Aufgabenstellungen: *Teil_03* [*http://www.uni-forst.gwdg.de/~wkurth/fs20_u03.pdf*](http://www.uni-forst.gwdg.de/~wkurth/fs20_u03.pdf)

auch unten auf den Folien, vor jeweiligen Lösungen

***Kurze Vorschau** auf die heutige Sitzung:*

Prädikatenlogik

etwas bearbeitete Version → ***Unifikationsalgorithmus von Robinson, Originalfassung, 1965***

```
Eingabe: nicht-leere Literalmenge L
sub := []
while |L sub| > 1 do
    begin
    durchsuche Literale in L sub von links nach rechts bis zur ersten Position, an der
    sich zwei Literale unterscheiden;
    if keines der sich unterscheidenden Zeichen ist Variable then stoppe mit "fail"
    else    begin
            Sei x die Variable und t der Term, der im anderen Literal beginnt;
            if x kommt in t vor then stoppe mit "fail" (occur check)
            else sub := sub [x/t] (Hintereinanderausführung der Substitutionen)
            end;
    end;
gib mgu sub aus;
```

```
function UNIFY(t1, t2) ⟹ (unifiable: Boolean, σ: substitution)
begin
  if t1 or t2 is a variable then
    begin
      let x be the variable, and let t be the other term
      if x = t, then (unifiable, σ) ← (true, ∅)
      else if occur(x, t) then unifiable ← false
      else(unifiable, σ) ← (true, {x ← t})
    end
  else
    begin
      assume t1 = f(x_1, ..., x_n) and t2 = g(y_1, ..., y_m)
      if f ≠ g or m ≠ n then unifiable ← false
      else
        begin
          k ← 0
          unifiable ← true
          σ ← nil
          while k < m and unifiable do
            begin
              k ← k + 1
              (unifiable, τ) ← UNIFY(σ(x_k), σ(y_k))
              if unifiable then σ ← compose(τ, σ)
            end
        end
    end
  return (unifiable, σ)
end
```

FunFact: steht für "nicht in der Liste" (leere Liste) – "null"-Analogon für LISP

***Die Schlüsselbegriffe** der heutigen Sitzung:*

- Term
- frei/gebunden
- allgemeinster Unifikator aka MGU
- Pränex-NF
- Skolem-NF
- Hornformel
- PROLOG

Inhaltsverzeichnis

Aufgabe T3.1: *Hornformel & Markierungsalgorithmus*

Ein Bäcker möchte Rosinenbrötchen backen. Leider hat er nicht alle benötigten Zutaten: Ihm fehlen die Rosinen. Er kann jedoch einige vorhandene Zutaten gegen andere tauschen, und zwar:

Mehl + Eier → Milch + Honig
Mandeln + Honig → Rosinen
Milch + Hefe → Mandeln.

Vorhanden sind Mehl, Eier und Hefe in großer Menge.

(a) Übersetzen Sie die Aussagen in eine Hornformel.

(b) Überprüfen Sie mit Hilfe des Markierungsalgorithmus (s. Schluss von Teil 1 des Skripts), ob der Bäcker die Rosinen erhalten kann. Geben Sie dabei an, in welchem Schritt Sie welche Aussagen markieren.
(Hinweis: Zu beweisen ist, dass die Verfügbarkeit der Rosinen eine Folgerung aus den oben genannten Aussagen ist. Übersetzen Sie die Frage, ob der Bäcker die Rosinen bekommt, in ein Unerfüllbarkeitsproblem.)

Aufgabe T3.1

Lösung: (a)

Schritt #1: Umbenennen der Bausteine

Symbol	Aussage	Symbol	Aussage
$\boldsymbol{M}$	**Mehl ist verfügbar**	$\boldsymbol{A}$	**Mandeln sind verfügbar**
$\boldsymbol{E}$	**Eier sind vorhanden**	$\boldsymbol{R}$	**Rosinen sind vorhanden**
$\boldsymbol{I}$	**Milch ist verfügbar**	$\boldsymbol{F}$	**Hefe ist verfügbar**
$\boldsymbol{H}$	**Honig ist vorhanden**		

Schritt #2: Übersetzung der Aussagen in Horn-Formeln (Implikationsschreibweise)

Teilschritt #1: erste Aussage

#	Kommentar	Vereinfachungsschritt
0		$\boldsymbol{B} \overset{\text{def}}{=} \boldsymbol{M \land E \longrightarrow I \land H}$
1		$\equiv \neg(\boldsymbol{M} \land \boldsymbol{E}) \lor (\boldsymbol{I} \land \boldsymbol{H})$
2		$\equiv (\neg(\boldsymbol{M} \land \boldsymbol{E}) \lor \boldsymbol{I}) \land (\neg(\boldsymbol{M} \land \boldsymbol{E}) \lor \boldsymbol{H})$
3	zwei definite Horn-Klauseln	$\equiv \boldsymbol{(M \land E \longrightarrow I)} \land \boldsymbol{(M \land E \longrightarrow H)}$

