

# Alternative Theorien: Paradigmenfunktionsmorphologie

Johannes Hein

Universität Potsdam  
johannes.hein@uni-potsdam.de

7. Juni 2018

# Einleitung

# Typologie morphologischer Theorien

Neben der Distribuierten Morphologie gibt es natürlich unzählige weitere Möglichkeiten, morphologische Phänomene zu formalisieren.

- ❖ Da wir nicht alle theoretischen Vorschläge besprechen können, lohnt es sich, über konkrete Theorien zu abstrahieren, um die theorieunterscheidenden Kernpunkte zu erfassen.
- ❖ Dazu schlägt Stump (2001) eine Typologie morphologischer Theorien vor.
- ❖ Laut dieser können Theorien anhand von zwei Fragen unterschieden werden:
  1. Werden einzelne Morpheme als unabhängige Entitäten im Lexikon gespeichert?
  2. Tragen diese einzelnen Morpheme zur Bedeutung des Wortes bei oder realisieren sie nur unabhängig vorhandene Bedeutung?

# Frage 1

Werden einzelne Morpheme als unabhängige Entitäten im Lexikon gespeichert?

# Frage 1

Werden einzelne Morpheme als unabhängige Entitäten im Lexikon gespeichert?

- ❖ Theorien, die diese Frage mit **Ja** beantworten, bezeichnet Stump als **lexikalisch**.

# Frage 1

Werden einzelne Morpheme als unabhängige Entitäten im Lexikon gespeichert?

- ❖ Theorien, die diese Frage mit **Ja** beantworten, bezeichnet Stump als **lexikalisch**.
- ❖ Theorien, die diese Frage mit **Nein** beantworten, bezeichnet Stump als **inferentiell**.

# Frage 1

Werden einzelne Morpheme als unabhängige Entitäten im Lexikon gespeichert?

- ❖ Theorien, die diese Frage mit **Ja** beantworten, bezeichnet Stump als **lexikalisch**.
  - ❖ Theorien, die diese Frage mit **Nein** beantworten, bezeichnet Stump als **inferentiell**.
- 
- ❖ In der Distribuierten Morphologie werden die einzelnen Morpheme unabhängig von der Wortform, in der sie auftauchen, im Lexikon als Vokabularelemente gespeichert.  
⇒ Daher ist DM in dieser Typologie eine lexikalische Theorie.

# Frage 1

## Werden einzelne Morpheme als unabhängige Entitäten im Lexikon gespeichert?

- ❖ Theorien, die diese Frage mit **Ja** beantworten, bezeichnet Stump als **lexikalisch**.
  - ❖ Theorien, die diese Frage mit **Nein** beantworten, bezeichnet Stump als **inferentiell**.
- 
- ❖ In der Distribuierten Morphologie werden die einzelnen Morpheme unabhängig von der Wortform, in der sie auftauchen, im Lexikon als Vokabularelemente gespeichert.  
⇒ Daher ist DM in dieser Typologie eine lexikalische Theorie.
  - ❖ Die Paradigmenfunktionsmorphologie nimmt nicht an, dass einzelne Morpheme im Lexikon gespeichert werden.  
⇒ Sie ist daher eine inferentielle Theorie.



## Frage 2

Tragen die einzelnen Morpheme zur Bedeutung des Wortes bei?

# Frage 2

Tragen die einzelnen Morpheme zur Bedeutung des Wortes bei?

- ❖ Theorien, die diese Frage mit **Ja** beantworten, bezeichnet Stump als **inkrementell**.

# Frage 2

Tragen die einzelnen Morpheme zur Bedeutung des Wortes bei?

- ❖ Theorien, die diese Frage mit **Ja** beantworten, bezeichnet Stump als **inkrementell**.
- ❖ Theorien, die diese Frage mit **Nein** beantworten, bezeichnet Stump als **realisational**.

# Frage 2

## Tragen die einzelnen Morpheme zur Bedeutung des Wortes bei?

- ❖ Theorien, die diese Frage mit **Ja** beantworten, bezeichnet Stump als **inkrementell**.
- ❖ Theorien, die diese Frage mit **Nein** beantworten, bezeichnet Stump als **realisational**.
  
- ❖ In der Distribuierten Morphologie tragen die einzelnen Morpheme nichts bei zur Bedeutung. Sie realisieren lediglich die unabhängig existierenden Merkmale.  
⇒ DM ist daher eine realisationale Theorie.

# Frage 2

## Tragen die einzelnen Morpheme zur Bedeutung des Wortes bei?

- ❖ Theorien, die diese Frage mit **Ja** beantworten, bezeichnet Stump als **inkrementell**.
  - ❖ Theorien, die diese Frage mit **Nein** beantworten, bezeichnet Stump als **realisational**.
- 
- ❖ In der Distribuierten Morphologie tragen die einzelnen Morpheme nichts bei zur Bedeutung. Sie realisieren lediglich die unabhängig existierenden Merkmale.  
⇒ DM ist daher eine realisationale Theorie.
  - ❖ Auch die Paradigmenfunktionsmorphologie realisiert unabhängig vorhandene Merkmale.  
⇒ PFM ist daher auch realisational.

# Theorie-Typ 1

Kreuzklassifiziert man diese beiden Dichotomien, bekommt man eine vollständige Typologie.

# Theorie-Typ 1

Kreuzklassifiziert man diese beiden Dichotomien, bekommt man eine vollständige Typologie.

- ❖ Typ 1: Lexikalisch-Inkrementell:  
Eine lexikalisch-inkrementelle Theorie nimmt an, dass es unabhängige Morpheme im Lexikon gibt, die ihren Teil zu den morphosyntaktischen Merkmalen des kombinierten Wortes beitragen:

# Theorie-Typ 1

Kreuzklassifiziert man diese beiden Dichotomien, bekommt man eine vollständige Typologie.

❖ Typ 1: Lexikalisch-Inkrementell:

Eine lexikalisch-inkrementelle Theorie nimmt an, dass es unabhängige Morpheme im Lexikon gibt, die ihren Teil zu den morphosyntaktischen Merkmalen des kombinierten Wortes beitragen:

(1) *Lexikoneinträge*  
 st {2.SG}  
 sag {V}

(2)        sagst  
               {V.2.SG}  
               /  \  
 sag          st  
 {V}      {2.SG}



# Theorie-Typ 1

Kreuzklassifiziert man diese beiden Dichotomien, bekommt man eine vollständige Typologie.

- ❖ Typ 1: Lexikalisch-Inkrementell:  
Eine lexikalisch-inkrementelle Theorie nimmt an, dass es unabhängige Morpheme im Lexikon gibt, die ihren Teil zu den morphosyntaktischen Merkmalen des kombinierten Wortes beitragen:

(1) *Lexikoneinträge*  
 st {2.SG}  
 sag {V}

(2)        sagst  
               {V.2.SG}  
               /  \  
 sag          st  
 {V}      {2.SG}

- ▶ Eine solche Theorie ist z.B. die von Lieber (1992) und die Minimalistische Morphologie (Wunderlich 1996 et seq.; Stiebels 2011).

# Theorie-Typ 2

- ❖ Typ 2: Lexikalisch-Realisational:  
Eine lexikalisch-realisationale Theorie nimmt an, dass es unabhängig gespeicherte Morpheme im Lexikon gibt, die selbst aber keine Information beitragen, sondern nur bereits vorhandene Information realisieren.

# Theorie-Typ 2

- ❖ Typ 2: Lexikalisch-Realisational:  
Eine lexikalisch-realisationale Theorie nimmt an, dass es unabhängig gespeicherte Morpheme im Lexikon gibt, die selbst aber keine Information beitragen, sondern nur bereits vorhandene Information realisieren.
  - ▶ DM gilt als lexikalisch-realisational, da die vollständige Spezifikation aller Merkmale schon von der Syntax bereitgestellt wird.
  - ▶ Die Vokabularelemente, die im Lexikon gespeichert sind, realisieren diese Information nur.

# Theorie-Typ 3

- ❖ Typ 3: Inferentiell-Realisational:  
Eine solche Theorie nimmt an, dass es keine Morpheme im Lexikon gibt. Alle Information ist schon unabhängig vorhanden und wird bloß mittels Realisierungsregeln o.ä. phonologisch hörbar gemacht.

# Theorie-Typ 3

- ❖ Typ 3: Inferentiell-Realisational:  
Eine solche Theorie nimmt an, dass es keine Morpheme im Lexikon gibt. Alle Information ist schon unabhängig vorhanden und wird bloß mittels Realisierungsregeln o.ä. phonologisch hörbar gemacht.
  - ▶ Hierunter fallen sogenannte Wort- und Paradigma-Ansätze wie etwa Paradigmenfunktionsmorphologie (Stump 2001), A-morphous Morphologie (Anderson 1992, Aronoff 1994).

# Theorie-Typ 3

- ❖ Typ 3: Inferentiell-Realisational:  
Eine solche Theorie nimmt an, dass es keine Morpheme im Lexikon gibt. Alle Information ist schon unabhängig vorhanden und wird bloß mittels Realisierungsregeln o.ä. phonologisch hörbar gemacht.
  - ▶ Hierunter fallen sogenannte Wort- und Paradigma-Ansätze wie etwa Paradigmenfunktionsmorphologie (Stump 2001), A-morphous Morphologie (Anderson 1992, Aronoff 1994).  
⇒ Die Form /sagst/ entspricht in solchen Ansätzen lediglich jener Paradigmenzelle, die für [sag-, 2.Person, Singular] spezifiziert ist.

# Theorie-Typ 3

- ❖ Typ 3: Inferentiell-Realisational:  
Eine solche Theorie nimmt an, dass es keine Morpheme im Lexikon gibt. Alle Information ist schon unabhängig vorhanden und wird bloß mittels Realisierungsregeln o.ä. phonologisch hörbar gemacht.
  - ▶ Hierunter fallen sogenannte Wort- und Paradigma-Ansätze wie etwa Paradigmenfunktionsmorphologie (Stump 2001), A-morphous Morphologie (Anderson 1992, Aronoff 1994).
    - ⇒ Die Form /sagst/ entspricht in solchen Ansätzen lediglich jener Paradigmenzelle, die für [sag-, 2.Person, Singular] spezifiziert ist.
    - ⇒ Die Form enthält aber nur ein Morphem.

# Theorie-Typ 4

- ❖ Typ 4: Inferentiell-Inkrementell:  
Logisch gesehen gibt es noch einen vierten Typ, der annähme, dass es keine unabhängigen Morpheme für Flexionsmarker, etc., im Lexikon gibt, diese aber Information beitragen.



# Theorie-Typ 4

- ❖ Typ 4: Inferentiell-Inkrementell:  
Logisch gesehen gibt es noch einen vierten Typ, der annähme, dass es keine unabhängigen Morpheme für Flexionsmarker, etc., im Lexikon gibt, diese aber Information beitragen.
- ▶ Dabei handelt es sich wahrscheinlich um einen logischen Widerspruch, wofür auch spricht, dass es derartige Ansätze praktisch nicht gibt.

# Paradigmenfunktionsmorphologie

# Paradigmenfunktionsmorphologie

Stump (2001 et seq.) schlägt einen inferentiell-realisationalen Ansatz vor, der versucht, die Daten im Bereich der Wortbildung in einem präsyntaktischen Modell zu erfassen.

- ❖ Die Theorie der Paradigmenfunktionsmorphologie (PFM) wurde mehrfach weiterentwickelt und um komplexe Mechanismen erweitert (siehe etwa Stump 2002, 2006, 2010, 2012; Stewart & Stump 2007; Bonami & Webelhuth 2013; Bonami & Boyé 2002, 2007).
- ❖ PFM basiert laut Bonami & Stump (2017) zentral auf 5 Prinzipien, die wir im Folgenden durchgehen werden. (Es sind eigentlich 6 Prinzipien, aber das zur Köpfigkeit von komplexen Wörtern würde uns hier etwas weit weg führen.)

# 1. Prinzip

Morphologie ist ein autonomes Modul der Grammatik

# 1. Prinzip

## Morphologie ist ein autonomes Modul der Grammatik

- ❖ Die Regeln dieses Moduls können nur bedingt auf Information von außerhalb des Moduls (sprich syntaktische oder phonologische Information) zugreifen.

# 1. Prinzip

## Morphologie ist ein autonomes Modul der Grammatik

- ❖ Die Regeln dieses Moduls können nur bedingt auf Information von außerhalb des Moduls (sprich syntaktische oder phonologische Information) zugreifen.
- ❖ Zugleich gibt es aber auch genuin morphologische Information (Flexionsklassen, Köpfigkeit von komplexen Wörtern), die für Regeln anderer Module nicht zugänglich ist.

# 1. Prinzip

## Morphologie ist ein autonomes Modul der Grammatik

- ❖ Die Regeln dieses Moduls können nur bedingt auf Information von außerhalb des Moduls (sprich syntaktische oder phonologische Information) zugreifen.
- ❖ Zugleich gibt es aber auch genuin morphologische Information (Flexionsklassen, Köpfigkeit von komplexen Wörtern), die für Regeln anderer Module nicht zugänglich ist.
- ▶ Z.B.: Eine morphologische Regel sieht, dass /*child*/ der morphologische Kopf von /*grandchild*/ ist und kann irregulären Plural bilden: /*grandchildren*/.

# 1. Prinzip

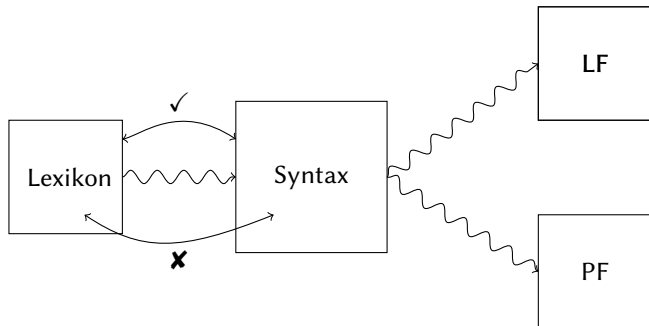
## Morphologie ist ein autonomes Modul der Grammatik

- ❖ Die Regeln dieses Moduls können nur bedingt auf Information von außerhalb des Moduls (sprich syntaktische oder phonologische Information) zugreifen.
- ❖ Zugleich gibt es aber auch genuin morphologische Information (Flexionsklassen, Köpfigkeit von komplexen Wörtern), die für Regeln anderer Module nicht zugänglich ist.
- ▶ Z.B.: Eine morphologische Regel sieht, dass /*child*/ der morphologische Kopf von /*grandchild*/ ist und kann irregulären Plural bilden: /*grandchildren*/.
- ▶ Eine syntaktische Regel kann keinen Bezug auf einen einzelnen Teil eines Wortes nehmen (Lexikale Integrität).



