

Merkmalestrukturen



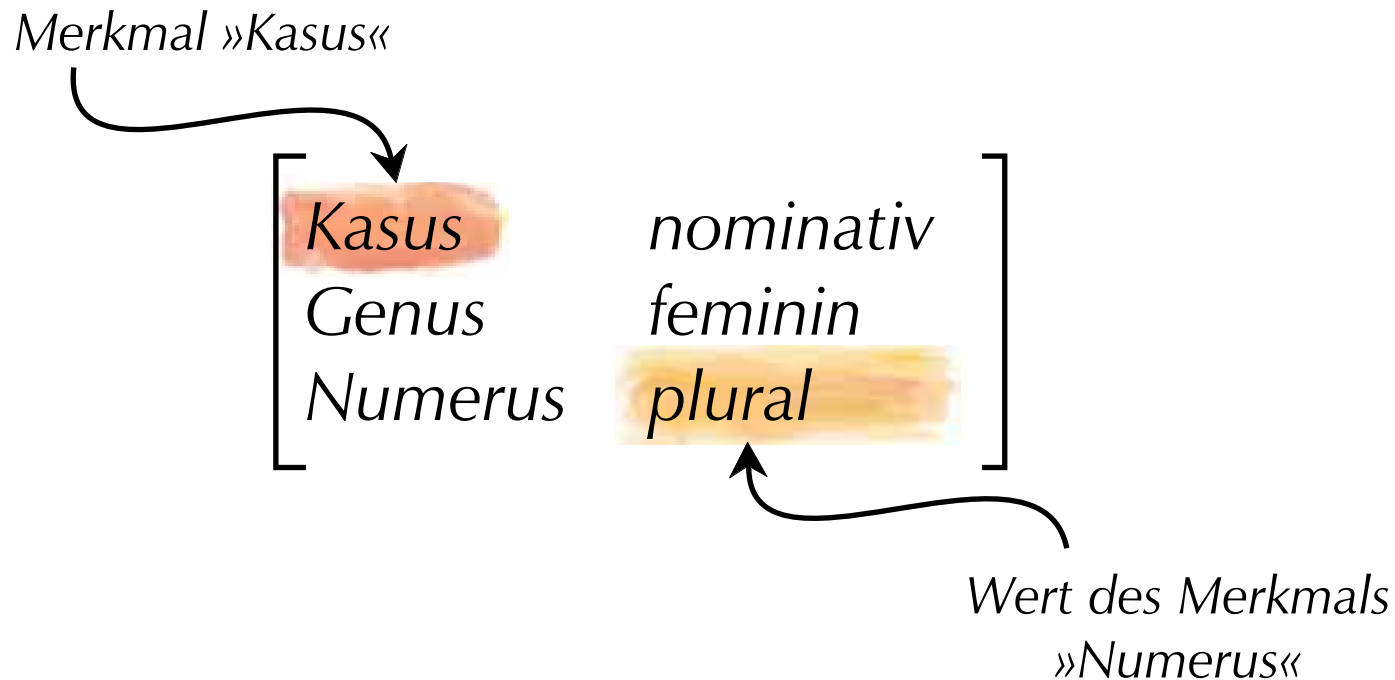
Übersicht

- ◆ Beispiele für Merkmalstrukturen
- ◆ Koreferenz, Unifikation
- ◆ Sinn und Zweck
- ◆ Merkmalstrukturen in Prolog: GULP
 - ◆ weitere Ansätze zur Übersetzung von Merkmalstrukturen
- ◆ Literatur

