Protokoll zur Vorlesung
Theoretische Informatik I

Datum : 28.04.2000 13:30 Uhr - 15:30 Uhr

Themen :

TOP 1	Satz G und H als Fortsetzung von Kapitel 2 " Grenzen der Algorithmisierung "
TOP 2	Church` sche These
TOP 3	Der Maschinenbegriff
TOP 3.1	Definition von abstrakten Maschinen
TOP 3.2	Beispielmaschinen
TOP 3.3	Einschränkungen des Maschinenbegriffs

Zu TOP 1 :

<u>Satz G :</u>

Behauptung : Es gibt keinen Algorithmus, der bei Eingabe eines beliebigen Algorithmus $\alpha \in X^*$ und beliebiger Wörter $w \in X^*$ feststellt, ob $w \in D(f_\alpha)$ gilt oder nicht.

Präziser : Die totale Funktion $h : X^* \times X^* \to X^*$ mit

$$h(\alpha,\omega) = \begin{cases} '\,ja', \ falls\ w \in D(f_\alpha) \\ '\,nein'\ sonst \end{cases}$$

ist nicht berechenbar.

Beweistechnik :
- Reduktion auf Selbstanwendungsproblem
- Widerspruchsbeweis

Beweis :

Annahme : h ist berechenbar, dass heisst es gibt einen Algorithmus, der das Halteproblem löst. Wir definieren eine neue Funktion:

$g : X^* \to X^*$ mit g total.
$g(\alpha) = h(\alpha, w)$

Beschreibt das Selbstanwendungs-problem

Mit h ist auch g berechenbar, d.h. es gibt mindestens einen Algorithmus γ, der g berechnet. g beschreibt hier das Selbstanwendungsproblem welches zu einem Widerspruch führt, daraus folgt, daß die Annahme falsch und die Behauptung der nicht Berechenbarkeit des obigen Algorithmus richtig ist.

Bemerkung : Satz G gilt für alle Algorithmen α und automatischen Analysen. Für speziell vorgegebene Algorithmen lassen sich durchaus Beweise zur Terminierung / Korrektheit finden.

<u>Satz H :</u> Dieser Satz ist nur umgangssprachlich, also nicht exakt formuliert. Sei E eine beliebige Eigenschaft, die von mindestens einer, aber nicht allen berechenbaren Funktionen $f : X^* \rightarrow X^*$ erfüllt wird. Dann gibt es keinen Algorithmus, der für beliebige Algorithmen $\alpha \in X^*$ ausgibt, ob f_α die Eigenschaft E erfüllt ist oder nicht. D.h. allgemeine Eigenschaften lassen sich nicht durch ein Programm testen.

Zu TOP 2 :

2.4 Church' sche These

Bisher haben wir Aussagen über Funktionen und Programmen / Algorithmen getroffen.
Frage :

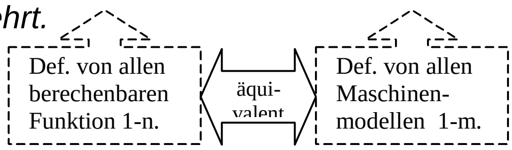
„Lassen sich die Aussagen auf Maschinen übertragen?" oder präziser „Habe ich mit Algorithmen den Leistungsumfang von Maschinen erfaßt, oder können Maschinen mehr?"

Problem :

Was ist ein Algorithmus und was ist die Leistungsfähigkeit von Algorithmen? Es besteht also ein Mangel an formalen Definitionen.

Schon 1936 hat A. Church dies in einer These formuliert:

Jede im intuitiven Sinne <u>berechenbare Funktion</u> ist <u>maschinell</u> berechenbar und umgekehrt.



Man beachte, daß die Church' sche These nicht mathematisch zu beweisen ist. Wann immer ein Algorithmus mechanisch dargestellt ist, so kann man ihn auf eine Maschine überführen:

Maschinenbegriffe	→	Äquivalent
Algorithmen	→	Äquivalent
Algorithmus	→	Maschine (überführbar).

Da das Adjektiv „intuitiv" in der Church 'schen These informell (**nicht formal**) verwendet ist, kann kein formaler Beweis geführt werden.

