
Lernbarkeitstheorie

Ein Lernbarkeitsresultat für Transformationsgrammatiken

Erinnerung

- Der Input für den Lerner besteht aus einer Folge von Paaren $\langle b_1, s_1 \rangle, \langle b_2, s_2 \rangle, \dots$ wobei
 1. b_i eine TS ist, die von der Basiskomponente B (eine ktf Grammatik) erzeugt wird, und
 2. s_i eine Terminalkette ist, die zu der OS gehört, welche aus b_i durch die Transformationskomponente der Zielgrammatik abgeleitet wird.
- Die Zielgrammatik ("adult grammar") wird kurz auch als A geschrieben, die Lernergrammatik ("child grammar") als C .
- LP ist erfolgreich, wenn sie eine Menge an Transformationen erschlossen hat, die es erlauben, TSen korrekt (im Sinne von A) auf Ketten unter OSen abzubilden (d.h., wenn gilt: $A(b_i) = s_i = C(b_i)$).

Annahmen

- Wexler & Culicover (1980) machen noch weitere Annahmen.
- Jede **Rekursion**, die durch B erzeugt wird, läuft über das Satzsymbol S :
 1. Angenommen, ein Symbol D dominiert eine andere Instanz von D .
 2. Dann muss es ein S -Symbol geben, so dass S von der ersten Instanz von D dominiert wird und seinerseits die zweite Instanz von D dominiert.

Annahmen 2

- Es gilt der **Zyklus**: Alle Transformationen applizieren zuerst auf der tiefsten zyklischen Domäne (Satzdomäne), dann auf der nächsthöheren, usw.
- Allen möglichen Grammatiken liegt dieselbe **Basis-komponente** zugrunde (vgl. Wexler & Culicover 1980, 85, Fußnote 1).
- Alle Transformationen sind **obligatorisch**: Wenn eine Struktur die strukturelle Beschreibung einer Transformation T erfüllt, dann muss T unmittelbar ausgeführt werden.

Annahmen 3

- Die Transformationskomponente von A ist **deterministisch**, d.h.,
 1. zu jedem Punkt der Derivation gibt es höchstens eine Transformation T , die angewandt werden kann, und
 2. T kann an jedem Punkt der Derivation nur auf eine Weise angewandt werden.
- Cs Transformationskomponente ist möglicherweise zunächst nicht deterministisch, aber gegen Ende schon.
- Determinismus vermeidet folgendes Problem:
 1. Angenommen, an einem Punkt der Derivation wäre die strukturelle Beschreibung von zwei obligatorischen Transformationen T und T' erfüllt.
 2. Dann sollten T und T' unmittelbar ausgeführt werden.
 3. Denn es ist unklar, was es heißt, dass T und T' **simultan** angewandt werden sollen.

Funktionslernbarkeit

- Sei F eine Klasse von Funktionen von einer Menge A (hier ist nicht die Zielgrammatik gemeint!) nach einer Menge R .
- Wenn $f \in F$, dann ist eine **Informationssequenz** von f eine Folge $\langle a_1, f(a_1) \rangle, \langle a_2, f(a_2) \rangle, \dots$, so dass
 1. $a \in A$, und
 2. für jedes $a \in A$ gibt es wenigstens eine Instanz $\langle a, f(a) \rangle$ in der Informationssequenz.
- Sei $I = \langle a_1, f(a_1) \rangle, \langle a_2, f(a_2) \rangle, \dots$ eine Informationssequenz von f . Dann ist $S_t(I) = \{\langle a_1, f(a_1) \rangle, \dots, \langle a_t, f(a_t) \rangle\}$ eine **Auswahl** von I zur Zeit t .
- F ist **funktionslernbar**, wenn ein Algorithmus LP existiert, so dass für jedes $f \in F$ und jede Informationssequenz I von f ein τ existiert, so dass:
 1. $LP(S_\tau(I)) = f$ und
 2. wenn $t > \tau$, dann $LP(S_t(I)) = LP(S_\tau(I))$.

Arten von Funktionen

- Eine Funktion f von A nach B heißt
 1. **injektiv**, gdw. für jedes $b \in B$ höchstens ein $a \in A$ existiert, so dass $f(a) = b$.
 2. **surjektiv**, gdw. für jedes $b \in B$ mindestens ein $a \in A$ existiert, so dass $f(a) = b$.
 3. **bijektiv**, gdw. für jedes $b \in B$ genau ein $a \in A$ existiert, so dass $f(a) = b$ (f ist bijektiv gdw. f injektiv und surjektiv ist).

Endliche und unendliche Mengen

- Mengen A und B sind gleich **mächtig**, wenn es eine Bijektion zwischen A und B gibt.
- Eine Menge A ist **endlich**, wenn es eine Bijektion zwischen A und $\{1, \dots, n\}$ gibt, für ein $n \in \mathcal{N}$ (\mathcal{N} = Menge der natürlichen Zahlen).
- Eine Menge A ist **unendlich**, wenn sie nicht endlich ist.