Teilschritt #2: restliche Aussagen

zweite	$\boldsymbol{C} \overset{\text{def}}{=} \boldsymbol{A \land H \longrightarrow R}$	definite Horn-Klausel
dritte	$\boldsymbol{D} \overset{\text{def}}{=} \boldsymbol{I \land F \longrightarrow A}$	definite Horn-Klausel
weitere [Faktenklauseln]	$\boldsymbol{M, E, F} \equiv \boldsymbol{M \land E \land F} \equiv \boldsymbol{(1 \longrightarrow M) \land (1 \longrightarrow E) \land (1 \longrightarrow F)} \overset{\text{def}}{=} \boldsymbol{G}$	

Aufgabe T3.1

Lösung: (b)

Schritt #3: Formulieren des Problems (Verfügbarkeit von Rosinen)

$$B, C, D, G \vDash R \Leftrightarrow \ \vDash B \wedge C \wedge D \wedge G \longrightarrow R \Leftrightarrow \ \vDash \neg(B \wedge C \wedge D \wedge G \wedge \neg R) \Leftrightarrow \ \nvDash B \wedge C \wedge D \wedge G \wedge \neg R$$

Schritt #4: Überprüfung auf Erfüllbarkeit mithilfe des Polynomialzeit-Markierungsverfahrens

QuickFact: markiere heißt „belege zwangsmäßig mit **1** aka WAHR“

$$B \wedge C \wedge D \wedge G \wedge \neg R \equiv (M \wedge E \longrightarrow I) \wedge (M \wedge E \longrightarrow H) \wedge (A \wedge H \longrightarrow R) \wedge (I \wedge F \longrightarrow A) \wedge (1 \longrightarrow M) \wedge (1 \longrightarrow E) \wedge (1 \longrightarrow F) \wedge (R \longrightarrow 0)$$

markiere alle ursprünglichen Fakten, d.h. M, E, F

$$(M \wedge E \longrightarrow I) \wedge (M \wedge E \longrightarrow H) \wedge (A \wedge H \longrightarrow R) \wedge (I \wedge F \longrightarrow A) \wedge (1 \longrightarrow M) \wedge (1 \longrightarrow E) \wedge (1 \longrightarrow F) \wedge (R \longrightarrow 0)$$

$$\equiv (1 \wedge 1 \longrightarrow I) \wedge (1 \wedge 1 \longrightarrow H) \wedge (A \wedge H \longrightarrow R) \wedge (I \wedge 1 \longrightarrow A) \wedge (1 \longrightarrow 1) \wedge (1 \longrightarrow 1) \wedge (1 \longrightarrow 1) \wedge (R \longrightarrow 0)$$

$$\equiv (1 \longrightarrow I) \wedge (1 \longrightarrow H) \wedge (A \wedge H \longrightarrow R) \wedge (I \longrightarrow A) \wedge 1 \wedge 1 \wedge 1 \wedge (R \longrightarrow 0)$$

$$\equiv (1 \longrightarrow I) \wedge (1 \longrightarrow H) \wedge (A \wedge H \longrightarrow R) \wedge (I \longrightarrow A) \wedge (R \longrightarrow 0)$$

markiere alle frischgebackenen ☺ Fakten (nach dem 1. Lauf), d.h. I, H

$$(M \wedge E \longrightarrow I) \wedge (M \wedge E \longrightarrow H) \wedge (A \wedge H \longrightarrow R) \wedge (I \wedge F \longrightarrow A) \wedge (1 \longrightarrow M) \wedge (1 \longrightarrow E) \wedge (1 \longrightarrow F) \wedge (R \longrightarrow 0)$$

$$\equiv (1 \longrightarrow 1) \wedge (1 \longrightarrow 1) \wedge (A \wedge 1 \longrightarrow R) \wedge (1 \longrightarrow A) \wedge (R \longrightarrow 0)$$

$$\equiv 1 \wedge 1 \wedge (A \longrightarrow R) \wedge (1 \longrightarrow A) \wedge (R \longrightarrow 0)$$

$$\equiv (A \longrightarrow R) \wedge (1 \longrightarrow A) \wedge (R \longrightarrow 0)$$

markiere alle Fakten (nach dem 2. Lauf), d.h. A

$$(M \wedge E \longrightarrow I) \wedge (M \wedge E \longrightarrow H) \wedge (A \wedge H \longrightarrow R) \wedge (I \wedge F \longrightarrow A) \wedge (1 \longrightarrow M) \wedge (1 \longrightarrow E) \wedge (1 \longrightarrow F) \wedge (R \longrightarrow 0)$$

$$\equiv (1 \longrightarrow R) \wedge (1 \longrightarrow 1) \wedge (R \longrightarrow 0)$$

$$\equiv (1 \longrightarrow R) \wedge 1 \wedge (R \longrightarrow 0)$$

$$\equiv (1 \longrightarrow R) \wedge (R \longrightarrow 0)$$

Anmerkung: **schattierte Felder** gehören nicht zum Markierungsalgorithmus – die dienen lediglich dem besseren Verständnis der vorgenommenen Schritte

