

---

## FGI 3 – Logik 12. Vorlesung

---

### Elaborierte Formalismen für Quantifikationsbereiche und zusammengesetzte Funktionen

- Einleitung
- Mehrsortige Prädikatenlogik
- Getypter Lambda-Kalkül
- (Logiken 2. Stufe und Höher)



## Bisher behandelt Logiken

---

### **Aussagenlogik / aussagenlogische Modallogik**

- **zusammengesetzte Ausdrücke:** Formeln
- **keine Variablen, Funktions- und Relationssymbole**
- reduzierte Ausdrucksfähigkeit aber leicht verarbeitbar

### **(unsortierte) Prädikatenlogik erster Stufe**

- **zusammengesetzte Ausdrücke:** Formeln und **Terme**
- **Funktions- und Relationssymbole:** syntaktisch spezifiziert durch Stelligkeit
- **Variablen sind Terme** ('Individuenvariablen')
- **Variablen-Bindung** (nur) durch Quantoren
- alle Terme sind von derselben Art (Sorte, Typ)
- ausdrucksmächtig, korrekte und vollständige Kalküle existieren, semi-entscheidbar

## Mehrsortige Prädikatenlogik erster Stufe

---

- atomare **Sorten** sind von endlicher Zahl, ggf. geordnet
- atomare Sorten determinieren **Evaluations-/Quantifikationsbereich**
- **Jeder Term hat eine eindeutige atomare Sorten**
- (Individuen-)Variablen für jede atomare Sorte
- Funktions- und Relationssymbole: syntaktisch spezifiziert durch **komplexe Sorten**
- **keine Vergrößerung der Ausdrucksmächtigkeit**, aber (für Menschen) besser nutzbar
  
- **Signatur**: Spezifikation der Sorten und Sortenzuordnung für verfügbare Symbole
- **Signatur-basierte Grammatik**: Prüfung von Sortenrestriktionen, Bestimmung der Sorte eines zusammengesetzten Terms

## Typen-basierter Formalismus

---

- (muss keine Formeln haben, deshalb auch nicht unbedingt 'Logik')
- Typen können atomar oder komplex sein
- Einheitliches Schema für (induktive) Generierung von (Funktions-) Typen zunehmender Komplexität
- Typen determinieren Evaluationsbereiche
- Jeder (wohlgeformte) Ausdruck hat einen eindeutigen Typ
- Variablen für beliebige Typen
- verfügbare Symbole: syntaktisch spezifiziert durch Typen
  
- **Signatur**: Spezifikation der Typen und Typenzuordnung für verfügbare Symbole
- **Signatur-basierte Grammatik**: Prüfung von Typenrestriktionen, Bestimmung des Typs eines zusammengesetzten Ausdrucks

- Polymorphe Typen sind parametrisierte Typen. (z.B.  $\text{stack}(\alpha)$ ,  $\text{list}(\alpha)$ )). Zu Ihrer Darstellung benötigt man auf Ebene der Typen-Darstellung (Signatur) Variablen (Java: Generische Typen).

## (Typen-basierter) Lambda-Kalkül

---

- (muss keine Formeln haben, deshalb auch nicht unbedingt 'Logik')
- **zusammengesetzte Ausdrücke**: Terme und **Funktionen**
- Variablen-Bindung durch **Lambda** (Funktionsbildungsoperator)
- Lambda bildet Ausdrücke von **komplexerem Typ als seine Teilausdrücke**
  
- Äquivalenz von Lambda-Termen ist für den **getypten Lambda-Kalkül entscheidbar**
- Der **typenfreie Lambda-Kalkül** ist ausdrucksmächtiger als der getypte Lambda-Kalkül, und Äquivalenz ist **nicht entscheidbar**.

## Prädikatenlogik höherer Stufe

---

- **Quantoren** binden auch Variablen mit **komplexen Typen**
- Ausdrucksmächtiger als PL1, **es gibt keine korrekten und vollständigen Kalküle**

### Ausdrückbar in Prädikatenlogik höherer Stufe

- Endlichkeit
  - (eine Menge ist endlich, wenn jede injektive Selbstabbildung surjektiv ist.)
- Induktionsaxiom der natürlichen Zahlen
  - (Jede Eigenschaft, die 0 hat und die jeder Nachfolger einer natürlichen Zahl hat, die die Eigenschaft hat, hat jede natürliche Zahl.)
  - $(\forall P) [(P(0) \wedge (\forall x) [P(x) \supset P(s(x))]) \supset (\forall x) [P(x)]]$
- Transitive Hülle
  - (die transitive Hülle einer binären Relation R ist die kleinste transitive Relation, die R enthält)

