4 Prädikatenlogik

Formale Grundlagen der Informatik I

Herbstsemester 2012

Robert Marti

Vorlesung teilweise basierend auf Unterlagen
von Prof. emer. Helmut Schauer

Prädikatenlogik als Erweiterung der Aussagenlogik

Motivation: Einfachere Schreibweisen, z.B. im Russian Spy Problem

There are three persons: Stirlitz, Mueller, and Eismann. It is known that exactly one of them is Russian, while the other two are Germans.

Aussagenlogik: $(RS \wedge GM \wedge GE) \vee (GS \wedge RM \wedge GE) \vee (GS \wedge GM \wedge RE)$

Prädikatenlogik: $(R(S) \wedge G(M) \wedge G(E)) \vee (G(S) \wedge R(M) \wedge G(E)) \vee (G(S) \wedge G(M) \wedge R(E))$

Objekt

Prädikat

Moreover, every Russian must be a spy.

Aussagenlogik: $(RS \rightarrow SS) \wedge (RM \rightarrow SM) \wedge (RE \rightarrow SE)$

Prädikatenlogik: $\forall x \; (R(x) \rightarrow S(x))$

Variable

Quantor

Prädikatenlogik als Erweiterung der Aussagenlogik

Motivation

- Einfachere Schreibweisen, z.B.

10 is greater than 5 $GreaterThan(10, 5)$ bzw. $>(10, 5)$ bzw. $10 > 5$

11 is greater than 5 $GreaterThan(11, 5)$ bzw. $>(11, 5)$ bzw. $11 > 5$

…

4 is greater than 5 $GreaterThan(4, 5)$ bzw. $>(4, 5)$ bzw. $4 > 5$

Wahrheitswert: **False**

- Verallgemeinerung

$\boldsymbol{x}$ is greater than 5 $GreaterThan(x, 5)$ bzw. $>(x, 5)$ bzw. $x > 5$

Syntax prädikatenlogischer Formeln (1)

- **Logische Symbole**:
  $\vee$, $\wedge$, $\neg$, $\exists$, $\forall$, $\rightarrow$, $\leftarrow$, $\leftrightarrow$, Klammern, Komma, Doppelpunkt

- **Bezeichner** (identifier):
  frei wählbare nicht-logische Symbole zur Bezeichnung von
  - Objekten ("Konstanten") — *Elisabeth_2* *Philipp* *Charles* *William* 2 4
  - Variablen ("Platzhalter für Objekte") — x y z
  - Funktionen — *Father_Of* *Wife_Of* $+$ $*$
  - Prädikaten — *Parent* *Ancestor* $=$ $\leq$

- **Terme**:
  - Objekte
  - Variablen
  - "strukturierte" Terme der Form
    $\varphi(\tau_1, \dots, \tau_n)$, wobei
    φ ein Funktionssymbol,
    τ_i Terme — *Wife_Of*(*Charles*) *Father_Of*(x)
    $+(x, 1)$ [meist in Infix-Notation: $x+1$]

Syntax prädikatenlogischer Formeln (2)

- **Atome** (auch: atomare Formeln, Primformeln) der Form $\pi(\tau_1, \ldots, \tau_n)$, wobei
  π ein Prädikatensymbol,
  τ_i Terme

  $Ancestor(Father\_Of(Elisabeth\_2), William)$

  $\geq (x, 0)$ [meist in Infix-Notation: $x \geq 0$]

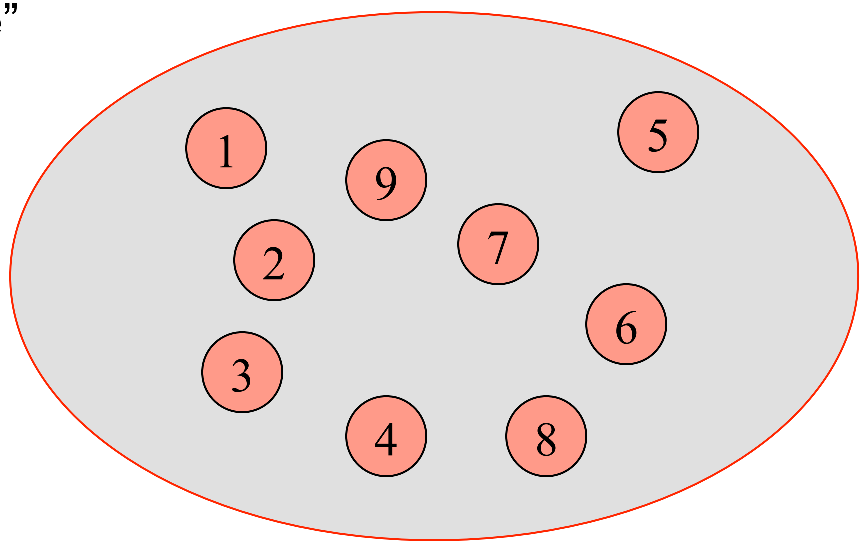
- **Formeln**:
  - Atome
  - $(\phi \vee \psi)$, $(\phi \wedge \psi)$, $(\neg\, \phi)$,
  - $(\exists x\, \phi)$, $(\forall x\, \phi)$,
  - $(\phi \rightarrow \psi)$, $(\phi \leftarrow \psi)$, $(\phi \leftrightarrow \psi)$,

