Automaten

mit

Termalphabeten

Jens-D. Doll

1.2 $\mathbb{F}$ = Vektoren über (Polynom-)Ringen

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ \dots \\ x_{n-1} \\ x_n \end{pmatrix} \qquad G_i(x) = \begin{pmatrix} g_0(x) \\ g_1(x) \\ g_2(x) \\ \dots \\ \dots \\ g_{n-1}(x) \\ g_n(x) \end{pmatrix}$$

$x_i \in R$ (Grundstruktur: Ring, Schiefkörper, Körper)

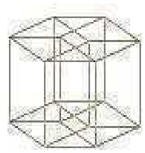
$g_i \in \mathbb{F}$: Modul über dem multivariaten Polynomring $R[x_, x^{-1}]$,
$g_i(x)$ univariat in x_i , weil x_j für $j \neq i$ als Parameter zu betrachten sind

1.3 $\mathbb{L}$ = Logik über den Vektoren

gegeben sei eine Ordnung auf $R[x, x^{-1}]$ und eine Logik $\mathbb{L}$ über $R[x, x^{-1}]$ mit Kombinatoren aus $\mathbb{L} \times \mathbb{L}$ um bedingte Funktionen zu definieren*:

$$F_i(x) \quad := \quad (c_i(x), G_i(x)) \quad := \quad \begin{pmatrix} c_i(x) \\ \\ g_0(x) \\ g_1(x) \\ g_2(x) \\ \ldots \\ \ldots \\ g_{n-1}(x) \\ g_n(x) \end{pmatrix}$$

$c_i(x)$ definieren Polytope im R^n



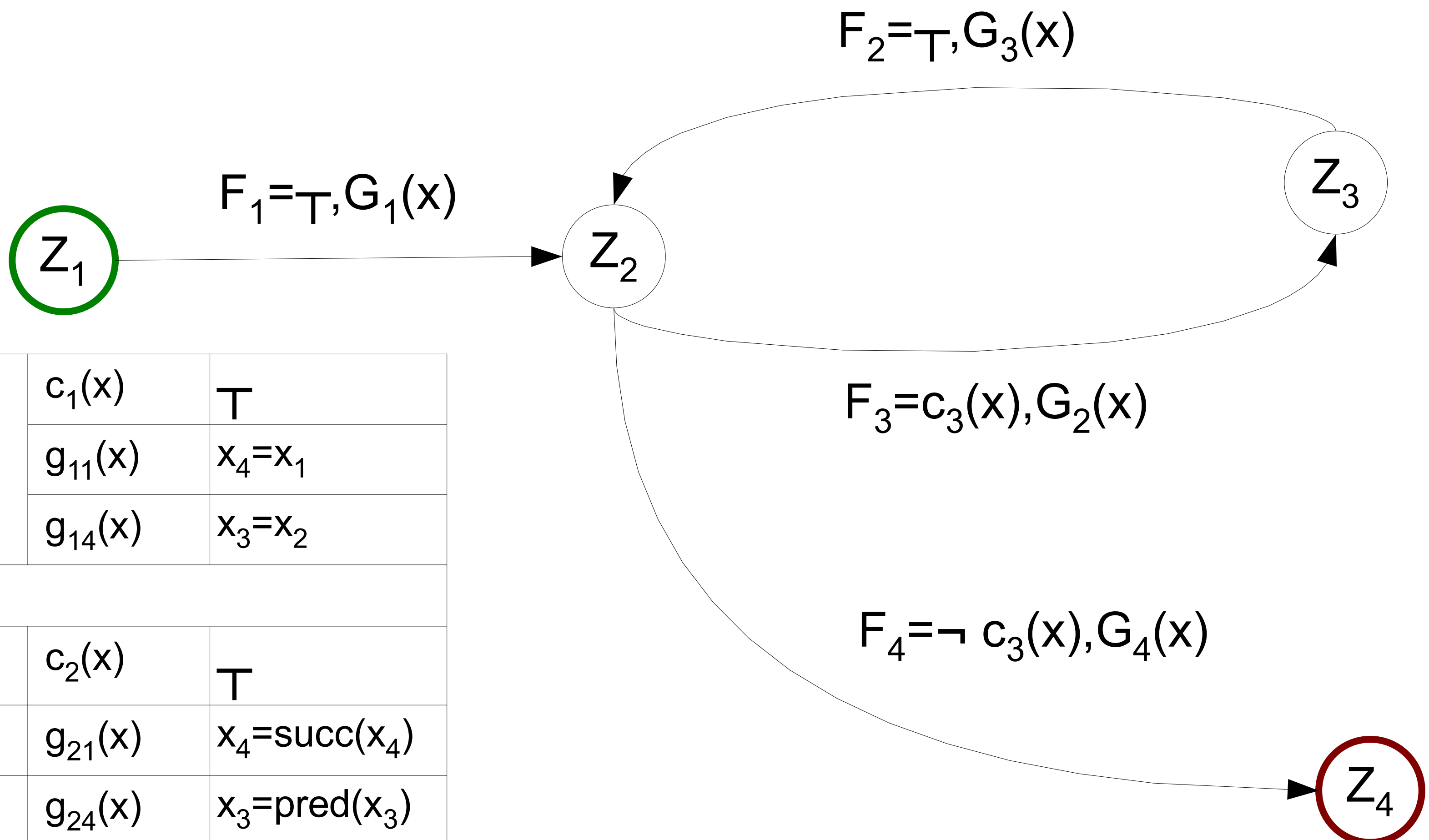
*(≈ guarded command)

