

Hisatsugu Kitahara (1997) – Shortest Derivation and Timing Effects

Zentrale Frage: Ökonomiebeschränkungen minimieren die Anzahl der Operationen in Derivationen – aber was zählt als eine solche Operation, im besonderen für die Shortest Derivation Condition (SDC)?

→ es wird eine Veränderung der SDC vorgenommen, die auf elementare Operationen Bezug nimmt; damit lassen sich Effekte wie die strikte Zyklizität und Timing Effekte wie Procrastinate durch eine Beschränkung ableiten

1. Elementare Operationen

Eine syntaktische Baumstruktur entsteht, indem lexikalische Elemente (LI), die zuvor aus dem Lexikon in die Numeration (N) gebracht wurden, durch strukturaufbauende Prozeduren miteinander verknüpft werden (Repräsentation R), um ein Paar (π, λ) zu bilden. Diese Prozeduren sind Merge, Move und Erase, die sich jedoch in noch elementarere Operationen zerlegen lassen.

a) Merge (Verkettung)

bisher: Angewandt auf 2 Objekte α und β , bildet Merge ein neues Objekt K, indem α und β konkateniert werden.

- appliziert nur an der Wurzel (= zyklisch, erweitert den Baum)
- operiert an separaten Phrasenmarkern
- eine elementare Operation: Konkatenation

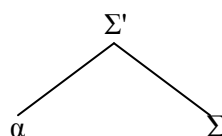
revidiert: Angewandt auf 2 Objekte α und Σ , das β dominiert, bildet Merge ein neues Objekt Σ' , indem α und β konkateniert werden. Wenn diese Operation nicht zyklisch ist, ersetzt sie β in Σ durch $L = \{\gamma, \{\alpha, \beta\}\}$.

- zyklische und nicht-zyklische (innerhalb des Dominanzbereiches der Wurzel, nicht strukturerweiternd) Anwendungen von Merge möglich
- elementare Operationen: Konkatenation und Ersetzung

- wenn $\Sigma = \beta$ (reflexive) → zyklische Applikation, nur Konkatenation

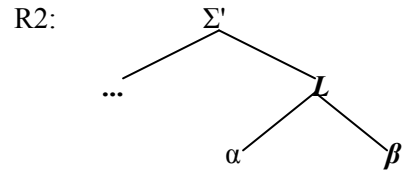
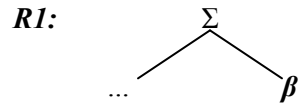
N: α, Σ

R:



- wenn $\Sigma \neq \beta \rightarrow$ nicht-zyklische Applikation, Konkatination und Ersetzung

N: α, β, \dots



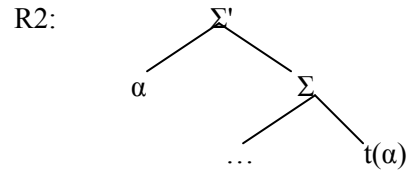
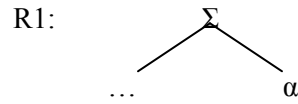
b) Move (Bewegung)

Angewandt auf eine Kategorie Σ , die K und α dominiert, bildet Move Σ' , indem es K und α konkateniert. Diese Operation ersetzt, wenn sie nicht-zyklisch appliziert, K in Σ durch $L = \{\gamma, \{\alpha, K\}\}$.

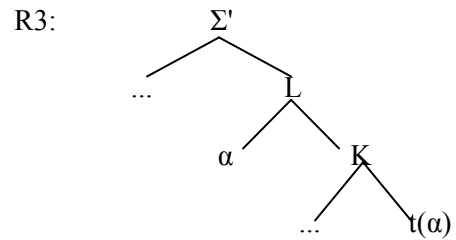
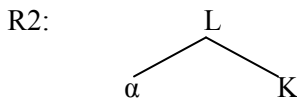
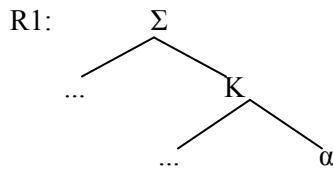
- operiert nur an einem Phrasenmarker
- kann an der Wurzel oder in deren Dominanzbereich applizieren