Aufgabe T3.1

Lösung: (b)

markiere alle Fakten (nach dem 3. Lauf), d.h. $\boldsymbol{R}$
$(\boldsymbol{M} \wedge \boldsymbol{E} \longrightarrow \boldsymbol{I}) \wedge (\boldsymbol{M} \wedge \boldsymbol{E} \longrightarrow \boldsymbol{H}) \wedge (\boldsymbol{A} \wedge \boldsymbol{H} \longrightarrow \boldsymbol{R}) \wedge (\boldsymbol{I} \wedge \boldsymbol{F} \longrightarrow \boldsymbol{A}) \wedge (\mathbf{1} \longrightarrow \boldsymbol{M}) \wedge (\mathbf{1} \longrightarrow \boldsymbol{E}) \wedge (\mathbf{1} \longrightarrow \boldsymbol{F}) \wedge (\boldsymbol{R} \longrightarrow \mathbf{0})$
$\equiv (1 \longrightarrow 1) \wedge (1 \longrightarrow 0)$ $\equiv 1 \wedge 0$ $\equiv \mathbf{0}$

$\boldsymbol{B} \wedge \boldsymbol{C} \wedge \boldsymbol{D} \wedge \boldsymbol{G} \wedge \neg \boldsymbol{R}$ ist unerfüllbar $\Longleftrightarrow$ $\not\models \boldsymbol{B} \wedge \boldsymbol{C} \wedge \boldsymbol{D} \wedge \boldsymbol{G} \wedge \neg \boldsymbol{R}$

Schritt #5: Rückübersetzung der Lösung

Rosinen sind verfügbar aka der Bäcker kann die Rosinen erhalten

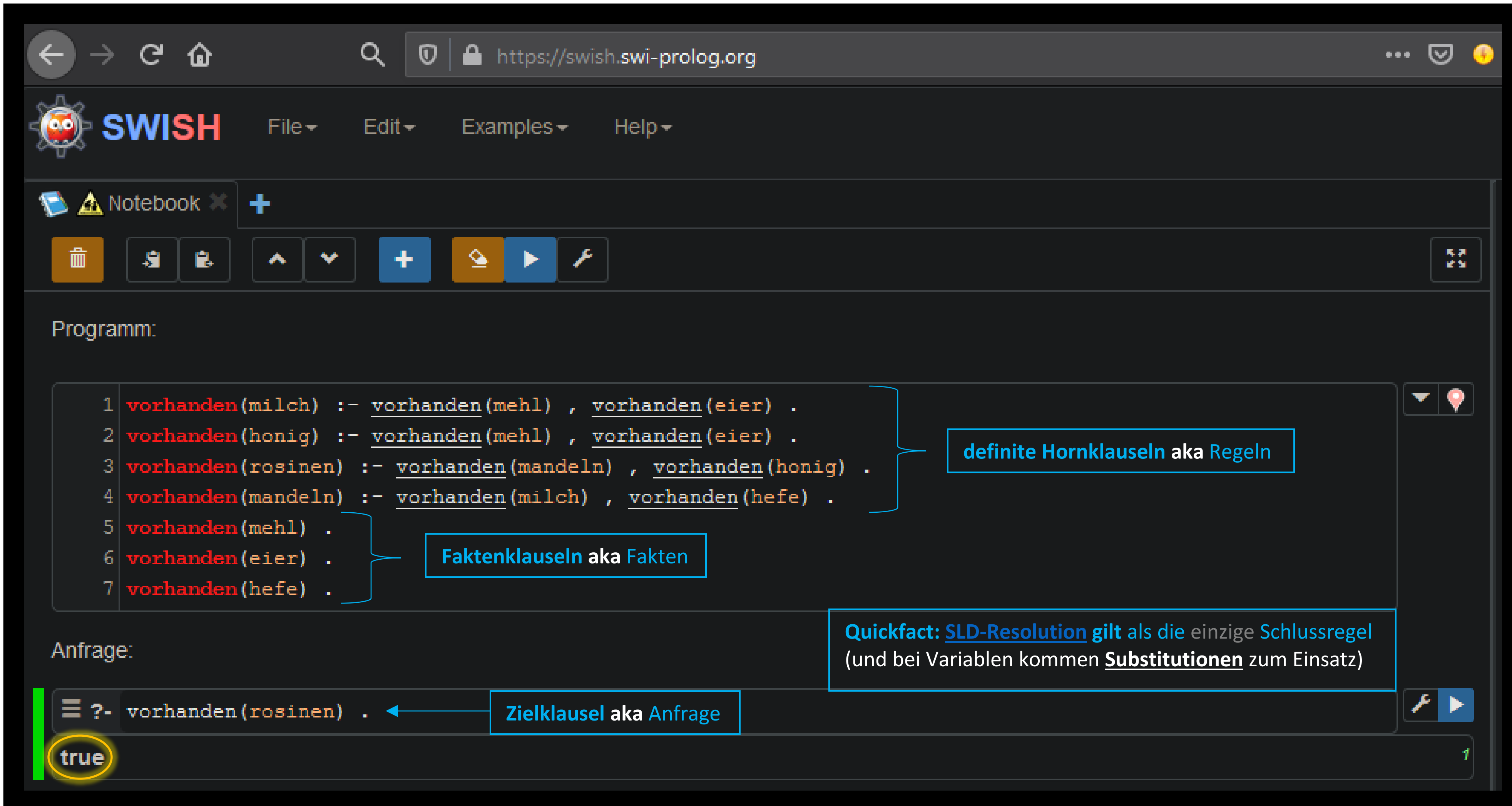
---

Aufgabe T3.2: *PROLOG auf Horn-Basis*

Recherchieren Sie im Internet die Grundlagen der PROLOG-Syntax. Schreiben Sie dann zu Aufgabe 1.1 ein PROLOG-Programm. Verwenden Sie das Prädikat "`vorhanden`".
Wie wird die Antwort auf die Anfrage "`?- vorhanden(rosinen)`" lauten?

