

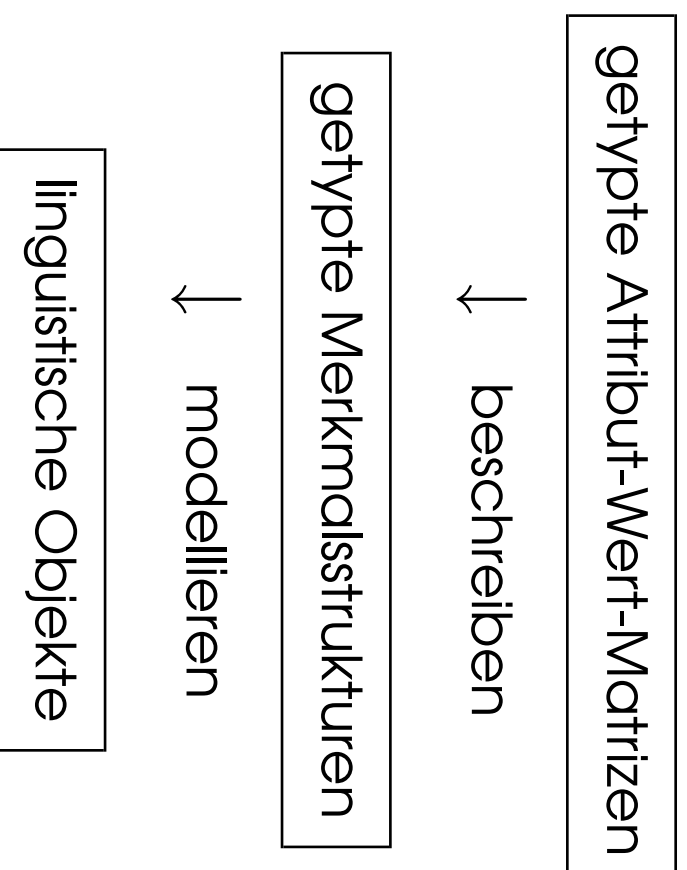
Typen/Sorten

- ▶ Bislang keine klare Strukturierung der Attribute und Werte

- ▶ **abschreckendes Beispiel:**

$$\left[\begin{array}{cc} \text{GENUS} & \text{akk} \\ \text{KAT} & \text{sg} \\ \text{NUMERUS} & \left[\begin{array}{cc} \text{GENUS} & \text{NP} \\ \text{KASUS} & \text{pl} \end{array} \right] \end{array} \right]$$

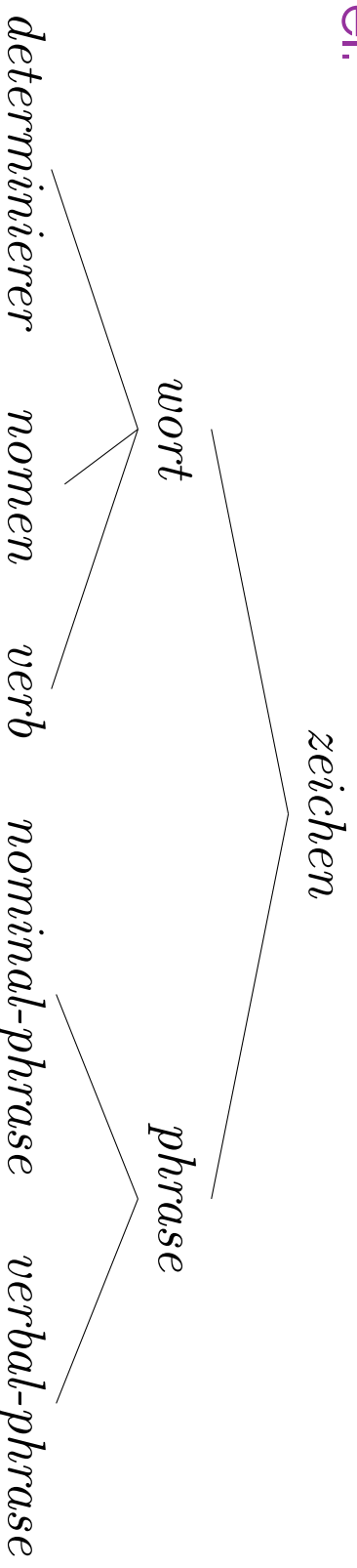
- ▶ Festlegung notwendig, welche Attribute welche Art von Wert haben
- ▶ **Typen/Sorten** stellen Klassen von Merkmalsstrukturen dar
- ▶ Legen damit genau fest, wie Merkmalsstrukturen aussehen können



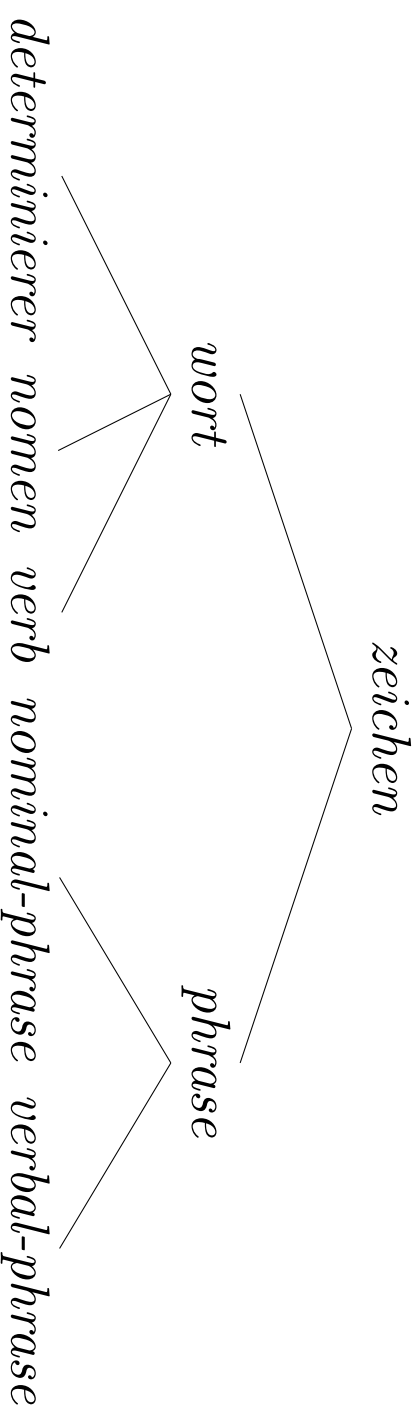
- ▶ Typen werden in zweierlei hinsicht verwendet:
 1. Auf der **Beschreibungsebene**, d.h. in Attribut-Wert-Matrizen
 2. Auf der **Modellierungsebene**, d.h. bei den Merkmalsstrukturen

- ▶ Typen werden in einer **Typhierarchie** angeordnet
- ▶ Diese Typhierarchie stellt eine Klassifikation der Merkmalsstrukturen (und damit der modellierten linguistischen Objekte) dar
- ▶ Welche es gibt und wie sie angeordnet sind hängt von der Grammatiktheorie ab

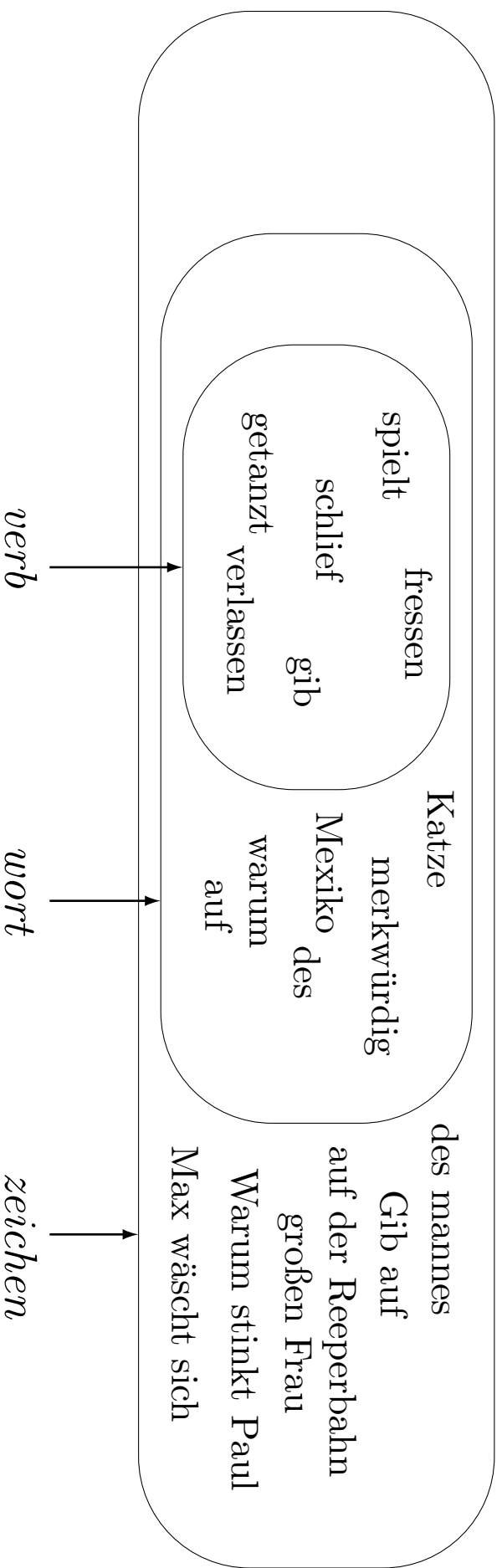
▶ **Beispiel:**



- ▶ Ein *verb* ist ein *wort* (und damit ein (linguistisches) *zeichen*), eine *verbal-phrase* ist eine *phrase* (und damit auch ein *zeichen*), etc.

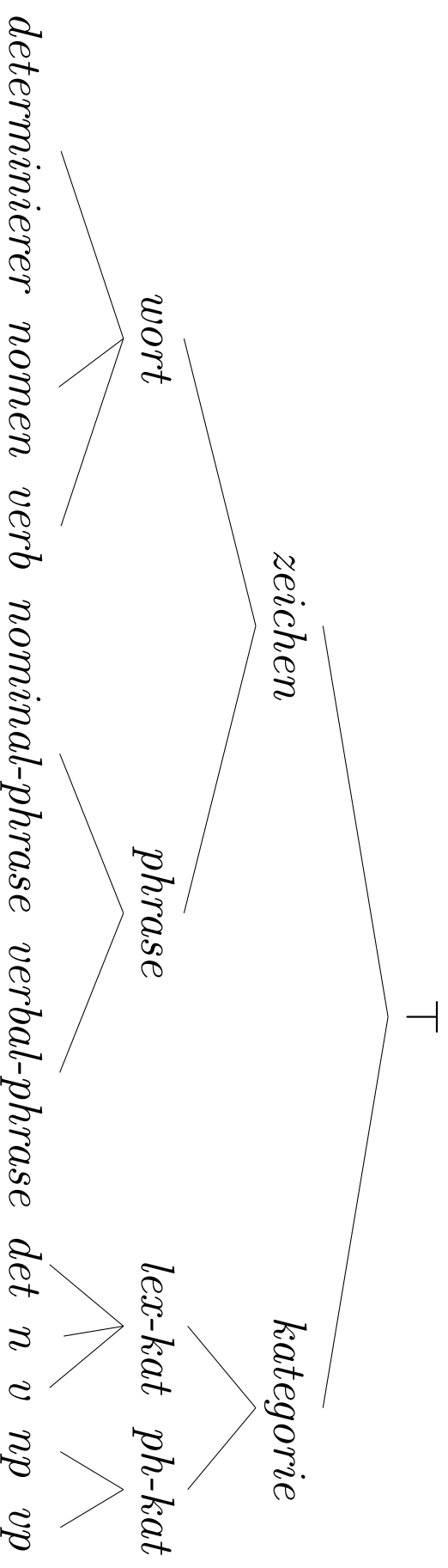


- ▶ Man spricht von **Supertypen** und **Subtypen**
- ▶ z.B. ist
 - wort* ein Subtyp von *zeichen*
 - zeichen* ein Supertyp von *wort*
 - nominal-phrase* ein Subtyp von *zeichen*, usw.
- ▶ Typen ohne Subtypen heissen **maximale** Typen



- ▶ Ist t Supertyp von t' , dann sind die MSen von $t \supseteq$ MSen von t'
- ▶ Typhierarchie muss die Ordnungseigenschaften erfüllen \Rightarrow **Subsumtion**
- ▶ t **subsumiert** t' , formal geschrieben als $t \sqsubseteq t'$
- ▶ hier: *zeichen* \sqsubseteq *wort* \sqsubseteq *verb*

Erweiterung:



- ▶ Neue Typen: *kategorie* mit Subtypen *lex-kat*, *ph-kat*, etc.
- ▶ Allgemeinster Typ T
- ▶ also z.B. *kategorie* \sqsubseteq *np*, *lex-kat* \sqsubseteq *v*, etc.

Angemessenheit von Attributen

- ▶ Für jeden Typ gibt es Information, welche Attribute mit welchen Werten für ihn **angemessen** sind.

- ▶ **Beispiel:**

Jedes *zeichen* hat ein Attribut KAT mit einem Wert vom Typ *kategorie*.

Angemessenheitsfunktion app (von engl. *appropriateness*)

$$app(typ, ATTRIBUT) = wert\text{-}typ$$

also:

$$app(\textit{zeichen}, KAT) = \textit{kategorie}$$

- ▶ Damit ist die Angemessenheitsfunktion für diese Werte **definiert** (\downarrow)

also: $app(\textit{zeichen}, KAT) = \downarrow$

Angemessenheit von Attributen

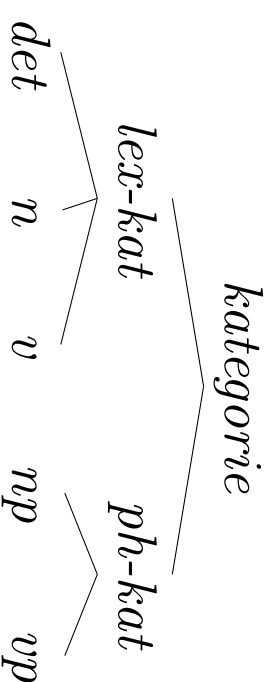
- ▶ Zunächst soll *zeichen* keine anderen Attribute haben (später: KGR)
- ▶ d.h. für andere Attribute ist *app* bei *zeichen* **undefiniert** (\uparrow)
also z.B.:
$$\text{app}(\textit{zeichen}, \text{GENUS}) = \uparrow$$
- ▶ An der Definiertheit/Undefiniertheit der Angemessenheitsfunktion ist ablesbar, welche Attribute eine AWM haben kann.

