

### Probleme mit natürlicher Sprache

1. Zuordnung von Wahrheitswerten zu natürlichsprachigen Aussagen ist problematisch. (Ich habe nur ein bißchen getrunken.)
2. Natürliche Sprache ist oft schwer verständlich.
3. Natürliche Sprache ist mehrdeutig.
4. Natürliche Sprache hängt von Kontext ab.

### Aussagenlogik

In der Aussagenlogik gehen wir von "Aussagen" aus, denen wir (zumindest prinzipiell) Wahrheitswerte zuordnen können. Die Aussagen werden durch "Operatoren" verbunden. Durch die Wahl der erlaubten Operatoren erhält man unterschiedliche "Logiken".

### Syntax der Aussagenlogik

Eine atomare Formel hat die Form  $p_i$  (wobei  $i \in \mathbb{N} = \{0, 1, \dots\}$ ). Formeln werden durch folgenden induktiven Prozess definiert:

1. Alle atomaren Formeln und  $\perp$  sind Formeln.
2. Falls  $\varphi$  und  $\psi$  Formeln sind, sind auch  $(\varphi \wedge \psi)$ ,  $(\varphi \vee \psi)$ ,  $(\varphi \rightarrow \psi)$  und  $\neg\varphi$  Formeln.
3. Nichts ist Formel, was sich nicht mittels der obigen Regeln erzeugen läßt.

Präzedenz der Operatoren:

- $\leftrightarrow$  bindet am schwächsten
- $\rightarrow$  Implikation...
- $\vee$  Disjunktion...
- $\wedge$  Konjunktion...
- $\neg$  Falsum bindet am stärksten

### Natürliches Schließen

Ein (mathematischer) Beweis zeigt, wie die Behauptung aus den Voraussetzungen folgt. Analog zeigt ein "Beweisbaum" (= "Herleitung"), wie eine Formel der Aussagenlogik aus Voraussetzungen (ebenfalls Formeln der Aussagenlogik) folgt. Diese "Deduktionen" sind Bäume, deren Knoten mit Formeln beschriftet sind:

- an der Wurzel steht die Behauptung (= Konklusion  $\varphi$ )
- an den Blättern stehen Voraussetzungen (= Hypothesen oder Annahmen aus  $\Gamma$ )
- an den inneren Knoten stehen "Teilergebnisse" und "Begründungen"

### Konstruktion von Deduktionen

Aus der Annahme der Aussage  $\varphi$  folgt  $\varphi$  unmittelbar eine triviale Deduktion:  $\varphi$  mit Hypothesen  $\{\varphi\}$  und Konklusion  $\varphi$ .

### Konjunktionseinführung

Ist D eine Deduktion von  $\varphi$  mit Hypothesen aus  $\Gamma$  und ist E eine Deduktion von  $\psi$  mit Hypothesen aus  $\Gamma$ , so ergibt sich die folgende Deduktion von  $\varphi \wedge \psi$  mit Hypothesen aus  $\Gamma$ :

$$\frac{\varphi \quad \psi}{\varphi \wedge \psi} (\wedge I)$$

### Konjunktionselimination

Ist D eine Deduktion von  $\varphi \wedge \psi$  mit Hypothesen aus  $\Gamma$ , so ergeben sich die folgenden Deduktionen von  $\varphi$  bzw. von  $\psi$  mit Hypothesen aus  $\Gamma$ :

$$\frac{\varphi \wedge \psi}{\varphi} (\wedge E_1)$$

$$\frac{\varphi \wedge \psi}{\psi} (\wedge E_2)$$

### Implikationseinführung

Ist D eine Deduktion von  $\psi$  mit Hypothesen aus  $\Gamma \cup \{\varphi\}$ , so ergibt sich die folgende Deduktion von  $\varphi \rightarrow \psi$  mit Hypothesen aus  $\Gamma$ :

$$\frac{\psi}{\varphi \rightarrow \psi} (\rightarrow I)$$

### Implikationselimination oder modus ponens

Ist D eine Deduktion von  $\varphi$  mit Hypothesen aus  $\Gamma$  und ist E eine Deduktion von  $\varphi \rightarrow \psi$  mit Hypothesen aus  $\Gamma$ , so ergibt sich die folgende Deduktion von  $\psi$  mit Hypothesen aus  $\Gamma$ :

$$\frac{\varphi \quad \varphi \rightarrow \psi}{\psi} (\rightarrow E)$$

### Disjunktionselimination

Ist D eine Deduktion von  $\varphi \vee \psi$  mit Hypothesen aus  $\Gamma$ , ist E eine Deduktion von  $\sigma$  mit Hypothesen aus  $\Gamma \cup \{\varphi\}$  und ist F eine Deduktion von  $\sigma$  mit Hypothesen aus  $\Gamma \cup \{\psi\}$ , so ergibt sich die folgende Deduktion von  $\sigma$  mit Hypothesen aus  $\Gamma$ :

$$\frac{\varphi \vee \psi \quad \frac{\sigma}{\sigma} \quad \sigma}{\sigma} (\vee E)$$

### Negationseinführung

Ist D eine Deduktion von  $\perp$  mit Hypothesen aus  $\Gamma \cup \{\varphi\}$ , so ergibt sich die folgende Deduktion von  $\neg\varphi$  mit Hypothesen aus  $\Gamma$ :

$$\frac{\perp}{\varphi} (\neg I)$$

### Negationselimination

Ist D eine Deduktion von  $\neg\varphi$  mit Hypothesen aus  $\Gamma$  und ist E eine Deduktion von  $\varphi$  mit Hypothesen aus  $\Gamma$ , so ergibt sich die folgende Deduktion von  $\perp$  mit Hypothesen aus  $\Gamma$ :

$$\frac{\neg\varphi \quad \varphi}{\perp} (\neg E)$$

### Falsum

Ist D eine Deduktion von  $\perp$  mit Hypothesen aus  $\Gamma$ , so ergibt sich die folgende Deduktion von  $\varphi$  mit Hypothesen aus  $\Gamma$ :

$$\frac{\perp}{\varphi} (\perp)$$

### reductio ad absurdum

Ist D eine Deduktion von  $\perp$  mit Hypothesen aus  $\Gamma \cup \{\neg\varphi\}$ , so ergibt sich die folgende Deduktion von  $\varphi$  mit Hypothesen aus  $\Gamma$ :

$$\frac{\perp}{\varphi} (raa)$$

### Regeln des natürlichen Schließens

**Definition** Für eine Formelmenge  $\Gamma$  und eine Formel  $\varphi$  schreiben wir  $\Gamma \Vdash \varphi$  wenn es eine Deduktion gibt mit Hypothesen aus  $\Gamma$  und Konklusion  $\varphi$ . Wir sagen "  $\varphi$  ist eine syntaktische Folgerung von  $\Gamma$ ". Eine Formel  $\varphi$  ist ein Theorem, wenn  $\emptyset \Vdash \varphi$  gilt.

$\Gamma \Vdash \varphi$  sagt nichts über den Inhalt der Formeln in  $\Gamma \cup \{\varphi\}$  aus, sondern nur über die Tatsache, dass  $\varphi$  mithilfe des natürlichen Schließens aus den Formeln aus  $\Gamma$  hergeleitet werden kann. Ebenso sagt "  $\varphi$  ist Theorem" nur, dass  $\varphi$  abgeleitet werden kann, über "Wahrheit" sagt dieser Begriff nichts aus.

**Satz** Für alle Formeln  $\varphi$  und  $\psi$  gilt  $\{\neg(\varphi \vee \psi)\} \Vdash \neg\varphi \wedge \neg\psi$ .

**Satz** Für jede Formel  $\varphi$  ist  $\neg\neg\varphi \rightarrow \varphi$  ein Theorem.

**Satz** Für jede Formel  $\varphi$  ist  $\varphi \vee \neg\varphi$  ein Theorem.

### Semantik

Formeln sollen Verknüpfungen von Aussagen widerspiegeln. Idee der Semantik: wenn man jeder atomaren Formel  $p_i$  einen Wahrheitswert zuordnet, so kann man den Wahrheitswert jeder Formel berechnen. Es gibt verschiedene Möglichkeiten, Wahrheitswerte zu definieren:

- Boolesche Logik  $B = \{0, 1\}$ : "wahr" = 1 und "falsch" = 0

- dreiwertige Kleene-Logik  $K_3 = \{0, \frac{1}{2}, 1\}$ : "unbekannt" =  $\frac{1}{2}$
- Fuzzy-Logik  $F = [0, 1]$ : "Grad der Überzeugtheit"
- unendliche Boolesche Algebra  $B_{\mathbb{R}} =$  Menge der Teilmengen von  $\mathbb{R}$
- Heyting-Algebra  $H_{\mathbb{R}} =$  Menge der offenen Teilmengen von  $\mathbb{R}$

### Wahrheitswertebereiche

**Definition** Sei  $W$  eine Menge und  $R \subseteq W \times W$  eine binäre Relation.

- $R$  ist reflexiv, wenn  $(a, a) \in R$  für alle  $a \in W$  gilt.
- $R$  ist antisymmetrisch, wenn  $(a, b), (b, a) \in R$  impliziert, dass  $a = b$  gilt (für alle  $a, b \in W$ ).
- $R$  ist transitiv, wenn  $(a, b), (b, c) \in R$  impliziert, dass  $(a, c) \in R$  gilt (für alle  $a, b, c \in W$ ).
- $R$  ist eine Ordnungsrelation, wenn  $R$  reflexiv, antisymmetrisch und transitiv ist. In diesem Fall heißt das Paar  $(W, R)$  eine partiell geordnete Menge.

**Definition** Sei  $(W, \leq)$  partiell geordnete Menge,  $M \subseteq W$  und  $a \in W$ .

- $a$  ist obere Schranke von  $M$ , wenn  $m \leq a$  für alle  $m \in M$  gilt.
- $a$  ist kleinste obere Schranke oder Supremum von  $M$ , wenn  $a$  obere Schranke von  $M$  ist und wenn  $a \leq b$  für alle oberen Schranken  $b$  von  $M$  gilt. Wir schreiben in diesem Fall  $a = \sup M$ .
- $a$  ist untere Schranke von  $M$ , wenn  $a \leq m$  für alle  $m \in M$  gilt.
- $a$  ist größte untere Schranke oder Infimum von  $M$ , wenn  $a$  untere Schranke von  $M$  ist und wenn  $b \leq a$  für alle unteren Schranken  $b$  von  $M$  gilt. Wir schreiben in diesem Fall  $a = \inf M$ .

**Definition** Ein (vollständiger) Verband ist eine partiell geordnete Menge  $(W, \leq)$ , in der jede Menge  $M \subseteq W$  ein Supremum  $\sup M$  und ein Infimum  $\inf M$  hat.

In einem Verband  $(W, \leq)$  definieren wir:

- $0_W = \inf W$  und  $1_W = \sup W$
- $a \wedge_W b = \inf\{a, b\}$  und  $a \vee_W b = \sup\{a, b\}$  für  $a, b \in W$

In jedem Verband  $(W, \leq)$  gelten  $0_W = \sup \emptyset$  und  $1_W = \inf \emptyset$  (jedes Element von  $W$  ist obere und untere Schranke von  $\emptyset$ ).

**Definition** Ein Wahrheitswertebereich ist ein Tupel  $(W, \leq, \rightarrow, \neg)$ , wobei  $(W, \leq)$  ein Verband und  $W \rightarrow W : W^2 \rightarrow W$  und  $\neg W : W \rightarrow W$  Funktionen sind.

- Boolesche Wahrheitswertebereich  $B$  mit Grundmenge  $B = \{0, 1\}$

$$\begin{aligned} \neg_B(a) &= 1 - a, & \rightarrow_B(a, b) &= \max(b, 1 - a) \\ 0_B &= 0, & 1_B &= 1, \\ a \wedge_B b &= \min(a, b), & a \vee_B b &= \max(a, b) \end{aligned}$$

- Kleenesche Wahrheitswertebereich  $K_3 = \{0, \frac{1}{2}, 1\}$

$$\begin{aligned} \neg_{K_3}(a) &= 1 - a, & \rightarrow_{K_3}(a, b) &= \max(b, 1 - a) \\ \neg_{K_3} &= 0, & 1_{K_3} &= 1 \\ a \wedge_{K_3} b &= \min(a, b), & a \vee_{K_3} b &= \max(a, b) \end{aligned}$$

- Fuzzy-Logik ist definiert durch Grundmenge  $F = [0, 1] \subseteq \mathbb{R}$

$$\begin{aligned} \neg_F(a) &= 1 - a, & \rightarrow_F(a, b) &= \max(b, 1 - a) \\ 0_F &= 0, & 1_F &= 1 \\ a \wedge_F b &= \min(a, b), & a \vee_F b &= \max(a, b) \end{aligned}$$

- Boolesche Wahrheitswertebereich  $B_{\mathbb{R}} = \{A \mid A \subseteq \mathbb{R}\}$

$$\begin{aligned} \neg_{B_{\mathbb{R}}}(A) &= \mathbb{R} \setminus A, & \rightarrow_{B_{\mathbb{R}}}(A, B) &= B \cup \mathbb{R} \setminus A \\ 0_{B_{\mathbb{R}}} &= \emptyset, & 1_{B_{\mathbb{R}}} &= \mathbb{R} \\ A \wedge_{B_{\mathbb{R}}} B &= A \cap B, & A \vee_{B_{\mathbb{R}}} B &= A \cup B \end{aligned}$$

- Heytingsche Wahrheitswertebereich  $H_{\mathbb{R}} = \{A \subseteq \mathbb{R} \mid A \text{ ist offen}\}$

$$\begin{aligned} \neg_{H_{\mathbb{R}}}(A) &= \text{Inneres}(\mathbb{R} \setminus A), & \rightarrow_{H_{\mathbb{R}}}(A, B) &= \text{Inneres}(B \cup \mathbb{R} \setminus A) \\ 0_{H_{\mathbb{R}}} &= \emptyset, & 1_{H_{\mathbb{R}}} &= \mathbb{R} \\ A \wedge_{H_{\mathbb{R}}} B &= A \cap B, & A \vee_{H_{\mathbb{R}}} B &= A \cup B \\ \text{Inneres}(A) &= \{a \in A \mid \exists \epsilon > 0 : (a - \epsilon, a + \epsilon) \subseteq A\} \end{aligned}$$