Aufgabe T3.2

Lösung:



Aufgabe T3.3: *Terme, Formeln, frei/gebunden*

Sind die folgenden Zeichenketten Terme oder Formeln der Prädikatenlogik erster Stufe? Welche Vorkommen welcher Variablen sind frei und welche gebunden?

a) $\exists x \, \forall y (g(f(y), f(z)) \rightarrow y)$

b) $\exists x \exists y (p(f(x)) \rightarrow q(y, g(z)))$

Schreibweise:	u, v, w, x, y, z	Variablen
	a, b, c	Konstanten
	f, g, h	Funktionssymbole
	p, q, r	Prädikatsymbole

Aufgabe T3.3

Lösung: (a)

Schritt #1: Test auf Zugehörigkeit einer Zeichenkette zur Menge $\boldsymbol{\mathcal{T}(\Sigma)}$ aller Terme über eine Signatur $\Sigma = \mathbb{K} \cup \mathbb{F} \cup \mathbb{P}$

Laut der induktiven (aka rekursiven) Definition von Termen:

$$\exists x \forall y\big(g\big(f(y), f(z)\big) \longrightarrow y\big) \in \mathcal{T}(\Sigma) \Leftrightarrow \exists \in \mathbb{F} \text{ und } x\forall y\big(g\big(f(y), f(z)\big) \longrightarrow y\big) \in \mathcal{T}(\Sigma) \xrightarrow{yields} \boldsymbol{falsch}$$

$$\exists x \forall y\big(g\big(f(y), f(z)\big) \longrightarrow y\big) \notin \mathcal{T}(\Sigma)$$

Schritt #2: Test auf Zugehörigkeit einer Zeichenkette zur Menge $\boldsymbol{\mathcal{F}(\Sigma)}$ aller Formeln über eine Signatur Σ

Laut der induktiven (aka rekursiven) Definition von Formeln:

$$\exists x \forall y\big(g\big(f(y), f(z)\big) \longrightarrow y\big) \in \mathcal{F}(\Sigma) \Leftrightarrow \forall y\big(g\big(f(y), f(z)\big) \longrightarrow y\big) \in \mathcal{F}(\Sigma) \Leftrightarrow g\big(f(y), f(z)\big) \longrightarrow y \in \mathcal{F}(\Sigma)$$

$$\Leftrightarrow g\big(f(y), f(z)\big) \in \mathcal{F}(\Sigma) \text{ und } y \in \mathcal{F}(\Sigma) \xrightarrow{yields} \boldsymbol{falsch}$$

$$\exists x \forall y\big(g\big(f(y), f(z)\big) \longrightarrow y\big) \notin \mathcal{F}(\Sigma)$$

Schritt #3: gebundene (aka nicht-freie) / freie (aka nicht-gebundene) Variablenvorkommen

Es wird **keine** Analyse benötigt **(da man keine gültige Formel hat)**.

Aufgabe T3.3

Lösung: (b)

Schritt #1: Test auf Zugehörigkeit einer Zeichenkette zur Menge $\mathcal{T}(\Sigma)$ aller Terme über eine Signatur $\Sigma = \mathbb{K} \cup \mathbb{F} \cup \mathbb{P}$

Laut der induktiven (aka rekursiven) Definition von Termen:

$$\exists x \exists y \left(p(f(x)) \rightarrow q(y, g(z))\right) \in \mathcal{T}(\Sigma) \Leftrightarrow \exists \in \mathbb{F} \text{ und } x\exists y \left(p(f(x)) \rightarrow q(y, g(z))\right) \in \mathcal{T}(\Sigma) \xrightarrow{yields} \boldsymbol{falsch}$$

$$\exists x \exists y \left(p(f(x)) \rightarrow q(y, g(z))\right) \notin \mathcal{T}(\Sigma)$$

Schritt #2: Test auf Zugehörigkeit einer Zeichenkette zur Menge $\mathcal{F}(\Sigma)$ aller Formeln über eine Signatur Σ

Laut der induktiven (aka rekursiven) Definition von Formeln:

$$\exists x \exists y \left(p(f(x)) \rightarrow q(y, g(z))\right) \in \mathcal{F}(\Sigma) \Leftrightarrow \exists y \left(p(f(x)) \rightarrow q(y, g(z))\right) \in \mathcal{F}(\Sigma) \Leftrightarrow p(f(x)) \rightarrow q(y, g(z)) \in \mathcal{F}(\Sigma)$$

$$\Leftrightarrow p(f(x)) \in \mathcal{F}(\Sigma) \text{ und } q(y, g(z)) \in \mathcal{F}(\Sigma) \Leftrightarrow p \in \mathbb{P} \text{ und } f(x) \in \mathcal{T}(\Sigma) \text{ und } q \in \mathbb{P} \text{ und } y, g(z) \in \mathcal{T}(\Sigma) \xrightarrow{yields} \boldsymbol{wahr}$$

$$\exists x \exists y \left(p(f(x)) \rightarrow q(y, g(z))\right) \in \mathcal{F}(\Sigma)$$

Schritt #3: gebundene (aka nicht-freie) / freie (aka nicht-gebundene) Variablenvorkommen

