1 Einleitung

1.1 Ziel: Motivation und Beispiele

Motivation

Ein Reduktionssystem ist

$(\mathcal{E}, \longrightarrow)$,

wobei $\mathcal{E}$=Menge, $\longrightarrow$=2-stellige Relation.

Dies dient zum

a) Rechnen,
nicht-deterministisch, eindeutige Ergebnisse.

b) Beweisen.

Ein **Wortproblem** ist gegeben durch

$(\mathcal{E}, \sim)$,

wobei $\mathcal{E}$=Menge, $\sim$=Äquivalenz-Relation.

Eingabe: $u, v \in \mathcal{E}$
Frage: Gilt $u \sim v$?

Beispiel: Wortersetzungssystem

$$\mathcal{E} = \Sigma^* \qquad \Sigma = \{a, b, c\}$$

$$\sim\; = \;=_E \qquad E\colon \begin{array}{l} ab^2 = ba \\ ca = a \\ cb = bc^3 \end{array}$$

$$\longrightarrow\; = \;\longrightarrow_R \qquad R\colon \begin{array}{l} ba \to ab^2 \\ ca \to a \\ cb \to bc^3 \end{array}$$

Rechnen

$$\underline{cb}a \longrightarrow bcc\underline{ca} \longrightarrow bc\underline{ca} \longrightarrow b\underline{ca} \longrightarrow \underline{ba} \longrightarrow ab^2$$

$$c\underline{ba} \longrightarrow \underline{ca}b^2 \longrightarrow ab^2$$

Beweisen: Es gilt

1. Jedes $u \in \mathcal{E}$ hat eindeutige Normalform $\hat{u}$ bzgl. $\longrightarrow$.
2. $u =_E v$ gdw $\hat{u} \equiv \hat{v}$.

Also: Das Wortproblem zu $(\mathcal{E}, \sim)$ ist effektiv lösbar.

Einsatz von Reduktionssystemen (1)

Beispiel

$$\begin{array}{ll} \mathcal{E} = \Sigma^* & \Sigma = \{a, b\} \\ E : ba = ab^2 & a^3 = \lambda \end{array}$$

Sei $\sim \; = \; =_E$, $[w] = \{v \mid v =_E w\}$.

Fragen:

1. Test, ob $u =_E v$ gilt.
2. Wieviele $[w]$ gibt es?
3. Ist $\mathcal{E}/\sim$ eine Gruppe, d. h. gilt

$$\bigwedge_u \bigvee_v u \cdot v =_E \lambda?$$

Setze $\longrightarrow \; = \; \longrightarrow_R$

$$R : ba \to ab^2 \qquad a^3 \to \lambda$$

Fragen:

a) Terminiert jede R-Ableitung? JA!

b) Hat jedes u eine eindeutige Normalform $\hat{u}$, also $u \xrightarrow{*} \hat{u}$?
NEIN:

$$\begin{tikzcd} & ab^2a^2 \arrow[r, "*"] & b^8 \\ ba^3 \arrow[ur] \arrow[dr] & & \\ & b & \end{tikzcd}$$

Einsatz von Reduktionssystemen (2)

Preprocessing (KB-Vervollständigung)

liefert $R' : ba \to ab^2, \qquad a^3 \to \lambda, \qquad b^8 \to b$

mit

a) $\longrightarrow_{R'}$ terminiert.

b) $\longrightarrow_{R'}$ ist konfluent.

c) Zu jedem u gibt es genau ein $\hat{u}$ mit $u \xrightarrow{*}_{R'} \hat{u}$ und $\hat{u}$ ist irreduzibel.

d) Es gilt $u =_E v \text{ gdw } \hat{u} \equiv \hat{v}$.

Dies liefert

- Das „Wortproblem" zu (Σ, E) ist entscheidbar.
- (Σ, E) ist eine Halbgruppe H, keine Gruppe.
- Die R'-Normalformen sind
  $\lambda, b, \ldots, b^7, a, ab, \ldots, ab^7, a^2, a^2b, \ldots, a^2b^7$
  $\leadsto H$ hat $3 \cdot 8 = 24$ Elemente.
- In H kann man mit R' effektiv rechnen: $[\hat{u}] \cdot [\hat{v}] = [\widehat{uv}]$.