# Modell der Grammatik in PFM

## (3) PFM-Modell der Grammatik



Syntax und Morphologie interagieren mittels ihrer Schnittstelle. Das bedeutet, dass die Syntax nur den Output der Morphologie sehen kann, aber nicht die Details der morphologischen Derivation.

## 2. Prinzip

Die Definition der Flexionsmorphologie einer Sprache ist die Menge ihrer Paradigmenfunktionen

## 2. Prinzip

Die Definition der Flexionsmorphologie einer Sprache ist die Menge ihrer Paradigmenfunktionen

- ❖ Flektierte Formen eines Stammes werden durch Paradigmenfunktionen erzeugt. Eine Paradigmenfunktion (PF) erzeugt eine Realisierung R auf der Basis eines Lexems L mit der vollen Merkmalspezifikation  $\sigma$ .

$$(4) \quad PF(\langle L, \sigma \rangle) = \langle R, \sigma \rangle$$

## 2. Prinzip

Die Definition der Flexionsmorphologie einer Sprache ist die Menge ihrer Paradigmenfunktionen

- ❖ Flektierte Formen eines Stammes werden durch Paradigmenfunktionen erzeugt. Eine Paradigmenfunktion (PF) erzeugt eine Realisierung R auf der Basis eines Lexems L mit der vollen Merkmalspezifikation  $\sigma$ .

$$(4) \quad PF(\langle L, \sigma \rangle) = \langle R, \sigma \rangle$$

Eine Beispielregel für */sagtest/* sieht also aus wie in (5).

$$(5) \quad PF(\langle \text{SAG}, \sigma : \{IND, 2, SG, PAST\} \rangle) = \langle \text{sagtest}, \sigma \rangle$$

## 2. Prinzip

Die Definition der Flexionsmorphologie einer Sprache ist die Menge ihrer Paradigmenfunktionen

- ❖ Flektierte Formen eines Stammes werden durch Paradigmenfunktionen erzeugt. Eine Paradigmenfunktion (PF) erzeugt eine Realisierung R auf der Basis eines Lexems L mit der vollen Merkmalspezifikation  $\sigma$ .

$$(4) \quad PF(\langle L, \sigma \rangle) = \langle R, \sigma \rangle$$

Eine Beispielregel für */sagtest/* sieht also aus wie in (5).

$$(5) \quad PF(\langle \text{SAG}, \sigma : \{IND, 2, SG, PAST\} \rangle) = \langle \text{sagtest}, \sigma \rangle$$

*/st/* in */sagtest/* ist also nicht im Lexikon gespeichert, sondern die Paradigmenfunktion in (5), die */sagtest/* erzeugt, ist Teil der deutschen Grammatik.

# Realisierungsregeln

Eine Paradigmenfunktion wie in (5) kann in mehrere Realisierungsregeln dekomponiert werden.

$$(6) \quad PF(\langle \text{SAG}, \sigma : \{IND, 2, SG, PAST\} \rangle) = \langle \text{sagtest}, \sigma \rangle$$

# Realisierungsregeln

Eine Paradigmenfunktion wie in (5) kann in mehrere Realisierungsregeln dekomponiert werden.

$$(6) \quad PF(\langle \text{SAG}, \sigma : \{IND, 2, SG, PAST\} \rangle) = \langle \text{sagtest}, \sigma \rangle$$

(7) Realisierungsregeln (vereinfacht):

$$a. \quad RR(\langle X, \sigma \rangle) = \langle X', \sigma \rangle \text{ wobei } X' \text{ der Stamm von } X \text{ ist.}$$

# Realisierungsregeln

Eine Paradigmenfunktion wie in (5) kann in mehrere Realisierungsregeln dekomponiert werden.

$$(6) \quad PF(\langle \text{SAG}, \sigma : \{IND, 2, SG, PAST\} \rangle) = \langle \text{sagtest}, \sigma \rangle$$

(7) Realisierungsregeln (vereinfacht):

a.  $RR(\langle X, \sigma \rangle) = \langle X', \sigma \rangle$  wobei  $X'$  der Stamm von  $X$  ist.

b.  $RR(\langle X, \sigma \rangle) = \langle Xte, \sigma \rangle$



# Realisierungsregeln

Eine Paradigmenfunktion wie in (5) kann in mehrere Realisierungsregeln dekomponiert werden.

$$(6) \quad PF(\langle \text{SAG}, \sigma : \{IND, 2, SG, PAST\} \rangle) = \langle \text{sagtest}, \sigma \rangle$$

(7) Realisierungsregeln (vereinfacht):

- a.  $RR(\langle X, \sigma \rangle) = \langle X', \sigma \rangle$  wobei  $X'$  der Stamm von  $X$  ist.
- b.  $RR(\langle X, \sigma \rangle) = \langle Xte, \sigma \rangle$
- c.  $RR(\langle X, \sigma \rangle) = \langle Xst, \sigma \rangle$

# Realisierungsregeln

Eine Paradigmenfunktion wie in (5) kann in mehrere Realisierungsregeln dekomponiert werden.

$$(6) \quad PF(\langle \text{SAG}, \sigma : \{IND, 2, SG, PAST\} \rangle) = \langle \text{sagtest}, \sigma \rangle$$

(7) Realisierungsregeln (vereinfacht):

- a.  $RR(\langle X, \sigma \rangle) = \langle X', \sigma \rangle$  wobei  $X'$  der Stamm von  $X$  ist.
- b.  $RR(\langle X, \sigma \rangle) = \langle Xte, \sigma \rangle$
- c.  $RR(\langle X, \sigma \rangle) = \langle Xst, \sigma \rangle$

Die Reihenfolge der RRs wird arbiträr festgelegt (siehe Regelblöcke).

### 3. Prinzip

Wettbewerb zwischen Flexionsregeln wird durch das Teilmengenprinzip aufgelöst

### 3. Prinzip

Wettbewerb zwischen Flexionsregeln wird durch das Teilmengenprinzip aufgelöst

- ❖ Wie die Vokabularelemente in DM können die Regeln für morphosyntaktische Merkmale unterspezifiziert sein.

### 3. Prinzip

Wettbewerb zwischen Flexionsregeln wird durch das Teilmengenprinzip aufgelöst

- ❖ Wie die Vokabularelemente in DM können die Regeln für morphosyntaktische Merkmale unterspezifiziert sein.
- ▶ Das führt unter Umständen zu Wettbewerb, der dann durch das Teilmengenprinzip aufgelöst wird.

# 3. Prinzip

## Wettbewerb zwischen Flexionsregeln wird durch das Teilmengenprinzip aufgelöst

- ❖ Wie die Vokabularelemente in DM können die Regeln für morphosyntaktische Merkmale unterspezifiziert sein.
- ▶ Das führt unter Umständen zu Wettbewerb, der dann durch das Teilmengenprinzip aufgelöst wird.
- ▶ Das bedeutet, dass die spezifischste Regel (vgl. (8-b)) gewinnt und Anwendung findet.

- (8) a.  $PF(\langle \text{SAG}, \sigma : \{IND, PL\} \rangle) = \langle \text{sagen}, \sigma \rangle$   
 b.  $PF(\langle \text{SAG}, \sigma : \{IND, 2, PL\} \rangle) = \langle \text{sagt}, \sigma \rangle$

# Merkmalsdekomposition

Der Wettbewerb zwischen konkurrierenden Markern wird in PFM wie auch in DM mittels des Teilmengenprinzips aufgelöst.

- ▶ Ähnlich wie in DM wird in PFM oftmals zusätzlich Dekomposition von Merkmalen angenommen, ist aber kein integraler Bestandteil der Theorie.

# Merkmalsdekomposition

Der Wettbewerb zwischen konkurrierenden Markern wird in PFM wie auch in DM mittels des Teilmengenprinzips aufgelöst.

- ▶ Ähnlich wie in DM wird in PFM oftmals zusätzlich Dekomposition von Merkmalen angenommen, ist aber kein integraler Bestandteil der Theorie.

Nebenbemerkung:

In PFM wird das Teilmengenprinzip “Pāṇini’s principle” genannt (nach einem Sanskritgrammatiker aus dem 6.–4. Jhdt. v. Chr.). Es tut aber genau dasselbe.



## 4. Prinzip

Paradigmenfunktionsregeln können implikative Regeln (Verweisregeln, *rules of referral*) enthalten

## 4. Prinzip

Paradigmenfunktionsregeln können implikative Regeln (Verweisregeln, *rules of referral*) enthalten

- ❖ Es kann implikationelle Regeln geben, die bestimmte Paradigmenzellen auf andere Paradigmenzellen verweisen.

## 4. Prinzip

Paradigmenfunktionsregeln können implikative Regeln (Verweisregeln, *rules of referral*) enthalten

- ❖ Es kann implikationelle Regeln geben, die bestimmte Paradigmenzellen auf andere Paradigmenzellen verweisen.
- ▶ Der berühmte NOM-ACC Synkretismus im Indo-Europäischen wird laut Bonami & Stump (2016) folgendermaßen abgeleitet.

(9) Wenn L ein Nomen im Neutrum ist und  
 $PF(\langle L, \sigma:\{\text{acc}, \text{NUM}:\alpha\} \rangle) = \langle Y, \sigma \rangle$ , dann gilt:  
 $PF(\langle L, \sigma':\{\text{nom}, \text{NUM}:\alpha\} \rangle) = \langle Y, \sigma' \rangle$ ,

# Verweisregeln

- ❖ Die Regel in (9) verweist eine Paradigmenzelle einfach arbiträr auf eine andere Paradigmenzelle.

# Verweisregeln

- ❖ Die Regel in (9) verweist eine Paradigmenzelle einfach arbiträr auf eine andere Paradigmenzelle.
- ❖ Das ist ein ungemein mächtiges Werkzeug, da es jede beliebige Zelle in einem Paradigma mit einer anderen Zelle in Bezug setzen kann.

# Verweisregeln

- ❖ Die Regel in (9) verweist eine Paradigmenzelle einfach arbiträr auf eine andere Paradigmenzelle.
- ❖ Das ist ein ungemein mächtiges Werkzeug, da es jede beliebige Zelle in einem Paradigma mit einer anderen Zelle in Bezug setzen kann.
- ▶ Im Vergleich zu Verarmung sind derartige Verweisregeln sogar noch mächtiger, da Verarmung auf Merkmalslöschung beschränkt ist. Verarmung kann im Gegensatz zu Verweisregeln also nicht einfach morphosyntaktische Merkmale beliebig ändern.

## 5. Prinzip

Die Paradigmenfunktionsregeln einer Sprache sind inferentiell und realisational

## 5. Prinzip

### Die Paradigmenfunktionsregeln einer Sprache sind inferentiell und realisational

- ❖ Eine Regel wie (10) ist eine Funktion, die einem sagt, wie ein bestimmtes Lexem in einem morphosyntaktischen Kontext realisiert wird.

$$(10) \quad PF(\langle \text{sag}, \sigma : \{IND, 2, PL\} \rangle) = \langle \text{sagt}, \sigma \rangle$$

Die Affixe selbst haben keinerlei eigenen Morphemstatus und tragen auch keinerlei Merkmale bei. Alle Merkmale  $\sigma$  sind bereits auf dem Lexem vorhanden.



# Regelblöcke

Laut Bonami & Stump (2016) keine zentrale Annahme des PFM-Ansatzes, aber doch ein zentraler Unterschied zu anderen Frameworks ist folgende Annahme:

# Regelblöcke

Laut Bonami & Stump (2016) keine zentrale Annahme des PFM-Ansatzes, aber doch ein zentraler Unterschied zu anderen Frameworks ist folgende Annahme:

- ▶ Paradigmenfunktionsregeln sind in Regelblöcken angeordnet, die zugleich die Ordnung der Affixe vorgeben.

$$(11) \quad \square_1 - \square_2 - \sqrt{\quad} - \square_3 - \square_4 - \square_5$$

# Regelblöcke

Laut Bonami & Stump (2016) keine zentrale Annahme des PFM-Ansatzes, aber doch ein zentraler Unterschied zu anderen Frameworks ist folgende Annahme:

- ▶ Paradigmenfunktionsregeln sind in Regelblöcken angeordnet, die zugleich die Ordnung der Affixe vorgeben.

$$(11) \quad \square_1 - \square_2 - \sqrt{\quad} - \square_3 - \square_4 - \square_5$$

- ❖ Jede Regel ist einem Block zugeordnet, der genau eine Stelle im Template besetzt.

# Regelblöcke

Laut Bonami & Stump (2016) keine zentrale Annahme des PFM-Ansatzes, aber doch ein zentraler Unterschied zu anderen Frameworks ist folgende Annahme:

- ▶ Paradigmenfunktionsregeln sind in Regelblöcken angeordnet, die zugleich die Ordnung der Affixe vorgeben.

$$(11) \quad \square_1 - \square_2 - \sqrt{\quad} - \square_3 - \square_4 - \square_5$$

- ❖ Jede Regel ist einem Block zugeordnet, der genau eine Stelle im Template besetzt.
- ❖ Nur die Regeln innerhalb eines Blockes treten in Wettbewerb zueinander und können sich gegenseitig blockieren.

# Regelblöcke

Die Reihenfolge der Regelblöcke und deren Inhalt ist vollkommen arbiträr und korreliert weder mit syntaktischem C-Kommando noch mit semantischem Skopus o.ä.

# Regelblöcke

Die Reihenfolge der Regelblöcke und deren Inhalt ist vollkommen arbiträr und korreliert weder mit syntaktischem C-Kommando noch mit semantischem Skopus o.ä.

- ❖ Wann immer es so aussieht, als gäbe es Skopuseffekte o.ä., sind immer funktionale oder diachrone Gründe anzuführen.

# Regelblöcke

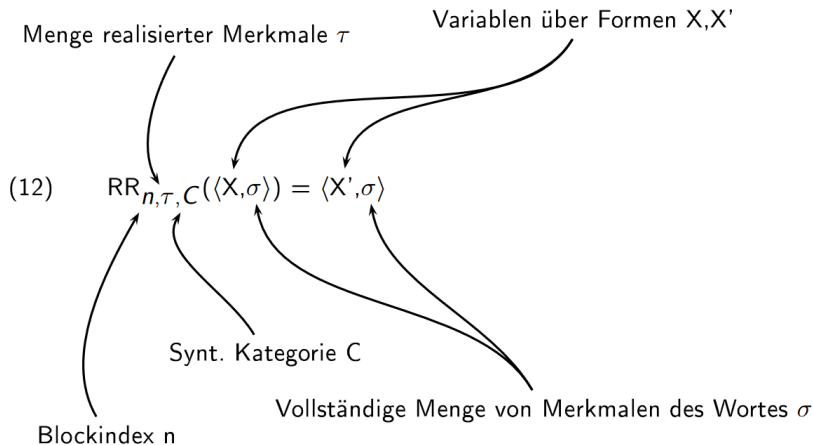
Die Reihenfolge der Regelblöcke und deren Inhalt ist vollkommen arbiträr und korreliert weder mit syntaktischem C-Kommando noch mit semantischem Skopus o.ä.

