensky: *Optimality Theory:
Constraint Interaction in Generative Grammar*.

*Prosodic Morphology I: Constraint Interaction and
Satisfaction*.



Was ist OT?

- Formale Theorie über sprachliche Universalien und somit der Universalgrammatik
- Gleichzeitig Theorie der Sprachvariation
- Die Grammatiken aller Einzelsprachen haben eine Menge von als universell angenommenen Beschränkungen gemeinsam (constraints).
 - ◇ Silben haben einen Anlaut (**Onset**)
 - ◇ Silben haben keinen Auslaut (**No-Coda**)
 - ◇ Optimale Silbe: **CV**
- Beschränkungen sind zwar universell gültig, aber verletzbar (*violable*).



Was ist OT?

- Diese Beschränkungen sind hinreichend einfach und generell, so dass sie in vielen Kontexten in Konflikt zueinander geraten: sie können nicht gleichzeitig erfüllt werden.
- Die jeweilige Grammatik einer Einzelsprache löst diese Konflikte, indem sie die Beschränkungen in einer Hierarchie ordnet, derart, dass höherrangige Beschränkungen absolute Priorität vor allen niedrigerrangigen haben.
- Grammatiken unterscheiden sich ausschließlich darin, wie sie die universellen Beschränkungen zu Hierarchien ordnen.
- Die Typologie aller möglichen Sprachen ist das Resultat aller möglichen Rangordnungen der universellen Beschränkungen.



OT als Formale Theorie

- Formal steht hier im Gegensatz zu substantiell (gegenstandsbezogen, engl. *substantive*).
- Die OT per se behandelt nicht spezifische substantielle Phänomene, egal ob sie aus der Phonologie, Morphologie oder der Syntax stammen.
- Sie liefert vielmehr einen theoretischen Rahmen, in dem eine substantielle Theorie ausgedrückt werden kann.
- Die OT ist also im Prinzip auf alle gegenstandsbezogene beschränkungs- oder prinzipienbasierte Theorien anwendbar, wenn deren Beschränkungen oder Prinzipien im Sinne der OT interpretiert oder reinterpretiert werden können.



Anwendungsbereiche

- Phonologie (generativ)
- Morphophonologie – Morphologie
- Syntax
 - ◇ Prinzipien und Parameter – Minimalist Program
 - ◇ Lexikalistisch Funktionale Grammatik (LFG)
- Semantik und Pragmatik
- Spracherwerbstheorie
 - ◇ Erstspracherwerb
 - ◇ Zweitspracherwerb (Interlanguage)
- Sprachgeschichte
- Sprachkontakt



Analogie 1: Verkehrsregeln

- Rechts vor links: Wer von rechts kommt, hat Vorfahrt
- Strasse vor Feldweg
- Verkehrszeichen
- Lichtzeichen
- Blaulicht, Einsatzhorn
- Verkehrspolizist



Analogie 2: Schaltjahre

- Ein Jahr ist dann ein Schaltjahr, wenn man die Jahreszahl durch vier teilen kann (Schaltjahr), außer, wenn man die Jahreszahl auch durch 100 teilen kann (kein Schaltjahr), außer, wenn man die Jahreszahl auch noch durch 400 teilen kann (Schaltjahr).
 1. Jahre, die durch 4 ohne Rest teilbar sind, haben 366 Tage
 2. Jahre, die durch 100 ohne Rest teilbar sind, haben 365 Tage
 3. Jahre, die durch 400 ohne Rest teilbar sind, haben 366 Tage
 4. Ein Jahr hat normalerweise 365 Tage (default)



Schaltjahre optimalitätstheoretisch

| Tableau für 1996 | | Alle durch 400 teilbaren Jahre haben 366 Tage | Alle durch 100 teilbaren Jahre haben 365 Tage | Alle durch 4 teilbaren Jahre haben 366 Tage | Alle Jahre haben 365 Tage |
|---|---------------------------|---|---|---|---------------------------|
| a. | *1996 hat 364 Tage | | | * ! | * |
| b. | *1996 hat 365 Tage | | | * ! | |
|  | *1996 hat 366 Tage | | | | * |



Schaltjahre optimalitätstheoretisch

| Tableau für 1900 | | Alle durch 400 teilbaren Jahre haben 366 Tage | Alle durch 100 teilbaren Jahre haben 365 Tage | Alle durch 4 teilbaren Jahre haben 366 Tage | Alle Jahre haben 365 Tage |
|---|---------------------------|---|---|---|---------------------------|
| a. | *1900 hat 364 Tage | | * ! | * | * |
|  | *1900 hat 365 Tage | | | * | |
| c. | *1900 hat 366 Tage | | * ! | | * |



Schaltjahre optimalitätstheoretisch

| Tableau für 2000 | | Alle durch 400 teilbaren Jahre haben 366 Tage | Alle durch 100 teilbaren Jahre haben 365 Tage | Alle durch 4 teilbaren Jahre haben 366 Tage | Alle Jahre haben 365 Tage |
|------------------|---------------------------|---|---|---|---------------------------|
| a. | *2000 hat 364 Tage | * ! | * | * | * |
| b. | *2000 hat 365 Tage | * ! | | * | |
| c. | *2000 hat 366 Tage | | * | | * |



Schaltjahre optimalitätstheoretisch

| Tableau für 2000 Falsches Ranking | | Alle durch 100 teilbaren Jahre haben 365 Tage | Alle durch 400 teilbaren Jahre haben 366 Tage | Alle durch 4 teilbaren Jahre haben 366 Tage | Alle Jahre haben 365 Tage |
|---|-------------------------------|--|--|--|---------------------------------|
| a. | *2000 hat 364 Tage | * ! | * | * | * |
|  | *2000 hat 365 Tage | | * | * | |
| c. | *2000 hat 366 Tage | * ! | | | * |