Beispiel: Gruppen

Sei $\mathcal{E}$ = Terme über $F = \{1, \cdot, {}^{-1}\}$
$\sim \; = \; =_E$
$\longrightarrow \; = \; \longrightarrow_R$

$$
\begin{aligned}
E: \quad & x \cdot 1 = x \\
& x \cdot x^{-1} = 1 \\
& x \cdot (y \cdot z) = (x \cdot y) \cdot z
\end{aligned}
$$

Die KB-Vervollständigung liefert

$$
\begin{array}{lll}
R: & x \cdot 1 \to x & 1 \cdot x \to x \\
& x \cdot x^{-1} \to 1 & x^{-1} \cdot x \to 1 \\
& 1^{-1} \to 1 & (x^{-1})^{-1} \to x \\
& (x \cdot y)^{-1} \to y^{-1} \cdot x^{-1} & (x \cdot y) \cdot z \to x \cdot (y \cdot z) \\
& x \cdot (x^{-1} \cdot y) \to y & x^{-1} \cdot (x \cdot y) \to y
\end{array}
$$

Frage: Gilt $x \cdot (y \cdot x)^{-1} =_E (x \cdot y)^{-1} \cdot x$?

Beispiel: Gruppen (Forts.)

Antwort: JA, denn

$$x \cdot \underbrace{(y \cdot x)^{-1}} \longrightarrow \underbrace{x \cdot (x^{-1} \cdot y^{-1})} \longrightarrow y^{-1}$$

$$\underbrace{(x \cdot y)^{-1}} \cdot x \longrightarrow \underbrace{(y^{-1} \cdot x^{-1}) \cdot x} \longrightarrow y^{-1} \cdot \underbrace{(x^{-1} \cdot x)}$$

$$\longrightarrow y^{-1} \cdot 1 \longrightarrow y^{-1}$$

Folgerung:

Das Wortproblem für freie Gruppen ist entscheidbar!

Spezifikationen

Eine **Spezifikation** spec besteht aus

- einer Signatur,
- definierenden Axiomen.

Hier

- spec = (sig, E).

- sig = (S, F, α) Signatur,
  wobei S=Sorten, F=Operatoren, α=Stelligkeit der Operatoren.
  Eine Signatur sig definiert die gültigen **Terme**.

- $E = \{u_i = v_i \mid i = 1, \ldots, n\}$
  definierendes Gleichungssystem.
  Alle Gleichungen implizit all-quantifiziert.

- Einige **Grundterme** (variablenfreie Terme) bilden die **Daten** der Spezifikation.

- Spezifikationen sind häufig hierarchisch.

Beispiel einer Spezifikation (1)

Boole'sche Werte

Sorte: BOOL $\}\ S$

Operatoren :
$$\left.\begin{array}{l}\textbf{true}, \textbf{false} :\to \text{BOOL}\\ \textbf{not} : \text{BOOL} \to \text{BOOL}\\ \textbf{and} : \text{BOOL}, \text{BOOL} \to \text{BOOL}\end{array}\right\} F, \alpha$$

Variable: $b \in V_{\text{BOOL}}$

E_1 :
$$\left.\begin{array}{l}\textbf{not}(\textbf{true}) = \textbf{false}\\ \textbf{not}(\textbf{false}) = \textbf{true}\\ \textbf{and}(b, \textbf{true}) = b\\ \textbf{and}(\textbf{true}, b) = b\\ \textbf{and}(b, \textbf{false}) = \textbf{false}\\ \textbf{and}(\textbf{false}, b) = \textbf{false}\end{array}\right\} E$$

Daten: **true**, **false**

Jeder Grundterm ist E_1-gleich zu einem Datum.