Sei  $W$  ein Wahrheitswertebereich und  $B$  eine  $W$ -Belegung. Induktiv über den Formelaufbau definieren wir den Wahrheitswert  $\hat{B}(\phi) \in W$  jeder zu  $B$  passenden Formel  $\phi$ :

- $\hat{B}(\perp) = 0_W$
- $\hat{B}(p) = B(p)$  falls  $p$  eine atomare Formel ist
- $\hat{B}((\phi \wedge \psi)) = \hat{B}(\phi) \wedge_W \hat{B}(\psi)$
- $\hat{B}((\phi \vee \psi)) = \hat{B}(\phi) \vee_W \hat{B}(\psi)$
- $\hat{B}((\phi \rightarrow \psi)) = \neg_W(\hat{B}(\phi), \hat{B}(\psi))$
- $\hat{B}(\neg\phi) = \neg_W(\hat{B}(\phi))$

### Folgerung und Tautologie

Sei  $W$  ein Wahrheitswertebereich. Eine Formel  $\phi$  heißt eine  $W$ -Folgerung der Formelmenge  $\Gamma$ , falls für jede  $W$ -Belegung  $B$ , die zu allen Formeln aus  $\Gamma \cup \{\phi\}$  paßt, gilt:  $\inf\{B(\gamma) | \gamma \in \Gamma\} \leq B(\phi)$ . Wir schreiben  $\Gamma \Vdash W\phi$ , falls  $\phi$  eine  $W$ -Folgerung von  $\Gamma$  ist. Eine  $W$ -Tautologie ist eine Formel  $\phi$  mit  $\emptyset \Vdash W\phi$ , d.h.  $B(\phi) = 1_W$  für alle passenden  $W$ -Belegungen  $B$ .

### Korrektheit

**Korrektheitslemma für nat. Schließen & Wahrheitswertebereich  $B$**  Sei  $D$  eine Deduktion mit Hypothesen in der Menge  $\Gamma$  und Konklusion  $\varphi$ . Dann gilt  $\Gamma \vdash_B \varphi$ , d.h.  $\inf\{B(\gamma) | \gamma \in \Gamma\} \leq B(\varphi)$  für alle passenden  $B$ -Belegungen  $B$ .

Beweis: Induktion über die Größe der Deduktion  $D$  (d.h. Anzahl der Regelanwendungen). Ist die letzte Schlußregel in der Deduktion  $D$  von der Form  $(\wedge I), (\vee E), (\rightarrow I)$  oder  $(raa)$ , so ist die Behauptung des Lemmas gezeigt.

**Korrektheitssatz für natürliches Schließen & Wahrheitswertebereich  $B$**  Für jede Menge von Formeln  $\Gamma$  und jede Formel  $\varphi$  gilt  $\Gamma \vdash \varphi \Rightarrow \Gamma \vdash_B \varphi$ .

Beweis: Wegen  $\Gamma \vdash \varphi$  existiert eine Deduktion  $D$  mit Hypothesen in  $\Gamma$  und Konklusion  $\varphi$ . Nach dem Korrektheitslemma folgt  $\Gamma \vdash_B \varphi$ .

**Korollar** Jedes Theorem ist eine  $B$ -Tautologie.

**Korrektheitssatz für natürliches Schließen & Wahrheitswertebereich  $B$**  Für jede Menge von Formeln  $\Gamma$  und jede Formel  $\varphi$  gilt  $\Gamma \vdash \varphi \Rightarrow \Gamma \vdash_{B_R} \varphi$ .

Beweis: verallgemeinere den Beweis von Korrektheitslemma und Korrektheitssatz für  $B$  auf  $B_R$  (Problem: wir haben mehrfach ausgenutzt, dass  $B = \{0, 1\}$  mit  $0 < 1$ )

**Korollar** Jedes Theorem ist eine  $B_R$ -Tautologie.

**Korrektheitslemma für nat. Schließen & Wahrheitswertebereich  $H_R$**  Sei  $D$  eine Deduktion mit Hypothesen in der Menge  $\Gamma$  und Konklusion  $\varphi$ , die die Regel  $(raa)$  nicht verwendet. Dann gilt  $\Gamma \vdash_{H_R} \varphi$ .

Beweis: ähnlich zum Beweis des Korrektheitslemmas für den Wahrheitswertebereich  $B$ . Nur die Behandlung der Regel  $(raa)$  kann nicht übertragen werden.

**Korrektheitssatz für nat. Schließen & Wahrheitswertebereich  $H_R$**  Für jede Menge von Formeln  $\Gamma$  und jede Formel  $\varphi$  gilt  $\Gamma \vdash \varphi$  ohne  $(raa) \Rightarrow \Gamma \vdash_{H_R} \varphi$ .

**Korollar** Jedes  $(raa)$ -frei herleitbare Theorem ist eine  $H_R$ -Tautologie.

Folgerung: Jede Deduktion der Theoreme  $\neg\neg\varphi \rightarrow \varphi$  und  $\varphi \vee \neg\varphi$  ohne Hypothesen verwendet  $(raa)$ .

### Vollständigkeit

- Sei  $W$  einer der Wahrheitswertebereiche  $B, K_3, F, B_R, H_R$
- z.z. ist  $\Gamma \vdash_W \varphi \Rightarrow \Gamma \vdash \varphi$
- dies ist äquivalent zu  $\Gamma \not\vdash \varphi \Rightarrow \Gamma \not\vdash_W \varphi$ 
  - $\Gamma \not\vdash_W \varphi$
  - $\Leftrightarrow \Gamma \cup \{\neg\varphi\}$  konsistent
  - $\Rightarrow \exists \Delta \subseteq \Gamma \cup \{\neg\varphi\}$  maximal konsistent
  - $\Rightarrow \Delta$  erfüllbar
  - $\Rightarrow \Gamma \cup \{\neg\varphi\}$  erfüllbar
  - $\Leftrightarrow \Gamma \not\vdash_B \varphi$
  - $\Rightarrow \Gamma \not\vdash \varphi$

### Konsistente Mengen

**Definition** Sei  $\Gamma$  eine Menge von Formeln.  $\Gamma$  heißt inkonsistent, wenn  $\Gamma \vdash \perp$  gilt. Sonst heißt  $\Gamma$  konsistent.

**Lemma** Sei  $\Gamma$  eine Menge von Formeln und  $\varphi$  eine Formel. Dann gilt  $\Gamma \not\vdash \varphi \Leftrightarrow \Gamma \cup \{\neg\varphi\}$  konsistent.

### Maximal konsistente Mengen

**Definition** Eine Formelmenge  $\Delta$  ist maximal konsistent, wenn sie konsistent ist und wenn gilt " $\Sigma \supseteq \Delta$  konsistent  $\Rightarrow \Sigma = \Delta$ ".

**Satz** Jede konsistente Formelmenge  $\Gamma$  ist in einer maximal konsistenten Formelmenge  $\Delta$  enthalten.

**Lemma 1** Sei  $\Delta$  maximal konsistent und gelte  $\Delta \vdash \varphi$ . Dann gilt  $\varphi \in \Delta$ .

**Lemma 2** Sei  $\Delta$  maximal konsistent und  $\varphi$  Formel. Dann gilt  $\varphi \notin \Delta \Leftrightarrow \neg\varphi \in \Delta$ .

### Erfüllbare Mengen

**Definition** Sei  $\Gamma$  eine Menge von Formeln.  $\Gamma$  heißt erfüllbar, wenn es eine passende  $B$ -Belegung  $B$  gibt mit  $B(\gamma) = 1_B$  für alle  $\gamma \in \Gamma$ .

- Die Erfüllbarkeit einer endlichen Menge  $\Gamma$  ist entscheidbar:
- Berechne Menge  $V$  von in  $\Gamma$  vorkommenden atomaren Formeln
- Probiere alle  $B$ -Belegungen  $B : V \rightarrow B$  durch
- Die Erfüllbarkeit einer endlichen Menge  $\Gamma$  ist NP-vollständig (Satz von Cook)

**Satz** Sei  $\Delta$  eine maximal konsistente Menge von Formeln. Dann ist  $\Delta$  erfüllbar.

Beweis: Definiere eine  $B$ -Belegung  $B$  mittels  $B(p_i) = \begin{cases} 1_B & \text{falls } p_i \in \Delta \\ 0_B & \text{sonst.} \end{cases}$  Wir zeigen für alle Formeln  $\varphi : B(\varphi) = 1_B \Leftrightarrow \varphi \in \Delta$  (\*). Der Beweis erfolgt per Induktion über die Länge von  $\varphi$

**Lemma** Sei  $\Gamma$  eine Menge von Formeln und  $\varphi$  eine Formel. Dann gilt  $\Gamma \not\vdash_B \varphi \Leftrightarrow \Gamma \cup \{\neg\varphi\}$  erfüllbar.

**Beobachtung** Sei  $W$  einer der Wahrheitswertebereiche  $B, K_3, F, H_R$  und  $B_R, \Gamma$  eine Menge von Formeln und  $\varphi$  eine Formel. Dann gilt  $\Gamma \Vdash W\varphi \Rightarrow \Gamma \Vdash B\varphi$ .

**Satz (Vollständigkeitssatz)** Sei  $\Gamma$  eine Menge von Formeln,  $\varphi$  eine Formel und  $W$  einer der Wahrheitswertebereiche  $B, K_3, F, B_R$  und  $H_R$ . Dann gilt  $\Gamma \Vdash_W \varphi \Rightarrow \Gamma \vdash \varphi$ . Insbesondere ist jede  $W$ -Tautologie ein Theorem.

Beweis (indirekt)

- $\Gamma \not\vdash$
- $\Gamma \cup \{\neg\varphi\}$  konsistent
- $\exists \Delta \supseteq \Gamma \cup \{\neg\varphi\}$  maximal konsistent
- $\Rightarrow \Delta$  erfüllbar
- $\Gamma \cup \{\neg\varphi\}$  erfüllbar
- $\Gamma \not\vdash_B \varphi$
- $\Gamma \not\vdash_W \varphi$

### Vollständigkeit und Korrektheit

**Satz** Seien  $\Gamma$  eine Menge von Formeln und  $\varphi$  eine Formel. Dann gilt  $\Gamma \vdash \varphi \Leftrightarrow \Gamma \Vdash_B \varphi$ . Insbesondere ist eine Formel genau dann eine  $B$ -Tautologie, wenn sie ein Theorem ist.

Beweis: Folgt unmittelbar aus Korrektheitssatz und Vollständigkeitssatz. Bemerkung:

- gilt für jede "Boolesche Algebra", z.B.  $B_R$
- $\Gamma \vdash \varphi$  ohne  $(raa) \Leftrightarrow \Gamma \Vdash_{H_R} \varphi$  (Tarski 1938)

### Folgerung 1: Entscheidbarkeit

**Satz** die Menge der Theoreme ist entscheidbar.

Beweis: Sei  $\varphi$  Formel und  $V$  die Menge der vorkommenden atomaren Formeln. Dann gilt  $\varphi$  Theorem

- $\Leftrightarrow \varphi$   $B$ -Tautologie
- $\Leftrightarrow$  für alle Abbildungen  $B : V \rightarrow \{0_B, 1_B\}$  gilt  $B(\varphi) = 1_B$

Da es nur endlich viele solche Abbildungen gibt und  $B(\varphi)$  berechnet werden kann, ist dies eine entscheidbare Aussage.

### Folgerung 2: Äquivalenzen und Theoreme

**Definition** Zwei Formeln  $\alpha$  und  $\beta$  heißen äquivalent ( $\alpha \equiv \beta$ ), wenn für alle passenden  $B$ -Belegungen  $B$  gilt:  $B(\alpha) = B(\beta)$ .

Es gelten die folgenden Äquivalenzen:

1.  $p_1 \vee p_2 \equiv p_2 \vee p_1$
2.  $(p_1 \vee p_2) \vee p_3 \equiv p_1 \vee (p_2 \vee p_3)$
3.  $p_1 \vee (p_2 \wedge p_3) \equiv (p_1 \vee p_2) \wedge (p_1 \vee p_3)$
4.  $\neg(p_1 \vee p_2) \equiv \neg p_1 \wedge \neg p_2$
5.  $p_1 \vee p_1 \equiv p_1$
6.  $(p_1 \wedge \neg p_1) \vee p_2 \equiv p_2$
7.  $\neg\neg p_1 \equiv p_1$
8.  $p_1 \wedge \neg p_1 \equiv \perp$
9.  $p_1 \vee \neg p_1 \equiv \top$
10.  $p_1 \rightarrow p_2 \equiv \neg p_1 \vee p_2$

Bemerkung: Mit den üblichen Rechenregeln für Gleichungen können aus dieser Liste alle gültigen Äquivalenzen hergeleitet werden.

### Zusammenhang zw. Theoremen und Äquivalenzen

**Satz** Seien  $\alpha$  und  $\beta$  zwei Formeln. Dann gilt  $\alpha \equiv \beta \Leftrightarrow (\alpha \leftrightarrow \beta)$  ist Theorem.

**Satz** Sei  $\alpha$  eine Formel. Dann gilt  $\alpha$  ist Theorem  $\Leftrightarrow \alpha \equiv \neg\perp$ .

### Folgerung 3: Kompaktheit

**Satz** Sei  $\Gamma$  eine u.U. unendliche Menge von Formeln und  $\varphi$  eine Formel mit  $\Gamma \Vdash_B \varphi$ . Dann existiert  $\Gamma' \subseteq \Gamma$  endlich mit  $\Gamma' \Vdash_B \varphi$ .