- wenn $K = \Sigma \rightarrow$ zyklische Applikation, Konkatination

N: α, \dots

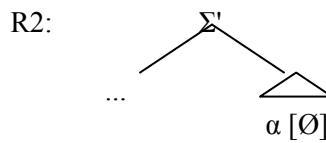
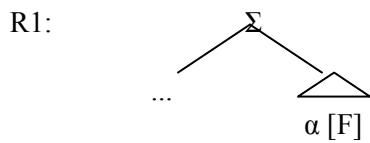


- wenn $K \neq \Sigma \rightarrow$ nicht-zyklische Applikation, Konkatination und Ersetzung



c) Erase (Ersetzung / Tilgung), Chomsky (1995)

Angewandt auf die Kategorie Σ , die α [F] dominiert, bildet Erase Σ' , indem es F in α durch ein leeres Element \emptyset ersetzt.



\rightarrow nicht-zyklische Anwendung von Move aus unabhängigen Gründen motiviert, z.B. Kopfbewegung; nicht-zyklische Anwendung von Merge nicht motiviert – die strikte Zyklizität ist eigentlich Teil der Definition von Merge (bei Chomsky)

\rightarrow aber: zyklische Operationen gegenüber nicht-zyklischen präferiert

Wie kann man diese Präferenz ableiten?

Komplexität:

- zyklische Applikation von Merge: Konkatination
 - zyklische Applikation von Move: Konkatination
 - (nicht-zyklische) Applikation von Erase: Ersetzung
 - nicht-zyklische Applikation von Merge: Konkatination + Ersetzung
 - nicht-zyklische Applikation von Move: Konkatination + Ersetzung
- } einfache Operationen
- } komplexe Operationen

Kitaharas Vorschlag:

Chomskys Ökonomiebeschränkung Shortest Derivation Condition (SDC, 1991), die besagt, dass die Anzahl der Operationen, die für die Konvergenz einer Derivation notwendig sind, minimiert werden sollen, wird geringfügig revidiert:

SDC: Minimiere die Anzahl der elementaren Operationen (Konkatenation und Ersetzung), die für die Konvergenz einer Derivation notwendig sind.

→ Obwohl Merge und Move sowohl zyklisch als auch nicht-zyklisch applizieren können, stellt die SDC sicher, dass die zyklischen Operationen bevorzugt werden, da sie weniger elementare Operationen benötigen.

2. Strict Cyclicity: frühere Analysen und SDC-Analyse

Problem: Überquerung von Inseln verhindern, 3 Analysen von Chomsky

*How₂ did John wonder [what₁ Mary fixed t₁ t₂] ?

nicht-zyklische Derivation:

1. a) *_{[CP how [C' C_{wh} [IP John wondered [CP C_{wh} [IP Mary fixed what t(how)]]]]]}

1. b) *_{[CP how [C' C_{wh} [IP John wondered [CP what [C' C_{wh} [IP Mary fixed t(what) t(how)]]]]]]]}

zyklische Derivation:

2. a) *_{[CP what [C' C_{wh} [IP Mary fixed t(what) how]]]}

2. b) *_{[CP how [C' C_{wh} [IP John wondered [CP what [C' C_{wh} [IP Mary fixed t(what) t(how)]]]]]]]}

Extension Condition Analysis, Chomsky (1993)

- *Extension Condition (EC)*: Generalized Transformation muss die gesamte Phrasenstruktur erweitern. d.h. die Phrasenstruktur „wächst nach oben“, eine neue Wurzel entsteht

- *Shortest Move Condition (SMC)*: Wenn 2 konvergente Derivationen D1 und D2 gegeben sind, die dieselbe Anzahl von Derivationsschritten involvieren, blockiert D1 D2, wenn ihre Links kürzer sind.