  wobei ϕ und ψ Formeln und x eine Variable

- Bemerkungen:
  - $\exists$ heisst Existenzquantor ("lineare" Notation: **exists**, manchmal auch **some**)
  - $\forall$ heisst Allquantor ("lineare" Notation: **all**, manchmal auch **forall**)
  - oft wird auch $(\exists x : \phi)$, $(\forall x : \phi)$ geschrieben
    oder es wird gar ein Bereich (domain, universe) U bzw. $U(x)$ für die Variable x angegeben:
    $(\exists x : U(x) : \phi)$, $(\forall x : U(x) : \phi)$ z.B. $(\forall i : 1 \leq i \leq n : a_i \geq 0)$

Semantik prädikatenlogischer Formeln mit Quantoren

"Universe of Discourse"
("Universum")
nicht notwendigerweise
explizit angegeben



bzw:
$U(x) := \{ x \mid 1 \leq x \leq 9 \}$

Semantik anhand von Beispielen:

$\exists x \; Prime(x)$

- heisst „**es existiert** im gegebenen Universum ein Objekt x, für das gilt: x ist prim."
- ist eine Abkürzung für $Prime(1) \vee Prime(2) \vee \ldots \vee Prime(9)$ (im gegebenen Universum)

$\forall x \; Digit(x)$

- heisst „**für alle** Objekte x im gegebenen Universum gilt: x ist eine Ziffer."
- ist eine Abkürzung für $Digit(1) \wedge Digit(2) \wedge \ldots \wedge Digit(9)$ (im gegebenen Universum)

Beispiele von Formeln in Prädikatenlogik (1. Stufe)

$Loves(Romeo, Juliet)$

$\forall x\ Ancestor(Adam, x)$

$\forall x\ (\ Person(x) \rightarrow Ancestor(Adam, x)\ )$

$Is\_Mother(Eve) \rightarrow \exists y\ (Is\_Child(y) \land Mother(Eve, y)\ )$

$\forall x\ (\ Is\_Mother(x) \rightarrow \exists y\ (\ Is\_Child(y) \land Mother(x, y)\ )\ )$

$\forall sub\ \forall s\ (\ Subset(sub, s) \leftarrow \forall x\ (\ Member(x, sub) \rightarrow Member(x, s)\ )\ )$

$\forall sub\ \forall s\ (\ sub \subseteq s \leftarrow \forall x\ (\ x \in sub \rightarrow x \in s\ )\ )$ -- Infix Notation mit üblichen Symbolen

$\exists x\ (\ Rich(x) \land \neg\ Rich(x)\ )$

nicht in Prädikatenlogik 1. Stufe ist z.B.

$\forall \mathbf{r}\ (\ Transitive(\mathbf{r}) \leftarrow \forall x\ \forall y\ \forall z\ (\ \mathbf{r}(x, y) \land \mathbf{r}(y, z) \rightarrow \mathbf{r}(x, z)\ )\ )$

wohl aber die folgende “**reifizierte**” Darstellung

$\forall \mathbf{r}\ (\ Transitive(\mathbf{r}) \leftarrow \forall x\ \forall y\ \forall z\ (\mathbf{Rel}(\mathbf{r}, x, y) \land \mathbf{Rel}(\mathbf{r}, y, z) \rightarrow \mathbf{Rel}(\mathbf{r}, x, z))\ )$

Einige bekannte Gesetze (vgl. auch Gesetze für Aussagenlogik)