- ❖ Wann immer es so aussieht, als gäbe es Skopuseffekte o.ä., sind immer funktionale oder diachrone Gründe anzuführen.

Das ist ein grundlegender Unterschied zu Frameworks, die das Spiegelprinzip als integralen Teil der Grammatik akzeptieren (etwa DM oder Minimalistische Morphologie) und auf die ein oder andere Art und Weise implementieren.

# Format von Realisierungsregeln

Eine Realisierungsregel in PFM sieht also insgesamt wie folgt aus.





# Hintergrundannahmen

## Grundannahme

Die Verknüpfung eines Wortes mit einer bestimmten Menge von morphosyntaktischen Eigenschaften determiniert eine Kette von Regelanwendungen, die die Flexionsform des Wortes bestimmen.

# Hintergrundannahmen

## Grundannahme

Die Verknüpfung eines Wortes mit einer bestimmten Menge von morphosyntaktischen Eigenschaften determiniert eine Kette von Regelanwendungen, die die Flexionsform des Wortes bestimmen.

## Traditionelle Terminologie

*Wort* ('Wort', 'Lexem'): z.B. BUCH; Wörter haben Paradigmen.

*Wortform* ('Flexionsform des Wortes'): z.B. *Buches*; Wortformen sind Teile von Paradigmen.

# Hintergrundannahmen

## Grundannahme

Die Verknüpfung eines Wortes mit einer bestimmten Menge von morphosyntaktischen Eigenschaften determiniert eine Kette von Regelanwendungen, die die Flexionsform des Wortes bestimmen.

## Traditionelle Terminologie

*Wort* ('Wort', 'Lexem'): z.B. BUCH; Wörter haben Paradigmen.

*Wortform* ('Flexionsform des Wortes'): z.B. *Buches*; Wortformen sind Teile von Paradigmen.

## Paradigmen (Behauptung)

In PFM sind Paradigmen keine Epiphänomene; vielmehr "konstituieren sie ein zentrales Prinzip der morphologischen Organisation". Paradigmen sind das Ergebnis von Paradigmenfunktionen.

# Hintergrundannahmen

## Grundannahme

Die Verknüpfung eines Wortes mit einer bestimmten Menge von morphosyntaktischen Eigenschaften determiniert eine Kette von Regelanwendungen, die die Flexionsform des Wortes bestimmen.

## Traditionelle Terminologie

*Wort* ('Wort', 'Lexem'): z.B. BUCH; Wörter haben Paradigmen.

*Wortform* ('Flexionsform des Wortes'): z.B. *Buches*; Wortformen sind Teile von Paradigmen.

## Paradigmen (Behauptung)

In PFM sind Paradigmen keine Epiphänomene; vielmehr "konstituieren sie ein zentrales Prinzip der morphologischen Organisation". Paradigmen sind das Ergebnis von Paradigmenfunktionen.

## Drei Typen morphologischer Ausdrücke

*Wurzel* ('root'): die "ultimative Default-Form" eines Lexems (Wortes).

*Stamm* ('stem'): ein Ausdruck, an den Flexionsexponenten angefügt werden können. (Jede Wurzel ist ein Stamm, nicht jeder Stamm ist eine Wurzel.)

*Wortform* ('word'(!)): eine freie, voll flektierte Form, die eine Paradigmenzelle besetzt.

# Hintergrundannahmen

## Realisierungsregeln

Paradigmen werden durch speziellere Realisierungsregeln definiert.

# Hintergrundannahmen

## Realisierungsregeln

Paradigmen werden durch speziellere Realisierungsregeln definiert.

Ein informelles Beispiel: Der Wert der Paradigmenfunktion

(⟨Mutter-,{dativ,plural}⟩) ergibt sich aus dem Ergebnis der Anwendung zweier Realisierungsregeln – einer, die die Umlautvariante des Stamms wählt, und einer, die *-n* suffigiert.

# Hintergrundannahmen

## Realisierungsregeln

Paradigmen werden durch speziellere Realisierungsregeln definiert.

Ein informelles Beispiel: Der Wert der Paradigmenfunktion

( $\langle \langle \text{Mutter-}, \{\text{dativ, plural}\} \rangle \rangle$ ) ergibt sich aus dem Ergebnis der Anwendung zweier Realisierungsregeln – einer, die die Umlautvariante des Stamms wählt, und einer, die *-n* suffigiert.

Terminologie:  $\langle \langle \text{Mutter-}, \{\text{dativ, plural}\} \rangle \rangle$  ist ein FPSP ('form/property-set pairing').

# Hintergrundannahmen

## Realisierungsregeln

Paradigmen werden durch speziellere Realisierungsregeln definiert.

Ein informelles Beispiel: Der Wert der Paradigmenfunktion

(⟨*Mutter*-,{dativ,plural}⟩) ergibt sich aus dem Ergebnis der Anwendung zweier Realisierungsregeln – einer, die die Umlautvariante des Stamms wählt, und einer, die *-n* suffigiert.

Terminologie: ⟨*Mutter*-,{dativ,plural}⟩ ist ein FPSP (‘form/property-set pairing’).

## Regelblöcke

- a. Die Realisierungsregeln einer Sprache sind in Blöcken organisiert.
- b. Regeln im selben Block konkurrieren miteinander; nur die spezifischste Regel kann applizieren (Pāṇinis Prinzip, Spezifizitätsprinzip).
- c. Regeln in verschiedenen Blöcken konkurrieren nicht; so treten in einer Wortform verschiedene Exponenten hintereinander.



# Einige Begriffe und Definitionen

## Paradigmenfunktion

Eine Paradigmenfunktion ist eine Funktion in der Menge der FPSPs (form/property-set pairings), die auf einem Wurzelpaar  $\langle X, \sigma \rangle$  appliziert (wobei  $X$  die Wurzel eines Lexems  $L$  ist und  $\sigma$  eine vollständige Menge morphosyntaktischer Eigenschaften für  $L$  ist) und eine  $\sigma$ -Zelle  $\langle Y, \sigma \rangle$  im Paradigma von  $L$  ergibt.

(12) Format von Paradigmenfunktionen:

$$PF(\langle X, \sigma \rangle) = \langle Y, \sigma \rangle$$

# Einige Begriffe und Definitionen

## Paradigmenfunktion

Eine Paradigmenfunktion ist eine Funktion in der Menge der FPSPs (form/property-set pairings), die auf einem Wurzelpaar  $\langle X, \sigma \rangle$  appliziert (wobei  $X$  die Wurzel eines Lexems  $L$  ist und  $\sigma$  eine vollständige Menge morphosyntaktischer Eigenschaften für  $L$  ist) und eine  $\sigma$ -Zelle  $\langle Y, \sigma \rangle$  im Paradigma von  $L$  ergibt.

(12) Format von Paradigmenfunktionen:

$$PF(\langle X, \sigma \rangle) = \langle Y, \sigma \rangle$$

## Realisierungsregel (*realization rules, rules of exponence*)

Eine Realisierungsregel ist eine Funktion in der Menge der FPSPs. Im Unterschied zu einer Paradigmenfunktion muss aber das Argument nicht unbedingt ein Wurzelpaar sein, und der Wert muss nicht unbedingt eine Paradigmenzelle sein.

(13) Format von Realisierungsregeln:

$$RR_{n,\tau,C}(\langle X, \sigma \rangle) = \langle Y', \sigma \rangle$$

# Einige Begriffe und Definitionen

(14) Format von Realisierungsregeln:

$$RR_{n,\tau,C}(\langle X,\sigma \rangle) = \langle Y',\sigma \rangle$$

- ❖  $n$ : Blockindex
- ❖  $\tau$ : Eigenschaftsmengenindex (die wohlgeformte Menge morphosyntaktischer Eigenschaften, die die Regel durch ihre Anwendung realisiert;  $\sigma$  muss Extension von  $\tau$  sein  $\rightarrow$  Unterspezifikation!)
- ❖  $C$ : Klassenindex (Klasse der Lexeme, deren Paradigmen die Regel mit definieren kann)
- ❖  $Y'$ : Im Default  $Y$ , aber Möglichkeit der Überschreibung durch morphophonologische Regeln

# Einige Begriffe und Definitionen

## Wohlgeformte Menge morphosyntaktischer Eigenschaften

Eine Menge  $\tau$  von morphosyntaktischen Eigenschaften für ein Lexem der Kategorie C ist wohlgeformt in einer Sprache L nur dann, wenn  $\tau$  die folgenden Bedingungen in L erfüllt.

- a. Für jede Eigenschaft  $F:v \in \tau$  gilt:  $F:v$  ist für Lexeme der Kategorie C zugänglich und  $v$  ist ein erlaubter Wert für  $F$ .
- b. Für jedes morphosyntaktische Merkmal  $F$ , das  $v_1, v_2$  als mögliche Werte hat, gilt: Wenn  $v_1 \neq v_2$  und  $F:v_1 \in \tau$ , dann  $F:v_2 \notin \tau$ .

# Einige Begriffe und Definitionen

## Wohlgeformte Menge morphosyntaktischer Eigenschaften

Eine Menge  $\tau$  von morphosyntaktischen Eigenschaften für ein Lexem der Kategorie C ist wohlgeformt in einer Sprache L nur dann, wenn  $\tau$  die folgenden Bedingungen in L erfüllt.

- Für jede Eigenschaft  $F:v \in \tau$  gilt:  $F:v$  ist für Lexeme der Kategorie C zugänglich und  $v$  ist ein erlaubter Wert für  $F$ .
- Für jedes morphosyntaktische Merkmal  $F$ , das  $v_1, v_2$  als mögliche Werte hat, gilt: Wenn  $v_1 \neq v_2$  und  $F:v_1 \in \tau$ , dann  $F:v_2 \notin \tau$ .

## Extension

Falls  $\sigma$  und  $\tau$  wohlgeformte Mengen morphosyntaktischer Eigenschaften sind, ist  $\sigma$  eine Extension von  $\tau$  gdw. (a.) und (b.) gelten.

- Für jedes atomwertige Merkmal  $F$  und jeden erlaubten Wert  $v$  für  $F$  gilt: Wenn  $F:v \in \tau$ , dann  $F:v \in \sigma$ .
- Für jedes mengenwertige Merkmal  $F$  und jeden erlaubten Wert  $p$  für  $F$  gilt: Wenn  $F:p \in \tau$ , dann  $F:p' \in \sigma$ , wobei  $p'$  eine Extension von  $p$  ist.

# Einige Begriffe und Definitionen

## Wohlgeformte Menge morphosyntaktischer Eigenschaften

Eine Menge  $\tau$  von morphosyntaktischen Eigenschaften für ein Lexem der Kategorie C ist wohlgeformt in einer Sprache L nur dann, wenn  $\tau$  die folgenden Bedingungen in L erfüllt.

- Für jede Eigenschaft  $F:v \in \tau$  gilt:  $F:v$  ist für Lexeme der Kategorie C zugänglich und  $v$  ist ein erlaubter Wert für  $F$ .
- Für jedes morphosyntaktische Merkmal  $F$ , das  $v_1, v_2$  als mögliche Werte hat, gilt: Wenn  $v_1 \neq v_2$  und  $F:v_1 \in \tau$ , dann  $F:v_2 \notin \tau$ .

## Extension

Falls  $\sigma$  und  $\tau$  wohlgeformte Mengen morphosyntaktischer Eigenschaften sind, ist  $\sigma$  eine Extension von  $\tau$  gdw. (a.) und (b.) gelten.

- Für jedes atomwertige Merkmal  $F$  und jeden erlaubten Wert  $v$  für  $F$  gilt: Wenn  $F:v \in \tau$ , dann  $F:v \in \sigma$ .
  - Für jedes mengenwertige Merkmal  $F$  und jeden erlaubten Wert  $p$  für  $F$  gilt: Wenn  $F:p \in \tau$ , dann  $F:p' \in \sigma$ , wobei  $p'$  eine Extension von  $p$  ist.
- z.B.  $\{\text{TNS:pres,AGR:\{PER:1,NUM:pl\}}\}$  ist eine Extension von  $\{\text{AGR:\{PER:1,NUM:pl\}}\}$ ,  $\{\text{AGR:\{NUM:pl\}}\}$ ,  $\{\}$ , usw.

# Einige Begriffe und Definitionen

**Eigenschaftskookkurrenzrestriktionen** (z.B. bulgarische Verbformen)

Eine Menge  $\tau$  von morphosyntaktischen Eigenschaften für ein Lexem der Kategorie V ist wohlgeformt nur, wenn  $\tau$  eine wohlgeformte Extension  $\sigma$  hat, so dass gilt:

# Einige Begriffe und Definitionen

**Eigenschaftskookkurrenzrestriktionen** (z.B. bulgarische Verbformen)

Eine Menge  $\tau$  von morphosyntaktischen Eigenschaften für ein Lexem der Kategorie V ist wohlgeformt nur, wenn  $\tau$  eine wohlgeformte Extension  $\sigma$  hat, so dass gilt:

- a.  $\sigma$  ist eine Extension von  $\{\text{VFORM:fin}\}$  gdw. für ein zulässiges  $\alpha$  gilt:  $\sigma$  ist eine Extension von  $\{\text{MOOD}:\alpha\}$ .



# Einige Begriffe und Definitionen

**Eigenschaftskookkurrenzrestriktionen** (z.B. bulgarische Verbformen)

Eine Menge  $\tau$  von morphosyntaktischen Eigenschaften für ein Lexem der Kategorie V ist wohlgeformt nur, wenn  $\tau$  eine wohlgeformte Extension  $\sigma$  hat, so dass gilt:

- a.  $\sigma$  ist eine Extension von  $\{\text{VFORM:fin}\}$  gdw. für ein zulässiges  $\alpha$  gilt:  $\sigma$  ist eine Extension von  $\{\text{MOOD}:\alpha\}$ .