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

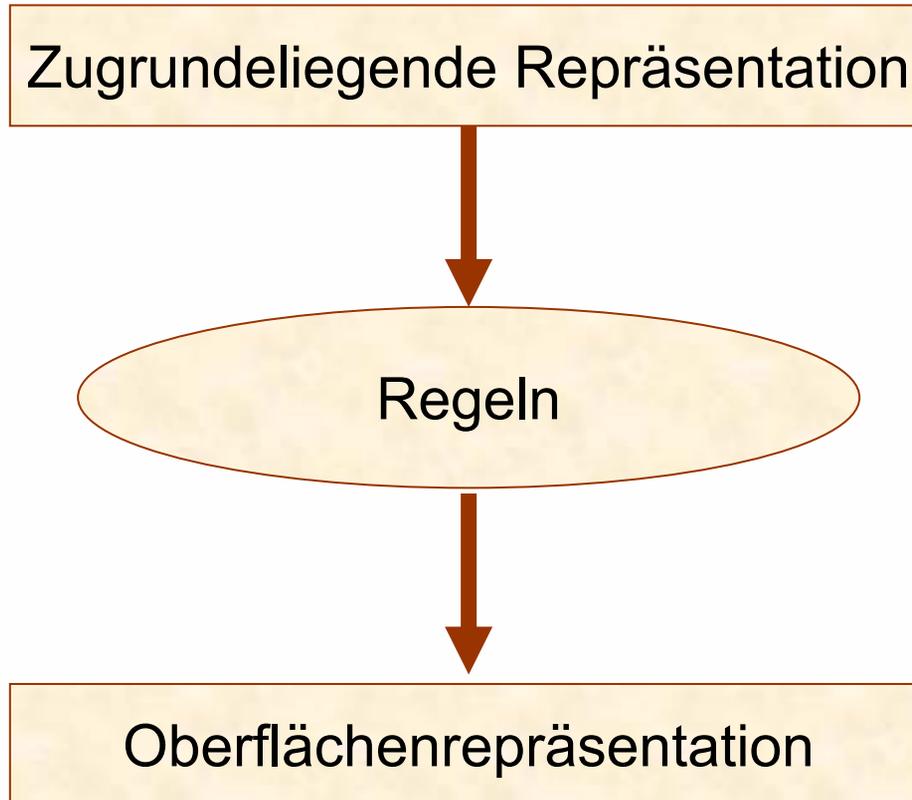
- ◇ $G(/sink/) = /sɪŋk/$
- ◇ $G(/sing/) = /sɪŋ/$

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





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.



Komponenten einer OT-Grammatik: Generator

- **Gen** ist universell, was heißt, dass die von **Gen** für einen bestimmten Input erzeugten Kandidaten für alle Sprachen gleich sind.
- Diese Kandidatenmenge muss jedoch auch hinreichend variabel sein, um alle in den Sprachen der Welt beobachtbaren Unterschiede erfassen zu können.
- Diese Eigenschaft von **Gen** wird "Freiheit der Analyse" (*freedom of analysis*) genannt.
- Diese Freiheit wird nur durch einige für alle Sprachen grundlegenden Strukturprinzipien eingeschränkt, in der Phonologie beispielsweise durch die Annahme eines spezifischen Inventars von distinktiven Merkmalen und in der Syntax durch so etwas wie die X'-Theorie.



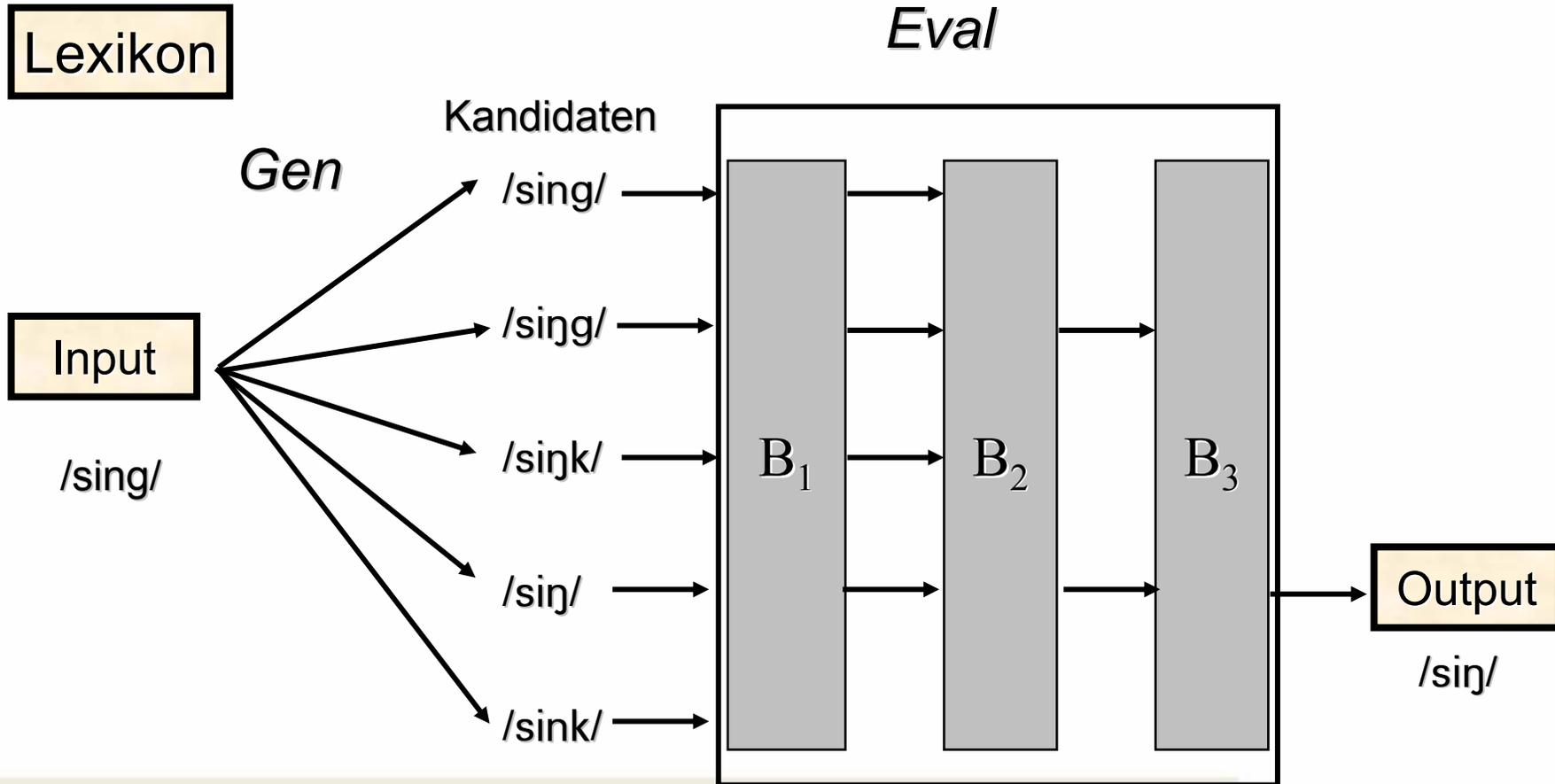
Komponenten einer OT-Grammatik: Evaluator