Beispiel einer Spezifikation (2)

Natürliche Zahlen

Sorte: NAT

Operatoren: $0 :\to$ NAT $\quad + :$ NAT, NAT $\to$ NAT
$s :$ NAT $\to$ NAT $\quad \leq :$ NAT, NAT $\to$ BOOL

Variable: $x, y \in V_{\mathsf{NAT}}$

E_2:

$$x + 0 = x \qquad 0 \leq x = \mathbf{true}$$
$$x + s(y) = s(x + y) \qquad s(x) \leq 0 = \mathbf{false}$$
$$s(x) \leq s(y) = x \leq y$$

Daten: $s^i(0)$ „Zahlen“

Jeder Grundterm der Sorte NAT ist $(E_1 \cup E_2)$-gleich zu einem Datum der Sorte NAT.

Beispiel einer Spezifikation (3)

Listen über NAT

Sorte: LIST

Operatoren:

nil $:\to$ LIST $\quad$ app : LIST, LIST $\to$ LIST
. : NAT, LIST $\to$ LIST $\quad$ if : BOOL, LIST, LIST $\to$ LIST

Variable: $q, q_1, q_2 \in V_{\mathsf{LIST}}$

E_3:

$$\mathsf{app}(\mathsf{nil}, q_2) = q_2$$
$$\mathsf{app}(x.q_1, q_2) = x.\mathsf{app}(q_1, q_2)$$
$$\mathsf{if}(\mathbf{true}, q_1, q_2) = q_1$$
$$\mathsf{if}(\mathbf{false}, q_1, q_2) = q_2$$

Daten: $a_1.a_2.\cdots a_n.$nil „Listen über $\mathbb{N}$“
a_i Daten der Sorte NAT

Jeder Grundterm der Sorte LIST ist $(E_1 \cup E_2 \cup E_3)$-gleich zu einem Datum der Sorte LIST.

Beispiel einer Spezifikation (4)

Programmieren: Quicksort

$$
\begin{array}{lll}
E_S\colon & \mathsf{sort}(\mathsf{nil}) & = \mathsf{nil} \\
& \mathsf{sort}(x.q) & = \mathsf{app}(\mathsf{sort}(\mathsf{low}(x,q)), x.\mathsf{sort}(\mathsf{up}(x,q))) \\
& \mathsf{low}(x,\mathsf{nil}) & = \mathsf{nil} \\
& \mathsf{low}(x,y.q) & = \mathsf{if}(\mathbf{not}(x \le y), y.\mathsf{low}(x,q), \mathsf{low}(x,q)) \\
& \mathsf{up}(x,\mathsf{nil}) & = \mathsf{nil} \\
& \mathsf{up}(x,y.q) & = \mathsf{if}(x \le y, y.\mathsf{up}(x,q), \mathsf{up}(x,q))
\end{array}
$$

Beispiel einer Spezifikation (5)

Rechnen in der Spezifikation

Anwendung der Gleichungen von links nach rechts.

a) $\mathbf{and}(\mathbf{not}(\mathbf{true}), \mathbf{and}(\mathbf{true}, \mathbf{true})) \longrightarrow \mathbf{and}(\mathbf{not}(\mathbf{true}), \mathbf{true})$
$\longrightarrow \mathbf{and}(\mathbf{false}, \mathbf{true}) \longrightarrow \mathbf{false}$ in E_1

b) $(0 + s(0)) \le s^2(0) \longrightarrow s(0+0) \le s^2(0)$
$\longrightarrow s(0) \le s^2(0) \longrightarrow 0 \le s(0) \longrightarrow \mathbf{true}$ in E_2

c) $\mathsf{app}(s(0).s^2(0).\mathsf{nil}, s^2(0).\mathsf{nil})$
$\longrightarrow s(0).\mathsf{app}(s^2(0).\mathsf{nil}, s^2(0).\mathsf{nil})$
$\longrightarrow s(0).s^2(0).\mathsf{app}(\mathsf{nil}, s^2(0).\mathsf{nil})$
$\longrightarrow s(0).s^2(0).s^2(0).\mathsf{nil}$ in E_3