(wenn Finitheit, dann Modus (Ind oder Konj))

# Einige Begriffe und Definitionen

**Eigenschaftskookkurrenzrestriktionen** (z.B. bulgarische Verbformen)

Eine Menge  $\tau$  von morphosyntaktischen Eigenschaften für ein Lexem der Kategorie V ist wohlgeformt nur, wenn  $\tau$  eine wohlgeformte Extension  $\sigma$  hat, so dass gilt:

- a.  $\sigma$  ist eine Extension von  $\{\text{VFORM:fin}\}$  gdw. für ein zulässiges  $\alpha$  gilt:  $\sigma$  ist eine Extension von  $\{\text{MOOD}:\alpha\}$ .  
(wenn Finitheit, dann Modus (Ind oder Konj))
- b. Wenn  $\sigma$  eine Extension ist von  $\{\text{MOOD:impv}\}$ , dann ist  $\sigma$  eine Extension von  $\{\text{AGR}:\{\text{PER:2}\}\}$ .

# Einige Begriffe und Definitionen

**Eigenschaftskookkurrenzrestriktionen** (z.B. bulgarische Verbformen)

Eine Menge  $\tau$  von morphosyntaktischen Eigenschaften für ein Lexem der Kategorie V ist wohlgeformt nur, wenn  $\tau$  eine wohlgeformte Extension  $\sigma$  hat, so dass gilt:

- a.  $\sigma$  ist eine Extension von  $\{\text{VFORM:fin}\}$  gdw. für ein zulässiges  $\alpha$  gilt:  $\sigma$  ist eine Extension von  $\{\text{MOOD}:\alpha\}$ .  
(wenn Finitheit, dann Modus (Ind oder Konj))
- b. Wenn  $\sigma$  eine Extension ist von  $\{\text{MOOD:impv}\}$ , dann ist  $\sigma$  eine Extension von  $\{\text{AGR}:\{\text{PER:2}\}\}$ .  
(wenn Imperativ, dann 2. Person)

# Einige Begriffe und Definitionen

**Eigenschaftskookkurrenzrestriktionen** (z.B. bulgarische Verbformen)

Eine Menge  $\tau$  von morphosyntaktischen Eigenschaften für ein Lexem der Kategorie V ist wohlgeformt nur, wenn  $\tau$  eine wohlgeformte Extension  $\sigma$  hat, so dass gilt:

- a.  $\sigma$  ist eine Extension von  $\{\text{VFORM:fin}\}$  gdw. für ein zulässiges  $\alpha$  gilt:  $\sigma$  ist eine Extension von  $\{\text{MOOD}:\alpha\}$ .  
(wenn Finitheit, dann Modus (Ind oder Konj))
- b. Wenn  $\sigma$  eine Extension ist von  $\{\text{MOOD:impv}\}$ , dann ist  $\sigma$  eine Extension von  $\{\text{AGR}:\{\text{PER}:2\}\}$ .  
(wenn Imperativ, dann 2. Person)
- c. Für jedes zulässige  $\alpha$  gilt:  $\sigma$  ist eine Extension von  $\{\text{TNS}:\alpha\}$  gdw.  $\sigma$  eine Extension ist von  $\{\text{MOOD:ind}\}$  oder von  $\{\text{VFORM:ptcp}\}$ .

# Einige Begriffe und Definitionen

**Eigenschaftskookkurrenzrestriktionen** (z.B. bulgarische Verbformen)

Eine Menge  $\tau$  von morphosyntaktischen Eigenschaften für ein Lexem der Kategorie V ist wohlgeformt nur, wenn  $\tau$  eine wohlgeformte Extension  $\sigma$  hat, so dass gilt:

- a.  $\sigma$  ist eine Extension von  $\{\text{VFORM:fin}\}$  gdw. für ein zulässiges  $\alpha$  gilt:  $\sigma$  ist eine Extension von  $\{\text{MOOD}:\alpha\}$ .  
(wenn Finitheit, dann Modus (Ind oder Konj))
- b. Wenn  $\sigma$  eine Extension ist von  $\{\text{MOOD:impv}\}$ , dann ist  $\sigma$  eine Extension von  $\{\text{AGR}:\{\text{PER}:2\}\}$ .  
(wenn Imperativ, dann 2. Person)
- c. Für jedes zulässige  $\alpha$  gilt:  $\sigma$  ist eine Extension von  $\{\text{TNS}:\alpha\}$  gdw.  $\sigma$  eine Extension ist von  $\{\text{MOOD:ind}\}$  oder von  $\{\text{VFORM:ptcp}\}$ .  
(V hat Tempus, wenn es Ind. oder Partizip ist)

# Einige Begriffe und Definitionen

**Eigenschaftskookkurrenzrestriktionen** (z.B. bulgarische Verbformen)

Eine Menge  $\tau$  von morphosyntaktischen Eigenschaften für ein Lexem der Kategorie V ist wohlgeformt nur, wenn  $\tau$  eine wohlgeformte Extension  $\sigma$  hat, so dass gilt:

- $\sigma$  ist eine Extension von  $\{\text{VFORM:fin}\}$  gdw. für ein zulässiges  $\alpha$  gilt:  $\sigma$  ist eine Extension von  $\{\text{MOOD}:\alpha\}$ .  
(wenn Finitheit, dann Modus (Ind oder Konj))
- Wenn  $\sigma$  eine Extension ist von  $\{\text{MOOD:impv}\}$ , dann ist  $\sigma$  eine Extension von  $\{\text{AGR}:\{\text{PER}:2\}\}$ .  
(wenn Imperativ, dann 2. Person)
- Für jedes zulässige  $\alpha$  gilt:  $\sigma$  ist eine Extension von  $\{\text{TNS}:\alpha\}$  gdw.  $\sigma$  eine Extension ist von  $\{\text{MOOD:ind}\}$  oder von  $\{\text{VFORM:ptcp}\}$ .  
(V hat Tempus, wenn es Ind. oder Partizip ist)
- Für jedes zulässige  $\alpha$  gilt:  $\sigma$  ist eine Extension von  $\{\text{AGR}:\{\text{GEN}:\alpha\}\}$  gdw.  $\sigma$  eine Extension ist von  $\{\text{VFORM:ptcp}\}$ , und  $\sigma$  ist eine Extension von  $\{\text{AGR}:\{\text{PER}:\alpha\}\}$  gdw.  $\sigma$  eine Extension ist von  $\{\text{VFORM:fin}\}$ .

# Einige Begriffe und Definitionen

**Eigenschaftskookkurrenzrestriktionen** (z.B. bulgarische Verbformen)

Eine Menge  $\tau$  von morphosyntaktischen Eigenschaften für ein Lexem der Kategorie V ist wohlgeformt nur, wenn  $\tau$  eine wohlgeformte Extension  $\sigma$  hat, so dass gilt:

- $\sigma$  ist eine Extension von  $\{\text{VFORM:fin}\}$  gdw. für ein zulässiges  $\alpha$  gilt:  $\sigma$  ist eine Extension von  $\{\text{MOOD}:\alpha\}$ .  
(wenn Finitheit, dann Modus (Ind oder Konj))
- Wenn  $\sigma$  eine Extension ist von  $\{\text{MOOD:impv}\}$ , dann ist  $\sigma$  eine Extension von  $\{\text{AGR}:\{\text{PER}:2\}\}$ .  
(wenn Imperativ, dann 2. Person)
- Für jedes zulässige  $\alpha$  gilt:  $\sigma$  ist eine Extension von  $\{\text{TNS}:\alpha\}$  gdw.  $\sigma$  eine Extension ist von  $\{\text{MOOD:ind}\}$  oder von  $\{\text{VFORM:ptcp}\}$ .  
(V hat Tempus, wenn es Ind. oder Partizip ist)
- Für jedes zulässige  $\alpha$  gilt:  $\sigma$  ist eine Extension von  $\{\text{AGR}:\{\text{GEN}:\alpha\}\}$  gdw.  $\sigma$  eine Extension ist von  $\{\text{VFORM:ptcp}\}$ , und  $\sigma$  ist eine Extension von  $\{\text{AGR}:\{\text{PER}:\alpha\}\}$  gdw.  $\sigma$  eine Extension ist von  $\{\text{VFORM:fin}\}$ .  
(Wenn Genus, dann Partizip; wenn Person, dann Finitheit)

# Einige Begriffe und Defintionen

## Unifikation

Falls  $\sigma$  und  $\tau$  wohlgeformte Mengen morphosyntaktischer Merkmale sind, ist die Unifikation  $\rho$  von  $\sigma$  und  $\tau$  die kleinste wohlgeformte Menge von morphosyntaktischen Eigenschaften, so dass  $\rho$  eine Extension sowohl von  $\sigma$  als auch von  $\tau$  ist.



# Einige Begriffe und Definitionen

## Unifikation

Falls  $\sigma$  und  $\tau$  wohlgeformte Mengen morphosyntaktischer Merkmale sind, ist die Unifikation  $\rho$  von  $\sigma$  und  $\tau$  die kleinste wohlgeformte Menge von morphosyntaktischen Eigenschaften, so dass  $\rho$  eine Extension sowohl von  $\sigma$  als auch von  $\tau$  ist.

- ❖  $\{\text{TNS:pres,MOOD:ind,AGR:\{PER:1,NUM:pl\}}\}$  ist die Unifikation von  $\{\text{TNS:pres,AGR:\{PER:1\}}\}$  und  $\{\text{TNS:pres,MOOD:ind,AGR:\{NUM:pl\}}\}$

# Einige Begriffe und Definitionen

## Unifikation

Falls  $\sigma$  und  $\tau$  wohlgeformte Mengen morphosyntaktischer Merkmale sind, ist die Unifikation  $\rho$  von  $\sigma$  und  $\tau$  die kleinste wohlgeformte Menge von morphosyntaktischen Eigenschaften, so dass  $\rho$  eine Extension sowohl von  $\sigma$  als auch von  $\tau$  ist.

- ❖  $\{\text{TNS:pres,MOOD:ind,AGR:}\{\text{PER:1,NUM:pl}\}\}$  ist die Unifikation von  $\{\text{TNS:pres,AGR:}\{\text{PER:1}\}\}$  und  $\{\text{TNS:pres,MOOD:ind,AGR:}\{\text{NUM:pl}\}\}$

## Vollständigkeit

Eine Menge  $\sigma$  von morphosyntaktischen Merkmalen für ein Lexem einer Kategorie ist vollständig gdw. (a.) und (b.) gelten:

# Einige Begriffe und Definitionen

## Unifikation

Falls  $\sigma$  und  $\tau$  wohlgeformte Mengen morphosyntaktischer Merkmale sind, ist die Unifikation  $\rho$  von  $\sigma$  und  $\tau$  die kleinste wohlgeformte Menge von morphosyntaktischen Eigenschaften, so dass  $\rho$  eine Extension sowohl von  $\sigma$  als auch von  $\tau$  ist.

- ❖  $\{\text{TNS:pres,MOOD:ind,AGR:}\{\text{PER:1,NUM:pl}\}\}$  ist die Unifikation von  $\{\text{TNS:pres,AGR:}\{\text{PER:1}\}\}$  und  $\{\text{TNS:pres,MOOD:ind,AGR:}\{\text{NUM:pl}\}\}$

## Vollständigkeit

Eine Menge  $\sigma$  von morphosyntaktischen Merkmalen für ein Lexem einer Kategorie ist vollständig gdw. (a.) und (b.) gelten:

- a.  $\sigma$  ist wohlgeformt.

# Einige Begriffe und Definitionen

## Unifikation

Falls  $\sigma$  und  $\tau$  wohlgeformte Mengen morphosyntaktischer Merkmale sind, ist die Unifikation  $\rho$  von  $\sigma$  und  $\tau$  die kleinste wohlgeformte Menge von morphosyntaktischen Eigenschaften, so dass  $\rho$  eine Extension sowohl von  $\sigma$  als auch von  $\tau$  ist.

- ❖  $\{\text{TNS:pres,MOOD:ind,AGR:\{PER:1,NUM:pl\}}\}$  ist die Unifikation von  $\{\text{TNS:pres,AGR:\{PER:1\}}\}$  und  $\{\text{TNS:pres,MOOD:ind,AGR:\{NUM:pl\}}\}$

## Vollständigkeit

Eine Menge  $\sigma$  von morphosyntaktischen Merkmalen für ein Lexem einer Kategorie ist vollständig gdw. (a.) und (b.) gelten:

- a.  $\sigma$  ist wohlgeformt.
- b. Für jede Menge  $\tau$  von morphosyntaktischen Merkmalen (so dass  $\sigma$  nicht eine Extension von  $\tau$  ist) gilt: die Unifikation von  $\tau$  und  $\sigma$  ist nicht wohlgeformt.

# Beispielanalyse: Bulgarische Verbflexion

# Konjugation von *krad-* ‘stehen’

Im Folgenden werden wir die besprochenen Prinzipien in Aktion sehen. Als Beispiel dient uns die Flexion des bulgarischen Verbs *krad-* ‘stehen’.

## (15) *Bulgarische Verbflexion*

	Präsens	Imperfekt	Aorist
1sg	krad-e-ə	krad-A-x	krád-o-x
2sg	krad-e-š	krad-A-x-e	krád-e
3sg	krad-e-e	krad-A-x-e	krád-e
1pl	krad-e-m	krad-A-x-me	krád-o-x-me
2pl	krad-e-te	krad-A-x-te	krád-o-x-te
3pl	krad-e-ət	krad-A-x-a	krád-o-x-a

# Regelblöcke

Die Regeln werden in vier Regelblöcke eingeteilt, um Formen wie *krád-o-x-me* abzuleiten.

# Regelblöcke

Die Regeln werden in vier Regelblöcke eingeteilt, um Formen wie *krád-o-x-me* abzuleiten.

- ❖ Der erste Regelblock selektiert dabei den Stamm, da das Verb zwischen den beiden Stämmen *krad* (default) und *krád* (aorist) alterniert.



# Regelblöcke

Die Regeln werden in vier Regelblöcke eingeteilt, um Formen wie *krád-o-x-me* abzuleiten.

- ❖ Der erste Regelblock selektiert dabei den Stamm, da das Verb zwischen den beiden Stämmen *krad* (default) und *krád* (aorist) alterniert.
- ❖ Die anderen Regelblöcke applizieren in der Reihenfolge der Templates (also in der Oberflächenreihenfolge der Morpheme).

# Regelblöcke

Die Regeln werden in vier Regelblöcke eingeteilt, um Formen wie *krád-o-x-me* abzuleiten.

- ❖ Der erste Regelblock selektiert dabei den Stamm, da das Verb zwischen den beiden Stämmen *krad* (default) und *krád* (aorist) alterniert.
- ❖ Die anderen Regelblöcke applizieren in der Reihenfolge der Templates (also in der Oberflächenreihenfolge der Morpheme).
- ❖ Im Gegensatz zu DM ist es dabei kein Problem, wenn mehrere Morpheme sich auf dieselben Merkmale beziehen, weil es in PFM keine Abarbeitung der Merkmale gibt.