- Die Grammatik einer Sprache besteht aus einer spezifischen Beschränkungshierarchie **H**, d.h. aus einer Menge von geordneten Beschränkungen $\{B_1 \gg B_2 \gg \dots B_n\}$.
- Der Evaluator **Eval** bewertet die Output-Kandidaten auf der Grundlage von **H** hinsichtlich ihrer "Harmonie-Werte" und wählt danach den optimalen Kandidaten aus. Genauer müsste man sagen, dass **Eval** die Kandidatenmenge **K** auf der Grundlage von **H** harmonisch ordnet:

$$\text{Eval}(K, H) = K_h, \text{ wobei } K_h = \left[K_{\text{Opt}} \succ K_i \succ K_j \succ \dots K_n \right].$$
- K_{Opt} ist der optimale Kandidat und \succ steht für die Relation "ist harmonischer als".



Architektur der Optimalitätstheorie



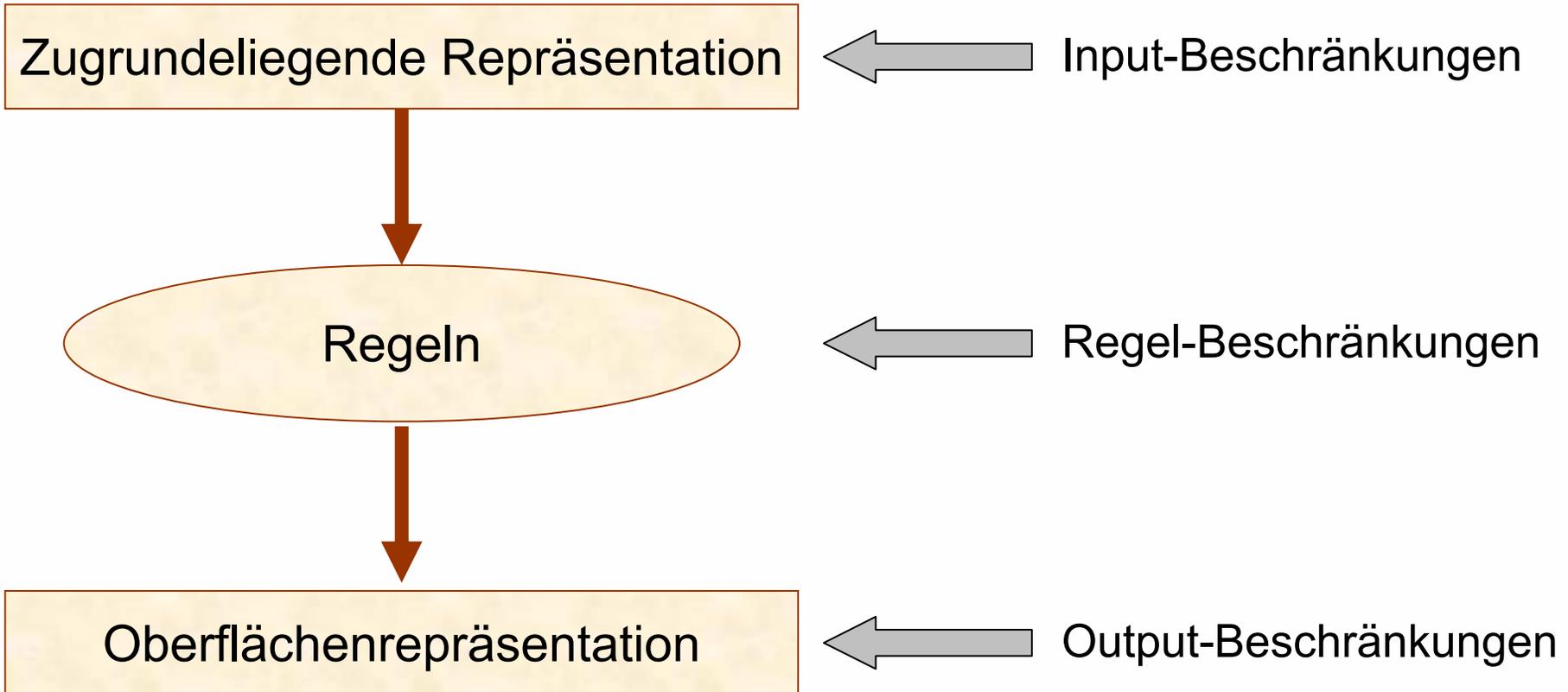


Richness of the Base

- Die Menge der möglichen Inputs für die Grammatiken aller Sprachen ist identisch und muss entsprechend reichhaltig sein (*Richness of the Base*).
- In der Phonologie sind sie z.B. bestimmt durch das allgemeine menschliche Lautbildungspotential.
- Die Formen des Lexikons einer Sprache sind nach dieser Annahme natürlich in der Menge der möglichen Inputs enthalten.