Seien

F, G und H prädikatenlogische Formeln (Well-Formed Formulas)

Kommutative Gesetze

$F \wedge G = G \wedge F$ $\qquad$ $F \vee G = G \vee F$

Assoziative Gesetze

$F \wedge (G \wedge H) = (F \wedge G) \wedge H$ $\qquad$ $F \vee (G \vee H) = (F \vee G) \vee H$

Distributive Gesetze

$F \wedge (G \vee H) = (F \wedge G) \vee (F \wedge H)$ $\qquad$ $F \vee (G \wedge H) = (F \vee G) \wedge (F \vee H)$

Weitere Gesetze (vgl. auch Gesetze für Aussagenlogik)

Seien

F, G und H prädikatenlogische Formeln (Well-Formed Formulas)

x eine Variable

Gesetz der doppelten Negation

$$\neg(\neg F) = F$$

De Morgan'sche Gesetze

$$\neg(F \wedge G) = \neg F \vee \neg G$$

$$\neg(F \vee G) = \neg F \wedge \neg G$$

$$\neg(\forall x\, F) = (\exists x\, \neg F)$$

$$\neg(\exists x\, F) = (\forall x\, \neg F)$$

Beweis der Äquivalenz prädikatenlogischer Formeln

Wie werden solche Gesetze in der Prädikatenlogik bewiesen?

Grundprinzip:

In Formeln mit Variablen müssen diese durch alle Objekte des Universums belegt werden. (Dies sind potentiell unendlich viele Objekte!)

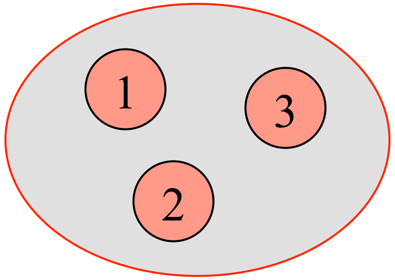
Bem: Wenn das Universum nicht explizit gegeben ist, dann werden die aus den in den Formeln verwendeten Objekt- und Funktionssymbolen "konstruierbaren" Terme ohne Variablen betrachtet (sog. Herbrand Universum).

Für jede **atomare** Formel muss bekannt sein, mit welchen Variablenbelegungen die atomare Formel wahr bzw. falsch wird.

Wenn für alle möglichen Variablenbelegungen die linken und rechten Seiten der Äquivalenzen das gleiche Resultat ergeben, dann ist die Äquivalenz bewiesen.

Kleines Beispiel zum Beweis von Äquivalenzen

Universum = $\{ 1, 2, 3 \}$



$P(2), P(3) = \mathbf{T}$ $\quad$ $P(1) = \mathbf{F}$

$Q(1), Q(2), Q(3) = \mathbf{T}$

Gilt $\neg (\forall x\, F) = (\exists x\, \neg F)$? (hier für P und Q ?)

für P

linke Seite: $\neg (\forall x\, P(x)) = \neg (P(1) \wedge P(2) \wedge P(3)) = \neg (\mathbf{F} \wedge \mathbf{T} \wedge \mathbf{T}) = \neg \mathbf{F} = \mathbf{T}$

rechte Seite: $\exists x\, \neg P(x) = \neg P(1) \vee \neg P(2) \vee \neg P(3) = \neg \mathbf{F} \vee \neg \mathbf{T} \vee \neg \mathbf{T} = \mathbf{T} \vee \mathbf{F} \vee \mathbf{F} = \mathbf{T}$

für Q

linke Seite: $\neg (\forall x\, Q(x)) = \neg (Q(1) \wedge Q(2) \wedge Q(3)) = \neg (\mathbf{T} \wedge \mathbf{T} \wedge \mathbf{T}) = \neg \mathbf{T} = \mathbf{F}$

rechte Seite: $\exists x\, \neg Q(x) = \neg Q(1) \vee \neg Q(2) \vee \neg Q(3) = \neg \mathbf{T} \vee \neg \mathbf{T} \vee \neg \mathbf{T} = \mathbf{F} \vee \mathbf{F} \vee \mathbf{F} = \mathbf{F}$

Gültigkeitsbereich von Quantoren

In den (Teil-) Formeln $(\exists x\ F)$ bzw. $(\forall x\ F)$ etabliert der Quantor einen **Gültigkeitsbereich** (scope) für die quantifizierte Variable x
(ähnlich wie dies Funktionen/Prozeduren in höheren Programmiersprachen für ihre Parameter tun).