Ziel

- ◆ Verstehen der Konzepte
- ◆ Nachvollziehen, wie GULP funktioniert

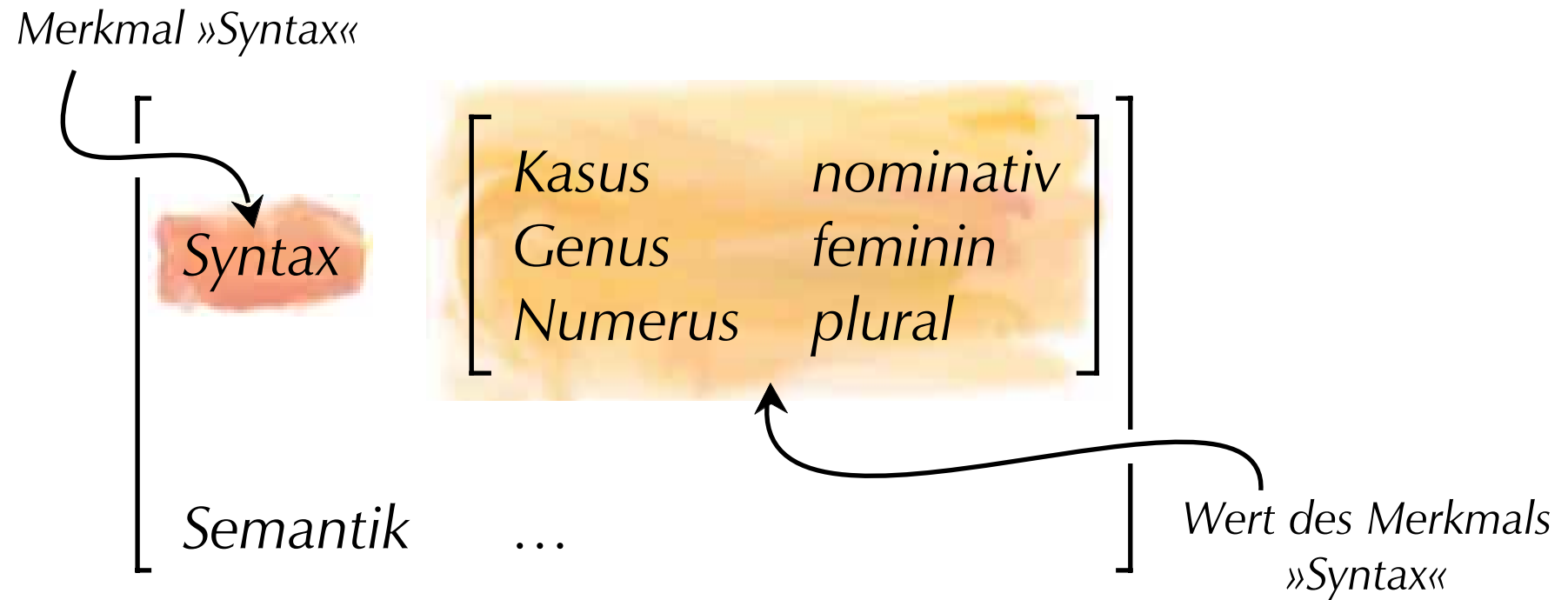
Einfache Merkmalstrukturen



Bestandteile einer Merkmalstruktur (*Feature Structure*)

- ◆ Merkmale/Attribute (*Features/Attributes*)
- ◆ Werte (*Values*)