Beschränkungen - *Constraints*





Beschränkungen - *Constraints*

- Eine **Beschränkung** in der OT ist eine **Strukturbedingung**, die von einer potentiellen Output-Form entweder **erfüllt** (engl. *satisfy*) oder **verletzt** (engl. *violate*) werden kann.
- Es können grob 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 Kopf 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**[imality]; früher: **Parse**)
 - ◇ Der Output muß die lineare Abfolge aller Segmente des Inputs beibehalten (keine Metathesen, **Linearity**)
 - ◇ Output-Segmente müssen Entsprechungen im Input haben (Epenthese-Verbot, **Dep**[endence]; früher: **Fill**)
 - ◇ Output-Segmente und Input-Segmente müssen identische Merkmalswerte aufweisen (**Ident**[ity](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'krætic dɪmɔkrəsi/ vs. *democrat(ic)*, *democracy*)



Beschränkungsfamilien

- Alignment
- Correspondence
- Faithfulness
- Markedness
- Sonority
- Structural



Optimalität, Konflikt und Herrschaft

- **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.



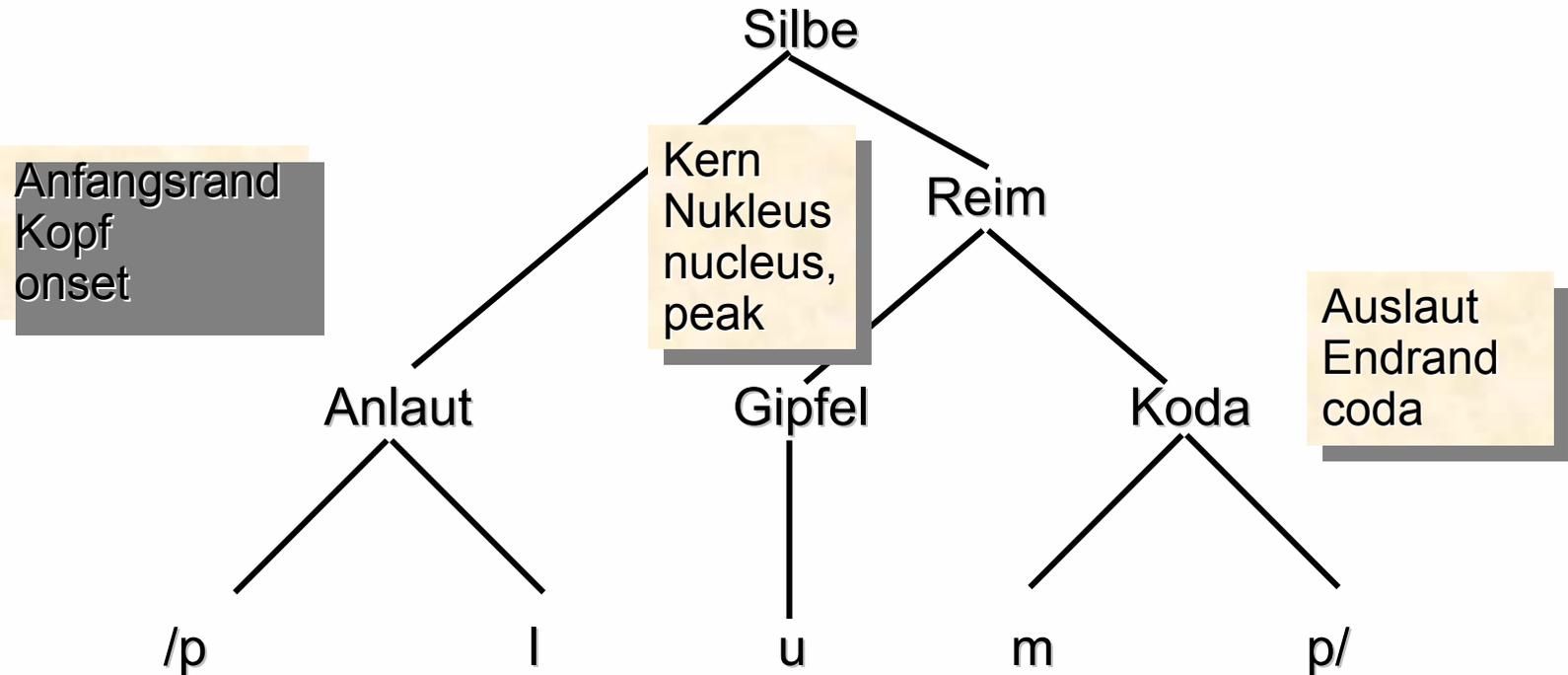
Beschränkungs-Interaktion – Constraint Interaction

- | | |
|--|--|
| <ul style="list-style-type: none"> <input type="checkbox"/> Lo[p] – Lo[b]es <input type="checkbox"/> Ra[t] – Ra[d]es <input type="checkbox"/> Sar[k] – Sär[g]e <input type="checkbox"/> akti[f] – akti[v]e <input type="checkbox"/> Gra[s] – Grä[z]er | <p>Auslautverhärtung im Deutschen: Obstruenten im Auslaut werden stimmlos (fortis). Genauer um den Silbenauslaut geht, womit sich die Frage erhebt, wie im Deutschen der Silbenauslaut bestimmt ist.</p> |
|--|--|

