# Universität Karlsruhe

## Informatik 1 WS 00/01

Institut für Telematik, Forschungsgruppe C&M
Prof. Dr. S. Abeck, R. Scholderer


Abgabe bis: 09.01.2001 14:00 Uhr

# Übungsblatt 9

# Aufgabe

## 1. Auswertungsstrategien für Termersetzungssysteme (2 Punkte)

Im folgenden betrachten wir zwei Auswertungsstrategien für Termersetzungssysteme:

E („eager"): Es wird in Termen immer möglichst weit innen reduziert. D.h.: Auf eine Funktionsapplikation wird erst dann eine Regel angewandt, wenn sich kein Argument mehr reduzieren läßt.
L („lazy"): Es wird in Termen immer möglichst weit außen reduziert. D.h.: Auf Funktionsargumente werden nur dann Regeln angewandt, wenn weiter außen keine Regel anwendbar ist.

Sind zwei Auswertungen auf gleicher Ebene möglich, so wird von links nach rechts vorgegangen.

NAT=$(\mathcal{S}_{\mathfrak{N}},\mathcal{F}_{\mathfrak{N}})$ definiere nun die Rechenstruktur NAT aus der Vorlesung.
Wir erweitern nun NAT zur Rechenstruktur NATSUM==$(\mathcal{S}_{\mathfrak{N}}, \mathcal{F} \cup \{$if . then . else . fi$^{BOOLIF}$, db$^{NATSUM}$, sum$^{NATUSM}\})$.

Die Funktionalität von if.then.else.fi ist wie in der Aufgabe *Rechenstruktur BOOLIF* , die Funktionalitäten von db und sum sind wie folgt gegeben:

fct db = (nat) nat und fct sum = (nat) nat

Für die Funktionen pred, add, $\overset{?}{=}$, db, sum und if.then.else.fi ist folgendes Termersetzungssystem über NATSUM gegeben:

(P) pred(succ(x)) $\rightarrow$ x
(G0) zero $\overset{?}{=}$ zero $\rightarrow$ true
(Gl) zero $\overset{?}{=}$ succ(y) $\rightarrow$ false
(G2) succ(x) $\overset{?}{=}$ zero $\rightarrow$ false
(G3) succ(x) $\overset{?}{=}$ succ(y) $\rightarrow$ x $\overset{?}{=}$ y
(A0) add(zero,y) $\rightarrow$ y
(Al) add(succ(x),y) $\rightarrow$ succ(add(x,y))
(D) db(x) $\rightarrow$ add(x,x)

(S) sum(x) $\rightarrow$ if x $\stackrel{?}{=}$ zero then zero else add(x, sum(pred(x))) fi
(I0) if true then x else y $\rightarrow$ x
(I1) if false then x else y $\rightarrow$ y

Werten Sie folgende Terme jeweils nach den Strategien E und L aus:
**Teilaufgabe 1:** add(succ(succ(zero)), succ(zero)) $\stackrel{?}{=}$ zero
**Teilaufgabe 2:** db(add(succ(zero), zero))

---

# Aufgabe

## 2. Termersetzungssysteme (2 Punkte)

Gegeben ist folgendes Termersetzungssystem über der Rechenstruktur NATSUM:

| | |
|---|---|
| (P)pred(succ(x))$\rightarrow$x | (G0) zero $\stackrel{?}{=}$ zero $\rightarrow$ true |
| | (Gl) zero $\stackrel{?}{=}$ succ(y) $\rightarrow$ false |
| (A0) add(zero,y)$\rightarrow$y | (G2) succ(x) $\stackrel{?}{=}$ zero $\rightarrow$ false |
| (Al) add(succ(x),y)$\rightarrow$succ(add(x,y)) | (G3) succ(x) $\stackrel{?}{=}$ succ(y)$\rightarrow$ x $\stackrel{?}{=}$ y |

a) Werten Sie den Ausdruck **add(pred(succ(zero)), add(zero, succ(zero))** $\stackrel{?}{=}$ **zero** nach der Eager -Strategie aus. Eager: Es wird in Termen immer möglichst weit innen reduziert. D.h.: Auf eine Funktionsapplikation wird erst dann eine Regel angewandt, wenn sich kein Argument mehr reduzieren läßt.

b) Werten Sie den Ausdruck **pred(succ(succ(add(succ(zero), zero))))** $\stackrel{?}{=}$ **succ(zero)** nach der Lazy-Strategie aus. Lazy: Es wird in Termen immer möglichst weit außen reduziert. D.h.: Auf Funktionsargumente werden nur dann Regeln angewandt, wenn weiter außen keine Regel anwendbar ist.