# Realisierungsregeln

- (16) a.  $RR_{1, \{T:Aor\}, V}(\langle X, \sigma \rangle) = \langle Y', \sigma \rangle$  wobei Y der 2. Stamm von X ist.  
 b.  $RR_{1, \{ \}, V}(\langle X, \sigma \rangle) = \langle Y', \sigma \rangle$  wobei Y der 1. Stamm von X ist.

# Realisierungsregeln

- (16) a.  $RR_{1,\{T:Aor\},V}(\langle X,\sigma \rangle) = \langle Y',\sigma \rangle$  wobei Y der 2. Stamm von X ist.  
 b.  $RR_{1,\{\},V}(\langle X,\sigma \rangle) = \langle Y',\sigma \rangle$  wobei Y der 1. Stamm von X ist.
- (17) a.  $RR_{2,\{T:Präsens\},V}(\langle X,\sigma \rangle) = \langle Xe',\sigma \rangle$   
 b.  $RR_{2,\{T:Imperf\},V}(\langle X,\sigma \rangle) = \langle XA',\sigma \rangle$   
 c.  $RR_{2,\{T:Aorist\},V}(\langle X,\sigma \rangle) = \langle Xo',\sigma \rangle$   
 d.  $RR_{[2\vee 3],\{T:Aorist,Pret:yes,3sg\},V}(\langle X,\sigma \rangle) = \langle X',\sigma \rangle$

# Realisierungsregeln

- (16) a.  $RR_{1,\{T:Aor\},V}(\langle X,\sigma \rangle) = \langle Y',\sigma \rangle$  wobei Y der 2. Stamm von X ist.  
 b.  $RR_{1,\{\},V}(\langle X,\sigma \rangle) = \langle Y',\sigma \rangle$  wobei Y der 1. Stamm von X ist.
- (17) a.  $RR_{2,\{T:Präsens\},V}(\langle X,\sigma \rangle) = \langle Xe',\sigma \rangle$   
 b.  $RR_{2,\{T:Imperf\},V}(\langle X,\sigma \rangle) = \langle XA',\sigma \rangle$   
 c.  $RR_{2,\{T:Aorist\},V}(\langle X,\sigma \rangle) = \langle Xo',\sigma \rangle$   
 d.  $RR_{[2\vee 3],\{T:Aorist,Pret:yes,3sg\},V}(\langle X,\sigma \rangle) = \langle X',\sigma \rangle$
- (18) a.  $RR_{3,\{Pret:yes\},V}(\langle X,\sigma \rangle) = \langle Xx',\sigma \rangle$

# Realisierungsregeln

- (16) a.  $RR_{1, \{T:Aor\}, V}(\langle X, \sigma \rangle) = \langle Y', \sigma \rangle$  wobei Y der 2. Stamm von X ist.  
 b.  $RR_{1, \{ \}, V}(\langle X, \sigma \rangle) = \langle Y', \sigma \rangle$  wobei Y der 1. Stamm von X ist.
- (17) a.  $RR_{2, \{T:Präsens\}, V}(\langle X, \sigma \rangle) = \langle Xe', \sigma \rangle$   
 b.  $RR_{2, \{T:Imperf\}, V}(\langle X, \sigma \rangle) = \langle XA', \sigma \rangle$   
 c.  $RR_{2, \{T:Aorist\}, V}(\langle X, \sigma \rangle) = \langle Xo', \sigma \rangle$   
 d.  $RR_{[2V3], \{T:Aorist, Pret:yes, 3sg\}, V}(\langle X, \sigma \rangle) = \langle X', \sigma \rangle$
- (18) a.  $RR_{3, \{Pret:yes\}, V}(\langle X, \sigma \rangle) = \langle Xx', \sigma \rangle$
- (19) a.  $RR_{4, \{T:Präsens, 1sg\}, V}(\langle X, \sigma \rangle) = \langle X\partial', \sigma \rangle$   
 b.  $RR_{4, \{T:Präsens, 2sg\}, V}(\langle X, \sigma \rangle) = \langle X\check{s}', \sigma \rangle$   
 c.  $RR_{4, \{3sg\}, V}(\langle X, \sigma \rangle) = \langle Xe', \sigma \rangle$   
 d.  $RR_{4, \{T:Präsens, 1pl\}, V}(\langle X, \sigma \rangle) = \langle Xm', \sigma \rangle$   
 e.  $RR_{4, \{1pl\}, V}(\langle X, \sigma \rangle) = \langle Xme', \sigma \rangle$   
 f.  $RR_{4, \{2pl\}, V}(\langle X, \sigma \rangle) = \langle Xte', \sigma \rangle$   
 g.  $RR_{4, \{T:Präsens, 3pl\}, V}(\langle X, \sigma \rangle) = \langle X\partial t', \sigma \rangle$   
 h.  $RR_{4, \{3pl\}, V}(\langle X, \sigma \rangle) = \langle Xa', \sigma \rangle$

# Bemerkungen

- ❖ Regeln wie (16-b) sind Identitätsfunktionen, die den Zweck eines radikal unterspezifizierten Markers erfüllen.
  - ▶ Auch in PFM gilt, dass der Default nicht immer zwingend morphologisch unmarkiert sein muss.
  - ▶ In PFM wird oft auch angenommen, dass es eine universelle Identitätsfunktion gibt, die garantiert, dass die Derivation voranschreitet, auch wenn in einem Block keine Regel angewendet werden kann.

$$(20) \quad \text{RR}_{n,\{\},\cup}(\langle X,\sigma \rangle) = \langle X,\sigma \rangle$$

# Bemerkungen

- ❖ Regeln wie (16-b) sind Identitätsfunktionen, die den Zweck eines radikal unterspezifizierten Markers erfüllen.
  - ▶ Auch in PFM gilt, dass der Default nicht immer zwingend morphologisch unmarkiert sein muss.
  - ▶ In PFM wird oft auch angenommen, dass es eine universelle Identitätsfunktion gibt, die garantiert, dass die Derivation voranschreitet, auch wenn in einem Block keine Regel angewendet werden kann.

$$(20) \quad RR_{n,\{\},\cup}(\langle X,\sigma \rangle) = \langle X,\sigma \rangle$$

- ❖ Die Disjunktion in Regel (17-d) ist nötig, da sichergestellt werden muss, dass sie einerseits Regel (17-c) blockiert, aber auch die Applikation von Regel (18-a) blockieren muss.
  - ▶ Solche Disjunktionen sprechen oft für ungelöste Probleme (z.B. unaufgelöste Synkretismen), da sie letztendlich in zwei Regeln dekomponiert werden müssen.
  - ▶ In diesem Fall müsste Regel (17-d) zweimal Teil der Grammatik sein; einmal in Block 2 und einmal in Block 3.



# Bemerkungen

- ❖ Die 2. Person Singular im Aorist hat nach den Realisierungsregeln der Blöcke 1–4 noch die falsche Form, nämlich *krád-o-x*.

# Bemerkungen

- ❖ Die 2. Person Singular im Aorist hat nach den Realisierungsregeln der Blöcke 1–4 noch die falsche Form, nämlich *krád-o-x*.
- ▶ Die korrekte Form wird mithilfe einer Verweisregel gewährleistet.

(21) *Verweisregel (informell)*

Im Präteritum (Imperfekt und Aorist, also: {PRET:yes}) richtet sich die Form der 2. Person Singular immer nach der Form der 3. Person Singular.

# Bemerkungen

- ❖ Die 2. Person Singular im Aorist hat nach den Realisierungsregeln der Blöcke 1–4 noch die falsche Form, nämlich *krád-o-x*.
- ▶ Die korrekte Form wird mithilfe einer Verweisregel gewährleistet.

(21) *Verweisregel (informell)*

Im Präteritum (Imperfekt und Aorist, also: {PRET:yes}) richtet sich die Form der 2. Person Singular immer nach der Form der 3. Person Singular.

- ▶ Diese Regel ist in keiner Weise beschränkt. Sie könnte [2.SG,Aorist] ebensogut nach [1.PL,Präsens] richten lassen.

# Bemerkungen

- ❖ Die 2. Person Singular im Aorist hat nach den Realisierungsregeln der Blöcke 1–4 noch die falsche Form, nämlich *krád-o-x*.
- ▶ Die korrekte Form wird mithilfe einer Verweisregel gewährleistet.

(21) *Verweisregel (informell)*

Im Präteritum (Imperfekt und Aorist, also: {PRET:yes}) richtet sich die Form der 2. Person Singular immer nach der Form der 3. Person Singular.

- ▶ Diese Regel ist in keiner Weise beschränkt. Sie könnte [2.SG,Aorist] ebenso gut nach [1.PL,Präsens] richten lassen.
  - ▶ Dies macht die Mächtigkeit derartiger Verweisregeln deutlich.
  - ▶ Jede Zelle eines Paradigmas lässt sich so mit jeder anderen Zelle in Bezug setzen.
  - ▶ Es ist demnach kein Synkretismus unableitbar bzw. es gibt keine unmöglichen Synkretismen.

# Nar<sub>n</sub>-Notation, Enge und Anwendbarkeit

## Pāṇinis Prinzip

Sei  $\sigma$  eine vollständige Menge von morphosyntaktischen Eigenschaften für Lexeme der Kategorie V. Dann ist  $PF(\langle X, \sigma \rangle) = \text{Nar}_4(\text{Nar}_3(\text{Nar}_2(\text{Nar}_1(\langle X, \sigma \rangle))))$ .

# Nar<sub>n</sub>-Notation, Enge und Anwendbarkeit

## Pāṇinis Prinzip

Sei  $\sigma$  eine vollständige Menge von morphosyntaktischen Eigenschaften für Lexeme der Kategorie V. Dann ist  $PF(\langle X, \sigma \rangle) = \text{Nar}_4(\text{Nar}_3(\text{Nar}_2(\text{Nar}_1(\langle X, \sigma \rangle))))$ .

## Nar<sub>n</sub>-Notation

Ist  $RR_{n,\tau,C}$  die **engste** Regel in Block  $n$ , die auf  $\langle X, \sigma \rangle$  **anwendbar** ist, so repräsentiert 'Nar<sub>n</sub>( $\langle X, \sigma \rangle$ )' das Resultat der Anwendung von  $RR_{n,\tau,C}$  auf  $\langle X, \sigma \rangle$ .

# Nar<sub>n</sub>-Notation, Enge und Anwendbarkeit

## Pāṇinis Prinzip

Sei  $\sigma$  eine vollständige Menge von morphosyntaktischen Eigenschaften für Lexeme der Kategorie V. Dann ist  $PF(\langle X, \sigma \rangle) = \text{Nar}_4(\text{Nar}_3(\text{Nar}_2(\text{Nar}_1(\langle X, \sigma \rangle))))$ .

## Nar<sub>n</sub>-Notation

Ist  $RR_{n,\tau,C}$  die **engste** Regel in Block  $n$ , die auf  $\langle X, \sigma \rangle$  **anwendbar** ist, so repräsentiert 'Nar<sub>n</sub>( $\langle X, \sigma \rangle$ )' das Resultat der Anwendung von  $RR_{n,\tau,C}$  auf  $\langle X, \sigma \rangle$ .

## Enge und Anwendbarkeit

- $RR_{n,\sigma,C}$  ist enger als  $RR_{n,\tau,C}$  gdw.  $\sigma$  eine Extension von  $\tau$  ist und  $\sigma \neq \tau$ .
- $RR_{n,\tau,C}$  ist anwendbar auf  $\langle X, \sigma \rangle$  gdw.  $RR_{n,\tau,C}(\langle X, \sigma \rangle)$  definiert ist.

# Nar<sub>n</sub>-Notation, Enge und Anwendbarkeit

## Pāṇinis Prinzip

Sei  $\sigma$  eine vollständige Menge von morphosyntaktischen Eigenschaften für Lexeme der Kategorie V. Dann ist  $\text{PF}(\langle X, \sigma \rangle) = \text{Nar}_4(\text{Nar}_3(\text{Nar}_2(\text{Nar}_1(\langle X, \sigma \rangle))))$ .

## Nar<sub>n</sub>-Notation

Ist  $\text{RR}_{n,\tau,C}$  die **engste** Regel in Block  $n$ , die auf  $\langle X, \sigma \rangle$  **anwendbar** ist, so repräsentiert 'Nar<sub>n</sub>( $\langle X, \sigma \rangle$ )' das Resultat der Anwendung von  $\text{RR}_{n,\tau,C}$  auf  $\langle X, \sigma \rangle$ .

## Enge und Anwendbarkeit

- $\text{RR}_{n,\sigma,C}$  ist enger als  $\text{RR}_{n,\tau,C}$  gdw.  $\sigma$  eine Extension von  $\tau$  ist und  $\sigma \neq \tau$ .
- $\text{RR}_{n,\tau,C}$  ist anwendbar auf  $\langle X, \sigma \rangle$  gdw.  $\text{RR}_{n,\tau,C}(\langle X, \sigma \rangle)$  definiert ist.

## Regel-Argument-Kohärenz

$\text{RR}_{n,\tau,C}(\langle X, \sigma \rangle)$  ist definiert gdw.

- $\sigma$  eine Extension von  $\tau$  ist (s.o.);
- L-Index(X)  $\in C$  ist; und
- $\sigma$  eine wohlgeformte Menge von morphosyntaktischen Eigenschaften für L-Index(X) ist.



# Formelle Verweisregel

Angenommen (a)–(c) sind der Fall:

- a.  $\tau$  ist eine beliebige vollständige Extension von  $\{\text{PRET:yes,AGR:\{PER:2,NUM:sg\}}\}$ .
- b.  $n$  ist ein beliebiger Regelblock in 1–4.
- c.  $\sigma' = \sigma / \{\text{AGR:\{PER:3\}}\}$ . (lies:  $\sigma$  modifiziert durch  $\{\text{AGR:\{PER:3\}}\}$ )

Dann gilt:

$$\text{RR}_{n,\tau,v}(\langle X,\sigma \rangle) = \langle Y,\sigma \rangle, \text{ wobei } \text{Nar}_n(\langle X,\sigma' \rangle) = \langle Y,\sigma' \rangle$$

# Expandierter Modus

# Argumentkodierung im Georgischen

Der Wettbewerb zwischen Realisierungsregeln in einem Block wird durch die spezifischste (engste) Regel gewonnen (Pāṇinis Prinzip).

# Argumentkodierung im Georgischen

Der Wettbewerb zwischen Realisierungsregeln in einem Block wird durch die spezifischste (engste) Regel gewonnen (Pāṇinis Prinzip).