$$[-sonor] \rightarrow [-sth] / \underline{\quad}]_{\sigma}$$

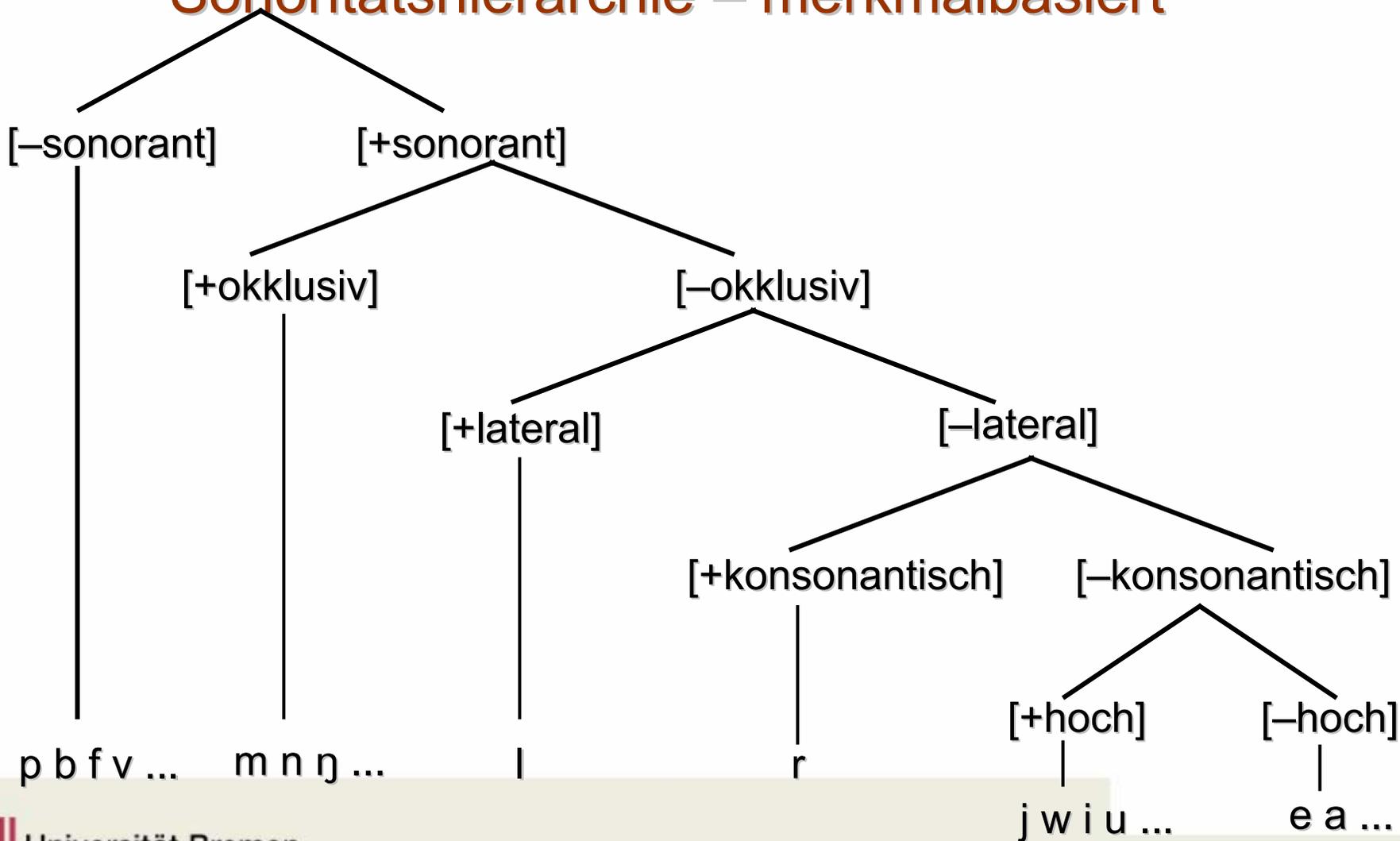


Silbenstruktur: Grundbegriffe





Sonoritätshierarchie – merkmalsbasiert





Die Kernsilbe

- Jede Silbe hat einen Sonoritätsgipfel, den Silbengipfel (Nukleus, Kern).
- Dieses Segment hat einen höheren Sonoritätsgrad als die Silbenschale (d.h. die linken und rechten Nachbarn).
- Die dem Silbengipfel vorausgehenden Segmente bilden eine phonologische Einheit, den Silbenanlaut (Kopf, Anfangsrand, Onset).
- Die Sonorität des Anlauts nimmt zum Gipfel hin zu.
- Die auf den Gipfel folgenden Segmente bilden eine phonologische Einheit, die Koda (Auslaut, Endrand, Coda).
- Die Sonorität der Koda nimmt vom Gipfel zum Rand hin ab.
- Gipfel und Koda bilden zusammen eine phonologische Einheit, den Reim.



Beschränkungs-Interaktion – Constraint Interaction

- Obstruenten in Kodaposition sind stimmlos: /hʊnt/ 'Hund' vs. /hʊn.də/ 'Hunde'.
- Als zugrundeliegende lexikalische Form ist /hʊnd/ anzunehmen.
- Zur optimalitätstheoretischen 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 [hunt] (neben vielen anderen wie [hun], [hunəd], [hunət], [hund]). Wir wollen uns auf die ersten beiden beschränken:
 - ◇ [hund] erfüllt **Ident-IO(sth)**, aber verletzt ***Voiced-Coda**
 - ◇ [hunt] verletzt **Ident-IO(sth)**, aber erfüllt ***Voiced-Coda**
- Wir erhalten die optimale Form [hunt], wenn wir von folgender Rangordnung der Beschränkungen ausgehen:
 - ◇ ***Voiced-Coda >> Ident-IO(sth)**
- Für das Englische würde umgekehrt gelten:
 - ◇ **Ident-IO(sth) >> *Voiced-Coda**



Beschränkungs-Interaktion: Auslautverhärtung Deutsch

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



Beschränkungs-Interaktion: Auslautverhärtung Deutsch

| /hʊn.də/ | | *Voiced-Coda | Ident-IO(sth) |
|----------|---|--------------|---------------|
| a. | [hʊn.də]  | | |
| b. | [hʊn.tə] | | * ! |