---

## **Sorten und Mehrsortige Logik**

---

## Logik und Logische Sprachen: Einige Zwischenbemerkungen

---

### Logische Sprachen

- Sind Repräsentationssprachen
- Erlauben Schlüsse über Domänen
- Basieren auf Operator-Operanden Konstruktionen

### Menschliche Konzeptualisierung von Domänen

- Basiert auf internen Repräsentationen
- Legt Kategorisierung der Entitäten in Domänen zugrunde

### Logische Sprachen sind das Resultat menschlicher Kognition

---

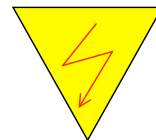
Ch. Habel, C. Eschenbach: FGI 3 – Logik

12 – 8

Diese Zwischenbemerkungen sind als Überleitung zu den beiden letzten Abschnitten des Logik-Teils der LOS-Vorlesung zu verstehen:

Sortenlogiken und typenbasierte Logiken stellen weitere syntaktische und semantische Mittel zur Verfügung, mit denen die interne Struktur von Domänen berücksichtigt werden kann.

- Die Begriffe „Sorte“ und „Typ“ der Logik sind zwar beide im Hinblick auf den Begriff des „Typs“ oder besser „Datentyps“ der Informatik relevant, dürfen aber **nicht** mit diesem identifiziert werden. Hierauf sollte im weiteren Verlauf der beiden Abschnitte sorgfältig geachtet werden.
- Zur Sortenlogik in der Informatik, siehe u.a.:
  - Gallier, Jean H. (1987). *Logic for Computer Science*. New York: John Wiley. (chapter 10).
  - Enderton, Herbert B. (2001). *A Mathematical Introduction to Logic*. Harcourt, Academic Press: San Diego.
  - Sperschneider, V. & G. Antoniou (1991). *Logic. A Foundation for Computer Science*. Addison-Wesley: Workingham, England. (chapter 6)
- Zu logischen Sprachen als Repräsentationssprachen (und zu einigen Anwendungen der Idee der Sortierung in der Wissensrepräsentation), siehe auch:
  - Habel, Christopher (1986). *Prinzipien der Referentialität*. Berlin: Springer-Verlag.  
[Anm. C Habel: Ich habe das Buch 1984–85 geschrieben, und würde jetzt einiges anders machen, aber im Wesentlichen ist dies meine Haltung zur Wissensrepräsentation.]



## Mehrsortige Logik – Ein motivierendes Beispiel (1)

### Ein Postulat der Euklidischen Geometrie (Playfairs Axiom)

Zu jeder Geraden und jedem Punkt, der nicht auf ihr liegt, existiert genau eine Gerade, die durch den Punkt verläuft und parallel zu der Geraden ist.

$$\begin{aligned} &(\forall x) (\forall y) ((\text{line}(x) \wedge \text{point}(y) \wedge \neg \text{on}(y, x)) \supset \\ &\quad (\exists z) (\text{line}(z) \wedge \text{on}(y, z) \wedge \text{parallel}(x, z) \wedge \\ &\quad (\forall v) ((\text{line}(v) \wedge \text{on}(y, v) \wedge \text{parallel}(x, v)) \supset (z = v)))) \end{aligned}$$

- Einsortige Logik: Terme / Quantoren tragen keine „Information“ über die intendierten Anwendungs- / Denotationsbereiche.

- Das hier formulierte Postulat ist eine Reformulierung von Euklids fünftem Postulat, dem „Parallelenpostulat“.  
„Originalformulierung“ (Übers. Thaer, Leipzig 1933):
- Gefordert soll sein:  
...  
*„Und dass, wenn eine gerade Linie beim Schnitt mit zwei geraden Linien bewirkt, dass innen auf derselben Seite entstehende Winkel zusammen kleiner als zwei Rechte werden, dann die zwei geraden Linien bei Verlängerung ins unendliche sich treffen auf der Seite, auf der die Winkel liegen, die zusammen kleiner als zwei Rechte sind.“*
- Diese Reformulierung, auch als Playfairs Axiom (1795) bezeichnet, hat den Vorteil, dass im Kontrast zur Forderung der Existenz einer parallelen Geraden durch den Punkt, Arten nicht-euklidischer Geometrien charakterisiert werden können.

