Formale Techniken der Software-Entwicklung

Matthias Hölzl, Christian Kroiß

19. Mai 2014

Aussagenlogik

- Beschäftigt sich mit deklarativen Sätzen (Aussagen) und Schlussfolgerungen, die aus (Mengen von) Aussagen getroffen werden können
- Einfache Aussagen werden durch *Aussagenvariablen* (propositionale Variablen) repräsentiert
- Zusammengesetzte Aussagen werden durch *Junktoren* wie „und“ oder „nicht“ aus einfachen Aussagen gebildet.

Einige Aussagen

- A_1 = Hans trank gestern zwei Flaschen Wein
- A_2 = Wer zwei Flaschen Wein trinkt ist betrunken
- A_3 = Hans war gestern betrunken
- A_4 = Alle Menschen sind sterblich
- A_5 = Plato war ein Mensch
- A_6 = Plato ist sterblich

Beschränkte Stärke der Prädikatenlogik

- A_1 = Hans trank gestern zwei Flaschen Wein
- A_2 = Wer zwei Flaschen Wein trinkt ist betrunken
- A_3 = Hans war gestern betrunken
- A_4 = Alle Menschen sind sterblich
- A_5 = Plato war ein Mensch
- A_6 = Plato ist sterblich

In der Aussagenlogik wird die *interne Struktur* von Aussagen ignoriert. Die Vorkommen von „Hans“, „Wein“, „betrunken“, usw. in den Aussagen A_1 bis A_6 sind für die Logik irrelevant. Es gilt z.B. nicht $A_1 \wedge A_2 \Rightarrow A_3$ oder $A_4 \wedge A_5 \Rightarrow A_6$.

Ideen von Constraint Netzwerken

- Repräsentiere *Objekte* im Modell, nicht Aussagen
- Repräsentiere Aussagen durch Relationen über Objekten

Aber: Betrachte allgemeinere Fragestellungen, nicht nur Erfüllbarkeit/Optimierung eines Constraint-Netzwerks

- Verschiedene Relationen über dem gleichen Scope
- Eigenschaften der Relationen (Reflexivität, Symmetrie, . . .)
- Beziehungen zwischen Relationen
- Notation für „für alle“ und „es gibt“
- Syntaktische und semantische Deduktionsbegriffe (Beweistheorie und Modelltheorie)

Prädikatenlogik

- Beschäftigt sich mit *Objekten*, deklarativen Sätzen (die sich auf Objekte beziehen können) und Schlussfolgerungen, die daraus getroffen werden können
- Objekte werden durch *Individuenvariablen* und *Konstanten* repräsentiert
- Einfache Aussagen entstehen, indem man *Prädikate* auf Objekte anwendet
- Zusammengesetzte Aussagen werden durch *Junktoren*, wie „und“ oder „nicht“, und *Quantoren*, wie „für alle“ oder „es gibt ein“, aus einfachen Aussagen gebildet.

Beispiel

- Variable x
- Konstanten Plato, Hans, Fido, ...
- Einstellige Prädikate M („ist/war ein Mensch“) und S („ist sterblich“)
- A_4 („Alle Menschen sind sterblich“) entspricht

$$\forall x.M(x) \Rightarrow S(x)$$

  Der Punkt nach $\forall x$ bedeutet, dass der Quantor $\forall$ so weit wie möglich gilt, die Formel bedeutet also $\forall x.(M(x) \Rightarrow S(x))$
- A_5 („Plato war ein Mensch“) entspricht $M(\text{Plato})$
- A_6 („Plato ist sterblich“) entspricht $S(\text{Plato})$
- In der Prädikatenlogik gilt $A_4 \wedge A_5 \Rightarrow A_6$

$$(\forall x.M(x) \Rightarrow S(x)) \wedge M(\text{Plato}) \Rightarrow S(\text{Plato})$$

Vorsicht!

- Die Formalisierung von „Alle Menschen sind sterblich“ ist

$$\forall x.M(x) \Rightarrow S(x)$$

nicht $\forall x.M(x) \wedge S(x)$

oder $(\forall x.M(x)) \Rightarrow (\forall x.S(x))$

- Die erste Ausage besagt, dass für alle Objekte x die folgende Eigenschaft gilt: Ist x ein Mensch, dann ist x sterblich
- Die zweite Aussage besagt, dass für alle Objekte x die folgende Eigenschaft gilt: x ist ein Mensch *und* x ist sterblich
- Die dritte Aussage besagt, dass wenn für alle Objekte x die Eigenschaft $M(x)$ gilt, dann gilt auch für alle Objekte x die Eigenschaft $S(x)$
- Für den Hund Fido gilt $S(x)$ aber $\neg M(x)$. Die zweite Aussage ist also falsch, die dritte Aussage ist wahr weil $\forall x.M(x)$ falsch ist, sagt aber nichts über die Sterblichkeit von Menschen aus

Quantoren

- $\forall$ steht für „für alle“
- $\exists$ steht für „es gibt (mindestens) ein“
- „Für alle ... gilt ...“ formalisiert man als

$$\forall x. \cdots \Rightarrow \cdots$$

- „Es gibt ... mit ...“ formalisiert man als

$$\exists x. \cdots \wedge \cdots$$

Funktionsymbole

Oft ist es sinnvoll Funktionen, die Objekte in andere Objekte abbilden, in Prädikatenlogischen Formeln zu verwenden:

- „Die Mutter jedes Menschen ist ein Mensch“

$$\forall x. M(x) \Rightarrow M(\text{mutter}(x))$$

(Funktionssymbol mutter)

- „Die Summe zweier gerader Zahlen ist eine gerade Zahl“

$$\forall x, y. \text{even}(x) \land \text{even}(y) \Rightarrow \text{even}(x + y)$$

(Funktionssymbol $+$) oder

$$\forall x, y. x \bmod 2 = 0 \land y \bmod 2 = 0 \Rightarrow (x + y) \bmod 2 = 0$$

(Funktionssymbole $+$, mod)

Syntax: Terme

Seien *Var* eine Menge von Variablensymbolen, *Fun* eine Menge von Funktionssymbolen und *Const* eine Menge von Konstantensymbolen. Sei $|.| : \mathit{Fun} \to \mathbb{N}$ eine Abbildung, die jedem Funktionssymbol eine *Stelligkeit* zuordnet. Ist $|f| = n$, so nennt man f n-stellig.
Die Menge der Prädikatenlogischen Terme $\mathcal{T}$ (über *Var*, *Fun* und *Const*) ist folgendermaßen rekursiv definiert.

- Jede Variable $x \in \mathit{Var}$ ist in $\mathcal{T}$ (d.h. $\mathit{Var} \subseteq \mathcal{T}$)
- Jede Konstante $c \in \mathit{Const}$ ist in $\mathcal{T}$ (d.h. $\mathit{Const} \subseteq \mathcal{T}$)
- Sei f ein n-stelliges Funktionssymbol und seien $t_1, \ldots, t_n$ Terme, dann ist $f(t_1, \ldots, t_n) \in \mathcal{T}$

Variablen und Konstanten nennt man auch *Primterme*. Alle anderen Terme heißen *Funktionsterme*.
Terme in denen keine Variablen vorkommen heißen *Grundterme*.
Grundterme kann es nur geben, wenn *Const* nichtleer ist.

Terminduktion

Seien $t = f(t_1, \ldots, t_n)$ und $s = g(s_1, \ldots, s_m)$ syntaktisch gleiche Terme, $t \equiv s$. Dann ist $m = n$, $f \equiv g$ und für alle i mit $1 \leq i \leq n$ gilt $t_i \equiv s_i$.

Das folgende Beweisprinzip wird häufig verwendet, um syntaktische Eigenschaften zu beweisen.

Proposition (Terminduktion)

Sei $\mathcal{E}$ eine Eigenschaft von Termen, für die folgendes gilt:

- *$\mathcal{E}(t)$ gilt für alle Primterme t*
- *Sei f ein n-stelliges Funktionssymbol und gelte $\mathcal{E}(t_i)$ für $t_1, \ldots, t_n$. Dann gilt auch $\mathcal{E}(f(t_1, \ldots, t_n))$*

Dann gilt $\mathcal{E}$ für alle Terme.

Entsprechend kann man Eigenschaften von Termen durch *Rekursion über die Termstruktur* definieren.

Syntax: Atomare Formeln

Sei $\mathcal{T}$ eine Menge von Termen (über *Var*, *Fun* und *Const*) und *Pred* eine Menge von Prädikatensymbolen. Sei $|.| : \mathit{Pred} \to \mathbb{N}$ eine Abbildung, die jedem Prädikatensymbol eine *Stelligkeit* zuordnet. Ist $|P| = n$, so nennt man P ein n-stelliges Prädikat.

Die *atomaren Formeln* (*Primformeln*) $\mathcal{A}$ (über *Var*, *Fun*, *Const* und *Pred*) sind dann folgendermaßen rekursiv definiert:

- Seien P ein n-stelliges Prädikatensymbol und $t_1, \ldots, t_n \in T$. Dann ist $P(t_1, \ldots, t_n) \in \mathcal{A}$
- Für $t_1, t_2 \in \mathcal{T}$ ist $t_1 = t_2 \in \mathcal{A}$

Syntax: Formeln

Seien *Var* eine Menge von Variablen und $\mathcal{A}$ die Menge von atomaren Formeln über *Var*, *Fun*, *Const* und *Pred*. Die Menge der prädikatenlogischen Formeln $\mathcal{L}$ (über *Var*, *Fun*, *Const* und *Pred*) ist dann folgendermaßen rekursiv definiert:

- Ist $\phi \in \mathcal{A}$ so ist $\phi \in \mathcal{L}$ (d.h. $\mathcal{A} \subseteq \mathcal{L}$)
- Sind $\phi, \psi \in \mathcal{L}$, so sind auch $\neg\phi$, $(\phi \wedge \psi)$, $(\phi \vee \psi)$, $(\phi \Rightarrow \psi)$ und $(\phi \Leftrightarrow \psi)$ in $\mathcal{L}$ enthalten
- Sind $x \in \mathit{Var}$ und $\phi \in \mathcal{L}$, so sind auch $\forall x.\phi$ und $\exists x.\phi$ in $\mathcal{L}$ enthalten

Formeln nennt man auch Ausdrücke oder Aussageformen.

Formeln

Formeln erlauben, genau wie Terme, Induktionsbeweise und rekursive Definitionen.