Abzählbarkeit

- Eine Menge A ist
 1. **abzählbar** (engl.: countable), wenn
 - (a) A endlich ist, oder
 - (b) A abzählbar unendlich ist.
 2. **abzählbar unendlich**, wenn A gleich mächtig ist wie \mathcal{N} (es gibt Bijektion zwischen A und \mathcal{N}).
 3. **überabzählbar**, wenn A nicht abzählbar ist.

Überabzählbare Mengen

- Theorem: Die Menge $2^{\mathcal{N}}$ (die Menge aller Teilmengen der natürlichen Zahlen) ist überabzählbar unendlich.
- Beweis durch Widerspruch (basierend auf Cantors (1890) **Diagonalisierungsprinzip**):
 1. Angenommen $2^{\mathcal{N}}$ sei abzählbar unendlich. Dann existiert eine Bijektion von \mathcal{N} nach $2^{\mathcal{N}}$ (d.h., man kann jedes Element von $2^{\mathcal{N}}$ mit einer natürlichen Zahl indizieren): $2^{\mathcal{N}} = \{R_0, R_1, R_2, \dots\}$.
 2. Bilde die Diagonalmenge $D = \{n \in \mathcal{N} \mid n \notin R_n\}$.
 3. D enthält nur natürliche Zahlen, muss also ein Element von $2^{\mathcal{N}}$ sein.
 4. Also muss es ein $k \in \mathcal{N}$ geben, so dass $D = R_k$.
 5. Das kann aber nicht sein, da $D \neq R_k$, für jedes k (nach Konstruktion von D). Man hat also einen Widerspruch.
 6. Dann muss die Annahme, dass $2^{\mathcal{N}}$ abzählbar ist, falsch sein.

Aufzählbarkeit

- Eine Menge A ist (rekursiv) **aufzählbar** (engl.: enumerable), wenn es eine Funktion f gibt, so dass
 1. f von \mathcal{N} nach A abbildet,
 2. f surjektiv ist und
 3. f berechenbar ist.
- Jede Menge, die aufzählbar ist, ist auch abzählbar, aber nicht umgekehrt.
- Wegen Surjektivität ist erlaubt, dass Elemente mehrfach aufgezählt werden.
- Intuitiv: Eine Menge ist abzählbar, wenn sich ihre Elemente durch ein mechanisches Verfahren nacheinander nennen lassen, so dass es für jedes Element e einen Zeitpunkt t gibt, zu dem e genannt wird.

Funktionslernbarkeit durch Aufzählung

- Proposition: Jede aufzählbare Menge F von Funktionen ist funktionslernbar.
- Beweis (durch Angabe des Verfahrens):
 1. Sei $f \in F$ die gesuchte Funktion.
 2. Zähle die Funktionen in F auf als f_1, f_2, \dots .
 3. Rate zuerst $f = f_1$. Wenn $f \neq f_1$, dann muss es ein $a \in A$ geben, so dass $f(a) \neq f_1(a)$.
 4. Nach Definition von Informationssequenz wird das Paar $\langle a, f(a) \rangle$ auch irgendwann präsentiert, so dass dies festgestellt werden kann.
 5. Wenn $f(a) \neq f_1(a)$, dann rate $f = f_i$, wobei f_i die nächste Funktion der Aufzählung ist, die mit allen bisher präsentierten Daten kompatibel ist.
 6. f wird irgendwann gewählt (ist in Aufzählung).
 7. Da f mit allen Daten kompatibel ist, wird keine andere Funktion mehr gewählt.
- Voraussetzung: Es muss ein (berechenbares) Verfahren geben, um festzustellen, ob eine Funktion mit den bisherigen Daten kompatibel ist.

Funktionslernbarkeit und Transformationsgrammatiken

- Sei \mathcal{T}_B die Klasse der Transformationsgrammatiken über einer fixen Basiskomponente B .
- Behauptung: \mathcal{T}_B ist funktionslernbar.
- Erinnerung: \mathcal{T}_B war nicht mengenlernbar, damit auch nicht textlernbar.
- Man kann die Behauptung beweisen, in dem man zeigt, dass die Menge der Transformationskomponenten (über B) aufzählbar ist (siehe vorherige Proposition).
- Daraus folgt dann die Behauptung, da jede Transformationskomponente eine Funktion von TSen in Oberflächenketten definiert.

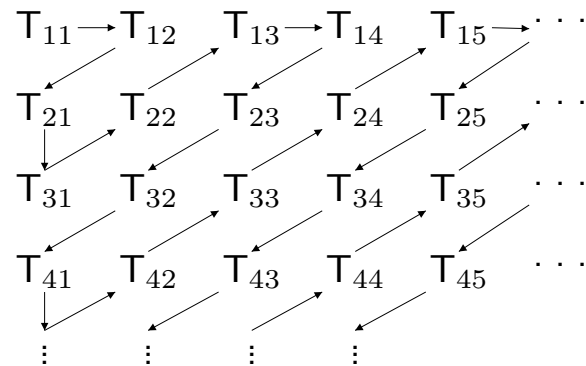
Funktionslernbarkeit und Transformationsgrammatiken 2