Beispiel:

$$(\ \forall z\ (\ Member(z, b) \rightarrow Member(z, a)\ )\ ) \rightarrow Subset(b, a)$$

Der Gültigkeitsbereich des Allquantors $\forall$ ist rot markiert.

Definition: Das Vorkommen einer Variable x in einer (Teil-) Formel F heisst **gebunden**, falls das Vorkommen dieser Variable x im Gültigkeitsbereich eines Quantors für x steht. Andernfalls heisst das Vorkommen der Variable x **frei**.

Beispiel:

$$(\ \forall z\ (\ Member(z, b) \rightarrow Member(z, a)\ )\ ) \rightarrow Subset(b, a)$$

In dieser Formel sind alle Vorkommen von z gebunden, und alle Vorkommen von a und b sind frei.

Umbenennung quantifizierter Variablen

In den (Teil-) Formeln $(\exists x\ F)$ bzw. $(\forall x\ F)$ kann die quantifizierte Variable x innerhalb von F umbenannt werden, sofern der neu gewählte Name (z.B. y) nicht bereits in der Teilformel F vorkommt.

Beispiel:

In der Formel

$$\forall z\ (\ Member(z, b) \ \rightarrow\ Member(z, a)\ ) \ \rightarrow\ Subset(b, a)$$

ist die Umbenennung von z zu w erlaubt:

$$\forall w\ (\ Member(w, b) \ \rightarrow\ Member(w, a)\ ) \ \rightarrow\ Subset(b, a)$$

nicht aber z.B. die Umbenennung von z zu b:

$$\forall b\ (\ Member(b, b) \ \rightarrow\ Member(b, a)\ ) \ \rightarrow\ Subset(b, a)$$

denn die zweite Aussage ist restriktiver …

($Member(b, b)$ fordert, dass beide Argumente gleich sein müssen!)

Weitere Rechenregeln für Quantoren

Seien P, Q Prädikate. Dann gelten folgende Regeln:

$$\forall x\,(P(x) \wedge Q(x)) = (\forall x\,P(x)) \wedge (\forall x\,Q(x)) = (\forall x\,P(x)) \wedge (\forall y\,Q(y))$$

$$\exists x\,(P(x) \vee Q(x)) = (\exists x\,P(x)) \vee (\exists x\,Q(x)) = (\exists x\,P(x)) \vee (\exists y\,Q(y))$$

$$(\forall x\,P(x)) \vee (\forall x\,Q(x)) \rightarrow \forall x\,(P(x) \vee Q(x))$$

$$\exists x\,(P(x) \wedge Q(x)) \rightarrow (\exists x\,P(x)) \wedge (\exists x\,Q(x))$$

Diese Regeln gelten auch für Prädikate mit mehreren Argumenten, z.B.

$$\forall x\,(P(x,y) \wedge Q(z,x,w)) = (\forall x\,P(x,y)) \wedge (\forall x\,Q(z,x,w))$$

Vorsicht beim Vertauschen von Quantoren!

$$\forall x\,(\forall y\,(P(x,y))) = \forall y\,(\forall x\,(P(x,y))) = \forall x\,\forall y\,P(x,y)$$ [oder gar: $\forall x, y\,P(x,y)$]

aber

$$\forall x\,(\exists y\,(P(x,y))) \neq \exists y\,(\forall x\,(P(x,y)))$$ [Beispiel: $P(x,y) := Loves(x,y)$]

Aussagen über Arrays (Beispiele)

Sei $\underline{a}$ ein Array (vgl. Vektor in Mathematik) der Länge N, mit Indices $0, 1, ..., N-1$

	a_0	a_1	a_2	a_3	…	a_{N-1}
$\underline{a}$	0	5	0	-3	…	22

Aussagen über Arrays (Beispiele, wobei die Aussagen auch **FALSE** sein können):

$\forall i : 0 \leq i < N : a_i = 0$	alle Elemente sind 0
$\forall i : 1 \leq i < N : a_{i-1} < a_i$	die Elemente sind streng monoton steigend
$\forall i, j : 0 \leq i < j < N : a_i < a_j$	die Elemente sind streng monoton steigend
$\forall i, j : 0 \leq i < j < N : a_i = a_j$	alle Elemente sind gleich
$\forall i : 0 \leq i < N : a_i \leq a_j$	das Element mit Index j ist das grösste
$\exists j: 0 \leq j < N: (\forall i: 0 \leq i < N: a_i \leq a_j)$	es existiert ein grösstes Element (mit Index j)

Einschub: Regeln und Definitionen Quantoren (→ Übungen)

Sei P ein (einstelliges) Prädikat, $U(x)$ ein Universum für x.