Zum Beispiel kann man den *Rang* rg einer Formel rekursiv definieren:

- $\mathsf{rg}(\pi) = 0$ für Primformeln π
- $\mathsf{rg}(\neg\phi) = \mathsf{rg}(\forall x.\phi) = \mathsf{rg}(\exists x.\phi) = \mathsf{rg}(\phi) + 1$
- $\mathsf{rg}(\phi \wedge \psi) = \mathsf{rg}(\phi \vee \psi) = \mathsf{rg}(\phi \Rightarrow \psi) = \mathsf{rg}(\phi \Leftrightarrow \psi) = \max(\mathsf{rg}(\phi), \mathsf{rg}(\psi))$

Auch der Begriff der *Teilformel* oder *Subformel* kann rekursiv definiert werden.

Die für die Aussagenlogik definierten Präzedenz- und Assoziativitätsregeln werden auch bei prädikatenlogischen Formeln verwendet um Klammern einzusparen.

Formeln

Für manche Arten von Formeln gibt es besondere Bezeichnungen:

- Primformeln und Negationen von Primformeln nennt man *Literale*
- Formeln, in denen $\forall$ und $\exists$ nicht vorkommen, nennt man *quantorenfreie Formeln*
- Sei X eine Menge von Formeln. Die Formeln, die man durch die Junktoren $(\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow)$ aus Elementen von X erzeugen kann, heißen *Boole'sche Kombinationen* aus X

Die Sprache $\mathcal{L}$

Die Mengen der Konstanten-, Funktions- und Prädikatssymbole, *Const*, *Fun*, *Pred*, zusammen mit den Stelligkeiten $|.|$ für Funktions- und Prädikatssymbole nennt man die *Signatur* von $\mathcal{L}$ und schreibt $\text{sig}(\mathcal{L})$.

Man nimmt an, dass die Menge der Variablen *Var* für alle Sprachen identisch ist. Dann bestimmt $\text{sig}(\mathcal{L})$ die Menge $\mathcal{L}$ eindeutig. Man verwendet deshalb das Symbol $\mathcal{L}$ oft auch als Bezeichnung für die Signatur $\text{sig}(\mathcal{L})$, und nennt $\mathcal{T}$ auch die Menge der $\mathcal{L}$-Terme und $\mathcal{L}$ die Menge der $\mathcal{L}$-Formeln.

Semantik (Vorbereitung)

Ziel: Wir wollen jeder Formel einen Wahrheitswert zuordnen. Informell kann man diese Idee folgendermaßen ausdrücken:

- Terme werden durch „Individuen“, d.h. Elemente aus einer vorgegebenen Menge A, repräsentiert
- Prädikate werden durch Relationen über A interpretiert: jedem n-stelligen Prädikat entspricht eine n-stellige Relation über A
- Gleichheit wird durch die Gleichheit auf A interpretiert: zwei Terme sind gleich, wenn sie durch das gleiche Element aus A repräsentiert werden

...

Semantik (Vorbereitung)

- Die Bedeutung der Junktoren soll der Aussagenlogik entsprechen, d.h., eine Aussage der Form $\phi(\vec{x}) \wedge \psi(\vec{x})$ wird durch eine Relation über $\vec{x}$ interpretiert, die genau dann wahr ist, wenn die Interpretation von $\phi(\vec{x})$ und $\psi(\vec{x})$ beide wahr sind
- Eine Formel der Form $\forall x.\phi$ soll wahr sein, wenn die durch ϕ definierte Relation wahr ist wenn man für die Variable x ein beliebiges Element $a \in A$ einsetzt, d.h., wenn die Projektion der durch ϕ definierten Relation auf x die ganze Menge A ist
- Eine Formel der Form $\exists x.\phi$ soll wahr sein, wenn die Interpretation von ϕ nicht die leere Relation ist, d.h., wenn die Projektion der durch ϕ definierten Relation auf x nicht leer ist

Diese Ideen wollen wir im Folgenden präzisieren.

Strukturen

Eine Struktur $\mathcal{A}$ zur Signatur $\sigma = (\mathit{Const}, \mathit{Fun}, \mathit{Pred}\ |.|)$ (kurz σ-Struktur) besteht aus

- einer nichtleeren Menge A, dem *Träger* (oder der *Grundmenge*) der Struktur
- einem Element $c^{\mathcal{A}} \in A$ für jedes Konstantensymbol $c \in \mathit{Const}$
- einer Funktion $f^{\mathcal{A}} : A^n \to A$ für jedes n-stellige Funktionssymbol $f \in \mathit{Fun}$
- einer Relation $P^{\mathcal{A}} \subseteq A^n$ für jedes n-stellige Prädikatssymbol P

Eine Struktur heißt *endlich* (bzw. *unendlich*), wenn die Trägermenge endlich (bzw. unendlich) ist.

Beispiel

Sei $\sigma = (\{1\}, \{\circ\}, \emptyset, |.|)$ mit $|\cdot| = 2$.