1.4 Definition des Termautomaten

Ein Termautomat ist ein Tupel ($\mathbb{T}$, $\mathbb{E}$, Z, S, E, U)

$\mathbb{T}$	Trägermenge T(∑,x) für $\mathbb{F}$ und $\mathbb{L}$
$\mathbb{E}$	$\epsilon\ \mathcal{L}_0$, definiert Axiome für R, $\mathbb{F}$ und $\mathbb{L}$
Z	endlich viele Zustände
S, E	Startzustand, Endzustände
U	Zustandsübergänge ϵ Z x $\mathbb{L}$ x $\mathbb{F}$ x Z mit c(x) T(∑,x) $\longrightarrow$ $\mathbb{L}$ mit F(x) T(∑,x) $\longrightarrow$ T(∑,x) (und ggf. $\forall$ z ϵ Z, k ϵ U, b ϵ $\mathbb{L}$: |k(z,b)| <= 2)

2.1 Beispielautomat



F_1	$c_1(x)$	$\top$
	$g_{11}(x)$	$x_4 = x_1$
	$g_{14}(x)$	$x_3 = x_2$
F_2	$c_2(x)$	$\top$
	$g_{21}(x)$	$x_4 = succ(x_4)$
	$g_{24}(x)$	$x_3 = pred(x_3)$
F_3	$c_3(x)$	$x_3 > 0$
F_4	$\neg\, c_3(x)$	$x_3 \leq 0$

es seien x1, x2 $\epsilon\, \mathbb{N}^0$ Parameter

2.1.1 Komposition, Partitionierung und Iteration

Sei $\mathbb{K} := \mathbb{L} \times \mathbb{F}$

$P(\mathbb{K}) :=$ die Potenzmenge

Komposition	o	$P(\mathbb{K}) \times P(\mathbb{K})$	$\rightarrow$	$P(\mathbb{K})$
Partitionierung	I	$P(\mathbb{K})$	$\rightarrow$	$P(\mathbb{K}) \times P(\mathbb{K})$
Iteration	*	$P(\mathbb{K})^*$	$\rightarrow$	$P(\mathbb{K})$

2.2 Bildung der transitiven Hülle

1) der Automat definiert die Relation der Zustandsübergänge

2) die Relation hat eine transitive Hülle
(e.g. Warshalls Algorithmus
für gewichtete Graphen)

$$A = F_4 \circ (F_3 \circ F_2)^* \circ F_1 \quad | \quad F_4 \circ F_1$$

Vorsicht: es ist nicht die Kleene-Algebra!

2.3 Reduktion der Hülle

Automat läßt sich eindeutig reduzieren

$$A = F_4 \circ F_5 \circ F_1 \mid F_6$$

mit reduzierten Funktionen

$$F_5 = (F_3 \circ F_2)^* \qquad x_4=x_1+x_2$$
$$F_6 = F_4 \circ F_1 \qquad x_4=0$$

* F_5 wird durch diskrete Integration errechnet, s.u.