- Beweisskizze:
 1. Jede Transformation T besteht aus
 - (a) einer strukturellen Beschreibung, die angibt, in welchem Kontext T angewandt wird, und
 - (b) einem strukturellen Wandel, der angibt, welche Operation T durchführt.
 2. Jede Transformation T kann also geschrieben werden als T_{ij} mit $i =$ strukturelle Beschreibung und $j =$ struktureller Wandel.
 3. Die Menge aller möglichen Transformationen lassen sich dann in einer Tabelle darstellen:

T_{11}	T_{12}	T_{13}	T_{14}	T_{15}	\dots
T_{21}	T_{22}	T_{23}	T_{24}	T_{25}	\dots
T_{31}	T_{32}	T_{33}	T_{34}	T_{35}	\dots
T_{41}	T_{42}	T_{43}	T_{44}	T_{45}	\dots
T_{51}	T_{52}	T_{53}	T_{54}	T_{55}	\dots
\vdots	\vdots	\vdots	\vdots	\vdots	

Funktionslernbarkeit und Transformationsgrammatiken 3

4. Die Tabelle kann systematisch durchlaufen werden, wobei jede Transformation irgendeinmal drankommt.
5. Beachte: Wenn die Tabelle spalten- oder zeilenweise durchlaufen wird, kommt keine Aufzählung zustande, da sowohl die Spalten als auch Zeilen unendlich sind.
6. Trick: Man durchläuft die Tabelle diagonal.



7. Konsequenz: Die Menge der möglichen Transformationen ist aufzählbar.

Funktionslernbarkeit und Transformationsgrammatiken 4

8. Durch Aufzählbarkeit der Transformationen kann man die Transformationskomponenten aufzählen.
9. Man bildet im Schritt k des Tabellendurchlaufs jeweils die Potenzmenge (Menge aller Teilmengen) aller bis k durchlaufenen Transformationen.
10. Da es immer nur endlich viele solcher Transformationen für jedes k gibt, ist auch die Potenzmenge daraus endlich und kann systematisch und in endlicher Zeit erzeugt werden.

$$\begin{aligned}
 k = 1: & \quad \{ \{T_{11}\} \} \\
 k = 2: & \quad \{ \{T_{11}\}, \{T_{12}\}, \{T_{11}, T_{22}\} \} \\
 k = 3: & \quad \{ \{T_{11}\}, \{T_{12}\}, \{T_{21}\}, \{T_{11}, T_{22}\}, \\
 & \quad \{T_{11}, T_{21}\}, \{T_{21}, T_{22}\}, \\
 & \quad \{T_{21}, T_{11}, T_{22}\} \} \\
 & \quad \dots \quad \text{etc.}
 \end{aligned}$$

11. Die Elemente der Potenzmengen liefern eine Aufzählung aller möglichen Transformationskomponenten (es ist ja erlaubt, dass manche Elemente mehrmals aufgezählt werden).

Obermengenproblem

- Aufzählbarkeit hilft bei Mengenlernbarkeit nicht unbedingt.
- Betrachte die Grammatikklasse $\mathcal{H} = \{H_0, H_1, H_2, \dots\}$:
 1. $L(H_0) = \{a, aa, aaa, aaaa, \dots\}$
 2. $L(H_1) = \{a\}$
 3. $L(H_2) = \{a, aa\}$
 4. . . .
- \mathcal{H} ist aufzählbar, aber nicht lernbar dadurch.
- Begründung:
 1. Irgendwo in der Aufzählung kommt H_0 vor.
 2. Angenommen, es werden Daten präsentiert aus H_i , mit H_i in der Aufzählung **hinter** H_0 .
 3. Angenommen, alle H_j in der Aufzählung **vor** H_0 wurden abgelehnt; als nächstes wird H_0 gewählt.
 4. Da alle verbleibenden Grammatiken mit H_0 **kompatibel** sind, wird immer H_0 beibehalten; H_i kann niemals gewählt werden.

Obermengenproblem 2

- Das Problem ist, dass der Lerner niemals merkt, dass es sich in einer **Obermenge** (H_0) der Zielmenge (H_i) befindet.
- Der Lerner erhält keine Information über Elemente, die nicht in der Zielmenge sind.
- Dieses Problem tritt bei Funktionslernbarkeit nicht auf, da dort kein vergleichbares Konzept der Obermenge existiert.
- Alle Funktionen in F sind auf derselben Domäne definiert (der Basiskomponente).
- Wenn das Kind lange genug wartet, dann erhält es für jede beliebige TS irgendwann Information darüber, auf welche Kette die TS abgebildet wird und kann überprüfen, ob seine Transformationskomponente dasselbe tut.

Speicherkapazität

- Erinnerung:
 1. Lernen von **unbeschränkten** TfGen durch Aufzählung von Paaren $\langle b, s \rangle$ ist möglich.
 2. Lernen durch Aufzählung verlangt, dass LP zu jedem Zeitpunkt t eine Grammatik wählt, die mit **allen** bis zu t präsentierten Daten kompatibel ist.
- Problem:
 1. Punkt 2. scheint kein realistisches Szenario für Kindern, die Sprache lernen.
 2. Aber wenn die Daten, die zur Verfügung stehen, beschränkt werden, dann kann der Lernerfolg nicht garantiert werden.
 3. Daraus folgt, dass Speicherbeschränkungen ein grundlegendes Problem für die Lernbarkeitstheorie sind.