- ❖ Damit kann es aber Probleme geben. Es muss mehr gesagt werden.

# Argumentkodierung im Georgischen

Der Wettbewerb zwischen Realisierungsregeln in einem Block wird durch die spezifischste (engste) Regel gewonnen (Pāṇinis Prinzip).

- ❖ Damit kann es aber Probleme geben. Es muss mehr gesagt werden.

Ausschnitt der Realisationsregeln für argumentkodierende Präfixe im Georgischen in (22) (Stump 2001:70).

- (22)
- a.  $RR_{\text{pref},\{\text{AGR}(\text{SU});\{\text{PER}:1\}\},V}(\langle X,\sigma \rangle) = \langle vX', \sigma \rangle$
  - b.  $RR_{\text{pref},\{\text{AGR}(\text{OB});\{\text{PER}:1\}\},V}(\langle X,\sigma \rangle) = \langle mX', \sigma \rangle$
  - c.  $RR_{\text{pref},\{\text{AGR}(\text{OB});\{\text{PER}:1,\text{NUM}:p\}\},V}(\langle X,\sigma \rangle) = \langle gvX', \sigma \rangle$
  - d.  $RR_{\text{pref},\{\text{AGR}(\text{OB});\{\text{PER}:2\}\},V}(\langle X,\sigma \rangle) = \langle gX', \sigma \rangle$

# Argumentkodierung im Georgischen

Der Wettbewerb zwischen Realisierungsregeln in einem Block wird durch die spezifischste (engste) Regel gewonnen (Pāṇinis Prinzip).

❖ Damit kann es aber Probleme geben. Es muss mehr gesagt werden.

Ausschnitt der Realisationsregeln für argumentkodierende Präfixe im Georgischen in (22) (Stump 2001:70).

- (22) a.  $RR_{\text{pref},\{\text{AGR}(\text{SU}):\{\text{PER}:1\}\},V}(\langle X,\sigma \rangle) = \langle vX', \sigma \rangle$   
 b.  $RR_{\text{pref},\{\text{AGR}(\text{OB}):\{\text{PER}:1\}\},V}(\langle X,\sigma \rangle) = \langle mX', \sigma \rangle$   
 c.  $RR_{\text{pref},\{\text{AGR}(\text{OB}):\{\text{PER}:1,\text{NUM}:p\}\},V}(\langle X,\sigma \rangle) = \langle gvX', \sigma \rangle$   
 d.  $RR_{\text{pref},\{\text{AGR}(\text{OB}):\{\text{PER}:2\}\},V}(\langle X,\sigma \rangle) = \langle gX', \sigma \rangle$

Problem: In einem Kontext wie ‘Ich werde dich töten’ sollten sowohl (22-a) als auch (22-d) passen, da beide Merkmalsmengen nicht in einem Extensionsverhältnis stehen. Empirisch korrekt aber wird (22-a) blockiert.

(23)	Präverb	Präfix	Stamm	Suffix	Glosse
	mo-	g-	klav		‘Ich werde dich töten’
	*mo-	v-	klav		‘Ich werde dich töten’
	mo-	g-	klav	-t	‘Ich werde dich töten’

# Lösungen

Extrinsische Regelordnung (Anderson 1992)

Regel (22-d) appliziert per Stipulation vor Regel (22-a).

# Lösungen

**Extrinsische Regelordnung** (Anderson 1992)

Regel (22-d) appliziert per Stipulation vor Regel (22-a).

**Expandierter Modus** (Stump 2001)

Regeln können aufgeblasen werden und sind dann maximal spezifisch.



# Lösungen

**Extrinsische Regelordnung** (Anderson 1992)

Regel (22-d) appliziert per Stipulation vor Regel (22-a).

**Expandierter Modus** (Stump 2001)

Regeln können aufgeblasen werden und sind dann maximal spezifisch.

(24) *Regelformate*

a. Normaler Modus

$$RR_{n,\tau,C}(\langle X,\sigma \rangle) = \langle Y'\sigma \rangle$$

b. Expandierter Modus

$$RR_{n,\leftarrow\tau\rightarrow,C}(\langle X,\sigma \rangle) = \langle Y'\sigma \rangle$$

“ $\leftarrow \tau \rightarrow$ ” bedeutet vereinfacht, dass  $\tau$  maximal erweitert wird.

# Lösungen

## Extrinsische Regelordnung (Anderson 1992)

Regel (22-d) appliziert per Stipulation vor Regel (22-a).

## Expandierter Modus (Stump 2001)

Regeln können aufgeblasen werden und sind dann maximal spezifisch.

### (24) *Regelformate*

a. Normaler Modus

$$RR_{n,\tau,C}(\langle X,\sigma \rangle) = \langle Y'\sigma \rangle$$

b. Expandierter Modus

$$RR_{n,\leftarrow\tau\rightarrow,C}(\langle X,\sigma \rangle) = \langle Y'\sigma \rangle$$

“ $\leftarrow\tau\rightarrow$ ” bedeutet vereinfacht, dass  $\tau$  maximal erweitert wird.

Fazit: Regel (22-d) im Georgischen arbeitet im expandierten Modus:

$$(25) \quad RR_{\text{pref},\leftarrow\{\text{AGR}(\text{ob})\}:\{\text{PER}:2\}\rightarrow,v}(\langle X,\sigma \rangle) = \langle gX',\sigma \rangle$$

# Vergleich mit DM

# Gemeinsamkeiten

Wenn man über den Formalismus abstrahiert und die Theorien mit etwas Abstand betrachtet, fallen einige Gemeinsamkeiten ins Auge:

# Gemeinsamkeiten

Wenn man über den Formalismus abstrahiert und die Theorien mit etwas Abstand betrachtet, fallen einige Gemeinsamkeiten ins Auge:

- ❖ Wie DM benutzt auch PFM Unterspezifikation und das Teilmengenprinzip von Markern bzw. Realisierungsregeln.

# Gemeinsamkeiten

Wenn man über den Formalismus abstrahiert und die Theorien mit etwas Abstand betrachtet, fallen einige Gemeinsamkeiten ins Auge:

- ❖ Wie DM benutzt auch PFM Unterspezifikation und das Teilmengenprinzip von Markern bzw. Realisierungsregeln.
- ❖ Auch in PFM gewinnt bei einem Wettbewerb zwischen verschiedenen passenden Markern/Regeln der/die spezifischste (Pāṇinis Prinzip).

# Gemeinsamkeiten

Wenn man über den Formalismus abstrahiert und die Theorien mit etwas Abstand betrachtet, fallen einige Gemeinsamkeiten ins Auge:

- ❖ Wie DM benutzt auch PFM Unterspezifikation und das Teilmengenprinzip von Markern bzw. Realisierungsregeln.
- ❖ Auch in PFM gewinnt bei einem Wettbewerb zwischen verschiedenen passenden Markern/Regeln der/die spezifischste (Pāṇinis Prinzip).
- ❖ Auch Dekomposition wird in vielen PFM-Analysen vorausgesetzt.

# Gemeinsamkeiten

Wenn man über den Formalismus abstrahiert und die Theorien mit etwas Abstand betrachtet, fallen einige Gemeinsamkeiten ins Auge:

- ❖ Wie DM benutzt auch PFM Unterspezifikation und das Teilmengenprinzip von Markern bzw. Realisierungsregeln.
- ❖ Auch in PFM gewinnt bei einem Wettbewerb zwischen verschiedenen passenden Markern/Regeln der/die spezifischste (Pāṇinis Prinzip).
- ❖ Auch Dekomposition wird in vielen PFM-Analysen vorausgesetzt.
- ❖ Beide Theorien benutzen einen maximal unterspezifizierte Elsewhere-Marker bzw. -Regeln.



# Gemeinsamkeiten

Wenn man über den Formalismus abstrahiert und die Theorien mit etwas Abstand betrachtet, fallen einige Gemeinsamkeiten ins Auge:

- ❖ Wie DM benutzt auch PFM Unterspezifikation und das Teilmengenprinzip von Markern bzw. Realisierungsregeln.
- ❖ Auch in PFM gewinnt bei einem Wettbewerb zwischen verschiedenen passenden Markern/Regeln der/die spezifischste (Pāṇinis Prinzip).
- ❖ Auch Dekomposition wird in vielen PFM-Analysen vorausgesetzt.
- ❖ Beide Theorien benutzen einen maximal unterspezifizierte Elsewhere-Marker bzw. -Regeln.
- ❖ Darüberhinaus verfügen beide Theorien über zusätzliche Werkzeuge, um bestimmte Paradigmenzellen in Bezug zu setzen (Verarmung bzw. Verweisregeln).

# Unterschiede: Grammatisches Wort

Allerdings gibt es auch gravierende Unterschiede zwischen beiden Theorien:

- ❖ Einer dieser Unterschiede ist der Status des Begriffs *grammatisches Wort*.

# Unterschiede: Grammatisches Wort

Allerdings gibt es auch gravierende Unterschiede zwischen beiden Theorien:

- ❖ Einer dieser Unterschiede ist der Status des Begriffs *grammatisches Wort*.
  - ▶ In DM genießt dieser Begriff keinen theoretischen Status. Die Konzepte, die wirklich für die Wortbildung relevant sind, sind Köpfe, Merkmale und deren strukturelle Beziehungen.

# Unterschiede: Grammatisches Wort

Allerdings gibt es auch gravierende Unterschiede zwischen beiden Theorien:

- ❖ Einer dieser Unterschiede ist der Status des Begriffs *grammatisches Wort*.
  - ▶ In DM genießt dieser Begriff keinen theoretischen Status. Die Konzepte, die wirklich für die Wortbildung relevant sind, sind Köpfe, Merkmale und deren strukturelle Beziehungen.
  - ▶ In PFM gilt das Wort, neben dem Paradigma, als das zentrale Konzept (Wort-und-Paradigma-Ansatz). Prozesse unterhalb der Wortgrenze laufen grundlegend anders ab als alle Prozesse oberhalb der Wortgrenze.

# Unterschiede: Grammatisches Wort

Allerdings gibt es auch gravierende Unterschiede zwischen beiden Theorien:

- ❖ Einer dieser Unterschiede ist der Status des Begriffs *grammatisches Wort*.
  - ▶ In DM genießt dieser Begriff keinen theoretischen Status. Die Konzepte, die wirklich für die Wortbildung relevant sind, sind Köpfe, Merkmale und deren strukturelle Beziehungen.
  - ▶ In PFM gilt das Wort, neben dem Paradigma, als das zentrale Konzept (Wort-und-Paradigma-Ansatz). Prozesse unterhalb der Wortgrenze laufen grundlegend anders ab als alle Prozesse oberhalb der Wortgrenze.
- ❖ Fälle, in denen die Grenze zwischen Wort, Affix, Klitikum, etc. zu verschwimmen scheint, können sich als problematisch für PFM erweisen.
- ❖ Auch Fälle, in denen Teile eines Paradigmas durch zwei Wörter ausgedrückt werden (Periphrase) können u.U. für PFM problematisch sein.

# Unterschiede: Paradigma

- ❖ Ebenfalls ein grundlegender Unterschied ist der Status eines Paradigmas.

# Unterschiede: Paradigma

- ❖ Ebenfalls ein grundlegender Unterschied ist der Status eines Paradigmas.
  - ▶ In DM sind Paradigmen keine Entitäten der Theorie selbst, sondern nur Epiphänomene. Regeln und Operationen in DM reden nie über Paradigmen oder Paradigmenzellen.

# Unterschiede: Paradigma

- ❖ Ebenfalls ein grundlegender Unterschied ist der Status eines Paradigmas.
  - ▶ In DM sind Paradigmen keine Entitäten der Theorie selbst, sondern nur Epiphänomene. Regeln und Operationen in DM reden nie über Paradigmen oder Paradigmenzellen.
  - ▶ In PFM sind Paradigmen 'ein zentrales Prinzip der morphologischen Organisation'. Sie sind definiert als Gesamtheit der wohlgeformten Paradigmenfunktionen einer Sprache.



# Unterschiede: Architektur der Grammatik

- ❖ Natürlich sind auch die grundlegenden Annahmen über die Architektur und Struktur der Grammatik verschieden.

# Unterschiede: Architektur der Grammatik

- ❖ Natürlich sind auch die grundlegenden Annahmen über die Architektur und Struktur der Grammatik verschieden.
  - ▶ In DM ist die Wortbildung Teil der Syntax und der syntaktische Kontext eines Wortes kann die Form des Wortes beeinflussen. (Man denke an die Implementation von Verarmung der Adjektivflexion bei Sauerland 1995.)

# Unterschiede: Architektur der Grammatik

- ❖ Natürlich sind auch die grundlegenden Annahmen über die Architektur und Struktur der Grammatik verschieden.
  - ▶ In DM ist die Wortbildung Teil der Syntax und der syntaktische Kontext eines Wortes kann die Form des Wortes beeinflussen. (Man denke an die Implementation von Verarmung der Adjektivflexion bei Suaerland 1995.)
  - ▶ PFM setzt voraus, dass syntaktische Regeln, Repräsentationen, und Prozesse keinen Zugriff auf morphologische Operationen haben. Daher muss PFM in einigen Fällen das morphosyntaktische Merkmalsinventar vergrößern.

# Unterschiede: Linearisierung

- ❖ Damit eng verbunden ist der zentrale Unterschied der beiden Theorien in Bezug auf Linearisierung.

# Unterschiede: Linearisierung

- ❖ Damit eng verbunden ist der zentrale Unterschied der beiden Theorien in Bezug auf Linearisierung.
  - ▶ In (Standard-)DM spiegelt die Ordnung der Affixe eines komplexen 'Wortes' die zugrundeliegende Syntax wider (Schlagwort: Spiegelprinzip). Etwaige Abweichungen müssen dann mittels zusätzlicher Operationen (Lokale Dislokation, Merger) erzeugt werden.

# Unterschiede: Linearisierung

- ❖ Damit eng verbunden ist der zentrale Unterschied der beiden Theorien in Bezug auf Linearisierung.
  - ▶ In (Standard-)DM spiegelt die Ordnung der Affixe eines komplexen ‘Wortes’ die zugrundeliegende Syntax wider (Schlagwort: Spiegelprinzip). Etwaige Abweichungen müssen dann mittels zusätzlicher Operationen (Lokale Dislokation, Merger) erzeugt werden.
  - ▶ In PFM wird angenommen, dass die Realisierungsregeln bestimmten Blöcken zugeordnet sind und die Affixe dementsprechend genau in der beobachteten Reihenfolge generiert werden.