Beschränkungs-Interaktion: Englisch

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



Die optimale Silbe: CV

Markiertheitsbeschränkungen (structural constraints)

- Onset Silben müssen einen Kopf haben
- Nucleus Silben müssen einen Gipfel (Nucleus) haben
- No-Coda Silben dürfen keine Koda haben
- *Complex Silbenkonstituenten dürfen nicht komplex sein
 - ◇ *Complex-Onset Silben dürfen keinen komplexen Anlaut haben
 - ◇ *Complex-Coda Silben dürfen keine komplexe Koda haben

Treuebeschränkungen (faithfulness constraints)

- Max Im Output müssen alle Segmente des Inputs erhalten bleiben (keine Tilgung: [Max\[imality\]](#))
- Dep Output-Segmente müssen Entsprechungen im Input haben (Epenthese-Verbot: [Dep\[endency\]](#))



Silbentypen

| | CV | CCV | CVC | CVV | V | C |
|----------|----|-----|-----|-----|---|---|
| Onset | ✓ | ✓ | ✓ | ✓ | * | ✓ |
| Nucleus | ✓ | ✓ | ✓ | ✓ | ✓ | * |
| No-Coda | ✓ | ✓ | * | ✓ | ✓ | ✓ |
| *Complex | ✓ | * | ✓ | * | ✓ | ✓ |



Tableau für die optimale Silbe CV

| Kandidaten | Onset | Nucleus | No-Coda | *Complex | Max | Dep |
|---|-------|---------|---------|----------|-----|-----|
| V | * ! | | | | | |
| C | | * ! | * ! | | | |
|  CV | | | | | | |
| VC | * ! | | * ! | | | |
| CVC | | | * ! | | | |
| CCVC | | | * ! | * ! | | |



Daten: frühkindliche Äußerungen (Timmy)

| 11 Monate | | 15 Monate | | 16 Monate | |
|---------------------|-------|-----------|------|--------------------|--------|
| p ^h ə | ball | bæ: | bird | pæ | baby |
| pæ | book | gɛ: | cow | pæ | block |
| ga: | duck | ka | cup | pæ | boat |
| ?a | hi | gæ: | dog | k'ak ^{hi} | cookie |
| hək: ^h a | key | kʌ | girl | ?ʌφæ | flower |
| kə | kitty | ?əm:a | moon | nʌmæ | Simon |



Struktur einfacher Silben im Englischen

| /εg/ | Dep | Max | Onset | No-Coda |
|---|-----|-----|-------|---------|
|  εg | | | * | * |
| εgə | * ! | | * | |
| ε | | * ! | * | |
| ?εg | * ! | | | * |



Struktur einfacher Silben im Englischen

| /kleɪ/ | Dep | Max | Onset | No-Coda | *Complex |
|---|-----|-------|-------|---------|----------|
|  kleɪ | | | | | * * |
| kle | | * ! | | | * |
| ke | | * ! * | | | |
| kəleɪ | * ! | | | | * |
| kəle | * ! | * ! | | | |



Daten: Englische Wörter im Japanischen

Englisch

- Christmas
- text
- club
- dress
- glass (zum Trinken)
- disc
- slum
- plus
- bolt
- grotesque

Japanisch

- kurisumasu
- tekisuto
- kurabu
- doresu
- gurasu
- disuku
- suramu
- purasu
- boruto
- gurotesuku



Englisch – Japanisch

| /tekst/ | | *Complex | No-Coda | Dep |
|---------|-----------------|----------|---------|-------|
| a. | [. tekst.] | * ! | * | |
| b. | [.te.k̚.s̚.t̚.] | | | * * * |





Silbentypen sprachvergleichend

| Sprachbeispiel | Typ(en) | | | | |
|----------------|---------|-----|--------|-----------|-----------|
| Hua | CV | | | | |
| Thargari | CV | CVC | | | |
| Cayuvava | CV | V | | | |
| Arabela | CV | CCV | | | |
| Sedang | CV | CVC | CCV(C) | | |
| Mazateco | CV | V | CCV | | |
| Klamath | CV | CVC | CVCC | | |
| Mokilese | CV | CVC | V(C) | | |
| Totonac | CV | CVC | CVCC | CCV(C)(C) | |
| Finnisch | CV | CVC | V(C) | (C)VCC | |
| Spanisch | CV | CVC | V(C) | CCV(C) | |
| Holländisch | CV | CVC | V(C) | CCV(C) | (C)(C)VCC |



„Faktorielle“ Typologie der Silbe

(a) Unmarkiert

Hua Onset No-Coda *Complex-O *Complex-C >>**Faith**

(b) Markiert I

Cayuvava No-Coda *Complex-O *Complex-C >>**Faith**>> Onset

Thargari Onset *Complex-O *Complex-C >>**Faith**>> No-Coda

Arabela Onset No-Coda *Complex-C >>**Faith**>> *Complex-O

(c) Markiert II

Sedang Onset *Complex-C >>**Faith**>> No-Coda *Complex-O

Mazateco No-Coda *Complex-C >>**Faith**>> Onset *Complex-O

Mokilese *Complex-O *Complex-C >>**Faith**>> Onset No-Coda

Klamath Onset *Complex-O >>**Faith**>> No-Coda *Complex-C



„Faktorielle“ Typologie der Silbe

(d) Markiert III

| | | | | | |
|----------|------------|-----------|---------|------------|------------|
| Totonac | Onset | >>Faith>> | No-Coda | *Complex-C | *Complex-O |
| Spanisch | *Complex-C | >>Faith>> | Onset | No-Coda | *Complex-O |
| Finnisch | *Complex-O | >>Faith>> | Onset | No-Coda | *Complex-C |

(e) Markiert IV

| | | | | | |
|-------------|---------|-------|---------|------------|------------|
| Holländisch | Faith>> | Onset | No-Coda | *Complex-O | *Complex-C |
|-------------|---------|-------|---------|------------|------------|