Vererbung

- ▶ Erinnerung:
Jedes *wort* ist ein *zeichen*, d.h. $\text{zeichen} \sqsubseteq \text{wort}$
- ▶ \Rightarrow weil KAT für *zeichen* angemessen, deshalb auch für *wort*
- ▶ Formaler:
 $\text{weil } \text{zeichen} \sqsubseteq \text{wort} \text{ und } \text{app}(\text{zeichen}, \text{KAT}) = \downarrow, \text{ deshalb } \text{app}(\text{wort}, \text{KAT}) = \downarrow$
- ▶ Allgemein: Ein Subtyp **erbt** alle Merkmale seiner Supertypen
- ▶ d.h. wenn $t \sqsubseteq t'$ und $\text{app}(t, A) = \downarrow$, dann auch $\text{app}(t', A) = \downarrow$
- ▶ Damit im Beispiel: *zeichen*, *wort*, *phrase*, *determinierer*, *nomen*, *verb*, *nominal-phrase*, *verbal-phrase* haben alle ein Merkmal KAT

Vererbung

Von welchem Typ ist der Wert von KAT bei den Subtypen von *zeichen*?



- ▶ Entweder auch vom Typ *kategorie*, oder von einem Subtyp von *kategorie*
- ▶ z.B. *wort* sollte nur Werte vom Typ *lex-kat* zulassen, und *phrase* nur Werte vom Typ *ph-kat*
- ▶ Also: $\text{app}(\text{wort}, \text{KAT}) = \text{lex-kat}$ und $\text{app}(\text{phrase}, \text{KAT}) = \text{ph-kat}$
- ▶ Allgemein muss gelten:
wenn $t \sqsubseteq t'$ und $\text{app}(t, A) = w$, dann $\text{app}(t', A) = w'$ wobei $w \sqsubseteq w'$

Vererbung

- Genauso: Ein *nomen* sollte nur KAT Werte vom Typ n zulassen, eine *verbal-phrase* nur Werte vom Typ vp , etc.
- Weiter Spezifizierung der vererbten Information:

$$\begin{aligned}\text{app}(\textit{determinierer}, KAT) &= \textit{det} \\ \text{app}(\textit{nomen}, KAT) &= \textit{n} \\ \text{app}(\textit{verb}, KAT) &= \textit{v} \\ \text{app}(\textit{nominal-phrase}, KAT) &= \textit{np} \\ \text{app}(\textit{verbal-phrase}, KAT) &= \textit{vp}\end{aligned}$$

Vererbung

- ▶ Weitere Spezifizierung: In Phrasen gibt es Töchter.
Jede Tochter ist vom Typ *zeichen*.

- ▶ Zwei weitere Attribute DTR1 und DTR2 sind für *phrase* angemessen:

$$\begin{aligned} \text{app}(\textit{phrase}, \text{DTR1}) &= \textit{zeichen} \\ \text{app}(\textit{phrase}, \text{DTR2}) &= \textit{zeichen} \end{aligned}$$

- ▶ Zusammenfassung:

- Subtypen können geerbte Merkmale weiter spezifizieren
- Subtypen können neue Merkmale einführen

Damit sind auch wohl-getyppt:

$zeichen \left[\begin{smallmatrix} KAT & ph-kat \end{smallmatrix} \right]$ denn $app(zeichen, KAT) = kategorie \sqsubseteq ph-kat$

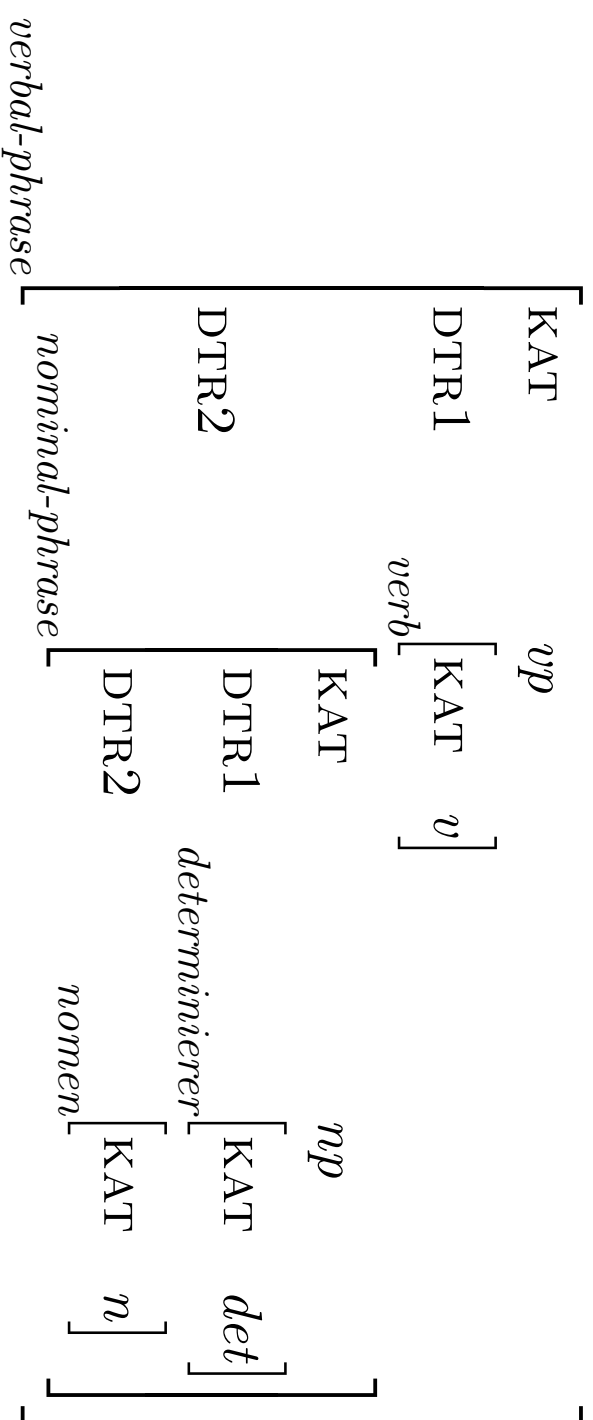
$zeichen \left[\begin{smallmatrix} KAT & np \end{smallmatrix} \right]$ denn $app(zeichen, KAT) = kategorie \sqsubseteq np$

$wort \left[\begin{smallmatrix} KAT & v \end{smallmatrix} \right]$ denn $app(wort, KAT) = lex-kat \sqsubseteq v$

$phrase \left[\begin{smallmatrix} KAT & vp \\ DTR2 & zeichen \end{smallmatrix} \right]$ denn $app(phrase, KAT) = ph-kat \sqsubseteq vp$
 $app(phrase, DTR2) = zeichen \sqsubseteq zeichen$

$verbal-phrase \left[\begin{smallmatrix} KAT & vp \\ DTR1 & verb \\ DTR2 & nominal-phrase \end{smallmatrix} \right]$ $app(verbal-phrase, KAT) = vp \sqsubseteq vp$
 $app(verbal-phrase, DTR1) = zeichen \sqsubseteq verb$
 $app(verbal-phrase, DTR2) = zeichen \sqsubseteq nominal-phrase$

Auch wohl-getypt:



Aber nicht:

$$\begin{array}{l} \text{zeichen} \left[\begin{array}{cc} \text{GENUS} & \text{T} \end{array} \right] \quad \text{⚡ GENUS Merkmal nicht angemessen für } \text{zeichen}, \\ \text{app}(\text{zeichen}, \text{GENUS}) = \uparrow \end{array}$$

$$\begin{array}{l} \text{wort} \left[\begin{array}{cc} \text{KAT} & \text{np} \end{array} \right] \quad \text{⚡ app}(\text{wort}, \text{KAT}) = \text{lex-kat} \not\subseteq \text{np} \end{array}$$

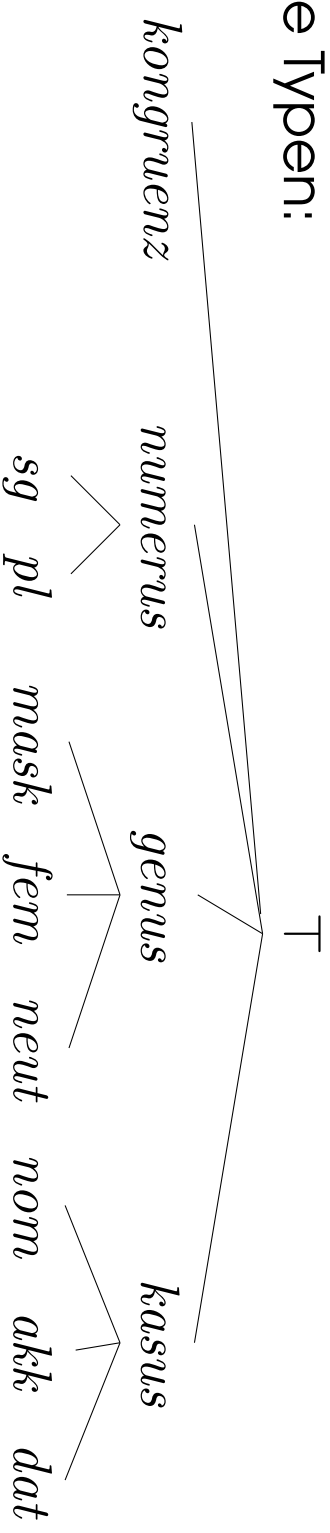
$$\begin{array}{l} \text{nomen} \left[\begin{array}{cc} \text{KAT} & \text{np} \end{array} \right] \quad \text{⚡ app}(\text{nomen}, \text{KAT}) = n \not\subseteq \text{np} \end{array}$$

Zusätzliche Restriktionen der Angemessenheit wären sinnvoll:

$$\begin{array}{l} \text{verbal-phrase} \left[\begin{array}{cc} \text{KAT} & \text{vp} \\ \text{DTR1} & \text{nomen} \\ \text{DTR2} & \text{nomen} \end{array} \right] \end{array}$$

Erweiterung:

► zusätzliche Typen:



► Für den Typ *kongruenz* :

$\text{app}(\textit{kongruenz}, \text{NUMERUS}) = \textit{numerus}$
 $\text{app}(\textit{kongruenz}, \text{GENUS}) = \textit{genus}$
 $\text{app}(\textit{kongruenz}, \text{KASUS}) = \textit{kasus}$
 $\text{app}(\textit{kongruenz}, \text{KAT}) = \uparrow$
 $\text{app}(\textit{kongruenz}, \text{KGR}) = \uparrow$

► Für den Typ *zeichen* :

$\text{app}(\textit{zeichen}, \text{KGR}) = \textit{kongruenz}$
 $\text{app}(\textit{kongruenz}, \text{NUMERUS}) = \uparrow$
 ...

► Damit sind auch wohl-getyppt:

$$\text{zeichen} \left[\begin{array}{c} \text{KGR} \\ \text{kongruenz} \end{array} \left[\begin{array}{cc} \text{NUMERUS} & \text{numerus} \\ \text{KASUS} & \text{dat} \end{array} \right] \right]$$

$$\text{app}(\text{kongruenz}, \text{NUMERUS}) = \text{numerus} \sqsubseteq \text{numerus}$$

$$\text{app}(\text{kongruenz}, \text{KASUS}) = \text{kasus} \sqsubseteq \text{dat}$$

$$\text{app}(\text{zeichen}, \text{KGR}) = \text{kongruenz} \sqsubseteq \text{kongruenz}$$

$$\text{phrase} \left[\begin{array}{c} \text{KAT} \\ \text{KGR} \\ \text{kongruenz} \end{array} \left[\begin{array}{c} \text{vp} \\ \left[\begin{array}{cc} \text{NUMERUS} & \text{sg} \\ \text{GENUS} & \text{fem} \\ \text{KASUS} & \text{dat} \end{array} \right] \end{array} \right] \right]$$

phrase erbt Angemessenheitsinfo
über KGR von *zeichen*

- ▶ **Achtung:** Wert-Typen für ein Merkmal heissen manchmal genauso wie das Merkmal selbst

siehe $\text{app}(\textit{kongruenz}, \text{NUMERUS}) = \textit{numerus}$

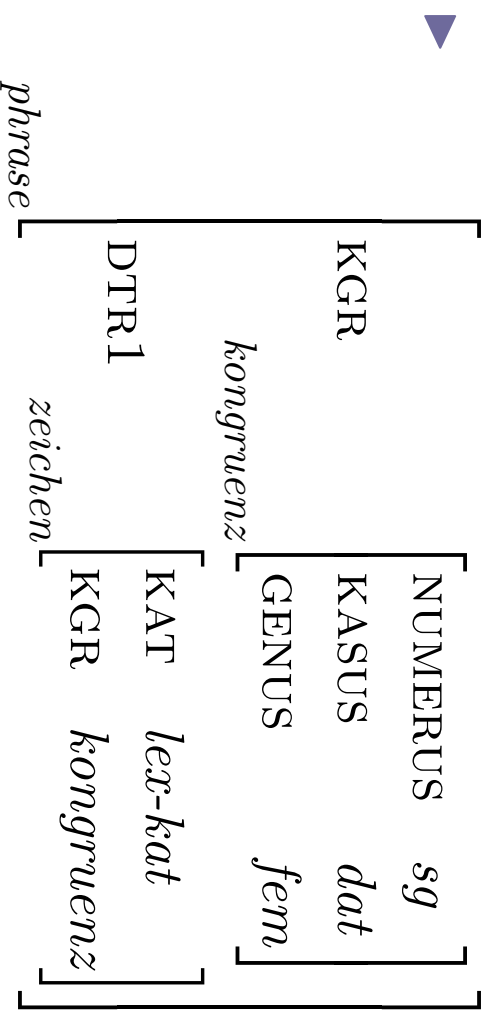
- ▶ Es *müssen* nicht alle angemessenen Merkmale angegeben werden
(\rightarrow **Unterspezifikation**)

- ▶ Falls doch alle angegeben sind, heisst die AWM **vollständig**

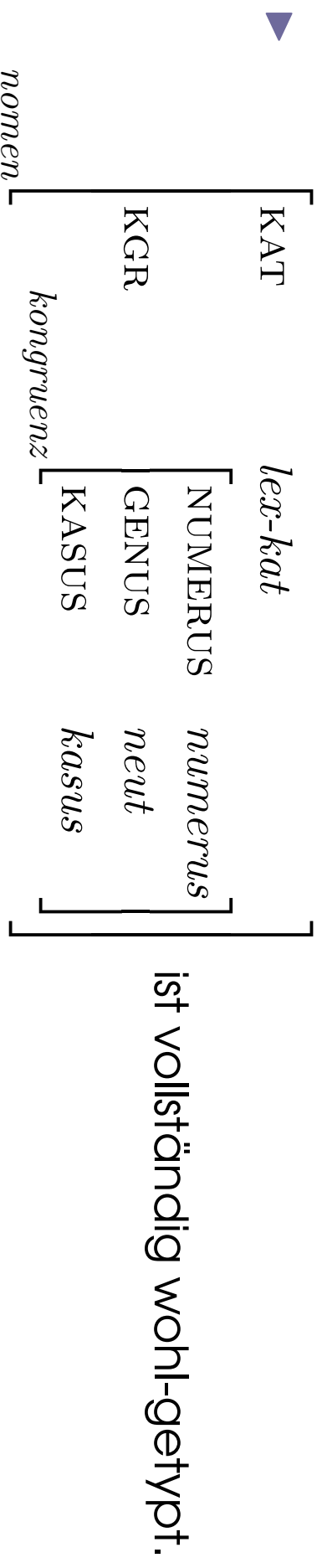
- ▶ **Beispiele:**

$$\textit{wort} \left[\begin{array}{cc} \text{KAT} & \textit{lex-kat} \\ \text{KGR} & \left[\begin{array}{cc} \text{NUMERUS} & \textit{sg} \\ \text{KASUS} & \textit{dat} \end{array} \right] \end{array} \right]$$