Speicherkapazität 2

- Vorschlag:
 1. LP wird geändert, so dass die Grammatik, die zum Zeitpunkt $t + 1$ gewählt wird, nur abhängt
 - (a) von der Grammatik, die zu t gewählt wurde, und
 - (b) von den Daten, die zu $t + 1$ präsentiert wurden.
 2. Um Lernbarkeit zu garantieren, müssen dann weitere Beschränkungen auf den möglichen TfGen in Kauf genommen werden.

Input

- Vorher: Jedes Paar $\langle b, s \rangle$ kommt wenigstens einmal in jeder Informationssequenz I vor.
- Jetzt: Jedes Paar $\langle b, s \rangle$ in I hat eine festgelegte **Wahrscheinlichkeit** > 0 , in I aufzutauchen.
- Idee:
 1. Wenn jedes Datum eine Wahrscheinlichkeit > 0 besitzt zu einem beliebigen Zeitpunkt t aufzutau-chen, dann besteht immer die Möglichkeit, dass im nächsten Moment nützliche Daten präsentiert werden.
 2. Das Kind muss dann nicht alle bisher präsentier-ten Daten gespeichert haben.

Lernbarkeitskriterium

- Es muss einen Zeitpunkt t geben, so dass
 1. der Lerner zu t die korrekte Grammatik gewählt hat und
 2. sich die Grammatik des Lerners nach t nicht mehr ändert.
- Die Grammatik ist korrekt, wenn ihre Transformationskomponente dieselben Paare $\langle b, s \rangle$ definiert wie die Transformationskomponente der Zielgrammatik.
- **Konvention** : Wenn eine Transformationskompo-nente K aus einer TS b die Kette s erzeugt, dann schreiben Wexler & Culicover (1980) auch $*K(b) = s$ (d.h., das Präfix $*$ vor P bezeichnet die Kette eines PMs P).

Lernprozedur

- Zu jedem Zeitpunkt t , zu dem ein Paar $\langle b, s \rangle$ präsentiert wird, kann C_t (die Transformationskomponente des Kindes zum Punkt t) entweder
 1. **unverändert** bleiben,
 2. eine alte Transformation **streichen**, oder
 3. eine neue Transformation **raten**.
- Wenn
 1. die Kette s' , auf die C_t b abbildet ($*C_t(b) = s'$) gleich der Kette s ist, dann bleibt C_t unverändert.
 2. die Kette s' , auf die C_t b abbildet ungleich der Kette s ist, dann
 - (a) hat C_t entweder eine falsche Transformation T angewandt und T muss gestrichen werden, oder
 - (b) C_t hat eine richtige Transformation T' nicht angewandt und T' muss erraten werden.
- Die Wahrscheinlichkeit, mit der gestrichen oder geraten wird, verteilt sich auf die Zahl der Transformationen, die dafür in Betracht kommen.

Streichen von Transformationen

- Frage: Welche Transformationen kommen in Betracht gestrichen zu werden?
- Angenommen
 1. zum Punkt t wird das Paar $\langle b, s \rangle$ präsentiert,
 2. $*C_t(b) = s' \neq s = *A_t(b)$ und
 3. $\{T_1, T_2, \dots, T_n\}$ sind von C_t angewandt worden, um s' zu generieren.
- Dann kommt jedes $T_i \in \{T_1, T_2, \dots, T_n\}$ in Betracht gestrichen zu werden.
- Idee:
 1. Jede der beteiligten Transformationen könnte den Fehler hervorgerufen haben.
 2. Daher wird jede der Transformationen mit gleicher Wahrscheinlichkeit gestrichen.

Raten von Transformationen

- Frage: Welche Transformationen kommen in Betracht geraten zu werden?
- Angenommen
 1. zum Zeitpunkt t wird das Paar $\langle b, s \rangle$ präsentiert,
 2. $*C_t(b) = s' \neq s = *A(b)$ und
 3. $M = \{T_i \mid \text{Wenn } T_i \text{ zu } C_t \text{ hinzugefügt würde, dann würde } C_t \text{ } s \text{ anstatt } s' \text{ aus } b \text{ ableiten}\}$
- Dann kommt jedes $T_i \in M$ als neue Hypothese in Betracht, wenn gilt: T_i involviert Symbole auf der **obersten Ebene** von b .
- Terminologie:
 1. Der Bereich zwischen zwei zyklischen (S -)Knoten ist eine Ebene.
 2. Wenn S_i S_{i+1} dominiert, dann ist der Bereich zwischen S_i und S_{i+1} die S_i -Ebene.