$\mathcal{A} = (\mathbb{N}, 1, \cdot)$ ist eine σ-Struktur, die 1 und $\circ$ die Zahl 1 und die Multiplikation zuordnet:

$$1^{\mathcal{A}} = 1$$
$$\circ^{\mathcal{A}} = \cdot$$

Ebenso ist $\mathcal{B} = (\mathbb{N}, 0, +)$ eine σ-Struktur, die 1 und $\circ$ die Zahl 0 und die Addition zuordnet:

$$1^{\mathcal{B}} = 0$$
$$\circ^{\mathcal{B}} = +$$

Modelle der Prädikatenlogik

Definition

Ein *Modell* $\mathcal{M}$ einer Sprache $\mathcal{L}$ ist ein Paar $(\mathcal{A}, w)$, bestehend aus einer $\mathcal{L}$-Struktur $\mathcal{A}$ (mit Träger A) und einer *Belegung* $w : Var \to A$. Wir schreiben $[\![c]\!]_{\mathcal{M}}$ (oder $c^{\mathcal{M}}$, oder $[\![c]\!]$ falls $\mathcal{M}$ klar ist) für $c^{\mathcal{A}}$, entsprechend für Funktions- und Prädikatensymbole.

Modelle einer Sprache $\mathcal{L}$ nennt man auch $\mathcal{L}$-Modelle oder Interpretationen von $\mathcal{L}$. Den Träger von $\mathcal{A}$ nennt man auch Träger von $\mathcal{M}$.

Wir schreiben $M[x \mapsto a]$ für das Modell (A, w') mit

$$w'(y) = \begin{cases} w(y) & \text{für } y \neq x \\ a & \text{für } y = x \end{cases}$$

Semantik von Termen

Durch ein Modell $\mathcal{M}$ wird jedem $\mathcal{L}$-Term ein Element aus A zugeordnet:

$$\begin{aligned}
[\![x]\!]_{\mathcal{M}} &= w(x) \\
[\![c]\!]_{\mathcal{M}} &= c^{\mathcal{A}} \\
[\![f(t_1, \ldots, t_n)]\!]_{\mathcal{M}} &= f^{\mathcal{A}}([\![t_1]\!]_{\mathcal{M}}, \ldots, [\![t_n]\!]_{\mathcal{M}})
\end{aligned}$$

...

Erfüllungsrelation

Die Semantik von Formeln lässt sich durch die *Erfüllungsrelation* $\mathcal{M} \models \phi$ ($\mathcal{M}$ erfüllt ϕ oder $\mathcal{M}$ ist ein Modell von ϕ) beschreiben:

$$
\begin{aligned}
\mathcal{M} \models R(t_1, \ldots, t_n) &\iff R^{\mathcal{A}}([\![t_1]\!]_{\mathcal{M}}, \ldots, [\![t_n]\!]_{\mathcal{M}}) \\
\mathcal{M} \models t_1 = t_2 &\iff [\![t_1]\!]_{\mathcal{M}} = [\![t_2]\!]_{\mathcal{M}} \\
\mathcal{M} \models \neg\phi &\iff \mathcal{M} \models \phi \text{ ist falsch } (M \not\models \phi) \\
\mathcal{M} \models \phi \wedge \psi &\iff \mathcal{M} \models \phi \text{ und } \mathcal{M} \models \psi \\
\mathcal{M} \models \phi \vee \psi &\iff \mathcal{M} \models \phi \text{ oder } \mathcal{M} \models \psi \\
\mathcal{M} \models \phi \Rightarrow \psi &\iff \mathcal{M} \not\models \phi \text{ oder } \mathcal{M} \models \psi \\
\mathcal{M} \models \phi \Leftrightarrow \psi &\iff (\mathcal{M} \models \phi \text{ und } \mathcal{M} \models \psi) \\
&\qquad \text{oder } (\mathcal{M} \not\models \phi \text{ und } \mathcal{M} \not\models \psi) \\
\mathcal{M} \models \forall x.\phi &\iff \mathcal{M}[x \mapsto a] \models \phi \text{ für alle } a \\
\mathcal{M} \models \exists x.\phi &\iff \text{es gibt } a \text{ mit } \mathcal{M}[x \mapsto a] \models \phi
\end{aligned}
$$

Semantik von Formeln

In einer an die Semantik der Aussagenlogik angelehnten Schreibweise kann man die Erfüllbarkeitsrelation auch folgendermaßen aufschreiben:

$$
\begin{aligned}
[\![P(t_1, \ldots, t_n)]\!]_{\mathcal{M}} &= P^{\mathcal{A}}([\![t_1]\!]_{\mathcal{M}}, \ldots, [\![t_n]\!]_{\mathcal{M}}) \\
[\![t_1 = t_2]\!]_{\mathcal{M}} &= [\![t_1]\!]_{\mathcal{M}} = [\![t_2]\!]_{\mathcal{M}} \\
[\![\neg\phi]\!]_{\mathcal{M}} &= ![\![\phi]\!]_{\mathcal{M}} \\
[\![\phi \wedge \psi]\!]_{\mathcal{M}} &= [\![\phi]\!]_{\mathcal{M}} \,\&\, [\![\psi]\!]_{\mathcal{M}} \\
[\![\phi \vee \psi]\!]_{\mathcal{M}} &= [\![\phi]\!]_{\mathcal{M}} \mid [\![\psi]\!]_{\mathcal{M}} \\
[\![\phi \Rightarrow \psi]\!]_{\mathcal{M}} &= (![\![\phi]\!]_{\mathcal{M}}) \mid [\![\psi]\!]_{\mathcal{M}} \\
[\![\phi \Leftrightarrow \psi]\!]_{\mathcal{M}} &= [\![\phi]\!]_{\mathcal{M}} = [\![\psi]\!]_{\mathcal{M}} \\
[\![\forall x.\phi]\!]_{\mathcal{M}} &= \text{wahr gdw. } [\![\phi]\!]_{\mathcal{M}[x \mapsto a]} = \text{wahr für alle } a \\
[\![\exists x.\phi]\!]_{\mathcal{M}} &= \text{wahr gdw. es ein } a \text{ gibt mit } [\![\phi]\!]_{\mathcal{M}[x \mapsto a]} = \text{wahr}
\end{aligned}
$$