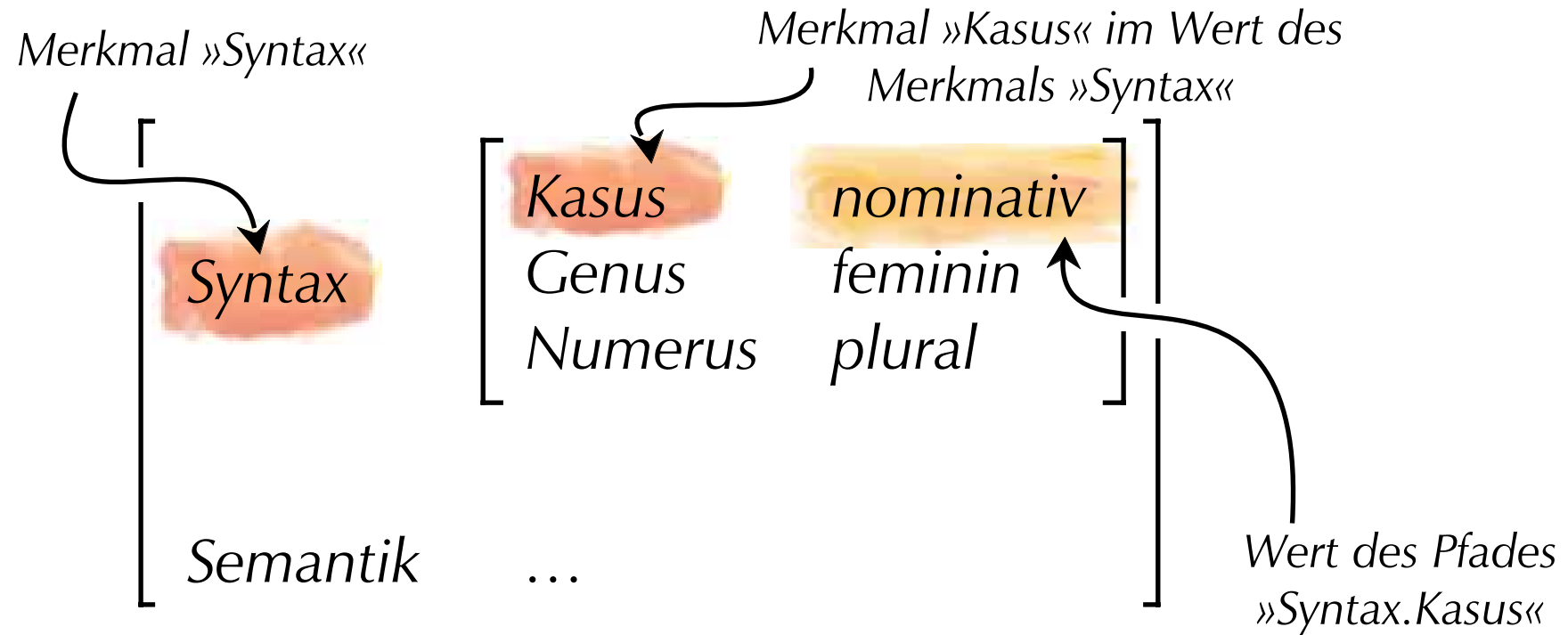
Verschachtelte Merkmalstrukturen



Eine etwas kompliziertere Merkmalstruktur

- ◆ der Wert des Merkmals »Syntax« ist selbst wiederum eine Merkmalstruktur ⇒ Verschachtelte Strukturen

Pfade



Ketten von Merkmalen heissen Pfade (*paths*)

- ◆ Wert des Pfades »Syntax.Kasus«
= Wert des Merkmals »Kasus« im Wert des Merkmals »Syntax«

Partielle Funktion: Pfad \rightarrow Wert

Für jeden Pfad enthält eine Merkmalstruktur höchstens einen Wert.

- ◆ damit ist eine Merkmalstruktur eine partielle Funktion, die Pfade zu Werten abbildet
- ◆ mindestens in dieser Einführung ...
 - ◆ ... es gibt durchaus auch Formalismen, wo genau ein Wert gefordert ist (totale Funktion) oder auch mehr als ein Wert zulässig wäre
 - ◆ sehr verbreitet
 - ◆ ... es gibt auch Formalismen, wo die Abbildung »unscharf« ist: »Mit einer Wahrscheinlichkeit von 80% ist der Kasus Nominativ«
 - ◆ vereinzelt vorgeschlagen, aber kaum verbreitet

Partielle Funktion: Pfad → Wert

Syntax	<table><tr><td>Kasus</td><td><i>nominativ</i></td></tr><tr><td>Genus</td><td><i>feminin</i></td></tr><tr><td>Numerus</td><td><i>plural</i></td></tr></table>	Kasus	<i>nominativ</i>	Genus	<i>feminin</i>	Numerus	<i>plural</i>
Kasus	<i>nominativ</i>						
Genus	<i>feminin</i>						
Numerus	<i>plural</i>						
Semantik	[]						

◆ Wert von *Syntax*:

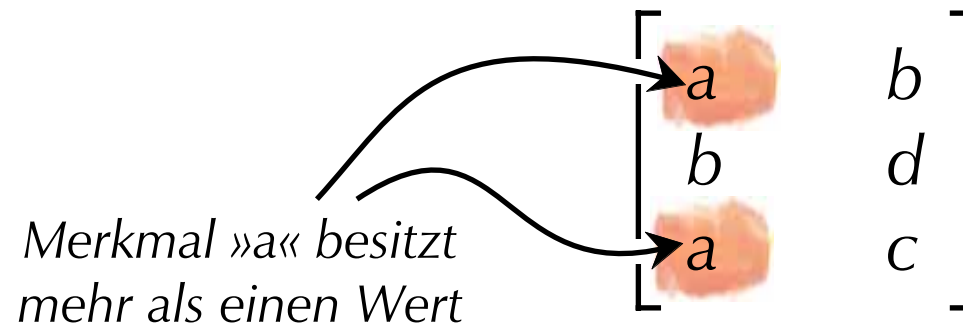
- ◆ Wert von *Syntax.Kasus*: *nominativ*
- ◆ Wert von *Syntax.Genus*: *feminin*
- ◆ Wert von *Syntax.Numerus*: *plural*
- ◆ Wert von *Syntax.Tempus*: **nicht definiert**

