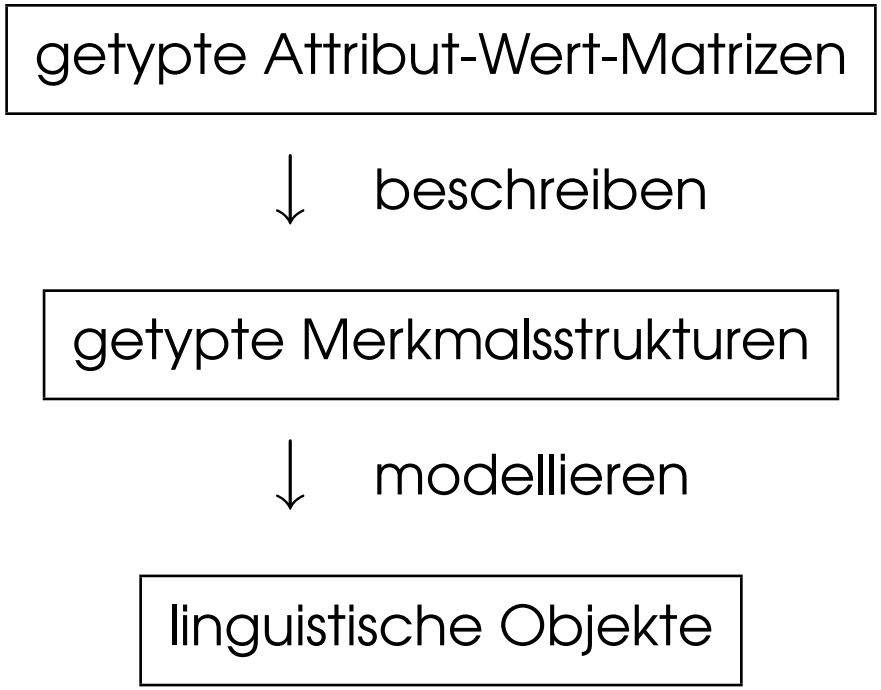
Typen/Sorten

- Bislang keine klare Strukturierung der Attribute und Werte
- abschreckendes Beispiel:

$$\begin{bmatrix} \textsc{genus} & akk \\ \textsc{kat} & sg \\ \textsc{numerus} & \begin{bmatrix} \textsc{genus} & NP \\ \textsc{kasus} & pl \end{bmatrix} \end{bmatrix}$$

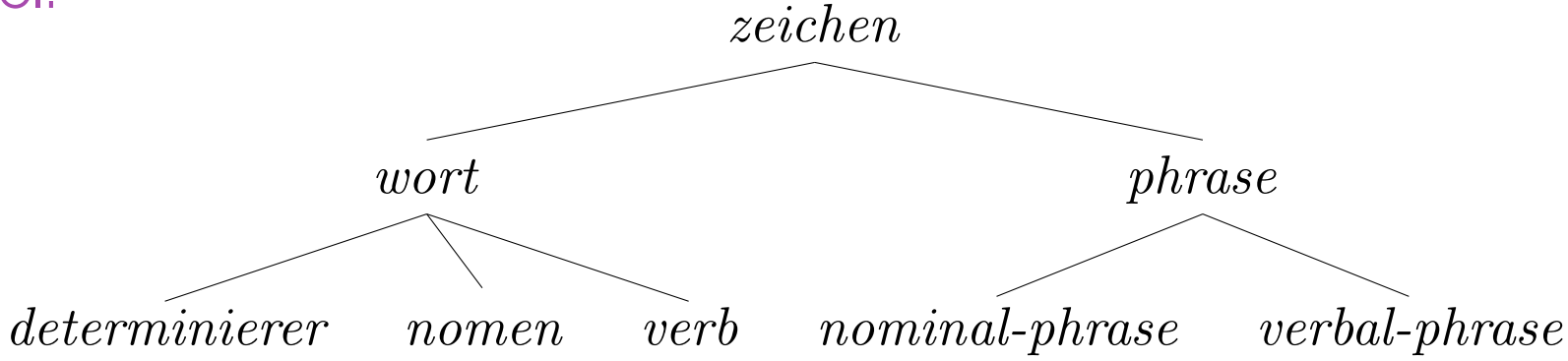
- Festlegung notwendig, welche Attribute welche Art von Wert haben
- Typen/Sorten stellen Klassen von Merkmalsstrukturen dar
- Legen damit genau fest, wie Merkmalsstrukturen aussehen können



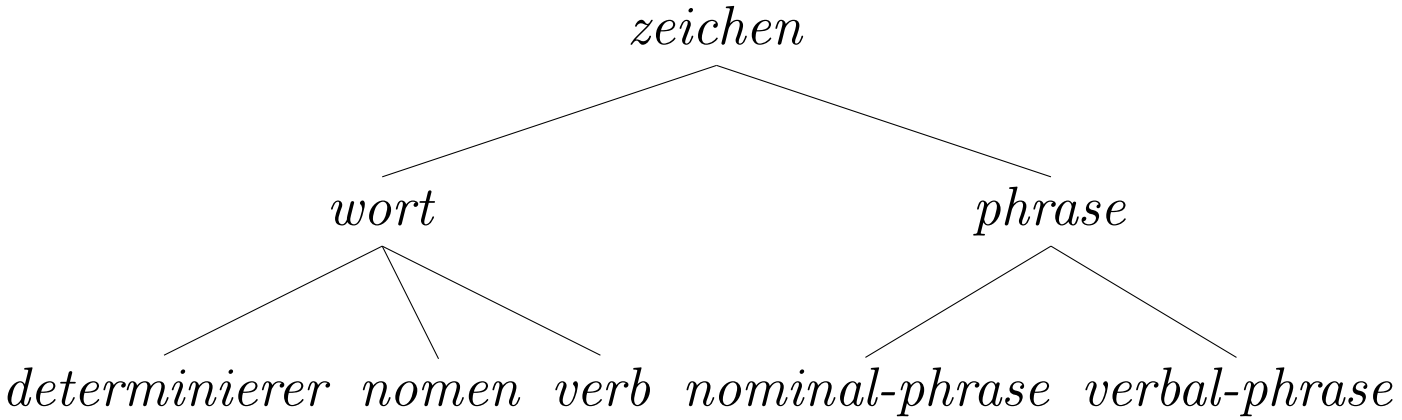
- Typen werden in zweierlei hinsicht verwendet:

  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.

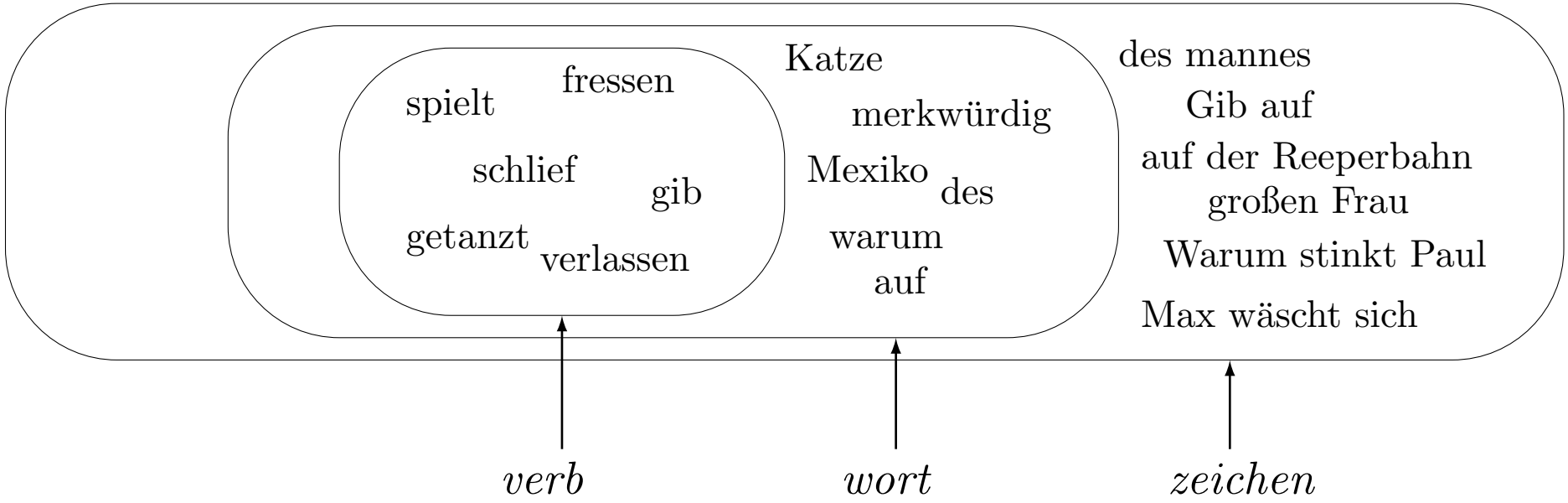


- Man spricht von Supertypen und Subtypen
- z.B. ist

  $wort$ ein Subtyp von $zeichen$

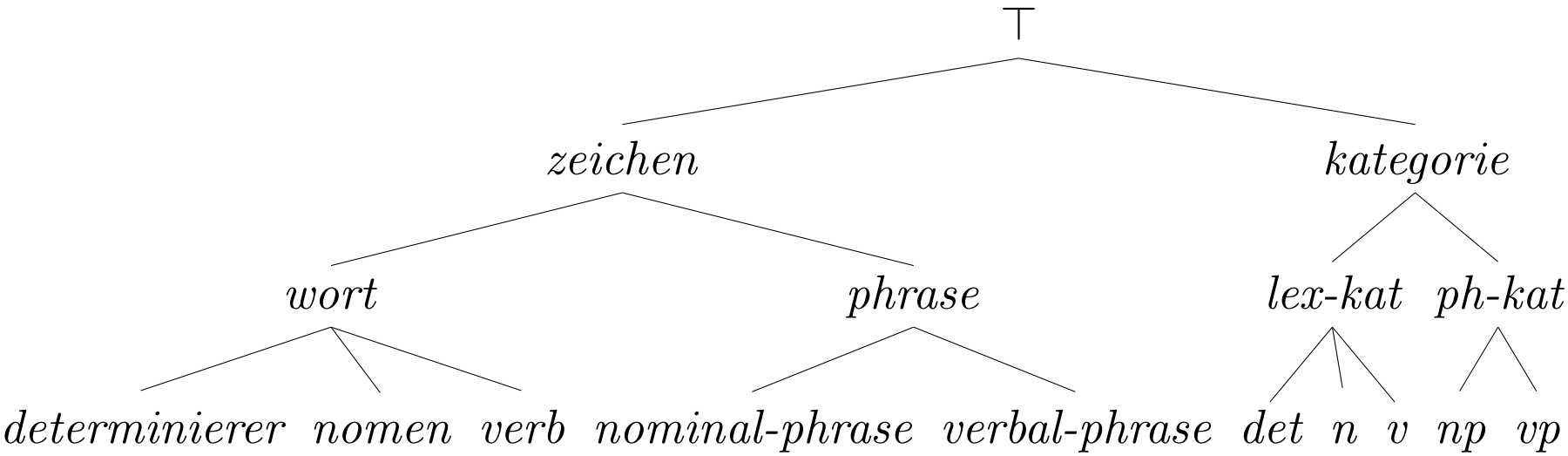
  $zeichen$ ein Supertyp von $wort$

  $nominal\text{-}phrase$ 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 $\top$
- also z.B. $kategorie \sqsubseteq np$, $lex\text{-}kat \sqsubseteq v$, 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 $\textsc{kat}$ mit einem Wert vom Typ $kategorie$.