1. a) Bewegung von *how* in die Matrix-SpecC-Position erfüllt beide Beschränkungen (wenn SpecC im eingebetteten Satz noch nicht vorhanden ist)
1. b) Bewegung von *what* erfüllt SMC, aber verletzt EC
2. a) Bewegung von *what* erfüllt EC und SMC
2. b) Bewegung von *how* in Matrix-SpecC-Position erfüllt die EC, aber verletzt die SMC, da SpecC im eingebetteten Satz übergangen wurde

→ EC und SMC gebraucht, um beide Derivationen auszuschließen

A Minimal Link Condition Analysis (Chomsky 1994)

- *Merge* (nur zyklisch) und *Move* wie oben definiert angenommen

- SMC durch die *Minimal Link Condition* ersetzt:

Minimiere die Ketten-Links. → Move α muss α zur nächstgelegenen Zielposition bewegen

1. a) Bewegung von *how* verletzt die MLC – nächstes Ziel wäre die eingebettete SpecC-Position
2. a) Bewegung von *what* erfüllt MLC
2. b) Bewegung von *how* verletzt die MLC – nächstes Ziel wäre die eingebettete SpecC-Position

Die MLC-Analyse erlaubt aber trotzdem einige nicht-zyklische Derivationen, die mit Hilfe der EC und der Subject Condition ausgeschlossen werden müssen.

3. ?? Who₂ did you say that [pictures of t₂]₁ were stolen t₁?

→ zykl. und nicht-zykl. Derivation werden nicht von der MLC ausgeschlossen; die zyklische kann mit Hilfe der **Subject Condition** ausgeschlossen werden: Man kann nur aus Komplementen herausbewegen. Die nicht-zyklische Derivation verletzt aber auch diese nicht.

Strong Feature Condition Analysis (Chomsky 1995)

Strong Feature Condition: Angenommen eine Derivation D hat Σ gebildet, das α mit einem starken Merkmal F enthält. D bricht zusammen, wenn α in einer Kategorie enthalten ist, deren Kopf nicht α ist. → unter Schwesterknoten sollte immer derjenige seine Merkmale auf den Mutterknoten projizieren, der starke Merkmale hat/Kategorien mit starken Merkmalen sind Köpfe v. Konstituenten



2. b) wondered [_{CP} C_{wh} [_{IP} Mary fixed what t(how)]]

3. [_{CP} that [_{TP} were stolen [_α pictures of who]]] (nicht-zyklisch)

1. b) wenn *wonder* und CP verkettet werden, sollte der Kopf C sein, da C ein starkes Merkmal [wh] trägt, aber die resultierende Konstituente ist eine VP

3. wenn *that* und die TP verkettet werden, sollte T der Kopf sein, weil er ein starkes D-Merkmal hat, aber die resultierende Konstituente ist eine CP

→ nicht-zyklische Derivationen ausgeschlossen

Um auch die zyklischen ausschließen zu können, revidiert Chomsky die **MLC**:

H(K) zieht α nur dann an, wenn es kein β gibt, das näher an α steht, so dass H(K) auch β anziehen könnte.

„Näher“ wird wie folgt definiert:

β ist näher an H(K) als α , wenn β α c-kommandiert und β sich nicht in der minimalen Domäne von CH befindet, wobei CH die Kette mit dem Kopf γ ist, und γ an H(K) adjungiert ist.

1. b) *how* wurde über *what* bewegt, aber *what* könnte das starke wh-Merkmal des C-Kopfes auch überprüfen und ist diesem näher

3. MLC erfüllt, man benötigt die Subject Condition um diese Derivation auszuschließen

Fazit: die erste Analyse benötigt die EC und die SDC, die zweite ersetzt die SDC durch die MLC, muss allerdings, um einige ungrammatische Fälle auszuschließen, auf die SC zurückgreifen. Die dritte Analyse benötigt sogar drei Beschränkungen, um die Daten abzuleiten: MLC, SC, SFC.

Kitaharas Shortest Derivation Condition Analysis

SDC: Minimiere die Anzahl der elementaren Operationen (Konkatenation und Ersetzung), die für die Konvergenz einer Derivation notwendig sind.