**Folgerung (Kompaktheits- oder Endlichkeitssatz)** Sei  $\Gamma$  eine u.U. unendliche Menge von Formeln. Dann gilt  $\Gamma$  unerfüllbar  $\Leftrightarrow \exists \Gamma' \subseteq \Gamma$  endlich:  $\Gamma'$  unerfüllbar

### 1. Anwendung des Kompaktheitsatzes: Färbbarkeit

**Definition** Ein Graph ist ein Paar  $G = (V, E)$  mit einer Menge  $V$  und  $E \subseteq \binom{V}{2} = \{X \subseteq V : |V| \geq 2\}$ . Für  $W \subseteq V$  sei  $G|_W = (W, E \cap \binom{W}{2})$  der von  $W$  induzierte Teilgraph. Der Graph  $G$  ist 3-färbbar, wenn es eine Abbildung  $f : V \rightarrow \{1, 2, 3\}$  mit  $f(v) \neq f(w)$  für alle  $\{v, w\} \in E$ .

Bemerkung: Die 3-Färbbarkeit eines endlichen Graphen ist NP-vollständig

**Satz** Sei  $G = (N, E)$  ein Graph. Dann sind äquivalent  
 1.  $G$  ist 3-färbbar.  
 2. Für jede endliche Menge  $W \subseteq N$  ist  $G|_W$  3-färbbar.

Behauptung: Jede endliche Menge  $\Delta \subseteq \Gamma$  ist erfüllbar. Nach dem Kompaktheitsatz ist  $\Gamma$  erfüllbar. Sei  $B$  erfüllende Belegung. Für  $n \in N$  existiert genau ein  $c \in \{1, 2, 3\}$  mit  $B(p_{n,c}) = 1$ . Setze  $f(n) = c$ . Dann ist  $f$  eine gültige Färbung des Graphen  $G$ .

### 2. Anwendung des Kompaktheitsatzes: Parkettierungen

**Definition** Ein Kachelsystem besteht aus einer endlichen Menge  $C$  von "Farben" und einer Menge  $K$  von Abbildungen  $\{N, O, S, W\} \rightarrow C$  von "Kacheln". Eine Kachelung von  $G \subseteq Z \times Z$  ist eine Abbildung  $f : G \rightarrow K$  mit

- $f(i, j)(N) = f(i, j + 1)(S)$  für alle  $(i, j), (i, j + 1) \in G$
- $f(i, j)(O) = f(i + 1, j)(W)$  für alle  $(i, j), (i + 1, j) \in G$

**Satz** Sei  $K$  ein Kachelsystem. Es existiert genau dann eine Kachelung von  $Z \times Z$ , wenn für jedes  $n \in N$  eine Kachelung von  $\{(i, j) : |i|, |j| \leq n\}$  existiert.

Bemerkung: Der Kompaktheitssatz gilt auch, wenn die Menge der atomaren Formeln nicht abzählbar ist.

### Erfüllbarkeit

**Erfüllbarkeitsproblem** Eingabe: Formel  $\Gamma$ . Frage: existiert eine B-Belegung  $B$  mit  $B(\Gamma) = 1_B$ .

### Hornformeln (Alfred Horn, 1918-2001)

**Definition** Eine Hornklausel hat die Form  $(\neg \perp \wedge p_1 \wedge p_2 \wedge \dots \wedge p_n) \rightarrow q$  für  $n \geq 0$ , atomare Formeln  $p_1, p_2, \dots, p_n$  und  $q$  atomare Formel oder  $q = \perp$ . Eine Hornformel ist eine Konjunktion von Hornklauseln.

- $\{p_1, p_2, \dots, p_n\} \rightarrow q$  für Hornklausel  $(\neg \perp \wedge p_1 \wedge p_2 \wedge \dots \wedge p_n) \rightarrow q$  insbes.  $\emptyset \rightarrow q$  für  $\neg \perp \rightarrow q$
- $\{(M_i \rightarrow q_i) \mid 1 \leq i \leq n\}$  für Hornformel  $\bigwedge_{1 \leq i \leq n} (M_i \rightarrow q_i)$

Bemerkung, in der Literatur auch:

- $\{\neg p_1, \neg p_2, \dots, \neg p_n, q\}$  für  $\{p_1, \dots, p_n\} \rightarrow q$  mit  $q$  atomare Formel
- $\{\neg p_1, \neg p_2, \dots, \neg p_n\}$  für  $\{p_1, \dots, p_n\} \rightarrow \perp$
- $\square$  für  $\emptyset \rightarrow \perp$ , die "leere Hornklausel"

### Markierungsalgorithmus

Eingabe: eine endliche Menge  $\Gamma$  von Hornklauseln.

- while es gibt in  $\Gamma$  eine Hornklausel  $M \rightarrow q$ , so dass alle  $p \in M$  markiert sind und  $q$  unmarkierte atomare Formel ist: do markiere  $q$  (in allen Hornklauseln in  $\Gamma$ )
- if  $\Gamma$  enthält eine Hornklausel der Form  $M \rightarrow \perp$ , in der alle  $p \in M$  markiert sind then return "unerfüllbar" else return "erfüllbar"

- Der Algorithmus terminiert: in jedem Durchlauf der while-Schleife wird wenigstens eine atomare Formel markiert. Nach endlich vielen Schritten terminiert die Schleife also.

- Wenn der Algorithmus eine atomare Formel  $q$  markiert und wenn  $B$  eine B-Belegung ist, die  $\Gamma$  erfüllt, dann gilt  $B(q) = 1_B$ . Beweis: wir zeigen induktiv über  $n$ : Wenn  $q$  in einem der ersten  $n$  Schleifendurchläufe markiert wird, dann gilt  $B(q) = 1_B$ .
- Wenn der Algorithmus "unerfüllbar" ausgibt, dann ist  $\Gamma$  unerfüllbar. Beweis: indirekt, wir nehmen also an, dass der Algorithmus "unerfüllbar" ausgibt,  $B$  aber eine B-Belegung ist, die  $\Gamma$  erfüllt.
- Wenn der Algorithmus "erfüllbar" ausgibt, dann erfüllt die folgende B-Belegung alle Formeln aus  $\Gamma$ :  

$$B(p_i) = \begin{cases} 1_B & \text{der Algorithmus markiert } p_i \\ 0_B & \text{sonst} \end{cases}$$
- Also gilt  $B(M \rightarrow q) = 1_B$  für alle Hornklauseln aus  $\Gamma$ , d.h.  $\Gamma$  ist erfüllbar.

**Satz** Sei  $\Gamma$  endliche Menge von Hornklauseln. Dann terminiert der Markierungsalgorithmus mit dem korrekten Ergebnis.

Beweis: Die Aussagen 1.-4. beweisen diesen Satz. Bemerkungen:

- Mit einer geeigneten Implementierung läuft der Algorithmus in linearer Zeit.
- Wir haben sogar gezeigt, dass bei Ausgabe von "erfüllbar" eine erfüllende B-Belegung berechnet werden kann.

### SLD-Resolution

**Definition** Sei  $\Gamma$  eine Menge von Hornklauseln. Eine SLD-Resolution aus  $\Gamma$  ist eine Folge  $(M_0 \rightarrow \perp, M_1 \rightarrow \perp, \dots, M_m \rightarrow \perp)$  von Hornklauseln mit

- $(M_0 \rightarrow \perp) \in \Gamma$
- für alle  $0 \leq n < m$  existiert  $(N \rightarrow q) \in \Gamma$  mit  $q \in M_n$  und  $M_{n+1} = M_n \setminus \{q\} \cup N$

Beispiel:

- $\Gamma = \{ \{BH\} \rightarrow AK, \{AK, BH\} \rightarrow \perp, \{RL, AK\} \rightarrow BH, \emptyset \rightarrow RL, \emptyset \rightarrow AK \}$
- $M_0 = \{AK, BH\}$
- $M_1 = M_0 \setminus \{BH\} \cup \{RL, AK\} = \{RL, AK\}$
- $M_2 = M_1 \setminus \{RL\} \cup \emptyset = \{AK\}$
- $M_3 = M_2 \setminus \{AK\} \cup \emptyset = \emptyset$

**Lemma A** Sei  $\Gamma$  eine (u.U. unendliche) Menge von Hornklauseln und  $(M_0 \rightarrow \perp, M_1 \rightarrow \perp, \dots, M_m \rightarrow \perp)$  eine SLD-Resolution aus  $\Gamma$  mit  $M_m = \emptyset$ . Dann ist  $\Gamma$  nicht erfüllbar.

**Lemma B** Sei  $\Gamma$  eine (u.U. unendliche) unerfüllbare Menge von Hornklauseln. Dann existiert eine SLD-Resolution  $(M_0 \rightarrow \perp, \dots, M_m \rightarrow \perp)$  aus  $\Gamma$  mit  $M_m = \emptyset$ .

Beweis: Endlichkeitssatz: es gibt  $\Delta \subseteq \Gamma$  endlich und unerfüllbar. Bei Eingabe von  $\Delta$  terminiert Markierungsalgorithmus mit "unerfüllbar"

- $r \geq 0$ .. Anzahl der Runden
- $q_i$ ... Atomformel, die in  $i$  Runde markiert wird ( $1 \leq i \leq r$ )

Behauptung: Es gibt  $m \leq r$  und SLD-Resolution  $(M_0 \rightarrow \perp, \dots, M_m \rightarrow \perp)$  aus  $\Delta$  mit  $M_m = \emptyset$  und  $M_n \subseteq \{q_1, q_2, \dots, q_{r-n}\}$  f.a.  $0 \leq n \leq m$ . (5)

**Satz** Sei  $\Gamma$  eine (u.U. unendliche) Menge von Hornklauseln. Dann sind äquivalent:

- $\Gamma$  ist nicht erfüllbar.
- Es gibt eine SLD-Resolution  $(M_0 \rightarrow \perp, M_1 \rightarrow \perp, \dots, M_m \rightarrow \perp)$  aus  $\Gamma$  mit  $M_m = \emptyset$ .

Die Suche nach einer SLD-Resolution mit  $M_m = \emptyset$  kann grundsätzlich auf zwei Arten erfolgen:

- Breitensuche:

- findet SLD-Resolution mit  $M_m = \emptyset$  (falls sie existiert), da Baum endlich verzweigend ist (d.h. Niveaus sind endlich)
- hoher Platzbedarf, da ganze Niveaus abgespeichert werden müssen (in einem Binärbaum der Tiefe  $n$  kann es Niveaus der Größe  $2^n$  geben)

- Tiefensuche:

- geringerer Platzbedarf (in einem Binärbaum der Tiefe  $n$  hat jeder Ast die Länge  $\leq n$ )
- findet existierende SLD-Resolution mit  $M_m = \emptyset$  nicht immer

## Kapitel 2: Prädikatenlogik

Um über Graphen Aussagen in der Aussagenlogik zu machen, verwenden wir Formeln  $\varphi_{i,j}$  für  $1 \leq i, j \leq 9$  mit  $\varphi_{i,j} = \begin{cases} \neg \perp & \text{falls } (v_i, v_j) \text{ Kante} \\ \perp & \text{sonst} \end{cases}$

- Die aussagenlogische Formel  $\bigvee_{1 \leq i, j \leq 9} \varphi_{i,j}$  sagt aus, dass der Graph eine Kante enthält.
- Die aussagenlogische Formel  $\bigwedge_{1 \leq i \leq 9} \bigvee_{1 \leq j \leq 9} \varphi_{i,j}$  sagt aus, dass jeder Knoten einen Nachbarn hat
- Die aussagenlogische Formel  $\bigvee_{1 \leq i, j, k \leq 9} \text{verschieden } \varphi_{i,j} \wedge \varphi_{j,k} \wedge \varphi_{k,i}$  sagt aus, dass der Graph ein Dreieck enthält.

### Kodierung in einer Struktur

- Grundmenge
- Teilmengen
- Relationen: Enthaltensein in einer Teilmenge oder Beziehungen zwischen Objekten ausdrücken
- Funktion: Objekte auf andere Objekte abbilden
- Konstante: Nullstellige Funktionen (ohne Argumente)

### Syntax der Prädikatenlogik

**Definition** Eine Signatur ist ein Tripel  $\Sigma = (\Omega, Rel, ar)$ , wobei  $\Omega$  und  $Rel$  disjunkte Mengen von Funktions- und Relationsnamen sind und  $ar : \Omega \cup Rel \rightarrow \mathbb{N}$  eine Abbildung ist.

Beispiel:  $\Omega = \{f, dk\}$  mit  $ar(f) = 1, ar(dk) = 0$

**Definition** Die Menge der Variablen ist  $Var = \{x_0, x_1, \dots\}$ .

**Definition** Sei  $\Sigma$  eine Signatur. Die Menge  $T_\Sigma$  der  $\Sigma$ -Terme ist induktiv definiert:

- Jede Variable ist ein Term, d.h.  $Var \subseteq T_\Sigma$
- ist  $f \in \Omega$  mit  $ar(f) = k$  und sind  $t_1, \dots, t_k \in T_\Sigma$ , so gilt  $f(t_1, \dots, t_k) \in T_\Sigma$
- Nichts ist  $\Sigma$ -Term, was sich nicht mittels der obigen Regeln erzeugen lässt.

**Definition** Sei  $\Sigma$  Signatur. Die atomaren  $\Sigma$ -Formeln sind die Zeichenketten der Form

- $R(t_1, t_2, \dots, t_k)$  falls  $t_1, t_2, \dots, t_k \in T_\Sigma$  und  $R \in Rel$  mit  $ar(R) = k$  oder
- $t_1 = t_2$  falls  $t_1, t_2 \in T_\Sigma$  oder
- $\perp$ .

