

Grundlagen der Syntax: Das minimalistische Programm

Martin Haase

Minimalistisches Programm

Noam Chomsky (1992): “A minimalist program for linguistic theory” in:

Kenneth Hale & Samuel Jay Keyser: *The view from building 20: essays in linguistics in honor of Sylvain Bromberger*. Cambridge, MA: MIT Press, 1993.

Zugrunde liegende Strukturen

Liedchen

/ˈliːdxən/ → [ˈliːtç(ə)n]

Zugrunde liegende Strukturen

Liedchen

#'li:d# + #-xen# → /'li:dxən/ → ['li:tçən] → ['li:tçn]

Zugrunde liegende Strukturen

Liedchen

/ˈliːdxən/ → [ˈliːtç(ə)n]

Ableitung

Oberflächenstruktur



Tiefenstruktur

Konzept: Tiefen- und Oberflächenstruktur

1. Transformation
2. Tiefen- und Oberflächenstruktur werden gleichzeitig generiert (also ohne Transformation), jedoch mit *move-a*.
3. keine Unterscheidung von Tiefen- und Oberflächenstruktur

Adäquatheitsebenen von Theorien

1. beobachtungsadäquat
2. erklärungsadäquat
3. psychisch adäquat

Adäquatheitsebenen von Theorien

1. beobachtungsadäquat
2. erklärungsadäquat
3. psychisch adäquat

oft falsch:
psychologisch adäquat (psychologically adequate)

Besonderheit

linguistisch \neq sprachlich, Sprach-
psychologisch \neq psychisch

Adäquatheitsebenen von Theorien

1. beobachtungsadäquat
2. erklärungsadäquat
3. psychisch adäquat

Psychische Adäquatheit von Transformationen?

... von Tiefen- und Oberflächenstruktur?

Psychische Adäquatheit

... von Tiefen- und Oberflächenstruktur?

Evidenz in der Phonologie:

Orthografie

Nonsense-Wörter

Versprecher

Wortspiele: *verlan*

verlan

l'envers → verlan

femme → meuf

fête → teuf

flic → keuf

pute → timpe

keine eindeutige Evidenz für Morphosyntax

keine eindeutige Evidenz für Morphosyntax

Charles Hockett (1954): “Two models of grammatical description”, *Word* 10: 210–234.

2–3 Modelle

IA: Item & Arrangement

IP: Item & Process
(Item in Paradigm)

Typische Frage im Staatsexamen

Ermitteln Sie die morphologische Struktur von:
remerciement
ringraziamento
agradecimiento