Die SDC schließt die nicht-zyklischen Derivationen 1 und 3 von vornherein aus, da diese die zwei elementaren Operationen Konkatenation und Ersetzung benötigen, während die zyklischen nur Konkatenation brauchen. Um 2 auszuschließen, wird auf die MLC zurückgegriffen und für 3 auf die SC. → auch 3 Beschränkungen gebraucht

3. Die Ableitung von Procrastinate

Mit der SDC können der Zeitpunkt von Verbbewegung, Object Shift und Expletiveinsetzung abgeleitet werden, die zuvor von Chomsky unter Bezug auf die Beschränkung Procrastinate erklärt wurden. Laut Kitahara kann Procrastinate deshalb aus dem Minimalistischen Programm entfernt werden.

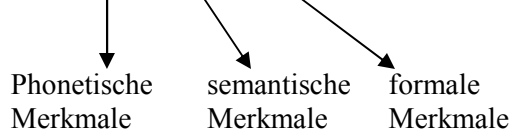
Procrastinate (Chomsky 1993): Minimiere die Anzahl der overt Operations, die für die Konvergenz einer Derivation notwendig sind. → covert Bewegung ist „besser“ als overte

Für die Ableitung der Daten durch die SDC spielt vor allem die Operation Erase eine wichtige Rolle. Erase appliziert nur, wenn ein Merkmal F ersetzt werden *muss*, um Konvergenz zu erreichen. Dazu müssen wir uns ansehen, welche Merkmale Erase unter welchen Bedingungen ersetzt.

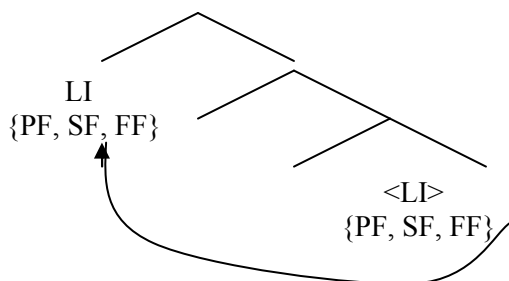
Merkmale und Bewegung

Ein lexikalisches Element LI ist ein Komplex von (mindestens) 3 Merkmalsbündeln:

LI {PF(LI), SF(LI), FF(LI)}



Die overte Bewegung eines LIs (auf der S-Struktur) hinterlässt eine Spur t(LI) als vollständige Kopie von LI mit allen Merkmalen. Es entsteht eine Kette CH (LI, t(LI)).



- das FF-Merkmal, das die Bewegung auslöst, wird überprüft und getilgt, d.h. durch Erase \emptyset mit ersetzt
- PF und SF bleiben erhalten
- angenommen, PF und SF können nur einmal interpretiert werden, dann muss nach der Bewegung SF von LI oder t(LI) auf LF getilgt werden durch Erase, um sie nicht doppelt zu interpretieren (Tilgung von PF ignoriert)
- per Annahme sind es die Merkmale von LI, die getilgt werden
- die Interpretation findet dort statt, wo das LI seine Thetarolle erhält

Bei coverter Bewegung auf LF wird hingegen nur FF kopiert, um überprüft und getilgt zu werden. PF wurde vorher schon abgearbeitet (Spell-Out) und SF wurde zurückgelassen, weil diese Merkmale dort interpretiert werden, wo das LI seine Thetarolle erhält.

→ overte Bewegung benötigt deshalb neben den Schritten, die für die Bewegung nötig sind, immer noch eine zusätzliche Ersetzungsoperation, um SF zu tilgen und ist damit immer unökonomischer als covert Bewegung, die diese Tilgung nicht braucht.