d) Sei $\underline{i} \equiv s^i(0)$ für $i \in \mathbb{N}$, $E = E_1 \cup E_2 \cup E_3 \cup E_S$
$\mathsf{sort}(\underline{5}.\underline{3}.\underline{6}.\underline{7}.\mathsf{nil})$
$\longrightarrow \mathsf{app}(\mathsf{sort}(\mathsf{low}(\underline{5}, \underline{3}.\underline{6}.\underline{7}.\mathsf{nil})), \underline{5}.\mathsf{sort}(\mathsf{up}(\underline{5}, \underline{3}.\underline{6}.\underline{7}.\mathsf{nil})))$
$\xrightarrow{*} \mathsf{app}(\mathsf{sort}(\underline{3}.\mathsf{nil}), \underline{5}.\mathsf{sort}(\underline{6}.\underline{7}.\mathsf{nil}))$
$\xrightarrow{*} \mathsf{app}(\underline{3}.\mathsf{nil}, \underline{5}.\underline{6}.\underline{7}.\mathsf{nil})$
$\xrightarrow{*} \underline{3}.\underline{5}.\underline{6}.\underline{7}.\mathsf{nil}$ in E

Beispiel einer Spezifikation (6)

Bisher

Spezifikation eines abstrakten Datentyps, dient zum Rechnen.

Jetzt

Spezifikation einer Korrektheitsaussage, dient zum Beweisen.

Setze

$$
\begin{aligned}
E_K\colon\ & \mathsf{sorted}(\mathsf{nil}) &&= \mathbf{true}\\
& \mathsf{sorted}(x.\mathsf{nil}) &&= \mathbf{true}\\
& \mathsf{sorted}(x.y.q) &&= \mathbf{and}(x \le y, \mathsf{sorted}(y.q))
\end{aligned}
$$

Zeige

$\mathsf{sorted}(\mathsf{sort}(q)) = \mathbf{true}$ in $E \cup E_K$

für alle

$$
\begin{aligned}
q &\equiv a_1.a_2.\cdots.a_n.\mathsf{nil}\\
a_i &\equiv s^{j_i}(0)
\end{aligned}
$$

Problem „Gültigkeit“ (1)

Gegeben: $\mathsf{spec} = (\mathsf{sig}, E)$
$\mathsf{sig} = (S, F, \alpha)$
s, t Terme

Was heißt: „$s = t$ ist gültig in spec“?

Beispiel: „Listen über NAT“

$F = \{0, s, \mathsf{nil}, ., \mathsf{app}\}$

$$
\begin{aligned}
E:\ & \mathsf{app}(\mathsf{nil}, l_2) = l_2\\
& \mathsf{app}(x.l_1, l_2) = x.\mathsf{app}(l_1, l_2)
\end{aligned}
$$

Gilt das Assoziativ-Gesetz?

$$\mathsf{app}(\mathsf{app}(l_1, l_2), l_3) = \mathsf{app}(l_1, \mathsf{app}(l_2, l_3))$$

Problem „Gültigkeit“ (2)

Mögliche Definitionen von „$s = t$ ist gültig in spec“

1) s lässt sich mit Gleichungen aus E in t überführen.
   Operationale E-Gleichheit.

2) $s = t$ gilt in allen Modellen von spec.
   Denotationale E-Gleichheit (Logik).

3) $s = t$ gilt im „Daten-Modell“ (im Bsp.: Listen über $\mathbb{N}$).
   Induktive E-Gleichheit.

Problem „Gültigkeit“ (3)

Es gilt

a) $s = t$ ist (1)-gültig
   gdw $s = t$ ist (2)-gültig. (Satz von Birkhoff)

b) $s = t$ ist (1)-gültig
   $\curvearrowright$ $s = t$ ist (3)-gültig
   $\not\curvearrowleft$ (i. Allg.)

$\Rightarrow$ saubere Definition der Semantik wichtig!