  Angemessenheitsfunktion app (von engl. *appropriateness*)

$$\mathrm{app}(typ, \textsc{attribut}) = wert\text{-}typ$$

  also:

$$\mathrm{app}(zeichen, \textsc{kat}) = kategorie$$

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

  also: $$\mathrm{app}(zeichen, \textsc{kat}) = \downarrow$$

Angemessenheit von Attributen

- Zunächst soll $zeichen$ keine anderen Attribute haben (später: $\textsc{kgr}$)
- d.h. für andere Attribute ist app bei $zeichen$ undefiniert ($\uparrow$)

  also z.B.: $\mathrm{app}(zeichen, \textsc{genus}) = \uparrow$
- An der Definiertheit/Undefiniertheit der Angemessenheitsfunktion ist ablesbar, welche Attribute eine AWM haben kann.

Vererbung

- Erinnerung:

  Jedes $wort$ ist ein $zeichen$, d.h. $zeichen \sqsubseteq wort$

- $\Rightarrow$ weil $\textsc{kat}$ für $zeichen$ angemessen, deshalb auch für $wort$

- Formaler:

  weil $zeichen \sqsubseteq wort$ und $\mathrm{app}(zeichen, \textsc{kat}) =\downarrow$, deshalb $\mathrm{app}(wort, \textsc{kat}) =\downarrow$

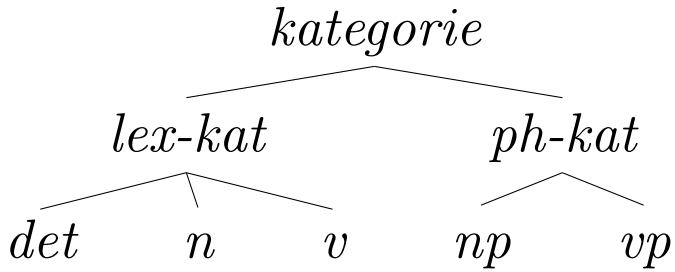
- Allgemein: Ein Subtyp erbt alle Merkmale seiner Supertypen

- d.h. wenn $t \sqsubseteq t'$ und $\mathrm{app}(t, A) =\downarrow$, dann auch $\mathrm{app}(t', A) =\downarrow$

- Damit im Beispiel: $zeichen$, $wort$, $phrase$, $determinierer$, $nomen$, $verb$, $nominal\text{-}phrase$, $verbal\text{-}phrase$ haben alle ein Merkmal $\textsc{kat}$

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 $lex\text{-}kat$ zulassen, und

  $phrase$ nur Werte vom Typ $ph\text{-}kat$
- Also: $\mathrm{app}(wort, \mathrm{KAT}) = lex\text{-}kat$ und $\mathrm{app}(phrase, \mathrm{KAT}) = ph\text{-}kat$
- Allgemein muss gelten:

  wenn $t \sqsubseteq t'$ und $\mathrm{app}(t, A) = w$, dann $\mathrm{app}(t', A) = w'$ wobei $w \sqsubseteq w'$

Vererbung

- Genauso: Ein $nomen$ sollte nur $\textsc{kat}$ Werte vom Typ n zulassen, eine $verbal\text{-}phrase$ nur Werte vom Typ vp, etc.

- Weiter Spezifizierung der vererbten Information:

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

Vererbung

- Weitere Spezifierung: In Phrasen gibt es Töchter.

  Jede Tochter ist vom Typ $zeichen$.

- Zwei weitere Attribute $\textsc{dtr1}$ und $\textsc{dtr2}$ sind für $phrase$ angemessen:

$$\begin{aligned} \mathrm{app}(phrase, \textsc{dtr1}) &= zeichen \\ \mathrm{app}(phrase, \textsc{dtr2}) &= zeichen \end{aligned}$$

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

Getypte AWMn

- In Attribut-Wert-Matrizen werden die Typen aussen an die Klammern geschrieben

$$zeichen\left[\text{KAT} \quad kategorie\right]$$

- Was vorher atomare Werte waren sind jetzt Typen!

- Eine AWM ist wohl-getypt, wenn sie die Angemessenheitsinformation wie folgt respektiert:

  Für jedes Attribut-Wert-Paar $\langle A, w\rangle$ in einer AWM vom Typ t muss gelten:

$$\text{app}(t, A) \sqsubseteq w$$

- Im Beispiel oben ist das erfüllt, denn

$$\text{app}(zeichen, \text{KAT}) = kategorie \quad \sqsubseteq kategorie$$

Damit sind auch wohl-getypt:

$$_{zeichen}\left[\textsc{kat} \quad ph\text{-}kat\right] \quad \textsf{denn} \quad \mathrm{app}(zeichen, \textsc{kat}) = kategorie \sqsubseteq ph\text{-}kat$$

$$_{zeichen}\left[\textsc{kat} \quad np\right] \quad \textsf{denn} \quad \mathrm{app}(zeichen, \textsc{kat}) = kategorie \sqsubseteq np$$

$$_{wort}\left[\textsc{kat} \quad v\right] \quad \textsf{denn} \quad \mathrm{app}(wort, \textsc{kat}) = lex\text{-}kat \sqsubseteq v$$

$$_{phrase}\begin{bmatrix}\textsc{kat} & vp \\ \textsc{dtr2} & zeichen\end{bmatrix} \quad \textsf{denn} \quad \begin{array}{l}\mathrm{app}(phrase, \textsc{kat}) = ph\text{-}kat \sqsubseteq vp \\ \mathrm{app}(phrase, \textsc{dtr2}) = zeichen \sqsubseteq zeichen\end{array}$$

$$_{verbal\text{-}phrase}\begin{bmatrix}\textsc{kat} & vp \\ \textsc{dtr1} & verb \\ \textsc{dtr2} & nominal\text{-}phrase\end{bmatrix} \quad \begin{array}{l}\mathrm{app}(verbal\text{-}phrase, \textsc{kat}) = vp \sqsubseteq vp \\ \mathrm{app}(verbal\text{-}phrase, \textsc{dtr1}) = zeichen \\ \qquad\qquad\qquad\qquad \sqsubseteq verb \\ \mathrm{app}(verbal\text{-}phrase, \textsc{dtr2}) = zeichen \\ \qquad\qquad\qquad\qquad \sqsubseteq nominal\text{-}phrase\end{array}$$

Auch wohl-getypt:

$$\begin{bmatrix} \textsc{kat} & vp \\ \textsc{dtr}1 & verb\begin{bmatrix} \textsc{kat} & v \end{bmatrix} \\ \textsc{dtr}2 & \begin{bmatrix} \textsc{kat} & np \\ \textsc{dtr}1 & determinierer\begin{bmatrix} \textsc{kat} & det \end{bmatrix} \\ \textsc{dtr}2 & nomen\begin{bmatrix} \textsc{kat} & n \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$verbal\text{-}phrase \qquad nominal\text{-}phrase$$

Aber nicht:

$$\mathit{zeichen}\left[\textsc{genus} \quad \top\right] \quad \lightning \quad \textsc{genus} \text{ Merkmal nicht angemessen für } \mathit{zeichen},$$

$$\mathrm{app}(\mathit{zeichen}, \textsc{genus}) = \uparrow$$

$$\mathit{wort}\left[\textsc{kat} \quad \mathit{np}\right] \quad \lightning \quad \mathrm{app}(\mathit{wort}, \textsc{kat}) = \mathit{lex\text{-}kat} \not\sqsubseteq \mathit{np}$$

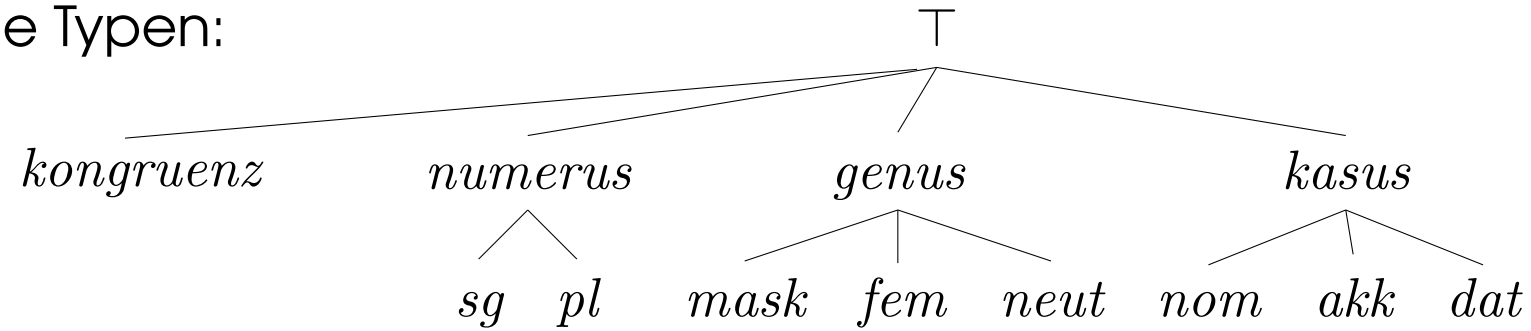
$$\mathit{nomen}\left[\textsc{kat} \quad \mathit{np}\right] \quad \lightning \quad \mathrm{app}(\mathit{nomen}, \textsc{kat}) = n \not\sqsubseteq \mathit{np}$$

Zusätzliche Restriktionen der Angemessenheit wären sinnvoll:

$$\mathit{verbal\text{-}phrase}\begin{bmatrix} \textsc{kat} & \mathit{vp} \\ \textsc{dtr}1 & \mathit{nomen} \\ \textsc{dtr}2 & \mathit{nomen} \end{bmatrix}$$