Raten von Transformationen 2

- Beachte: Es ist **nicht** notwendig, dass Transformationen als Hypothesen in Betracht gezogen werden, die auf **eingebetteten** Ebenen $S_{i>0}$ applizieren:
 1. Dies erhöht nur den rechnerischen Aufwand.
 2. Das Fehlen solcher Transformationen verursacht auch immer Fehler in PMn, bei denen diese Ebenen die höchste Ebene darstellen, weswegen diese Transformationen dort geraten werden können.
- Begründung von 2.:
 1. Sei S_0 die oberste Ebene von b .
 2. Angenommen T müsste von C_t auf $S_{i>0}$ in b angewandt werden, um s zu erzeugen.
 3. Dann gibt es b' , so dass $b' = b$, nur dass b' nicht die Ebenen $S_{0 \leq j < i}$ besitzt.
 4. Da $T \notin C_t$, kann T nicht auf b' angewandt werden, was zu einem Fehler führt: $*C(b') \neq *A(b')$.

Raten von Transformationen 3

5. Bei Präsentation des Paares $\langle b', s'' \rangle$ entsteht also ein Fehler, so dass bei Eingabe $\langle b', s'' \rangle$ die Hypothese T auf der Ebene S_0 von b' aufgestellt werden kann.
- Einschränkung:
 1. Dies gilt nur, wenn die fehlende Transformation T nicht **ketteninvariant** applizieren kann.
 2. Kann sie das, dann entsteht auf Ebene S_0 von b' **kein** Fehler und T wird nicht bei b' geraten.
 3. Allerdings kann T auf der Ebene S_i von b die **Struktur** von b verändern (auch wenn die Kette gleich bleibt).
 4. Wenn die Veränderung nicht stattfindet, kann dies schließlich zu einem (erkennbaren) Fehler in b auf einer Ebene $< i$ führen.
 5. W&C nehmen daher an: Es darf keine Transformationen geben, die ketteninvariant applizieren.

Streichen von Transformationen 2

- Frage: Kann das Argument für Beschränkung auf der S_0 -Ebene beim Raten nicht auch angewandt werden auf das Streichen von Transformationen?
 1. Angenommen, eine falsche Transformation T appliziert auf Ebene S_i von b .
 2. Dann gibt es einen PM b' , welcher derjenige Teilbaum von b ist, der von S_i dominiert wird (d.h., die S_i -Ebene von $b =$ die S_0 -Ebene von b').
 3. Dann appliziert T auch auf der S_0 -Ebene von b' und verursacht dort einen Fehler.
 4. T kann also auf der S_0 -Ebene von b' gestrichen werden.
 5. Dann würde es genügen, Transformationen zur Streichung zulassen, die auf einer S_0 -Ebene applizieren.

Streichen von Transformationen 3

- Problem:
 1. Inkorrekte T kann die korrekte Kette für b' ableiten, gleichzeitig aber die falsche Struktur.
 2. Dies kann passieren, auch wenn T **nicht** ketteninvariant appliziert: C_t und A können b auf Ebene S_i transformieren und dabei die gleichen Ketten unter unterschiedlichen Strukturen generieren.
 3. Dann gäbe es keinen erkennbaren Fehler auf b' und es würde keine Transformation gestrichen.
 4. Der Fehler würde erst erkennbar auf der obersten Ebene S_0 von b .
 5. Verursacht wurde der Fehler aber durch eine Transformation T , die auf S_i appliziert, **nicht** auf S_0 .
 6. Daher reicht es nicht, sich auf Transformationen zu beschränken, die auf S_0 applizieren.

Nichtdeterminismus

- Problem: Da C_t nichtdeterministisch sein kann, sind manche Derivationen nicht definiert, wenn es nur obligatorische Transformationen gibt.
- Ausweg:
 1. Angenommen, C_t ist nicht-deterministisch auf Input $d = \langle b, s \rangle$ (d.h., es gibt nicht nur eine Kette, die C_t mit b assoziiert).
 2. Dann gilt: $C_t(b)$ kann nicht berechnet werden; es wird so getan, als ob C_t auf d einen Fehler macht.
 3. Die Transformationen, die den Nichtdeterminismus verursachen, sollen dann alle für Streichung in Betracht kommen.
- Konsequenz:
 1. Die T_i , die potentiell gestrichen werden, sind nicht nur die T_i , die auf b angewandt wurden, sondern
 2. alle T_i , die an irgendeinem Punkt der Derivation **hätten** angewandt werden können.

Zusammenfassung

- Zu Beginn (Zeitpunkt 0) enthält die Lernerkomponente C_0 keine Transformationen.
- Angenommen, zum Zeitpunkt t wird das Paar $\langle b, s \rangle$ präsentiert.
 1. Wenn $*C_t(b) = s$, dann $C_{t+1} = C_t$.
 2. Wenn $*C_t(b) = s' \neq s$, dann
 - (a) rate ein T_i aus der Menge der in Frage kommenden Transformationen, oder
 - (b) streiche ein T_i aus der Menge der in Frage kommenden Transformationen.
- Lernbarkeitskriterium: Die Wahrscheinlichkeit, dass für alle b , die von B erzeugt werden, 1. und 2. gelten, kann mit wachsender Zeit beliebig nahe an 1 herangeführt werden:
 1. $*C_t(b) = *A(b)$ und
 2. für alle $\tau > t$: $C_\tau = C_t$.

