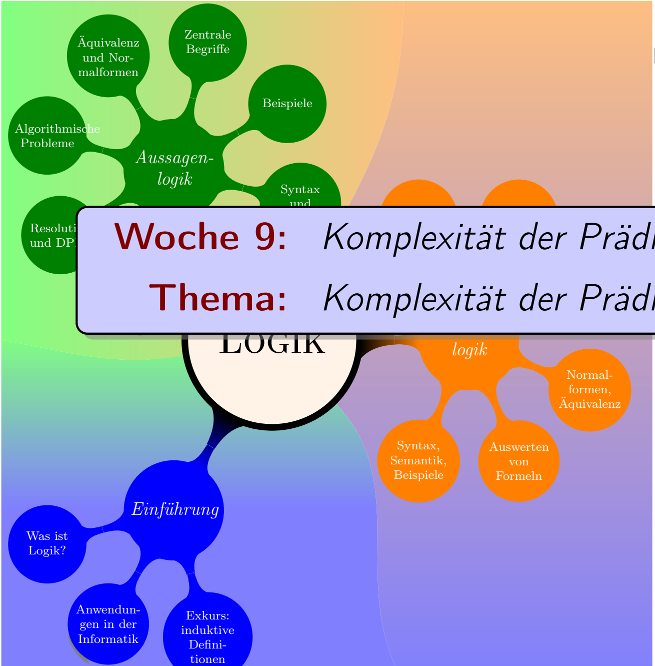
Logik im Wintersemester 2022/2023



Woche 9: *Komplexität der Prädikatenlogik*

Thema: *Komplexität der Prädikatenlogik*

Nov. 17 18 19 20 21 22 23 *Einführung*
24 25 26 27 28 29 30
31 1 2 3 4 5 6
7 8 9 10 11 12 13
Dez. 14 15 16 17 18 19 20
25 26 27
2 3 4
9 10 11
16 17 18
Jan. 19 20 21 22 23 24 25
26 27 28 29 30 31 1
2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
Feb. 23 24 25 26 27 28 29
30 31 1 2 3 4 5
6 7 8 9 10 11 12
13 14 15 16 17 18 19

Formeln mit freien Variablen vs. Sätze

$\mathcal{A}$ für welche $a \in A$ gilt:

Formeln mit freien Variablen

$(\mathcal{A}, [x/a]) \models \varphi$

$\varphi_1(x) := \forall y \, \forall z (y * z = x \to (y = 1 \lor z = 1))$

$\varphi_4(x, y) := \exists z (x * x = y + z)$

Sätze

$\varphi_2 := \forall y \exists x \, (y < x \land \varphi_1(x))$

$\varphi_3 := \forall x \forall y \forall z ((x < y \land y < z) \to x < z)$

Formeln $\varphi(x)$.

Eine Formel $\varphi(x)$ sagt etwas über ein Element innerhalb einer Struktur aus. D.h. $\varphi(x)$ beschreibt eine Eigenschaft eines Elements.

Wenn $\beta(x) = a$ eine Belegung von x ist, dann gilt $(\mathcal{A}, \beta) \models \varphi(x)$, wenn a die Eigenschaft φ hat.

Sätze ψ.

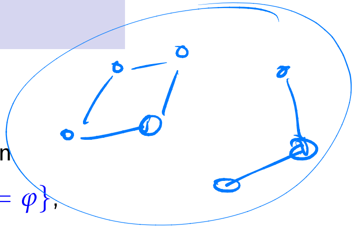
Ein Satz ψ sagt etwas über die Struktur insgesamt aus.

Ohne freie Variablen brauchen wir keine Belegung β.

D.h. $\mathcal{A} \models \psi$, wenn die Struktur die Eigenschaft ψ hat.

Modellklassen und definierbare Relationen

G



Definition (definierbare Relationen).

Sei $\mathcal{A}$ eine σ-Struktur und $\varphi(x_1, \dots, x_k) \in \text{FO}[\sigma]$. Wir definieren

$$\varphi(\mathcal{A}) := \{(a_1, \dots, a_k) \in A^k : (\mathcal{A}, [x_1/a_1, \dots, x_k/a_k]) \models \varphi\},$$

und sagen, dass φ die Relation $\varphi(\mathcal{A})$ *in $\mathcal{A}$ definiert*.

$R = \{(a,b) :$ b ist von a aus erreichbar in G$\}$

Umgekehrt nennen wir eine Relation $R \subseteq A^k$ FO-*definierbar in $\mathcal{A}$*, wenn es eine Formel $\varphi(x_1, \dots, x_k) \in \text{FO}$ gibt, so dass $\varphi(\mathcal{A}) = R$.