Erweiterung:

- zusätzliche Typen:



- Für den Typ $kongruenz$:

$$\begin{aligned}
&\mathrm{app}(kongruenz, \textsc{numerus}) = numerus\\
&\mathrm{app}(kongruenz, \textsc{genus}) = genus\\
&\mathrm{app}(kongruenz, \textsc{kasus}) = kasus\\
&\mathrm{app}(kongruenz, \textsc{kat}) = \uparrow\\
&\mathrm{app}(kongruenz, \textsc{kgr}) = \uparrow
\end{aligned}$$

- Für den Typ $zeichen$:

$$\begin{aligned}
&\mathrm{app}(zeichen, \textsc{kgr}) = kongruenz\\
&\mathrm{app}(kongruenz, \textsc{numerus}) = \uparrow\\
&\ldots
\end{aligned}$$

- Damit sind auch wohl-getypt:

$$zeichen\left[\begin{array}{ll}\textsc{kgr} & \\ & kongruenz\left[\begin{array}{ll}\textsc{numerus} & numerus \\ \textsc{kasus} & dat\end{array}\right]\end{array}\right]$$

$$\begin{aligned}\mathrm{app}(kongruenz, \textsc{numerus}) = numerus &\sqsubseteq numerus\\ \mathrm{app}(kongruenz, \textsc{kasus}) = kasus &\sqsubseteq dat\\ \mathrm{app}(zeichen, \textsc{kgr}) = kongruenz &\sqsubseteq kongruenz\end{aligned}$$

$$phrase\left[\begin{array}{ll}\textsc{kat} & vp \\ \textsc{kgr} & kongruenz\left[\begin{array}{ll}\textsc{numerus} & sg \\ \textsc{genus} & fem \\ \textsc{kasus} & dat\end{array}\right]\end{array}\right]$$

phrase erbt Angemessenheitsinfo über KGR von *zeichen*

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

  siehe $\mathrm{app}(kongruenz, \textsc{numerus}) = numerus$

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

- Falls doch alle angegeben sind, heisst die AWM vollständig

- Beispiele:

$$wort\left[\begin{array}{ll} \textsc{kat} & lex\text{-}kat \\ \textsc{kgr} & kongruenz\left[\begin{array}{ll} \textsc{numerus} & sg \\ \textsc{kasus} & dat \end{array}\right] \end{array}\right]$$

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

- $$phrase\left[\begin{array}{ll} \textsc{kgr} & kongruenz\left[\begin{array}{ll} \textsc{numerus} & sg \\ \textsc{kasus} & dat \\ \textsc{genus} & fem \end{array}\right] \\ \textsc{dtr}1 & zeichen\left[\begin{array}{ll} \textsc{kat} & lex\text{-}kat \\ \textsc{kgr} & kongruenz \end{array}\right] \end{array}\right]$$

  ist wohl-getypt, aber nicht vollständig (KAT und DTR2 sind auch angemessen für *phrase*)

- $$nomen\left[\begin{array}{ll} \textsc{kat} & lex\text{-}kat \\ \textsc{kgr} & kongruenz\left[\begin{array}{ll} \textsc{numerus} & numerus \\ \textsc{genus} & neut \\ \textsc{kasus} & kasus \end{array}\right] \end{array}\right]$$

  ist vollständig wohl-getypt.

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

- $$wort\left[\begin{array}{ll}\text{KAT} & lex\text{-}kat \\ \text{KGR} & kongruenz\left[\begin{array}{ll}\text{NUMERUS} & numerus \\ \text{GENUS} & neut \\ \text{KASUS} & kasus\end{array}\right]\end{array}\right]$$

  ist nicht typen-aufgelöst ($lex\text{-}kat$, $numerus$, $kasus$ sind nicht maximal)

- $$nomen\left[\begin{array}{ll}\text{KAT} & n \\ \text{KGR} & kongruenz\left[\begin{array}{ll}\text{NUMERUS} & pl \\ \text{GENUS} & neut \\ \text{KASUS} & dat\end{array}\right]\end{array}\right]$$

  ist typen-aufgelöst

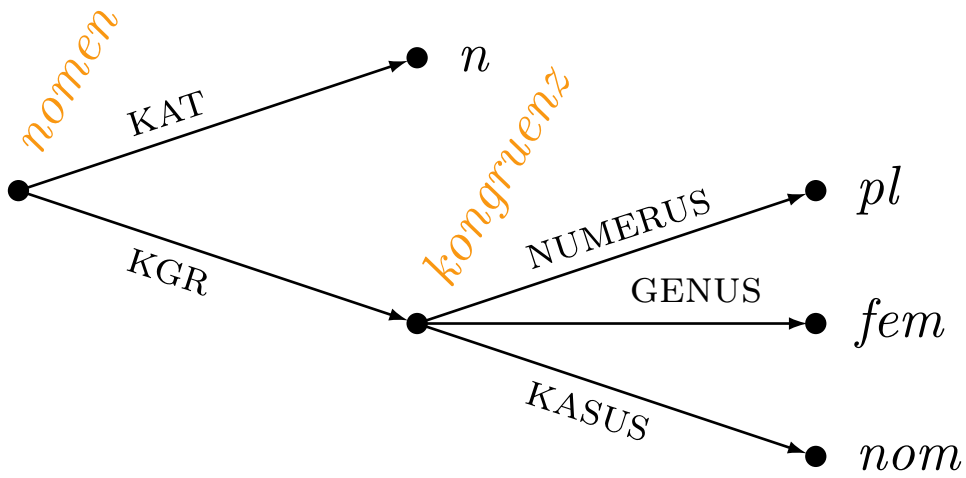
Zusammenfassung

- 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

Getypte Merkmalsstrukturen

- Erinnerung:
  Merkmalsstruktur $\equiv$ gerichteter azyklischer etikettierter Graph
- Kantenetikette $\equiv$ Attribute
- Alle getypten Merkmalsstrukturen sind
  - wohl-getypt
  - 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
- Typaufgelöstheit $\equiv$ alle Typen maximal