Dann gilt:

$\forall\, x : U(x) : P\,(x) \quad = \quad \textbf{TRUE}$ falls $U(x) = \emptyset$ ($\emptyset$ die leere Menge)

$\exists\, x : U(x) : P\,(x) \quad = \quad \textbf{FALSE}$ falls $U(x) = \emptyset$

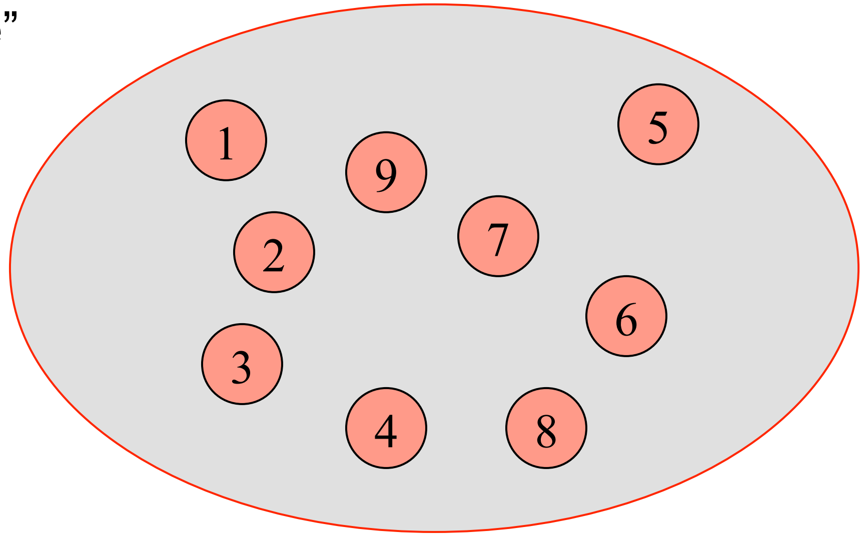
Definition des **Anz** Quantors (eigentlich eine Funktion, die eine Zahl liefert):

$\textbf{Anz}\, x : U(x) : P(x)$ die Anzahl von Objekten im Universum $U(x)$ für welche die Formel $P(x)$ **TRUE** ist

$\textbf{Anz}\, x : U(x) : P\,(x) \quad = \quad 0$ falls $U(x) = \emptyset$

Einschub: “Anzahl” Quantor (→ Übungen)

“Universe of Discourse”
(“Universum”)
nicht notwendigerweise
explizit angegeben



Semantik anhand eines Beispiels im obigen Universum:

$\mathbf{Anz}\ x\ \ Prime(x)$

- heisst „**Anzahl** der Objekte x im ‚Universum‘, für die gilt: x ist prim.“
- liefert eine Zahl, keinen Wahrheitswert! (Dies ist letztlich eine **Funktion** und in der Logik – im Gegensatz zu Programmiersprachen – nicht sehr gebräuchlich!)

Aussagen über Arrays (Beispiele, wobei die Aussagen auch $\textsc{False}$ sein können):

$\mathbf{Anz}\ i\ :\ 0 \leq i < N :\ a_i = 0$ Anzahl der Elemente, die 0 sind

Nebenbemerkung zur Notation

Die Notation für Bereiche werden i.a. in Programmiersprachen nicht unterstützt.

Beispiel:

$0 \leq i < N$ müsste ausformuliert werden: $0 \leq i \ \wedge \ i < N$

Die gewählte Notation für Arrays wäre in der Prädikatenlogik im Prinzip "illegal".

Beispiel:

$\forall i : 0 \leq i \ \wedge \ i < N : a_i = 0$ müsste z.B. wie folgt umgeschrieben werden

$\forall i : 0 \leq i \ \wedge \ i < N : Array(a) \ \wedge \ Element(x, a, i) \ \wedge \ x = 0$

bzw.