**Definition** Sei  $\Sigma$  Signatur.  $\Sigma$ -Formeln werden durch folgenden induktiven Prozess definiert:

- Alle atomaren  $\Sigma$ -Formeln sind  $\Sigma$ -Formeln.
- Falls  $\varphi$  und  $\psi$   $\Sigma$ -Formeln sind, sind auch  $(\varphi \wedge \psi), (\varphi \vee \psi)$  und  $(\varphi \rightarrow \psi)$   $\Sigma$ -Formeln.
- Falls  $\varphi$  eine  $\Sigma$ -Formel ist, ist auch  $\neg \varphi$  eine  $\Sigma$ -Formel.
- Falls  $\varphi$  eine  $\Sigma$ -Formel und  $x \in Var$ , so sind auch  $\forall x \varphi$  und  $\exists x \varphi$   $\Sigma$ -Formeln.
- Nichts ist  $\Sigma$ -Formel, was sich nicht mittels der obigen Regeln erzeugen lässt.

**Definition** Sei  $\Sigma$  eine Signatur. Die Menge  $FV(\varphi)$  der freien Variablen einer  $\Sigma$ -Formel  $\varphi$  ist induktiv definiert:

- Ist  $\varphi$  atomare  $\Sigma$ -Formel, so ist  $FV(\varphi)$  die Menge der in  $\varphi$  vorkommenden Variablen.
- $FV(\varphi \square \Psi) = FV(\varphi) \cup FV(\Psi)$  für  $\square \in \{\wedge, \vee, \rightarrow\}$
- $FV(\neg\varphi) = FV(\varphi)$
- $FV(\exists x\varphi) = FV(\forall x\varphi) = FV(\varphi) \setminus \{x\}$ .

Eine  $\Sigma$ -Formel  $\varphi$  ist geschlossen oder ein  $\Sigma$ -Satz, wenn  $FV(\varphi) = \emptyset$  gilt.

**Definition** Sei  $\Sigma$  eine Signatur. Eine  $\Sigma$ -Struktur ist ein Tupel  $A = (U_A, (f^A)_{f \in \Omega}, (R^A)_{R \in Rel})$ , wobei

- $U_A$  eine nichtleere Menge, das Universum,
- $R^A \supseteq U_A^{ar(R)}$  eine Relation der Stelligkeit  $ar(R)$  für  $R \in Rel$  und
- $f^A : U_A^{ar(f)} \rightarrow U_A$  eine Funktion der Stelligkeit  $ar(f)$  für  $f \in \Omega$  ist.

Bemerkung:  $U_A^0 = \{\()\}$ .

- Also ist  $a^A : U_A^0 \rightarrow U_A$  für  $a \in \Omega$  mit  $ar(a) = 0$  vollständig gegeben durch  $a^A(\()) \in U_A$ . Wir behandeln 0-stellige Funktionen daher als Konstanten.
- Weiterhin gilt  $R^A = \emptyset$  oder  $R^A = \{\()\}$  für  $R \in Rel$  mit  $ar(R) = 0$ .

**Definition** Sei  $\Sigma$  eine Signatur,  $\varphi$  eine  $\Sigma$ -Formel,  $\Delta$  eine Menge von  $\Sigma$ -Formeln und  $A$  eine  $\Sigma$ -Struktur.

- $A \models \varphi$  ( $A$  ist Modell von  $\varphi$ ) falls  $A \models_p \varphi$  für alle Variableninterpretationen  $\rho$  gilt.
- $A \models \Delta$  falls  $A \models \Psi$  für alle  $\Psi \in \Delta$ .

**Definition** Sei  $\Sigma$  eine Signatur,  $\varphi$  eine  $\Sigma$ -Formel,  $\Delta$  eine Menge von  $\Sigma$ -Formeln und  $A$  eine  $\Sigma$ -Struktur.

- $\Delta$  ist erfüllbar, wenn es  $\Sigma$ -Struktur  $B$  und Variableninterpretation  $\rho : Var \rightarrow U_B$  gibt mit  $B \models_\rho \Psi$  für alle  $\Psi \in \Delta$ .
- $\varphi$  ist semantische Folgerung von  $\Delta$  ( $\Delta \models \varphi$ ) falls für alle  $\Sigma$ -Strukturen  $B$  und alle Variableninterpretationen  $\rho : Var \rightarrow U_B$  gilt: Gilt  $B \models_\rho \Psi$  für alle  $\Psi \in \Delta$ , so gilt auch  $B \models_\rho \varphi$ .
- $\varphi$  ist allgemeingültig, falls  $B \models \rho\varphi$  für alle  $\Sigma$ -Strukturen  $B$  und alle Variableninterpretationen  $\rho$  gilt.

Bemerkung:  $\varphi$  allgemeingültig gdw.  $\emptyset \models \varphi$  gdw.  $\{\neg\varphi\}$  nicht erfüllbar. Hierfür schreiben wir auch  $\models \varphi$ .

### Substitutionen

**Definition** Eine Substitution besteht aus einer Variable  $x \in Var$  und einem Term  $t \in T_\Sigma$ , geschrieben  $[x := t]$ .

Die Formel  $\varphi[x := t]$  ist die Anwendung der Substitution  $[x := t]$  auf die Formel  $\varphi$ . Sie entsteht aus  $\varphi$ , indem alle freien Vorkommen von  $x$  durch  $t$  ersetzt werden. Sie soll das über  $t$  aussagen, was  $\varphi$  über  $x$  ausgesagt hat.

**Lemma** Seien  $\Sigma$  Signatur,  $A$   $\Sigma$ -Struktur,  $\rho : Var \rightarrow U_A$  Variableninterpretation,  $x \in Var$  und  $s, t \in T_\Sigma$ . Dann gilt  $\rho(s[x := t]) = \rho(x \rightarrow \rho(t))(s)$ .

**Definition** Sei  $[x := t]$  Substitution und  $\varphi$   $\Sigma$ -Formel. Die Substitution  $[x := t]$  heißt für  $\varphi$  zulässig, wenn für alle  $y \in Var$  gilt:  $y$  Variable in  $t \Rightarrow \varphi$  enthält weder  $\exists y$  noch  $\forall y$

**Lemma** Sei  $\Sigma$  Signatur,  $A$   $\Sigma$ -Struktur,  $\rho : Var \rightarrow U_A$  Variableninterpretation,  $x \in Var$  und  $t \in T_\Sigma$ . Ist die Substitution  $[x := t]$  für die  $\Sigma$ -Formel  $\varphi$  zulässig, so gilt  $A \models_p \varphi[x := t] \Leftrightarrow A \models_{\rho[x \rightarrow \rho(t)]} \varphi$ .

### Korrektheit

Der Beweis des Korrektheitslemmas für das natürliche Schließen kann ohne große Schwierigkeiten erweitert werden.

**Lemma V0** Sei  $\Sigma$  eine Signatur,  $\Gamma$  eine Menge von  $\Sigma$ -Formeln und  $\varphi$  eine  $\Sigma$ -Formel. Sei weiter  $D$  eine Deduktion mit Hypothesen in  $\Gamma$  und Konklusion  $\varphi$ , die die Regeln des natürlichen Schließens der Aussagenlogik verwendet. Dann gilt  $\Gamma \Vdash \varphi$ .

Umgekehrt ist nicht zu erwarten, dass aus  $\Gamma \Vdash \varphi$  folgt, dass es eine Deduktion mit Hypothesen in  $\Gamma$  und Konklusion  $\varphi$  gibt, denn die bisher untersuchten Regeln erlauben keine Behandlung von  $=, \forall$  bzw.  $\exists$ . Solche Regeln werden wir jetzt einführen. Zunächst kümmern wir uns um Atomformeln der Form  $t_1 = t_2$ . Hierfür gibt es die zwei Regeln ( $R$ ) und ( $GfG$ ):

**Reflexivität** Für jeden Term  $t$  ist  $\frac{}{t = t}$  eine hypothesenlose Deduktion mit Konklusion  $t = t$ .

**Gleiches-für-Gleiches** Seien  $s$  und  $t$  Terme und  $\varphi$  Formel, so dass die Substitutionen  $[x := s]$  und  $[x := t]$  für  $\varphi$  zulässig sind. Sind  $D$  und  $E$  Deduktionen mit Hypothesen in  $\Gamma$  und Konklusionen  $\varphi[x := s]$  bzw.  $s = t$ , so ist das folgende eine Deduktion mit Hypothesen in  $\Gamma$  und Konklusion  $\varphi[x := t]$

Bedingung: über keine Variable aus  $s$  oder  $t$  wird in  $\varphi$  quantifiziert

**Lemma V1** Sei  $\Sigma$  eine Signatur,  $\Gamma$  eine Menge von  $\Sigma$ -Formeln und  $\varphi$  eine  $\Sigma$ -Formel. Sei weiter  $D$  eine Deduktion mit Hypothesen in  $\Gamma$  und Konklusion  $\varphi$ , die die Regeln des natürlichen Schließens der Aussagenlogik, ( $R$ ) und ( $GfG$ ) verwendet. Dann gilt  $\Gamma \Vdash \varphi$ .

**$\forall$ -Einführung** Sei  $D$  eine Deduktion mit Hypothesen in  $\Gamma$  und Konklusion  $\varphi$  und sei  $x$  eine Variable, die in keiner Formel aus  $\Gamma$  frei vorkommt. Dann ist das folgende eine Deduktion mit Hypothesen in  $\Gamma$  und Konklusion  $\forall x\varphi : \frac{\varphi}{\forall x\varphi}$   
Bedingung:  $x$  kommt in keiner Hypothese frei vor

**Lemma V2** Sei  $\Sigma$  eine Signatur,  $\Gamma$  eine Menge von  $\Sigma$ -Formeln und  $\varphi$  eine  $\Sigma$ -Formel. Sei weiter  $D$  eine Deduktion mit Hypothesen in  $\Gamma$  und Konklusion  $\varphi$ , die die Regeln des natürlichen Schließens der Aussagenlogik, ( $R$ ), ( $GfG$ ) und ( $\forall$ -I) verwendet. Dann gilt  $\Gamma \Vdash \varphi$ .

**$\forall$ -Elimination** Sei  $D$  eine Deduktion mit Hypothesen in  $\Gamma$  und Konklusion  $\forall x\varphi$  und sei Term, so dass Substitution  $[x:=t]$  für  $\varphi$  zulässig ist. Dann ist das folgende eine Deduktion mit Hypothesen in  $\Gamma$  und Konklusion  $\varphi[x := t] : \frac{\forall x\varphi}{\varphi[x:=t]}$   
Bedingung: über keine Variable aus  $t$  wird in  $\varphi$  quantifiziert

**Lemma V3** Sei  $\Sigma$  eine Signatur,  $\Gamma$  eine Menge von  $\Sigma$ -Formeln und  $\varphi$  eine  $\Sigma$ -Formel. Sei weiter  $D$  eine Deduktion mit Hypothesen in  $\Gamma$  und Konklusion  $\varphi$ , die die Regeln des natürlichen Schließens der Aussagenlogik, ( $R$ ), ( $GfG$ ), ( $\forall$ -I) und ( $\forall$ -E) verwendet.  $\iota$  Dann gilt  $\Gamma \Vdash \varphi$ .

**$\exists$ -Elimination** Sei  $\Gamma$  eine Menge von Formeln, die die Variable  $x$  nicht frei enthalten und enthalte die Formel  $\sigma$  die Variabel  $x$  nicht frei. Wenn  $D$  eine Deduktion mit Hypothesen in  $\Gamma$  und Konklusion  $\exists x\varphi$  und  $E$  eine Deduktion mit Hypothesen in  $\Gamma \cup \{\varphi\}$  und Konklusion  $\sigma$  ist, dann ist das folgende eine Deduktion mit Hypothesen in  $\Gamma$  und Konklusion  $\sigma : \frac{\exists x\varphi \quad \sigma}{\sigma}$   
Bedingung:  $x$  kommt in den Hypothesen und in  $\sigma$  nicht frei vor.

**Lemma V4** Sei  $\sigma$  eine Signatur,  $\Gamma$  eine Menge von  $\Sigma$ -Formeln und  $\varphi$  eine  $\sigma$ -Formel. Sei weiter  $D$  eine Deduktion mit Hypothesen in  $\Gamma$  und Konklusion  $\varphi$ , die die Regeln des natürlichen Schließens der Aussagenlogik, ( $R$ ), ( $GfG$ ), ( $\forall$ -I), ( $\forall$ -E) und ( $\exists$ -E) verwendet. Dann gilt  $\Gamma \Vdash \varphi$ .

**$\exists$ -Einführung** Sei die Substitution  $[x := t]$  für die Formel  $\varphi$  zulässig. Sei weiter  $D$  eine Deduktion mit Hypothesen in  $\Gamma$  und Konklusion  $\varphi[x := t]$ . Dann ist das folgende eine Deduktion mit Hypothesen in  $\Gamma$  und Konklusion  $\exists x\varphi : \frac{\varphi[x:=t]}{\exists x\varphi}$   
Bedingung: über keine Variable in  $t$  wird in  $\varphi$  quantifiziert

**Korrektheitslemma für das natürliche Schließen in der Prädikatenlogik** Sei  $\sigma$  eine Signatur,  $\Gamma$  eine Menge von  $\Sigma$ -Formeln und  $\varphi$  eine  $\sigma$ -Formel. Sei weiter  $D$  eine Deduktion mit Hypothesen in  $\Gamma$  und Konklusion  $\varphi$ , die die Regeln des natürlichen Schließens der Aussagenlogik, ( $R$ ), ( $GfG$ ), ( $\forall$ -I), ( $\forall$ -E), ( $\exists$ -E) und ( $\exists$ -I) verwendet. Dann gilt  $\Gamma \Vdash \varphi$ .

**Definition** Für eine Menge  $\Gamma$  von  $\Sigma$ -Formeln und eine  $\Sigma$ -Formel  $\varphi$  schreiben wir  $\Gamma \vdash \varphi$  wenn es eine Deduktion gibt mit Hypothesen in  $\Gamma$  und Konklusion  $\varphi$ . Wir sagen " $\varphi$  ist eine syntaktische Folgerung von  $\Gamma$ ". Eine Formel  $\varphi$  ist ein Theorem, wenn  $\emptyset \vdash \varphi$  gilt.