Kommentar

Alle anderen Rangordnungen sind also in den Sprachen der Welt attestiert und es gibt offenbar keine anderen Sprachen, für die andere Beschränkungen und/oder Rangordnungen postuliert werden müssten, um ihr Inventar an Silbentypen charakterisieren zu können. Die Rangordnungen sind nach dem im Output zugelassenen Komplexitätsgrad gruppiert:

- ◇ In (a) finden wir die *unmarkierte Rangordnung* für Hua,
- ◇ in (b) haben wir Rangordnungen des *Markiertheitsgrades I*: sie führen zu Silbenstrukturen im Output, die in maximal einer Hinsicht strukturell markiert sein können.
- ◇ die Rangordnungen in (c) haben den *Markiertheitsgrad II*: sie lassen *wohlgeformte Silben* zu, die in zweifacher Hinsicht markiert sind.
- ◇ in (d) (*Markiertheitsgrad III*) haben wird Rangordnungen, die in dreifacher Weise markierte Silben zulassen.
- ◇ in (e) schließlich (*Markiertheitsgrad IV*) können Silben auf vierfache Weise markiert sein.



Faktorielle Typologie und Ontogenese

- Die ontogenetische Sprachentwicklung führt von einem Initialzustand mit einer Grammatik G_i zu einem Endzustand (final state) mit einer Grammatik G_f
- Es wird angenommen, daß G_i einen Output erzeugt, der maximal unmarkiert ist.
- G_f ist entweder identisch mit G_i oder eine Grammatik, die einen strukturell stärker markierten Output zuläßt.
- Der Lernweg von der Initialgrammatik G_i zur Finalgrammatik G_f ist eine Folge von Grammatiken, die einen zunehmend stärker markierten Output zulassen.
- Aus der faktoriellen Typologie für die Silbenstruktur lassen sich zwölf verschiedene Lernwege ableiten, welche die Grammatik unmarkierter Sprachen über die der Markiertheitsgrade I, II und III mit der am stärksten markierter Sprachen des Grades IV verbinden.



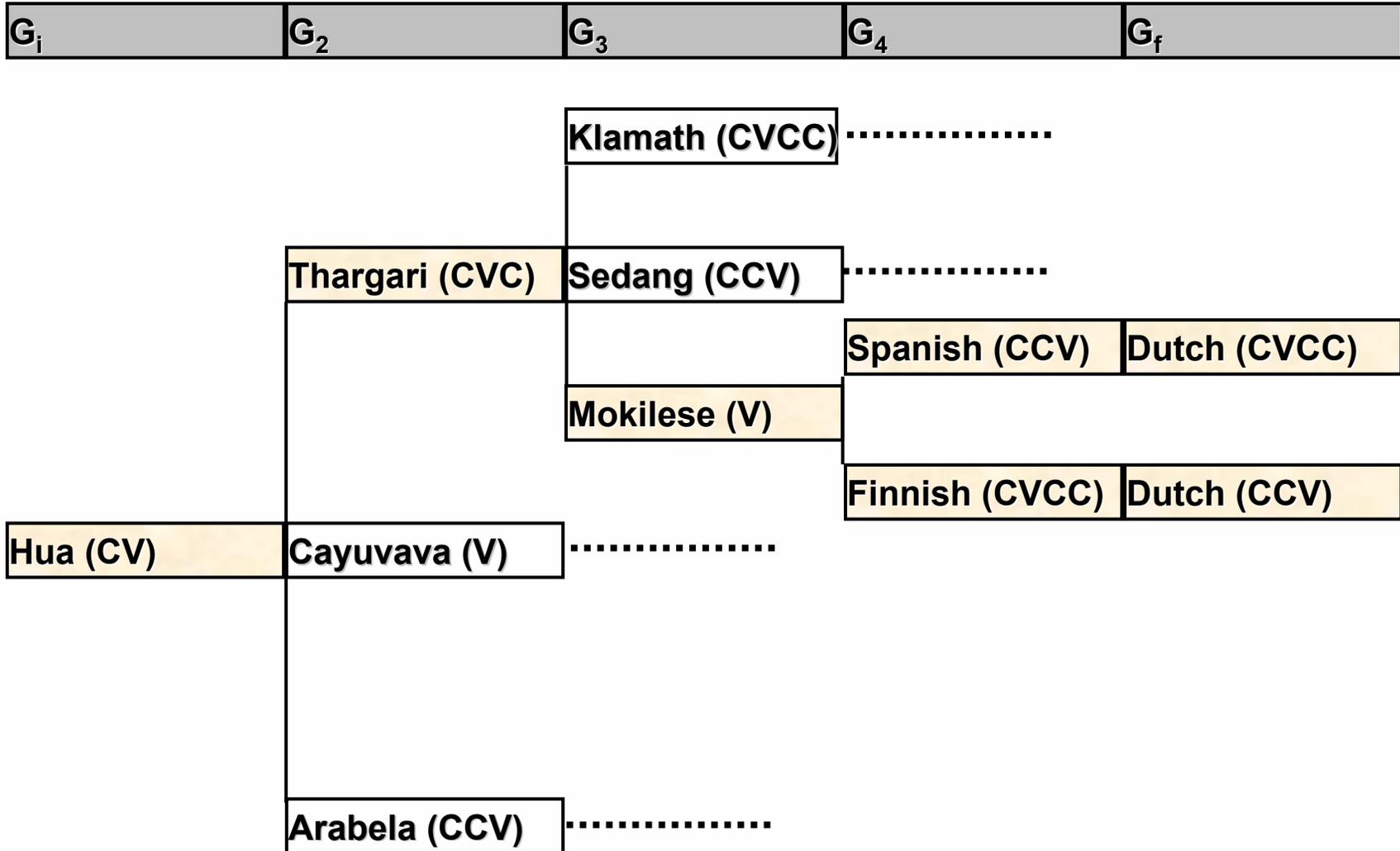
Faktorielle Typologie und Ontogenese

- Die Initialgrammatik zeichnet sich dadurch aus, daß alle Markiertheitsbeschränkungen vor den Treuebeschränkungen (Faith) rangieren.
- {Onset No-Coda *Complex-O *Complex-C} >> **Faith**
- Mit zunehmender Markiertheit wandert **Faith** in der Rangordnung nach oben.