Erfüllbarkeit, Tautologien

- Eine Formel heißt *erfüllbar*, wenn sie ein Modell besitzt
- Eine Formel $\phi \in \mathcal{L}$ heißt *allgemeingültig*, *logisch gültig* oder *Tautologie*, wenn sie in allen Modellen wahr ist, wenn also für alle $\mathcal{L}$-Modelle gilt $\mathcal{M} \models \phi$
- Formeln ϕ und ψ heißen *logisch äquivalent*, wenn sie von den gleichen $\mathcal{L}$-Modellen erfüllt werden, wenn also $\mathcal{M} \models \phi$ genau dann gilt, wenn $\mathcal{M} \models \psi$ gilt
- Sei Φ eine Menge von Formeln. Wir schreiben $\mathcal{M} \models \Phi$, wenn $\mathcal{M} \models \phi$ für alle $\phi \in \Phi$ gilt
- Wir schreiben $\Phi \models \psi$ (aus Φ folgt ψ) wenn jedes Modell von Φ auch ψ erfüllt, wenn also gilt $\mathcal{M} \models \Phi \implies \mathcal{M} \models \psi$.

Vorsicht: Die letzten beiden Definitionen sind anders als bei Sequenzen!

Beispiele

- $\forall x.\exists y.x \neq y$ ist erfüllbar. (Wähle ein beliebiges Modell, dessen Träger mindestens zwei Elemente enthält.)
- $\forall x.\forall y.x = y$ ist erfüllbar. (Wähle ein beliebiges Modell, dessen Träger genau ein Element enthält.)
- Keine dieser Formeln ist allgemeingültig
- $\forall x.x = x$ ist allgemeingültig
- $x = y$ und $\neg(y \neq x)$ sind logisch äquivalent
- $\{\phi \Rightarrow \psi, \phi\} \models \psi$. (Sei $\mathcal{M}$ ein Modell für das $\mathcal{M} \models \phi \Rightarrow \psi$ und $\mathcal{M} \models \phi$ gilt. Die erste Relation ist äquivalent zu „$\mathcal{M} \not\models \phi$ oder $\mathcal{M} \models \psi$.“ Da $\mathcal{M} \models \phi$ gilt muss also $\mathcal{M} \models \psi$ gelten.)

Zusammenfassung

- Ein $\mathcal{L}$-Modell $\mathcal{M}$ besteht aus einer $\mathcal{L}$-Struktur $\mathcal{A}$ und einer Variablenbelegung w
- Durch die Wahl eines Modells lässt sich die Erfüllbarkeitsrelation für prädikatenlogische Formeln $\mathcal{M} \models \phi$ definieren
- Durch die Erfüllbarkeitsrelation lassen sich die Begriffe Erfüllbarkeit und Allgemeingültigkeit für Formeln sowie ein Folgerungsbegriff (zwischen einer Formelmenge und einer Formel) definieren

Beweissysteme

Wie bei der Aussagenlogik wollen wir auch für die Prädikatenlogik Beweissysteme angeben, die durch syntaktische Manipulationen die Erfüllbarkeit oder Allgemeingültigkeit von Formeln herleiten können. Wir werden dazu Erweiterungen des von PVS verwendeten Sequenzenkalküls und einen Resolutionskalkül betrachten.

Dazu führen wir auf den nächsten Folien einige Begriffe ein.

Freie Variablen

Die freien Variablen in Termen und Formeln sind folgendermaßen rekursiv definiert:

$$
\begin{aligned}
\mathrm{fv}(c) &= \emptyset \\
\mathrm{fv}(x) &= \{x\} \\
\mathrm{fv}(f(t_1, \ldots, t_n)) &= \bigcup_{i=1}^{n} \mathrm{fv}(t_i) \\
\\
\mathrm{fv}(R(t_1, \ldots, t_n)) &= \bigcup_{i=1}^{n} \mathrm{fv}(t_i) \\
\mathrm{fv}(t_1 = t_2) &= \mathrm{fv}(t_1) \cup \mathrm{fv}(t_2) \\
\mathrm{fv}(\neg\phi) &= \mathrm{fv}(\phi) \\
\mathrm{fv}(\phi \,\mathrm{op}\, \psi) &= \mathrm{fv}(\phi) \cup \mathrm{fv}(\psi) \quad \text{für } \mathrm{op} \in \{\wedge, \vee, \Rightarrow, \Leftrightarrow\} \\
\mathrm{fv}(\forall x.\phi) &= \mathrm{fv}(\phi) \setminus \{x\} \\
\mathrm{fv}(\exists x.\phi) &= \mathrm{fv}(\phi) \setminus \{x\}
\end{aligned}
$$

Gebundene Variablen

Die gebundenen Variablen in Termen und Formeln sind folgendermaßen rekursiv definiert:

$$\begin{aligned}
\mathrm{bv}(t) &= \emptyset \\
\mathrm{bv}(R(t_1, \ldots, t_n)) &= \emptyset \\
\mathrm{bv}(t_1 = t_2) &= \emptyset \\
\mathrm{bv}(\neg\phi) &= \mathrm{bv}(\phi) \\
\mathrm{bv}(\phi \,\mathrm{op}\, \psi) &= \mathrm{bv}(\phi) \cup \mathrm{bv}(\psi) \quad \text{für } \mathrm{op} \in \{\wedge, \vee, \Rightarrow, \Leftrightarrow\} \\
\mathrm{bv}(\forall x.\phi) &= \{x\} \cup \mathrm{bv}(\phi) \\
\mathrm{bv}(\exists x.\phi) &= \{x\} \cup \mathrm{bv}(\phi)
\end{aligned}$$

Man nennt ϕ in der Formel $\forall x.\phi$ (bzw. $\exists x.\phi$) den *Wirkungsbereich* des Quantors $\forall x$ (bzw. $\exists x$).

