Theoretische Informatik II
Vorlesung vom 15.12.00

Fortsetzung des Beweises für die Simulation einer TM durch eine RM

(2.3) $\delta(s,a)=(s',a',B)$

Fall 1: B=h (stop)

R1:=R5*(2^{R4})
R2:=R6
Codiere R2 per Primzahlcodierung in R1 hinein
[R1 enthält dann die Codierung des TM-Bandes]
Gehe in Endzustand

Fall 2: B=R

Nun muß $\prod_{j=2}^{r} p_j^{nr(v_j)}$ in $\begin{cases} \prod_{j=2}^{r} p_{j-1}^{nr(v_j)}, \text{falls } r \geq 2 \\ p_1^{nr(b)}, \text{falls } r = 1 \end{cases}$

und $\prod_{j=1}^{r} p_j^{nr(u_j)}$ in $p_1^{nr(a')} \cdot \prod_{j=1}^{r'} p_{j+1}^{nr(u_j)}$ umgewandelt werden.

Folgendes Programmstück leistet dies (n(a') steht in R4):

R2:=2^{R4}; R1:=1; R7:=2; {N-te Primzahl}
R8:=2; {(N-1)-te Primzahl}
while R5≠1 do {R5 wird fortlaufend durch alle Primzahlen geteilt}
    R9:= R7-te Primzahl;
    [R10:= d ∧ R5:=R5 div $(R9)^d$];
    R1:=R1*($R8^{R10}$); {Nächster Buchstabe x mit nr(x)=d wurde umverschlüsselt}
    R8:=R9; R7:=R7+1;
    od;
if R7=2 then R1:=$2^{nr(b)}$ fi {Sonderfall: r=1, fertig mit v}
while R6≠1 do
    R9:=R7-te Primzahl;
    [R10:=d ∧ R6:=R6 div $(R8)^d$];
    R2:=R2*($R9^{R10}$);
    R8:=R9; R7:=R7+1;
    } analog wie oben für a
od;

Fall 3: B=L (analog)

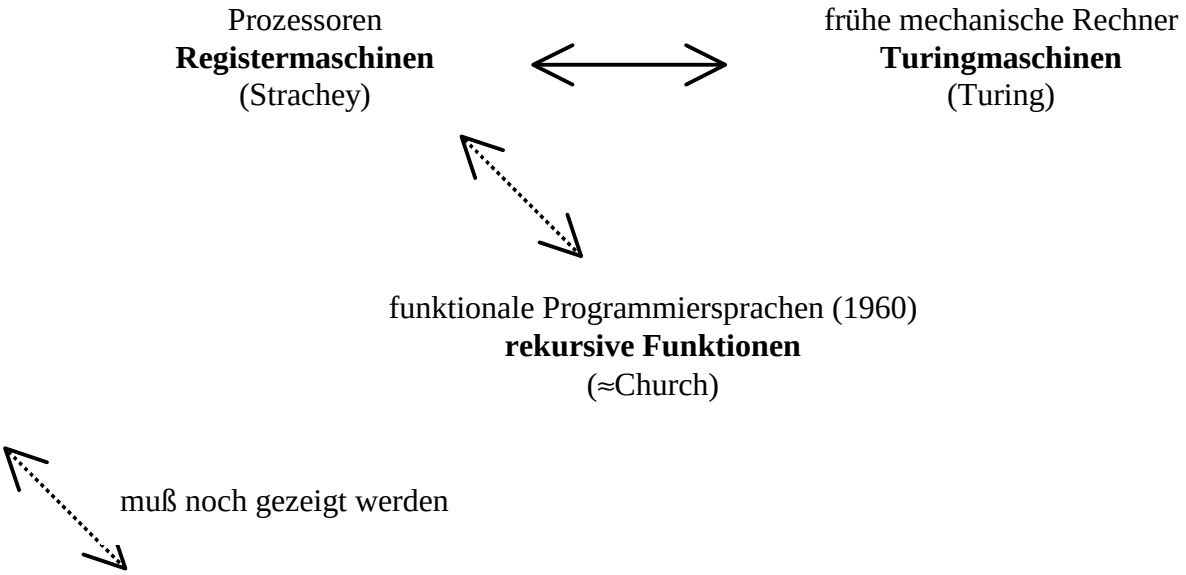
(3) Ausgabe: Konkateniere die Folge der decodierten Exponenten (v rückwärts lesen). Entferne hinten und Blanks.

Die so definierte RM simuliert jeden Schritt von τ. Dies ist für (1) und (3), sowie (2.1) und (2.2) klar. Für (2.3) müsste man die Korrektheit der Äquivalenz der Programme in a) bis o) zeigen und damit beweisen, dass die Simulation korrekt ist.