Definition (Modellklassen).

Sei σ eine Signatur und $\Phi \subseteq \text{FO}[\sigma]$ eine Menge von σ-Sätzen.

Die *Modellklasse* von Φ, geschrieben $\text{Mod}(\Phi)$, ist die Klasse aller σ-Strukturen $\mathcal{A}$ mit $\mathcal{A} \models \Phi$.

Falls $\Phi := \{\varphi\}$ nur einen Satz enthält, schreiben wir kurz $\text{Mod}(\varphi)$.

Auswerten prädikatenlogischer Formeln

Das Auswerten prädikatenlogischer Formeln ist viel schwerer als das Auswerten aussagenlogischer Formeln.

Top-Down Auswertung. Auswerten der Formel von „außen“ nach „innen“.

D.h. beginnend bei den äußersten Quantoren $\exists x \dots / \forall x \dots$ testen wir jede mögliche Belegung der Variablen durch.

$$\exists x \forall y (x = y \lor E(x,y))$$

Bottom-up Auswertung. Auswerten der Formel von „innen“ nach „außen“.

Beginnend bei den atomaren Formeln $\varphi(\overline{x}) := R(x_1, \dots, x_r)$ berechnen wir alle erfüllenden Variablenbelegungen, d.h. $\varphi(\mathcal{A})$.

Top-Down Auswertung prädikatenlogischer Formeln

$\mathrm{MC}(\mathcal{A}, \beta, \varphi)$.

Eingabe: endliche σ-Struktur $\mathcal{A}$, $\varphi(x_1, \ldots, x_k) \in \mathrm{FO}$
Belegung β der freien Variablen von φ.

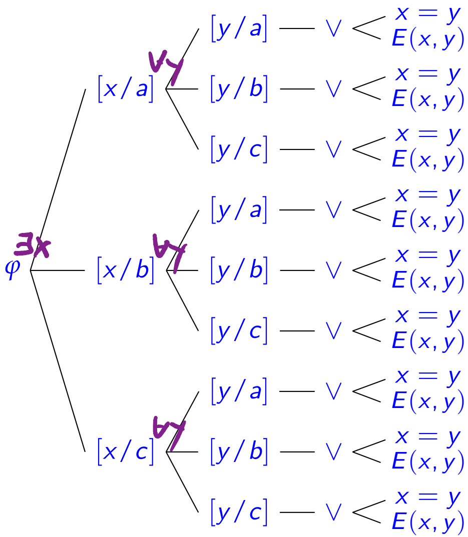
Ausgabe: 1 wenn $(\mathcal{A}, \beta) \models \varphi$, 0 sonst.

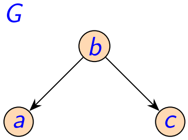
Algorithmus. Fallunterscheidung anhand des Formelaufbaus.

- $\varphi = \exists x_i \psi(x_i)$.
  for all $a \in A$:
  if $\mathrm{MC}(\mathcal{A}, \beta[x_i/a], \psi) = 1$ **then return** 1.
  return 0.
- $\varphi = \forall x_i \psi(x_i)$.
  for all $a \in A$:
  if $\mathrm{MC}(\mathcal{A}, \beta[x_i/a], \psi) = 0$ **then return** 0.
  return 1.
- $\varphi = (\varphi_1 \vee \varphi_2)$.
  return $\max\{\mathrm{MC}(\mathcal{A}, \beta, \varphi_1), \mathrm{MC}(\mathcal{A}, \beta, \varphi_2)\}$
- $\varphi = R(t_1, \ldots, t_k)$.
  für ein k-stelliges Relationssymbol $R \in \sigma$.
  Berechne $a_1 := t_1, \ldots, a_k := t_k$ in $\mathcal{A}$
  if $(a_1, \ldots, a_k) \in R^{\mathcal{A}}$ **then return** 1 **else return** 0.

(Weitere Fälle analog)

Beispiel: Top-Down Auswertung





$$\varphi := \exists x \forall y\ (x = y \vee E(x, y)).$$

Bottom-Up Auswertung prädikatenlogischer Formeln

$\mathrm{MC2}(\mathcal{A}, \varphi)$.

Eingabe: endliche σ-Struktur $\mathcal{A}$, $\varphi(x_1, \ldots, x_k) \in \mathrm{FO}$

Ausgabe: $\varphi(\mathcal{A}) = \{(a_1, \ldots, a_k) \in A^k : (\mathcal{A}, [x_1/a_1, \ldots, x_k/a_k]) \models \varphi\}$.

Hinweis. Hier auch Terme betrachten.