◆ Wert von *Semantik*:

- ◆ *eine leere Merkmalstruktur*

Partielle Funktion: Pfad \rightarrow Wert

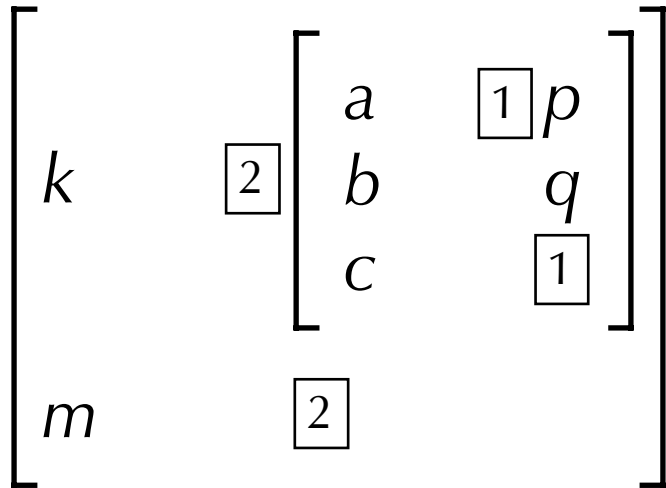
Beispiel für eine unzulässige Merkmalstruktur:



Koreferente Pfade

Koreferente Pfade besitzen denselben Wert

- ◆ beschreiben ein Gleichungssystem
- ◆ ähnlich wie die Variablen von Prolog
 - ◆ daher hier nicht ausführlicher erklärt
- ◆ Notation als Nummer innerhalb eines Kästchens



$Wert(k.a) = p$
 $Wert(k.b) = q$
 $Wert(k.c) = Wert(k.a)$
 $Wert(m) = Wert(k)$

 $Wert(k.c) = p$
 $Wert(m.a) = p$
...