$\tau_{A,A} \subseteq Rm$ ist gezeigt. (bei geeigneter Codierung)□

Bem.: Man kommt in der Theorie mit 2 Registern statt 20 aus. Überaus mühsame Codierung und Simulation. 1 Register reicht nicht.

Berechnungsmodelle:



6.3 μ-rekursive Funktionen

Bezeichnungen:

- Nachfolgerfunktion $N: \mathbb{N}_0 \to \mathbb{N}_0$ mit $N(x)=x+1$
- Projektion $\prod_i^n : \mathbb{N}_0^n \to \mathbb{N}_0$ mit $\prod_i^n (x_1,...,x_n) = x_i$
- Konstante $C_i^n : \mathbb{N}_0^n \to \mathbb{N}_0$ mit $C_i^n (x_1,...,x_n) = i$

Def. K: Seien $g: \mathbb{N}_0^m \to \mathbb{N}_0$, $h_i: \mathbb{N}_0^k \to \mathbb{N}_0$, i=1, ..., m Funktionen.
Die Funktion $f:=g(h_1,..., h_m)$ mit
$f: \mathbb{N}_0^k \to \mathbb{N}_0$ und $f(x_1,..., x_k)=g(h_1(x_1,..., x_k), ..., h_m(x_1,..., x_k))$ geht aus g, h_1, ..., h_m durch simultane Einsetzung hervor.

Def. L: 1) $f: \mathbb{N}_0^k \to \mathbb{N}_0$ geht aus $g: \mathbb{N}_0^{k-1} \to \mathbb{N}_0$ und $h: \mathbb{N}_0^{k+1} \to \mathbb{N}_0$ durch primitive Rekursion hervor, wenn $f(x_1,..., x_{k-1}, 0)=g(x_1,..., x_{k-1})$, $\forall x_1,..., x_{k-1} \in \mathbb{N}_0$ und
$f(x_1,...,x_{k-1},y+1)=h(x_1,\ldots,x_{k-1},y,f(x_1,\ldots,x_{k-1},y))$ $\forall x_1,\ldots,x_{k-1},y \in \mathbb{N}_0$

2) f entsteht $g: \mathbb{N}_0^{k+1} \to \mathbb{N}_0$ durch Anwenden des μ-Operators, wenn
$f(x_1,...,x_k)=\text{Min}\{y \mid g(x_1,...,x_k,y)=0$ und $g(x_1,...,x_k,y)$ ist definiert für alle $j<y\}$, falls Min defininert.
Schreibweise: $f=\mu g$

Bsp.: k=1:
g(x,700)=0, $g(x,430)=\perp$
700 ist erstes y mit g(...)=0. $\mu g=\perp$

g(x,700)=0, $g(x,j)\neq 0$, aber definert $\forall\ 0\leq j\leq 699 \Rightarrow f(x)=700$

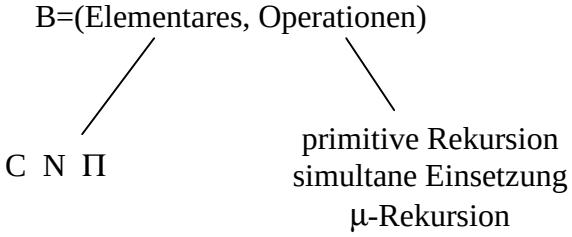
Programm für μ:

```
y:=0;
while g(x1,...,xk,y)≠0 do y:=y+1 od;
```

Def. M: 1) f: $\mathbb{N}_0^k \to \mathbb{N}_0$ heißt primitiv-rekursiv, wenn f Nachfolgerfunktion, Projektion, Konstante ist, oder wenn f durch simultane Einsetzung oder durch primitive Rekursion aus primitiv-rekursiven Funktionen hervorgeht. Die Klasse der primitiv-rekursiven Funktionen bezeichnen wir mit F_{prim}.

2) f heißt μ-rekursiv, wenn f primitiv-rekursiv ist oder durch Anwenden des μ-Operators aus einer μ-rekursiven Funktion entsteht. Die Klasse der μ-rekursiven Funktionen bezeichnen wir mit F_μ , die Teilklasse der totalen μ-rekursiven Funktionen mit R_μ.

Bem: Fundamentale Idee der Informatik: Bausteinsystem:



Beispiele: 1) Addition: x+y=(x+y-1)+1 primitiv rekursiv:

$g:=\prod_1^1$, $k:=N\circ\prod_3^3$, also f(x,y)=x+y wird definiert durch

$f(x,0)=\prod_1^1(x)$ (=x)

$f(x,y+1)=N(\prod_3^3(x,y,f(x,y))$ (=N(f(x,y)) = f(x,y)+1 = (x+y) +1)