Konvergenz

- Ziel: Zeigen, dass LP **konvergiert**, d.h., dass LP tatsächlich das Lernbarkeitskriterium erfüllt. Der Beweis wird nur skizziert.
- Erinnerung:
 1. LP rät oder streicht Transformationen nur, wenn ein erkennbarer Fehler auftritt.
 2. Jedes Paar $\langle b, s \rangle$ wird zu jedem Zeitpunkt t mit Wahrscheinlichkeit > 0 präsentiert, so dass das Kind seine Fehler immer erkennen kann.
 3. Nur wenn der Lerner Fehler erkennt, kann er sich korrigieren.
 4. Und nur wenn sich der Lerner korrigiert, kann er schließlich die korrekte Grammatik finden.

Konvergenz 2

- Problematisches Szenario 1:
 1. Die Wahrscheinlichkeiten, mit denen Paare $d_i = \langle b_i, s_i \rangle$ präsentiert werden, sind über die d_i verteilt ($\sum_{i=1}^{\infty} \text{Prob}(d_i) = 1$).
 2. Angenommen, erkennbare Fehler würden nur in Strukturen hohen Grades auftauchen.
 3. Dann folgt, dass sich ihre Auftretenswahrscheinlichkeit 0 annähert, denn es gibt unendlich viele Strukturen hohen Grades.
 4. Konsequenz: Die Auftretenswahrscheinlichkeit der Daten, die erkennbare Fehler enthalten, würde so gering sein, dass Konvergenz (Lernbarkeit) nicht möglich wäre.
 5. Um Konvergenz zu garantieren, muss folgende Eigenschaft gelten:
- Eigenschaft P1: Fehler treten ausreichend oft auf.

Konvergenz 3

- Problematisches Szenario 2:
 1. Angenommen, ein Fehler tritt auf bei Präsentation von $d = \langle b, s \rangle$ zum Zeitpunkt $t - 1$ auf.
 2. Je höher der Grad von b und je länger s (also: je größer d), desto mehr Transformationen T_i können potentiell gestrichen oder geraten werden.
 3. Je größer die Menge der potentiellen T_i , desto geringer die Wahrscheinlichkeit, dass zu t die **korrekte** T_i geraten/gestrichen wird.
 4. Werden die d zu groß, wird die Wahrscheinlichkeit, die korrekte T_i zu raten/streichen u.U. so klein, dass Konvergenz nicht mehr garantiert ist.
 5. Um Konvergenz zu garantieren, muss folgendes gelten:
- Eigenschaft P2: Es gibt Fehlerpaare $\langle b, s \rangle$, für die die Auswahl an zu ratenden und zu streichenden T_i so eingeschränkt ist, dass die Wahrscheinlichkeit, eine korrekte T_i zu wählen, ausreichend hoch ist.

Konvergenz 4

- Behauptung 1: P2 kann garantiert werden, wenn P3 gilt.
- Eigenschaft P3: Wenn Fehler auftreten, dann treten sie auch in PMn von ausreichend niedrigem Grad auf.
- Kommentar: P3 garantiert P2, da ein PM von niedrigem Grad wenig alternative T_i zulässt, womit die Chance hoch genug ist, die richtige T_i zu wählen.
- Behauptung 2:
 1. Die Anzahl der PM niedrigen Grades ist endlich.
 2. Daher gibt es eine untere Schranke, die die Auftretenswahrscheinlichkeit solcher PM von 0 trennt (im Gegensatz zu PMn hohen Grades, vgl. Szenario 1).
 3. Konsequenz: P3 stellt auch sicher, dass P1 gilt.

Konvergenz 5

- Behauptung 3: Um Konvergenz zu garantieren, müssen P1 und P2 für **dieselben** Paare $\langle b, s \rangle$ gelten (siehe P4); das wird von P3 garantiert.
- Eigenschaft P4: Es gibt Fehlerpaare $d = \langle b, s \rangle$, so dass
 1. die Auswahl der T_i für d so eingeschränkt ist, dass die Wahrscheinlichkeit eine korrekte T_i zu wählen ausreichend hoch ist, und
 2. diese d treten ausreichend oft auf.
- Schlussfolgerung: Aus P3 folgen P1, P2, und damit P4. P3 ist also die zentrale Eigenschaft, von der gezeigt werden muss, dass sie gilt.

Konvergenz 6

- Vorweg:
 1. Der Beweis von P3 ist unabhängig von LP , dafür aber abhängig von den Eigenschaften der TfGen.
 2. P3 kann formuliert werden als BDE.
- Boundedness of Minimal Degree of Error (BDE): Für jede Basiskomponente B gibt es eine Zahl U , so dass für jedes A und C und für alle b , die von B generiert werden, gilt: Wenn $*A(b) \neq *C(b)$, dann gibt es ein b' , so dass 1.-3. gelten:
 1. $*A(b') \neq *C(b')$,
 2. b' ist von B generiert,
 3. b' ist höchstens vom Grad U .