FunFact: falls ein Vorkommen einer Variable im Skopus mehrerer Quantoren liegt, die die gleiche Variable quantifizieren, dann gilt es **ausschließlich** im Bezug auf den nächsten Quantor als **gebunden**

#	Vorkommnis	Gebunden?	Kommentar
1	$\exists \boldsymbol{x} \exists y \left(p(f(x)) \rightarrow q(y, g(z))\right)$	**Ja**	steht **unmittelbar** nach dem Quantorzeichen
2	$\exists x \exists \boldsymbol{y} \left(p(f(x)) \rightarrow q(y, g(z))\right)$	**Ja**	steht **unmittelbar** nach dem Quantorzeichen
3	$\exists x \exists y \left(p(f(\boldsymbol{x})) \rightarrow q(y, g(z))\right)$	**Ja**	liegt im **Bereich** (aka Skopus aka Bindungs-/Geltungsbereich) des ExistenzQuantors $\exists x$
4	$\exists x \exists y \left(p(f(x)) \rightarrow q(\boldsymbol{y}, g(z))\right)$	**Ja**	liegt im **Bereich** des ExistenzQuantors $\exists y$
5	$\exists x \exists y \left(p(f(x)) \rightarrow q(y, g(\boldsymbol{z}))\right)$	**Nein**	die Variable z wird gar nicht quantifiziert

Aufgabe T3.4: *Teilformeln, Terme & frei/gebunden*

Geben Sie sämtliche Teilformeln und Terme an, die in der Formel

$F = \neg(\forall x(p(x, y) \rightarrow q(f(x)))) \wedge \forall x \forall y\, p(x, f(y))$

enthalten sind. Bestimmen Sie für jedes Vorkommen einer Variablen, ob es frei oder gebunden ist.

Schreibweise:	u, v, w, x, y, z	Variablen
	a, b, c	Konstanten
	f, g, h	Funktionssymbole
	p, q, r	Prädikatsymbole

Aufgabe T3.4

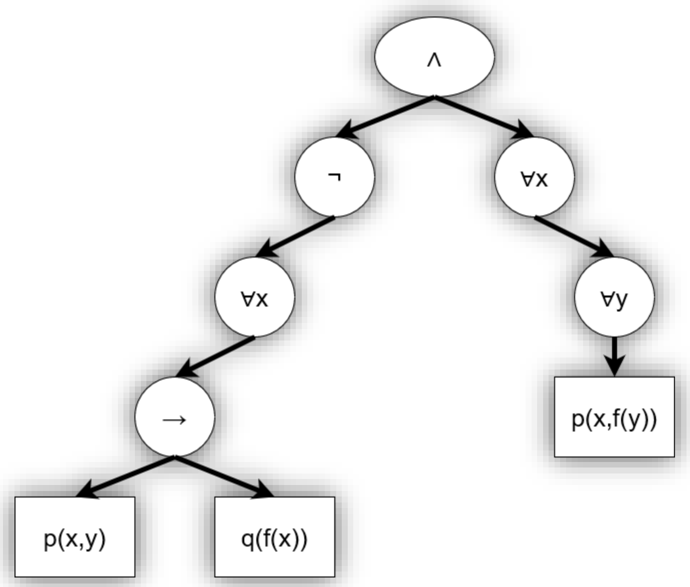
Lösung

Schritt #1: Teilformeln am besten durch „Baumschütteln“ ☺ ernten

Anmerkung: untere Tabelle ist kein Muss - sie hilft allerdings bei der Fehlervermeidung (durch ihren systematischen Ansatz)

$$\neg\Big(\forall x\big(p(x,y)\longrightarrow q(f(x))\big)\Big)\land\forall x\forall y p(x,f(y))$$

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
¬	(	∀	x	(	p	(	x	,	y	)	→	q	(	f	(	x	)	)	)	)	∧	∀	x	∀	y	p	(	x	,	f	(	y	)	)
1		2			3						4	5									6	7		8		9								
0	1	1	1	2	2	3	3	3	3	-3	2	2	3	3	4	4	-4	-3	-2	-1	0	0	0	0	0	0	1	1	1	1	2	2	-2	-1
II		III			V						IV	V									I	II		III		IV								



#	Knoten	Teilformel
1	$\land$	$\neg\Big(\forall x\big(p(x,y)\longrightarrow q(f(x))\big)\Big)\land\forall x\forall y p(x,f(y))$
2	$\neg$	$\neg\Big(\forall x\big(p(x,y)\longrightarrow q(f(x))\big)\Big)$
3	$\forall x$	$\forall x\forall y p(x,f(y))$
4	$\forall x$	$\forall x\big(p(x,y)\longrightarrow q(f(x))\big)$
5	$\forall y$	$\forall y p(x,f(y))$
6	$\longrightarrow$	$p(x,y)\longrightarrow q(f(x))$
7	$p(x,f(y))$	$p(x,f(y))$
8	$p(x,y)$	$p(x,y)$
9	$q(f(x))$	$q(f(x))$