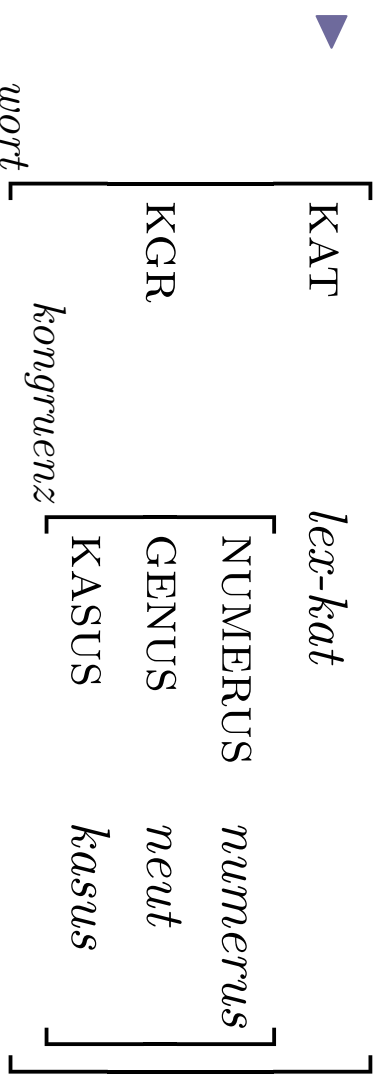
ist wohl-getypt, aber nicht vollständig
(GENUS ist angemessen für *kongruenz* aber nicht vorhanden)



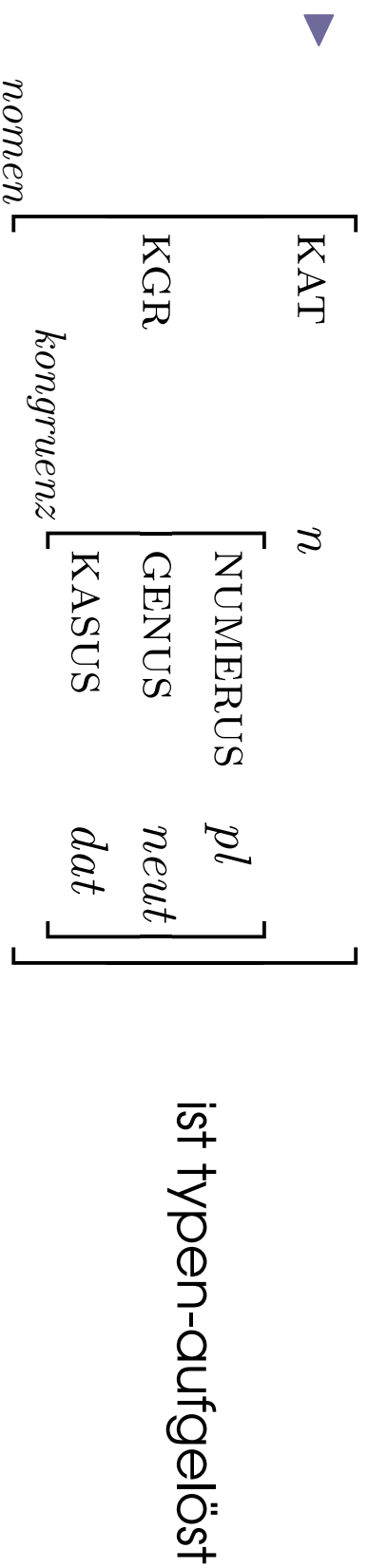
ist wohl-getyppt, aber nicht vollständig (KAT und DTR2 sind auch ange-messen für *phrase*)



Eine AWM in der alle Typen maximal sind heisst **typen-aufgelöst** (engl. type-resolved)



ist nicht typen-aufgelöst (*lex-kat*, *numerus*, *kasus* sind nicht maximal)



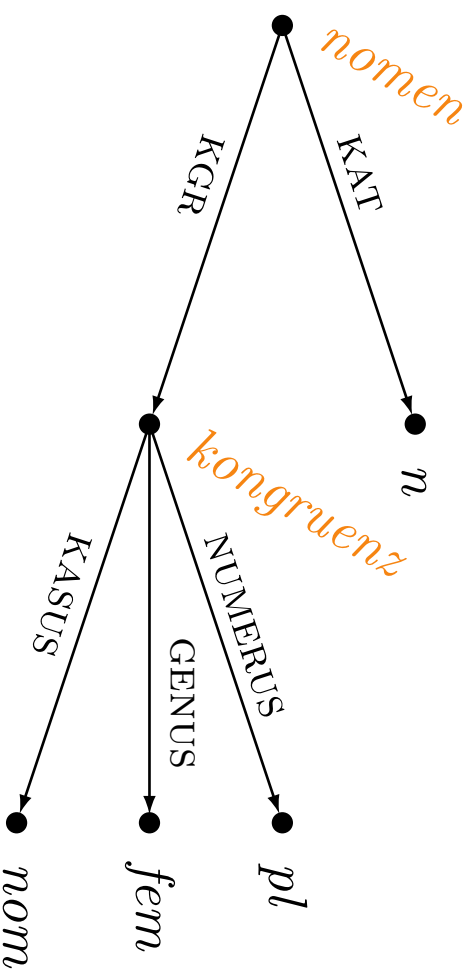
Zusammenfassung

- ▶ Typen klassifizieren Merkmalsstrukturen und damit die modellierten linguistischen Objekte
- ▶ Typen sind in einer Subsumtionshierarchie angeordnet
- ▶ Für jeden Typ können Merkmale angemessen sein
- ▶ angemessene Merkmale vererben sich auf Subtypen, können aber im Wert weiter spezifiziert werden
- ▶ AWM sind damit auch getypt und können wohl-getypt, vollständig und typen-aufgelöst sein

Getypete Merkmalsstrukturen

- ▶ **Erinnerung:**
Merkmalsstruktur \equiv gerichteter azyklischer etikettierter Graph
- ▶ Kantenetikette \equiv Attribute
- ▶ Alle getypeten Merkmalsstrukturen sind
 - wohl-getyp
 - vollständig
 - typen-aufgelöst
- ▶ Neu: Die Knoten sind mit den Typen etikettiert

Beispiel:



- ▶ Knotenetikette \equiv Typen
- ▶ Kantenetikette \equiv Attribute
- ▶ Wohlgetyptheit \equiv nur Kanten für angemessene Attribute
- ▶ Vollständigkeit \equiv Kanten für alle angemessenen Attribute vorhanden
- ▶ Typauflösththeit \equiv alle Typen maximal

AWM \rightarrow Merkmalsstruktur

getypte Attribut-Wert-Matrizen

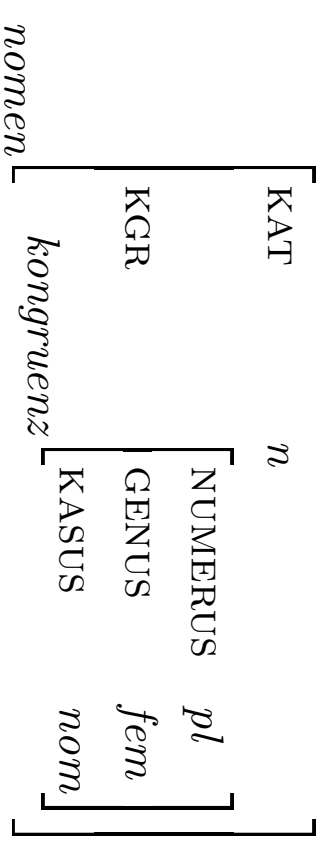
\downarrow beschreiben

getypte Merkmalsstrukturen

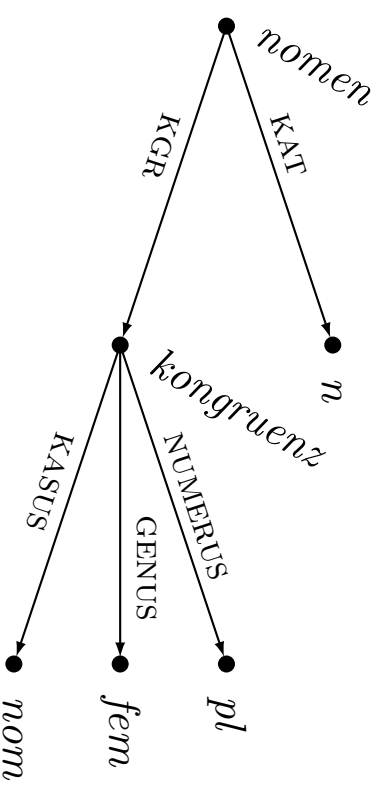
\downarrow modellieren

linguistische Objekte

- ▶ Eine wohlgetypte, vollständige, typenaufgelöste AWM beschreibt genau eine Merkmalsstruktur