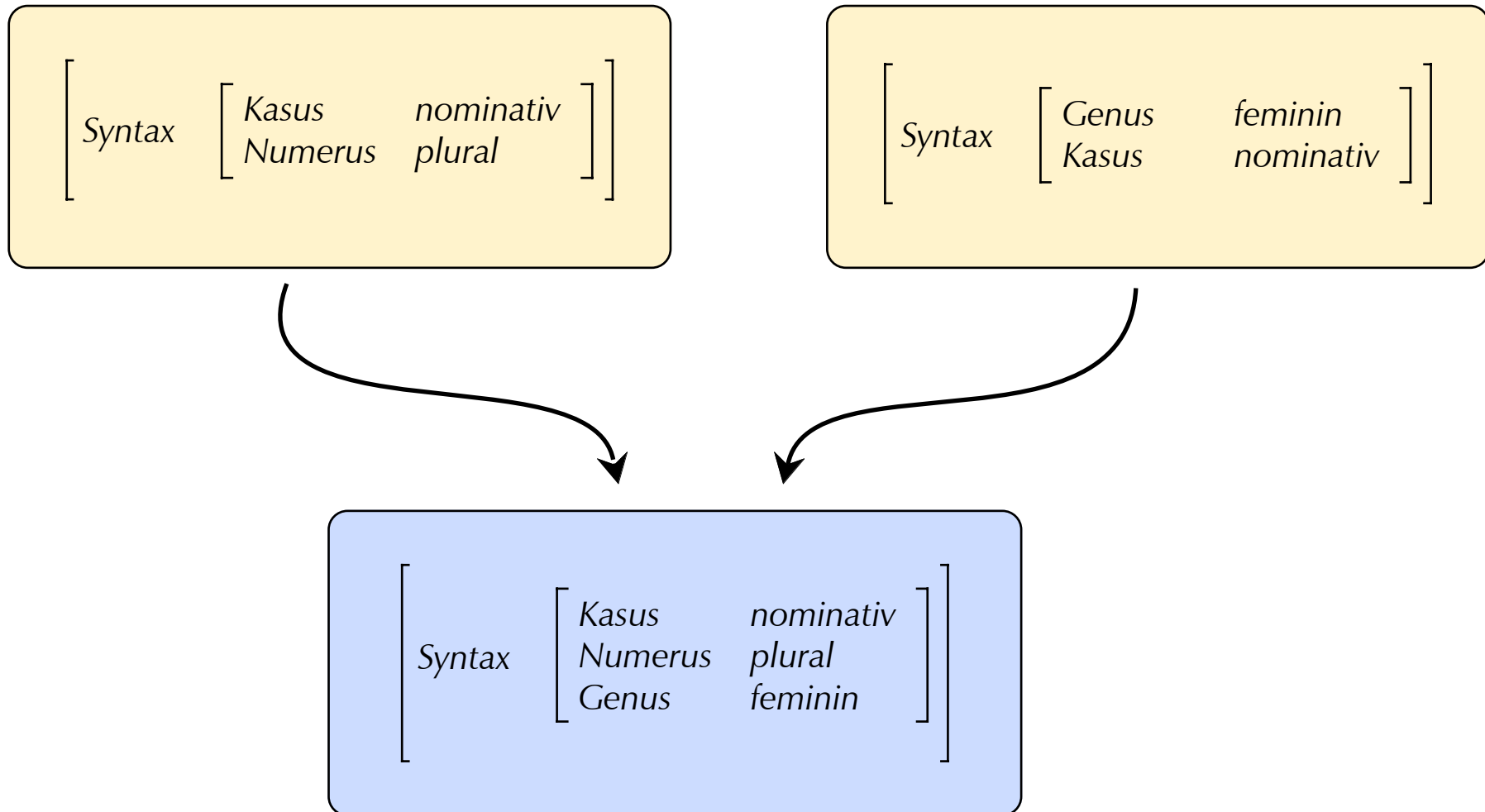
Unifikation



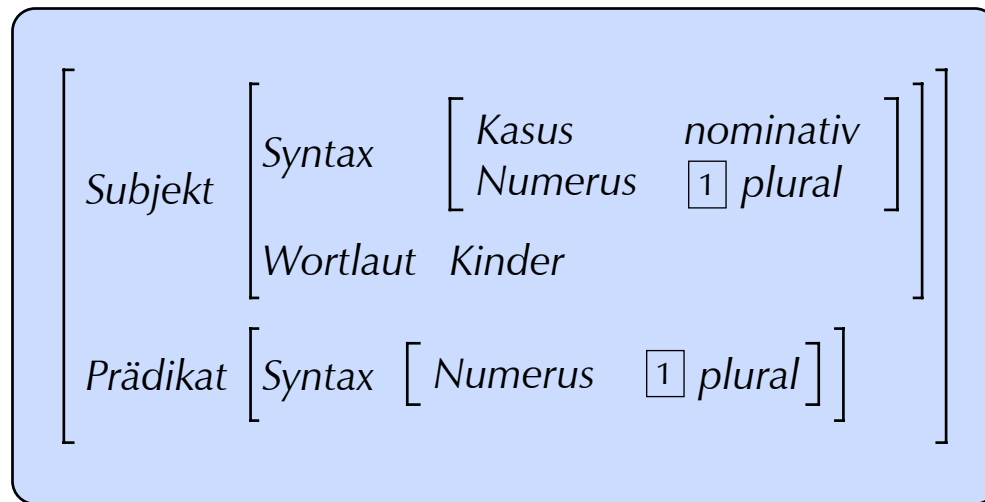
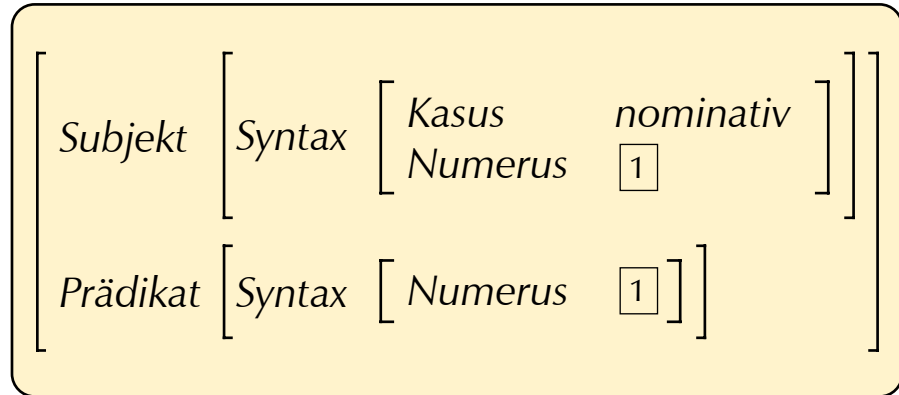
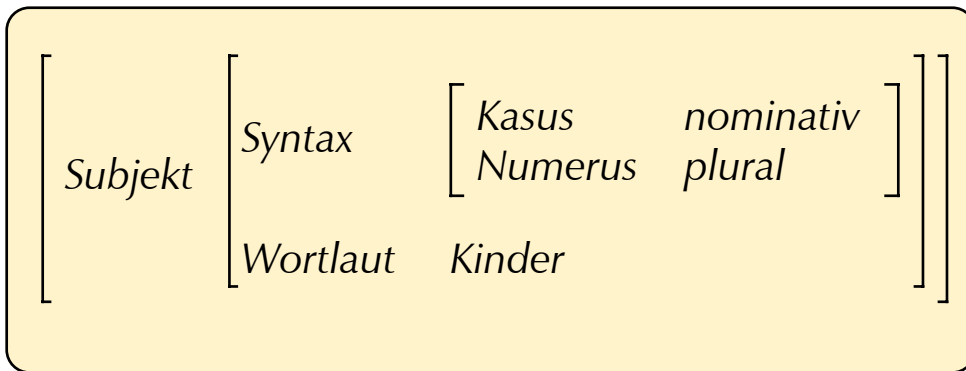
Merkmalstrukturen können unifiziert werden