$\forall i : 0 \leq i \ \wedge \ i < N : Array(a) \ \wedge \ ElementAtPosition(a, i) = 0$

wobei:

$Array(a)$ ein Prädikat "a ist ein Array"

$Element(x, a, i)$ ein Prädikat "x ist Element von a an der Position i "

$ElementAtPosition(a, i)$ eine Funktion "Element von a an der Position i "

Ambiguität natürlichsprachlicher Aussagen

Ambiguität (ambiguity) = Mehrdeutigkeit.

Die Elemente eines Arrays a der Länge N sind entweder 0 oder 1:

$$\forall i : 0 \leq i < N : (a_i = 0 \vee a_i = 1)$$

oder

$$(\forall i : 0 \leq i < N : a_i = 0) \vee (\forall i : 0 \leq i < N : a_i = 1)$$

?

Die Elemente der Arrays a und b der Länge N sind gleich:

$$\forall i : 0 \leq i < N : a_i = b_i$$

oder

$$\forall i : 0 < i < N : (a_i = a_{i-1} \wedge b_i = b_{i-1})$$

oder

$$(\forall i : 0 < i < N : a_i = a_{i-1}) \wedge (\forall i : 0 \leq i < N : a_i = b_i)$$

oder

$$\forall i : 0 \leq i < N : (\exists j : 0 \leq j < N : a_i = b_j)$$

?

Natürlichsprachliche Aussagen in Prädikatenlogik

Wortart	Darstellung in Logik	Beispiel
Eigenname	Konstante	*Garfield*
(allgemeines) Substantiv	1-stelliges Prädikat	$Cat(who)$
Adjektiv	1-stelliges Prädikat	$Lazy(who)$
intransitives Verb	1-stelliges Prädikat	$Sleeps(who)$
transitives Verb	2-stelliges Prädikat	$Sees(who, what)$
bitransitives Verb	3-stelliges Prädikat	$Gives(who, what, to)$

"A $noun_{sg}$ $verb_{sg}$". $\exists x\ (\ noun_{sg}(x) \wedge verb_{sg}(x)\ )$

"All $noun_{pl}$ $verb_{pl}$". $\forall x\ (\ noun_{pl}(x) \rightarrow verb_{pl}(x)\ )$

Inferenzregeln und syntaktische Folgerung

Def.: Eine **Inferenzregel** (inference rule) gibt an, wie aus einer Menge bekannter Formeln eine „neue“ Formel hergeleitet – d.h. erzeugt und zu dieser Menge hinzugefügt – wird.

Beispiele von Inferenzregeln:

And Introduction	Modus Ponens [MP]	Universal Instantiation [UI]	
$\phi\,,\ \psi$	$\phi\,,\ \phi \rightarrow \psi$	$\forall x\ \phi$	bekannte Formel(n)
$\phi \wedge \psi$	ψ	$\phi\{x/\tau\}$	„neue“ Forrmel

Formel in der alle Vorkommen von Variable x in Formel ϕ durch Term τ ersetzt wurden

Eine Formel ψ heisst **syntaktische Folgerung** der Formelmenge $\Delta := \{\ \phi_1, \ldots, \phi_n\ \}$,

geschrieben $\Delta \vdash \psi$ bzw. $\{\ \phi_1, \ldots, \phi_n\ \} \vdash \psi$,

falls ψ durch eine **Folge von Anwendungen von Inferenzregeln** aus Δ hergeleitet werden kann.

Syntaktische Folgerung (Beispiel)

$$\{\ Human(Socrates),\ \forall x\ (\ Human(x) \rightarrow Fallible(x)\ )\ \} \vdash\ Fallible(Socrates)$$

$$\frac{\forall x\ (\ Human(x) \rightarrow Fallible(x)\ )}{Human(Socrates) \rightarrow Fallible(Socrates)}\quad [\ \mathrm{UI}\ ]$$

sofern *Socrates* im Universum vorkommt

$$\frac{Human(Socrates) \rightarrow Fallible(Socrates) \qquad Human(Socrates)}{Fallible(Socrates)}\quad [\ \mathrm{MP}\ ]$$

♦

Horn-Klauseln

- Formeln der Form

$\boxed{H} \leftarrow \boxed{B_1 \land \ldots \land B_n}$ $\quad n \geq 0,\ H, B_i\ (1 \leq i \leq n)$ Atome (auch **Regeln** genannt)

Kopf (head) Körper (body)

$\leftarrow\ G_1 \land \ldots \land G_m$ $\quad n \geq 1,\ G_j\ (1 \leq j \leq m)$ Atome (auch **Anfragen** genannt)

heissen **Horn-Klauseln** (Horn clauses, definite clauses), wobei alle vorkommenden Variablen implizit allquantifiziert sind.