Freie und Gebundene Variablen

Die Variable x kommt gebunden, die Variable y frei in $\exists x.P(x,y)$ vor:

$$\begin{aligned}
\mathrm{fv}(P(x,y)) &= \mathrm{fv}(x) \cup \mathrm{fv}(y) \\
&= \{x, y\} \\
\mathrm{fv}(\exists x.P(x,y)) &= \mathrm{fv}(P(x,y)) \setminus \{x\} \\
&= \{y\}
\end{aligned}$$

$$\begin{aligned}
\mathrm{bv}(P(x,y)) &= \emptyset \\
\mathrm{bv}(\exists x.P(x,y)) &= \{x\} \cup \mathrm{bv}(P(x,y)) \\
&= \{x\}
\end{aligned}$$

Freie und Gebundene Variablen

Eine Variable kann in einer Formel gleichzeitig frei und gebunden vorkommen:

$$\mathrm{fv}(M(x) \wedge \exists x.M(x)) = \{x\}$$
$$\mathrm{bv}(M(x) \wedge \exists x.M(x)) = \{x\}$$

Das erste Vorkommen von x ist frei, das zweite gebunden.

Manchmal schreibt man $\phi(x_1, \ldots, x_n)$ um anzudeuten, dass in der Formel ϕ höchstens die Variablen $x_1, \ldots, x_n$ frei vorkommen.

Substitutionen

Eine wichtige Umformung von Formeln ist das Ersetzen von Variablen durch Terme. Eine *Substitution* beschreibt die Ersetzung von einer Variable durch einen Term, oder auch die gleichzeitige Ersetzung mehrerer Variablen durch mehrere Terme. Formal ist eine Substitution eine partielle Abbildung $\sigma : \mathit{Var} \rightharpoonup \mathcal{T}$.

Wir schreiben einfache Substitutionen in der Form $[x \mapsto t]$ und die gleichzeitige Substitution mehrerer Variablen als $[x_1 \mapsto t_1, \ldots, x_n \mapsto t_n]$ oder $\phi[\,\vec{x} \mapsto \vec{t}\,]$. Die Anwendung einer Substitution σ auf einen Term ϕ notiert man $\phi\sigma$ oder ϕ^σ.

Ersetzt man zum Beispiel in $M(x)$ die Variable x durch die Konstante Plato, so erhält man die Formel $M(\text{Plato})$, kürzer

$$M(x)[x \mapsto \text{Plato}] \equiv M(\text{Plato}).$$

Umbenennen von Variablen

Gebundene Variablen kann man (fast) beliebig umbenennen ohne dass sich die Bedeutung eines Terms ändert. Z.B. bedeutet die Formel $\forall x.M(x) \Rightarrow S(x)$ genau das Gleiche wie $\forall y.M(y) \Rightarrow S(y)$.

Man schreibt $P \simeq Q$ wenn sich P und Q nur durch die korrekte Umbenennung gebundener Variablen unterscheiden.

Die Umbenennung darf keine freien Variablen „einfangen," d.h. keine vorher freie Variable darf sich nach der Umbenennung im Wirkungsbereich eines Quantors befinden:

$$\forall x.P(x, y) \not\simeq \forall y.P(y, y)$$

($\mathcal{M} \models \forall x.P(x, y)$ gdw. $P^{\mathcal{A}}(a, w(y))$ für alle $a \in A$ gilt, $\mathcal{M} \models \forall y.P(y, y)$ gdw. $P^{\mathcal{A}}(a, a)$ für alle $a \in A$ gilt.)

Substitution und Gebundene Variablen

Das Vorkommen von gebundenen Variablen verkompliziert die Definition der Substitution. Gebundene Variablen werden nicht substituiert, und freie Variablen im substituierten Term dürfen nicht eingefangen werden:

$$(\forall x.M(x))[x \mapsto \text{Plato}] \simeq \forall x.M(x)$$
$$(\forall y.P(x,y))[x \mapsto y] \simeq \forall z.P(y,z)$$

Wir schränken daher Substitutionen so ein, dass Variablen im substituierten Term nicht von Quantoren eingefangen werden können:

Definition

ϕ und $[x \mapsto t]$ heißen *kollisionsfrei*, wenn für alle $y \in \text{fv}(t) \setminus \{x\}$ gilt $y \notin \text{bv}(\phi)$. Entsprechend für $[\vec{x} \mapsto \vec{t}\,]$.

Substitution in Formeln

Durch kollisionsfreie Substitutionen werden keine Variablen eingefangen.
Durch Umbenennen der gebundenen Variablen in einem Term ϕ kann man (für endliche Substitutionen) immer erreichen, dass eine gegebene Substitution σ kollisionsfrei mit ϕ ist.