Algorithmus. Fallunterscheidung anhand des Formelaufbaus.

Hinweis. $\overline{x} := \mathit{frei}(\varphi_1) \vee \mathit{frei}(\varphi_2)$

- $\varphi = R(x_1, \ldots, x_k)$. für ein k-stelliges Relationssymbol $R \in \sigma$.
  Return $R^{\mathcal{A}} = \{(a_1, \ldots, a_k) \in A^k : \overline{a} \in R^{\mathcal{A}}\}$.
- $\varphi = (\varphi_1(\overline{x}) \vee \varphi_2(\overline{x}))$. **Berechne** $R_1 = \mathrm{MC2}(\mathcal{A}, \varphi_1)$ und $R_2 = \mathrm{MC2}(\mathcal{A}, \varphi_2)$
  Return $R_1 \cup R_2$

Weitere Fälle analog

- $\varphi = \exists x_i \psi(x_1, \ldots, x_i, \ldots, x_r)$. **Berechne** $R = \mathrm{MC2}(\mathcal{A}, \psi)$
  return $\{(a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_r) : (a_1, \ldots, a_r) \in R\}$.

  Bsp: $R = \{(b)\}$ i-te Stelle
- $\varphi = \forall x_i \psi(x_1, \ldots, x_i, \ldots, x_r)$. **Berechne** $R = \mathrm{MC2}(\mathcal{A}, \psi)$
  return $\{(a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_r) : (a_1, \ldots, a_{i-1}, a, a_{i+1}, \ldots, a_r) \in R$ für alle $a \in A\}$.

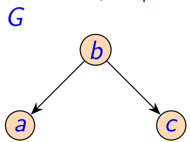
Anmerkung zu $\varphi_1(\overline{x}) \vee \varphi_2(\overline{y})$. Es ist effizienter, nur $\mathrm{MC2}(\mathcal{A}, \varphi_1(\overline{x}))$ und $\mathrm{MC2}(\mathcal{A}, \varphi_2(\overline{y}))$ auszurechnen und die Ergebnisse „sinnvoll“ zusammenzusetzen.

Beispiel: Bottom-Up Auswertung

$B(x,y) \wedge (E(x,z) \vee E(y,z))$

$(E(x,y) \vee E(y,z))$

$MC_2($ $\{x,y,z\}$



Bottom-Up Auswertung.

Unterformel	Auswertung
$x = y$	$\{(a,a),(b,b),(c,c)\}$
$E(x,y)$	$\{(b,a),(b,c)\}$
$(x = y \vee E(x,y))$	$\{(a,a),(b,b),(c,c),(b,a),(b,c)\}$
$\psi(x) := \forall y(x = y \vee E(x,y))$	$\{(b)\}$
$\varphi = \exists x \forall y(x = y \vee E(x,y))$	$\{()\}$ also gilt $G \models \varphi$

$\emptyset$

z

$\cancel{(a,b)}$

(b,a) (b,b) (b,c)

$\varphi := \exists x \forall y \, (x = y \vee E(x,y))$.

$\varphi' := \exists x \underbrace{\forall y \, (x = y \vee E(x,y))}_{\emptyset}$

Auswerten prädikatenlogischer Formeln

Das Auswerten prädikatenlogischer Formeln ist viel schwerer als das Auswerten aussagenlogischer Formeln.

Top-Down Auswertung. Auswerten der Formel von „außen“ nach „innen“.

D.h. beginnend bei den äußersten Quantoren $\exists x \ldots / \forall x \ldots$ testen wir jede mögliche Belegung der Variablen durch.

Vorteil. Es wird relativ wenig Platz benötigt.

Bottom-up Auswertung. Auswerten der Formel von „innen“ nach „außen“.

Beginnend bei den atomaren Formeln $\varphi(\overline{x}) := R(x_1, \ldots, x_r)$ berechnen wir alle erfüllenden Variablenbelegungen, d.h. $\varphi(\mathcal{A})$.

Vorteil. Wir sparen uns die vielen rekursiven Aufrufe, verbrauchen aber eventuell sehr viel Platz.

Komplexität des Auswertungsproblems

$A = \{a_1 \dots a_n\}$

Laufzeitabschätzung des top-down Algorithmus'.

Sei $\mathcal{A}$ eine Struktur mit Universum A und φ eine Formel der Länge $|\varphi|$.

Der Algorithmus durchläuft für jeden Quantor in φ alle Elemente in A.

Es ergibt sich eine Laufzeit von $|A|^{O(|\varphi|)}$.

Allerdings wird nur $O(|\varphi| \cdot |A|)$ Platz benötigt.