sehr „schwache“ Lösung: 0 Punkte

re+merci+e+ment

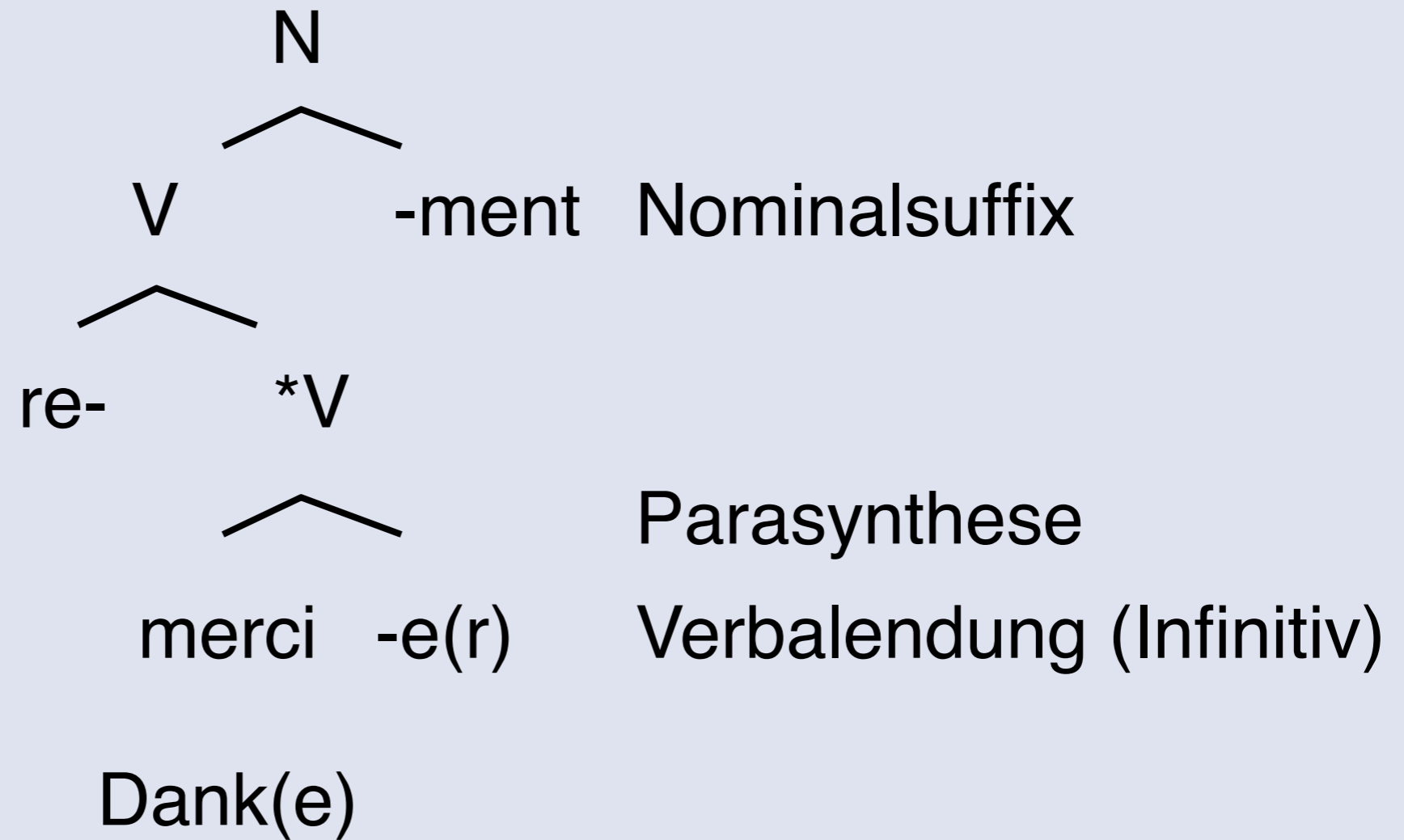
wieder+Dank+Femininum+Adverbialendung

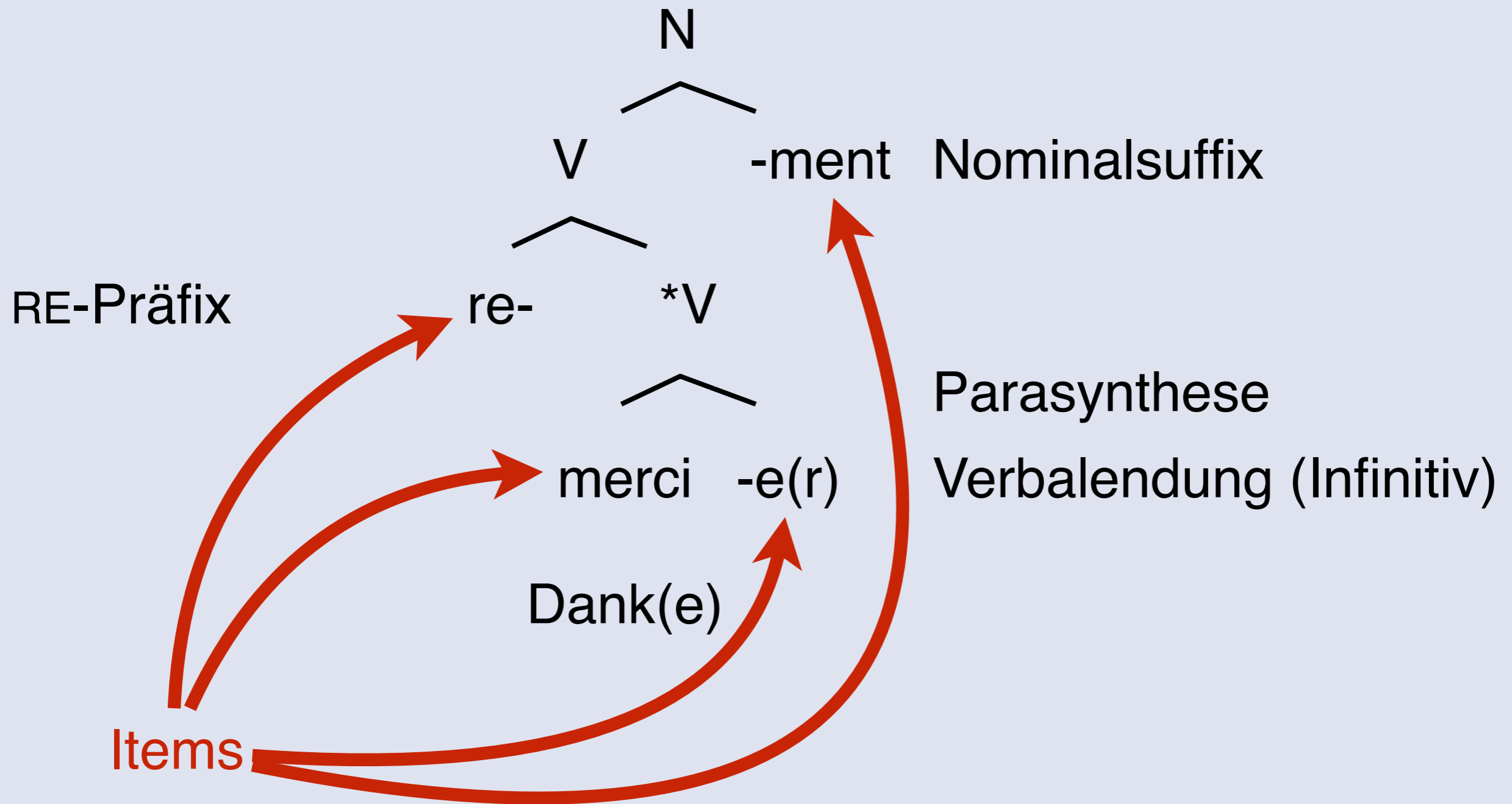
„schwache“ Lösung: 1 Punkt

re+merci+e+ment

wieder+Dank+Fugenelement+Nominalendung

RE-Präfix





merci *Ableitungsbasis (Input)*

*mercier

remercier

remerciement *(Output)*