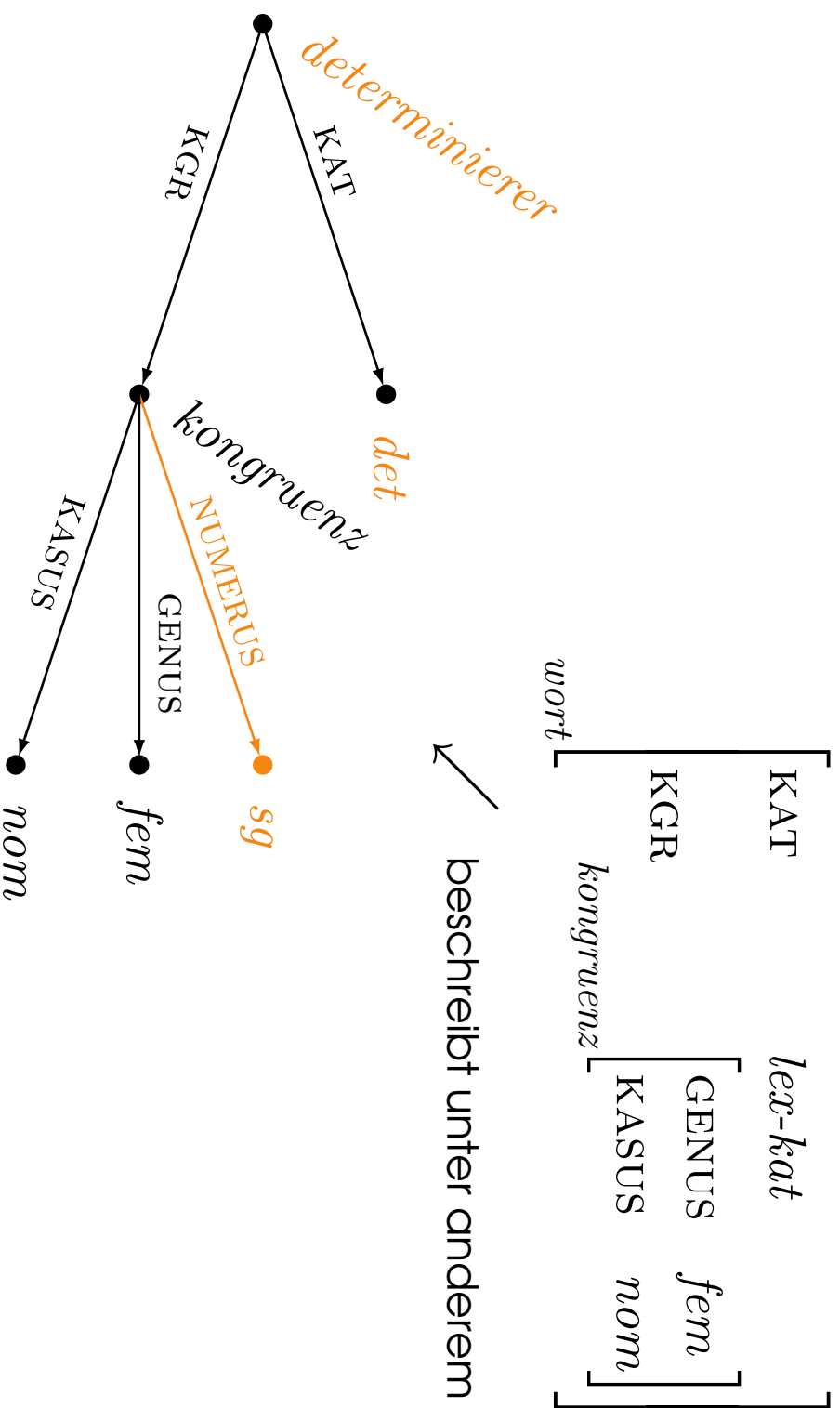
\downarrow beschreibt

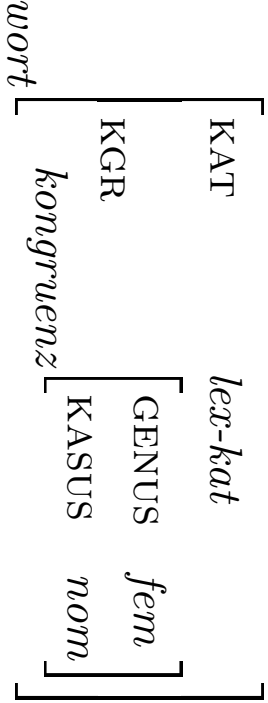


\downarrow modelliert

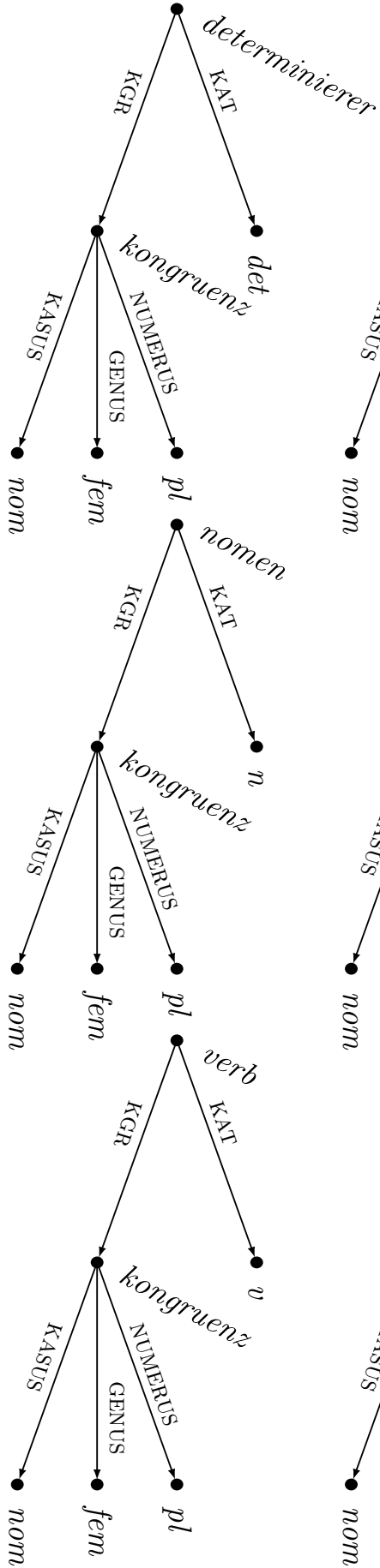
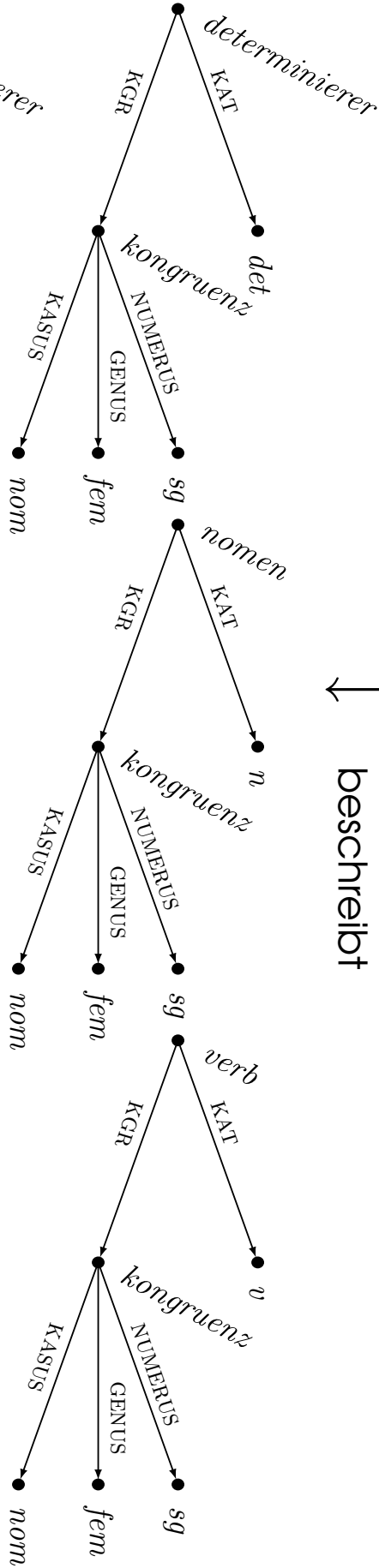
Frauen

Eine unvollständige oder nicht-typenaufgelöste AWM beschreibt mehrere Msen (\rightarrow **Unterspezifikation**)





↑ beschreibt



Subsumtion getypter AWMen

► Erinnerung:

Die Typhierarchie muss die Eigenschaften einer Subsumtionsrelation (\equiv Ordnungsrelation) aufweisen.

Typ t ist Supertyp von t' gdw. $t \sqsubseteq t'$

- Damit wird die Subsumtion von getypten AWMen definiert.
Eine AWM A vom Typ t subsumiert eine AWM A' vom Typ t' genau dann, wenn:
 - $t \sqsubseteq t'$ und
 - jedes Attribut in A ist auch in A' vorhanden und der Wert in A subsumiert den Wert in A
 - koreferente Attribute/Pfade in A sind auch koreferent in A' .

Beispiele:

$zeichen \left[\begin{smallmatrix} KAT \\ ph-kat \end{smallmatrix} \right] \sqsubseteq phrase \left[\begin{smallmatrix} KAT \\ np \end{smallmatrix} \right]$

$zeichen \sqsubseteq phrase$
 $ph-kat \sqsubseteq np$

$wort \not\sqsubseteq verbal-phrase \left[\begin{smallmatrix} KAT \\ vp \end{smallmatrix} \right]$

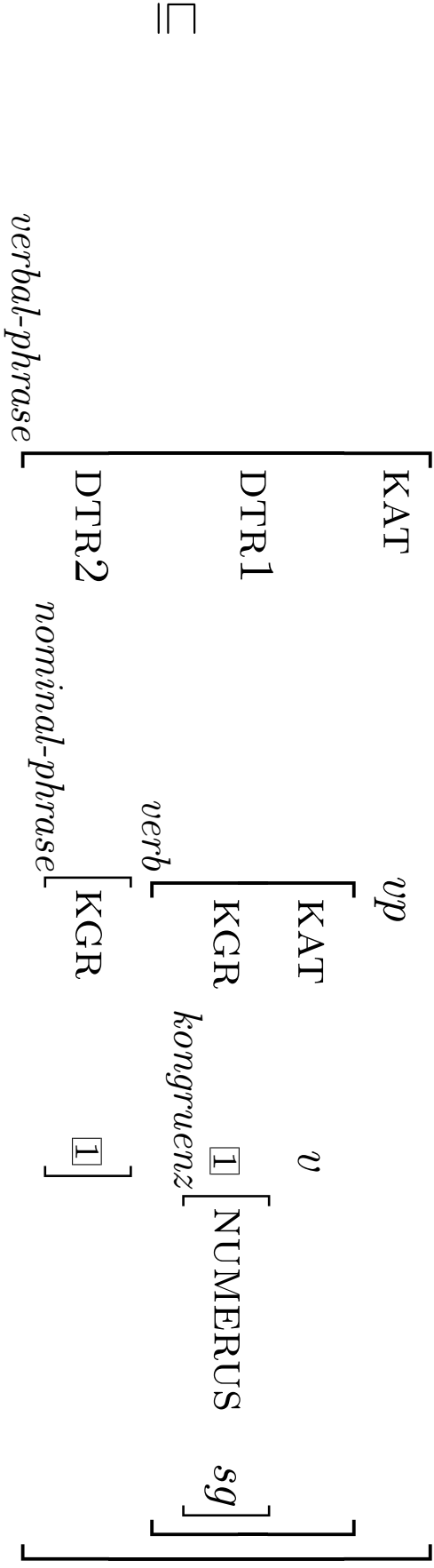
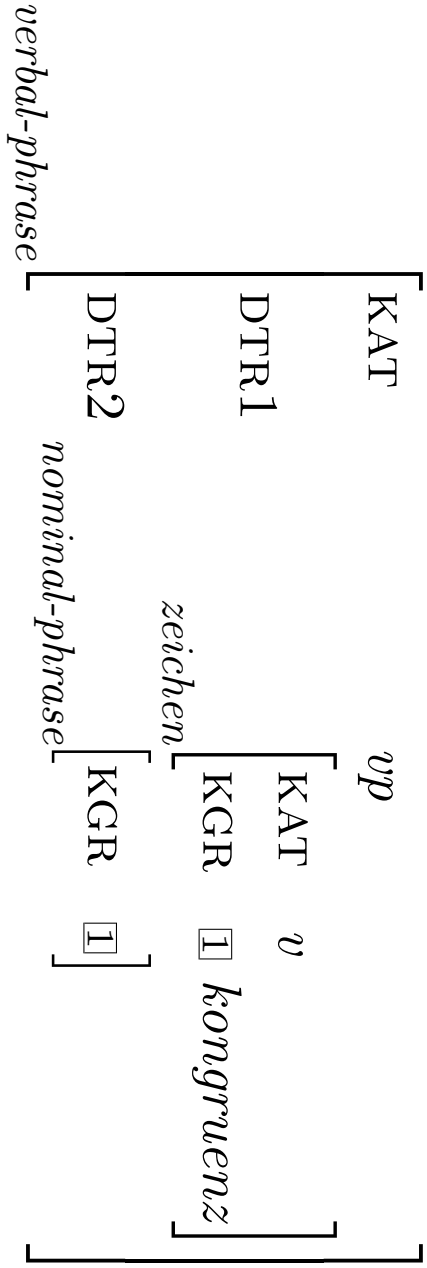
$wort \not\sqsubseteq verbal-phrase$

$phrase \left[\begin{smallmatrix} KAT \\ vp \end{smallmatrix} \right] \sqsubseteq verbal-phrase \left[\begin{smallmatrix} KAT \\ KGR \\ kongruenz \end{smallmatrix} \begin{smallmatrix} vp \\ NUMERUS \\ numerus \end{smallmatrix} \right]$

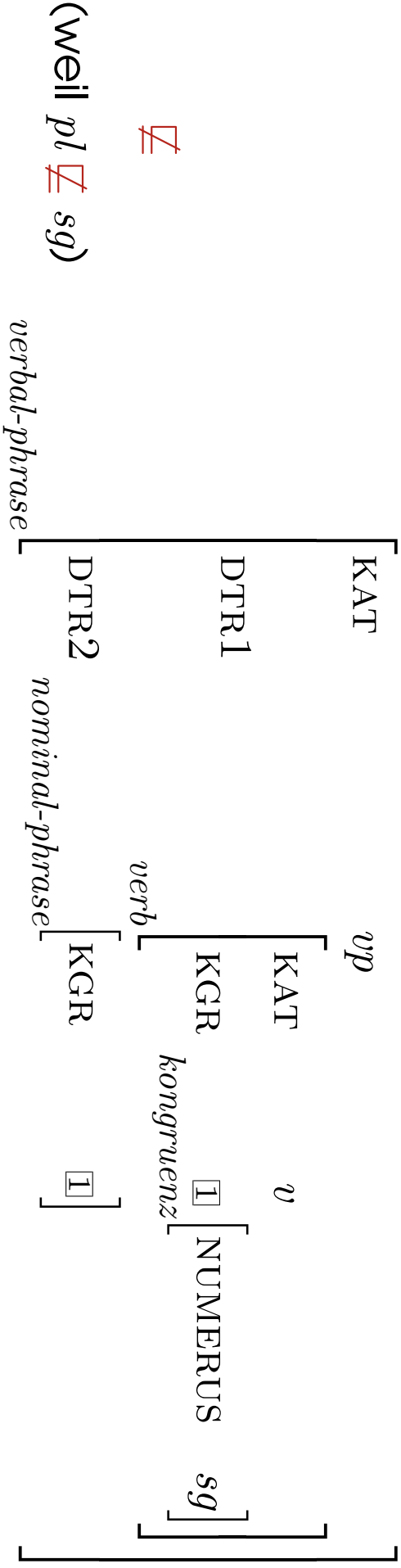
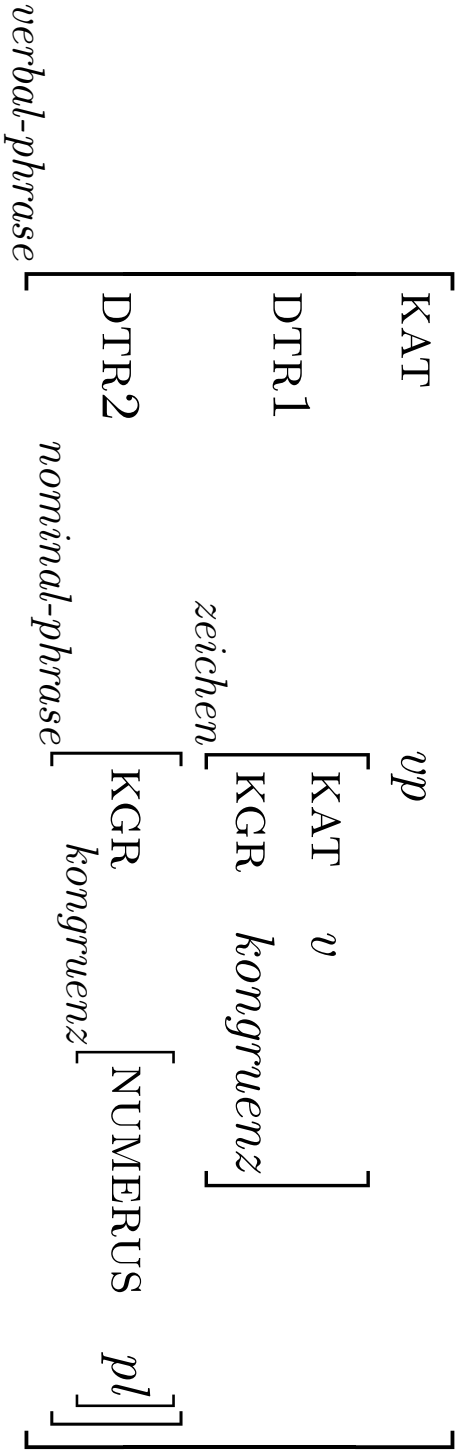
$phrase \left[\begin{smallmatrix} KAT \\ KGR \\ kongruenz \end{smallmatrix} \begin{smallmatrix} ph-kat \\ NUMERUS \\ GENUS \\ numerus \\ fem \end{smallmatrix} \right] \not\sqsubseteq verbal-phrase \left[\begin{smallmatrix} KAT \\ KGR \\ kongruenz \end{smallmatrix} \begin{smallmatrix} vp \\ NUMERUS \\ GENUS \\ sg \\ genus \end{smallmatrix} \right]$

$fem \not\sqsubseteq genus$

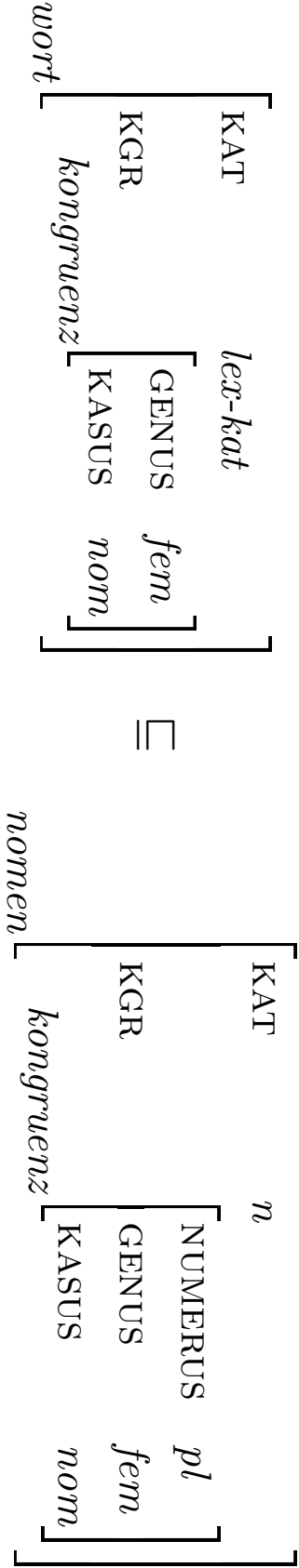
Weitere Beispiele:



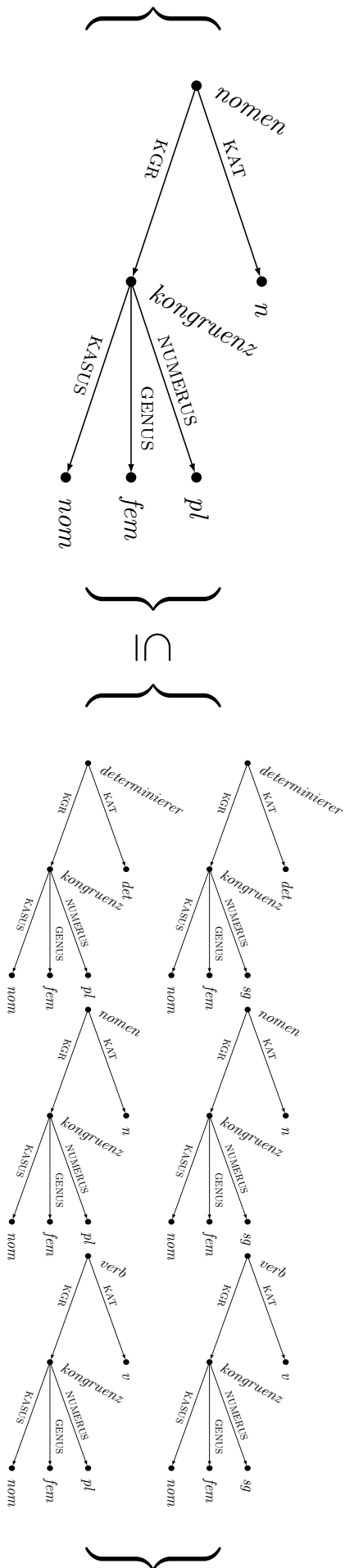
Weitere Beispiele:



Subsumtion \leftrightarrow MS-Teilmengenbeziehung $A \sqsubseteq B \Leftrightarrow MS(B) \subseteq MS(A)$



\Leftrightarrow



Unifikation getypter AWMen

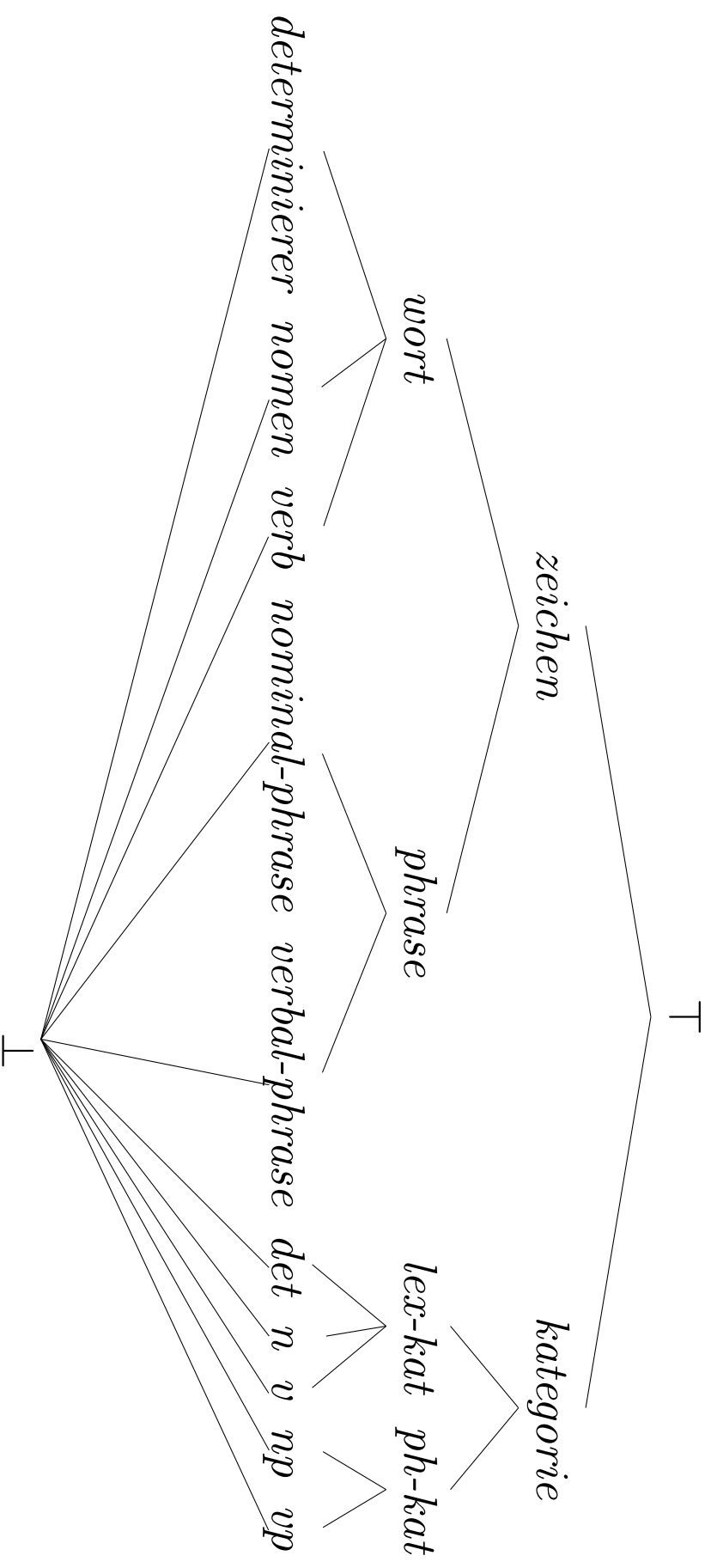
- ▶ Genauso wie bei ungetypten AWMen!
- ▶ $A \sqcup B$ = die allgemeinste AWM C , die $A \sqsubseteq C$ und $B \sqsubseteq C$
- ▶ Dafür muss aber die Unifikation von Typen $t \sqcup t'$ definiert sein ...

- ▶ **Preisfrage:**

Welche Bedingung muss die Typhierarchie dann erfüllen ????

- ▶ Für je zwei Typen t und t' muss das **Infimum** existieren, bzw.
- ▶ die Typhierarchie muss ein **(Halb-)Verband** sein

Noch kein Verband:



► Der alte Trick: \perp steht für fehlschlagende Unifikation

Beispiele:

$$\text{zeichen} \sqcup \text{wort} = \text{wort}$$

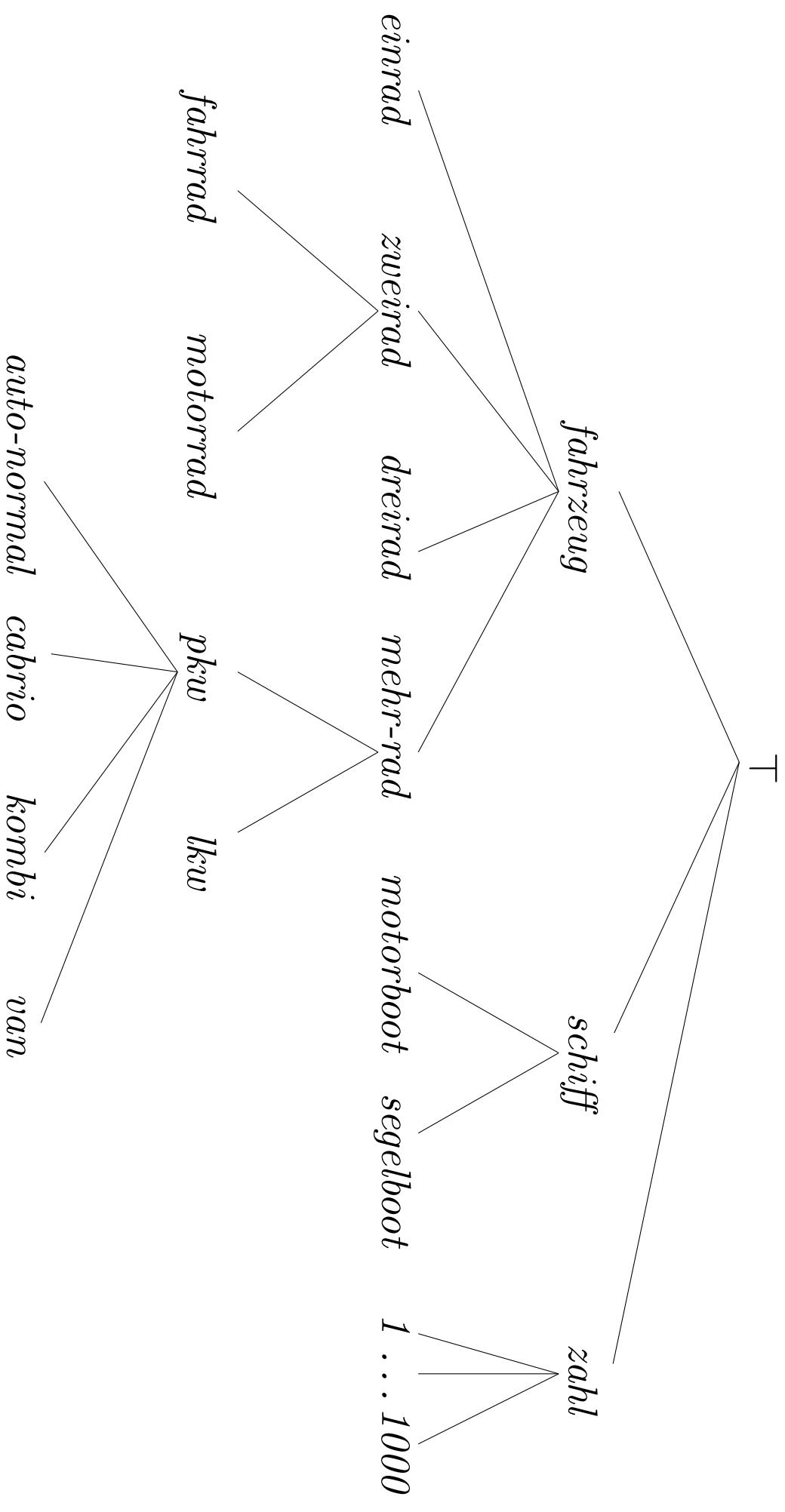
$$\text{lex-kat} \sqcup \text{nomen} = \perp$$

$$\text{zeichen} \left[\begin{array}{c} \text{KAT} \\ n \end{array} \right] \sqcup \text{wort} \left[\begin{array}{c} \text{KAT} \\ \text{lex-kat} \end{array} \right] = \text{wort} \left[\begin{array}{c} \text{KAT} \\ n \end{array} \right]$$

$$\text{phrase} \left[\begin{array}{c} \text{DTR1} \\ \text{zeichen} \end{array} \right] \sqcup \text{zeichen} \left[\begin{array}{c} \text{KAT} \\ \text{kategorie} \end{array} \right] = \text{phrase} \left[\begin{array}{c} \text{KAT} \\ \text{DTR1} \\ \text{ph-kat} \\ \text{zeichen} \end{array} \right]$$

$$\text{zeichen} \left[\begin{array}{c} \text{KGR} \\ \text{DTR1} \end{array} \begin{array}{c} \text{kongruenz} \\ \text{nomen} \end{array} \left[\begin{array}{c} \text{NUMERUS} \\ \text{GENUS} \end{array} \begin{array}{c} \text{pl} \\ \text{genus} \end{array} \right] \right] \sqcup \text{verbal-phrase} \left[\begin{array}{c} \text{KGR} \\ \text{DTR1} \end{array} \begin{array}{c} \text{kongruenz} \\ \text{zeichen} \end{array} \left[\begin{array}{c} \text{NUMERUS} \\ \text{GENUS} \end{array} \begin{array}{c} \text{numerus} \\ \text{fem} \end{array} \right] \right] = \perp$$

Typhierarchie (diesmal für nicht-linguistische Objekte); ⊥ nicht gezeigt



Angemessenheitsfunktion:

- ▶ Ein *fahrzeug* hat eine gewisse Anzahl Räder

$$\text{app}(\textit{fahrzeug}, \text{RÄDER}) = \textit{zahl}$$

- ▶ Spezialisierung: Ein *einrad* hat ein Rad

$$\text{app}(\textit{einrad}, \text{RÄDER}) = 1$$

(Erinnerung: $t \sqsubseteq t'$ und $\text{app}(t, A) = \downarrow$, dann $\text{app}(t, A) \sqsubseteq \text{app}(t', A)$)

- ▶ Spezialisierung: Ein *zweirad* hat zwei Räder

$$\text{app}(\textit{zweirad}, \text{RÄDER}) = 2$$

- ▶ Spezialisierung: Ein *dreirad* hat drei Räder

$$\text{app}(\textit{dreirad}, \text{RÄDER}) = 3$$

Angemessenheitsfunktion:

- **Spezialisierung**: nicht für *mehr-rad*, aber: ein *pkw* hat vier Räder

$$\text{app}(\textit{pkw}, \text{RÄDER}) = 4$$

- **Einführung neuer Attribute**: Ein *mehr-rad* hat Türen

$$\text{app}(\textit{mehr-rad}, \text{TÜREN}) = \textit{zahl}$$

- **Spezialisierung**: Ein *cabrio* hat 2 Türen

$$\text{app}(\textit{cabrio}, \text{TÜREN}) = 2$$

- **Spezialisierung**: Ein *kombi* hat 4 Türen

$$\text{app}(\textit{kombi}, \text{TÜREN}) = 4$$

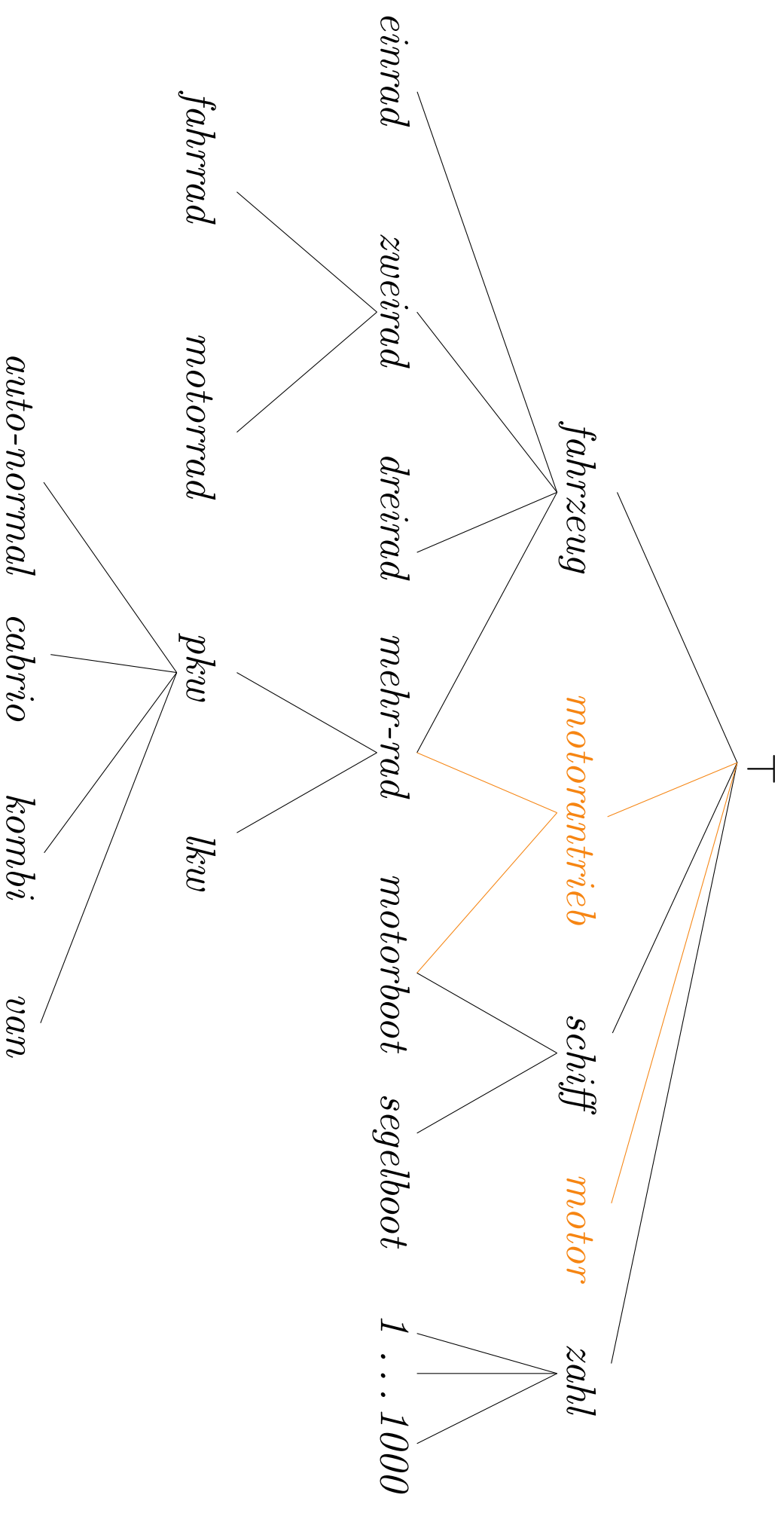
- Sonst für alle Typen t und Attribute A :

$$\text{app}(t, A) = \uparrow$$

AWMn über dieser Typhierarchie

- ▶ $motorad \left[\begin{array}{c} RÄDER \\ 2 \end{array} \right]$ wohl-getyppt, vollständig, typen-aufgelöst
- ▶ $motorboot \left[\begin{array}{c} RÄDER \\ zahl \end{array} \right]$ nicht wohl-getyppt (app(*motorboot*, RÄDER) = \uparrow)
- ▶ $fahrzeug \left[\begin{array}{c} RÄDER \\ TÜRREN \\ 4 \\ 2 \end{array} \right]$ nicht wohl-getyppt (app(*fahrzeug*, TÜRREN) = \uparrow)
- ▶ $mehr-rad \left[\begin{array}{c} RÄDER \\ 6 \end{array} \right]$ wohl-getyppt, nicht vollständig (TÜRREN),
nicht typen-aufgelöst (*mehr-rad*)
- ▶ $kombi \left[\begin{array}{c} TÜRREN \\ 4 \end{array} \right]$ wohl-getyppt, nicht vollständig (RÄDER),
typen-aufgelöst
- ▶ $auto-normal \left[\begin{array}{c} RÄDER \\ TÜRREN \\ 4 \\ zahl \end{array} \right]$ wohl-getyppt, vollständig,
nicht typen-aufgelöst (*zahl*)

Erweiterung der Typhierarchie



Erweiterung der Angemessenheitsfunktion:

- ▶ Ein Ding vom Typ *motorantrieb* hat einen Motor (vom Typ *motor*)

$\text{app}(\text{motorantrieb}, \text{MOTOR}) = \text{motor}$

- ▶ Ein *motor* hat eine gewisse Anzahl Zylinder und PS

$\text{app}(\text{motor}, \text{ZYLINDER}) = \text{zahl}$

$\text{app}(\text{motor}, \text{PS}) = \text{zahl}$

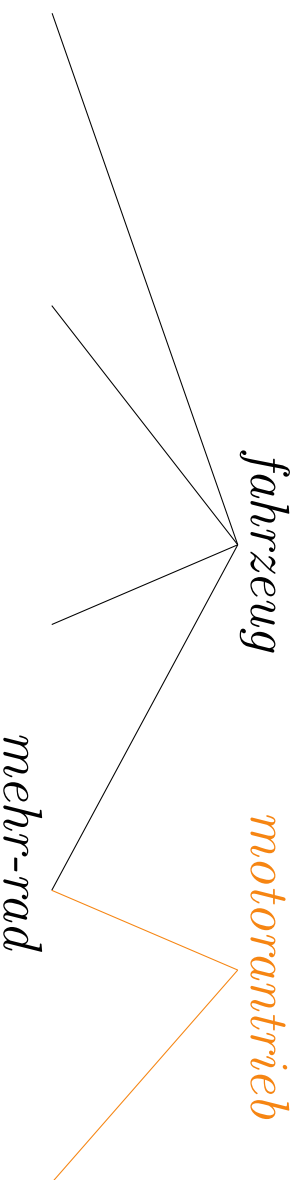
- ▶ wohl-getyppte und vollständige AWM z.B.

$$\text{motorantrieb} \left[\begin{array}{c} \text{MOTOR} \\ \text{PS} \end{array} \right] \left[\begin{array}{c} \text{ZYLINDER} \\ \text{PS} \end{array} \right] \left[\begin{array}{c} 4 \\ 150 \end{array} \right]$$

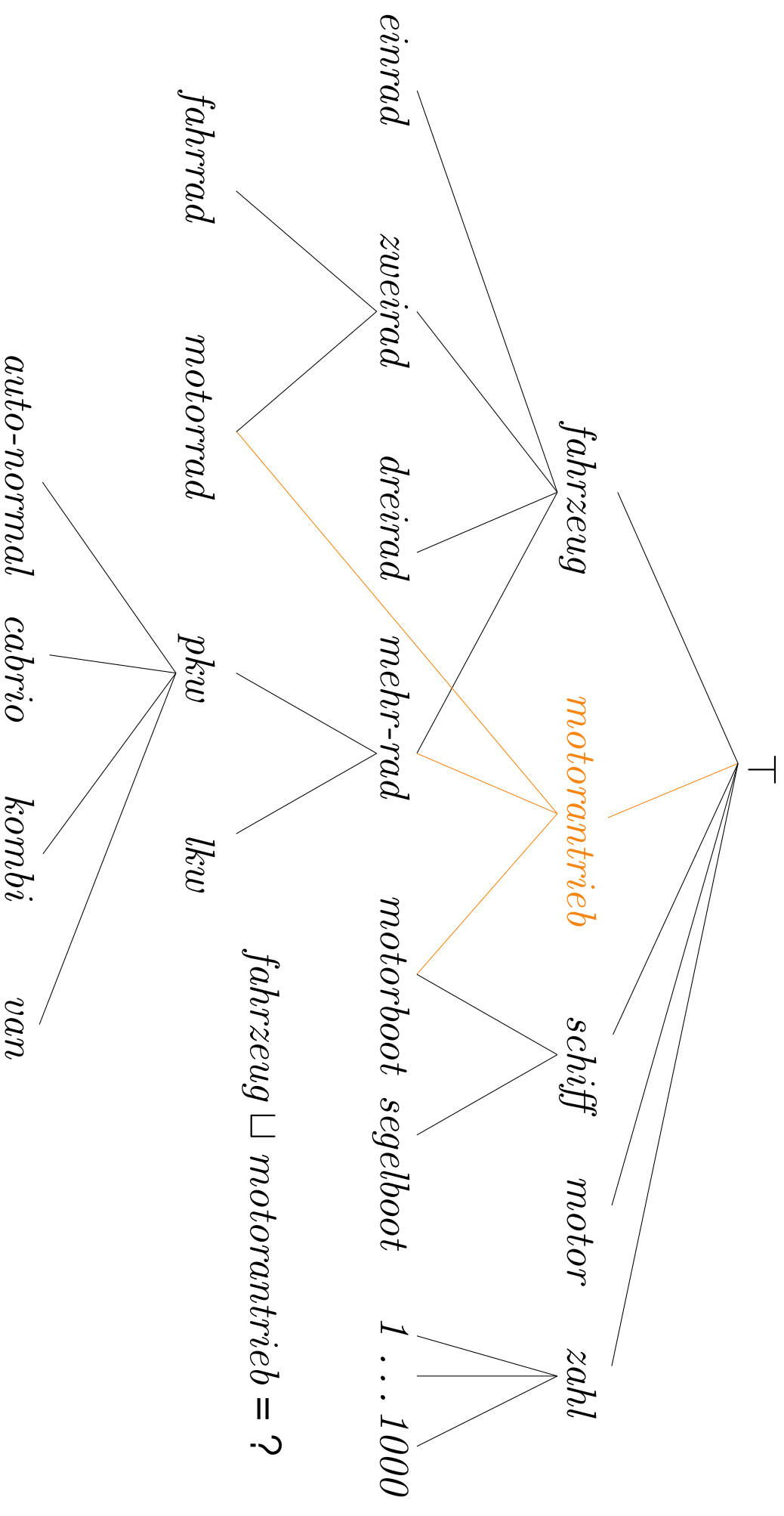
- ▶ *motorboot* hat zwei (unmittelbare) Supertypen: *schiff* und *motorantrieb*
- ▶ *mehr-rad* hat zwei (unmittelbare) Supertypen: *fahrzeug* und *motorantrieb*
- ▶ Eine Typenhierarchie in der manche Typen mehrere (unmittelbare) Supertypen haben heisst **multi-dimensional**
- ▶ Vorsicht! Infimum zweier Typen muss weiterhin existieren für Unifikation

▶ **Beispiel:**

fahrzeug \sqcup *motorantrieb* = *mehr-rad*



Beispiel einer unerlaubten Typhierarchie:



► **Erinnerung:**

Typen *erben* die Angemessenheitsinformation ihrer Supertypen

- In multi-dimensionalen Typhierarchien kann ein Typ mehrere Supertypen haben, d.h. er erbt von mehreren Supertypen/Dimensionen

- Das nennt man **Mehrfachvererbung** (*multiple inheritance*)

► **Beispiel:**

Ein *motor-fahrzeug* erbt von *fahrzeug* und von *motorantrieb*:

von *fahrzeug* $\text{app}(\text{fahrzeug}, \text{RÄDER}) = \text{zahl}$

von *motorantrieb* $\text{app}(\text{motorantrieb}, \text{MOTOR}) = \text{motor}$

- Damit ist z.B.

motor-fahrzeug $\left[\begin{array}{cc} \text{RÄDER} & 4 \\ \text{MOTOR} & \left[\begin{array}{cc} \text{ZYLINDER} & 4 \\ \text{PS} & 120 \end{array} \right] \end{array} \right]$ wohlgetypt und vollständig

Beispiele (**W**ohlgetyptheit, **V**ollständigkeit, **T**ypen-aufgelöstheit):

$$\begin{array}{c} \blacktriangledown \\ \text{mehr-rad} \end{array} \left[\begin{array}{cc} \text{RÄDER} & \text{zahl} \\ \text{MOTOR} & \left[\begin{array}{cc} \text{ZYLINDER} & 9877 \\ \text{PS} & \text{zahl} \end{array} \right] \end{array} \right]$$

+W **-V** **-T**

$$\begin{array}{c} \blacktriangledown \\ \text{motorrad} \end{array} \left[\begin{array}{cc} \text{RÄDER} & \text{zahl} \\ \text{MOTOR} & \left[\begin{array}{cc} \text{ZYLINDER} & 9877 \\ \text{PS} & \text{zahl} \end{array} \right] \end{array} \right]$$

-W app(*zweirad*, RÄDER) = 2

$$\begin{array}{c} \blacktriangledown \\ \text{kombi} \end{array} \left[\begin{array}{cc} \text{RÄDER} & 4 \\ \text{TÜREN} & 4 \\ \text{MOTOR} & \left[\begin{array}{cc} \text{ZYLINDER} & 24 \\ \text{PS} & 170 \end{array} \right] \end{array} \right]$$

+W **+V** **+T**

Beispiele (Subsumtion):

$$\text{motorantrieb} \left[\begin{array}{c} \text{MOTOR} \\ \text{motor} \left[\begin{array}{c} \text{ZYLINDER} \quad 8 \\ \text{PS} \quad \text{zahl} \end{array} \right] \end{array} \right] \quad \not\sqsubseteq \quad \text{pkw} \left[\begin{array}{c} \text{RÄDER} \quad 4 \\ \text{MOTOR} \left[\begin{array}{c} \text{ZYLINDER} \quad 16 \\ \text{PS} \quad 200 \end{array} \right] \end{array} \right]$$

$$\text{motorantrieb} \quad \not\sqsubseteq \quad \text{zweirad} \left[\begin{array}{c} \text{RÄDER} \quad 2 \end{array} \right]$$

$$\text{motor-fahrzeug} \left[\begin{array}{c} \text{MOTOR} \\ \text{motor} \left[\begin{array}{c} \text{PS} \quad \text{zahl} \end{array} \right] \end{array} \right] \quad \not\sqsubseteq \quad \text{motorboot} \left[\begin{array}{c} \text{MOTOR} \left[\begin{array}{c} \text{ZYLINDER} \quad 16 \\ \text{PS} \quad 250 \end{array} \right] \end{array} \right]$$

$$\text{fahrzeug} \left[\begin{array}{c} \text{RÄDER} \quad 2 \end{array} \right] \quad \not\sqsubseteq \quad \text{pkw} \left[\begin{array}{c} \text{RÄDER} \quad 4 \\ \text{MOTOR} \left[\begin{array}{c} \text{ZYLINDER} \quad 16 \\ \text{PS} \quad 250 \end{array} \right] \end{array} \right]$$