Konvergenz 7

- Skizze der Ableitung von BDE:
 1. Erinnerung:
 - (a) Rekursion in B ist beschränkt auf S -Knoten.
 - (b) Zwei S -Knoten umfassen eine Ebene.
 2. Jeder PM kann in Ebenen zerlegt werden, wobei jede Ebene in Tiefe und Breite beschränkt ist.
 3. Diese Beschränkung der Ebenen folgt
 - (a) aus der Annahme 1a und
 - (b) daraus, dass B aus endlichen Regeln besteht.
 4. Es gilt das **Binärprinzip** (BP): Eine Transformation kann ausschließlich Symbole auf der höchsten S_i -Ebene involvieren, plus Symbole auf der nächst tieferen S_{i+1} -Ebene.
 5. Wenn ein Fehler auf der höchsten Ebene eines PMs b entsteht, dann wegen BP auch in PMn, die dadurch aus b abgeleitet sind, dass man alle Ebenen von b außer den beiden obersten weglässt.
 6. Da Ebenen in Tiefe und Breite beschränkt sind, folgt, dass Fehler, wenn sie überhaupt auftreten, auch in PMn beschränkten Grades auftreten: BDE.

Konvergenz 8

- Beachte (vgl. W&C 1981, 110): Aus dem, was in 5. gesagt wurde, scheint wegen BP sogar zu folgen, dass $U = 2$, d.h., dass wenn Fehler auftreten, dann treten Sie in Strukturen des Grades 2 auf.
- Tatsächlich ist die Sache aber komplizierter, da T_i , die auf tieferen Zyklen applizieren, Einfluss haben können auf T_j , die auf höheren Zyklen applizieren.
- Damit erweitert sich der Bereich U . Ohne weitere Annahmen scheint es nicht möglich zu sein, ein U für ein bestimmtes B zu fixieren.
- In Kapitel 4 wird gezeigt, wie man U auf 2 fixieren kann.

Konvergenz 9

- Die Transformationskomponenten A und C sind **moderat äquivalent** bzgl. einer Basiskomponente B ($C \cong_M A$) genau dann, wenn für alle b , die von B generiert werden, gilt: $*C(b) = *A(b)$.
- **Lernbarkeitsresultat** (Konvergenz): Die Wahrscheinlichkeit, dass der Lerner die korrekte Transformationskomponente findet (und nicht wieder verliert), kann mit wachsender Zeitspanne beliebig nahe an 1 herangeführt werden.
- Ausgangspunkt:
 1. Seien $A = \{T_1, T_2, \dots, T_n\}$, $C_t = \{T'_1, T'_2, \dots, T'_m\}$.
 2. Angenommen alle T_i von C_t sind falsch.
 3. Der Weg von C_t nach A benötigt mindestens $m + n$ Schritte (nur korrekte T_i werden geraten, keine korrekten gestrichen, es gibt nur Fehlerdaten).
 4. Zu zeigen: Die Wahrscheinlichkeit, von C_t nach A in weniger als k Schritten zu wechseln, ist größer als eine untere Schranke > 0 .

Konvergenz 10

- Beweisskizze:
 1. BDE: Für jeden der $m + n$ Schritte gilt, dass es ausreichend wahrscheinlich ist, dass
 - (a) ein Fehler durch $d = \langle b, s \rangle$ erkennbar wird, und
 - (b) bs Grad ausreichend niedrig ist, um die Wahl einer korrekten T_i wahrscheinlich zu machen.
 2. Zu zeigen:
 - (a) Eigenschaft P5: Es gibt PM von geringem Grad, so dass die Menge der potentiell zu wählenden T_i wenigstens eine korrekte T_i enthält.
 - (b) Es gibt eine obere Schranke für die Zahl der T_i in C_t , die zu allen Zeitpunkten t konstant ist.
- Idee von 2b:
 1. Die Wahrscheinlichkeit in **mehr** als k Schritten von C_t zu A zu wechseln ist nach unten beschränkt: C_t kann nur beschränkte Zahl **falscher** T_i enthalten (C_t ist nicht beliebig schlecht).
 2. Kombinationen aus Raten und Streichen, die nicht konvergieren, sind zu unwahrscheinlich.

Konvergenz 11

- 2a (P5):
 1. Es wird das Fehlerpaar $d = \langle b, s \rangle$ präsentiert.
 2. BDE garantiert, dass es solche Fehlerpaare für $\text{Grad}(b) \leq U$ gibt.
 3. Fall 1: C_t wendet falsche T_i auf b an.
 - (a) Da der Fehler bei d auftritt, steht T_i zur Streichung zur Verfügung.
 - (b) Und da T_i falsch war, ist T_i eine korrekte Wahl.
 4. Fall 2: C_t wendet korrekte T_i nicht auf b an.
 - (a) Betrachte die tiefste Ebene von b , auf der eine korrekte T_i von A appliziert, und $T_i \notin C$.
 - (b) Sei dies die oberste Ebene des Teilbaums b' .
 - (c) b' kann dann auch von b isoliert auftreten (Ebenen sind immer von S dominiert).
 - (d) An der Top-Ebene von b' kann in einem solchen Fall dann T_i geraten werden.
 - (e) Da b' ohne T_i falsch ist, folgt, dass T_i eine korrekte Wahl ist.
 - (f) Da $\text{Grad}(b) \leq U$, folgt dass $\text{Grad}(b') \leq U$.