$O(\underline{|\varphi|} \cdot \underline{\log |A|})$

x / i

$\log n$ bits

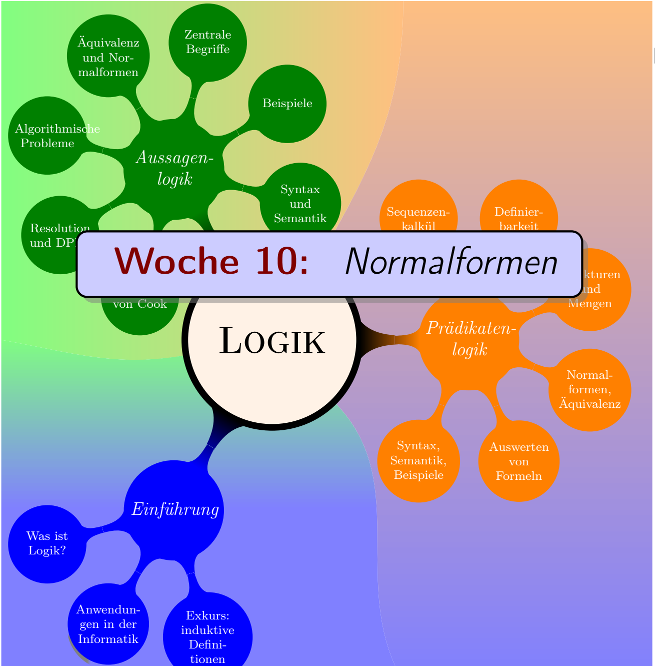
Komplexität des Auswertungsproblems.

Das Auswertungsproblem für die Prädikatenlogik ist

- lösbar in Zeit exponentiell in der Formellänge aber nur polynomiell in der Strukturgröße.
- PSPACE vollständig.

Logik im Wintersemester 2022/2023

Woche 10: *Normalformen*



Nov.	17	18	19	20	21	22	23	*Einführung*
	24	25	26	27	28	29	30	*Aussagenlogik*
	31	1	2	3	4	5	6	*Normalformen*
	7	8	9	10	11	12	13	*Resolution*
Dez.	14	15	16	17	18	19	20	*Kompaktheit*
	21	22	23	24	25	26	27	*DPLL, Satz von Cook*
	28	29	30	1	2	3	4	*Strukturen und FO*
	5	6	7	8	9	10	11	*Prädikatenlogik*
	12	13	14	15	16	17	18	*Komplexität von FO*
Jan.	19	20	21	22	23	24	25	*Weihnachten*
	26	27	28	29	30	31	1	*Neujahr*
	2	3	4	5	6	7	8	*Normalformen*
	9	10	11	12	13	14	15	*Definierbarkeit*
	16	17	18	19	20	21	22	*EF-Spiele*
Feb.	23	24	25	26	27	28	29	*EF-Spiele*
	30	31	1	2	3	4	5	*Sequenzenkalkül AL*
	6	7	8	9	10	11	12	*Sequenzenkalkül FO*
	13	14	15	16	17	18	19	*Ausblick*

10.1 Substitution

Erinnerung: Beispiele

$\sigma := \{+, *, <, 0, 1\}$: Signatur der Arithmetik

$\mathcal{A} := (\mathbb{N}, +^{\mathcal{A}}, *^{\mathcal{A}}, <^{\mathcal{A}}, 0^{\mathcal{A}}, 1^{\mathcal{A}})$: Struktur über den natürlichen Zahlen mit der üblichen Interpretation von $+, *, <, 0, 1$.

Beispiele.

- $\underline{\varphi_1(x)} := \forall y \left(\exists z \left(y * z = x \rightarrow (y = 1 \lor z = 1)\right)\right)$

- $\varphi_2 := \forall y \exists x \left(y < x \land \varphi_1(x)\right)$

  $\forall y \exists z \, (y < z \land \underline{\varphi_1(z)})$

  Anmerkung. Hier wird φ_1 in die Formel φ_2 „eingesetzt". Im allgemeinen nicht unproblematisch, siehe nächste Woche.

Formeln als Unterformeln. Sei $\varphi(x_1, x_2)$ eine Formel.

Wir wollen eine andere Formel $\psi := \ldots . \exists y \exists z \varphi(y, z) \ldots$ definieren.

Dazu wollen wir in φ die freien Variablen x_1 und x_2 durch y und z ersetzen. $\rightsquigarrow$ *Substitution*

Substitution: informell

Analog zur Aussagenlogik wollen wir einen Begriff der Substitution einführen.

Ziel ist es, Variablen sinnvoll durch *Terme* zu ersetzen.