Beispiele (Unifikation):

$$\blacktriangleright \text{motorantrieb} \sqcup \text{zweirad} \left[\begin{array}{c} \text{RÄDER} \\ 2 \end{array} \right] = \text{motorrad} \left[\begin{array}{c} \text{RÄDER} \\ 2 \end{array} \right]$$

$$\blacktriangleright \text{motorantrieb} \left[\begin{array}{c} \text{MOTOR} \\ \text{motor} \left[\begin{array}{c} \text{PS} \\ 340 \end{array} \right] \end{array} \right] \sqcup \text{lkw} \left[\begin{array}{c} \text{TÜREN} \\ \text{MOTOR} \left[\begin{array}{c} \text{PS} \\ 355 \end{array} \right] \end{array} \right] = \perp$$

$$\blacktriangleright \text{fahrzeug} \left[\begin{array}{c} \text{RÄDER} \\ 8 \end{array} \right] \sqcup \text{motorantrieb} \left[\begin{array}{c} \text{MOTOR} \\ \text{motor} \left[\begin{array}{c} \text{ZYLINDER} \\ \text{PS} \end{array} \right] \left[\begin{array}{c} 8 \\ \text{zahl} \end{array} \right] \end{array} \right]$$

$$= \text{motor-fahrzeug} \left[\begin{array}{c} \text{RÄDER} \\ \text{MOTOR} \left[\begin{array}{c} \text{ZYLINDER} \\ \text{PS} \end{array} \right] \left[\begin{array}{c} 8 \\ \text{zahl} \end{array} \right] \end{array} \right]$$

Erweiterungen: Disjunktion

- Disjunktion: wie bei ungetypten AWMn (unter Beachtung der Typen)

- **Beispiel:**

steht für

$$lk_w \left[\begin{array}{cc} \text{RÄDER} & 6 \vee 8 \\ \text{MOTOR} & lk_{motor} \left[\begin{array}{cc} \text{ZYLINDER} & 12 \vee 16 \end{array} \right] \end{array} \right]$$

$$lk_w \left[\begin{array}{cc} \text{RÄDER} & 6 \\ \text{MOTOR} & lk_{motor} \left[\begin{array}{cc} \text{ZYLINDER} & 12 \end{array} \right] \end{array} \right]$$

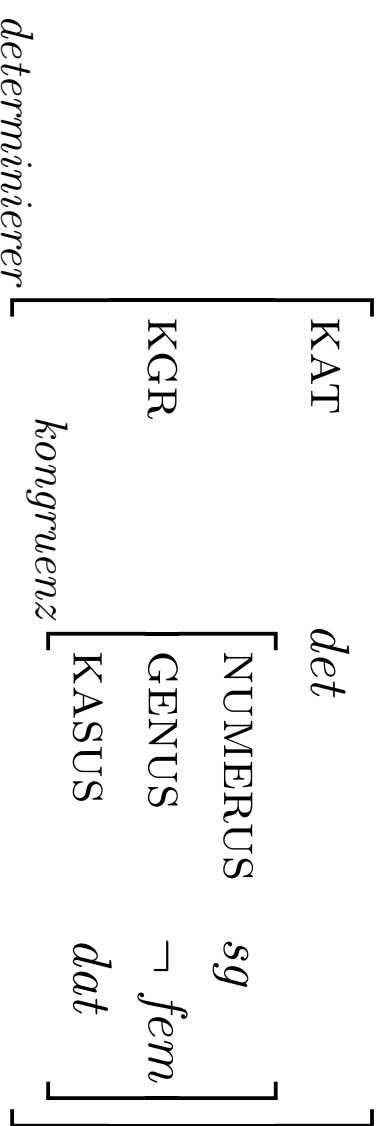
$$lk_w \left[\begin{array}{cc} \text{RÄDER} & 8 \\ \text{MOTOR} & lk_{motor} \left[\begin{array}{cc} \text{ZYLINDER} & 12 \end{array} \right] \end{array} \right]$$

$$lk_w \left[\begin{array}{cc} \text{RÄDER} & 6 \\ \text{MOTOR} & lk_{motor} \left[\begin{array}{cc} \text{ZYLINDER} & 16 \end{array} \right] \end{array} \right]$$

$$lk_w \left[\begin{array}{cc} \text{RÄDER} & 8 \\ \text{MOTOR} & lk_{motor} \left[\begin{array}{cc} \text{ZYLINDER} & 16 \end{array} \right] \end{array} \right]$$

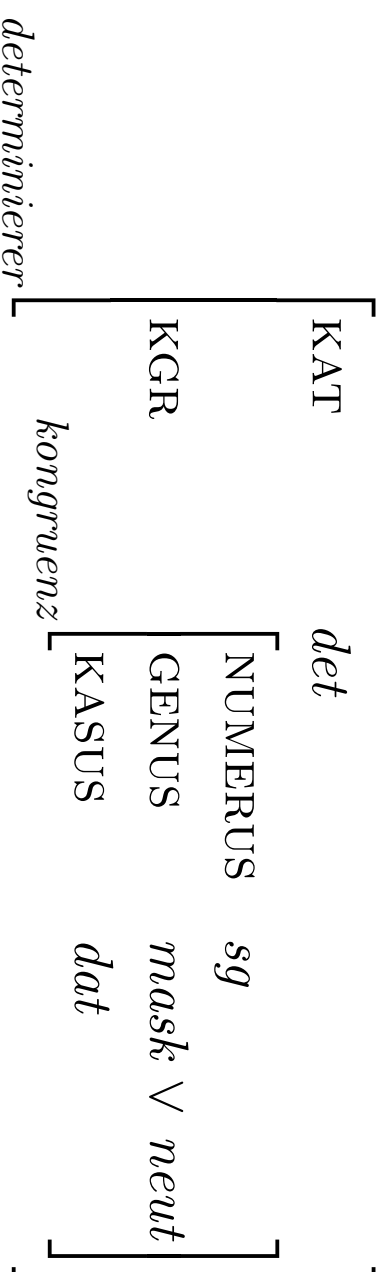
Erweiterungen: Negation

- ▶ Mit Typen auch möglich: **Negation** \neg
- ▶ Auch wieder nur abkürzend für mehrere AWMn
- ▶ z.B. Determinierer dem ist singular, im Dativ und *nicht* feminin
und durch folgende AWM beschrieben (mit der *zeichen*-Typhierarchie):



Erweiterungen: Negation

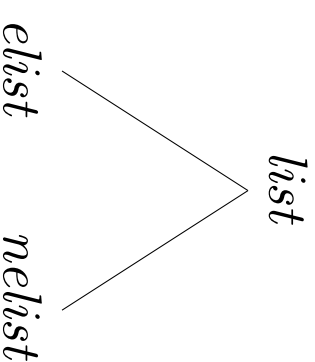
- ▶ Weil $\text{app}(\textit{kongruenz}, \text{GENUS}) = \textit{genus} \quad \dots$
- ▶ \dots und *genus* die Subtypen *mask*, *fem*, *neut* hat...
- ▶ \dots steht diese AWM für



Erweiterungen: Listen

- ▶ Listen mittels Typen:

Eine Liste ist leer (*elist*) oder nicht-leer (*nelist*)

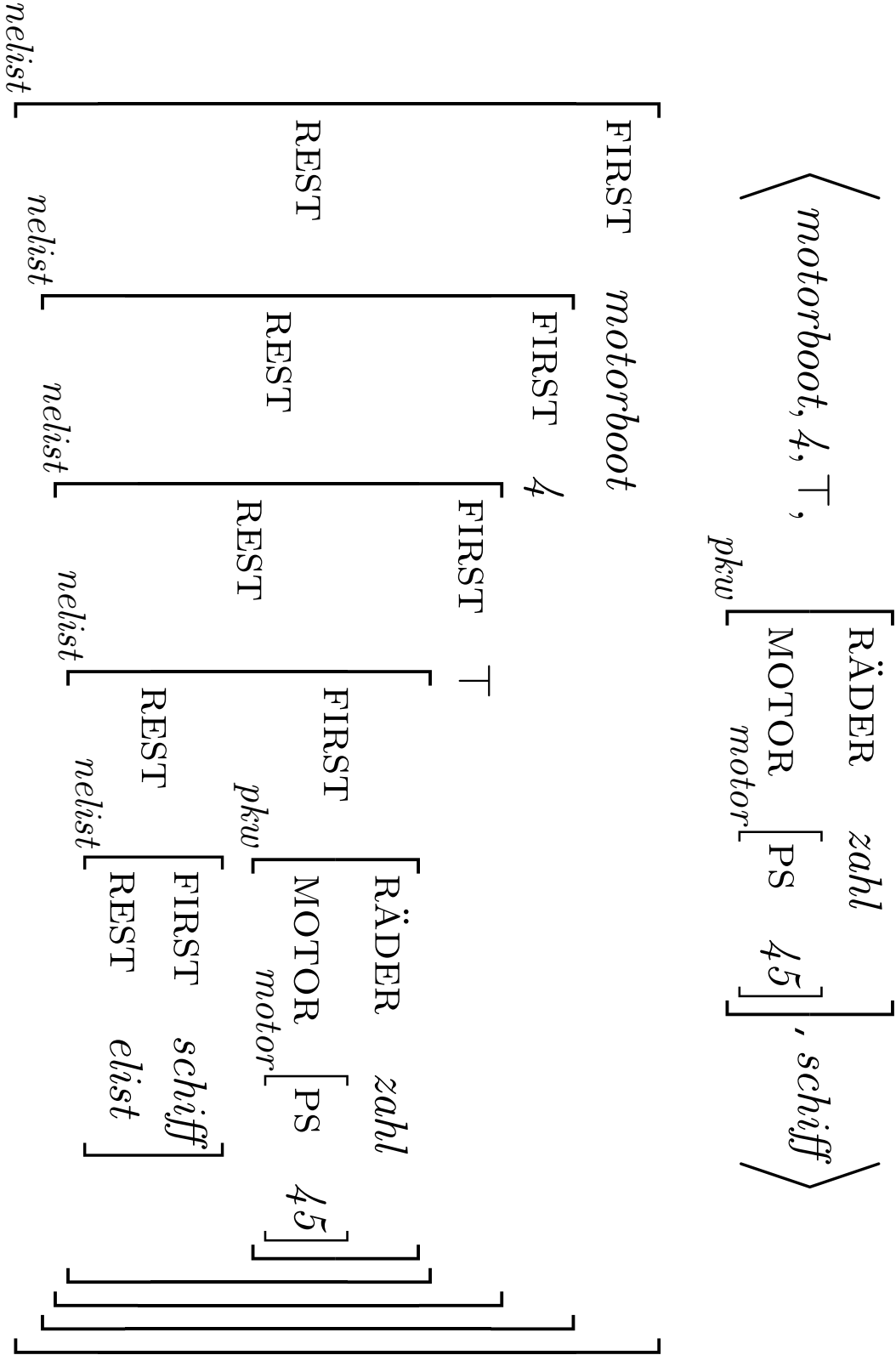


- ▶ Ist die Liste nicht-leer, dann besteht sie aus einem ersten Element (FIRST) und einer Restliste (REST)

$$\begin{aligned} \text{app}(\text{nelist}, \text{FIRST}) &= \text{T} \\ \text{app}(\text{nelist}, \text{REST}) &= \text{list} \end{aligned}$$

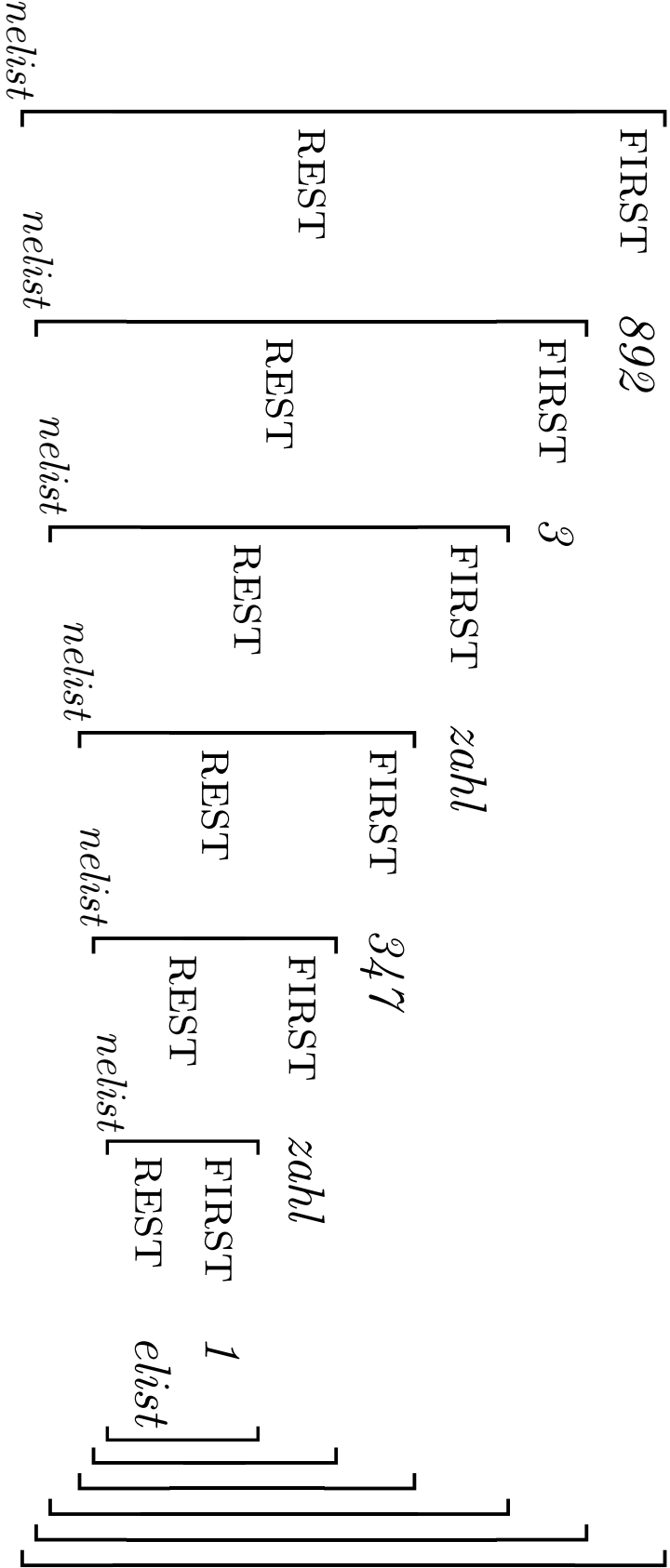
- ▶ Anstelle von T kann auch ein anderer Typ stehen, z.B. Liste von Zahlen \rightsquigarrow *zahl*, Liste von Fahrzeugen \rightsquigarrow *fahrzeug*, etc.