AWM → Merkmalsstruktur

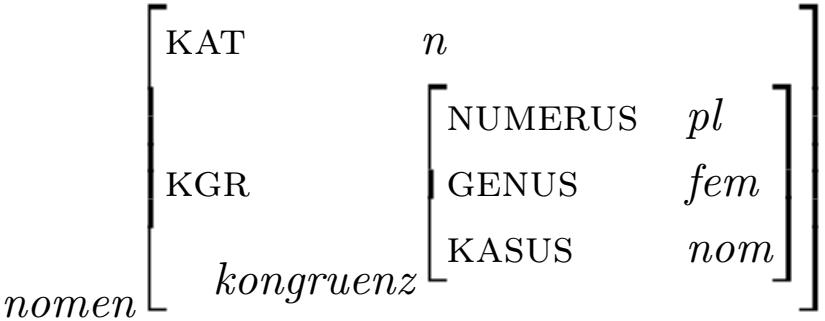
getypte Attribut-Wert-Matrizen

↓ beschreiben

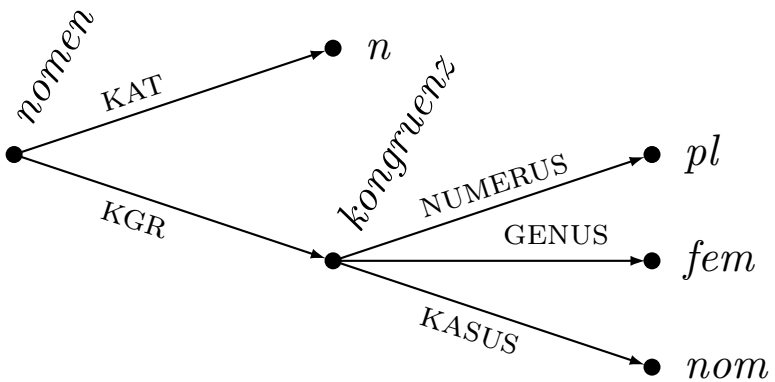
getypte Merkmalsstrukturen

↓ modellieren

linguistische Objekte



↓ beschreibt



↓ modelliert

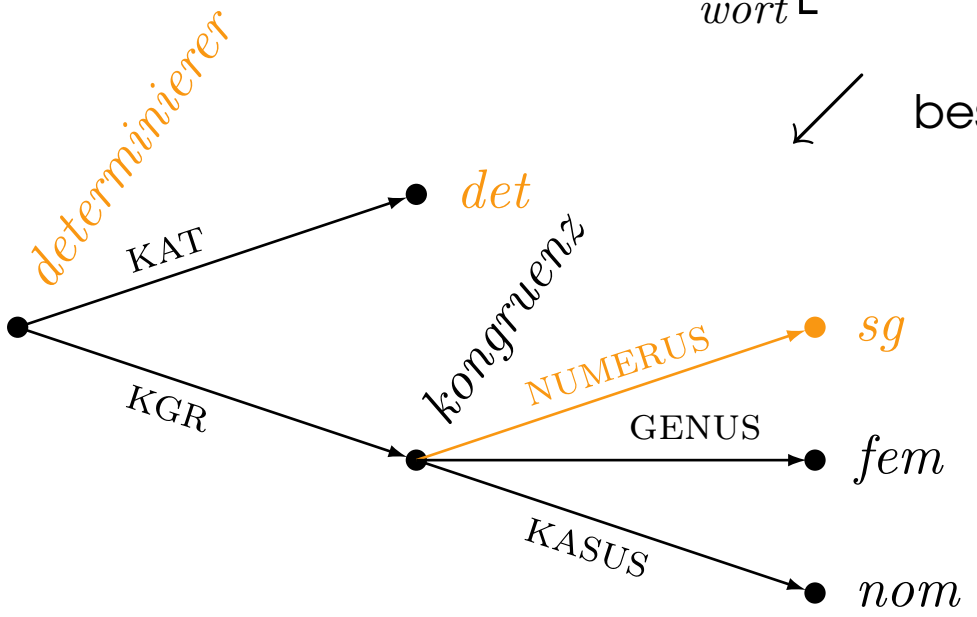
Frauen

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

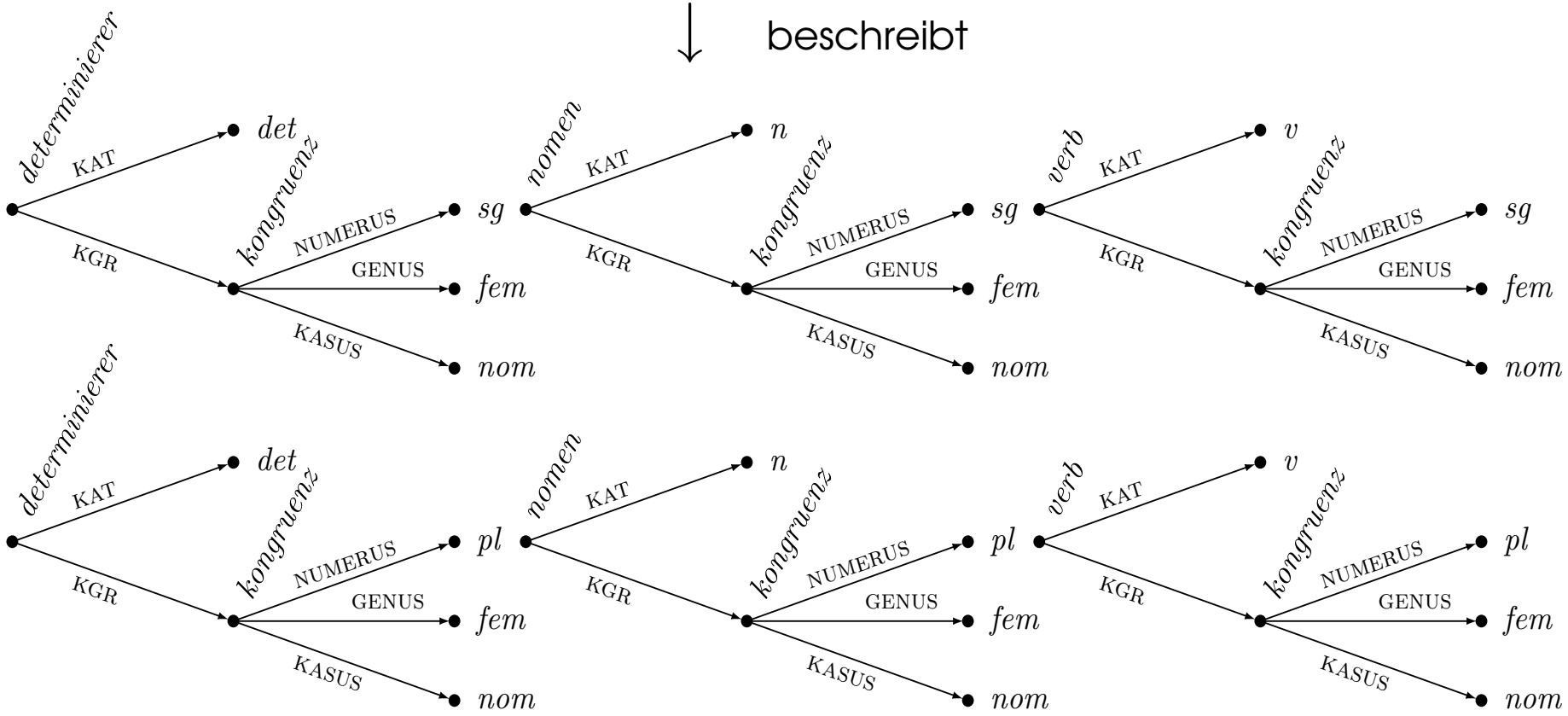
Eine unvollständige oder nicht-typenaufgelöste AWM beschreibt mehrere MSen (→ Unterspezifikation)

$$wort\left[\begin{array}{ll}\text{KAT} & lex\text{-}kat \\ \text{KGR} & kongruenz\left[\begin{array}{ll}\text{GENUS} & fem \\ \text{KASUS} & nom\end{array}\right]\end{array}\right]$$

beschreibt unter anderem



$$wort\left[\begin{array}{ll}\text{KAT} & lex\text{-}kat \\ \text{KGR} & kongruenz\left[\begin{array}{ll}\text{GENUS} & fem \\ \text{KASUS} & nom\end{array}\right]\end{array}\right]$$



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 AWMn 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:

$$\mathit{zeichen}\left[\text{KAT} \quad \mathit{ph\text{-}kat}\right] \sqsubseteq \mathit{phrase}\left[\text{KAT} \quad \mathit{np}\right] \qquad \begin{array}{r}\mathit{zeichen} \sqsubseteq \mathit{phrase}\\ \mathit{ph\text{-}kat} \sqsubseteq \mathit{np}\end{array}$$

$$\mathit{wort} \not\sqsubseteq \mathit{verbal\text{-}phrase}\left[\text{KAT} \quad \mathit{vp}\right] \qquad \mathit{wort} \not\sqsubseteq \mathit{verbal\text{-}phrase}$$

$$\mathit{phrase}\left[\text{KAT} \quad \mathit{vp}\right] \sqsubseteq \mathit{verbal\text{-}phrase}\left[\begin{array}{ll}\text{KAT} & \mathit{vp}\\ \text{KGR} & \mathit{kongruenz}\left[\text{NUMERUS} \quad \mathit{numerus}\right]\end{array}\right]$$

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

$$\mathit{fem} \not\sqsubseteq \mathit{genus}$$

Weitere Beispiele:

$$\textit{verbal-phrase}\left[\begin{array}{ll}\textsc{kat} & \textit{vp}\\ \textsc{dtr}1 & \textit{zeichen}\left[\begin{array}{lll}\textsc{kat} & \textit{v} & \\ \textsc{kgr} & \boxed{1} & \textit{kongruenz}\end{array}\right]\\ \textsc{dtr}2 & \textit{nominal-phrase}\left[\textsc{kgr}\quad \boxed{1}\right]\end{array}\right]$$

$$\sqsubseteq$$

$$\textit{verbal-phrase}\left[\begin{array}{ll}\textsc{kat} & \textit{vp}\\ \textsc{dtr}1 & \textit{verb}\left[\begin{array}{lll}\textsc{kat} & \textit{v} & \\ \textsc{kgr} & \boxed{1} & \textit{kongruenz}\left[\textsc{numerus}\quad \textit{sg}\right]\end{array}\right]\\ \textsc{dtr}2 & \textit{nominal-phrase}\left[\textsc{kgr}\quad \boxed{1}\right]\end{array}\right]$$

Weitere Beispiele:

$$\textit{verbal-phrase}\left[\begin{array}{ll}\textsc{kat} & \textit{vp}\\ \textsc{dtr1} & \textit{zeichen}\left[\begin{array}{ll}\textsc{kat} & \textit{v}\\ \textsc{kgr} & \textit{kongruenz}\end{array}\right]\\ \textsc{dtr2} & \textit{nominal-phrase}\left[\textsc{kgr}\ \textit{kongruenz}\left[\textsc{numerus}\ \textit{pl}\right]\right]\end{array}\right]$$

$$\not\sqsubseteq$$

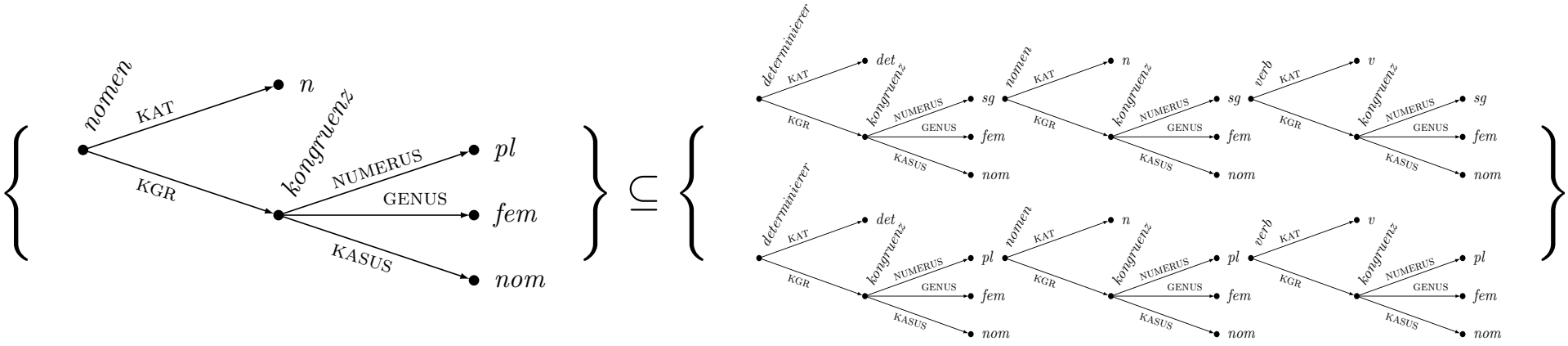
(weil $pl \not\sqsubseteq sg$)

$$\textit{verbal-phrase}\left[\begin{array}{ll}\textsc{kat} & \textit{vp}\\ \textsc{dtr1} & \textit{verb}\left[\begin{array}{ll}\textsc{kat} & \textit{v}\\ \textsc{kgr} & \boxed{1}\ \textit{kongruenz}\left[\textsc{numerus}\ \textit{sg}\right]\end{array}\right]\\ \textsc{dtr2} & \textit{nominal-phrase}\left[\textsc{kgr}\ \boxed{1}\right]\end{array}\right]$$

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

$$wort\begin{bmatrix}\text{KAT} & lex\text{-}kat \\ \text{KGR} & kongruenz\begin{bmatrix}\text{GENUS} & fem \\ \text{KASUS} & nom\end{bmatrix}\end{bmatrix} \sqsubseteq nomen\begin{bmatrix}\text{KAT} & n \\ \text{KGR} & kongruenz\begin{bmatrix}\text{NUMERUS} & pl \\ \text{GENUS} & fem \\ \text{KASUS} & nom\end{bmatrix}\end{bmatrix}$$