## Mehrsortige Logik – Ein motivierendes Beispiel (2)

---

### Unterscheidung von Objektsorten

- bei der Anwendung logischer Sprachen
- Beispiel: klassische Euklidische Geometrie  
Geraden, Punkt, Ebenen, Kreise, ...
- Mehrsortige Logik: Terme / Quantoren tragen „Information“ über die intendierten Anwendungs- / Denotationsbereiche.
- Beispiel: Variablen, die mit  $l$  beginnen, stehen für Geraden  
Variablen, die mit  $p$  beginnen, stehen für Punkte

### Eine Reformulierung von Playfairs Axiom

$$(\forall l_1)(\forall p) (\neg \text{on}(p, l_1) \supset (\exists l_2) (\text{on}(p, l_2) \wedge \text{parallel}(l_1, l_2) \wedge (\forall l_3) (\text{on}(p, l_3) \wedge \text{parallel}(l_1, l_3)) \supset (l_2 = l_3))))$$

## Mehrsortige Logik – Die Grundidee

---

### Die Menge der **Terme** von $\mathcal{L}_{PL}$ ist in Sorten unterteilt

- Sorten: eine nicht-leere endliche Menge
- Sortierung der Konstanten
- Sortierung der Argumentstruktur von Funktions- und Relationsymbolen
- Sortierung von Funktionswerten

### Notationelle Varianten sortierter Quantifikation

- $(\forall I)$  die Variablenmengen sind unterschieden.  
bzw.  $(\forall p)$
- $(\forall x:LINE)$  Variablen einer Sorte werden durch einen Index  
bzw.  $(\forall x^{LINE})$  ausgezeichnet.
- $(\forall_{LINE} x)$  Für jede Sorte gibt es eigene Quantoren.

## Mehrsortige Logik – Ausdrucksadäquatheit

---

### In jedem Dreieck beträgt die Winkelsumme $180^\circ$

einsortig:  $(\forall x) (\text{dreieck}(x) \supset \text{winkelsumme}(x) = 180^\circ)$

mehrsortig:  $(\forall dr) \text{winkelsumme}(dr) = 180^\circ$

### Es gibt ein gleichseitiges Dreieck

einsortig:  $(\exists x) (\text{dreieck}(x) \wedge \text{gleichseitig}(x))$

mehrsortig:  $(\exists dr) \text{gleichseitig}(dr)$

Hierbei sei jeweils  $dr$  eine Variable der Sorte ‚Dreieck‘

## Sortenkorrektheit

---

### Was ist von

(#)  $\pi$  ist eine Primzahl  $\text{prim}(\pi)$

### zu halten?

### Zwei mögliche Reaktionen

- (#) ist falsch,  
dann ist  $\neg\text{prim}(\pi)$  wahr.  
Dies kann zu Fehlschlüssen führen:  
Eigenschaften von Nicht-Primzahlen: sie haben echte Teiler.
- (#) ist *nicht sinnvoll*, ist *sorteninkorrekt, sortal nicht korrekt*  
denn die Eigenschaft *...ist Primzahl* kann nur natürliche Zahlen  
zu- oder abgesprochen werden.

- Das Konzept der „sortalen Korrektheit“ ist ein zentrales Konzept für die Semantik natürlicher Sprachen.

## Definition eines logischen Systems: Generelles Schema

---

### Spezifikation

- einer formalen Sprache (zur Repräsentation)
- von Evaluations- / Interpretationsprinzipien
- semantischer Kategorisierungen und Beziehungen
- Ableitungs-, Beweisverfahren

## Signaturen

---

### Definition

- Eine **mehr-sortige Signatur** ist ein Paar  $\langle S, \Sigma \rangle$ , wobei die Menge der Sortensymbole  $S$  eine endliche nicht-leere Menge ist und  $\Sigma$  eine abzählbare Menge von Zeichenketten der Form  $f:s_1s_2\dots s_n \rightarrow s$  oder  $P:s_1s_2\dots s_n$ , mit  $s_1, s_2, \dots, s_n, s \in S$ .
- Wenn  $f:s_1s_2\dots s_n \rightarrow s \in \Sigma$ , dann nennen wir  $f$  ein **Funktionssymbol der Sorte**  $s_1s_2\dots s_n \rightarrow s$ .
- Wenn  $c:\rightarrow s \in \Sigma$ , dann nennen wir  $c$  eine **Konstante der Sorte**  $s$ .
- Wenn  $P:s_1s_2\dots s_n \in \Sigma$ , dann nennen wir  $P$  ein **Relationssymbol der Sorte**  $s_1s_2\dots s_n$ .

- Die folgenden Formulierungen (und Symbolisierungen) folgen der Darstellung in Sperschneider & Antoniou (1991).
- Andere Logiker gehen sogar so weit, die Unterscheidung zwischen Termen und Formeln aufzuheben, indem sie eine spezifische Sorte (‚bool‘, ‚assert‘) einführen, so dass die Ausdrücke der Sorte bool genau die Formeln sind.
- Auch die in der Vorlesung 'Semantik der Programmierung' verwendeten 'types of phrases' (Reynolds, 1998, p. 1) wie 'intexp', 'assert', 'comm' und 'spec' können als Unterscheidung von Sorten aufgefasst werden.