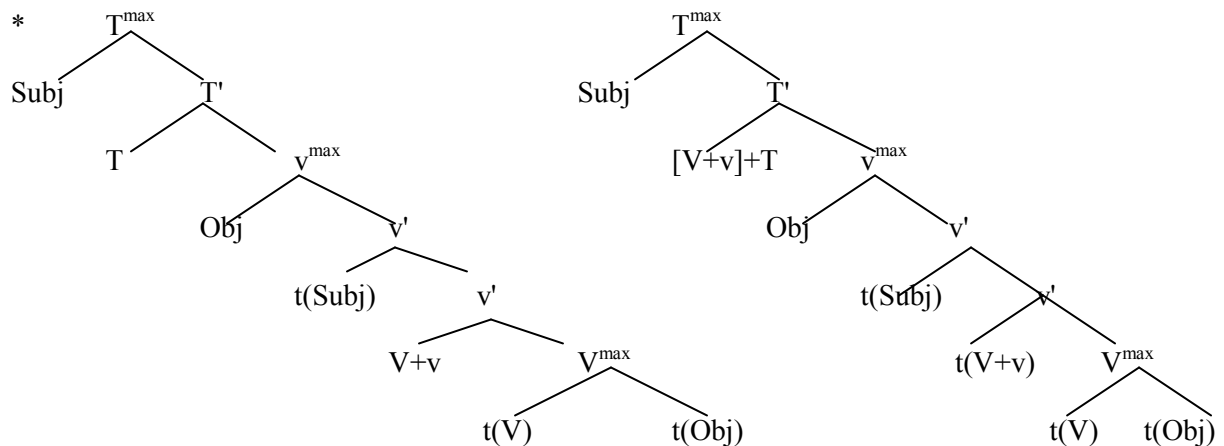
→ Procrastinate ist mit der SDC abgeleitet.

Das Ergebnis entspricht folgender *Generalisierung über Object Shift von Holmberg*:

Overtes Object Shift kann nur mit overter V-nach-T-Bewegung stattfinden.

Da oben gezeigt wurde, dass overte Verbbewegung von Hauptverben im englischen durch die SDC ausgeschlossen ist, kann auch kein overtes Object Shift stattfinden.

6. b) Object Shift mit overter Verbbewegung:



Schlussfolgerung: Wenn eine Sprache overte Verbbewegung hat, sollte sie zumindest optional auch overtes Object Shift zulassen.

Beispiel: Isländisch

7. a) Risarnir átu [ekki t(átu) ríkisstjórnia].
the-giants ate not the-government
7. b) Risarnir átu ríkisstjórnia [ekki t(átu) t(ríkisstjórnia)].
the-giants ate the-government not
- } overte Verbbewegung,
in b) Object Shift

Wie kann man Optionalität in Derivationen ableiten?

In Chomskys Feature Checking Theory ist das nicht möglich, da Bewegungen nicht wahllos stattfinden, sondern durch entsprechende starke Merkmale ausgelöst werden müssen. Dann ist es aber unklar, weshalb in ein und derselben Derivation einmal ein solches Merkmal vorhanden ist und einmal nicht. Ausweg: optionale Bewegungen geschehen aus stilistischen Gründen

7. a) $[_{CP} [_{IP} \text{Subj} [V+v]+T [_{VP} \text{adv } t(\text{subj}) t(v) [_{VP} t(V) \text{Obj}]]]]]$
7. b) $[_{CP} [_{IP} \text{Subj} [V+v]+T [_{VP} \text{Obj } \text{adv } t(\text{subj}) t(v) [_{VP} t(V) t(\text{Obj})]]]]]$

SDC leitet Optionalität ab:

-erfolgt kein Object Shift, bleibt das Objekt in situ und wird erst auf LF bewegt = nicht-zyklische Operation von Move: Konkatenation und Ersetzung

- da bei dieser coverten Bewegung aber nur die Merkmale FF bewegt werden, muss auf LF kein Erase mehr stattfinden

- erfolgt Object Shift, findet zyklische Bewegung des Objekts statt: Konkatenation

- da bei overter Bewegung aber alle Merkmale kopiert werden, muss auf LF Erase stattfinden, um die Merkmale SF zu tilgen: Ersetzung

→ sowohl overte als auch coverte Bewegung benötigen 2 elementare Operationen: Konkatenation und Ersetzung, deshalb ist es optional, welche Derivation gewählt wird

3. Anwendung – Ableitung des Zeitpunkts der Expletiveinsetzung

8. a) $There_1$ seems [t_1 to be someone in the room].
8. b) * $There$ seems [TP someone $_1$ to be t_1 in the room].