- Wenn der Körper der Regel fehlt ($n = 0$) wird die Formel als **Fakt** bezeichnet.

- Beispiele

$Human(Socrates)$

$Fallible(x)\ \leftarrow\ Human(x)$

$Defect(System)\ \leftarrow\ Has\_Part(System, Monitor) \land Defect(Monitor)$

$\leftarrow\ Fallible(who)$

Konsequenzen der Einschränkung auf Horn-Klauseln

- jede Formel kann zu einer oder mehreren Klauseln der Form

  $$H_1 \vee \ldots \vee H_k \leftarrow B_1 \wedge \ldots \wedge B_n$$

  umgeformt werden (Formel in einer Variation der Disjunktiven NormalForm);
  bei Horn-Klauseln darf im Kopf aber keine Disjunktion stehen (d.h., $k = 1$).

- Beispiele von Formeln, die nicht dargestellt werden können:
  $President(Ballmer, Microsoft) \vee President(Ballmer, Oracle)$
  $Healthy(x) \vee Married(x) \leftarrow Happy(x)$
  $Poor(x) \leftarrow \neg Rich(x)$

- Beispiele von Formeln, die nicht „sinnvoll“ dargestellt werden können:
  $\exists X\ (\ Rich(x) \wedge President(x, Oracle)\ )$ [“Ausweg”: $Rich(Sk_1) \wedge President(Sk_1, Oracle)$]
  $\neg President(Jobs, IBM)$ [wäre als Anfrage möglich …]
  $\neg Poor(x) \leftarrow Rich(x)$

Sk_1 ist ein Name für ein unbekanntes Objekt

Resolution (vereinfachte Form)

$\leftarrow A_1 \wedge \ldots \wedge A_{i-1} \wedge \boldsymbol{A_i} \wedge A_{i+1} \wedge \ldots \wedge A_m$ $\quad (m \geq 1)$ $\quad$ Elternklausel

$\boldsymbol{H} \leftarrow B_1 \wedge \ldots \wedge B_n$ $\quad (n \geq 0)$ $\quad$ „

$\leftarrow A_1 \wedge \ldots \wedge A_{i-1} \wedge B_1 \wedge \ldots \wedge B_n \wedge A_{i+1} \wedge \ldots \wedge A_m$ $\quad$ Resolvente

sofern $\boldsymbol{A_i}$ und $\boldsymbol{H}$ syntaktisch gleich sind

(Diese Einschränkung ist etwas zu stark. Sie wird bald gelockert …)

Resolution im Rahmen eines Widerlegungsbeweises

- Sei Δ eine Menge von Fakten und Regeln
- Sei $A_1 \wedge \dots \wedge A_m$ eine Formel mit Variablen $x_1, \dots, x_k$
- Zu beweisen: $\Delta \vdash_{Resolution} \exists x_1 \dots \exists x_k (A_1 \wedge \dots \wedge A_m)$
- Umwandlung in Widerlegungsbeweis:

  Gilt $\Delta \cup \{ \neg \exists x_1 \dots \exists x_k (A_1 \wedge \dots \wedge A_m) \} \vdash_{Resolution} \Box$?

  bzw. $\Delta \cup \{ \leftarrow A_1 \wedge \dots \wedge A_m \} \vdash_{Resolution} \Box$?

- Die Resolutionregel wird angewendet bis
  - die leere Klausel erzeugt wird und damit ein Widerspruch herbeigeführt wurde: **Erfolg** (success) $\Box$
  - kein zu einem Atom A_i passender Regelkopf H mehr gefunden werden kann: **Fehlschlag** (failure) $\blacksquare$

SLD Resolution

Aufruf: $success := prove(\Delta, \leftarrow A_1 \wedge \ldots \wedge A_m);$

procedure *prove*(Δ, *goal*): boolean;

(* returns true if $\Delta \vdash goal$, false otherwise *)

begin

    repeat

        select an atom A_i in the goal;

        choose the next clause $H \leftarrow B_1 \wedge \ldots \wedge B_n$ $(n \geq 0)$ in Δ matching with goal A_i; (*)

        if there is no matching clause **then** **return** *false* **end**;