Wenn wir z.B. in der σ_{ar}-Formel

$$\exists y \; y * y = x + x$$

die Variable x durch $(1+1)$ ersetzen, erhalten wir

$$\exists y \; y * y = (1+1) \; + \; (1+1).$$

Substitution: informell

Analog zur Aussagenlogik wollen wir einen Begriff der Substitution einführen.

Ziel ist es, Variablen sinnvoll durch *Terme* zu ersetzen.

Variablen durch Formeln zu ersetzen wäre sinnlos. Warum?

Wenn wir z.B. in der σ_{ar}-Formel

$$\exists y \; y * y = x + x$$

die Variable x durch $(1+1)$ ersetzen, erhalten wir

$$\exists y \; y * y = (1+1) \; + \; (1+1).$$

Substitution: informell

Das folgende Beispiel zeigt potentielle Probleme.

Beispiel. Sei $\varphi := \exists y\, y * y = x + x$. $\varphi(z) := \exists y \;\; y \cdot y = z + z$

1. Wenn wir in φ die freie Variable x durch y ersetzen, erhalten wir

$$\exists y\, y * y = y + y$$

was eine andere Bedeutung hat.

Wir müssen also auf Konflikte mit gebundenen Variablen achten.

Substitution: informell

Das folgende Beispiel zeigt potentielle Probleme.

Beispiel. Sei $\varphi := \exists y\, y * y = x + x$.

1. Wenn wir in φ die freie Variable x durch y ersetzen, erhalten wir

   $$\exists y\, y * y = y + y$$

   was eine andere Bedeutung hat.

   Wir müssen also auf Konflikte mit gebundenen Variablen achten.

2. Wenn wir in φ die gebundene Variable y durch x ersetzen, erhalten wir die Formel

   $$\exists x\, x * x = x + x$$

   ebenfalls mit anderer Bedeutung.

   Wir sollten daher nur freie Variablen substituieren.

Substitution

Definition. Sei σ eine Signatur.

1. Eine σ-*Substitution* ist eine Abbildung $\mathcal{S} : \mathrm{def}(\mathcal{S}) \to \mathcal{T}_\sigma$ mit endlichem Wertebereich $\mathrm{def}(\mathcal{S}) \subseteq \mathrm{Var}$.

2. Für eine Substitution $\mathcal{S}$ definieren wir $\mathrm{var}(\mathcal{S})$ als die Menge der Variablen, die in einem Term im Bild der Substitution vorkommen, d.h.

$$\mathrm{var}(\mathcal{S}) := \bigcup_{x \in \mathrm{def}(\mathcal{S})} \mathrm{var}(\mathcal{S}(x)).$$

Substitution

Beispiel. $a \longmapsto z$

$$\mathcal{S} : \begin{array}{l} x \mapsto y + z \\ y \mapsto z + v. \end{array}$$

$\text{var}(\mathcal{S}) := \{y, z, v\}$.

Definition. Sei σ eine Signatur.

1. Eine σ-*Substitution* ist eine Abbildung $\mathcal{S} : \text{def}(\mathcal{S}) \to \mathcal{T}_\sigma$ mit endlichem Wertebereich $\text{def}(\mathcal{S}) \subseteq \text{Var}$.

2. Für eine Substitution $\mathcal{S}$ definieren wir $\text{var}(\mathcal{S})$ als die Menge der Variablen, die in einem Term im Bild der Substitution vorkommen, d.h.

$$\text{var}(\mathcal{S}) := \bigcup_{x \in \text{def}(\mathcal{S})} \text{var}(\mathcal{S}(x)).$$

Substitution in Termen

Beispiel.

$$\mathcal{S} : \begin{array}{l} x \mapsto y + z \\ y \mapsto z + v. \end{array}$$

$\mathrm{var}(\mathcal{S}) := \{y, z, v\}$.

Definition. Sei $\mathcal{S}$ eine σ-Substitution.

Induktiv über die Struktur von Termen definieren wir für jeden Term $t \in \mathcal{T}_\sigma$ den Term $t\mathcal{S}$, der durch Anwendung von $\mathcal{S}$ auf t entsteht, als:

- Wenn $t := x$, wobei $x \in \mathrm{Var}$, dann
$$t\mathcal{S} := \begin{cases} \mathcal{S}(x) & \text{wenn } x \in \mathrm{def}(\mathcal{S}) \\ x & \text{sonst.} \end{cases}$$

$\exists z\,(x + z = y + z)$

$\exists a\,(x + a = y + a)$

$\longrightarrow \exists a\,((y+z) + a = (z+v) + a)$

- Wenn $t := c$, für ein Konstantensymbol $c \in \sigma$, dann $t\mathcal{S} := c$.