## Signatur: Ein Beispiel

### Spezifikation eines Kellerspeichers (über natürlichen Zahlen)

Signatur:  $\langle S, \Sigma \rangle$

Sorten:  $S = \{\text{stack}, \text{nat}\}$

$\Sigma = \{0: \rightarrow \text{nat}, \text{succ}: \text{nat} \rightarrow \text{nat}, +: \text{nat nat} \rightarrow \text{nat}, *: \text{nat nat} \rightarrow \text{nat}, \epsilon: \rightarrow \text{stack}, \text{isempty}: \text{stack}, \text{push}: \text{stack nat} \rightarrow \text{stack}, \text{pop}: \text{stack} \rightarrow \text{stack}, \text{top}: \text{stack} \rightarrow \text{nat}, \text{depth}: \text{stack} \rightarrow \text{nat}\}$

### Formeln dieser Sprache könnten sein

$(\forall n) (\forall s) \text{pop}(\text{push}(s, n)) = s$

$(\forall n) (\forall s) \text{top}(\text{push}(s, n)) = n$

#### Aufgabe 12-1

a) Ergänzen Sie die Spezifikation um folgende Bedingungen:

- 1)  $\epsilon$  ist der einzige Stack, auf den `isempty` zutrifft.
- 2) `depth` gibt die Anzahl der im Stack gespeicherten Objekte an
- 3) Liefert die Anwendung von `succ` auf zwei natürliche Zahlen dasselbe Ergebnis, dann sind die Zahlen identisch.
- 4) Addition von natürlichen Zahlen und Multiplikation

b) Beweisen Sie aus der Spezifikation (wenn Sie mögen, mit einem geeigneten Tableau):

$(\forall n) (\forall s) (\forall m) (\forall s') (\text{push}(s, n) = \text{push}(s', m) \supset (s = s' \wedge n = m))$

c) Das obige Beispiel hat das Problem, dass die Annahme, es handele sich bei `top` und `pop` um Funktionen zu unerwünschten Konsequenzen führen kann, denn was soll `top( $\epsilon$ )` oder `pop( $\epsilon$ )` sein? [Grundsätzlich hat die Logik Probleme mit partiellen Funktionen und diese Probleme werden durch die Sortierung nur zu einem Teil gelöst. In den freien Logiken wird die Möglichkeit, dass ein Term kein Objekt bezeichnet, gezielt behandelt.]

Machen Sie Vorschläge, wie man bei der Spezifikation mit `top( $\epsilon$ )` oder `pop( $\epsilon$ )` umgehen kann.

#### Aufgabe 12-2

Was verändert sich, wenn wir in dem Stack nicht natürliche Zahlen, sondern Objekte einer Sorte `item` speichern?

Aufgabe (ohne Nummer)

Spezifizieren Sie das Verhalten einer First-In-First-Out-Struktur (queue) in einer mehrsortigen Logik.

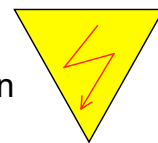
## Mehrsortige Sprachrahmen (2)

---

### Definition

- Eine **sortenbasierte logische Sprache** wird definiert durch eine Signatur  $\langle \mathcal{S}, \Sigma \rangle$  und eine sortierte Menge von Variablen.
- Für jedes  $s \in \mathcal{S}$  existiert eine abzählbar unendliche Menge  $\mathcal{V}_s$  von **Variablen der Sorte  $s$** .
  - Die Variablen der verschiedenen Sorten sind voneinander verschieden und die Variablen sind auch von den Konstantensymbolen aus  $\langle \mathcal{S}, \Sigma \rangle$  verschieden.
  - Die Abbildung, die jeder Sorte ihre Variablen zuordnet, nennen wir die ‚**sortierte Variablenmenge**‘  $\mathcal{V}$ .

- In dieser Variante der mehrsortigen Logik müssen die Sortensymbole nicht als Teile der Formeln auftreten, da die Variablenmengen von vorne herein unterschieden sind.
- Die Disjunktheitsforderung für Sortenzugehörigkeit von Variablen bzw. Konstantensymbolen sollte nicht mit einer Disjunktheitsforderung für die spätere Interpretation verwechselt werden.
  - Das Konstantensymbol „1“ für eine Konstante der Sorte *nat* ist eben etwas anderes als das Konstantensymbol „1,0“ für eine Konstante der Sorte *real*.



## Terme der mehrsortigen Logik

---

### Definition

Es sei  $\langle S, \Sigma \rangle$  eine Signatur und  $\mathcal{V}$  eine sortierte Variablenmenge.