$N \rightarrow V$

parasyntetisch

RE-Präfigierung

$V \rightarrow N$

merci

(Input)

*mercier

merci + V-Endung

remercier

re + *mercier

remerciement

remercie(r) + -ment

(Output)

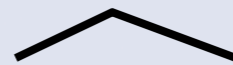
$N \rightarrow V$

parasyntetisch

RE-Präfigierung

$V \rightarrow N$

*V



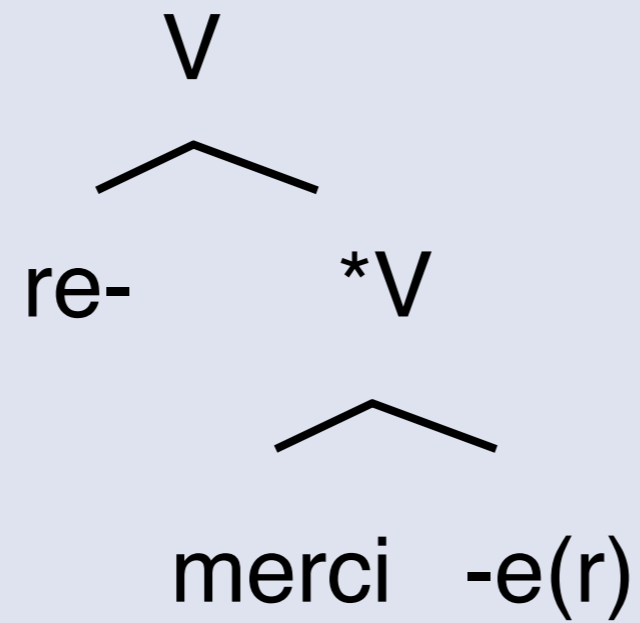
merci -e(r)