**Korrektheitssatz** Für eine Menge von  $\Sigma$ -Formeln  $\Gamma$  und eine  $\Sigma$ -Formel  $\varphi$  gilt  $\Gamma \vdash \varphi \Rightarrow \Gamma \Vdash \varphi$ .

### Vollständigkeit

**Definition** Eine Menge  $\Delta$  von Formeln hat Konkretisierungen, wenn für alle  $\exists x\varphi \in \Delta$  ein variablenloser Term  $t$  existiert mit  $\varphi[x := t] \in \Delta$ .

**Satz** Sei  $\Delta$  eine maximal konsistente Menge von  $\Sigma$ -Formeln. Dann existiert eine Signatur  $\Sigma^+ \supseteq \Sigma$  und eine maximal konsistente Menge von  $\Sigma^+$ -Formeln mit Konkretisierungen, so dass  $\Delta \subseteq \Delta^+$ .

**Satz** Sei  $\Delta^+$  maximal konsistente Menge von  $\Sigma^+$ -Formeln mit Konkretisierungen. Dann ist  $\Delta^+$  erfüllbar.

**Satz: Vollständigkeitssatz der Prädikatenlogik** Sei  $\Gamma$  eine Menge von  $\Sigma$ -Formeln und  $\varphi$  eine  $\Sigma$ -Formel. Dann gilt  $\Gamma \Vdash \varphi \Rightarrow \Gamma \vdash \varphi$ . Insbesondere ist jede allgemeingültige Formel ein Theorem.

**Satz** Sei  $\Gamma$  höchstens abzählbar unendliche und konsistente Menge von Formeln. Dann hat  $\Gamma$  ein höchstens abzählbar unendliches Modell.

### Vollständigkeit & Korrektheit für Prädikatenlogik

**Satz** Seien  $\Gamma$  eine Menge von  $\Sigma$ -Formeln und  $\varphi$  eine  $\Sigma$ -Formel. Dann gilt  $\Gamma \vdash \varphi \Leftrightarrow \Gamma \Vdash \varphi$ . Insbesondere ist eine  $\Sigma$ -Formel genau dann allgemeingültig, wenn sie ein Theorem ist.

Beweis: Folgt unmittelbar aus Korrektheitssatz und Vollständigkeitssatz.

### Folgerung 1: Kompaktheit

**Satz** Seien  $\Gamma$  eine u.U. unendliche Menge von  $\Sigma$ -Formeln und  $\varphi$  eine  $\Sigma$ -Formel mit  $\Gamma \Vdash \varphi$ . Dann existiert  $\Gamma' \subseteq \Gamma$  endlich mit  $\Gamma' \Vdash \varphi$ .

**Folgerung (Kompaktheits- oder Endlichkeitssatz)** Sei  $\Gamma$  eine u.U. unendliche Menge von  $\Sigma$ -Formeln. Dann gilt  $\Gamma$  erfüllbar  $\Leftrightarrow \forall \Gamma' \subseteq \Gamma$  endlich:  $\Gamma'$  erfüllbar

**Satz** Sei  $\Delta$  eine u.U. unendliche Menge von  $\Sigma$ -Formeln, so dass für jedes  $n \in \mathbb{N}$  eine endliche Struktur  $A_n$  mit  $A_n \models \Delta$  existiert, die wenigstens  $n$  Elemente hat. Dann existiert eine unendliche Struktur  $A$  mit  $A \models \Delta$ .

**Folgerung 2: Löwenheim-Skolem**

Frage: Gibt es eine Menge  $\Gamma$  von  $\Sigma$ -Formeln, so dass für alle Strukturen  $A$  gilt:  $A \models \Gamma \Leftrightarrow A \cong (\mathbb{R}, +, *, 0, 1)$ ?

**Satz von Löwenheim-Skolem** Sei  $\Gamma$  erfüllbare und höchstens abzählbar unendliche Menge von  $\Sigma$ -Formeln. Dann existiert ein höchstens abzählbar unendliches Modell von  $\Gamma$ .

**Folgerung 3: Semi-Entscheidbarkeit**

**Satz** Die Menge der allgemeingültigen  $\Sigma$ -Formeln ist semi-entscheidbar.

Teste für jede Zeichenkette  $w$  nacheinander, ob sie hypothesenlose Deduktion mit Konklusion  $\varphi$  ist. Wenn ja, so gib aus "  $\varphi$  ist allgemeingültig ". Ansonsten gehe zur nächsten Zeichenkette über.

**Der Satz von Church**

Die Menge der allgemeingültigen  $\Sigma$ -Formeln ist nicht entscheidbar. Wegen  $\varphi$  allgemeingültig  $\Leftrightarrow \neg\varphi$  unerfüllbar reicht es zu zeigen, dass die Menge der erfüllbaren Sätze nicht entscheidbar ist.

**Definition** Eine Horn-Formel ist eine Konjunktion von  $\Sigma$ -Formeln der Form  $\forall x_1 \forall x_2 \dots \forall x_n ((\neg \perp \wedge \alpha_1 \wedge \alpha_2 \wedge \dots \wedge \alpha_m) \rightarrow \beta)$ , wobei  $\alpha_1, \dots, \alpha_m$  und  $\beta$  atomare  $\Sigma$ -Formeln sind.

**Lemma** Angenommen, das Korrespondenzsystem  $I$  hat keine Lösung. Dann ist die Horn-Formel  $\varphi_I = \psi_I \wedge \forall x(R(x, x) \rightarrow x = e)$  erfüllbar.

**Lemma** Sei  $B$  Struktur mit  $B \models \psi_I$ . Dann gilt  $(f_{i_1 i_2 \dots i_n}^B(e^B), f_{i_1 i_2 \dots i_n}^B(e^B)) \in R^B$  für alle  $n \geq 0, 1 \leq i_1, i_2, \dots, i_n \leq k$ .

**Lemma** Angenommen,  $(i_1, \dots, i_n)$  ist eine Lösung von  $I$ . Dann ist die  $\Sigma$ -Formel  $\varphi_I$  unerfüllbar.

**Satz** Die Menge der unerfüllbaren Horn-Formeln ist nicht entscheidbar.

**Folgerung (Church 1936)** Die Menge der allgemeingültigen  $\Sigma$ -Formeln ist nicht entscheidbar.

Beweis: Eine  $\Sigma$ -Formel  $\varphi$  ist genau dann unerfüllbar, wenn  $\neg\varphi$  allgemeingültig ist. Also ist  $\varphi \rightarrow \neg\varphi$  eine Reduktion der unentscheidbaren Menge der unerfüllbaren  $\Sigma$ -Formeln auf die Menge der allgemeingültigen  $\Sigma$ -Formeln, die damit auch unentscheidbar ist.

**Theorie der natürlichen Zahlen**

**Definition** Sei  $A$  eine Struktur. Dann ist  $Th(A)$  die Menge der prädikatenlogischen  $\Sigma$ -Formeln  $\varphi$  mit  $A \models \varphi$ . Diese Menge heißt die (elementare) Theorie von  $A$ .

Beispiel: Sei  $N = (N, \leq, +, *, 0, 1)$ . Dann gelten

- $(\forall x \forall y : x + y = y + x) \in Th(N)$
- $(\forall x \exists y : x + y = 0) \notin Th(N)$
- aber  $(\forall x \exists y : x + y = 0) \in Th((\mathbb{Z}, +, 0))$ .

**Lemma** Die Menge  $Th(N)$  aller Sätze  $\varphi$  mit  $N \models \varphi$  ist nicht entscheidbar.

**Zahlentheoretisches Lemma** Für alle  $n \in N, x_0, x_1, \dots, x_n \in N$  existieren  $c, d \in N$ , so dass für alle  $0 \leq i \leq n$  gilt:  $x_i = c \text{ mod}(1 + d * (i + 1))$ .

**Satz** Sei  $A$  eine Struktur, so dass  $Th(A)$  semi-entscheidbar ist. Dann ist  $Th(A)$  entscheidbar.

**Korollar** Die Menge  $Th(N)$  der Aussagen  $\varphi$  mit  $N \models \varphi$  ist nicht semi-entscheidbar.

**Korollar (1. Gödelscher Unvollständigkeitssatz)** Sei  $\Gamma$  eine semi-entscheidbare Menge von Sätzen mit  $N \models \gamma$  für alle  $\gamma \in \Gamma$ . Dann existiert ein Satz  $\varphi$  mit  $\Gamma \not\models \varphi$  und  $\Gamma \not\models \neg\varphi$  (d.h. "  $\Gamma$  ist nicht vollständig ").

**2. Semi Entscheidungsverfahren für allgemeingültige Formeln**

bekanntes Verfahren mittels natürlichem Schließen: Suche hypothesenlose Deduktion mit Konklusion  $\psi$ .

**Definition** Zwei  $\Sigma$ -Formeln  $\varphi$  und  $\psi$  heißen erfüllbarkeitsäquivalent, wenn gilt:  $\varphi$  ist erfüllbar  $\Leftrightarrow \psi$  ist erfüllbar

**Elimination von Gleichungen**

**Definition** Eine  $\Sigma$ -Formel ist gleichungsfrei, wenn sie keine Teilformel der Form  $s = t$  enthält.

Idee: Die Formel  $\varphi'$  entsteht aus  $\varphi$ , indem alle Teilformeln der Form  $x = y$  durch  $GI(x, y)$  ersetzt werden, wobei  $GI$  ein neues Relationssymbol ist. Notationen

- Sei  $\Sigma = (\Omega, Rel, ar)$  endliche Signatur und  $\varphi$   $\Sigma$ -Formel
- $\Sigma_{GI} = (\Omega, Rel \cup \{GI\}, ar_{GI})$  mit  $ar_{GI}(f)$  für alle  $f \in \Omega \cup Rel$  und  $ar_{GI}(GI) = 2$
- Für eine  $\Sigma$ -Formel  $\varphi$  bezeichnet  $\varphi_{GI}$  die  $\Sigma_{GI}$ -Formel, die aus  $\varphi$  entsteht, indem alle Vorkommen und Teilformen  $s = t$  durch  $GI(s, t)$  ersetzt werden.

**Definition** Sei  $A$  eine  $\Sigma$ -Struktur und  $\sim$  eine binäre Relation auf  $U_A$ . Die Relation  $\sim$  heißt Kongruenz auf  $A$ , wenn gilt:

- $\sim$  ist eine Äquivalenzrelation (d.h. reflexiv, transitiv und symmetrisch)
- für alle  $f \in \Omega$  mit  $k = ar(f)$  und alle  $a_1, b_1, \dots, a_k, b_k \in U_A$  gilt  $a_1 \sim b_1, a_2 \sim b_2, \dots, a_k \sim b_k \Rightarrow f^A(a_1, \dots, a_k) \sim f^A(b_1, \dots, b_k)$
- für alle  $R \in Rel$  mit  $k = ar(R)$  und alle  $a_1, b_1, \dots, a_k, b_k \in U_A$  gilt  $a_1 \sim b_1, \dots, a_k \sim b_k, (a_1, \dots, a_k) \in R^A \Rightarrow (b_1, \dots, b_k) \in R^A$ .

**Definition** Sei  $A$  eine  $\Sigma$ -Struktur und  $\sim$  eine Kongruenz auf  $A$ .

- Für  $a \in U_A$  sei  $[a] = \{b \in U_A \mid a \sim b\}$  die Äquivalenzklasse von  $a$  bzgl  $\sim$ .
- Dann definieren wir den Quotienten  $B = A \setminus \sim$  von  $A$  bzgl  $\sim$  wie folgt:
  - $U_B = U_A \setminus \sim = \{[a] \mid a \in U_A\}$
  - Für jedes  $f \in \Omega$  mit  $ar(f) = k$  und alle  $a_1, \dots, a_k \in U_A$  setzen wir  $f^B([a_1], \dots, [a_k]) = [f^A(a_1, \dots, a_k)]$
  - für jede  $R \in Rel$  mit  $ar(R) = k$  setzen wir  $R^B = \{([a_1], [a_2], \dots, [a_k]) \mid (a_1, \dots, a_k) \in R^A\}$
- Sei  $p : Var \rightarrow U_A$  Variableninterpretation. Dann definieren die Variableninterpretation  $p \setminus \sim : Var \rightarrow U_B : x \rightarrow [p(x)]$ .

**Lemma 1** Sei  $A$  Struktur,  $p : Var \rightarrow U_A$  Variableninterpretation und  $\sim$  Kongruenz. Seien weiter  $B = A \setminus \sim$  und  $p_B = p \setminus \sim$ . Dann gilt für jeden Term  $t : [p(t)] = p_B(t)$

**Lemma 2** Sei  $A$   $\Sigma$ -Struktur,  $\sim$  Kongruenz und  $B = A \setminus \sim$ . Dann gilt für alle  $R \in Rel$  mit  $k = ar(R)$  und alle  $c_1, \dots, c_k \in U_A : ([c_1], [c_2], \dots, [c_k]) \in R^B \Leftrightarrow (c_1, c_2, \dots, c_k) \in R^A$

**Satz** Seien  $A$   $\Sigma_{GI}$ -Struktur und  $p : Var \rightarrow U_A$  Variableninterpretation, so dass  $\sim = GI^A$  Kongruenz auf  $A$  ist. Seien  $B = A \setminus \sim$  und  $p_B = p \setminus \sim$ . Dann gilt für alle  $\Sigma$ -Formeln  $\varphi : A \models_p \varphi_{GI} \Leftrightarrow B \models_{p_B} \varphi$

**Lemma** Aus einer endlichen Signatur  $\Sigma$  kann ein gleichungsfreier Horn-Satz  $Kong_\Sigma$  über  $\Sigma_{GI}$  berechnet werden, so dass für alle  $\Sigma_{GI}$ -Strukturen  $A$  gilt:  $A \models Kong_\Sigma \Leftrightarrow GI^A$  ist eine Kongruenz

**Satz** Aus einer endlichen Signatur  $\Sigma$  und einer  $\Sigma$ -Formel  $\varphi$  kann eine gleichungsfreie und erfüllbarkeitsäquivalente  $\Sigma_{GI}$ -Formel  $\varphi'$  berechnet werden. Ist  $\varphi$  Horn Formel, so ist auch  $\varphi'$  Horn Formel.