# Unterschiede: Linearisierung

- ❖ Damit eng verbunden ist der zentrale Unterschied der beiden Theorien in Bezug auf Linearisierung.
  - ▶ In (Standard-)DM spiegelt die Ordnung der Affixe eines komplexen 'Wortes' die zugrundeliegende Syntax wider (Schlagwort: Spiegelprinzip). Etwaige Abweichungen müssen dann mittels zusätzlicher Operationen (Lokale Dislokation, Merger) erzeugt werden.
  - ▶ In PFM wird angenommen, dass die Realisierungsregeln bestimmten Blöcken zugeordnet sind und die Affixe dementsprechend genau in der beobachteten Reihenfolge generiert werden.
- ❖ PFM benötigt daher keine zusätzlichen Operationen, verpasst es aber (womöglich), interessante Generalisierungen zu erfassen (z.B. Julien 2002).
- ❖ Darüberhinaus muss PFM natürlich den Blockindex auf jeder Realisierungsregel stipulieren.

# Unterschiede: Linearisierung

- ❖ Die Annahme, dass Realisierungsregeln in Blöcken angeordnet sind, macht aber auch starke Voraussagen in Bezug auf die Flexibilität innerhalb eines Wortes.



# Unterschiede: Linearisierung

- ❖ Die Annahme, dass Realisierungsregeln in Blöcken angeordnet sind, macht aber auch starke Voraussagen in Bezug auf die Flexibilität innerhalb eines Wortes.
  - ▶ Fälle variabler Affixordnung und Fälle von kontextsensitiver Affixordnung (in Bezug auf syntaktische oder semantische Kontexte) sind auf den ersten Blick problematisch für Template-Ansätze.

# Unterschiede: Linearisierung

- ❖ Die Annahme, dass Realisierungsregeln in Blöcken angeordnet sind, macht aber auch starke Voraussagen in Bezug auf die Flexibilität innerhalb eines Wortes.
  - ▶ Fälle variabler Affixordnung und Fälle von kontextsensitiver Affixordnung (in Bezug auf syntaktische oder semantische Kontexte) sind auf den ersten Blick problematisch für Template-Ansätze.
  - ▶ In diesen Fällen muss die Theorie deutlich kompliziertere Annahmen über die Natur des Blockindexes machen.

# Unterschiede: Linearisierung

- ❖ Die Annahme, dass Realisierungsregeln in Blöcken angeordnet sind, macht aber auch starke Voraussagen in Bezug auf die Flexibilität innerhalb eines Wortes.
  - ▶ Fälle variabler Affixordnung und Fälle von kontextsensitiver Affixordnung (in Bezug auf syntaktische oder semantische Kontexte) sind auf den ersten Blick problematisch für Template-Ansätze.
  - ▶ In diesen Fällen muss die Theorie deutlich kompliziertere Annahmen über die Natur des Blockindex machen.
  - ▶ In DM folgen derartige Effekte (womöglich) mittels der Werkzeuge (Lokale Dislokation, etc.), die ohnehin angenommen werden müssen.

# Unterschiede: Merkmalsnutzung

- ❖ Ein weiterer Unterschied bezieht sich auf die konkrete Implementation von Flexion und die zugänglichen Merkmalsinventare eines Wortes/Kopfes.

# Unterschiede: Merkmalsnutzung

- ❖ Ein weiterer Unterschied bezieht sich auf die konkrete Implementation von Flexion und die zugänglichen Merkmalsinventare eines Wortes/Kopfes.
  - ▶ In PFM sind die für Realisierungsregeln zugänglichen Merkmale immer als ganzes auf dem Wort gespeichert ( $\sigma$ ).

# Unterschiede: Merkmalsnutzung

- ❖ Ein weiterer Unterschied bezieht sich auf die konkrete Implementation von Flexion und die zugänglichen Merkmalsinventare eines Wortes/Kopfes.
  - ▶ In PFM sind die für Realisierungsregeln zugänglichen Merkmale immer als ganzes auf dem Wort gespeichert ( $\sigma$ ).
  - ▶ In DM wird oftmals angenommen, dass Merkmale bei der Vokabulareinsetzung abgearbeitet werden und dann nicht mehr (oder nur bedingt) zugänglich sind (*discharge effect*).

# Unterschiede: Merkmalsnutzung

- ❖ Ein weiterer Unterschied bezieht sich auf die konkrete Implementation von Flexion und die zugänglichen Merkmalsinventare eines Wortes/Kopfes.
  - ▶ In PFM sind die für Realisierungsregeln zugänglichen Merkmale immer als ganzes auf dem Wort gespeichert ( $\sigma$ ).
  - ▶ In DM wird oftmals angenommen, dass Merkmale bei der Vokabulareinsetzung abgearbeitet werden und dann nicht mehr (oder nur bedingt) zugänglich sind (*discharge effect*).
- ❖ DM macht in dieser Hinsicht natürlich stärkere Voraussagen, da für die gleiche Zahl von Affigierungen/Formänderungen weniger Information zur Verfügung steht.

# Unterschiede: Merkmalsnutzung

In der Tat scheint es aber Fälle von Erweiterter Exponenz zu geben, in denen auch bereits realisierte morphosyntaktische Information verfügbar sein muss.



# Unterschiede: Merkmalsnutzung

In der Tat scheint es aber Fälle von Erweiterter Exponenz zu geben, in denen auch bereits realisierte morphosyntaktische Information verfügbar sein muss.

- ❖ In diesen Fällen muss DM sich mit zusätzlichen Werkzeugen behelfen (z.B. sekundäre Einsetzungsmerkmale), die die Theorie verkomplizieren.

# Unterschiede: Merkmalsnutzung

In der Tat scheint es aber Fälle von Erweiterter Exponenz zu geben, in denen auch bereits realisierte morphosyntaktische Information verfügbar sein muss.

- ❖ In diesen Fällen muss DM sich mit zusätzlichen Werkzeugen behelfen (z.B. sekundäre Einsetzungsmerkmale), die die Theorie verkomplizieren.
- ❖ In PFM hingegen ist es erwartet, dass Merkmale mehrfach realisiert werden, allerdings (da immer nur eine Regel pro Block applizieren kann) müssen diese Mehrfachrealisierungen auf verschiedene Blöcke verteilt sein. Zudem sollten sie wohl häufiger auftreten, als sie es in den Sprachen der Welt tun.

# Unterschiede: Verarmungs- vs. Verweisregel

Ein weiterer Unterschied betrifft die Konzepte Verarmungs- und Verweisregel (*rule of referral*).

# Unterschiede: Verarmungs- vs. Verweisregel

Ein weiterer Unterschied betrifft die Konzepte Verarmungs- und Verweisregel (*rule of referral*).

- ❖ Verweiseregeln sind deutlich mächtiger als Verarmungsregeln, weil sie unbeschränkt sind.
- ❖ Eine Verarmungsregel kann immer nur Merkmale löschen und so sozusagen nur Unterschiede neutralisieren.
- ❖ Eine Verweisregel kann zwei beliebige Paradigmenzellen miteinander in Bezug setzen.

# Unterschiede: Verarmungs- vs. Verweisregel

Ein weiterer Unterschied betrifft die Konzepte Verarmungs- und Verweisregel (*rule of referral*).

- ❖ Verweiseregeln sind deutlich mächtiger als Verarmungsregeln, weil sie unbeschränkt sind.
- ❖ Eine Verarmungsregel kann immer nur Merkmale löschen und so sozusagen nur Unterschiede neutralisieren.
- ❖ Eine Verweisregel kann zwei beliebige Paradigmenzellen miteinander in Bezug setzen.
- ▶ Eine Verweisregel ist also in gewisser Weise äquivalent zu einer Verarmungsregel, die Merkmale auch hinzufügen oder verändern kann.

# Zusammenfassung

Die Paradigmenfunktionsmorphologie ist eine alternative morphologische Theorie, die sich in einigen grundlegenden Annahmen von DM unterscheidet.

# Zusammenfassung

Die Paradigmenfunktionsmorphologie ist eine alternative morphologische Theorie, die sich in einigen grundlegenden Annahmen von DM unterscheidet.

- ❖ Allgemein lässt sich wohl sagen, dass PFM in vielen Fällen übereinzelsprachlich weniger starke Voraussagen macht.

# Zusammenfassung

Die Paradigmenfunktionsmorphologie ist eine alternative morphologische Theorie, die sich in einigen grundlegenden Annahmen von DM unterscheidet.

- ❖ Allgemein lässt sich wohl sagen, dass PFM in vielen Fällen übereinzelsprachlich weniger starke Voraussagen macht.
  - ▶ Vielfach werden Regeln weniger restriktiv formuliert oder Merkmalsinventare derart angereichert, dass Sprachen einen deutlich größeren Variationsspielraum haben.



# Zusammenfassung

Die Paradigmenfunktionsmorphologie ist eine alternative morphologische Theorie, die sich in einigen grundlegenden Annahmen von DM unterscheidet.

- ❖ Allgemein lässt sich wohl sagen, dass PFM in vielen Fällen übereinzelsprachlich weniger starke Voraussagen macht.
  - ▶ Vielfach werden Regeln weniger restriktiv formuliert oder Merkmalsinventare derart angereichert, dass Sprachen einen deutlich größeren Variationsspielraum haben.
  - ▶ Diese Beobachtung speist sich wohl aus der Herangehensweise, dass PFM eher einen deskriptiven Standpunkt einnimmt als einen explanativen. Phänomene sollen lediglich möglichst exakt und formal beschrieben werden.

# Zusammenfassung

Die Paradigmenfunktionsmorphologie ist eine alternative morphologische Theorie, die sich in einigen grundlegenden Annahmen von DM unterscheidet.

- ❖ Allgemein lässt sich wohl sagen, dass PFM in vielen Fällen übereinzelsprachlich weniger starke Voraussagen macht.
  - ▶ Vielfach werden Regeln weniger restriktiv formuliert oder Merkmalsinventare derart angereichert, dass Sprachen einen deutlich größeren Variationsspielraum haben.
  - ▶ Diese Beobachtung speist sich wohl aus der Herangehensweise, dass PFM eher einen deskriptiven Standpunkt einnimmt als einen explanativen. Phänomene sollen lediglich möglichst exakt und formal beschrieben werden.
  - ▶ Es sollen möglichst wenig Voraussagen über hypothetische Sprachtypen angestellt werden (die sich später als falsch herausstellen und teils starke Anpassungen der Theorie erfordern).

# Fallstudie: Kugu Nghanhcara

# Fallstudie: Pronomen in Kugu Nganhcara

Bonitz (2008) präsentiert eine PFM-Analyse der Pronomen im Kugu Nganhcara (Pama-Nyungan, Australien).

(26) *Deklination der freien Pronomen in NOM, ACC, DAT, und ABL*

	NOM	ACC	DAT	ABL
1.SG	ngaye	nganyi	ngathu	ngathurumu
1.DU.INK	ngale	ngalina	ngalina	ngalinamu
1.DU.EXK	ngana	nganana	nganana	ngananamu
1.PL.INK	ngampa	ngampara	ngampara	ngamparamu
1.PL.EXK	nganhca	nganhcara	nganhcara	nagnhcaramu
2.SG	nhinta	nina	nhingku	nhingkurumu
2.DU	nhipa	nhipana	nhipana	nhipanamu
2.PL	nhiya	nhiyana	nhiyana	nhiyanamu
3.SG	nhila	nhunha	nhingu	nhingurumu
3.DU	pula	pulana	pulana	pulanamu
3.PL	thana	thaarana	thaarana	thaaaranamu

# Fallstudie: Pronomen in Kugu Nganhcara

Außer im Singular ist der Akkusativ synkretisch mit dem Dativ.

(26) *Deklination der freien Pronomen in NOM, ACC, DAT, und ABL*

	NOM	ACC	DAT	ABL
1.SG	ngaye	nganyi	ngathu	ngathurumu
1.DU.INK	ngale		ngalina	ngalinamu
1.DU.EXK	ngana		nganana	ngananamu
1.PL.INK	ngampa		ngampara	ngamparamu
1.PL.EXK	nganhca		nganhcara	nganhcaramu
2.SG	nhinta	nina	nhingku	nhingkurumu
2.DU	nhipa		nhipana	nhipanamu
2.PL	nhiya		nhiyana	nhiyanamu
3.SG	nhila	nhunha	nhingu	nhingurumu
3.DU	pula		pulana	pulanamu
3.PL	thana		thaarana	thaanamu

# Fallstudie: Pronomen in Kugu Nganhcara

1. und 2. Person haben jeweils einen einzigen Stamm. In der 3. Person gibt es drei Stämme.

(26) *Deklination der freien Pronomen in NOM, ACC, DAT, und ABL*

	NOM	ACC	DAT	ABL
1.SG	nga-ye	nga-nyi	nga-thu	nga-thurumu
1.DU.INK	nga-le		nga-lina	nga-linamu
1.DU.EXK	nga-na		nga-nana	nga-nanamu
1.PL.INK	nga-mpa		nga-mpara	nga-mparamu
1.PL.EXK	nga-nhca		nga-nhcara	nga-nhcaramu
2.SG	nhi-nta	ni-na	nhi-ngku	nhi-ngkurumu
2.DU	nhi-pa		nhi-pana	nhi-panamu
2.PL	nhi-ya		nhi-yana	nhi-yanamu
3.SG	nhi-la	nhu-nha	nhi-ngu	nhi-ngurumu
3.DU	pu-la		pu-lana	pu-lanamu
3.PL	tha-na		thaa-rana	thaa-ranamu

# Fallstudie: Pronomen in Kugu Nghanhcara

Im Regelfall scheint der Ablativ auf Basis des Dativs (und Akkusativs) aufgebaut zu sein, der wiederum auf dem Nominativ aufgebaut ist: Das nennt man auch *Priscianische* oder *parasitäre Bildung*.

## (26) *Deklination der freien Pronomen in NOM, ACC, DAT, und ABL*

	NOM	ACC	DAT	ABL
1.SG	nga-ye	nga-nyi	nga-thu	nga-thu-ru-mu
1.DU.INK	nga-le		nga-li-na	nga-li-na-mu
1.DU.EXK	nga-na		nga-na-na	nga-na-na-mu
1.PL.INK	nga-mpa		nga-mpa-ra	nga-mpa-ra-mu
1.PL.EXK	nga-nhca		nga-nhca-ra	nga-nhca-ra-mu
2.SG	nhi-nta	ni-na	nhi-ngku	nhi-ngku-ru-mu
2.DU	nhi-pa		nhi-pa-na	nhi-pa-na-mu
2.PL	nhi-ya		nhi-ya-na	nhi-ya-na-mu
3.SG	nhi-la	nhu-nha	nhi-ngu	nhi-ngu-ru-mu
3.DU	pu-la		pu-la-na	pu-la-na-mu
3.PL	tha-na		thaa-ra-na	thaa-ra-na-mu

# Fallstudie: Pronomen in Kugu Nghanhcara

Im Regelfall scheint der Ablativ auf Basis des Dativs (und Akkusativs) aufgebaut zu sein, der wiederum auf dem Nominativ aufgebaut ist: Das nennt man auch *Priscianische* oder *parasitäre Bildung*.