Offtop: solche Syntaxbäume sind ein wichtiger Bestandteil jedes Kompilierers

Aufgabe T3.4

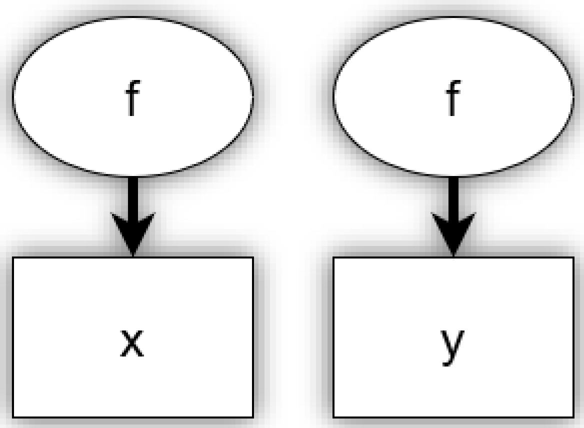
Lösung

Schritt #2: Bäume auch beim Auffinden/Auflisten von Termen nicht verkehrt ☺

$$\neg\left(\forall x\left(p(x,y) \longrightarrow q(f(x))\right)\right) \land \forall x \forall y p(x, f(y))$$

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18
$\neg$	(	$\forall$	x	(	p	(	x	,	y	)	$\rightarrow$	q	(	f	(	x	)
			1				1		2					3		1	
0	1	1	1	2	2	3	3	3	3	-3	2	2	3	3	4	4	-4
			I				I		I					II		I	

19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
)	)	)	$\land$	$\forall$	x	$\forall$	y	p	(	x	,	f	(	y	)	)
					1		2			1		4		2		
-3	-2	-1	0	0	0	0	0	0	1	1	1	1	2	2	-2	-1
					I		I			I		II		I		



#	Knoten	Term
1	x	x
2	y	y
3	f	$f(x)$
4	f	$f(y)$

Schritt #3: freie/gebundene Variablenvorkommen

Analog zur Aufgabe #3 (b:3) – gilt als **Hausaufgabe**

Aufgabe T3.5: *allgemeinster Unifikator #1*

Geben Sie zu folgenden Formeln F und G einen allgemeinsten Unifikator μ sowie das Ergebnis μ(F) = μ(G) der Unifikation an, sofern das möglich ist:

a) F = q(f(f(x,y),x)) und G = q(f(f(g(c),z),g(z)))

b) F = p(x,y) und G = p(f(y),f(x))

Schreibweise:	u, v, w, x, y, z	Variablen
	a, b, c	Konstanten
	f, g, h	Funktionssymbole
	p, q, r	Prädikatsymbole

Aufgabe T3.5

> **Anmerkung**: der **Schritt #1** ist nicht unbedingt nötig – man könnte direkt mit der Unifikation der Literalmenge $\{F, G\}$ anfangen (siehe das linke Einleitungsbild)

Lösung: (a)

Schritt #1: Umwandlung der Aufgabe in eine Standardunifikation der Termmenge

Da die Unifikation auf die Substitutionen $\boldsymbol{\sigma}: \mathbb{V} \subset \mathcal{T} \longmapsto \mathcal{T}$ mit $|\mathbb{V}| < \infty$ basiert, die aber auf (verneinte) Prädikatensymbole aka Relationssymbole keinerlei Auswirkung aufweisen:

$$\sigma\big(p(t_1, t_2, \dots, t_n)\big) \stackrel{\text{def}}{=} p\big(\sigma(t_1), \sigma(t_2), \dots, \sigma(t_n)\big) \text{ bei } s(p) = n \text{ und } t_{1 \le i \le n} \in \mathcal{T}$$

$$\sigma\big(\neg p(t_1, t_2, \dots, t_n)\big) \stackrel{\text{def}}{=} \neg p\big(\sigma(t_1), \sigma(t_2), \dots, \sigma(t_n)\big) \text{ bei } s(p) = n \text{ und } t_{1 \le i \le n} \in \mathcal{T}$$

Und da die beiden Formeln $\boldsymbol{F}$ und $\boldsymbol{G}$ identische Prädikatensymbole (hier: gleichstellige $\boldsymbol{q}$) haben, wird die ursprüngliche Aufgabe auf die Unifikation von Termen reduziert (und zwar direkt in der **Robinsonschen** Fassung: siehe das rechte Einleitungsbild):

$$UNIFY\Big(f\big(f(x,y),x\big), f\big(f(g(c),z),g(z)\big)\Big)$$

> **QuickFact**: der Ausdruck $p(t_1, t_2, \dots, t_n)$ $(p \in \mathbb{P})$ sowie dessen Verneinung sind prädikatenlogische Literale

Schritt #2: Unifikation der Terme

#	Differenz aka *Abweichung*	Substitution	Termmenge
	Aktualisierung der Mengen		
0	$D_0 = \emptyset$	$\sigma_0 = \{x \mapsto x, y \mapsto y, z \mapsto z\}$ alternativ $\sigma_0 = \emptyset$	$TM_0 = \Big\{f\big(f(x,y),x\big), f\big(f(g(c),z),g(z)\big)\Big\}$
1	$D_1 = \big\{\big(x, g(c)\big)\big\}$	$\sigma_1 = [x \mapsto g(c)] \circ \sigma_0$	$TM_1 = \Big\{f\big(f(g(c),y),g(c)\big), f\big(f(g(c),z),g(z)\big)\Big\}$
2	$D_2 = \{(y,z)\}$	$\sigma_2 = [y \mapsto z] \circ \sigma_1$	$TM_2 = \Big\{f\big(f(g(c),z),g(c)\big), f\big(f(g(c),z),g(z)\big)\Big\}$
3	$D_3 = \{(c,z)\}$	$\sigma_3 = [z \mapsto c] \circ \sigma_2$	$TM_3 = \Big\{f\big(f(g(c),c),g(c)\big), f\big(f(g(c),c),g(c)\big)\Big\}$
4	$D_4 = \emptyset$	$\sigma = [z \mapsto c][y \mapsto z][x \mapsto g(c)] = \{x \mapsto g(c), y \mapsto c, z \mapsto c\}$	$TM = \Big\{f\big(f(g(c),c),g(c)\big)\Big\}$