**Skolemform**

Ziel: Jede  $\Sigma$ -Formel  $\varphi$  ist erfüllbarkeitsäquivalent zu einer  $\Sigma'$ -Formel  $\varphi' = \forall x_1 \forall x_2 \dots \forall x_k \psi$ , wobei  $\psi$  kein Quantoren enthält,  $\varphi'$  heißt in Skolemform. 2 Schritte:

- Quantoren nach vorne (d.h. Pränexform)
- Existenzquantoren eliminieren

**Definition** Zwei  $\Sigma$ -Formeln  $\varphi$  und  $\psi$  sind äquivalent (kurz:  $\varphi \equiv \psi$ ), wenn für alle  $\Sigma$ -Strukturen  $A$  und alle Variableninterpretationen  $\rho$  gilt:  $A \models_\rho \varphi \Leftrightarrow A \models_\rho \psi$ .

**Lemma** Seien  $Q \in \{\exists, \forall\}$  und  $\oplus \in \{\wedge, \vee, \rightarrow, \leftarrow\}$ . Sei  $\varphi = (Qx\alpha) \oplus \beta$  und sei  $y$  eine Variable, die weder in  $\alpha$  noch in  $\beta$  vorkommt. Dann gilt  $\varphi \equiv \begin{cases} Qy(\alpha[x := y] \oplus \beta) & \text{falls } \oplus \in \{\wedge, \vee, \leftarrow\} \\ \forall y(\alpha[x := y] \rightarrow \beta) & \text{falls } \oplus = \rightarrow, Q = \exists \\ \exists y(\alpha[x := y] \rightarrow \beta) & \text{falls } \oplus = \rightarrow, Q = \forall \end{cases}$

**Lemma** Seien  $Q \in \{\exists, \forall\}$  und  $\oplus \in \{\wedge, \vee, \rightarrow, \leftarrow\}$ . Sei  $\varphi = (Qx\alpha) \oplus \beta$  und sei  $y$  eine Variable, die weder in  $\alpha$  noch in  $\beta$  vorkommt. Dann gilt  $\varphi \equiv \begin{cases} Qy(\alpha[x := y] \oplus \beta) & \text{falls } \oplus \in \{\wedge, \vee, \leftarrow\} \\ \forall y(\alpha[x := y] \rightarrow \beta) & \text{falls } \oplus = \rightarrow, Q = \exists \\ \exists y(\alpha[x := y] \rightarrow \beta) & \text{falls } \oplus = \rightarrow, Q = \forall \end{cases}$

**Satz** Aus einer endlichen Signatur  $\Sigma$  und einer  $\Sigma$ -Formel  $\varphi$  kann eine äquivalente  $\Sigma$ -Formel  $\varphi' = Q_1 x_1 Q_2 x_2 \dots Q_k x_k \psi$  (mit  $Q_i \in \{\exists, \forall\}$ ,  $\psi$  quantorenfrei und  $x_i$  paarweise verschieden) berechnet werden. Eine Formel  $\varphi'$  dieser Form heißt Pränexform. Ist  $\varphi$  gleichungsfrei, so ist auch  $\varphi'$  gleichungsfrei.

Idee:

- wandle Formel in Pränexform um
- eliminiere  $\exists$ -Quantoren durch Einführen neuer Funktionssymbole

**Lemma** Die Formeln  $\varphi$  und  $\varphi'$  sind erfüllbarkeitsäquivalent.

**Satz** Aus einer Formel  $\varphi$  kann man eine erfüllbarkeitsäquivalente Formel  $\varphi$  in Skolemform berechnen. Ist  $\varphi$  gleichungsfrei, so auch  $\varphi$ .

**Herbrand-Strukturen und Herbrand-Modelle**

Sei  $\Sigma = (\Omega, Rel, ar)$  eine Signatur. Wir nehmen im folgenden an, dass  $\Omega$  wenigstens ein Konstantensymbol enthält. Das Herbrand-Universum  $D(\Sigma)$  ist die Menge aller variablenfreien  $\Sigma$ -Terme.

Beispiel:  $\Omega = \{b, f\}$  mit  $ar(b) = 0$  und  $ar(f) = 1$ . Dann gilt  $D(\Sigma) = \{b, f(b), f(f(b)), f(f(f(b))), \dots\}$

**Satz** Sei  $\varphi$  eine gleichungsfreie Aussage in Skolemform.  $\varphi$  ist genau dann erfüllbar, wenn  $\varphi$  ein Herbrand-Modell besitzt.

**Behauptung 1** Ist  $\psi$  eine quantoren- und gleichungsfreie Aussage, so gilt  $A \vdash_{\rho} \psi \Leftrightarrow B \vdash_{\rho_B} \psi$ . Diese Behauptung wird induktiv über den Aufbau von  $\psi$  gezeigt.  
Für allgemeine Formeln in Skolemform (also u.U. mit Quantoren) können wir Behauptung 1 nicht zeigen, sondern höchstens die folgende Abschwächung.

**Behauptung 2** Ist  $\psi$  eine gleichungsfreie Aussage in Skolemform, so gilt  $A \vdash_{\rho} \psi \Rightarrow B \vdash_{\rho_B} \psi$  (hieraus folgt dann  $B \vdash_{\rho_B} \varphi$  wegen  $A \vdash_{\rho} \varphi$ ). Diese Behauptung wird induktiv über die Anzahl  $n$  der Quantoren in  $\psi$  bewiesen.

### Die Herbrand-Expansion

Jedes Herbrand-Modell  $A$  von  $\varphi$

- hat als Universum das Herbrand-Universum  $D(\Sigma) = \{a, f(a), f^2(a), \dots\} = \{f^n(a) | n \geq 0\}$
- erfüllt  $f^A(f^n(a)) = f^{n+1}(a)$  für alle  $n \geq 0$

**Konstruktion** Sei  $B : \{P(t_1, \dots, t_k) | P \in Rel, k = ar(P), t_1, \dots, t_k \in D(\Sigma)\} \rightarrow B$  eine B-Belegung. Die hiervon induzierte Herbrand-Struktur  $A_B$  ist gegeben durch  $P^{A_B} = \{(t_1, \dots, t_k) | t_1, \dots, t_k \in D(\Sigma), B(P(t_1, \dots, t_k)) = 1\}$  für alle  $P \in Rel$  mit  $ar(P) = k$ .

**Lemma** Für jede quantoren- und gleichungsfreie Aussage  $\alpha$  und jede Variableninterpretation  $\rho$  in  $A_B$  gilt  $A_B \models_{\rho} \alpha \Leftrightarrow B(\alpha) = 1$ .

**Lemma** Sei  $\varphi = \forall y_1 \forall y_2 \dots \forall y_n \psi$  gleichungsfreie Aussage in Skolemform. Sie hat genau dann ein Herbrand-Modell, wenn die Formelmenge  $E(\varphi)$  (im aussagenlogischen Sinn) erfüllbar ist.

**Satz von Gödel-Herbrand-Skolem** Sei  $\varphi$  gleichungsfreie Aussage in Skolemform. Sie ist genau dann erfüllbar, wenn die Formelmenge  $E(\varphi)$  (im aussagenlogischen Sinn) erfüllbar ist.

**Satz von Herbrand** Eine gleichungsfreie Aussage  $\varphi$  in Skolemform ist genau dann unerfüllbar, wenn es eine endliche Teilmenge von  $E(\varphi)$  gibt, die (im aussagenlogischen Sinn) unerfüllbar ist. (Jacques Herbrand (1908-1931))

### Algorithmus von Gilmore

Eingabe:  $\varphi$

```
n:=0;
repeat n := n + 1;
until { alpha_1, alpha_2, ..., alpha_n } ist unerfüllbar;
//(dies kann mit Mitteln der Aussagenlogik getestet werden)
Gib 'unerfüllbar' aus und stoppe.
```

Folgerung: Sei  $\varphi$  eine gleichungsfreie Aussage in Skolemform. Dann gilt:

- Wenn die Eingabeformel  $\varphi$  unerfüllbar ist, dann terminiert der Algorithmus von Gilmore und gibt "unerfüllbar" aus.
- Wenn die Eingabeformel  $\varphi$  erfüllbar ist, dann terminiert der Algorithmus von Gilmore nicht, d.h. er läuft unendlich lange.

### Berechnung von Lösungen

- Wir müssen die Unerfüllbarkeit einer gleichungsfreien Horn-Formel der Prädikatenlogik testen.
- Ist  $\varphi$  gleichungsfreie Horn-Klausel der Prädikatenlogik, so ist  $E(\varphi)$  eine Menge von Horn-Klauseln der Aussagenlogik.

Schreib- und Sprechweise

- $\{\alpha_1, \alpha_2, \dots, \alpha_n\} \rightarrow \beta$  für Horn-Klausel der Prädikatenlogik  $(\neg \perp \wedge \alpha_1 \wedge \alpha_2 \wedge \dots \wedge \alpha_n) \rightarrow \beta$  insbes.  $\emptyset \rightarrow \beta$  für  $\neg \perp \rightarrow \beta$

- $\{(N_i \rightarrow \beta_i) | 1 \leq i \leq m\}$  für Horn-Formel  $\bigwedge_{1 \leq i \leq m} (N_i \rightarrow \beta_i)$

Folgerung: Sei  $\varphi = \bigwedge_{1 \leq i \leq n} \varphi_i$  gleichungsfreie Horn-Formel der Prädikatenlogik. Dann ist  $\varphi$  genau dann unerfüllbar, wenn  $\bigcup_{1 \leq i \leq n} E(\varphi_i)$  im aussagenlogischen Sinne unerfüllbar ist.  
Folgerung: Eine gleichungsfreie Horn-Formel der Prädikatenlogik  $\varphi = \bigwedge_{1 \leq i \leq n} \varphi_i$  ist genau dann unerfüllbar, wenn es eine SLD-Resolution  $(M_0 \rightarrow \perp, M_1 \rightarrow \perp, \dots, M_m \rightarrow \perp)$  aus  $\bigcup_{1 \leq i \leq n} E(\varphi_i)$  mit  $M_m = \emptyset$  gibt.

### Substitutionen

Eine verallgemeinerte Substitution  $\sigma$  ist eine Abbildung der Menge der Variablen in die Menge aller Terme, so daß nur endlich viele Variable  $x$  existieren mit  $\sigma(x) \neq x$ .  
Sei  $Def(\sigma) = \{x \text{ Variable} | x \neq \sigma(x)\}$  der Definitionsbereich der verallgemeinerten Substitution  $\sigma$ . Für einen Term  $t$  definieren wir den Term  $t\sigma$  wie folgt induktiv:

- $x\sigma = \sigma(x)$
- $[f(t_1, \dots, t_k)]\sigma = f(t_1\sigma, \dots, t_k\sigma)$  für Terme  $t_1, \dots, t_k, f \in \Omega$  und  $k = ar(f)$
- Für eine atomare Formel  $\alpha = P(t_1, \dots, t_k)$  (d.h.  $P \in Rel, ar(P) = k, t_1, \dots, t_k$  Terme) sei  $\alpha\sigma = P(t_1\sigma, \dots, t_k\sigma)$

**Lemma** Seien  $\sigma$  Substitution,  $x$  Variable und  $t$  Term, so dass

- (i)  $x \notin Def(\sigma)$  und
- (ii)  $x$  in keinem der Terme  $y\sigma$  mit  $y \in Def(\sigma)$  vorkommt. Dann gilt  $[x := t]\sigma = \sigma[x := t\sigma]$ .

### Unifikator/Allgemeinster Unifikator

Gegeben seien zwei gleichungsfreie Atomformeln  $\alpha$  und  $\beta$ . Eine Substitution  $\sigma$  heißt Unifikator von  $\alpha$  und  $\beta$ , falls  $\alpha\sigma = \beta\sigma$ . Ein Unifikator  $\sigma$  von  $\alpha$  und  $\beta$  heißt allgemeinster Unifikator von  $\alpha$  und  $\beta$ , falls für jeden Unifikator  $\sigma'$  von  $\alpha$  und  $\beta$  eine Substitution  $\tau$  mit  $\sigma' = \sigma\tau$  existiert.

**Lemma** Sind  $\sigma_1$  und  $\sigma_2$  allgemeinste Unifikatoren von  $\alpha$  und  $\beta$ , so existiert eine Variablenumbenennung  $\rho$  mit  $\sigma_2 = \sigma_1\rho$ .

### Unifikationsalgorithmus

- Eingabe: Paar  $(\alpha, \beta)$  gleichungsfreier Atomformeln  $\sigma :=$  Substitution mit  $Def(\sigma) = \emptyset$  (d.h. Identität)
- while  $\alpha\sigma \neq \beta\sigma$  do
  - Suche die erste Position, an der sich  $\alpha\sigma$  und  $\beta\sigma$  unterscheiden
  - if keines der beiden Symbole an dieser Position ist eine Variable
  - then stoppe mit "nicht unifizierbar"
  - else sei  $x$  die Variable und  $t$  der Term in der anderen Atomformel (möglicherweise auch eine Variable)
    - \* if  $x$  kommt in  $t$  vor
    - \* then stoppe mit "nicht unifizierbar"
    - \* else  $\sigma := \sigma[x := t]$
- endwhile
- Ausgabe:  $\sigma$

**Satz**

- (A) Der Unifikationsalgorithmus terminiert für jede Eingabe.
- (B) Wenn die Eingabe nicht unifizierbar ist, so terminiert der Unifikationsalgorithmus mit der Ausgabe "nicht unifizierbar".
- (C) Wenn die Eingabe  $(\alpha, \beta)$  unifizierbar ist, dann findet der Unifikationsalgorithmus einen allgemeinsten Unifikator von  $\alpha$  und  $\beta$ .