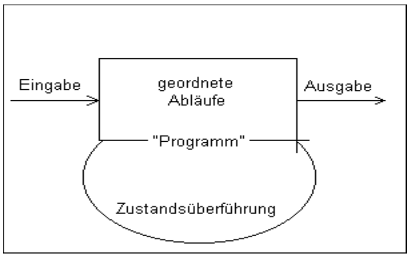
Zu TOP 3

3. Maschinen

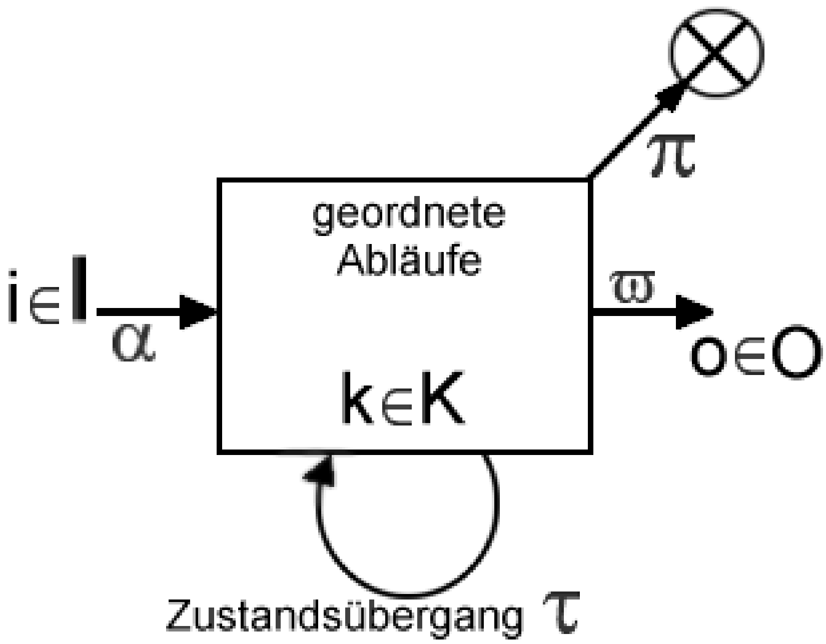
Welches sind die typischen / wesentlichen / unveränderlichen Merkmale von Maschinen ?

Merkmale :

- Inputs (Eingaben)
- Outputs (Ausgaben)
- Zustände (innere)
- „fertig“ – Mechanismus



- „Programm“: Ablaufsteuerung, die in Reaktion auf Eingaben und den aktuellen Zustand eine Zustandsänderung herbeiführt.



Beispiel : Ein Getränkeautomat für Getränke zum Preis von 20 Pfennig.

Eingaben = {5,10} Pfennig
Ausgaben = {G, GW, nichts} - Getränk
- Wechselgeld + Getränk
- nichts

Eingaben → $\boxed{\text{Zustand}}$ → Ausgaben

mögliche Zustände: { 0 Pf eingen. , 5 Pf eingen., 10 Pf eingen., 15 Pf eingen., 20 Pf eingen., 25 Pf eingen., 30 Pf eingen., Getr. ausgeben}

Zustandsübergangsfunktion :

Ausgangszustand	**Eingabe** ***5 Pf***	***10 Pf***	**Ausgabe**
0 (Pf)	5	10	-
5 (Pf)	10	15	-
10 (Pf)	15	20	-
15 (Pf)	20	25	-
20 (Pf)	25	30	-
25 (Pf)	-	-	Getränk ausgeben, zurück zu 0 Pfennig
30 (Pf)	-	-	Getränk + 5 Pf ausgeben, zurück zu 0 Pfennig

Zu TOP 3.1

Definition A : Eine Abstrakte Maschine M beschreibt man durch ein 7 – Tupel $M = \{I, O, K, \alpha, \omega, \tau, \pi\}$.

Wobei folgendes gilt :

I	Menge der Eingaben
O	Menge der Ausgaben
K	Menge der Konfigurationen
$\alpha : I \to K$	Eingabefunktion
$\omega : K \to O$	Ausgabefunktion
$\tau : K \to K$	Überführungsfunktion (Zustandsüberführungdfunktion)
$\pi : K \to \{1,0\}$	Halteprädikat

$E = \{k \in K \mid \pi(k) = 1\}$ sei die Menge der Endkonfiguration.

Allgemeine Arbeitsweise eines solchen Automaten :

$$\begin{tikzcd} I \arrow[r, "\alpha"'] \arrow[rrrrrr, bend left, "f_m"] & K \arrow[r, "\tau"'] & K \arrow[r] & \ldots \arrow[r, "\tau"'] & K \arrow[r, "\tau"'] & E \arrow[r, "\omega"'] & O \end{tikzcd}$$

Laufzeit der Maschine

Definition B : Sei M = (I, O, K, α, ω, τ, π) eine Abstrakte Maschine. Die Laufzeit t_M von M für eine Eingabe $i \in I$ ist definiert durch
$t_M(i) = \min \{ m \in N \mid \pi(\tau^m(\alpha(i))) = 1\}$.
Falls min nicht definiert ist, dann setze $t_M = \infty$. D.h. die Maschine hält nicht an.