Für jede Sorte  $s$  gilt: Die Menge  $(\mathcal{T}er_s(S, \Sigma, \mathcal{V}))$  der **Terme der Sorte  $s$**  ist die kleinste Menge für die gilt:

1. Jedes Konstantensymbol  $c$  mit  $c: \rightarrow s \in \Sigma$  und jede Variable  $v \in \mathcal{V}_s$  ist ein Element von  $\mathcal{T}er_s(S, \Sigma, \mathcal{V})$ .
2. Wenn  $t_1, \dots, t_n$  Terme sind, mit  $t_i$  ist von der Sorte  $s_i$ , und  $f: s_1 s_2 \dots s_n \rightarrow s \in \Sigma$ , so ist  $f(t_1, \dots, t_n) \in \mathcal{T}er_s(S, \Sigma, \mathcal{V})$ .

### Definition

Eine Substitution  $\sigma$  ist **sortentreu**, wenn für jede Variable  $v \in \mathcal{V}_s$  gilt:  $v\sigma \in \mathcal{T}er_s(S, \Sigma, \mathcal{V})$

Die Bedingung 2 stellt die Sortenkorrektheit sicher und weist außerdem eine Ergebnissorte zu.

Es ist durchaus zulässig, dass dasselbe Funktionssymbol mit verschiedenen Sorten in der Signatur auftritt. Allerdings sollte die Ergebnissorte von den Argumentsorten funktional abhängig sein. Deshalb dürfen auch die Konstanten nicht mehrfach auftreten.

## Atomare Formeln der mehrsortigen Logik

---

### Definition

Es sei  $\langle S, \Sigma \rangle$  eine Signatur und  $\mathcal{V}$  eine sortierte Variablenmenge.

1.  $\perp, \top \in \mathcal{A}tFor(S, \Sigma, \mathcal{V})$  sind atomare Formeln.
2. Wenn  $t_1, \dots, t_n$  Terme sind, mit  $t_i$  ist von der Sorte  $s_i$ , und  $P: s_1 s_2 \dots s_n \in \Sigma$ , so ist  $P(t_1, \dots, t_n) \in \mathcal{A}tFor(S, \Sigma, \mathcal{V})$  eine atomare Formel.
3. In einer mehrsortigen Logik mit Identität zusätzlich:  
Wenn  $s$  eine Sorte ist und  $t_1, t_2$  Terme der Sorte  $s$  sind, dann ist  $(t_1 = t_2) \in \mathcal{A}tFor(S, \Sigma, \mathcal{V})$  eine atomare Formel
4. Das sind alle atomaren Formeln.

Die Bedingung 2 stellt wieder die Sortenkorrektheit sicher.  
Für die Anwendung der Gleichheitsrelation wird die  
Sortenübereinstimmung grundsätzlich vorausgesetzt.

## Formeln der mehrsortigen Logik

---

### Definition

Es sei  $\langle S, \Sigma \rangle$  eine Signatur und  $\mathcal{V}$  eine sortierte Variablenmenge.  
Die Menge der Formeln ( $\text{For}(S, \Sigma, \mathcal{V})$ ) ist die kleinste Menge für die gilt:

1. Jede atomare Formel ist eine Formel.
2. Wenn  $F, G$  Formeln sind, so ist  $\neg F \in \text{For}(S, \Sigma, \mathcal{V})$  und für jeden binären Junktor  $\odot$  ist  $(F \odot G) \in \text{For}(S, \Sigma, \mathcal{V})$ .
3. Wenn  $F$  eine Formel ist und  $v \in \mathcal{V}_s$ , so sind  $(\forall v) F, (\exists v) F \in \text{For}(S, \Sigma, \mathcal{V})$ .

Wie man sieht: Für den Aufbau von komplexeren Formeln spielt die Sortendifferenzierung keine Rolle mehr.

## Definition eines logischen Systems: Generelles Schema

---

### Spezifikation

- einer formalen Sprache (zur Repräsentation)
- von Evaluations- / Interpretationsprinzipien
- semantischer Kategorisierungen und Beziehungen
- Ableitungs-, Beweisverfahren

## Modell für eine Signatur

---

### Definition

Ein *Modell* für eine Signatur  $\langle \mathcal{S}, \Sigma \rangle$  ist ein Paar  $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$ , wobei

- $\mathcal{D}$  ordnet jeder Sorte  $s \in \mathcal{S}$  eine nicht-leere Menge (*Domäne*)  $\mathcal{D}_s$  zu.
- $\mathcal{I}$  ist eine Abbildung (*Interpretation*), die folgendes leistet:
  - für jedes  $c: \rightarrow s \in \Sigma$  ist  $\mathcal{I}(c: \rightarrow s) \in \mathcal{D}_s$
  - für jedes  $f: s_1 s_2 \dots s_n \rightarrow s \in \Sigma$  ist  
 $\mathcal{I}(f: s_1 s_2 \dots s_n \rightarrow s): \mathcal{D}_{s_1} \times \dots \times \mathcal{D}_{s_n} \rightarrow \mathcal{D}_s$
  - für jedes  $P: s_1 s_2 \dots s_n \in \Sigma$  ist  $\mathcal{I}(P: s_1 s_2 \dots s_n) \subseteq \mathcal{D}_{s_1} \times \dots \times \mathcal{D}_{s_n}$

## Sortentreue Zuweisung, Auswertung von Termen

### Definition

Es sei  $\langle S, \Sigma \rangle$  eine Signatur und  $\mathcal{V}$  eine sortierte Variablenmenge.