- Wenn $t := f(t_1, \ldots, t_k)$, für ein k-stelliges Funktionssymbol $f \in \sigma$ und σ-Terme $t_1, \ldots, t_k \in \mathcal{T}_\sigma$, dann $t\mathcal{S} := f(t_1\mathcal{S}, \ldots, t_k\mathcal{S})$.

$x \mapsto y$

$y \mapsto x$

Substitution in Termen

Beispiel.

$$\mathcal{S} : \begin{array}{l} x \mapsto y + z \\ y \mapsto z + v. \end{array}$$

$\text{var}(\mathcal{S}) := \{y, z, v\}$.

Für $t := x + y$ gilt

$t\mathcal{S} := \big((y + z) + (z + v)\big)$.

Definition. Sei $\mathcal{S}$ eine σ-Substitution.

Induktiv über die Struktur von Termen definieren wir für jeden Term $t \in \mathcal{T}_\sigma$ den Term $t\mathcal{S}$, der durch Anwendung von $\mathcal{S}$ auf t entsteht, als:

- Wenn $t := x$, wobei $x \in \text{Var}$, dann
  $$t\mathcal{S} := \begin{cases} \mathcal{S}(x) & \text{wenn } x \in \text{def}(\mathcal{S}) \\ x & \text{sonst.} \end{cases}$$
- Wenn $t := c$, für ein Konstantensymbol $c \in \sigma$, dann $t\mathcal{S} := c$.
- Wenn $t := f(t_1, \dots, t_k)$, für ein k-stelliges Funktionssymbol $f \in \sigma$ und σ-Terme $t_1, \dots, t_k \in \mathcal{T}_\sigma$, dann $t\mathcal{S} := f(t_1\mathcal{S}, \dots, t_k\mathcal{S})$.

Substitution in Formeln

Definition. Sei $\mathcal{S}$ eine σ-Substitution.
Induktiv definieren wir für $\varphi \in \mathrm{FO}[\sigma]$ die Formel $\varphi\mathcal{S}$ als:

- Für $\varphi := R(t_1, \dots, t_k)$ gilt $\varphi\mathcal{S} := R(t_1\mathcal{S}, \dots, t_k\mathcal{S})$.
  (wobei $t_1, \dots, t_k \in \mathcal{T}_\sigma$, R k-stell. Relationssymbol)
- Für $\varphi := t_1 = t_2$ gilt $\varphi\mathcal{S} := t_1\mathcal{S} = t_2\mathcal{S}$ (wobei $t_1, t_2 \in T_\sigma$).
- Für $\varphi := \neg\psi$ gilt $\varphi\mathcal{S} := \neg\, \psi\mathcal{S}$.
- Für $\varphi := (\psi_1 * \psi_2)$ gilt $\varphi\mathcal{S} := (\psi_1\mathcal{S} * \psi_2\mathcal{S})$
  (wobei $\psi_1, \psi_2 \in \mathrm{FO}[\sigma]$ und $* \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$).
- Wenn $\varphi := \exists x\psi$, wobei $x \in \mathrm{Var}$ und $\psi \in \mathrm{FO}[\sigma]$, dann gilt:
  1. $\varphi\mathcal{S} := \exists x\psi\mathcal{S}'$, falls $x \notin \mathrm{var}(\mathcal{S})$, wobei $\mathcal{S}' := \mathcal{S}_{|\mathrm{def}(\mathcal{S})\setminus\{x\}}$.
  2. Wenn $x \in \mathrm{var}(\mathcal{S})$, wähle $y \in \mathrm{Var} \setminus (\mathit{frei}(\varphi) \cup \mathrm{var}(\mathcal{S}))$ und setze $\varphi\mathcal{S} := \exists y\psi\mathcal{S}'$, wobei $\mathcal{S}' := \mathcal{S}_{\mathrm{def}(\mathcal{S})\setminus\{x\}} \cup \{x \mapsto y\}$.
- Der Fall $\varphi := \forall x\psi$ ist analog.

Beispiel.
$\mathcal{S}: \begin{array}{l} x \mapsto y + z \\ y \mapsto z + v. \end{array}$
$\mathrm{var}(\mathcal{S}) := \{y, z, v\}$.

Für $t := x + y$ gilt
$t\mathcal{S} := ((y + z) + (z + v))$.

$(\exists x\, \psi)S =$
$\exists x\ \psi S'$
$S': y \mapsto z+v$

$(\exists y\ \psi)S$ $\quad S': x \mapsto y+z$
$\rightarrow \exists a\ \psi S'$ $\quad$ ~~$y \mapsto z+v$~~
$y \mapsto a$

Substitution in Formeln

Definition. Sei $\mathcal{S}$ eine σ-Substitution.