Hausaufgabe: warum nicht $\{x \mapsto g(c), y \mapsto z, z \mapsto c\}$?

Hausaufgabe: in welchen Fällen gilt

$$[v_n \mapsto t_n] \dots [v_2 \mapsto t_2][v_1 \mapsto t_1] = \{v_1 \mapsto t_1, v_2 \mapsto t_2, \dots, v_n \mapsto t_n\} ?$$

Aufgabe T3.5

Lösung: (b)

Analog zur Aufgabe #5 (a) – gilt als **Hausaufgabe**

Hier ist eine Skizze für die Lösung:

	Aktualisierung der Mengen		
#	**Differenz** *aka* ***Abweichung***	**Substitution**	**Literalmenge**
0	$D_0 = \emptyset$	$\sigma_0 = \{x \mapsto x, y \mapsto y\}$	$LM_0 = \{p(x, y), p(f(y), f(x))\}$
1	$D_1 = \{(x, f(y))\}$	$\sigma_1 = [x \mapsto f(y)] \circ \sigma_0$	$LM_1 = \{p(f(y), y), p(f(y), f(f(y)))\}$
2	$D_2 = \{(y, f(f(y)))\}$	$\sigma_2 = [y \mapsto f(f(y))] \circ \sigma_1$	$LM_2 = \{p(f(f(f(y))), f(f(y))), p(f(f(f(y))), f(f(f(f(y)))))\}$
3	$D_3 = D_2$	$\sigma_3 = \sigma_2$	Verschachtelungen werden immer länger (endlose Schleife)
		Keine Unifikation (bei endlichen Termen aka Bäumen) möglich	

Anmerkung: alternativ könnte man den Unifikationsvorgang bereits bei $D_2 = \{(y, f(f(y)))\}$ mit der Ausgabe „**nicht unifizierbar** („Occurs check“ ist fehlgeschlagen)“ terminieren.

Aufgabe T3.6: *allgemeinster Unifikator #2*

Wenden Sie den Unifikationsalgorithmus auf die Literalmengen

$K = \{p(f(v, g(v, z)), z), p(f(h(u), v), h(a))\}$

$L = \{p(x, y), p(f(a), g(x)), p(f(z), g(f(z)))\}$

an.

Schreibweise:	u, v, w, x, y, z	Variablen
	a, b, c	Konstanten
	f, g, h	Funktionssymbole
	p, q, r	Prädikatsymbole

Aufgabe T3.6

Lösung

Analog zur Aufgabe #5 (a) – gilt als **Hausaufgabe**

Hier ist eine Skizze für die Lösung (Literalmenge L):

	Aktualisierung der Mengen		
#	**Differenz** *aka* ***Abweichung***	**Substitution**	**Literalmenge**
0	$D_0 = \emptyset$	$\sigma_0 = \emptyset$	$LM_0 = \{p(x,y), p(f(a), g(x)), p(f(z), g(f(z)))\}$
1	$D_1 = \{(x, f(a), f(z))\}$	$\sigma_1 = [z \mapsto a][x \mapsto f(a)]$	$LM_1 = \{p(f(a), y), p(f(a), g(f(a))), p(f(a), g(f(a)))\}$
2	$D_2 = \{(y, g(f(a)))\}$	$\sigma_2 = [y \mapsto g(f(a))] \circ \sigma_1$	$LM_2 = \{p(f(a), g(f(a))), p(f(a), g(f(a))), p(f(a), g(f(a)))\}$
		$\sigma = [y \mapsto g(f(a))][z \mapsto a][x \mapsto f(a)] = \{x \mapsto f(a), y \mapsto g(f(a)), z \mapsto a\}$ $LM = \{p(f(a), g(f(a)))\}$	

Aufgabe T3.7: *Pränex- & Skolem-NF*

Berechnen Sie eine Pränex-Normalform und eine Skolem-Normalform für die prädikatenlogische Formel:

$F = \forall z \exists y (p(x, g(y), z) \vee \neg \forall x\, q(x)) \wedge \neg \forall z \exists x\, \neg r(f(x, z), z)$

Schreibweise:	u, v, w, x, y, z	Variablen
	a, b, c	Konstanten
	f, g, h	Funktionssymbole
	p, q, r	Prädikatsymbole

Aufgabe T3.7

Lösung

Schritt #0:

Vorsichtshinweis: unterschiedliche Quantoren kommutieren nicht

$$F = \forall z \exists y\big(p(x, g(y), z) \ \lor \ \neg\forall x q(x)\big) \quad \land \quad \neg\forall z \exists x \neg r(f(x,z),z)$$

$$F = \forall z \exists y\big(p(x, g(y), z) \ \lor \quad \boldsymbol{G} \quad \big) \quad \land \quad \boldsymbol{H}$$