- ◆ Um zwei Merkmalstrukturen zu unifizieren, unifiziere die Werte aller Merkmale
- ◆ Wenn ein Merkmal nur in einer der beiden Strukturen vorkommt:
 - ◆ kopiere das Merkmal (zusammen mit seinem Wert) in die resultierende Struktur
- ◆ Wenn ein Merkmal in beiden Strukturen vorkommt, unifiziere die beiden Werte:
 - ◆ beides Atome: Wenn die beiden Atome gleich sind, ist das Ergebnis eben dieses; sind sie verschieden, schlägt die Unifikation fehl
 - ◆ beides Merkmalstrukturen: Unifikation der Strukturen (Rekursion!)
 - ◆ ein Atom, eine Merkmalstruktur: Unifikation schlägt fehl

Unifikation



Unifikation



Unifikation

$\left[\text{Syntax} \left[\begin{array}{l} \text{Kasus} \\ \text{Numerus} \end{array} \begin{array}{l} \text{nominativ} \\ \text{plural} \end{array} \right] \right]$

$\left[\text{Syntax} \left[\begin{array}{l} \text{Genus} \\ \text{Kasus} \end{array} \begin{array}{l} \text{feminin} \\ \text{dativ} \end{array} \right] \right]$

Diese Merkmalstrukturen sind nicht unifizierbar

- ◆ weil im rekursiven Schritt, beim Unifizieren von des Werts von »Syntax«, die Werte von »Kasus« nicht unifizierbar sind
- ◆ bei komplizierten Strukturen kann die Unifikation »tief innen« fehlschlagen

Sinn und Zweck

In the early days of generative grammar, the linguistic types singled out for attention were the sentences, considered as strings of phonetic shapes. Correspondingly, a grammar was just a computational device, for example, a context-free grammar or a transformational grammar, that enumerated a set of strings. Most linguistic theories, of course, are much more demanding: the linguistic types par excellence, the expressions—or signs (in roughly the Saussurean sense)—include not only sentences, but also the words and subsentential phrases, even multisentence discourses. And a sign is taken to consist not only of a phonetic form, but of other attributes or features as well. That is, we conceive of signs as structured complexes of phonological, syntactic, semantic, and phrase-structural information.

*Head-Driven Phrase-Structure Grammar, 1994
Beginn von Kapitel 1, »HPSG: A System of Signs«*

Sinn und Zweck



Wozu Merkmalstrukturen?

- ◆ einheitlicher (und mathematisch sauber definierter) Formalismus für die verschiedenen Ebenen der Linguistik
 - ◆ Phonologie, Morphologie, Syntax, Semantik, Pragmatik, ...
- ◆ Repräsentation *aller* Information, die zu einem linguistischen Zeichen bekannt ist
 - ◆ Formalisierung des modellierten Objekts
 - ◆ Beispiel: Ergebnis der syntaktischen Analyse (Struktur-Baum) als Merkmalstruktur
- ◆ aber auch: einheitliche Repräsentation des Wissens über die Regularitäten der modellierten Sprache
 - ◆ Formalisierung der Modellierungswerkzeuge
 - ◆ Beispiel: Grammatikregel als Merkmalstruktur

Sinn und Zweck



The role of the linguistic theory is to give a precise specification of which feature structures are to be considered admissible; the types of linguistic entities that correspond to the admissible feature structures constitute the prediction of the theory.