Eine *sortentreue Zuweisung* für  $\mathcal{V}$  in ein Modell  $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$  für  $\langle S, \Sigma \rangle$  ist eine Abbildung  $\mathcal{A}: \mathcal{V} \rightarrow \mathcal{D}$ , so dass für jede Sorte  $s \in S$  gilt:  $\mathcal{A}(\mathcal{V}_s) \subseteq \mathcal{D}_s$ .  $\mathcal{B}$  ist eine *sortentreue x-Variante* von  $\mathcal{A}$ , falls  $\mathcal{B}$  eine *x-Variante* von  $\mathcal{A}$  und sortentreu ist.

### Definition (Termauswertung)

Sei  $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$  Modell für eine Signatur  $\langle S, \Sigma \rangle$  und  $\mathcal{A}$  eine sortentreue Zuweisung für  $\mathcal{V}$  in  $\mathcal{M}$ .

1. Für  $c: \rightarrow s \in \Sigma$  ist  $c^{\mathcal{I}, \mathcal{A}} = \mathcal{I}(c: \rightarrow s)$
2. Für  $v \in \mathcal{V}_s$  ist  $v^{\mathcal{I}, \mathcal{A}} = \mathcal{A}(v)$
3. Für  $f: s_1 s_2 \dots s_n \rightarrow s \in \Sigma$  und  $t_1, \dots, t_n \in \text{Ter}_{s_i}(S, \Sigma, \mathcal{V})$  ist  
 $[f(t_1, \dots, t_n)]^{\mathcal{I}, \mathcal{A}} = \mathcal{I}(f: s_1 s_2 \dots s_n \rightarrow s)(t_1^{\mathcal{I}, \mathcal{A}}, \dots, t_n^{\mathcal{I}, \mathcal{A}})$

- Hiermit ergibt sich: Wenn  $t$  ein Term der Sorte  $s$  ist, dann ist  $t^{\mathcal{I}, \mathcal{A}} \in \mathcal{D}_s$

## Auswertung für Formeln

---

### Definition

Sei  $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$  Modell für eine Signatur  $\langle S, \Sigma \rangle$  und  $\mathcal{A}$  eine sortentreue Zuweisung für  $\mathcal{V}$  in  $\mathcal{M}$ .

1.  $\perp^{\mathcal{I}, \mathcal{A}} = \mathbf{f}$        $\top^{\mathcal{I}, \mathcal{A}} = \mathbf{t}$
2. Für  $P: s_1 s_2 \dots s_n \in \Sigma$  und  $t_1, \dots, t_n \in \text{Ter}_{s_i}(S, \Sigma, \mathcal{V})$  ist  
 $[P(t_1, \dots, t_n)]^{\mathcal{I}, \mathcal{A}} = \mathbf{t}$  gdw.  $\langle t_1^{\mathcal{I}, \mathcal{A}}, \dots, t_n^{\mathcal{I}, \mathcal{A}} \rangle \in \mathcal{I}(P: s_1 s_2 \dots s_n)$
3.  $[\neg X]^{\mathcal{I}, \mathcal{A}} = \neg[X]^{\mathcal{I}, \mathcal{A}}$
4.  $[(X \odot Y)]^{\mathcal{I}, \mathcal{A}} = X^{\mathcal{I}, \mathcal{A}} \odot Y^{\mathcal{I}, \mathcal{A}}$
5.  $[(\forall x) \Phi]^{\mathcal{I}, \mathcal{A}} = \mathbf{t}$  gdw. für alle sortentreuen  $x$ -Varianten  $\mathcal{B}$  zu  $\mathcal{A}$  gilt  $\Phi^{\mathcal{I}, \mathcal{B}} = \mathbf{t}$ , und sonst  $\mathbf{f}$ .
6.  $[(\exists x) \Phi]^{\mathcal{I}, \mathcal{A}} = \mathbf{t}$  gdw. für eine sortentreue  $x$ -Variante  $\mathcal{B}$  zu  $\mathcal{A}$  gilt:  $\Phi^{\mathcal{I}, \mathcal{B}} = \mathbf{t}$ , und sonst  $\mathbf{f}$

## Definition eines logischen Systems: Generelles Schema

---

### Spezifikation

- einer formalen Sprache (zur Repräsentation)
- von Evaluations- / Interpretationsprinzipien
- **semantischer Kategorisierungen und Beziehungen**
- Ableitungs-, Beweisverfahren

## Folgerbarkeit und Gültigkeit in mehrsortigen Logiken

---

### Definition (Folgerbarkeit mit Sorten)

Sei  $\langle S, \Sigma \rangle$  eine Signatur,  $\mathcal{V}$  eine sortierte Variablenmenge und  $X \in gFor(S, \Sigma, \mathcal{V})$ .