        $goal := \leftarrow A_1 \wedge \ldots \wedge A_{i-1} \wedge B_1 \wedge \ldots \wedge B_n \wedge A_{i+1} \wedge \ldots \wedge A_m;$

        until $goal = \square$;

    return *true*;

end *prove*;

(*) choice point: hier sind evt. verschiedene Alternativen möglich

SLD = **L**inear resolution for **D**efinite clauses with **S**election function

Widerlegungsbeweis mit Resolution (Beispiel)

$\Delta := \{$ $Has\_part(System, Monitor)$, [1]

$Not\_plugged\_in(System)$, [2]

$Defect(System) \leftarrow Has\_part(System, Monitor) \wedge Defect(Monitor)$, [3]

$Does\_not\_work(System) \leftarrow Defect(System)$, [4]

$Does\_not\_work(System) \leftarrow Not\_plugged\_in(System)$ $\}$ [5]

$goal :=$ $\leftarrow Does\_not\_work(System)$

$\leftarrow Does\_not\_work(System)$

Resolution mit [4]

Resolution mit [5]

$\leftarrow Defect(System)$

$\leftarrow Not\_plugged\_in(System)$

Resolution mit [3]

Resolution mit [2]

$\leftarrow Has\_part(System, Monitor) \wedge Defect(Monitor)$

$\square$ **Erfolg**

Resolution mit [1]

$\leftarrow Defect(Monitor)$

$\blacksquare$ **Fehlschlag**

“Gleichmachen” verschiedener Terme

- Gegeben: Wissensbasis $\Delta = \{\, Human(Socrates),\ Fallible(x) \leftarrow Human(x) \,\}$
  Anfrage $\leftarrow Fallible(Socrates)$

- Atome $A_i = Fallible(Socrates)$ und $H = Fallible(x)$ sind zwar nicht gleich, können aber **syntaktisch gleich gemacht** werden, indem x mit $Socrates$ belegt wird

  dabei müssen alle Vorkommen von x in der betreffenden Formel mit $Socrates$ belegt werden, im Beispiel: $Fallible(Socrates) \leftarrow human(Socrates)$

  $\Rightarrow$ Resolvente ist nicht $Human(x)$, sondern “nur” $Human(Socrates)$

Achtung: folgende Terme sind **syntaktisch** nicht gleich:

a) $4,\ 3+1,\ 1+3,\ 2+2,\ 11-7,\ 2*2$ bzw. $plus(3,1)$ bzw. '+'$(3,1)$ etc.
Bem.: $+$, $-$, * ... sind 2-stellige Funktionssymbole

b) $Prince\_Charles,\ Son\_Of(Queen\_Elizabeth\_II),\ Former\_Husband\_Of(Lady\_Diana)$

Dies ist eine Folge der sog. **U**nique **N**ame **A**ssumption (UNA).

Unifikation

- **Definition**. Zwei Terme (bzw. atomare Formeln) τ_1 und τ_2 **unifizieren**,
  - falls sie syntaktisch gleich sind, d.h., $\tau_1 = \tau_2$ oder
  - falls ihre Variablen so durch Terme belegt werden können, dass die beiden Terme (bzw. atomaren Formeln) identisch werden, d.h., dass eine Substitution θ existiert, so dass $\tau_1\,\theta = \tau_2\,\theta$.

- **Beispiel**

$$\begin{tikzcd}
Date(d, m, 1969) \arrow[dr, "{d / 21,\ m / July}"'] & & Date(e, July, y) \arrow[dl, "{e / 21,\ y / 1969}"] \\
& Date(21, July, 1969) &
\end{tikzcd}$$

Die Substitution $\theta = \{\, d/21\ ,\ e/21,\ m/july,\ y/1969 \,\}$ unifiziert die beiden Terme

Widerlegungsbeweis mit Resolution (Beispiel)

$\Delta := \{ \quad Hate(Marcus, Paulus)\,,$ [1]

$\qquad Hate(Marcus, Caesar)\,,$ [2]

$\qquad Ruler(Caesar)\ \}$ [3]

$goal := \quad \leftarrow Hate(Marcus, x) \wedge Ruler(x)$

$\leftarrow Hate(Marcus, x) \wedge Ruler(x)$

Resolution mit [1] — x / $Paulus$ — $\leftarrow Ruler(Paulus)$

Resolution mit [2] — x / $Caesar$ — $\leftarrow Ruler(Caesar)$

Resolution mit [3]

■ **Fehlschlag**

□ **Erfolg**