Vorsichtshinweis: ist ein Vorkommen einer Variable (in Bezug auf einen Quantor) frei, dann muss es auch nach der Umbenennung/Quantorverschiebung frei bleiben (in Bezug auf denselben Quantor)

Schritt #1: Umformung von F in eine äquivalente Formel in Pränex-Normalform (PNF)

$$
\begin{aligned}
F &\equiv \forall z \exists y\big(p(x,g(y),z) \lor \exists x \neg q(x)\big) \land \exists z \neg \exists x \neg r(f(x,z),z) && \text{Negation nach innen schieben}\\
&\equiv \forall z \exists y\big(p(x,g(y),z) \lor \exists x \neg q(x)\big) \land \exists z \forall x \neg\neg r(f(x,z),z) && \text{Negation nach innen schieben}\\
&\equiv \forall z \exists y\big(p(x,g(y),z) \lor \exists x \neg q(x)\big) \land \exists z \forall x r(f(x,z),z) && \text{Doppelnegation}\\
&\equiv \forall z \exists y\big(p(x,g(y),z) \lor \exists t \neg q(t)\big) \land \exists u \forall v r(f(v,u),u) && \text{Umbennenung gebundener Variablen}
\end{aligned}
$$

(d.h. **Ersetzung** durch neue Variablen):

in G: x/t, **in H**: z/u und x/v

$$
\begin{aligned}
&\equiv \forall z \exists y \exists t\big(p(x,g(y),z) \lor \neg q(t)\big) \land \exists u \forall v r(f(v,u),u) && \text{Quantoren nach außen ziehen}\\
&\equiv \exists u(\forall z \exists y \exists t\big(p(x,g(y),z) \lor \neg q(t)\big) \land \forall v r(f(v,u),u)) && \text{Quantoren nach außen ziehen}\\
&\equiv \exists u \forall v(\forall z \exists y \exists t\big(p(x,g(y),z) \lor \neg q(t)\big) \land r(f(v,u),u)) && \text{Quantoren nach außen ziehen}\\
&\equiv \exists u \forall v \forall z(\exists y \exists t\big(p(x,g(y),z) \lor \neg q(t)\big) \land r(f(v,u),u)) && \text{Quantoren nach außen ziehen}\\
&\equiv \exists u \forall v \forall z \exists y(\exists t\big(p(x,g(y),z) \lor \neg q(t)\big) \land r(f(v,u),u)) && \text{Quantoren nach außen ziehen}\\
&\equiv \exists u \forall v \forall z \exists y \exists t(\big(p(x,g(y),z) \lor \neg q(t)\big) \land r(f(v,u),u)) && \text{Quantoren nach außen ziehen}
\end{aligned}
$$

Pränex-Normalform von F:

$$\exists u \forall v \forall z \exists y \exists t\big((p(x,g(y),z) \lor \neg q(t)) \land r(f(v,u),u)\big)$$

Aufgabe T3.7

Lösung

Schritt #2: weitere Umformung in eine erfüllbarkeitsäquivalente Formel in Skolem-Normalform (SNF)

$$\equiv \forall x \exists u \forall v \forall z \exists y \exists t\big((p(x, g(y), z) \vee \neg q(t)) \wedge r(f(v, u), u)\big)$$

$$\equiv^* \forall x \forall v \forall z\left((p(x, g(h(x, v, z)), z) \vee \neg q(i(x, v, z))) \wedge r\left(f(v, j(x)), j(x)\right)\right)$$

Allabschluß (betrifft jeweils **alle freien Variablen** – hier ist solche nur x)

Skolemisierung (erfüllbarkeitsäquivalent)

Skolem-Normalform von F:

$$\boxed{\forall x \forall v \forall z\left((p(x, g(h(x, v, z)), z) \vee \neg q(i(x, v, z))) \wedge r\left(f(v, j(x)), j(x)\right)\right)}$$

Skolemisierungsweg:

Aus $\forall x_1 \dots \forall x_n \exists y\, Q_1 z_1 \dots Q_m z_m\, M$	Q_i **ist** $\exists$ **oder** $\forall$
mache $\forall x_1 \dots \forall x_n\, Q_1 z_1 \dots Q_m z_m\, M[y/f(x_1 \dots x_n)]$	$M[y/f(x_1 \dots x_n)]$ bedeutet das Ergebnis der in M durchgeführten Ersetzung von y durch **eine neue Funktion** f. Die Argumenten dieser Funktion sind genau all die allquantifizierten Variablen (hier - $x_1 \dots x_n$), in denen Bindungsbereich die Variable y lag. Wie man sieht, die Stelligkeit solcher Funktion ist gleich der **Anzahl** von allen allquantifizierten Variablen, die vor der zu ersetzenden existenzquantifizierten Variablen stehen: sollte keine allquantifizierten Variablen vor so einer zu ersetzenden Variablen stehen ***(die oben erwähnte Anzahl ist dann einfach* 0*)***, wird diese zu ersetzende Variable durch eine neue 0-stellige Funktion (anders gesagt – **durch eine neue Konstante**) ersetzt.
wiederhole, bis alle $\exists$ eliminiert sind	