Definition C : Sei M = (I, O, K, α, ω, τ, π) eine Abstrakte Maschine. Die vom M berechnete Funktion f_M ist definiert durch $f_M : I \to O$ mit

$$f_M(i) = \begin{cases} \omega\left(\tau^{t_M(i)}(a(i))\right), & falls\ t_M(i) < \infty \\ \perp\ sonst. \end{cases}$$

Zu TOP 3.2

Beispiele :

1) Kommen wir nun auf unseren Getränkeautomaten noch einmal zurück und betrachten ihn nun anhand unserer Definitionen.

M_1 = (I_1, O_1, K_1, α_1, ω_1, τ_1, π_1) mit

I_1 = {5,10},

O_1 = {G} x N_0 — N_0 = *Wechselgeld*

K_1 = I_1^* x N_0 — *Speichergeld*

α_1 : $I_1 \rightarrow K_1$ mit ω(x, s) = (G, S – 25)

Gibt Getränk und vermindert das eingegebene Geld um 25 und gibt den Restbetrag aus.

$$\pi : K_1 \rightarrow \{0,1\} \; mit \; \pi(x,s) = \begin{cases} 1, \; falls \;\; x = \in, s \geq 25 \\ 0, sonst \end{cases}$$

τ(x, s) = (y, s + α), falls x = αy, $\alpha \in$ {5, 10}, y $\in$ I*

Ablauf :

$$i = 510510 \xrightarrow{\alpha} (510510{,}0) \xrightarrow{\tau} (10510{,}5) \xrightarrow{\tau} \ldots$$
$$\xrightarrow{\tau} (10{,}20) \rightarrow (\varepsilon{,}30) \xrightarrow{\omega} (G, S)$$

Aus diesem Aufbau folgt das am ende des Vorgangs ein Getränk und 5 Pf Wechselgeld vom Automaten zurückgegeben werden.

Beispiel 2 :

2.) Abstrakte Maschinen sind außerordentlich leistungsfähig, sie können jede beliebige Funktion berechnen, sogar in nur einem einzigem Schritt. Hier ist das Beispiel einer abstrakten Maschine M die einfach das Ergebnis einer totalen Funktion ausgibt.

Sei f: $\mathrm{N} \rightarrow \mathrm{N}$ einer beliebigen totalen Funktion.

Def: Abstrakte Maschine M mit $f=f_M$.

$M_2 = (I_2, O_2, K_2, \alpha_2, \omega_2, \tau_2, \pi_2)$ mit $I_2=O_2=N$ und $K_2=N_0\times N_0$

$\alpha_2 : N\rightarrow N_0\times N_0$ mit $\alpha_2(n) = (n,0)$

$\omega_2 : N_0\times N_0\rightarrow N$ mit $\omega_2(n) = (0,n)n$

$\tau_2 : N_0\times N_0\rightarrow N_0\times N_0$ mit $\tau_2(n,0) = (0,f(n))$ für alle $n\in N_0$

$$\pi_2 : N_0\times N_0\rightarrow\{0,1\} \quad \text{mit } \pi_2(m,n) \begin{cases} 1 & \text{falls } m=0 \\ & \text{sonst } 0 \end{cases}$$

$t_{M_2}(n) = \min \{m\in N\ (\pi (\tau^m (\alpha (n))) = 1 \} = 1,$

$f_{M_2}(n) = \omega (\tau^1 (\alpha (n))) = \omega (\tau(n, 0)) = \omega (0, f (n)) = f (n).$

Zu TOP 3.2

1.) Wir nehmen an es existieren Maschinen die mehr vermögen Algorithmen, dies steht jedoch im Widerspruch mit der Church´schen These daraus folgt :

2.) Abstrakte Maschinen sind nicht realistisch!

Realistische Automaten müssen endlich erzeugt, d.h. :

- Endlicher Speicher
- Endliche Menge von Grundzuständen
- Irgendwie auf realistische Weise beschränken

Daraus lässt sich schliessen, das zu obengenannter Maschine M_2 kein auf diese Weise realisierter, endlicher Automat existieren kann, da von einer unendlichen Menge auf eine andere abgebildet wurde.

Beispiele für endliche Automaten (endliche Menge von Zuständen, reduzierter Zugriff auf die Eingabe…):

- Kellerautomaten (Stack) [zusätzlicher Speicher m als Stack, Operationen push,pop usw.)
- Linear beschränkter Automat (speicherbeschränkt in Abhängigkeit von der Eingabe)
- Turingmaschinen (Operationsbeschränkung, mit Speicherband)
- Registermaschinen (Operationsbeschränkung, mit Speicherband und Arbeitsregistern)

Von oben nach unten nähern sich diese Automaten immer weiter den heutigen Computern an, und werden somit mächtiger.