Induktiv definieren wir für $\varphi \in \mathrm{FO}[\sigma]$ die Formel $\varphi\mathcal{S}$ als:

- Für $\varphi := R(t_1, \dots, t_k)$ gilt $\varphi\mathcal{S} := R(t_1\mathcal{S}, \dots, t_k\mathcal{S})$.
  (wobei $t_1, \dots, t_k \in \mathcal{T}_\sigma$, R k-stell. Relationssymbol)
- Für $\varphi := t_1 = t_2$ gilt $\varphi\mathcal{S} := t_1\mathcal{S} = t_2\mathcal{S}$ (wobei $t_1, t_2 \in T_\sigma$).
- Für $\varphi := \neg\psi$ gilt $\varphi\mathcal{S} := \neg\, \psi\mathcal{S}$.
- Für $\varphi := (\psi_1 * \psi_2)$ gilt $\varphi\mathcal{S} := (\psi_1\mathcal{S} * \psi_2\mathcal{S})$
  (wobei $\psi_1, \psi_2 \in \mathrm{FO}[\sigma]$ und $* \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$).
- Wenn $\varphi := \exists x\psi$, wobei $x \in \mathrm{Var}$ und $\psi \in \mathrm{FO}[\sigma]$, dann gilt:
  1. $\varphi\mathcal{S} := \exists x\psi\mathcal{S}'$, falls $x \notin \mathrm{var}(\mathcal{S})$, wobei $\mathcal{S}' := \mathcal{S}_{|\mathrm{def}(\mathcal{S})\setminus\{x\}}$.
  2. Wenn $x \in \mathrm{var}(\mathcal{S})$, wähle $y \in \mathrm{Var} \setminus (\mathit{frei}(\varphi) \cup \mathrm{var}(\mathcal{S}))$ und setze $\varphi\mathcal{S} := \exists y\psi\mathcal{S}'$, wobei $\underline{\mathcal{S}'} := \mathcal{S}_{\mathrm{def}(\mathcal{S})\setminus\{x\}} \cup \{x \mapsto y\}$.
- Der Fall $\varphi := \forall x\psi$ ist analog.

Beispiel.

$$\mathcal{S} : \begin{array}{l} x \mapsto y + z \\ y \mapsto z + v. \end{array}$$

$\mathrm{var}(\mathcal{S}) := \{y, z, v\}$.

Für $t := x + y$ gilt

$t\mathcal{S} := ((y + z) + (z + v))$.

Sei $\varphi := \exists a\forall z(x + y + z = a + x)$:

$\varphi\mathcal{S} := \exists a\big(\forall z(x + y + z = a + x)\mathcal{S}\big)$

$= \exists a\ \forall v_0 (x + y + z = a + x)\mathcal{S}'$

$\mathcal{S}' := \mathcal{S} \cup \{z \mapsto v_0\}$

$= \exists a\ \forall v_0\ (x + y + z)\mathcal{S}' = (a + x)\mathcal{S}'$

$= \exists a\ \forall v_0\ ((y + z) + (z + v) + v_0$

$= (a + (y + z)))$.

$\underline{\forall z} (\exists a (\quad))\mathcal{S}'$

$\mathcal{S}'$: $z \mapsto a$, $x \mapsto y+z$, $y \mapsto z+v$

$\mathcal{S}''$: $z \mapsto a$, $x, y \mapsto \dots$, $a \mapsto v_0$

Notation

Notation.

- Analog zur Aussagenlogik schreiben wir für eine Substitution $\mathcal{S}$ mit $\mathrm{def}(\mathcal{S}) := \{x_1, \dots, x_n\}$ und $\mathcal{S}(x_i) := t_i$, $1 \leq i \leq n$,

  $$[x_1/t_1, \dots, x_n/t_n].$$

  Das erlaubt uns, $\varphi[x_1/t_1, \dots, x_n/t_n]$ statt $\varphi\mathcal{S}$ zu schreiben.

Notation

Notation.

- Analog zur Aussagenlogik schreiben wir für eine Substitution $\mathcal{S}$ mit $\text{def}(\mathcal{S}) := \{x_1, \dots, x_n\}$ und $\mathcal{S}(x_i) := t_i$, $1 \leq i \leq n$,

$$[x_1 / t_1, \dots, x_n / t_n].$$

  Das erlaubt uns, $\varphi[x_1 / t_1, \dots, x_n / t_n]$ statt $\varphi\mathcal{S}$ zu schreiben.
- Für $\varphi(x_1, \dots, x_k) \in \text{FO}$ und $t_1, \dots, t_k \in \mathcal{T}_\sigma$ schreiben wir

$$\varphi[t_1, \dots, t_k] \quad \text{statt} \quad \varphi[x_1 / t_1, \dots, x_k / t_k].$$

Notation

Notation.