$\Leftrightarrow$

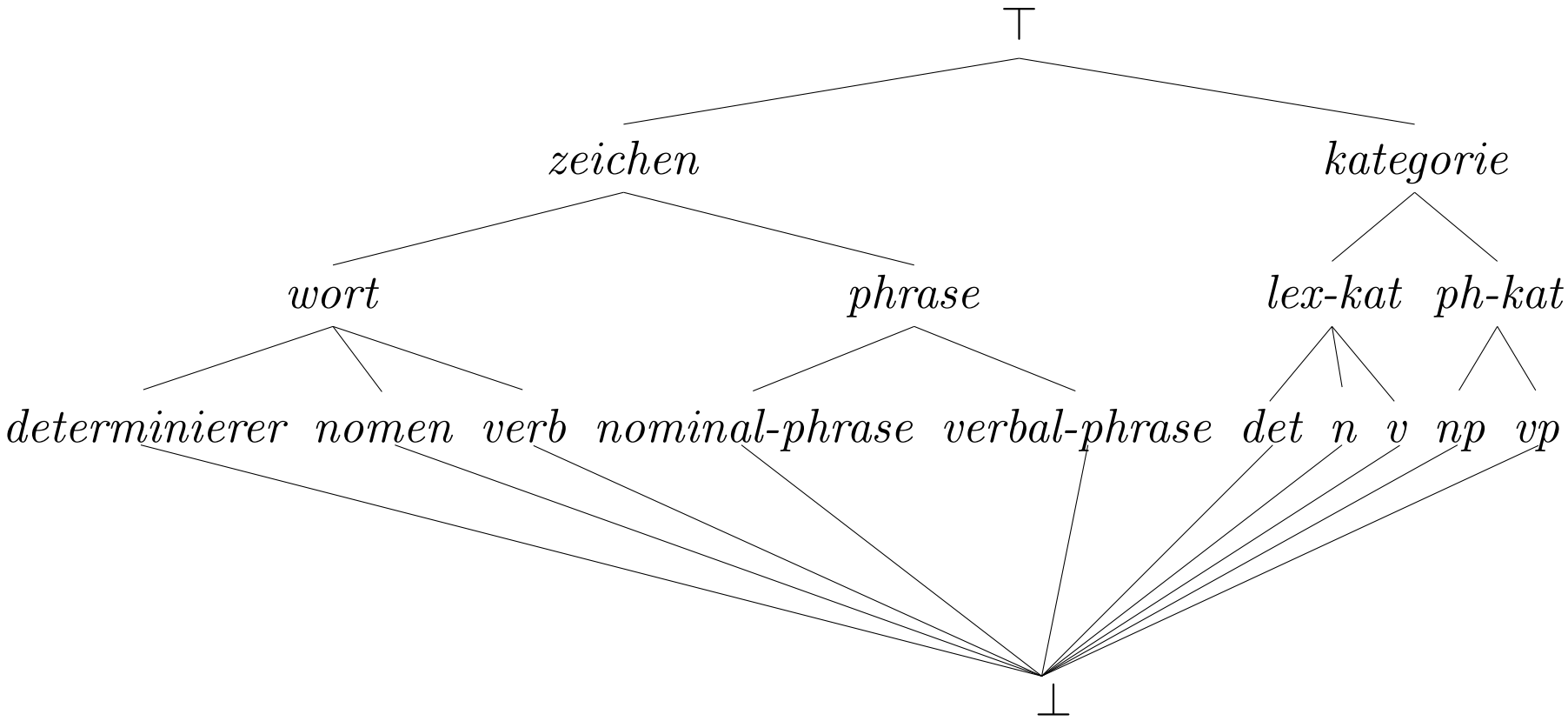


Unifikation getypter AWMen

- 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: $\bot$ steht für fehlschlagende Unifikation

Beispiele:

$$zeichen \sqcup wort = wort$$

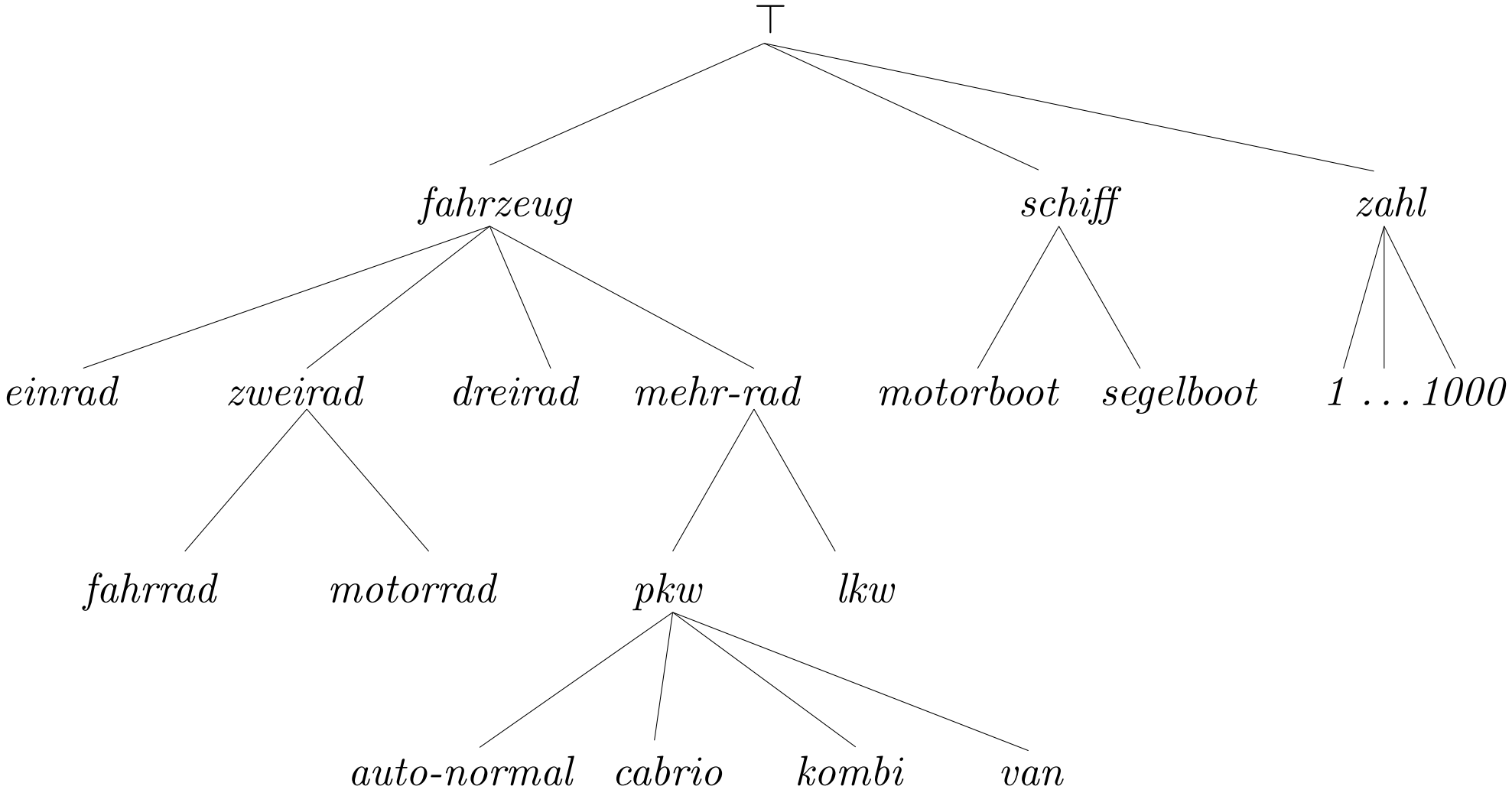
$$lex\text{-}kat \sqcup nomen = \bot$$

$$_{zeichen}\left[\text{KAT} \quad n\right] \sqcup {}_{wort}\left[\text{KAT} \quad lex\text{-}kat\right] = {}_{wort}\left[\text{KAT} \quad n\right]$$

$${}_{phrase}\left[\text{DTR1} \quad zeichen\right] \sqcup {}_{zeichen}\left[\text{KAT} \quad kategorie\right] = {}_{phrase}\begin{bmatrix}\text{KAT} & ph\text{-}kat \\ \text{DTR1} & zeichen\end{bmatrix}$$

$${}_{zeichen}\begin{bmatrix}\text{KGR} & {}_{kongruenz}\begin{bmatrix}\text{NUMERUS} & pl \\ \text{GENUS} & genus\end{bmatrix} \\ \text{DTR1} & nomen\end{bmatrix} \sqcup {}_{verbal\text{-}phrase}\begin{bmatrix}\text{KGR} & {}_{kongruenz}\begin{bmatrix}\text{NUMERUS} & numerus \\ \text{GENUS} & fem\end{bmatrix} \\ \text{DTR1} & {}_{zeichen}\left[\text{KAT} \quad v\right]\end{bmatrix} = \bot$$

Typhierarchie (diesmal für nicht-linguistische Objekte); $\bot$ nicht gezeigt



Angemessenheitsfunktion:

- Ein $fahrzeug$ hat eine gewisse Anzahl Räder

$$\mathrm{app}(fahrzeug, \textsc{räder}) \;=\; zahl$$

- Spezialisierung: Ein $einrad$ hat ein Rad

$$\mathrm{app}(einrad, \textsc{räder}) \;=\; 1$$

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

- Spezialisierung: Ein $zweirad$ hat zwei Räder

$$\mathrm{app}(zweirad, \textsc{räder}) \;=\; 2$$

- Spezialisierung: Ein $dreirad$ hat drei Räder

$$\mathrm{app}(dreirad, \textsc{räder}) \;=\; 3$$

Angemessenheitsfunktion:

- Spezialisierung: nicht für $mehr\text{-}rad$, aber: ein pkw hat vier Räder

$$\mathrm{app}(pkw, \textsc{räder}) = 4$$

- Einführung neuer Attribute: Ein $mehr\text{-}rad$ hat Türen

$$\mathrm{app}(mehr\text{-}rad, \textsc{türen}) = zahl$$

- Spezialisierung: Ein $cabrio$ hat 2 Türen

$$\mathrm{app}(cabrio, \textsc{türen}) = 2$$

- Spezialisierung: Ein $kombi$ hat 4 Türen

$$\mathrm{app}(kombi, \textsc{türen}) = 4$$

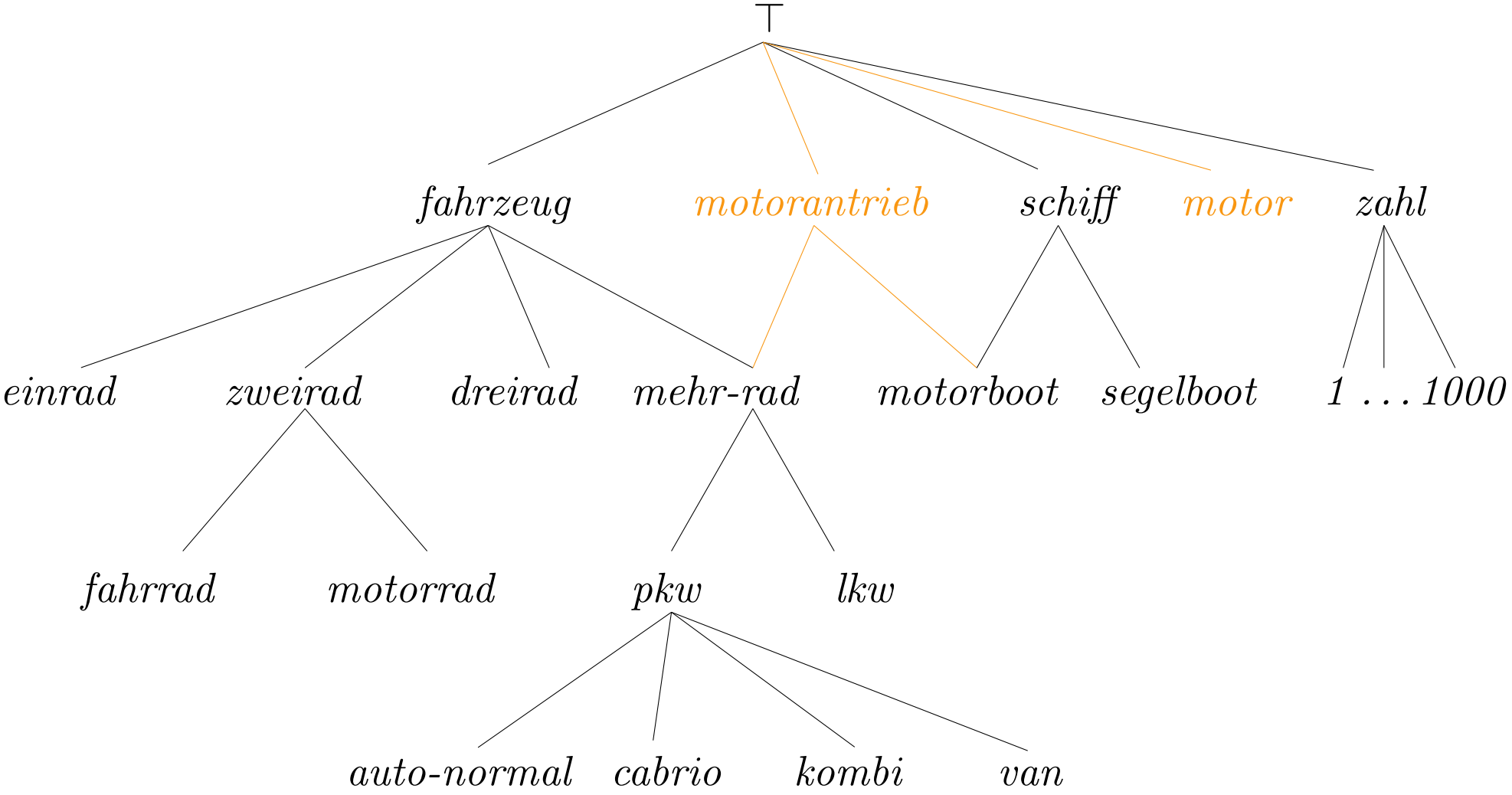
- Sonst für alle Typen t und Attribute A:

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

AWMn über dieser Typhierarchie

- $motorad\left[\begin{array}{ll}\text{RÄDER} & 2\end{array}\right]$ — wohl-getypt, vollständig, typen-aufgelöst
- $motorboot\left[\begin{array}{ll}\text{RÄDER} & zahl\end{array}\right]$ — nicht wohl-getypt ($\text{app}(motorboot, \text{RÄDER}) = \uparrow$)
- $fahrzeug\left[\begin{array}{ll}\text{RÄDER} & 4\\ \text{TÜREN} & 2\end{array}\right]$ — nicht wohl-getypt ($\text{app}(fahrzeug, \text{TÜREN}) = \uparrow$)
- $mehr\text{-}rad\left[\begin{array}{ll}\text{RÄDER} & 6\end{array}\right]$ — wohl-getypt, nicht vollständig (TÜREN), nicht typen-aufgelöst ($mehr\text{-}rad$)
- $kombi\left[\begin{array}{ll}\text{TÜREN} & 4\end{array}\right]$ — wohl-getypt, nicht vollständig (RÄDER), typen-aufgelöst
- $auto\text{-}normal\left[\begin{array}{ll}\text{RÄDER} & 4\\ \text{TÜREN} & zahl\end{array}\right]$ — wohl-getypt, vollständig, nicht typen-aufgelöst ($zahl$)

Erweiterung der Typhierarchie



Erweiterung der Angemessenheitsfunktion:

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

$$\mathrm{app}(motorantrieb, \textsc{motor}) \;=\; motor$$

- Ein $motor$ hat eine gewisse Anzahl Zylinder und PS

$$\mathrm{app}(motor, \textsc{zylinder}) \;=\; zahl$$

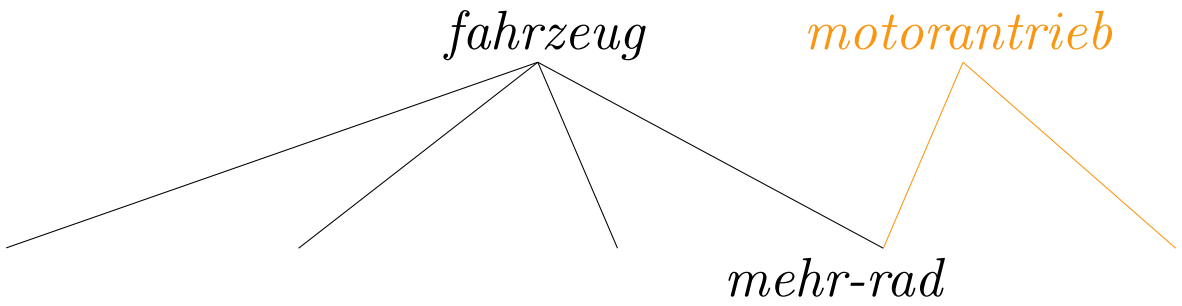
$$\mathrm{app}(motor, \textsc{ps}) \;=\; zahl$$

- wohl-getypte und vollständige AWM z.B.

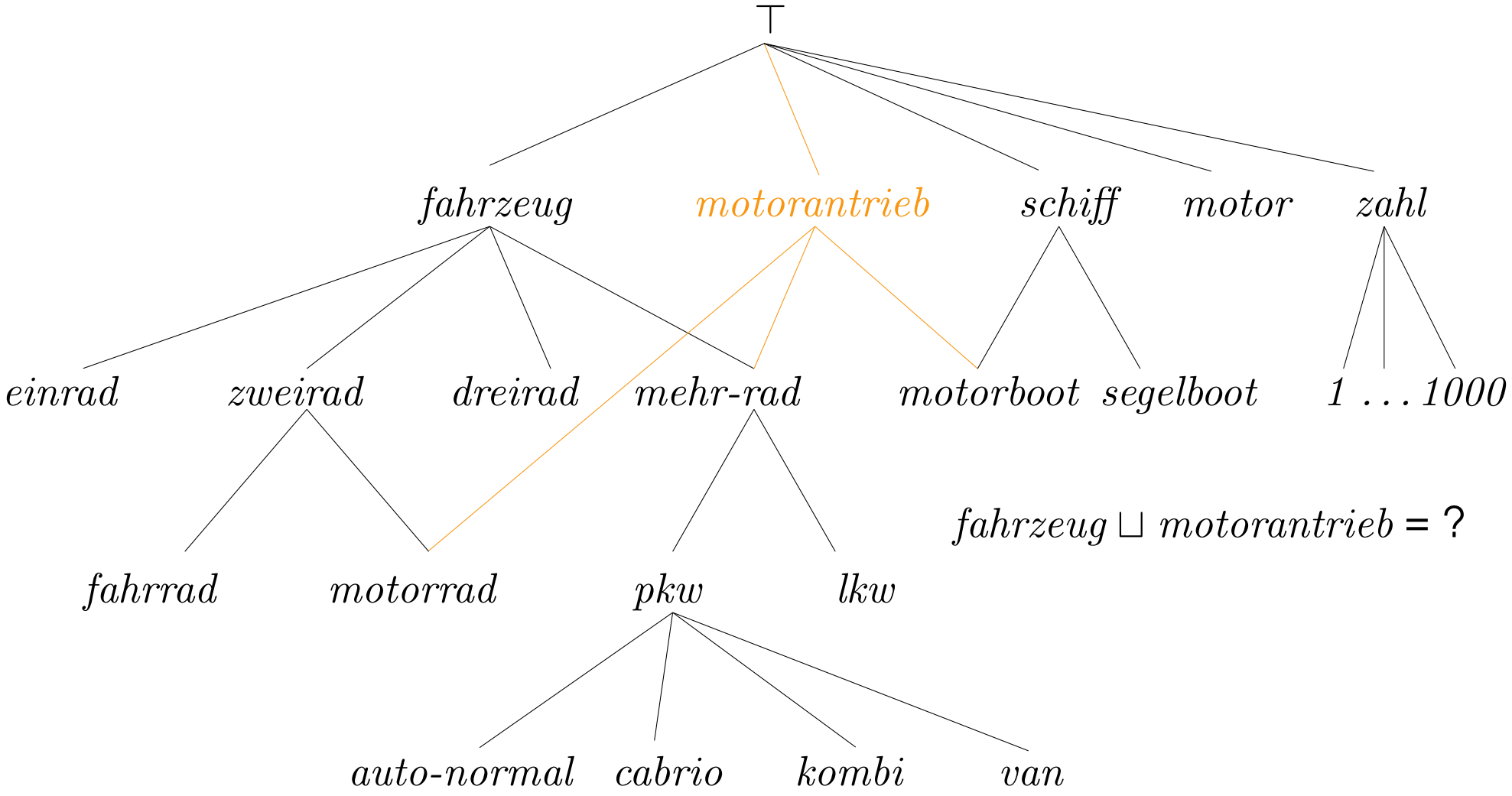
$$motorantrieb\left[\begin{array}{ll} \textsc{motor} & motor\left[\begin{array}{ll} \textsc{zylinder} & 4 \\ \textsc{ps} & 150 \end{array}\right] \end{array}\right]$$

- $motorboot$ hat zwei (unmittelbare) Supertypen: $schiff$ und $motorantrieb$
- $mehr\text{-}rad$ 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:

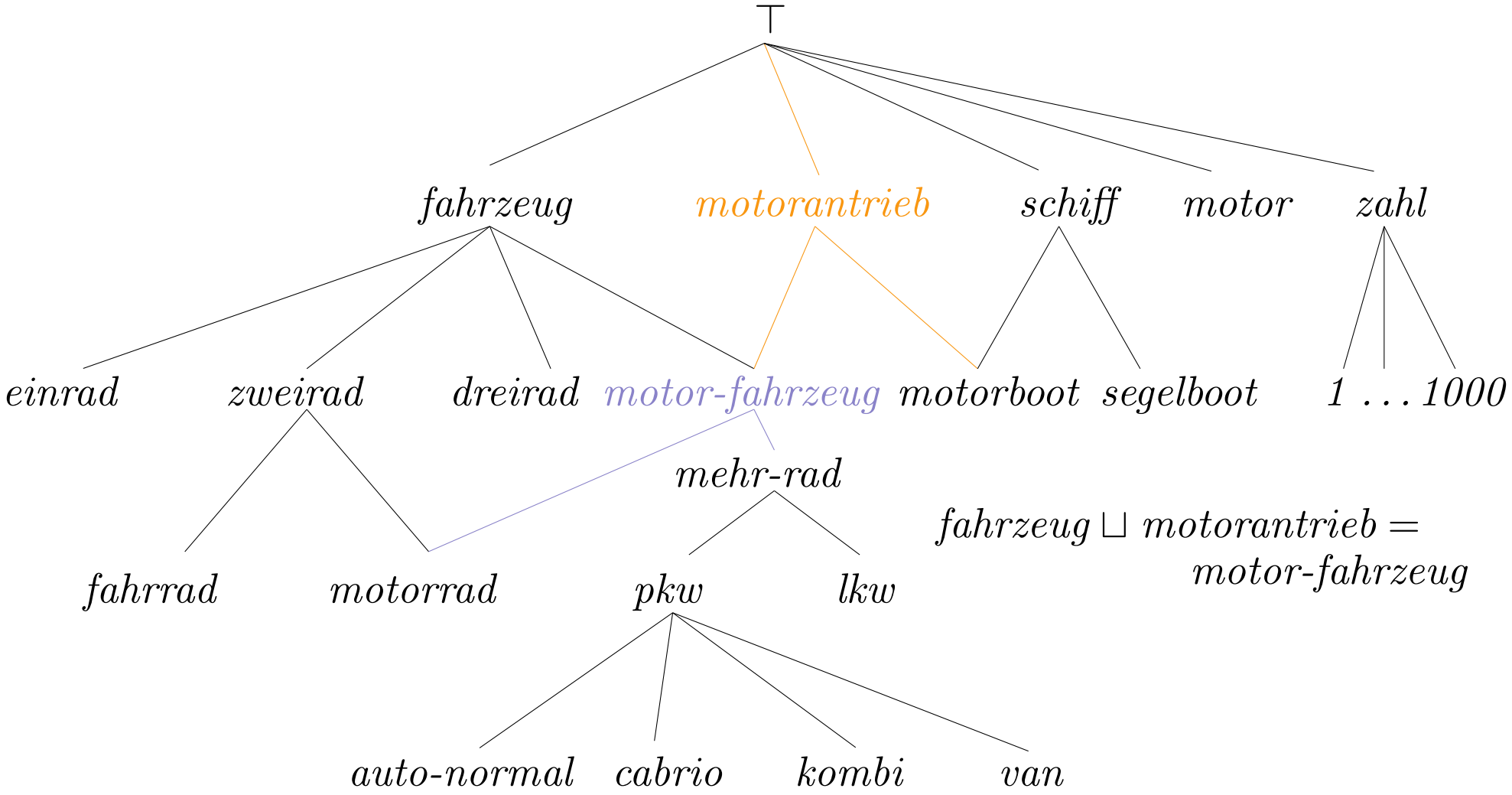
$$fahrzeug \sqcup motorantrieb = mehr\text{-}rad$$



Beispiel einer unerlaubten Typhierarchie:



Abhilfe: Einführung von Hilfstypen:



- 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 $motor\text{-}fahrzeug$ erbt von $fahrzeug$ und von $motorantrieb$:

  von $fahrzeug$ $\quad \mathrm{app}(fahrzeug, \textsc{räder}) = zahl$

  von $motorantrieb$ $\quad \mathrm{app}(motorantrieb, \textsc{motor}) = motor$

- Damit ist z.B. $motor\text{-}fahrzeug\begin{bmatrix}\textsc{räder} & 4 \\ \textsc{motor} & motor\begin{bmatrix}\textsc{zylinder} & 4 \\ \textsc{ps} & 120\end{bmatrix}\end{bmatrix}$ wohlgetypt und vollständig

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

- $$mehr\text{-}rad\begin{bmatrix}\text{RÄDER} & zahl \\ \text{MOTOR} & motor\begin{bmatrix}\text{ZYLINDER} & 9877 \\ \text{PS} & zahl\end{bmatrix}\end{bmatrix}$$ **+W -V -T**

- $$motorrad\begin{bmatrix}\text{RÄDER} & zahl \\ \text{MOTOR} & motor\begin{bmatrix}\text{ZYLINDER} & 9877 \\ \text{PS} & zahl\end{bmatrix}\end{bmatrix}$$ **-W** $\text{app}(zweirad, \text{RÄDER}) = 2$

- $$kombi\begin{bmatrix}\text{RÄDER} & 4 \\ \text{TÜREN} & 4 \\ \text{MOTOR} & motor\begin{bmatrix}\text{ZYLINDER} & 24 \\ \text{PS} & 170\end{bmatrix}\end{bmatrix}$$ **+W +V +T**

Beispiele (Subsumtion):

- $motorantrieb\left[\begin{array}{l}\textsc{motor}\quad motor\left[\begin{array}{ll}\textsc{zylinder} & 8\\ \textsc{ps} & zahl\end{array}\right]\end{array}\right] \not\sqsubseteq pkw\left[\begin{array}{ll}\textsc{räder} & 4\\ \textsc{motor}\quad motor\left[\begin{array}{ll}\textsc{zylinder} & 16\\ \textsc{ps} & 200\end{array}\right] & \end{array}\right]$
- $motorantrieb \not\sqsubseteq zweirad\left[\textsc{räder}\quad 2\right]$
- $motor\text{-}fahrzeug\left[\begin{array}{l}\textsc{motor}\quad motor\left[\textsc{ps}\quad zahl\right]\end{array}\right] \not\sqsubseteq motorboot\left[\begin{array}{l}\textsc{motor}\quad motor\left[\begin{array}{ll}\textsc{zylinder} & 16\\ \textsc{ps} & 250\end{array}\right]\end{array}\right]$
- $fahrzeug\left[\textsc{räder}\quad 2\right] \not\sqsubseteq pkw\left[\begin{array}{ll}\textsc{räder} & 4\\ \textsc{motor}\quad motor\left[\begin{array}{ll}\textsc{zylinder} & 16\\ \textsc{ps} & 250\end{array}\right] & \end{array}\right]$

Beispiele (Unifikation):

- $motorantrieb \sqcup {}_{zweirad}\left[\text{RÄDER} \quad 2\right] = {}_{motorrad}\left[\text{RÄDER} \quad 2\right]$
- ${}_{motorantrieb}\left[\text{MOTOR} \; {}_{motor}\left[\text{PS} \quad 340\right]\right] \sqcup {}_{lkw}\left[\begin{array}{ll}\text{TÜREN} & 2 \\ \text{MOTOR} & {}_{motor}\left[\text{PS} \quad 355\right]\end{array}\right] = \bot$
- ${}_{fahrzeug}\left[\text{RÄDER} \quad 8\right] \sqcup {}_{motorantrieb}\left[\text{MOTOR} \; {}_{motor}\left[\begin{array}{ll}\text{ZYLINDER} & 8 \\ \text{PS} & zahl\end{array}\right]\right]$

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

Erweiterungen: Disjunktion

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

$$lkw\left[\begin{array}{ll}\text{RÄDER} & 6 \vee 8 \\ \text{MOTOR} & motor\left[\text{ZYLINDER} \quad 12 \vee 16\right]\end{array}\right]$$

steht für

$$lkw\left[\begin{array}{ll}\text{RÄDER} & 6 \\ \text{MOTOR} & motor\left[\text{ZYLINDER} \quad 12\right]\end{array}\right] \qquad lkw\left[\begin{array}{ll}\text{RÄDER} & 6 \\ \text{MOTOR} & motor\left[\text{ZYLINDER} \quad 16\right]\end{array}\right]$$

$$lkw\left[\begin{array}{ll}\text{RÄDER} & 8 \\ \text{MOTOR} & motor\left[\text{ZYLINDER} \quad 12\right]\end{array}\right] \qquad lkw\left[\begin{array}{ll}\text{RÄDER} & 8 \\ \text{MOTOR} & motor\left[\text{ZYLINDER} \quad 16\right]\end{array}\right]$$

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):

$$\mathit{determinierer}\left[\begin{array}{lll}\textsc{kat} & \mathit{det} & \\ \textsc{kgr} & \mathit{kongruenz}\left[\begin{array}{ll}\textsc{numerus} & \mathit{sg} \\ \textsc{genus} & \neg\ \mathit{fem} \\ \textsc{kasus} & \mathit{dat}\end{array}\right] & \end{array}\right]$$

Erweiterungen: Negation

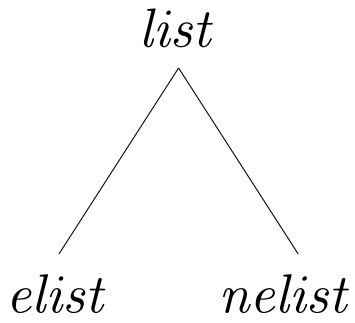
- Weil $\mathrm{app}(kongruenz, \textsc{genus}) = genus \quad \ldots$
- . . . und $genus$ die Subtypen $mask,\ fem,\ neut$ hat. . .
- . . . steht diese AWM für

$$determinierer\left[\begin{array}{lll} \textsc{kat} & det & \\ \textsc{kgr} & kongruenz\left[\begin{array}{lll} \textsc{numerus} & sg & \\ \textsc{genus} & mask \vee & neut \\ \textsc{kasus} & dat & \end{array}\right] & \end{array}\right]$$

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)

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

- Anstelle von $\top$ kann auch ein anderer Typ stehen,

  z.B. Liste von Zahlen $\rightsquigarrow zahl$, Liste von Fahrzeugen $\rightsquigarrow fahrzeug$, etc.