Konvention

Wir setzen im Folgenden voraus, dass alle Substitutionen kollisionsfrei sind.

Definition der Substitution

Seien alle vorkommenden Terme und Formeln mit $[x \mapsto t]$ kollisionsfrei.

$$
\begin{aligned}
x[x \mapsto t] &= t \\
y[x \mapsto t] &= y \quad \text{falls } y \neq x \\
c[x \mapsto t] &= c \\
f(t_1, \ldots, t_n)[x \mapsto t] &= f(x_1[x \mapsto t], \ldots, x_n[x \mapsto t]) \\
\\
R(t_1, \ldots, t_n)[x \mapsto t] &= R(x_1[x \mapsto t], \ldots, x_n[x \mapsto t]) \\
(t_1 = t_2)[x \mapsto t] &= t_1[x \mapsto t] = t_2[x \mapsto t] \\
(\neg\phi)[x \mapsto t] &= \neg(\phi[x \mapsto t]) \\
(\phi \operatorname{op} \psi)[x \mapsto t] &= (\phi[x \mapsto t]) \operatorname{op} (\psi[x \mapsto t]) \quad \operatorname{op} \in \{\wedge, \vee, \Rightarrow, \Leftrightarrow\} \\
(\mathrm{Q}\, x.\phi)[x \mapsto t] &= (\mathrm{Q}\, x.\phi) \qquad \mathrm{Q} \in \{\forall, \exists\} \\
(\mathrm{Q}\, y.\phi)[x \mapsto t] &= (\mathrm{Q}\, y.\phi[x \mapsto t]) \quad \mathrm{Q} \in \{\forall, \exists\}, y \neq x
\end{aligned}
$$

Substitutionssatz

Sei $\mathcal{M}$ ein $\mathcal{L}$-Modell. Wir defineren $\mathcal{M}\sigma$ als das Modell $(\mathcal{A}, w^\sigma)$, wobei $w^\sigma(x) = (x\sigma)^\mathcal{M}$ für alle $x \in \mathit{Var}$. Für $\sigma : \mathit{Var} \rightharpoonup A$ definieren wir $\mathcal{M}\sigma$ als $(\mathcal{A}, w^\sigma)$ mit $w^\sigma(x) = x\sigma$ falls $x \in \mathrm{dom}(\sigma)$, $w^\sigma(x) = w(x)$ sonst.

Theorem (Substitutionssatz)

Sei $\mathcal{M}$ ein Modell und σ eine Substitution. Für alle Formeln ϕ, die mit σ kollisionsfrei sind gilt

$$\mathcal{M} \models \phi\sigma \iff \mathcal{M}\sigma \models \phi$$

Der Beweis erfolgt durch Induktion über ϕ.

Mit der eingeführten Notation können wir die Definition von $\mathcal{M} \models \forall x.\phi$ folgendermaßen schreiben:

$$\mathcal{M} \models \forall x.\phi \iff \text{für alle } a \in A\text{: } \mathcal{M}[x \mapsto a] \models \phi$$

Folgerungen aus dem Substitutionssatz

Aus dem Substitutionssatz folgt das wichtige Korollar

Korollar (aus dem Substitutionssatz)

Seien ϕ und $[x \mapsto t]$ kollisionsfrei. Dann gelten

- $\forall x.\phi \models \phi[x \mapsto t]$
- $\phi[x \mapsto t] \models \exists x.\phi$
- $\phi[x \mapsto t_1], t_1 = t_2 \models \phi[x \mapsto t_2]$

Gelte $\mathcal{M} \models \forall x.\phi$, also $\mathcal{M}[x \mapsto a] \models \phi$ für alle $a \in A$. Da $t^{\mathcal{M}} \in A$ ist gilt somit auch $\mathcal{M}[x \mapsto t] \models \phi$, nach dem Substitutionssatz also $\mathcal{M} \models \phi[x \mapsto t]$. Die anderen Aussagen zeigt man analog.

Sequenzenkalkül (1)

$$\frac{\Gamma_1 \vdash \Delta_1}{\Gamma_2 \vdash \Delta_2}\mathbf{W} \qquad \text{if } \Gamma_1 \subseteq \Gamma_2 \wedge \Delta_1 \subseteq \Delta_2$$

$$\frac{}{\Gamma, \phi \vdash \psi, \Delta}\mathbf{Ax} \qquad \text{if } \phi \simeq \psi$$

$$\frac{}{\Gamma, \bot \vdash \Delta}\bot \qquad \frac{}{\Gamma \vdash \top, \Delta}\top$$

$$\frac{\Gamma \vdash \phi, \Delta}{\Gamma, \neg\phi \vdash \Delta}\neg\vdash \qquad \frac{\Gamma, \phi \vdash \Delta}{\Gamma \vdash \neg\phi, \Delta}\vdash\neg$$