Entwicklungsstadien beim Erwerb von Silbenstrukturen

| | | | | | | | |
|-----------------|-----------|------------|------------|------------|-----------|---------------|---------|
| G _i | *Compl=&C | *Complex-C | *Complex-O | Ons&NC | Onset | No-Coda | >>Faith |
| G ₂ | *Compl=&C | *Complex-C | *Complex-O | Ons&NC | Onset | >>Faith> > | No-Coda |
| G ₃ | *Compl=&C | *Complex-C | *Complex-O | Ons&NC | >>Faith>> | Onset | No-Coda |
| G ₄ | *Compl=&C | *Complex-C | *Complex-O | >>Faith>> | Ons&NC | Onset | No-Coda |
| G _{5A} | *Compl=&C | *Complex-O | >>Faith>> | *Complex-C | Ons&NC | Onset | No-Coda |
| G _{5B} | *Compl=&C | *Complex-C | >>Faith>> | *Complex-O | Ons&NC | Onset | No-Coda |
| G ₆ | *Compl=&C | >>Faith>> | *Complex-C | *Complex-O | Ons&NC | Onset | No-Coda |
| G _f | Faith>> | *Complex-C | *Complex-C | *Complex-O | Ons&NC | Onset | No-Coda |





OT-Syntax: Input – Gen

- $\langle \text{sing}(x), x=\text{John}, x=\text{topic} \rangle$
- $\langle \text{give}(x,y,z), x=\text{topic}, z=\text{focus}; x=\text{John}, y=\text{present}, z=\text{Mary} \rangle$
- GEN generiert aus dem Input die Menge der Strukturbeschreibungen, die den Beschränkungen der X'-Theorie genügen.
- Aus $\langle \text{sing}(x), x=\text{topic}, x=\text{John} \rangle$
 - ◇ $[_{IP} \text{ has sung}]$
 - ◇ $[_{IP} \text{ John}_i \text{ has } [t_i \text{ sung}]]$
 - ◇ $[_{IP} \text{ has } [t_i \text{ sung}] \text{ John}_i]$



OT-Syntax: Gen

- Project(X): diese Operation projiziert ein lexikalisches oder funktionales Element X in eine X'-Theorie-konforme Projektion
- Compose(X,Y): diese Operation fügt Projektionen zusammen und muß dem Strukturerhaltungsprinzip folgen.
- Move- α (Y)
- Case-assign
- Agree



Syntax: Eval

- **Subject:** Die Subjektposition in einer erweiterten Projektion muß gefüllt sein, d.h. Sätze haben ein Subjekt.
- **Full-Int:** Jedes Element eines sprachlichen Ausdrucks trägt zu dessen Interpretation bei, d.h. keine semantisch leeren Elemente.
- **DropTopic:** Argumente mit Topic-Funktion werden nicht ausgedrückt.
- **Max:** Alle Elemente des Inputs müssen Entsprechungen im Output haben



Optimale Subjekte

DropTopic >> Max >> Subject

Input: <cantare(x), x=topic, x=lui>

| Kandidaten | DropTopic | Max | Full-Int | Subject |
|---|-----------|-----|----------|---------|
|  ha cantato | | * | | * |
| lui ha cantato | * ! | | | |
| ha cantato lui | * ! | | | * |



Optimale Subjekte: Englisch

Parse >> DropTopic

Input: <sing(x), x=he, x=topic>

| Kandidaten | Max | DropTopic | Subject | Full-Int |
|---|-----|-----------|---------|----------|
| has sung | * ! | | * | |
|  he has sung | | * | | |
| has sung he | | * | * ! | |



Subjekt nicht Topic

DropTopic >> Parse >> Subject

Input: <cantare(x), x=lui>

| Kandidaten | DropTopic | Max | Full-Int | Subject |
|--|-----------|-----|----------|---------|
| ha cantato | | * ! | | * |
|  lui ha cantato | | | | |
| ha cantato lui | | | | * ! |



Subjekt nicht Topic - Englisch

Parse >> DropTopic

Input: <sing(x), x=he>

| Kandidaten | Max | DropTopic | Subject | Full-Int |
|---|-----|-----------|---------|----------|
| has sung | * ! | | * | |
|  he has sung | | | | |
| has sung he | | | * ! | |



Expletiva - Englisch

{Parse, Subjet} >> Full-Int

Input: <seem(x), x=leave(y), y=John>

| Kandidaten | Max | DropTopic | Subject | Full-Int |
|---|-----|-----------|---------|----------|
| seems [that ...] | | | *! | |
|  it seems [that ...] | | | | * |



Expletiva - Englisch

Parse >> DropTopic

Input: <sing(x), x=he, x=topic>

| Kandidaten | Max | DropTopic | Subject | Full-Int |
|--|-----|-----------|---------|----------|
|  he has sung | | * | | |
| Expl has sung | * ! | | | * |
| Expl has sung he | | * | | * ! |



Expletiva - Italienisch

{Full-Int, Max} >> Subject

Input: <sembra(x), x=andare(y), y=Gianni>

| Kandidaten | DropTopic | Max | Full-Int | Subject |
|---|-----------|-----|----------|---------|
|  sembra [che ...] | | | | * |
| Expl sembra [che ...] | | | * ! | |



Expletiva - Italienisch

Full-Int >> Subject

Input: <cantare(x), x=Gianni, x=topic>

| Kandidaten | DropTopic | Max | Full-Int | Subject |
|---|-----------|-----|----------|---------|
|  ha cantato | | * | | * |
| Expl. ha cantato | | * | * ! | |
| Expl. ha lui cantato | * ! | | * | |