Sowohl 8. a) als auch 8. b) erfüllen die SFC, denn das starke D-Merkmal des eingebetteten T wird in a) durch die Einsetzung von *there* und in b) durch die Bewegung von *someone* überprüft. Das starke D-Merkmal des Matrix-T wird dann überprüft, wenn sich *there* bzw. *someone* aus der eingebetteten SpecT-Position in die Matrix-SpecT-Position bewegt.

- Chomskys Lösung: Procrastinate leitet ab, dass erst *there* eingesetzt wird, da overte Bewegung von *someone* im eingebetteten Satz „kostspielig“ ist.
- Kitaharas Analyse mit der SDC:
Das Expletivum *there* ist eine phonetisch overte realisierte, aber semantisch leere Kategorie, die ein Kategorienmerkmal D hat, das das starke D-Merkmal von T überprüfen kann.

→ {PF(EXP), FF(EXP)}

Wenn das Expletivum overte bewegt wird, werden wieder alle Merkmale kopiert, aber auf LF muss Erase nicht applizieren, um SF zu tilgen, da das Expletivum von vornherein keine semantischen Merkmale hat (PF bei Spell-Out gestrichen).

Wird aber wie in 8. b) *someone* bewegt, das semantische Merkmale besitzt, müssen diese auf LF wieder durch Erase einmal getilgt werden, um sie nicht doppelt in der Verkettungsposition und der Landestelle nach der Bewegung zu interpretieren.

a) benötigt also nur Konkatenation, während b) Konkatenation und Ersetzung benötigt und a) nach der SDC ökonomischer ist.

4. Fazit:

Kitahara zerlegt die strukturaufbauenden Operationen Merge, Move und Erase in die elementaren Operationen Konkatenation und Ersetzung und stellt die transderivationelle Beschränkung SDC auf, die besagt, dass die Anzahl dieser elementaren Operationen zu minimieren sei. Damit kann er ableiten, warum zyklische Derivationen gegenüber nicht-zyklischen präferiert werden und es gelingt ihm zu zeigen, warum overte Bewegung oft overter Bewegung vorgezogen wird. Dadurch kann er Procrastinate, das besagt, dass LF-Bewegung „billiger“ ist als overte Bewegung, als Beschränkung eliminieren und darüber hinaus noch erklären, warum das so sein muss. Procrastinate an sich war eher eine Festlegung, die die Daten nahe legen, ohne Begründung für den Vorrang der LF-Bewegung.

Besonders hervorzuheben ist, dass Kitahara Optionalität in Derivationen durch die SDC syntaktisch ableiten kann, ohne auf andere Komponenten des Sprachsystems ausweichen zu müssen, wie z. B. die stilistische Ebene. Im Minimalistischen Programm ist es ansonsten nicht möglich Optionalität zu erklären, da man davon ausgeht, dass Bewegungen nur applizieren, wenn es auch einen Auslöser dafür gibt (Merkmalstilgung).

5. Fragen /Diskussionspunkte:

- Prinzip „Merge before Move“ nicht ableitbar, wenn beide zyklisch bzw. nicht-zyklisch applizieren
- keine Argumente für die Annahme, dass Merge auch nicht-zyklisch applizieren kann
- verschiedene Lesarten bei den Bakersätzen abgeleitet: Optionalität auf LF – egal zu welcher Position die wh-Phrase bewegt wird – es handelt sich immer um nicht-zyklische Applikationen von Move
- Bedingung des strikten Zyklus von overter Nicht-Kopf- Bewegung abgeleitet: zyklische Bewegung immer ökonomischer als nicht-zyklische

6. Quellen:

- Kitahara, Hisatsugu (1997): Elementary Operations and Optimal Derivations. MIT Press, Cambridge, Massachusetts.
- Chomsky, Noam (1993): Some Notes on Economy of Derivation and Representation. In: R. Freidin, ed., Principles and Parameters in Generative Grammar. MIT Press, Cambridge, Massachusetts.
- Adger, David (2003): Core Syntax, Oxford University Press.