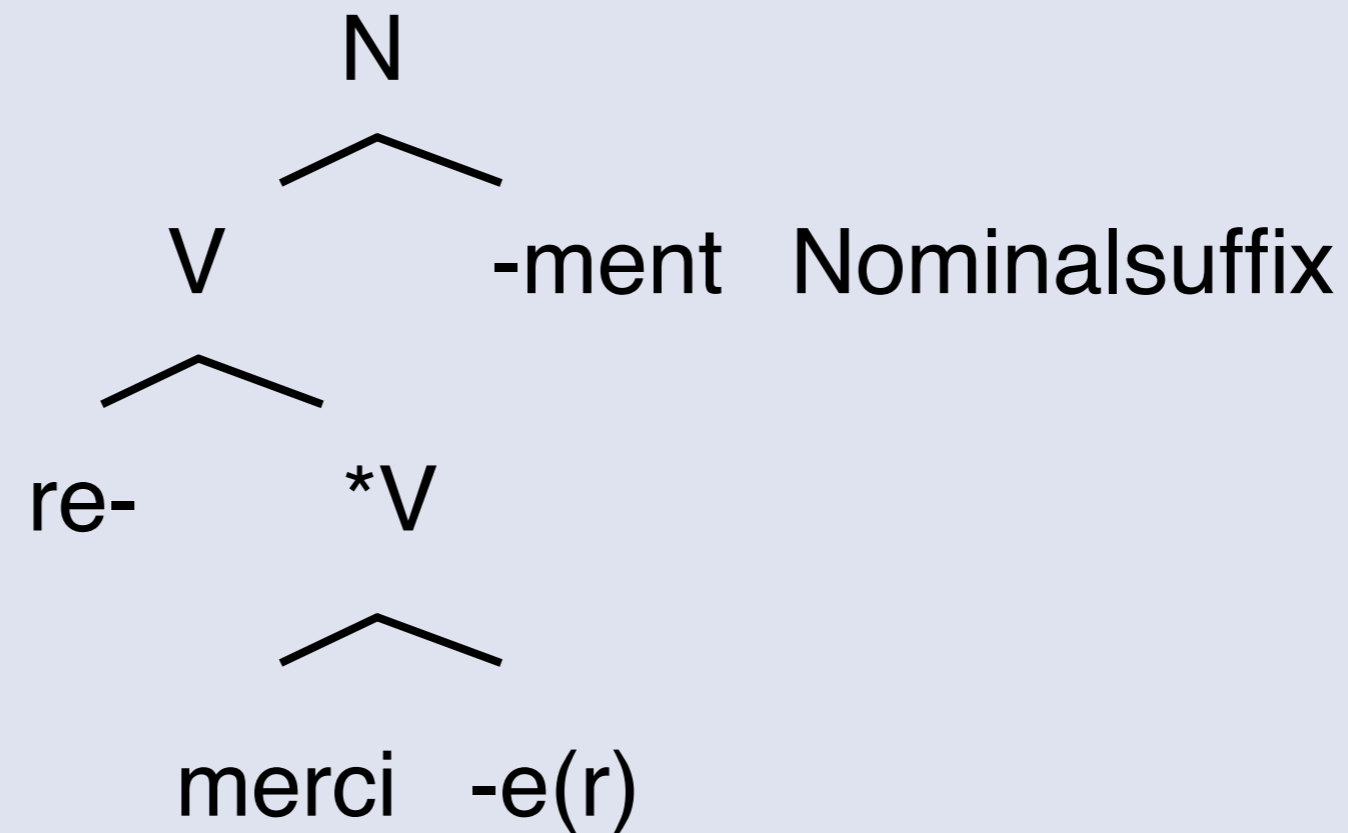
Parasynthese

Verbalendung (Infinitiv)

Dank(e)

RE-Präfix





Funktion *merge* ('zusammenfügen')



Input: merci, -e(r), re-, -ment

1. merci + -e(r)
2. re- + [merci -e(r)]
3. [re- [merci -e(r)]] + -ment

Output: [[re- [merci -e]] -ment]

Funktion *merge* ('zusammenfügen')



Input: merci

1. merci + -e(r)
2. re- + [merci -e(r)]
3. [re- [merci -e(r)]] + -ment

Output: remerciement

Input: merci

1. merci + -e(r)
2. re- + [merci -e(r)]
3. [re- [merci -e(r)]] + -ment

Output: remerciement

Derivation

„Derivation“ übertragen auf Syntax

María quiere comprar un coche.
Marie veut acheter une voiture.
Maria vuole comprare una macchina.
Maria will ein Auto kaufen.