- Analog zur Aussagenlogik schreiben wir für eine Substitution $\mathcal{S}$ mit $\mathrm{def}(\mathcal{S}) := \{x_1, \dots, x_n\}$ und $\mathcal{S}(x_i) := t_i$, $1 \leq i \leq n$,

$$[x_1 / t_1, \dots, x_n / t_n].$$

  Das erlaubt uns, $\varphi[x_1 / t_1, \dots, x_n / t_n]$ statt $\varphi\mathcal{S}$ zu schreiben.
- Für $\varphi(x_1, \dots, x_k) \in \mathrm{FO}$ und $t_1, \dots, t_k \in \mathcal{T}_\sigma$ schreiben wir

$$\varphi[t_1, \dots, t_k] \quad \text{statt} \quad \varphi[x_1 / t_1, \dots, x_k / t_k].$$

Vergleiche mit Methoden in Java.

```
Boolean phi(int x1, ..., int xk)
```

Indem wir $x_1, \dots, x_k$ spezifizieren, fixieren wir eine Ordnung der Parameter.

```
Boolean b = phi(3, 5, ..., 17);
```

Das Substitutionslemma

Lemma. Sei $\mathcal{S}$ eine σ-Substitution. Für alle σ-Formeln φ, ψ:

$$\varphi \equiv \psi \quad \Longrightarrow \quad \varphi\mathcal{S} \equiv \psi\mathcal{S}$$

Das Substitutionslemma

Lemma. Sei $\mathcal{S}$ eine σ-Substitution. Für alle σ-Formeln φ, ψ:

$$\varphi \equiv \psi \quad \Longrightarrow \quad \varphi\mathcal{S} \equiv \psi\mathcal{S}$$

$\psi := \dots \vartheta$

$\psi' := \dots \varphi \dots$

Lemma (Ersetzungslemma).

Sei τ eine Signatur und seien $\varphi, \psi, \vartheta \in \mathrm{FO}[\tau]$.

Sei ϑ eine Teilformel von ψ und $\vartheta \equiv \varphi$. Ferner, sei ψ' die Formel, die aus ψ entsteht, indem ϑ durch φ ersetzt wird.

Dann gilt $\psi \equiv \psi'$.

Beispiele

$\sigma := \{+, *, <, 0, 1\}$: Signatur der Arithmetik

$\mathcal{A} := (\mathbb{N}, +^{\mathcal{A}}, *^{\mathcal{A}}, <^{\mathcal{A}}, 0^{\mathcal{A}}, 1^{\mathcal{A}})$: Struktur über den natürlichen Zahlen mit der üblichen Interpretation von $+, *, <, 0, 1$

Beispiele.

- $\varphi_1(x) := \forall y\, (\exists z\, (y * z = x \rightarrow (y = 1 \vee z = 1)))$

$\forall y \exists z\, (y < x \wedge \varphi_1[z])$

- $\varphi_2 := \forall y \exists x\, (y < x \wedge \varphi_1(x))$

  Anmerkung. Hier wird φ_1 in die Formel φ_2 „eingesetzt". Im allgemeinen nicht unproblematisch, siehe nächste Woche.

$\varphi_1[x/z]$

$\varphi_1 S$

$S: x \mapsto z$

Unterformeln

Mit Hilfe der Substitution können wir nun Unterformeln benutzen.

Beispiele

$\sigma := \{+, *, <, 0, 1\}$: Signatur der Arithmetik

$\mathcal{A} := (\mathbb{N}, +^{\mathcal{A}}, *^{\mathcal{A}}, <^{\mathcal{A}}, 0^{\mathcal{A}}, 1^{\mathcal{A}})$: Struktur über den natürlichen Zahlen mit der üblichen Interpretation von $+, *, <, 0, 1$

Beispiele.

- $\varphi_1(x) := \forall y \, (\exists z \, (y * z = x \rightarrow (y = 1 \lor z = 1)))$

- $\varphi_2 := \forall y \exists x \, (y < x \land \varphi_1(x))$ $\qquad \exists z (\varphi_1(z) \land z * z = x)$

  Anmerkung. Hier wird φ_1 in die Formel φ_2 „eingesetzt". Im allgemeinen nicht unproblematisch, siehe nächste Woche.

Unterformeln

Mit Hilfe der Substitution können wir nun Unterformeln benutzen.