Sequenzenkalkül (2)

$$\frac{\phi, \psi, \Gamma \vdash \Delta}{\phi \wedge \psi, \Gamma \vdash \Delta} \wedge\vdash \qquad \frac{\Gamma \vdash \phi, \Delta \quad \Gamma \vdash \psi, \Delta}{\Gamma \vdash \phi \wedge \psi, \Delta} \vdash\wedge$$

$$\frac{\phi, \Gamma \vdash \Delta \quad \psi, \Gamma \vdash \Delta}{\phi \vee \psi, \Gamma \vdash \Delta} \vee\vdash \qquad \frac{\Gamma \vdash \phi, \psi, \Delta}{\Gamma \vdash \phi \vee \psi, \Delta} \vdash\vee$$

$$\frac{\psi, \Gamma \vdash \Delta \quad \Gamma \vdash \phi, \Delta}{\phi \Rightarrow \psi, \Gamma \vdash \Delta} \Rightarrow\vdash \qquad \frac{\Gamma, \phi \vdash \psi, \Delta}{\Gamma \vdash \phi \Rightarrow \psi, \Delta} \vdash\Rightarrow$$

Sequenzenkalkül (3)

$$\frac{}{\Gamma \vdash t = t, \Delta}\textbf{Refl} \qquad \frac{t_1 = t_2, \Gamma[x \mapsto t_1] \vdash \Delta[x \mapsto t_1]}{t_1 = t_2, \Gamma[x \mapsto t_2] \vdash \Delta[x \mapsto t_2]}\textbf{Repl}$$

$$\frac{\phi[x \mapsto t], \Gamma \vdash \Delta}{\forall x.\phi, \Gamma \vdash \Delta}\forall\vdash \qquad \frac{\Gamma \vdash \phi[x \mapsto c], \Delta}{\Gamma \vdash \forall x.\phi, \Delta}\vdash\forall \quad c \text{ frisch}$$

$$\frac{\phi[x \mapsto c], \Gamma \vdash \Delta}{\exists x.\phi, \Gamma \vdash \Delta}\exists\vdash \quad c \text{ frisch} \qquad \frac{\Gamma \vdash \phi[x \mapsto t], \Delta}{\Gamma \vdash \exists x.\phi, \Delta}\vdash\exists$$

Alternative Syntax: PVS

Terme:

$t \in \mathcal{T}$	$t' \in$PVS
x	`x`
c	`c`
$f(t_1, \ldots, t_n)$	`f(`$t'_1, \ldots, t'_n$`)`

Formeln:

$\xi \in \mathcal{L}$	$\xi' \in$ PVS
$P(t_1, \ldots, t_n)$	`P(`$t'_1, \ldots, t'_n$`)`
$t_1 = t_2$	t'_1 `=` t'_2
$\neg\phi$	`not(`ϕ'`)`
$\phi \wedge \psi$	ϕ' `and` ψ', ϕ' `&` ψ'
$\phi \vee \psi$	ϕ' `or` ψ'
$\phi \Rightarrow \psi$	ϕ' `implies` ψ', ϕ' `=>` ψ'
$\phi \Leftrightarrow \psi$	ϕ' `iff` ψ', ϕ' `<=>` ψ'
$\forall x.\phi$	`forall (x:A):` ϕ'
$\exists x.\phi$	`exists (x:A):` ϕ'

Beispiel

$\xi \in \mathcal{L}$	$\xi' \in$ Snark
$M(\text{Plato})$	`M(plato)`
$M(x)$	`M(x)`
$\forall x.M(x) \Rightarrow S(x)$	`forall (x:A): M(x) => S(x)`
$\forall x.M(x) \Rightarrow M(\text{mutter}(x))$	`forall (x:A): M(x) => S(mutter(x))`

PVS verwendet eine mehrsortige (typisierte) Logik. Daher muss bei jeder gebundenen Variable der Typ der Variable angegeben werden. Man kann eine einsortige (ungetypte) Logik simulieren, indem man einen Typ, z.B. `A`, für alle Variablen verwendet.

Alternative Syntax: Snark/Poem/KIF

Terme:

$t \in \mathcal{T}$	$t' \in$ Snark
x	`?x, x`
c	`c`
$f(t_1, \ldots, t_n)$	`(f` $t'_1 \ldots t'_n$`)`

Formeln:

$\xi \in \mathcal{L}$	$\xi' \in$ Snark
$P(t_1, \ldots, t_n)$	`(p` $t'_1 \ldots t'_n$`)`
$t_1 = t_2$	`(=` t'_1 t'_2`)`
$\neg\phi$	`(not` ϕ'`)`
$\phi \wedge \psi$	`(and` ϕ' ψ'`)`
$\phi \vee \psi$	`(or` ϕ' ψ'`)`
$\phi \Rightarrow \psi$	`(implies` ϕ' ψ'`)`
$\phi \Leftrightarrow \psi$	`(iff` ϕ' ψ'`)`
$\forall x.\phi$	`(forall (?x)` ϕ'`), (forall (x)` ϕ'`)`
$\exists x.\phi$	`(exists (?x)` ϕ'`), (exists (x)` ϕ'`)`

Beispiel

$\xi \in \mathcal{L}$	$\xi' \in$ Snark
$M(\text{Plato})$	`(m plato)`
$M(x)$	`(m ?x)`

$\forall x.M(x) \Rightarrow S(x)$	`(forall (?x)`
	`  (implies (m ?x) (s ?x)))`
	oder
	`(forall (x)`
	`  (implies (m x) (s x)))`

$\forall x.M(x) \Rightarrow M(\text{mutter}(x))$	`(forall (x)`
	`  (implies (m x)`
	`  (m (mutter x))))`

Variablen können immer in der Form $?x$ geschrieben werden. Bei quantifizierten Variablen kann das führende Fragezeichen auch weggelassen werden.