Numeration: {María, INFL, querer, PRO, comprar, un, coche}

Derivation:

- Select: {un, coche} – Merge: N'' [un coche]
- Select: {comprar, N'' } – Merge: V' [comprar N'']
- Select: {PRO, V' } – Merge: V'' [PRO V']
- Select: {INFL_[-fin], V'' } – Merge: $I'_{[-fin]}$ [INFL_[-fin] V'']
- Select: {querer, $I'_{[-fin]}$ } – Merge: V' [querer $I'_{[-fin]}$]
- Select: {María, V' } – Merge: V'' [María, V']
- Select: {INFL, V'' } – Merge: I' [INFL V'']
- Move: I'' [María_k I' [quiere_i V'' [t_k V' [t_i $I'_{[-fin]}$]]]]]

Spell-Out: María quiere comprar un coche.



Numeration: lexikalische Entitäten (*items*), auch INFL

Derivation:

- Select – Merge
- Move

Spell-Out

Numeration: lexikalische Entitäten (*items*), auch INFL

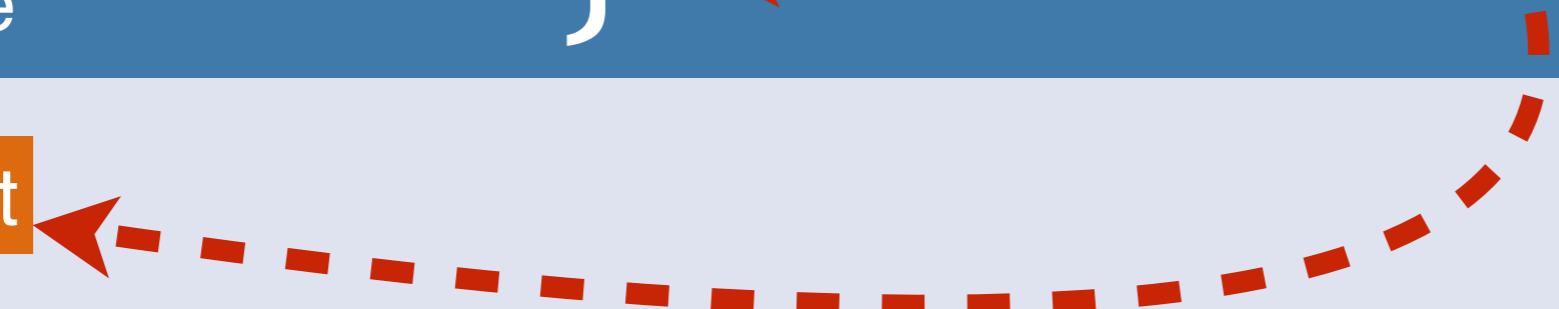
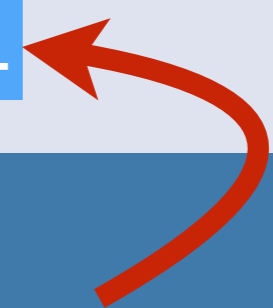
Derivation:

- Select – Merge
- Move



mentale Operationen

Spell-Out



Numeration: lexikalische Entitäten (*items*), auch INFL

Derivation:

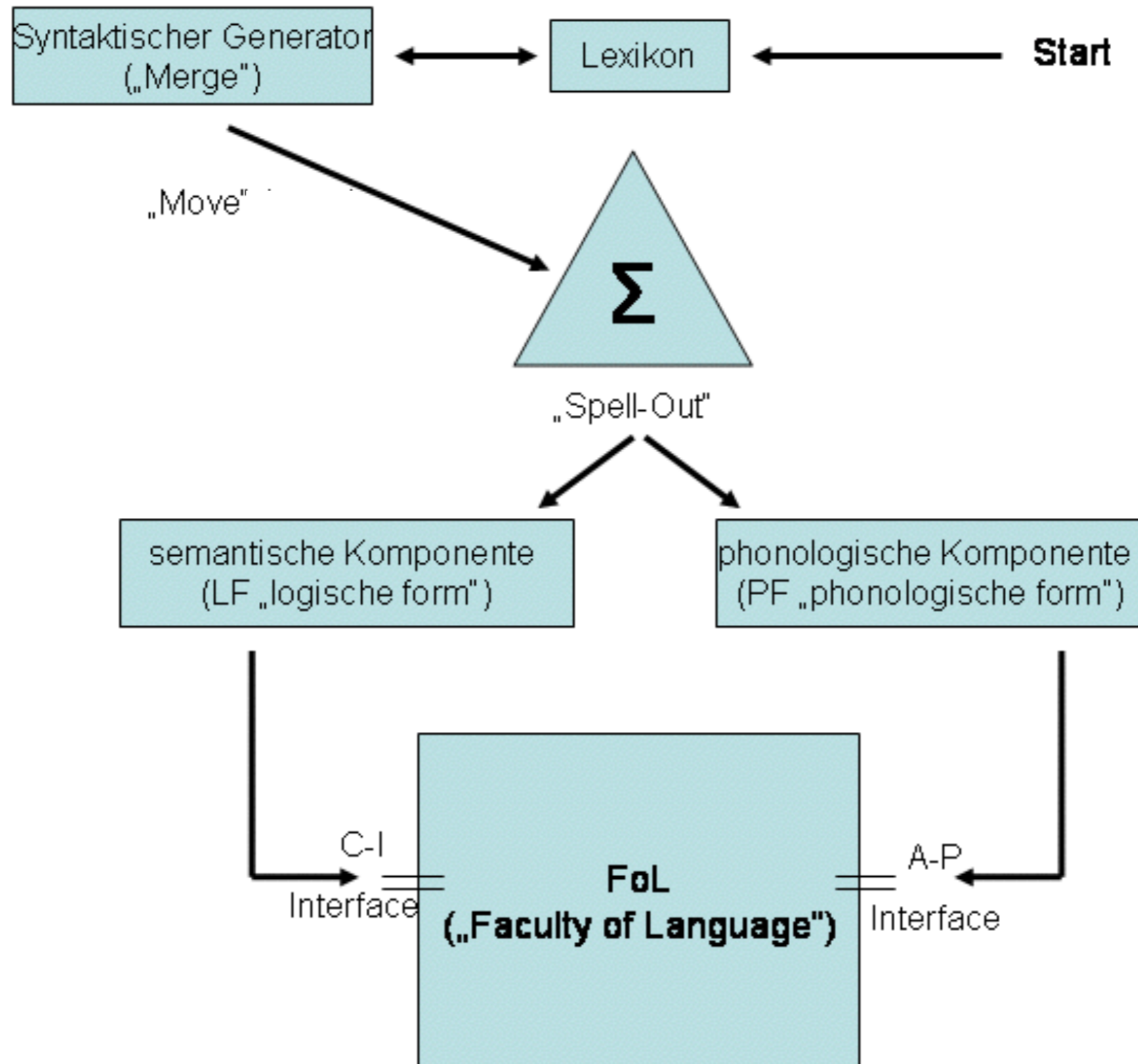
- Select – Merge
- Move



mentale Operationen

Spell-Out

überführt mentale Operationen
in logische Form (LF) & phonologische Form (PF)



Problem

produktionsorientiert
was geschieht beim Verstehen?