## (26) *Deklination der freien Pronomen in NOM, ACC, DAT, und ABL*

	NOM	ACC	DAT	ABL
1.SG	nga-ye	nga-nyi	nga-thu	nga-thu-ru-mu
1.DU.INK	nga-le		nga-li-na	nga-li-na-mu
1.DU.EXK	nga-na		nga-na-na	nga-na-na-mu
1.PL.INK	nga-mpa		nga-mpa-ra	nga-mpa-ra-mu
1.PL.EXK	nga-nhca		nga-nhca-ra	nga-nhca-ra-mu
2.SG	nhi-nta	ni-na	nhi-ngku	nhi-ngku-ru-mu
2.DU	nhi-pa		nhi-pa-na	nhi-pa-na-mu
2.PL	nhi-ya		nhi-ya-na	nhi-ya-na-mu
3.SG	nhi-la	nhu-nha	nhi-ngu	nhi-ngu-ru-mu
3.DU	pu-la		pu-la-na	pu-la-na-mu
3.PL	tha-na		thaa-ra-na	thaa-ra-na-mu



# Fallstudie: Pronomen in Kugu Nghanhara

Wir brauchen mind. 4 Blöcke: 4. Block: *mu* [ABL]; 3. Block: *na/ra* [ACC/DAT], *ru* immer dann, wenn ABL nicht auf NOM basiert; 2. Block: diverse Person/Numerus/(Kasus)-Morpheme; 1. Block: Stämme.

## (26) *Deklination der freien Pronomen in NOM, ACC, DAT, und ABL*

	NOM	ACC	DAT	ABL
1.SG	nga-ye	nga-nyi	nga-thu	nga-thu- <b>ru-mu</b>
1.DU.INK	nga-le	nga-li- <b>na</b>		nga-li- <b>na-mu</b>
1.DU.EXK	nga-na	nga-na- <b>na</b>		nga-na- <b>na-mu</b>
1.PL.INK	nga-mpa	nga-mpa- <b>ra</b>		nga-mpa- <b>ra-mu</b>
1.PL.EXK	nga-nhca	nga-nhca- <b>ra</b>		nga-nhca- <b>ra-mu</b>
2.SG	nhi-nta	ni-na	nhi-ngku	nhi-ngku- <b>ru-mu</b>
2.DU	nhi-pa	nhi-pa- <b>na</b>		nhi-pa- <b>na-mu</b>
2.PL	nhi-ya	nhi-ya- <b>na</b>		nhi-ya- <b>na-mu</b>
3.SG	nhi-la	nhu-nha	nhi-ngu	nhi-ngu- <b>ru-mu</b>
3.DU	pu-la	pu-la- <b>na</b>		pu-la- <b>na-mu</b>
3.PL	tha-na	thaa-ra- <b>na</b>		thaa-ra- <b>na-mu</b>

# Fallstudie: Pronomen im Kugu Nganhcara

## Dekomposition:

(27) *Person:*

1.INK [+1,+2]

1.EXK [+1,-2]

2. [-1,+2]

3. [-1,-2]

(28) *Numerus:*

SG [+sg,-pl]

DU [-sg,-pl]

PL [-sg,+pl]

# Fallstudie: Pronomen im Kugu Nganhcara

Dekomposition:

(27) *Person:*

1.INK	[+1,+2]
1.EXK	[+1,-2]
2.	[-1,+2]
3.	[-1,-2]

(28) *Numerus:*

SG	[+sg,-pl]
DU	[-sg,-pl]
PL	[-sg,+pl]

Die Dekomposition von Kasus gestaltet sich hier schwierig, da ACC und DAT eine natürliche Klasse bilden müssen, aber auch DAT und ABL. Die Merkmale, die diese Klassen definieren müssten durch den /na/- bzw. /ra/-Marker realisiert werden, da dieser in allen drei Kasus auftaucht. Zwei binäre Merkmale reichen also nicht aus.

(29) *Unnütze Dekomposition von Kasus:*

NOM	[-obj,-obl]
ACC	[+obj,-obl]
DAT	[+obj,+obl]
ABL	[-obj,+obl]

# Fallstudie: Pronomen im Kugu Nganhcara

Dekomposition:

(27) *Person:*

1.INK	[+1,+2]
1.EXK	[+1,-2]
2.	[-1,+2]
3.	[-1,-2]

(28) *Numerus:*

SG	[+sg,-pl]
DU	[-sg,-pl]
PL	[-sg,+pl]

Eine Dekomposition mit drei binären Merkmalen ist notwendig. Diese stellt noch vier zusätzliche Kasus zur Verfügung. In der Tat gibt es in der Sprache noch vier andere Kasus: Reflexiv, Komitativ, Privativ, und Lokativ. Die Merkmal hier haben keinen semantischen Inhalt.

(29) *Nützliche Kasusdekomposition:*

NOM	[-A,-B,-C]	REFL	[-A,+B,+C]
ACC	[+A,-B,-C]	KOM	[-A,-B,+C]
DAT	[+A,+B,-C]	PRIV	[+A,-B,+C]
ABL	[+A,+B,+C]	LOK	[-A,+B,-C]

# Fallstudie: Pronomen im Kugu Nganhcara

Die Regeln des 1. Blocks: *ni*, *pu*, *tha*, *nhu*, *nhi*, *nga*

- (30) a.  $RR_{1,\{-1,+2,+sg,+A,-B\},P}(\langle X,\sigma \rangle) = \langle Y,\sigma \rangle$ , wobei  $Y=ni$   
 b.  $RR_{1,\{-1,+sg,+A,-B\},P}(\langle X,\sigma \rangle) = \langle Y,\sigma \rangle$ , wobei  $Y=nhu$   
 c.  $RR_{1,\{-1,-2,-sg,-pl\},P}(\langle X,\sigma \rangle) = \langle Y,\sigma \rangle$ , wobei  $Y=pu$   
 d.  $RR_{1,\{-1,-2,+pl\},P}(\langle X,\sigma \rangle) = \langle Y,\sigma \rangle$ , wobei  $Y=tha$   
 e.  $RR_{1,\{+1\},P}(\langle X,\sigma \rangle) = \langle Y,\sigma \rangle$ , wobei  $Y=nga$   
 f.  $RR_{1,\{\},P}(\langle X,\sigma \rangle) = \langle Y,\sigma \rangle$ , wobei  $Y=nhi$

1.SG	nga-ye	nga-nyi	nga-thu	nga-thu-ru-mu
1.DU.INK	nga-le	nga-li-na		nga-li-na-mu
1.DU.EXK	nga-na	nga-na-na		nga-na-na-mu
1.PL.INK	nga-mpa	nga-mpa-ra		nga-mpa-ra-mu
1.PL.EXK	nga-nhca	nga-nhca-ra		nga-nhca-ra-mu
2.SG	nhi-nta	ni-na	nhi-ngku	nhi-ngku-ru-mu
2.DU	nhi-pa	nhi-pa-na		nhi-pa-na-mu
2.PL	nhi-ya	nhi-ya-na		nhi-ya-na-mu
3.SG	nhi-la	nhu-nha	nhi-ngu	nhi-ngu-ru-mu
3.DU	pu-la	pu-la-na		pu-la-na-mu
3.PL	tha-na	thaa-ra-na		thaa-ra-na-mu

# Fallstudie: Pronomen im Kugu Nganhcara

Die Regeln des 2. Blocks: *nyi, nha, le, li, ye, nta, ngu, pa, na, thu, ngku, ya, ra, la, mpa, nhca*

- (31) a.  $RR_{2,\{+1,-2,+sg,+A,-B\},P}(\langle X,\sigma \rangle) = \langle Xnyi,\sigma \rangle$   
 b.  $RR_{2,\{-1,-2,+sg,+A,-B\},P}(\langle X,\sigma \rangle) = \langle Xnha,\sigma \rangle$   
 c.  $RR_{2,\{+1,+2,-sg,-pl,-A\},P}(\langle X,\sigma \rangle) = \langle Xle,\sigma \rangle$   
 d.  $RR_{2,\{+1,+2,-sg,-pl\},P}(\langle X,\sigma \rangle) = \langle Xli,\sigma \rangle$   
 e.  $RR_{2,\{+1,-2,+sg,-A\},P}(\langle X,\sigma \rangle) = \langle Xye,\sigma \rangle$   
 f.  $RR_{2,\{-1,+2,+sg,-A\},P}(\langle X,\sigma \rangle) = \langle Xnta,\sigma \rangle$   
 g.  $RR_{2,\{-1,-2,+sg,+A\},P}(\langle X,\sigma \rangle) = \langle Xngu,\sigma \rangle$   
 h.  $RR_{2,\{-1,+2,-sg,-pl\},P}(\langle X,\sigma \rangle) = \langle Xpa,\sigma \rangle$   
 i.  $RR_{2,\{+1,-2,-sg,-pl\},P}(\langle X,\sigma \rangle) = \langle Xna,\sigma \rangle$   
 j.  $RR_{2,\{+1,-2,+sg\},P}(\langle X,\sigma \rangle) = \langle Xthu,\sigma \rangle$   
 k.  $RR_{2,\{-1,+2,+sg\},P}(\langle X,\sigma \rangle) = \langle Xngku,\sigma \rangle$   
 l.  $RR_{2,\{-1,+2,+pl\},P}(\langle X,\sigma \rangle) = \langle Xya,\sigma \rangle$   
 m.  $RR_{2,\{-1,-2,+pl\},P}(\langle X,\sigma \rangle) = \langle Xra,\sigma \rangle$   
 n.  $RR_{2,\{-1,-2,-pl\},P}(\langle X,\sigma \rangle) = \langle Xla,\sigma \rangle$   
 o.  $RR_{2,\{+1,+2,+pl\},P}(\langle X,\sigma \rangle) = \langle Xmpa,\sigma \rangle$   
 p.  $RR_{2,\{+1,-2,+pl\},P}(\langle X,\sigma \rangle) = \langle Xnhca,\sigma \rangle$

# Fallstudie: Pronomen im Kugu Nganhcara

Noch generieren wir die falsche Form *\*tha-ra* in der 3.PL.NOM und *\*ni-ngku* in der 2.SG.DAT. In beiden Fällen sollte eigentlich *-na* an den jeweiligen Stamm angefügt werden.

# Fallstudie: Pronomen im Kugu Nghanhcara

Noch generieren wir die falsche Form *\*tha-ra* in der 3.PL.NOM und *\*ni-ngku* in der 2.SG.DAT. In beiden Fällen sollte eigentlich *-na* an den jeweiligen Stamm angefügt werden.

Die drei Kontexte für *-na* lassen sich nur schwerlich durch Unterspezifikation zusammenführen.

- (32)
- a. 1.DU.EXK: [+1,-2,-sg,-pl]
  - b. 2.SG.DAT: [-1,+2,+sg,-pl,+A,+B,-C]
  - c. 3.PL.NOM: [-1,-2,-sg,+pl,-A]



# Fallstudie: Pronomen im Kugu Nganhcara

Noch generieren wir die falsche Form *\*tha-ra* in der 3.PL.NOM und *\*ni-ngku* in der 2.SG.DAT. In beiden Fällen sollte eigentlich *-na* an den jeweiligen Stamm angefügt werden.

Die drei Kontexte für *-na* lassen sich nur schwerlich durch Unterspezifikation zusammenführen.

- (32) a. 1.DU.EXK: [+1,-2,-sg,-pl]  
 b. 2.SG.DAT: [-1,+2,+sg,-pl,+A,+B,-C]  
 c. 3.PL.NOM: [-1,-2,-sg,+pl,-A]

Daher bietet es sich an, hier zwei Verweisregeln zu formulieren, die sicherstellen, dass sich in Block 2 die Form der 2.SG.DAT und 3.PL.NOM nach der 1.DU.EXK richten.

$$(33) \text{RR}_{2,\{-1,+2,+sg,+A,+B,-C\},P}(\langle X,\sigma \rangle) = \langle Y,\sigma \rangle, \text{ wobei } \text{Nar}_2(\langle X,\sigma / \{+1,-2,-sg\} \rangle) = \langle Y,\sigma / \{+1,-2,-sg\} \rangle$$

$$(34) \text{RR}_{2,\{-1,-2,+pl,-A\},P}(\langle X,\sigma \rangle) = \langle Y,\sigma \rangle, \text{ wobei } \text{Nar}_2(\langle X,\sigma / \{+1,-pl\} \rangle) = \langle Y,\sigma / \{+1,-pl\} \rangle$$

# Fallstudie: Pronomen im Kugu Nganhcara

Die Regeln des 3. Blocks: *na*, *ra*, *ru*

- (35) a.  $RR_{3,\{+1,+pl,+A\},P}(\langle X,\sigma \rangle) = \langle Xra,\sigma \rangle$   
 b.  $RR_{3,\{-sg,+A\},P}(\langle X,\sigma \rangle) = \langle Xna,\sigma \rangle$   
 c.  $RR_{3,\{+sg,+C\},P}(\langle X,\sigma \rangle) = \langle Xru,\sigma \rangle$   
 d.  $RR_{3,\{+sg\},P}(\langle X,\sigma \rangle) = \langle X,\sigma \rangle$

Die Regel (35-d) stellt sicher, dass der dritte Slot eines Wortes nicht unbedingt besetzt werden muss.

# Fallstudie: Pronomen im Kugu Nghanhara

Die Regeln des 4. Blocks: *mu*

- (36) a.  $RR_{4,\{+C\},P}(\langle X,\sigma \rangle) = \langle Xmu,\sigma \rangle$   
 b.  $RR_{4,\{\},P}(\langle X,\sigma \rangle) = \langle X,\sigma \rangle$

Wieder stellt die letzte Regel sicher, dass die Derivation nicht scheitert, wenn eine Zelle nicht im Ablativ steht.

# Fallstudie: Pronomen im Kugu Nganhcara

## Zusammenfassung:

- ❖ Das Paradigma konnte komplett mit PFM analysiert werden.
- ❖ Der Vorteil von PFM ist, dass Merkmale mehrfach (in verschiedenen Blöcken) für die Realisierung von Wortformen genutzt werden können. Dadurch ist eine Priscianische Wortbildung, wie im vorliegenden Fall, leichter abzuleiten als in DM.