[Hinweis: Sind zwei Auswertungen auf gleicher Ebene möglich, so wird von links nach rechts vorgegangen.]

---

# Aufgabe

## 3. Theoreme der Aussagenlogik (2 Punkte)

In der folgenden Aufgabe sollen die Ableitungsregeln und Lemmata der Aussagenlogik für die in der Vorlesung behandelten Theoreme angewendet werden.

Um mit der Aussagenlogik komplexere Fragestellungen zu behandeln, sind Kalküle erforderlich, die innerhalb der Vorlesung Informatik 1 nicht eingeführt werden.

a) Gegeben sei das Theorem zur Inkonsistenz einer Theorie aus der Vorlesung [Kap.3.4, Folie5]:

Ist in einer Theorie der Aussagenlogik H $\vdash$ false ableitbar, so ist jede Aussage t ableitbar also H $\vdash$ t (*).

- Zeigen Sie zunächst (Vervollständigung des Beweises der Vorlesung), dass {false} $\vdash$ t (**) gilt.
- Zeigen Sie nun, dass mit (**) und der Voraussetzung des Theorems H $\vdash$ false die Aussage (*) gilt.

b) Sei nun das Theorem zur Korrektheit von Ableitungen aus der Vorlesung [Kap 3.4, Folie 6] gegeben:

Jeder aus einer konsistenten Axiomenmenge ableitbaren Aussage t mit freien Identifikatoren $x_1, \dots, x_n$ der Sorte bool wird für beliebige Belegungen von $x_1, \dots, x_n$ durch L oder 0 der Wert L („wahr“) zugeordnet.

Beweisen Sie dieses Theorem unter zur Hilfenahme der Beweisskizze aus der Vorlesung.

*Hinweis:* Unterscheiden Sie beim Beweis die Fälle, dass die Axiomenmenge leer ist oder nicht. Beweisen Sie die Korrektheit von Theorien mit konsistenter Axiomenmenge (vgl. Vorlesung).

---

# Aufgabe

## 4. Prädikatenlogische Formel (2 Punkte)

Gegeben sei die prädikatenlogische Formel

$$F = Q(x) \vee \exists x\, \forall y\, (P(\, f(z), z\,) \wedge Q(a)\,) \vee \forall\, y\, R(x, z, g(y))$$

Es sei a eine Konstante, x, y, z seien Variablen.

Geben Sie für jedes Vorkommen einer jeden Variablen in den Teiltermen der Formel an, ob die Variable frei oder gebunden ist.

(*Hinweis:* Quantoren mit einer Variablen (z.B. $\exists x$) gehören zu keinem Teilterm!)

---

# Aufgabe

## 5. Modelle (2 Punkte)

**Teilaufgabe 1:**

Gegeben ist die Formel $F = \exists x\, \exists y\, \exists z\, (P(x,y,z) \wedge P(x,z,y) \wedge P(z,y,x))$ und die Struktur A mit dem Paar $(I_A, U_A)$, wobei

$I_A(P) = \{(r,s,t), r,s,t \in \mathbf{N}, r < s \leq t\}$

$U_A = \mathbf{N}$ ($\mathbf{N}$ = Natürliche Zahlen ohne Null)

Untersuchen Sie, ob die Struktur A ein Modell für die Formel F ist.

**Teilaufgabe 2:**

Gegeben ist die Formel $F = \exists x \exists y \exists z\, (P(x, y) \wedge P(z, y) \wedge P(x, z) \wedge \neg P(z,x))$ und die Struktur A mit dem Paar $(I_A, U_A)$, wobei

$I_A(P) = \{(A, B) \mid A, B \subseteq \mathbf{N}, A \subseteq B\}$

$U_A = \boldsymbol{P}(\mathbf{N})$ ($\boldsymbol{P}$ ($\mathbf{N}$) ist die Potenzmenge der natürlichen Zahlen)

Untersuchen Sie, ob die Struktur A ein Modell für die Formel F ist.