$X$  folgt unter Berücksichtigung der Sorten aus  $C \subseteq gFor(S, \Sigma, \mathcal{V})$ , falls in allen Modellen  $\mathcal{M}$  von  $\langle S, \Sigma \rangle$  (und sortentreuen Zuweisungen für  $\mathcal{V}$  in  $\mathcal{M}$ ), die alle Elemente von  $C$  wahr machen,  $X$  wahr ist. Wir schreiben:  $C \models_{\langle S, \Sigma \rangle} X$ .

$X$  ist unter Berücksichtigung der Sorten gültig, falls  $X$  in allen Modellen  $\mathcal{M}$  von  $\langle S, \Sigma \rangle$  (und sortentreuen Zuweisungen für  $\mathcal{V}$  in  $\mathcal{M}$ ) wahr ist. Wir schreiben:  $\models_{\langle S, \Sigma \rangle} X$ .

- Die Zuweisungen spielen eigentlich keine Rolle, da hier ja nur ueber geschlossene Formeln geredet wird. Andererseits werden die Quantoren ja evaluierte, indem die globale Zuweisung modifiziert wird (Variantenbildung).

## Ordnungssortierte Logiken

---

### Obersorten und Untersorten

- In einer Signatur  $\langle S, \Sigma \rangle$  ist  $S$  partiell geordnet
- Im Modell spiegelt sich diese Ordnung wider als
  - Wenn  $s_1$  Untersorte von  $s_2$  ist, dann gilt  $\mathcal{D}_{s_1} \subseteq \mathcal{D}_{s_2}$ .
- Für Substitutionen gilt, dass Variablen mit übergeordneten Sorten durch Terme mit untergeordneten Sorten ersetzt werden können.

### Beispiel

- Natürliche Zahlen als Untersorte der ganzen Zahlen.

## Definition eines logischen Systems: Generelles Schema

---

### Spezifikation

- einer formalen Sprache (zur Repräsentation)
- von Evaluations- / Interpretationsprinzipien
- semantischer Kategorisierungen und Beziehungen
- **Ableitungs-, Beweisverfahren**

## Simulation mehrsortiger Logiken in einsortigen Logiken

---

### Einsortige Sprache mit Sortenprädikaten

- Es sei  $\langle \mathcal{S}, \Sigma \rangle$  eine Signatur und  $\mathcal{V}$  eine sortierte Variablenmenge
- Die unsortierte Sprache  $\mathcal{L}_{\text{PL}}(\Sigma)$  ergibt sich aus:
  - Alle Konstanten, Funktionssymbole und Relationssymbole aus  $\Sigma$  werden in  $\mathcal{L}_{\text{PL}}(\Sigma)$  übernommen, die Sortenspezifikation dabei ‚vergessen‘, die Stelligkeit erhalten
  - $\text{Sort}(\mathcal{S}) = \{\text{sort}_s \mid s \in \mathcal{S}\} \subseteq \text{Rel}_1(\mathcal{L}_{\text{PL}}(\Sigma))$  eine Menge einstelliger Prädikatssymbole, genannt *Sortenprädikate*.
  - Die Variablenmenge  $\text{Var}(\mathcal{L}_{\text{PL}}(\Sigma)) = \bigcup_{s \in \mathcal{S}} \mathcal{V}_s$  ist unsortiert.
- Dann ist  $\text{For}(\mathcal{S}, \Sigma, \mathcal{V}) \subseteq \text{For}(\mathcal{L}_{\text{PL}}(\Sigma))$