Linguistic theory has become sufficiently modular and complex that a need for formalization has become apparent, especially to researchers concerned with the computational implementation of current theories.

Feature logics [have been developed that are] appropriate for the formalization of linguistic theories, languages whose formulas serve as the linguist's analog of the physicist's differential equations.

*Head-Driven Phrase-Structure Grammar, 1994
Gekürzter Auszug aus der Einführung*

Anwendungen



Theoretische Linguistik

- ◆ Unifikationsgrammatiken zur Beschreibung der Syntax, Morphologie und anderer Ebenen
 - ◆ Lexikalisch-Funktionale Grammatik (LFG)
 - ◆ Kopf-getriebene Phrasenstruktur-Grammatik (HPSG)
 - ◆ Unifikations-basierte Varianten der Kategorialgrammatik
 - ◆ einige andere

Computerlinguistik

- ◆ Entwicklungsumgebungen für Unifikationsgrammatiken
 - ◆ ALE, AMALIA, GULP, PATR, ProFIT, TDL (und viele andere)

Merkmalstrukturen in Prolog

Die Konzepte bei Merkmalstrukturen ähneln denen von Prolog

- ◆ Atomarer Wert ~ Prolog-Atom
- ◆ Merkmalstruktur ~ komplexer Prolog-Term
 - ◆ Reihenfolge der Merkmale spielt keine Rolle, bei Prolog-Termen aber schon!
- ◆ Koreferenz ~ Prolog-Variablen
- ◆ gleiches Unifikationsverfahren

Entsprechend nahe liegt es, Merkmalstrukturen in Prolog zu implementieren.

Merkmalstrukturen in Prolog

$$\left[\text{Syntax} \left[\begin{array}{ll} \text{Genus} & \text{feminin} \\ \text{Kasus} & \text{dativ} \end{array} \right] \right]$$

Merkmalstruktur → komplexer Prolog-Term

- ◆ `(syntax(genus(feminin), kasus(dativ)))`
- ◆ Probleme:
 - ◆ `syntax(genus(feminin))` ist nicht mit `syntax(Numerus(plural))` unifizierbar; die entsprechenden Merkmalstrukturen wären es aber
 - ◆ ganz so einfach geht es also nicht!

Merkmalestrukturen in Prolog

$$\left[\textit{Syntax} \left[\begin{array}{ll} \textit{Genus} & \textit{feminin} \\ \textit{Kasus} & \textit{dativ} \end{array} \right] \right]$$

Merkmalestruktur → Prolog-Liste

- ◆ `[syntax([genus(feminin), kasus(dativ)])]`
 - ◆ `[syntax([genus(feminin)])]` ist nicht mit `[syntax([numerusb(plural)])]` unifizierbar; die entsprechenden Merkmalestrukturen wären es aber!
 - ◆ ein Prolog-Prädikat zur »Unifikation« (im Sinn von Merkmalestrukturen!) dieser Terme ist zwar relativ leicht zu definieren, jedoch nicht sonderlich effizient

Merkmalestrukturen in Prolog

$$\left[\textit{Syntax} \left[\begin{array}{ll} \textit{Genus} & \textit{feminin} \\ \textit{Kasus} & \textit{dativ} \end{array} \right] \right]$$

Merkmalestruktur \rightarrow Prolog-Term/-Liste mit fest zugewiesenen Argumentstellen

- ◆ z.B. *Genus* \rightarrow 5. Argumentstelle, *Kasus* \rightarrow 2. Argumentstelle
- ◆ `struktur(struktur(_, dativ, _, _, feminin))`
 - ◆ normale Prolog-Unifikation verwendbar: sehr effizient
 - ◆ mühsam zum Eingeben
 - ◆ Terme völlig unleserlich

Merkmalstrukturen in Prolog

Merkmalstrukturen sind also direkt in Prolog realisierbar

- ◆ aber dann ausgesprochen mühsam zu benutzen
- ◆ Lösung: Automatische Übersetzung
 - ◆ beim Einlesen der Grammatik: leserliche Ausdrücke → Prolog-Terme
 - ◆ beim Ausgeben der Ergebnisse: Prolog-Terme → leserliche Ausdrücke

Ein Beispiel für eine solche Übersetzung ist GULP.