führt PF/LDF den mentalen Operationen
des Verstehens zu

Parsing

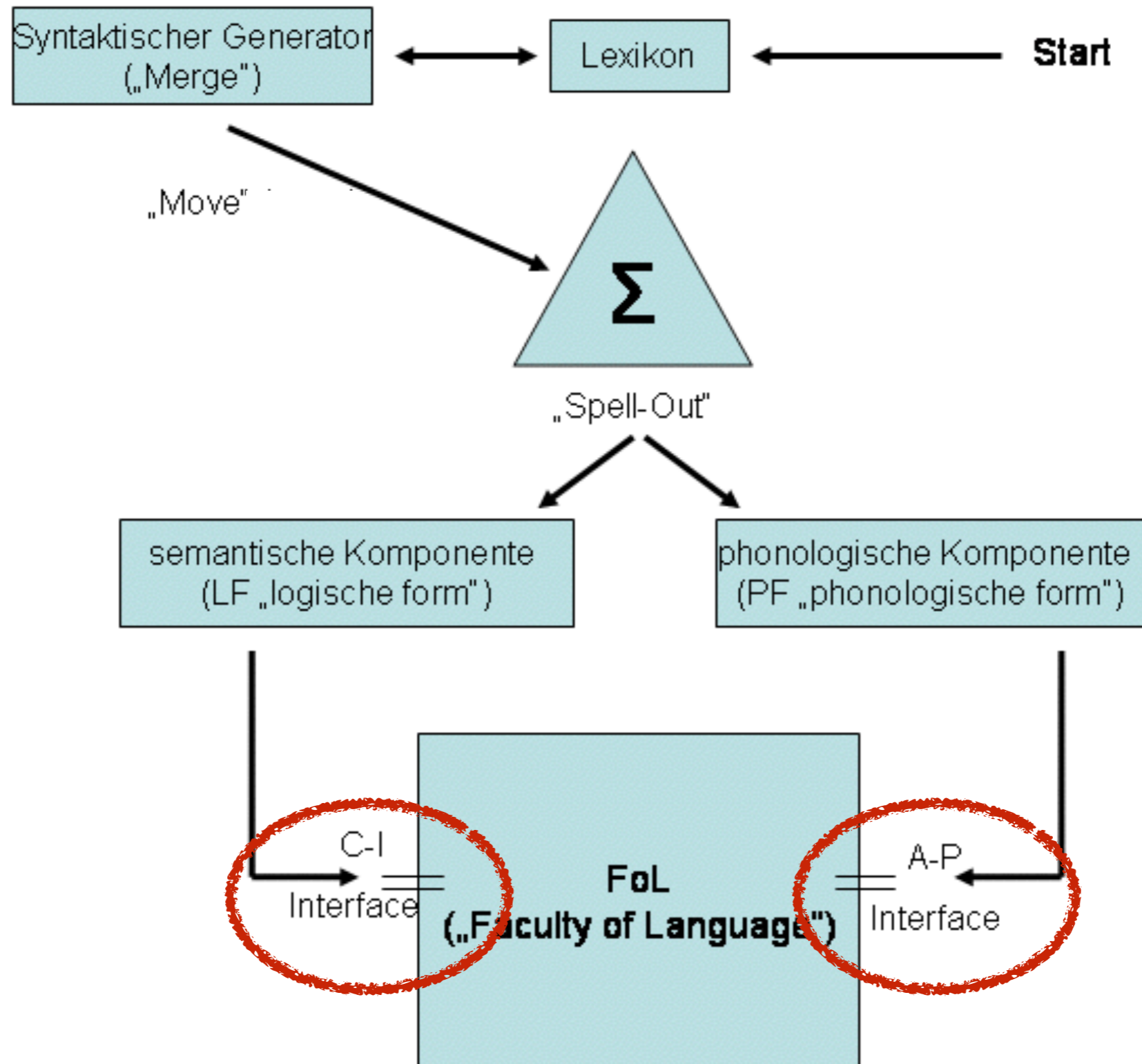
Analyse:

- Parse
- Move?

}

mentale Operationen

Numeration: lexikalische Entitäten (*items*)



C-I: conceptional-intentional interface

A-P: articulatory-phonetic interface

C-I: conceptual-intentional interface

A-P: articulatory-phonetic interface

Pragmatik

C-I: conceptual-intentional interface

A-P: articulatory-phonetic interface

Pragmatik

C-I: conceptual-intentional interface

A-P: articulatory-phonetic interface

Gebärdensprache

Numeration: lexikalische Entitäten (*items*), auch INFL

Derivation:

- Select {X, Y} – Merge Z[X Y]
- Move

Spell-Out

Numeration: lexikalische Entitäten (*items*), auch INFL

Derivation:

- Select {X, Y} – Merge Z[X Y]
- Move

Spell-Out

bare phrase structure:

- X''-Schema nötig?
- also Unterscheidung von Spezifizierern und Schwesterkonstituenten?
- Move ein spezieller Fall von Merge?