## Simulation mehrsortiger Logiken in einsortigen Logiken

---

### Sortenaxiome

- Es sei  $\langle \mathcal{S}, \Sigma \rangle$  eine Signatur und  $\mathcal{V}$  eine sortierte Variablenmenge und es seien  $x, x_1, \dots, x_n \in \text{Var}(\mathcal{L}_{\text{PL}}(\Sigma))$
- Die **Sortenaxiome** zu  $\langle \mathcal{S}, \Sigma \rangle$  sind:
  - $\text{SortAx}(\mathcal{S}) = \{(\exists x) \text{sort}_s(x) \mid s \in \mathcal{S}\} \subseteq \text{For}(\mathcal{L}_{\text{PL}}(\Sigma))$
  - $\text{SortAx}(\Sigma) = \{(\forall x_1) \dots (\forall x_n) ((\text{sort}_{s_1}(x_1) \wedge \dots \wedge \text{sort}_{s_n}(x_n)) \supset \text{sort}_s(f(x_1, \dots, x_n) \mid f: s_1 s_2 \dots s_n \rightarrow s \in \Sigma) \subseteq \text{For}(\mathcal{L}_{\text{PL}}(\Sigma))$
  - $\text{OrdAx}(\mathcal{S}) = \{(\forall x) (\text{sort}_{s_1}(x) \supset \text{sort}_{s_2}(x)) \mid \text{Wenn } s_1 \text{ Untersorte von } s_2\} \subseteq \text{For}(\mathcal{L}_{\text{PL}}(\Sigma))$

## Simulation mehrsortiger Logiken in einsortigen Logiken

---

### Übersetzung von Formeln

Es sei  $\langle S, \Sigma \rangle$  eine Signatur und  $\mathcal{V}$  eine sortierte Variablenmenge. Die Übersetzungsfunktion  $\phi: \text{For}(S, \Sigma, \mathcal{V}) \rightarrow \text{For}(\mathcal{L}_{\text{PL}}(\Sigma))$ , ist wie folgt definiert:

- Wenn  $F$  eine atomare Formel ist, dann ist  $\phi(F) = F$ .
- Wenn  $F, G$  Formeln sind und  $\odot$  ein binärer Junktor, so sind
  - $\phi(\neg F) = \neg\phi(F)$
  - $\phi((F \odot G)) = (\phi(F) \odot \phi(G))$
- Wenn  $F$  eine Formel ist und  $v \in \mathcal{V}_s$ , so sind
  - $\phi((\forall v) F) = (\forall v) (\text{sort}_s(v) \supset \phi(F))$
  - $\phi((\exists v) F) = (\exists v) (\text{sort}_s(v) \wedge \phi(F))$
- Wenn  $C$  eine Formelmenge ist, dann ist  $\phi(C) = \{\phi(F) \mid F \in C\}$

- Die Übersetzungsfunktion  $\phi$  bildet jede Formel der mehrsortigen Logik auf eine Formel der einsortigen Logik ab.

## Beziehung: Mehrsortige Logik – Einsortige Logik

---

mehrsortige Logik	einsortige Logik
Sorten $S$	Sortenprädikate $Sort(S)$ Sortenaxiome $SorAx(S)$
$\Sigma$	Stelligkeit, Sortenaxiome $SorAx(\Sigma)$
Ober- / Untersorte	Ordnungsaxiome $OrdAx(S)$
Formel $F$	$\phi(F)$ : Übersetzung der Formel
Sortierung quantifizierter Variablen	Relativierte Quantifikation
$(\forall v) F$	$(\forall v) (sort_s(v) \supset \phi(F))$
$(\exists v) F$	$(\exists v) (sort_s(v) \wedge \phi(F))$

## Wirksamkeit der Sortenaxiome

---

### Theorem

Es sei  $\langle S, \Sigma \rangle$  eine Signatur,

$\mathcal{Ax}(\langle S, \Sigma \rangle) = \text{SorAx}(S) \cup \text{SorAx}(\Sigma) \cup \text{OrdAx}(S)$ ,

$\mathcal{V}$  eine sortierte Variablenmenge,

$X \in gFor(S, \Sigma, \mathcal{V})$  und  $\mathbf{C} \subseteq gFor(S, \Sigma, \mathcal{V})$ .

Dann gilt:

$\mathcal{Ax}(\langle S, \Sigma \rangle) \models \phi(X)$  genau dann, wenn  $\models_{\langle S, \Sigma \rangle} X$

$\phi(\mathbf{C}) \cup \mathcal{Ax}(\langle S, \Sigma \rangle) \models \phi(X)$  genau dann, wenn  $\mathbf{C} \models_{\langle S, \Sigma \rangle} X$

- Dieses Theorem sagt nur über Formeln, die sortenkorrekt gebildet sind, etwas aus.

## Tableau-Verfahren für mehrsortige Logiken

---

### Parametermenge

- Parameter müssen Sorten zugeordnet werden.
- Für jede Sorte abzählbar viele Parameter.
- Es werden nur sortenkorrekte Instanzen gebildet.
  
- Konsequenz: Allquantifizierte Formeln müssen nur mit passenden Parametern instanziiert werden.
- (Das ändert nichts an der Semi-Entscheidbarkeit, kann aber trotzdem effizienzsteigernd wirken.)