3.1 Algebra über den Vektoren

$(x \mid y) \mid z$	$=$	$x \mid (y \mid z)$	(1)	$x \mid y$	$=$	$y \mid x$	(5)
$(x \circ y) \circ z$	$=$	$x \circ (y \circ z)$	(2)	$1 \circ x$	$=$	x	(6)
$(x \mid y) \circ z$	$=$	$(x \circ z) \mid (y \circ z)$	(3)	x^*	$=$	$(x^* \circ x) \mid 1$	(7)
$x \circ (y \mid z)$	$=$	$(x \circ y) \mid (x \circ z)$	(4)				

$\circ$ Komposition

$\mid$ Partitionierung

$*$ Iteration

$x, y, z \in \mathbb{L} \times \mathbb{F}$

es ensteht ein unitärer Semiring

4.1 Interpretation von Variablen

Variable werden nicht als Werte

$x_i = x_i + 1,$

sondern als Funktionen interpretiert

$f(x_i) := f(x_i) + 1,$

und es wird nach der zu x_i gehörigen Funktion gefragt.

4.2 Diskretes Integral - unbestimmt

$n \in \mathbb{N}^0$ sei der Iterator
$\oint \subseteq \mathbb{F} \to \mathbb{F}$

Konstante	$\oint c \, dx_i$	=	c
Variable	$\oint x_j \, dx_i$	=	x_j
Nachfolger	$\oint succ(x_i) \, dx_i$	=	$x+n$
Vorgänger	$\oint pred(x_i) \, dx_i$	=	$x-n$
Summand	$\oint x_i + c \, dx_i$	=	x_i+n*c
Faktor	$\oint x_i * c \, dx_i$	=	$x_i * c^n$
Kettenregel	$\oint f_i(g_j(x),y) \, dx_i$	=	$\oint f_i(\oint g_j(x) \, dx_j, y) \, dx_i$
Vektoren	$\oint x \, dx$	=	$(\oint x_1 \, dx_1, \ldots \oint x_n \, dx_n)$

Konstanten oder Wörter mit Linksfaktor lassen sich integrieren

4.3 Diskretes Integral - bestimmt

$n \in \mathbb{N}^0$ sei der Iterator,
$b(x)$ das Integrationsgebiet, $b^{-1}(i)$ Umkehrfunktion
$\oint \subseteq \mathbb{F} \rightarrow \mathbb{F}$

Konstante	${}_{b(x)}\oint c\ dx_i$	=	c
Variable	${}_{b(x)}\oint x_j\ dx_i$	=	x_j
Nachfolger	${}_{b(x)}\oint succ(x_i)\ dx_i$	=	$x+b^{-1}(i)$
Vorgänger	${}_{b(x)}\oint pred(x_i)\ dx_i$	=	$x-b^{-1}(i)$
Summand	${}_{b(x)}\oint x_i+ c\ dx_i$	=	$x_i+b^{-1}(i)*c$
Faktor	${}_{b(x)}\oint x_i* c\ dx_i$	=	$x_i* c^{b^{-1}(i)}$
Kettenregel	${}_{b(x)}\oint f_i(g_j(x),y)\ dx_i$	=	${}_{b(x)}\oint f_i(\oint g_j(x)\ dx_j,y)\ dx_i$
Vektoren	${}_{b(x)}\oint x\ dx$	=	$({}_{b^{-1}(i)}\oint x_1\ dx_1, \ldots {}_{b^{-1}(x)}\oint x_n\ dx_n)$

$b^{-1}(x)$ muß einen Linksfaktor enthalten

5 Grenzen und Ziele

Einschränkungen, z.Zt.:

- geschlossene Lösung für Polynome mit deg(p(x))>2
- funktionsalternierende Folgen (e.g. Collatz-Folge)
- komplexe, μ-Rekursion

...

Zielvorstellung:

Elegantes Beweissystem für

- prozedurale Sprachen
- Hardwaresprachen
- ...

Vielen Dank für Ihre Aufmerksamkeit