Beispiel: (mit Typenhierarchie von vorher)



Beispiel: Liste von Zahlen (d.h. $\text{app}(\text{nelist}, \text{FIRST}) = \text{zahl}$)

$$\langle 892, 3, \text{zahl}, 347, \text{zahl}, 1 \rangle$$



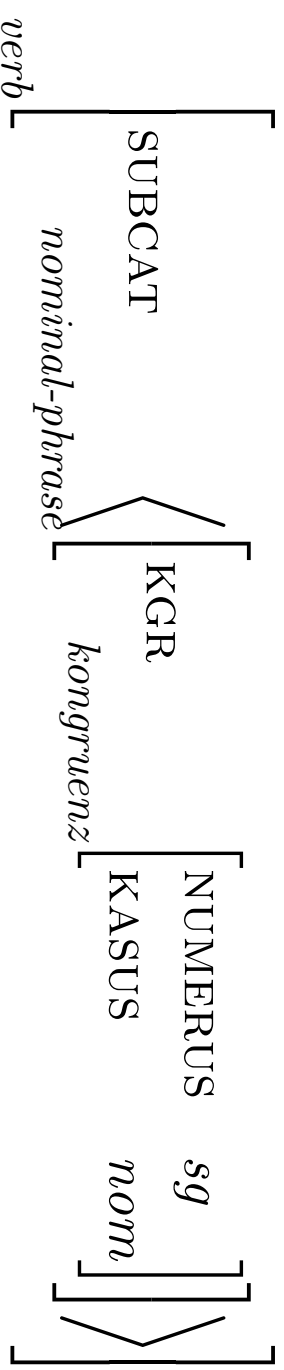
Prinzipien & Type-Constraints

- ▶ Prinzipien/Typ-Constraints stellen zusätzliche Anforderungen an zugelassene AWMen
- ▶ Sie sind von der Form $A \Rightarrow B$
- ▶ Bedeutung: Jede AWM, die von A subsumiert wird, muss auch von B subsumiert werden.
- ▶ Damit lassen sich zusätzliche Generalisierungen ausdrücken.

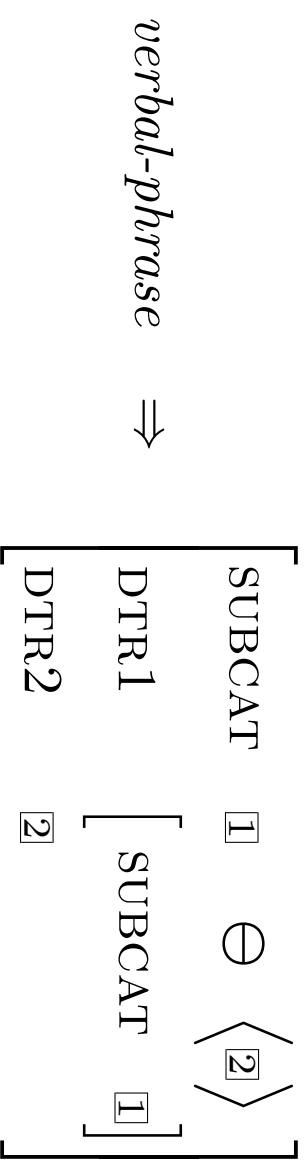
Beispiel: In einer NP kongruieren die beiden Töchter und vererben die Kongruenzmerkmale an die NP

$$nominal\text{-}phrase \Rightarrow \left[\begin{array}{cc} KGR & [1] \\ DTR1 & [KGR \quad [1]] \\ DTR2 & [KGR \quad [1]] \end{array} \right]$$

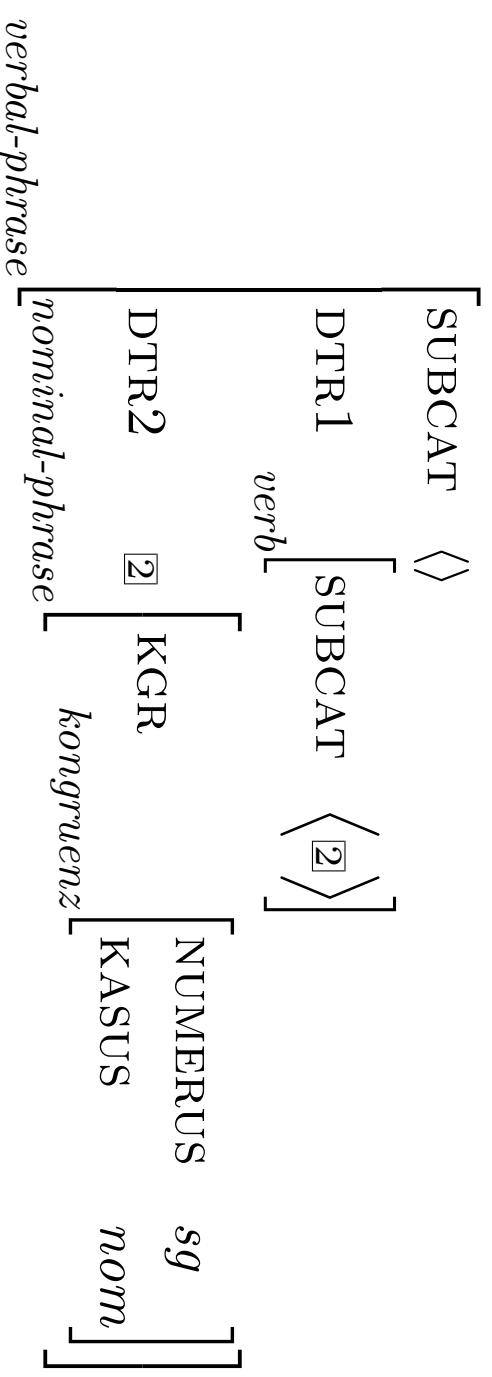
- ▶ **Erinnerung:** Komplementanforderung via SUBCAT-Liste
- ▶ **Beispiel:** Intransitives Verb schläft



- ▶ **Subkategorisierungsprinzip:** SUBCAT-Liste wird abgebaut



- ▶ Damit u.a. zugelassen:



- ▶ Prinzipien/Typ-Constraints sind zentraler Bestandteil von unifikationsbasierten Grammatiken
- ▶ Keine Regeln mehr, sondern nur noch zugelassene linguistische Zeichen → **sign-based Grammar**

Relationen

Eine **Relation** R über einer Menge D ist eine Menge von geordneten Paaren von Elementen aus D . R ist ...

- ▶ **reflexiv**: für **alle** $x \in D$: $\langle x, x \rangle \in R$
irreflexiv: für **kein** $x \in D$: $\langle x, x \rangle \in R$
- ▶ **symmetrisch**: $\langle x, y \rangle \in R$, dann auch $\langle y, x \rangle \in R$
asymmetrisch: $\langle x, y \rangle \in R$, dann **nicht** $\langle y, x \rangle \in R$
antisymmetrisch: $\langle x, y \rangle \in R$ und $\langle y, x \rangle \in R$, dann $x = y$
- ▶ **transitiv**: $\langle x, y \rangle \in R$ und $\langle y, z \rangle \in R$, dann auch $\langle x, z \rangle \in R$
(sonst einfach nicht transitiv)

- ▶ **total/konnex/linear**: für $x, y \in D$ gilt $\langle x, y \rangle \in R$ oder $\langle y, x \rangle \in R$
(sonst **partiell**)
- ▶ Relation reflexiv, antisymmetrisch und transitiv: (**schwache**) **Ordnung**
(Merke: 'kleiner-gleich' \leq bei Zahlen)
- ▶ Relation irreflexiv, asymmetrisch und transitiv: **strikte Ordnung**
(Merke: 'echt-kleiner' $<$ bei Zahlen)
- ▶ bei beiden noch zusätzlich: partiell vs. total)
- ▶ Relation reflexiv, symmetrisch und transitiv: **Äquivalenzrelation**
(Merke: 'gleich' = bei Zahlen)

Beispiele:

- Relation: $\langle x, y \rangle \in R$ gdw. x 'sitzt-in-einer-Reihe-vor' y .
irreflexiv, asymmetrisch, transitiv, partiell
 R ist partielle strikte Ordnung

Wie müssten Sie sitzen, damit die Relation total wird?

In einer Reihe hintereinander!

Beispiele:

- Relation: $\langle x, y \rangle \in R$ gdw. x 'schreibt-in-der-Klausur-ab-von' y .

irreflexiv, asymmetrisch, nicht-transitiv, partiell

R ist keine spezielle Relation

Wie müssten Sie voneinander abschreiben, damit die Relation eine strikte Ordnung wird?

Transitivität:

*Wenn A von B abschreibt und B von C abschreibt,
dann muss A auch von C abschreiben.*

Beispiele:

- Relation: $\langle x, y \rangle \in R$ gdw. x 'hat-die-gleiche-Klausernote-wie' y .
reflexiv, symmetrisch, transitiv
 R ist eine Äquivalenzrelation

Welche Eigenschaften ändern sich, wenn man stattdessen die Relation R'

x 'hat-die-gleiche-oder-eine-bessere-Klausernote-wie' y
betrachtet?

symmetrisch \leadsto antisymmetrisch

R' ist damit eine (schwache) Ordnung

Verbände

Verband als Ordnung

(ausgehend von Ordnung R)

- ▶ **Supremum** $\sup(x, y)$ zweier Elemente: *kleinste obere Schranke*
 $\langle x, c \rangle, \langle y, c \rangle \in R$ und für alle d mit $\langle x, d \rangle, \langle y, d \rangle \in R \Rightarrow \langle c, d \rangle \in R$
- ▶ **Infimum** $\inf(x, y)$ zweier Elemente: *größte untere Schranke*
 $\langle c, x \rangle, \langle c, y \rangle \in R$ und für alle d mit $\langle d, x \rangle, \langle d, y \rangle \in R \Rightarrow \langle d, c \rangle \in R$
- ▶ Eine Ordnung R ist ein Verband, wenn Supremum und Infimum für beliebige zwei Elemente x, y immer existieren.

Verbände

Verband als algebraische Struktur

Verband $\langle D, \vee, \wedge \rangle$ ist Menge D mit zwei Operationen **join** \vee und **meet** \wedge :

$$1 \text{ (a) } x \vee y = y \vee x$$

$$\text{(b) } x \wedge y = y \wedge x$$

Kommutativgesetze

$$2 \text{ (a) } x \vee (y \vee z) = (x \vee y) \vee z$$

$$\text{(b) } x \wedge (y \wedge z) = (x \wedge y) \wedge z$$

Assoziativgesetze

$$3 \text{ (a) } x \vee x = x$$

$$\text{(b) } x \wedge x = x$$

Idempotenzgesetze

$$4 \text{ (a) } x \vee (x \wedge y) = x$$

$$\text{(b) } x \wedge (x \vee y) = x$$

Absorptionsgesetze

Beispiele:

- Grundmenge: $\{a, b, c, d, e\}$

Relation: $R = \{ \langle a, a \rangle, \langle c, c \rangle, \langle d, d \rangle, \langle e, e \rangle, \langle a, b \rangle, \langle a, c \rangle, \langle c, d \rangle, \langle a, e \rangle, \langle b, e \rangle, \langle d, e \rangle \}$

weder reflexiv noch irreflexiv, antisymmetrisch, nicht transitiv

R ist keine spezielle Relation

Welche Paare muss man dazufügen, damit aus R eine Ordnung wird?

Reflexivität: $\langle b, b \rangle$

Transitivität: $\langle a, d \rangle, \langle c, e \rangle$

Ist R ein Verband?

Ja, denn für zwei Elemente gibt es immer das Supremum und Infimum

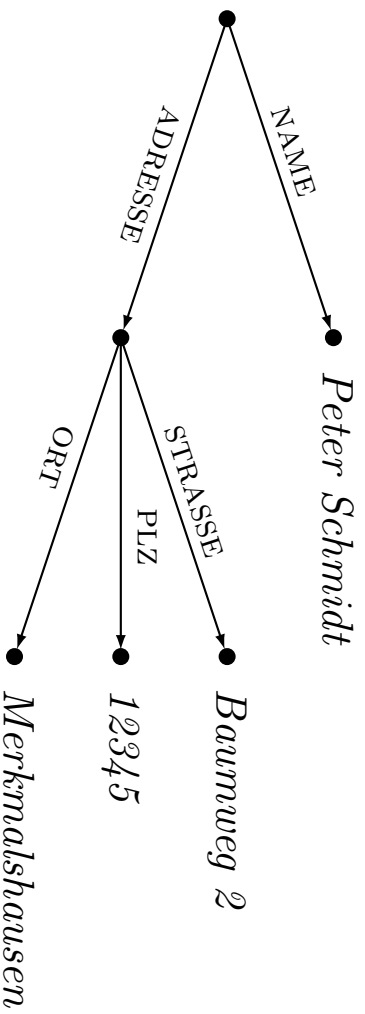
z.B. $\sup(b, c) = e$ $\inf(b, c) = a$.

Logische Formalisierung nach Johnson

- ▶ Beschreibung der Merkmalsstrukturen (d.h. der gerichteten azyklischen Graphen) mittels logischen Formeln.
- ▶ Objekte der Domäne: Knoten dieser Graphen
- ▶ Konstantensymbole \rightsquigarrow Knoten für atomare Werte
Variablen \rightsquigarrow Knoten von komplexen Werten/MS
Prädikatusymbole \rightsquigarrow Kanten

Beispiel:

| | | | | | | | |
|---------|---|---------|------------------|-----|--------------|-----|-----------------------|
| NAME | <i>Peter Schmidt</i> | | | | | | |
| ADRESSE | <table> <tr> <td>STRASSE</td><td><i>Baumweg 2</i></td></tr> <tr> <td>PLZ</td><td><i>12345</i></td></tr> <tr> <td>ORT</td><td><i>Merkmalshausen</i></td></tr> </table> | STRASSE | <i>Baumweg 2</i> | PLZ | <i>12345</i> | ORT | <i>Merkmalshausen</i> |
| STRASSE | <i>Baumweg 2</i> | | | | | | |
| PLZ | <i>12345</i> | | | | | | |
| ORT | <i>Merkmalshausen</i> | | | | | | |



$$\begin{aligned}
 & \exists x \exists y \left(\text{NAME}(x, \text{peter-schmidt}) \wedge \text{ADRESSE}(x, y) \right. \\
 & \quad \wedge \text{STRASSE}(y, \text{baumweg-2}) \wedge \text{PLZ}(y, 12345) \\
 & \quad \left. \wedge \text{ORT}(y, \text{merkmalshausen}) \right)
 \end{aligned}$$

- ▶ Einschränkung der möglichen Modelle mittels folgender Axiome:

- ▶ (AX1) Atomare Werte haben keine Attribute/ausgehenden Kanten:

z.B. $\forall x (\neg \text{ADRESSE}(\text{merkmalshausen}, x))$

- ▶ (AX2) Werte sind eindeutig/Kantenrelationen sind Funktionen

z.B. $\forall x \forall y \forall z ((\text{ADRESSE}(x, y) \wedge \text{ADRESSE}(x, z)) \rightarrow y = z)$

- ▶ (AX3) atomare Werte sind unterschiedlich/keine Namensgleichheit

z.B. $\neg(12345 = \text{merkmalshausen})$

- ▶ Unifikation zweier Beschreibungen $\exists x(\varphi)$ und $\exists x'(\varphi')$:
neue Beschreibung $\exists x \exists x' (\varphi \wedge \varphi') \wedge x = x'$
mittels Axiomen vereinfachen.

▶ d