Konvergenz 12

- Beachte: Für die Argumentation war notwendig, dass der Zyklus gilt.
 1. Gilt der Zyklus nicht, dann könnte zuerst eine T_j auf der obersten Ebene von b angewandt werden und dadurch Anwendung von einer T_i auf einer tieferen Ebene (Top-Ebene von b') auslösen.
 2. Würde T_i dann nicht angewandt, entstünde ein Fehler.
 3. Aber T_i kann in so einem Fall
 - (a) nicht auf b geraten werden, da T_i nicht auf der obersten Ebene von b angewandt würde, sondern auf b' (siehe Definition von "Raten").
 - (b) nicht auf b' geraten werden, da auf b' noch kein Fehler auftritt.

Exkurs: Empirische Evidenz für Zyklus

- Empirische Evidenz für den Zyklus kommt z.B. von W-Inselverletzungen wie in (1).
 - Annahmen:
 1. Die W-Phrase *wem* in (1) muss vom eingebetteten \bar{S} in den übergeordneten \bar{S} bewegt werden.
 2. Bewegung muss **sukzessiv zyklisch**, d.h., via den Rand des eingebetteten \bar{S} erfolgen.
 3. Dieser Rand ist in (1) aber schon von der W-Phrase *wie* besetzt.
- (1) *Wem_i weißt [_S du nicht [_{\bar{S}} wie [_S du _i gefallen könntest]]] ?
- Zyklus notwendig, um folgende Derivation auszuschließen (Schritt 3. ist blockiert):
 1. *Wem* bewegt sich zu Rand des eingebetteten \bar{S} .
 2. *Wem* bewegt sich zu Rand des übergeordneten \bar{S} .
 3. *Wie* bewegt sich zu Rand des eingebetteten \bar{S} .

Konvergenz 13

- 2b (Obere Schranke für Zahl der T_i in C_t):
 1. Jede T_i in C_t wurde an einem Punkt geraten.
 2. Dabei musste die strukturelle Beschreibung von T_i auf einen **zugänglichen** Teil des PMs passen.
 3. Wegen Determinismus gibt es immer nur eine T_i , die auf eine strukturelle Beschreibung passt.
 4. Wegen bestimmter Beschränkungen auf TfGen kann man zeigen, dass es eine beschränkte Zahl an zugänglichen Strukturen gibt.
 5. Es folgt, dass C_t nur eine beschränkte, über alle t konstante Zahl (falscher) T_i raten (und damit enthalten) kann.
- Eine der Beschränkungen auf TfGen, von denen in 4. die Rede ist, ist das Binärprinzip.

Verwandte des Binärprinzips

- Das Binärprinzip ist sehr ähnlich zu den linguistisch motivierten Prinzipien
 1. **Subjazen** (siehe Chomsky 1973) und
 2. **Phase Impenetrability Condition** (siehe Chomsky 2001)
- Binärprinzip (BP): Eine Transformation kann ausschließlich Symbole auf der höchsten S_i -Ebene involvieren, plus Symbole auf der nächst tieferen S_{i+1} -Ebene.
- Subjazen (vereinfacht): Keine Transformation kann zwei Symbole involvieren, die durch zwei Knoten der Kategorie S oder NP voneinander getrennt sind.
- Phase Impenetrability Condition (PIC, vereinfacht): Keine Transformation kann zwei Symbole X , Y involvieren, wobei Y innerhalb (nicht am Rand) einer \overline{S} -Domäne ist und X außerhalb von \overline{S} .

Verwandte des Binärprinzips 2

- Beobachtung:

1. Subjanz und PIC schließen beide die W-Inselerletzungen in (3) aus; BP nicht.
2. Nur Subjanz schließt die Verletzungen komplexer NPn in (2) aus.
3. Alle drei erlauben lange W-Bewegung in (4).

- (2)
- a. *Wen_i hast [_S du [_{NP} ein Gerücht gehört [_S _i dass [_S Fritz _i mag]]]]
 - b. *der Hut, den_i [_S ich [_{NP} die Behauptung gehört habe [_S _i dass [_S Fritz _i trägt]]]]
- (3)
- a. *Wem_i weißt [_S du nicht [_S wie [_S du _i gefallen könntest]]] ?
 - b. *Wie_i möchtest [_S du wissen [_S wen [_S Fritz _i überzeugt hat]]] ?
- (4)
- a. Wem_i glaubst [_S du [_S _i könntest [_S du _i gefallen]]] ?
 - b. Wie_i glaubst [_S du [_S _i könntest [_S du sie _i überreden]]] ?

Literatur

- Cantor, Georg (1890): 'Über eine elementare Frage der Mannigfaltigkeitslehre', *Deutsche Mathematiker-Vereinigung* **1**, 75–78.
- Chomsky, Noam (1973): Conditions on Transformations. In: S. Anderson & P. Kiparsky, eds, *A Festschrift for Morris Halle*. Academic Press, New York, pp. 232–286.
- Chomsky, Noam (2001): Derivation by Phase. In: M. Kenstowicz, ed., *Ken Hale. A Life in Language*. MIT Press, Cambridge, Massachusetts, pp. 1–52.
- Wexler, Ken & Peter Culicover (1980): *Formal Principles of Language Acquisition*. MIT Press, Cambridge, Massachusetts.