**Lemma (A)** Der Unifikationsalgorithmus terminiert für jede Eingabe  $(\alpha, \beta)$ .

**Lemma (B)** Wenn die Eingabe nicht unifizierbar ist, so terminiert der Unifikationsalgorithmus mit der Ausgabe "nicht unifizierbar".

**Lemma (C1)** Sei  $\sigma'$  ein Unifikator der Eingabe  $(\alpha, \beta)$ , so dass keine Variable aus  $\alpha$  oder  $\beta$  auch in einem Term aus  $\{y\sigma' | y \in Def(\sigma')\}$  vorkommt. Dann terminiert der Unifikationsalgorithmus erfolgreich und gibt einen Unifikator  $\sigma$  von  $\alpha$  und  $\beta$  aus. Außerdem gibt es eine Substitution  $\tau$  mit  $\sigma' = \sigma\tau$ .

die Aussage des Lemmas:

- Nach (2) terminiert der Algorithmus erfolgreich mit der Substitution  $\sigma_N$ . Daher gilt aber  $\alpha\sigma_N = \beta\sigma_N$ , d.h.  $\sigma_N$  ist ein Unifikator.
- Nach (1) gibt es auch eine Substitution  $\tau_n$  mit  $\sigma' = \sigma_N\tau_n$ .

**Lemma (C)** Sei die Eingabe  $(\alpha, \beta)$  unifizierbar. Dann terminiert der Unifikationsalgorithmus erfolgreich und gibt einen allgemeinsten Unifikator  $\sigma$  von  $\alpha$  und  $\beta$  aus.

**Satz**

- (A) Der Unifikationsalgorithmus terminiert für jede Eingabe.
- (B) Wenn die Eingabe nicht unifizierbar ist, so terminiert der Unifikationsalgorithmus mit der Ausgabe "nicht unifizierbar".
- (C) Wenn die Eingabe  $(\alpha, \beta)$  unifizierbar ist, dann findet der Unifikationsalgorithmus immer einen allgemeinsten Unifikator von  $\alpha$  und  $\beta$ .

### Prädikatenlogische SLD-Resolution

**Definition** Sei  $\Gamma$  eine Menge von gleichungsfreien Horn-Klauseln der Prädikatenlogik. Eine SLD-Resolution aus  $\Gamma$  ist eine Folge  $((M_0 \rightarrow \perp, \sigma_0), (M_1 \rightarrow \perp, \sigma_1), \dots, (M_m \rightarrow \perp, \sigma_m))$  von Horn-Klauseln und Substitutionen mit

- $(M_0 \rightarrow \perp) \in \Gamma$  und  $Def(\sigma_0) = \emptyset$
- für alle  $0 \leq n < m$  existieren  $\emptyset \neq Q \subseteq M_n, (N \rightarrow \alpha) \in \Gamma$  und Variablenumbenennung  $\rho$ , so dass
  - $(N \cup \{\alpha\})\rho$  und  $M_n$  variabelndisjunkt sind,
  - $\sigma_{n+1}$  ein allgemeinster Unifikator von  $\alpha\rho$  und  $Q$  ist und
  - $M_{n+1} = (M_n \setminus Q \cup N\rho)\sigma_{n+1}$ .

Dann sind äquivalent:

1.  $\Gamma \Vdash \Psi(s_1, \dots, s_i)$ .
2. Es gibt eine SLD-Resolution  $((M_n \rightarrow \perp, \sigma_n))_{0 \leq n \leq m}$  aus  $\Gamma \cup \{M_0 \rightarrow \perp\}$  mit  $M_0 = \{\Psi(x_1, \dots, x_i)\}$  und  $M_m = \emptyset$  und eine Substitution  $\tau$ , so dass  $s_i = x_i\sigma_0\sigma_1\dots\sigma_m\tau$  für alle  $1 \leq i \leq i$  gilt.

**Lemma** Sei  $\Gamma$  Menge von gleichungsfreien Horn-Klauseln der Prädikatenlogik und  $(M_n \rightarrow \perp, \sigma_n)_{0 \leq n \leq m}$  eine SLD-Resolution aus  $\Gamma \cup \{M_0 \rightarrow \perp\}$  mit  $M_m = \emptyset$ . Dann gilt  $\Gamma \Vdash \Psi\sigma_0\sigma_1\dots\sigma_m$  für alle  $\Psi \in M_0$ .

**Lemma** Sei  $\Gamma$  eine Menge von definiten gleichungsfreien Horn-Klauseln der Prädikatenlogik, sei  $M \rightarrow \perp$  eine gleichungsfreie Horn-Klausel und sei  $\nu$  Substitution, so dass  $M\nu$  variabelnlos ist und  $\Gamma \Vdash M\nu$  gilt. Dann existieren eine prädikatenlogische SLD-Resolution  $((M_n \rightarrow \perp, \sigma_n))_{0 \leq n \leq m}$  und eine Substitution  $\tau$  mit  $M_0 = M, M_m = \emptyset$  und  $M_0\sigma_0\sigma_1\dots\sigma_m\tau = M\nu$ .

**Satz** Sei  $\Gamma$  eine Menge von definiten gleichungsfreien Horn-Klauseln der Prädikatenlogik, sei  $M \rightarrow \perp$  eine gleichungsfreie Horn-Klausel und sei  $\nu$  Substitution, so dass  $M\nu$  variabelnlos ist. Dann sind äquivalent:

- $\Gamma \Vdash M\nu$
- Es existieren eine SLD-Resolution  $((M_n \rightarrow \perp, \sigma_n))_{0 \leq n \leq m}$  aus  $\Gamma \cup \{M\nu \rightarrow \perp\}$  und eine Substitution  $\tau$  mit  $M_0 = M, M_m = \emptyset$  und  $M_0\sigma_0\sigma_1\dots\sigma_m\tau = M\nu$ .

### Zusammenfassung Prädikatenlogik

- Das natürliche Schließen formalisiert die "üblichen" Argumente in mathematischen Beweisen.
- Das natürliche Schließen ist vollständig und korrekt.
- Die Menge der allgemeingültigen Formeln ist semi-entscheidbar, aber nicht entscheidbar.
- Die Menge der Aussagen, die in  $(\mathbb{N}, +, *, 0, 1)$  gelten, ist nicht semi-entscheidbar.
- Die SLD-Resolution ist ein praktikables Verfahren, um die Menge der "Lösungen"  $(s_1, \dots, s_n)$  von  $\Gamma \vdash \psi(s_1, \dots, s_n)$  zu bestimmen (wobei  $\Gamma$  Menge von gleichungsfreien Horn-Klauseln und  $\psi$  Konjunktion von gleichungsfreien Atomformeln sind).

### Logische Programmierung

Grundidee (nach G.W. Leibniz)

1. lingua characteristica - Wissensdarstellungssprache
2. calculus ratiocinator - Wissensverarbeitungskalkül

### PROLOG - ein Folgerungstool

Sei  $M$  eine Menge von Aussagen,  $H$  eine Hypothese.  $H$  folgt aus  $M (M \models H)$ , falls jede Interpretation, die zugleich alle Elemente aus  $M$  wahr macht (jedes Modell von  $M$ ), auch  $H$  wahr macht.

#### Aussagen in PROLOG: HORN-Klauseln des PK1

$$\forall X_1 \dots \forall X_n (A(X_1, \dots, X_n) \leftarrow \bigwedge_{i=1}^m A_i(X_1, \dots, X_n))$$

Klauselkopf                      Klauselkörper

1. Regeln (vollständige HORN-Klauseln)  
 $\forall X_1 \dots \forall X_n (A(X_1, \dots, X_n) \leftarrow \bigwedge_{i=1}^n A_i(X_1, \dots, X_n))$
2. Fakten (HORN-Klauseln mit leerem Klauselkörper)  
 $\forall X_1 \dots \forall X_n (A(X_1, \dots, X_n) \leftarrow true)$
3. Fragen (HORN-Klauseln mit leerem Klauselkopf)  
 $\forall X_1 \dots \forall X_n (false \leftarrow \bigwedge_{i=1}^n A_i(X_1, \dots, X_n))$
4. leere HORN-Klauseln (mit leeren Kopf & leerem Körper)  
 $false \leftarrow true$

Effekte der Beschränkung auf HORN-Logik

1. Über HORN-Klauseln gibt es ein korrektes und vollständiges Ableitungsverfahren.  $\{K_1, \dots, K_n\} \models H$ , gdw.  $\{K_1, \dots, K_n\} \vdash_{ROB} H$
2. Die Suche nach einer Folge von Resolutionsschritten ist algorithmisierbar.
  - Das Verfahren "Tiefensuche mit Backtrack" sucht systematisch eine Folge, die zur leeren Klausel führt.
  - Rekursive und/oder metalogische Prädikate stellen dabei die Vollständigkeit in Frage.
3. Eine Menge von HORN-Klauseln mit nichtleeren Klauselköpfen ist stets erfüllbar; es lassen sich keine Widersprüche formulieren.  
 $K_1 \wedge K_2 \dots \wedge K_n \neq false$

Die systematische Erzeugung von (HORN-) Klauseln

1. Verneinungstechnischen Normalform (VTNF):  $\neg$  steht nur vor Atomformeln  $\neg \neg A \equiv A$
2. Erzeugung der Pränexen Normalform (PNF):  $\forall, \exists$  stehen vor dem Gesamtausdruck
  - $\forall X A(X) \rightarrow B \equiv \exists X (A(X) \rightarrow B)$
  - $\exists X A(X) \rightarrow B \equiv \forall X (A(X) \rightarrow B)$
3. Erzeugung der SKOLEM'schen Normalform (SNF):  $\exists$  wird eliminiert
4. Erzeugung der Konjunktiven Normalform (KNF): Durch systematische Anwendung des Distributivgesetzes  
 $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$

5. Erzeugung der Klauselform (KF): Jede der Elementardisjunktionen  $(L_i^j \vee \dots \vee L_1^{j_k})$  der KNF kann man als äquivalente Implikation (Klausel)  
 $(L_i^j \vee \dots \vee L_1^{j_m}) \leftarrow (L_i^{j_{m+1}} \wedge \dots \wedge L_1^{j_k})$  notieren, indem man alle positiven Literale  $L_i^j, \dots, L_i^{j_m}$  disjunktiv verknüpft in den DANN-Teil (Klauselkopf) und alle negativen Literale  $L_i^{j_{m+1}}, \dots, L_i^{j_k}$  konjunktiv verknüpft in den WENN-Teil (Klauselkörper) notiert.
6. In dem Spezialfall, dass alle Klauselköpfe dabei aus genau einem Literal bestehen, war die systematische Erzeugung von HORN-Klauseln erfolgreich; anderenfalls gelingt sie auch nicht durch andere Verfahren.

### Inferenz in PROLOG: Resolution nach ROBINSON

**Satz von ROBINSON**  $M' \equiv \bigwedge_{i=1}^n K_i \wedge \neg H$  ist kontradiktorisch (kt  $M'$ ), gdw. durch wiederholte Resolutionen in endlich vielen Schritten die negierte Hypothese  $\neg H \equiv false \leftarrow H$  durch die leere Klausel  $false \leftarrow true$  ersetzt werden kann.

Substitution

- Eine Substitution  $\nu$  einer Atomformel  $A$  ist eine Abbildung der Menge der in  $A$  vorkommenden Variablen  $X$  in die Menge der Terme
- Sie kann als Menge von Paaren  $[Variable, Ersetzung]$  notiert werden:  $\nu = \{[x, t] : x \in X, t = \nu(x)\}$
- Für strukturierte Terme wird die Substitution auf deren Komponenten angewandt:  $\nu(f(t_1, \dots, t_n)) = f(\nu(t_1), \dots, \nu(t_n))$
- Verkettungsoperator  $\circ$  für Substitutionen drückt Hintereinander-anwendung aus:  $\sigma \circ \nu(t) = \sigma(\nu(t))$
- Substitutionen, die zwei Terme syntaktisch identisch machen, heißen Unifikator:  $\nu$  unifiziert zwei Atomformeln  $s$  und  $t$ , falls dessen Einsetzung  $s$  und  $t$  syntaktisch identisch macht.

**Unifikation** Zwei Atomformeln  $p(t_{11}, \dots, t_{1n})$  und  $p_2(t_{21}, \dots, t_{2n})$  sind unifizierbar, gdw.

- sie die gleichen Prädikatsymbole aufweisen ( $p_1 = p_2$ ),
- sie die gleichen Stelligkeiten aufweisen ( $n = m$ ) und
- die Terme  $t_{1i}$  und  $t_{2i}$  jeweils miteinander unifizierbar sind.

Die Unifizierbarkeit zweier Terme richtet sich nach deren Sorte:

1. Zwei Konstanten  $t_1$  und  $t_2$  sind unifizierbar, gdw.  $t_1 = t_2$
2. Zwei strukturierte Terme  $f(t_{11}, \dots, t_{1n})$  und  $f(t_{21}, \dots, t_{2n})$  sind unifizierbar, gdw.
  - sie die gleichen Funktionssymbole aufweisen ( $f_1 = f_2$ ),
  - sie die gleichen Stelligkeiten aufweisen ( $n = m$ ) und
  - die Terme  $t_{1i}$  und  $t_{2i}$  jeweils miteinander unifizierbar sind.
3. Eine Variable  $t_1$  ist mit einer Konstanten oder einem strukturierten Term  $t_2$  unifizierbar.  $t_1$  wird durch  $t_2$  ersetzt (instanziiert):  $t_1 := t_2$
4. Zwei Variablen  $t_1$  und  $t_2$  sind unifizierbar und werden gleichgesetzt:  $t_1 := t_2$  bzw.  $t_2 := t_1$

**Allgemeinster Unifikator** Eine Substitution heißt allgemeinster Unifikator zweier Terme  $s$  und  $t$  ( $\nu = m.g.u.(s, t)$ ), gdw.