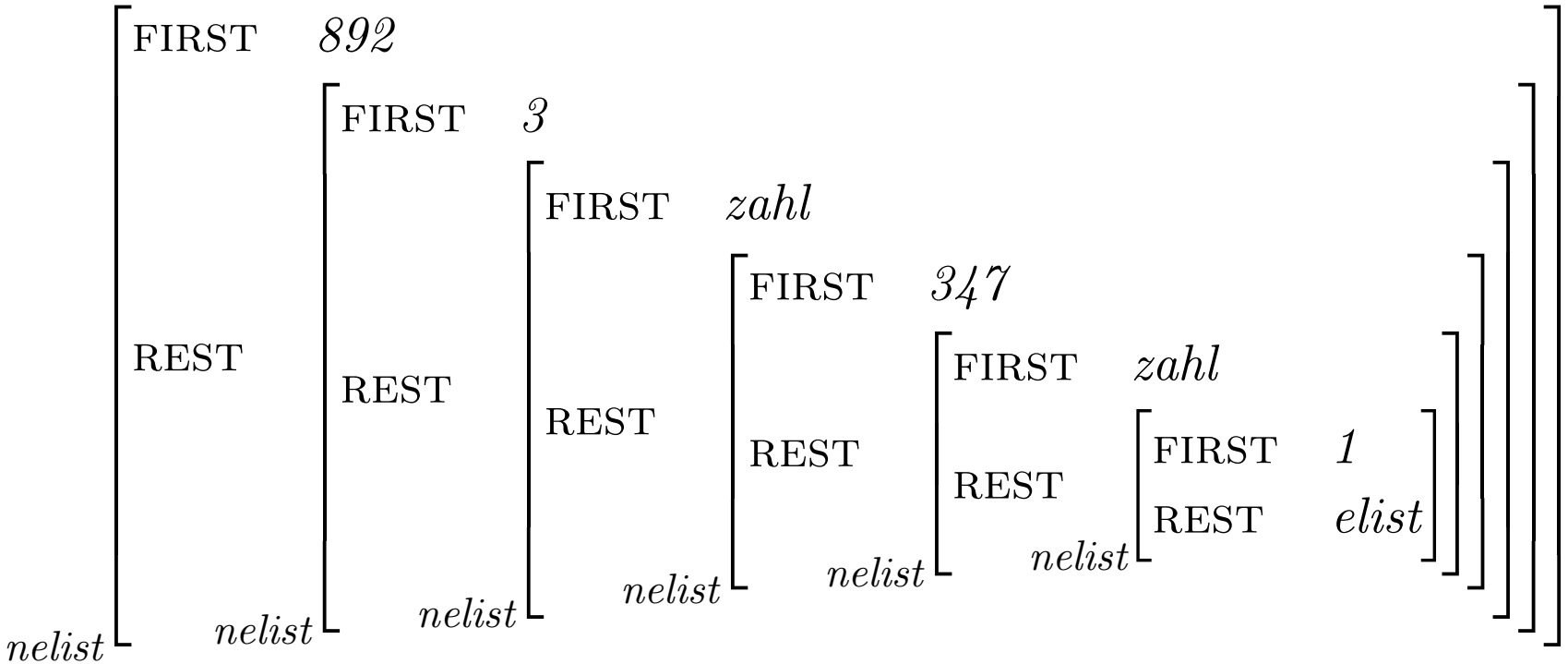
Beispiel: (mit Typenhierarchie von vorher)

$$\left\langle motorboot, 4, \top, \begin{bmatrix} \text{RÄDER} & zahl \\ \text{MOTOR}_{motor} & \begin{bmatrix} \text{PS} & 45 \end{bmatrix} \end{bmatrix}_{pkw}, schiff \right\rangle$$

$$\begin{bmatrix} \text{FIRST} & motorboot \\ \text{REST} & \begin{bmatrix} \text{FIRST} & 4 \\ \text{REST} & \begin{bmatrix} \text{FIRST} & \top \\ \text{REST} & \begin{bmatrix} \text{FIRST} & \begin{bmatrix} \text{RÄDER} & zahl \\ \text{MOTOR}_{motor} & \begin{bmatrix} \text{PS} & 45 \end{bmatrix} \end{bmatrix}_{pkw} \\ \text{REST} & \begin{bmatrix} \text{FIRST} & schiff \\ \text{REST} & elist \end{bmatrix}_{nelist} \end{bmatrix}_{nelist} \end{bmatrix}_{nelist} \end{bmatrix}_{nelist} \end{bmatrix}_{nelist}$$

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

$$\langle 892,\ 3,\ zahl,\ 347,\ zahl,\ 1 \rangle$$



Prinzipien & Type-Constraints

- Prinzipien/Typ-Constraints stellen zusätzliche Anforderungen an zugelassene AWMen
- Sie sind von der Form $A \quad \Rightarrow \quad B$
- 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

$$\mathit{nominal\text{-}phrase} \quad \Rightarrow \quad \begin{bmatrix} \textsc{kgr} & \boxed{1} \\ \textsc{dtr}1 & \begin{bmatrix} \textsc{kgr} & \boxed{1} \end{bmatrix} \\ \textsc{dtr}2 & \begin{bmatrix} \textsc{kgr} & \boxed{1} \end{bmatrix} \end{bmatrix}$$

- Wie oben werden oft die Typen weggelassen, wenn sie klar sind.
- Damit ist z.B. folgende AVM (z.B. für dem Hund) zugelassen:

$$
\begin{bmatrix}
\textsc{kat} & np \\
\textsc{kgr} & \boxed{1} \\
\textsc{dtr}1 & \begin{bmatrix} \textsc{kat} & det \\ \textsc{kgr} & \boxed{1} \end{bmatrix}_{determinierer} \\
\textsc{dtr}2 & \begin{bmatrix} \textsc{kat} & n \\ \textsc{kgr} & \boxed{1} \end{bmatrix}_{kongruenz} \begin{bmatrix} \textsc{numerus} & sg \\ \textsc{genus} & mask \\ \textsc{kasus} & dat \end{bmatrix}_{nomen}
\end{bmatrix}_{nominal\text{-}phrase}
$$

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

$$\textit{verb}\left[\begin{array}{l}\textsc{subcat}\ \left\langle\left[\begin{array}{l}\textsc{kgr}\ \left[\begin{array}{ll}\textsc{numerus} & \textit{sg}\\ \textsc{kasus} & \textit{nom}\end{array}\right]\\ \textit{kongruenz}\end{array}\right]\right\rangle\\ \textit{nominal-phrase}\end{array}\right]$$

- Subkategorisierungsprinzip: SUBCAT-Liste wird abgebaut

$$\textit{verbal-phrase} \Rightarrow \left[\begin{array}{ll}\textsc{subcat} & \boxed{1} \ominus \left\langle \boxed{2} \right\rangle\\ \textsc{dtr}1 & \left[\textsc{subcat}\ \boxed{1}\right]\\ \textsc{dtr}2 & \boxed{2}\end{array}\right]$$

- Damit u.a. zugelassen:

$$\textit{verbal-phrase}\left[\begin{array}{ll}\textsc{subcat} & \langle\rangle \\ \textsc{dtr}1 & \textit{verb}\left[\textsc{subcat} \quad \left\langle \boxed{2} \right\rangle\right] \\ \textsc{dtr}2 & \boxed{2}\left[\textsc{kgr} \quad \textit{kongruenz}\left[\begin{array}{ll}\textsc{numerus} & \textit{sg} \\ \textsc{kasus} & \textit{nom}\end{array}\right]\right] \\ \textit{nominal-phrase} & \end{array}\right]$$

- 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 änderen 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 (a) $x \vee y = y \vee x$
(b) $x \wedge y = y \wedge x$ — Kommutativgesetze

2 (a) $x \vee (y \vee z) = (x \vee y) \vee z$
(b) $x \wedge (y \wedge z) = (x \wedge y) \wedge z$ — Assoziativgesetze

3 (a) $x \vee x = x$
(b) $x \wedge x = x$ — Idempotenzgesetze

4 (a) $x \vee (x \wedge y) = x$
(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?

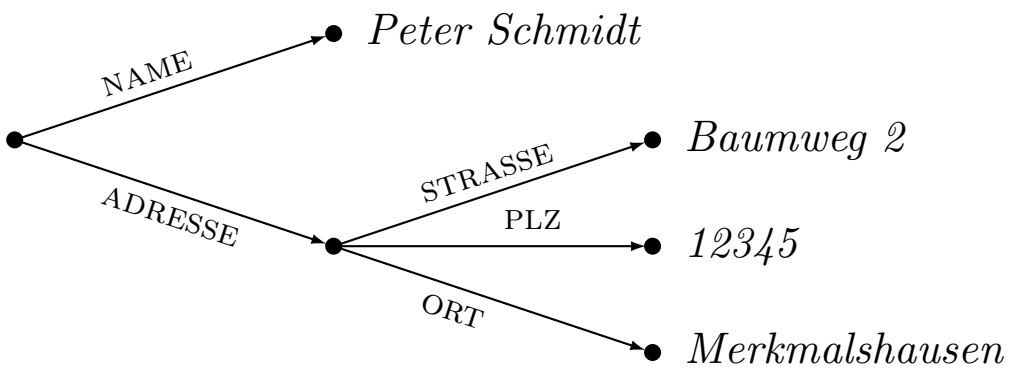
  Ja, denn für zwei Elemente gibt es immer das Supremum und Infimum
  z.B. $\sup(b, c) = e \qquad \inf(b, c) = a.$

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
  Prädikatensymbole $\rightsquigarrow$ Kanten

Beispiel:

$$\begin{bmatrix} \textsc{name} & \textit{Peter Schmidt} \\ \textsc{adresse} & \begin{bmatrix} \textsc{strasse} & \textit{Baumweg 2} \\ \textsc{plz} & \textit{12345} \\ \textsc{ort} & \textit{Merkmalshausen} \end{bmatrix} \end{bmatrix}$$



$$\exists x \exists y \big(\textsc{name}(x, \textit{peter\_schmidt}) \wedge \textsc{adresse}(x, y) \wedge \textsc{strasse}(y, \textit{baumweg\_2}) \wedge \textsc{plz}(y, \textit{12345}) \wedge \textsc{ort}(y, \textit{merkmalshausen}) \big)$$

- Einschränkung der möglichen Modelle mittels folgender Axiome:
- (AX1) Atomare Werte haben keine Attribute/ausgehenden Kanten:

  z.B. $\forall x(\neg \textsc{adresse}(merkmalshausen, x))$
- (AX2) Werte sind eindeutig/Kantenrelationen sind Funktionen

  z.B. $\forall x \forall y \forall z((\textsc{adresse}(x, y) \wedge \textsc{adresse}(x, z)) \rightarrow y = z)$
- (AX3) atomare Werte sind unterschiedlich/keine Namensgleichheit

  z.B. $\neg(12345 = merkmalshausen)$
- 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