- ◆ siehe ausgeteilte Kopien aus [Covington, 1994]

```
syntax:(genus:feminin..kasmus:dativ)
_g([_g([dativ, _, feminin]), _])
```

<i>Syntax</i>	<i>Genus</i>	<i>feminin</i>
	<i>Kasmus</i>	<i>dativ</i>

Merkmalestrukturen übersetzen

Einige weitere Ansätze, um Merkmalstrukturen in etwas anderes umzuwandeln (es existieren noch einige mehr):

- ◆ **ProFIT** — ähnlich wie GULP, jedoch mit mehr Ausdrucksmöglichkeiten; beim Konsultieren von Grammatik und Lexikon werden Merkmalstrukturen in Prolog-Terme übersetzt
- ◆ **ALE** — Compiler übersetzt die Merkmalstrukturen der Grammatik in Prolog-Programme und die Merkmalstrukturen des Lexikons in Prolog-Terme. Letztere verarbeitet das aus der Grammatik erzeugte Prolog-Programm dann effizient.
- ◆ **AMALIA** — Compiler übersetzt die Merkmalstrukturen der Grammatik in ein Programm für einen Interpreter, der in C geschrieben ist — ähnlich wie Prolog-Programme in Code für die Warren Abstract Machine übersetzt werden
- ◆ **Patti** — Compiler übersetzt die Merkmalstrukturen der Grammatik in optimierten Assembler-Code für einen RISC-Chip (PowerPC), wodurch bestimmte Mechanismen (Caches, Pipelines, Superskalarität) zur Effizienzsteigerung ausgenutzt werden

Literatur



Theoretische Grundlagen (getypter) Merkmalstrukturen

- ◆ Carpenter, Bob: The Logic of Typed Feature Structures with Applications to Unification Grammars, Logic Programs and Constraint Resolution. Cambridge Tracts in Theoretical Computer Science 32. Cambridge: Cambridge University Press, 1992. ISBN 0-521-41932-8 (gebunden).
 - ◆ das Standardwerk zum Thema
 - ◆ schafft sehr gute Grundlage für das tiefere Verständnis des Themas
 - ◆ dürfte ohne mathematische Vorkenntnisse allerdings nur schwer verständlich sein

Literatur



Linguistische Anwendungen: Übersicht

- ◆ Shieber, Stuart M.: An Introduction to Unification-Based Approaches to Grammar. CSLI Lecture Notes, vol. 4. Center for the Study of Language and Information: Palo Alto, 1986.
 - ◆ gute Einführung in praktische linguistische Anwendungen von Merkmalstrukturen
 - ◆ bespricht den Einsatz von Merkmalstrukturen in den damals gebräuchlichen linguistischen Theorien
 - ◆ empfehlenswert, auch wenn das Buch etwas an der Oberfläche bleibt

Literatur

Grammatiktheorien: Übersicht

- ◆ Sells, Peter: Lectures on Contemporary Syntactic Theories. An Introduction to Government-Binding Theory, Generalized Phrase Structure Grammar, and Lexical-Functional Grammar. CSLI Lecture Notes 3. 1985. ISBN 0-937073-13-X (gebunden) und 0-937073-14-8 (Paperback).
 - ◆ etwas veraltet, bleibt eher an der Oberfläche
 - ◆ lesenswert: Kapitel zur LFG

Grammatiktheorien mit Merkmalstrukturen: LFG

- ◆ Bresnan, Joan [Hrsg.]: The Mental Representation of Grammatical Relations. Cambridge, Mass.: MIT Press, 1982.

Literatur

Grammatiktheorien mit Merkmalstrukturen: HPSG

- ◆ Carl Pollard/Ivan A. Sag: Head-Driven Phrase Structure Grammar. Chicago/London: The University of Chicago Press, 1994. ISBN 0-226-67446-0 (gebunden) und 0-226-67447-9 (Paperback)
 - ◆ Sehr lesenswerte Einführungskapitel: Was bringt die Formalisierung der Linguistik? Wo sind die Fehler bei transformationellen Grammatiken? Auf welche Prinzipien stützt sich die HPSG? Wie ist das Konzept einer Universalgrammatik modellierbar?
 - ◆ Weitere Kapitel besonders aus linguistischer Sicht interessant, da zahlreiche Erscheinungen in natürlichen Sprachen (z.B. Kongruenz, Bindung, Schmarotzerlücken) im HPSG-Rahmen modelliert werden
 - ◆ keine besonderen Voraussetzungen zum Verständnis nötig