1. die Substitution  $\nu$  ein Unifikator von  $s$  und  $t$  ist und
2. für jeden anderen Unifikator  $\sigma$  von  $s$  und  $t$  eine nichtleere und nicht identische Substitution  $\tau$  existiert, so dass  $\sigma = \tau \circ \nu$  ist.

Algorithmus zur Bestimmung des allgemeinsten Unifikators 2er Terme

```
input: s, t
output: Unifizierbarkeitsaussage, ggf. $nu = m.g.u.( s, t )
$!: = 0; $nu_i := \varnothing; $s_i := s; $t_i := t$
Start: $s_i$ und $t_i$ identisch?
ja $ \rightarrow$ s und t sind unifizierbar, $nu = $nu_i = m.g.u.(s, t)
```

```
nein $ \rightarrow$ Bilde die Unterscheidungsterme $s_i^*$ und $t_i^*$
$s_i^*$ oder $t_i^*$ Variable?
nein $ \rightarrow$ $$ und $t$ sind nicht unifizierbar (
ja $ \rightarrow$ sei (o.B.d.A.) $s_i^*$ eine Variable
$s_i^* \subseteq t_i^*$? (enthaltet $t_i^*$ die Variabl
ja $ \rightarrow$ $$ und $t$ sind nicht unifizierbar
nein $ \rightarrow$
$nu := {[s, t]: [s, t] \in nu, t' := t |_{s*} \rightarrow$
$s' := nu'(s_i); t' := nu'(t_i); i := i+1; $
\nu_i := \nu'; s_i := s'; t_i := t'; $
gehe zu Start
```

### Logische Programmierung

"deskriptives" Programmierparadigma =

1. Problembeschreibung

- Die Aussagenmenge  $M = \{K_1, \dots, K_n\}$ , über denen gefolgert wird, wird in Form von Fakten und Regeln im PK1 notiert.
- Eine mutmaßliche Folgerung (Hypothese)  $H$  wird in Form einer Frage als negierte Hypothese hinzugefügt.

2. (+) Programmverarbeitung

- Auf der Suche eines Beweises für  $M \models H$  werden durch zugesteuerte Prozedur-Aufrufe Resolutions-Schritte zusammengestellt.
- Dem "Programmierer" werden (begrenzte) Möglichkeiten gegeben, die systematische Suche zu beeinflussen.

### Syntax

BACKUS-NAUR-Form

- PROLOG-Programm ::= Wissensbasis Hypothese
- Wissensbasis ::= Klausel — Klausel Wissensbasis
- Klausel ::= Fakt — Regel
- Fakt ::= Atomformel.
- Atomformel ::= Prädikatsymbol (Termfolge)
- Prädikatsymbol ::= Name
- Name ::= Kleinbuchstabe — Kleinbuchstabe Restname — "Zeichenfolge" — Sonderzeichenfolge
- RestName ::= Kleinbuchstabe — Ziffer — \_ — Kleinbuchstabe RestName — Ziffer RestName — \_ RestName
- ...

Was muss der Programmierer tun?

- Formulierung einer Menge von Fakten und Regeln
- Formulierung einer negierten Hypothese (Frage, Ziel)

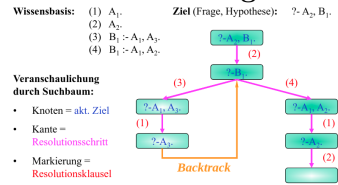
Was darf der Programmierer erwarten? Dass das "Deduktionstool" PROLOG  $M \models H$  zu zeigen versucht, indem systematisch die Resolutionsmethode auf  $\neg H$  und eine der Klauseln aus  $M$  angewandt wird, solange bis  $\neg H \equiv false \leftarrow true$  entsteht

**Formulierung von Wissensbasen** Visual Prolog benötigt aber einen Deklarationsteil für Datentypen und Aritäten der Prädikate

```
domains
person = symbol
predicates
vater_von(person, person)
mutter_von(person, person).
grossvater_von(person, person)
geschwister(person, person)
clauses
< Wissensbasis einfüegen >
// Klauseln gleichen Kopfraedikates gruppieren
goal
< Frage ohne '?'- ' einfüegen >
```

### Verarbeitung Logischer Programme

### Veranschaulichung ohne Unifikation



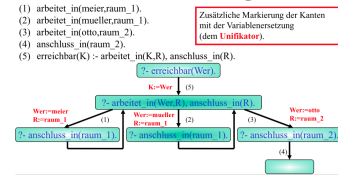
Tiefensuche mit Backtrack: Es werden anwendbare Klauseln für das erste Teilziel gesucht. Gibt es ...

- ... genau eine, so wird das 1. Teilziel durch deren Körper ersetzt.
- ... mehrere, so wird das aktuelle Ziel inklusive alternativ anwendbarer Klauseln im Backtrack-Keller abgelegt und die am weitesten oben stehende Klausel angewandt.
- ... keine (mehr), so wird mit dem auf dem Backtrack-Keller liegendem Ziel die Bearbeitung fortgesetzt.

Dies geschieht solange, bis

- das aktuelle Ziel leer ist oder
- keine Klausel (mehr) anwendbar ist und der Backtrack-Keller leer ist.

### Veranschaulichung mit Unifikation



Zusätzliche Markierung der Kanten mit der Variablensubstitution (dem Unifikator).

### PROLOG aus prozeduraler Sicht

**Deklarative Interpretation** In einem Objektbereich  $I = \{\text{mueller, mayer, schulze, ...}\}$  bildet das Prädikat  $\text{weisungsrecht}(X, Y)$   $[X, Y]$  auf wahr ab, gdw.

- das Prädikat  $\text{chef_von}(X, Y)$  das Paar  $[X, Y]$  auf wahr abbildet oder
- es ein  $Z \in I$  gibt, so dass
  - das Prädikat  $\text{chef_von}(X, Z)$  das Paar  $[X, Z]$  auf wahr abbildet und
  - das Prädikat  $\text{weisungsrecht}(Z, Y)$  das Paar  $[Z, Y]$  auf wahr abbildet.

**Prozedurale Interpretation** Die Prozedur  $\text{weisungsrecht}(X, Y)$  wird abgearbeitet, indem

1. die Unterprozedur  $\text{chef_von}(X, Y)$  abgearbeitet wird. Im Erfolgsfall ist die Abarbeitung beendet; anderenfalls werden
2. die Unterprozeduren  $\text{chef_von}(X, Z)$  und  $\text{weisungsrecht}(Z, Y)$  abgearbeitet; indem systematisch Prozedurvarianten beider Unterprozeduren aufgerufen werden. Dies geschieht bis zum Erfolgsfall oder erfolgloser erschöpfender Suche.

Prädikate zur Steuerung der Suche nach einer Folge von Resolutionschritten!

Das Prädikat  $!/0$  ist stets wahr. In Klauselkörpern eingefügt verhindert es ein Backtrack der hinter  $!/0$  stehenden Teilziele zu den vor  $!/0$  stehenden Teilzielen sowie zu alternativen Klauseln des gleichen Kopfprädikats. Die Verarbeitung von  $!/0$  schneidet demnach alle vor der Verarbeitung verbliebenen Lösungswege betreffenden Prozedur ab.

### Listen und rekursive Problemlösungsstrategien

Listen

1.  $[]$  ist eine Liste.
2. Wenn  $T$  ein Term und  $L$  eine Liste ist, dann ist
  - (a)  $[T|L]$  eine Liste.
  - (b)  $T.L$  eine Liste. (ungebräuchlich)
  - (c)  $.(T, L)$  eine Liste. (ungebräuchlich)
  - (d) Das erste Element  $T$  heißt Listenkopf,  $L$  heißt Listenkörper oder Restliste.
3. Wenn  $t_1, \dots, t_n$  Terme sind, so ist  $[t_1, \dots, t_n]$  eine Liste.
4. Weitere Notationsformen von Listen gibt es nicht.

### Rekursion

Eine Prozedur heißt rekursiv, wenn in mindestens einem der Klauselkörper ihrer Klauseln ein erneuter Aufruf des Kopfprädikates erfolgt. Ist der Selbstaufzuruf die letzte Atomformel des Klauselkörpers der letzten Klausel dieser Prozedur - bzw. wird er es durch vorheriges "Abschneiden" nachfolgender Klauseln mit dem Prädikat  $!/0$  -, so spricht man von Rechtsrekursion; anderenfalls von Linksrekursion. Eine Prozedur heißt indirekt rekursiv, wenn bei der Abarbeitung ihres Aufrufes ein erneuter Aufruf derselben Prozedur erfolgt.

### Unifikation 2er Listen

1. Zwei leere Listen sind miteinander unifizierbar.
2. Zwei nichtleere Listen  $[K_1|R_1]$  und  $[K_2|R_2]$  sind miteinander unifizierbar, wenn ihre Köpfe ( $K_1$  und  $K_2$ ) und ihre Restlisten ( $R_1$  und  $R_2$ ) jeweils miteinander unifizierbar sind.
3. Eine Liste  $L$  und eine Variable  $X$  sind miteinander unifizierbar, wenn die Variable selbst nicht in der Liste enthalten ist. Die Variable  $X$  wird bei erfolgreicher Unifikation mit der Liste  $L$  instanziiert:  $X := L$ .

### Differenzlisten

1. Die Differenz aus einer leeren Liste und einer (beliebigen) Liste ist die leere Liste:  $[] - L = []$
2. Die Differenz aus einer Liste  $[E|R]$  und der Liste  $L$ , welche  $E$  enthält, ist die Liste  $D$ , wenn die Differenz aus  $R$  und  $L$  (abzögl.  $E$ ) die Liste  $D$  ist:  $[E|R] - L = D$ , wenn  $E \in L$  und  $R - (L - [E]) = D$
3. Die Differenz aus einer Liste  $[E|R]$  und einer Liste  $L$ , welche  $E$  nicht enthält, ist die Liste  $[E|D]$ , wenn die Differenz aus  $R$  und  $L$  die Liste  $D$  ist:  $[E|R] - L = [E|D]$ , wenn  $E \in L$  und  $R - L = D$

### Prolog-Fallen

- Nicht terminierende Programme
- Alternierende Zielklauseln: Ein aktuelles Ziel wiederholt sich und die Suche nach einer Folge von Resolutionschritten endet nie oder die Suche nach Resolutionschritten endet mit einem Überlauf des Backtrack-Kellers
- Expandierende Zielklauseln: Das erste Teilziel wird in jeden Resolutionschritt durch mehrere neue Teilziele ersetzt; die Suche endet mit einem Speicherüberlauf
- Metalogische Prädikate und konstruktive Lösungen: Das Prädikat  $\text{not}/1$  hat eine Aussage als Argument und ist somit eine Aussage über eine Aussage, also metalogisch.

### Rekursive Problemlösungsstrategien

**Botschaft 1** Man muss ein Problem nicht in allen Ebenen überblicken, um eine Lösungsverfahren zu programmieren. Es genügt die Einsicht, wie man aus der Lösung eines einfacheren Problems die Lösung des präsenten Problems macht und

1. wie man aus der Lösung eines einfacheren Problems die Lösung des präsenten Problems macht und
2. wie es im Trivialfall zu lösen ist.

**Botschaft 2** Wann immer man Objekte mit Mustern vergleicht, z.B. eine Struktur durch "Auflegen von Schablonen" identifiziert, Gemeinsamkeiten mehrerer Objekte identifiziert, d.h. "eine Schablone entwirft" oder "gemeinsame Beispiele für mehrere Schablonen" sucht, mache man sich den Unifikations-Mechanismus zu nutzen.

**Botschaft 3** Es ist mitunter leichter (oder überhaupt erst möglich), für komplexe Probleme eine potentielle Lösung zu "erraten" und dazu ein Verfahren zu entwickeln, welches diese Lösung auf Korrektheit testet,

als zielgerichtet die korrekte Lösung zu entwerfen. Hierbei kann man den Backtrack-Mechanismus nutzen.

**Botschaft 4** Heuristiken sind eine Chance, auch solche Probleme einer Lösung zuzuführen, für die man keinen (determinierten) Lösungsalgorithmus kennt und das klassische Einsatzgebiet zahlreicher KI-Tools - auch der Logischen Programmierung.

**Botschaft 5**

1. Für die systematische Suche eines Pfades kann der Suchprozess einer Folge von Resolutionschritten genutzt werden. Man muss den Suchprozess nicht selbst programmieren.
2. Für eine heuristische Suche eines Pfades gilt Botschaft 4: Sie ist das klassische Einsatzgebiet zahlreicher KI-Tools - auch der Logischen Programmierung.

**Botschaft 6**

1. "Logeleien" sind oft Aussagen über Belegungen von Variablen mit endlichem Wertebereich, ergänzt um eine Frage zu einem nicht explizit gegebenen Wert.
2. Dabei handelt es sich um Gründe um eine Deduktionsaufgabe mit einer Hypothese zu einem mutmaßlichen Wert der gesuchten Variablen. Deshalb ist es oft auch mit dem "Deduktionstool" Prolog lösbar, denn Prolog tut im Grunde nichts anderes als ein zielgerichtetes "Durchprobieren" legitimer Deduktionsschritte im "Generate - and - Test" - Verfahren.

**Botschaft 7** Auch in der formalen Logik gibt es Deduktionsaufgaben, bei der Variablenbelegungen gesucht sind, welche eine Aussage wahr machen:

1. Meist geschieht das durch systematische Auswertung der Aussage, wozu das Suchverfahren von Prolog genutzt werden kann.
2. Auch hier geht es oft um gesuchte Werte für Variablen. Deshalb ist es oft auch mit dem "Deduktionstool" Prolog lösbar, denn Prolog tut im Grunde nichts anderes als ein zielgerichtetes "Durchprobieren" legitimer Deduktionsschritte im "Generate - and - Test" - Verfahren.