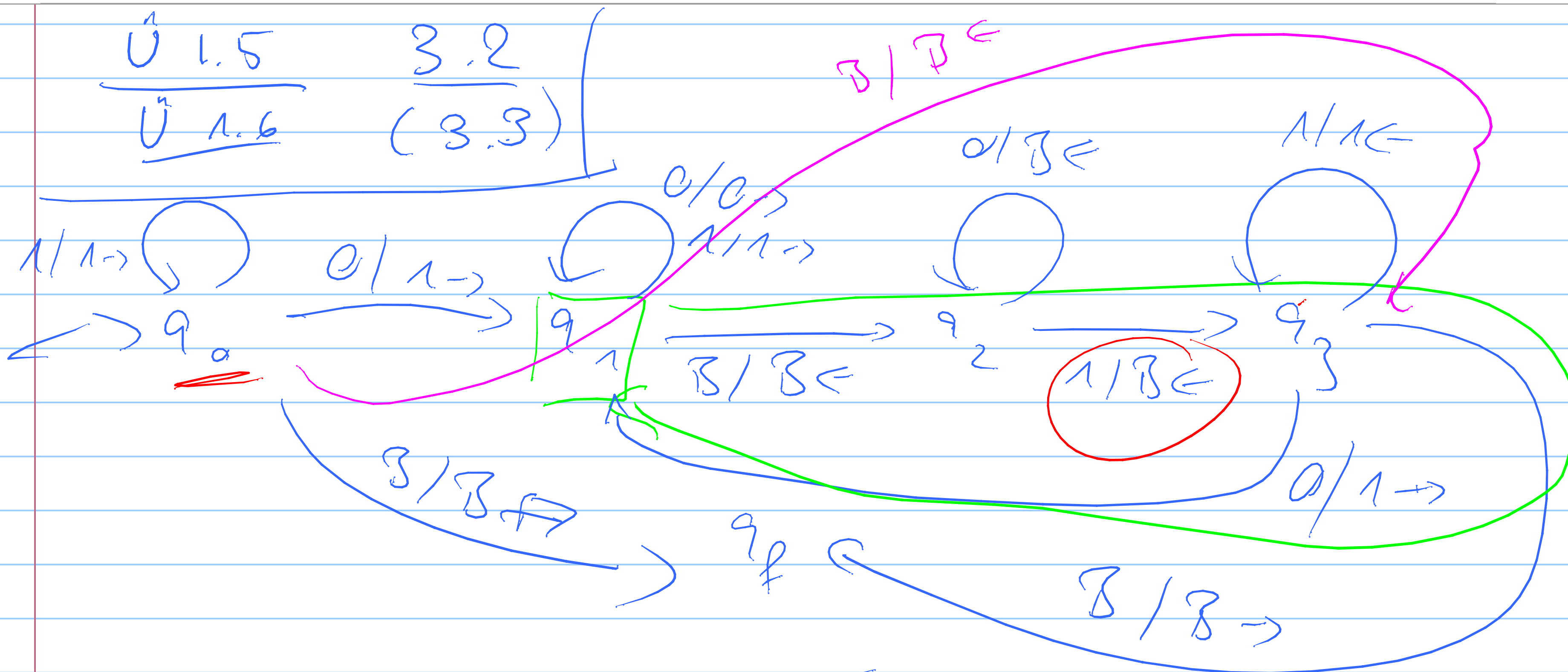
Tutorium 09-05-13

Notiztitel 5/6/2009

Ü 1.5 3.2

Ü 1.6 (3.3)



- Entfernt alle Nullen aus Eingabe
  Sammelt dabei die Einse auf

Vermutung $f(w) = 1^{\#_1(w)}$

Ausnahme $w = 1^n$ $f(w) = \varepsilon$

weil Maschine rechts von w endet

↗ Maschine verstößt gegen Konvention, daß Ausgabe rechts von Kopf steht

ALT

$$f(w) = \begin{cases} \varepsilon & w = 1^n \text{ für ein } n \\ \#_1(w) & \\ 1 & \text{sonst} \end{cases}$$

bei Einhalten der Konvention

Analyse: $\alpha(w) = (\varepsilon, q_a, w)$

$\boxed{\text{1. Fall} \quad w = 1^n}$ für ein n

$\alpha(w) \vdash^n (1^n, q_a, B) \vdash (1^n B, q_e, B)$ terminal

$\omega(1^n B, q_e, B) = \varepsilon$

also $f(1^n) = \varepsilon$

2. Fall $\boxed{w = 1^i 0 v}$ $v \in \{0,1\}^*$

$$\alpha(w) \vdash^{i} (1^{i}, q_{\alpha}, 0v) \vdash (1^{i+1}, q_1, v)$$

$$\vdash (1^{i+1} v, q_1, B)$$

müssen v aufschlüsseln

$$v = u x 1\, 0^{k} \qquad k > 1 \;?$$

d.h.

$$\alpha(w) \vdash^{*} (1^{i+1} u 1 0^{k}, q_1, B)$$

$$\vdash (1^{i+1} u x 1 0^{k-1}, q_2, 0)$$

$$\vdash^{k-1} (1^{i+1} u x, q_2, 1)$$

$$\vdash (1^{i+1} u, q_3, x)$$

v könnte auch anders aussehen

- keine 0 am Ende
- keine 1 vor den Nullen

$\longrightarrow$ zu kompliziert!!

komplizierte Form

$$w = 1^{i_1} 0^{j_1} 1^{i_2} 0^{j_2} \dots 1^{i_n} 0^{j_n} 1^{i_{n+1}}$$

$i_k, j_k > 0 \qquad n \geq 1$

$i_{n+1} \geq 0$

Zeige

$$(v', q_1, v) \vdash^* (u', q_1, u)$$

dann $\#_1(v') + \#_1(v) = \#_1(u') + \#_1(u)$ Anzahl 1 unverändert

und $\#_0(v'v) > \#_0(u'u)$

Detailbeweis Induktion

oder plausibles verbales Argument

1) die weggenommene Eins wird wieder zugefügt im letzten Schritt

2) eine Null verschwindet mindestens

Iterativ: (Induktion)

alle Nullen verschwinden

anzahl Einsen bleibt

im letzte durchlauf

$(\varepsilon, q_1, 1^n) \vdash^{n+1} (1^{n-1}, q_2, 1)$

$\vdash (1^{n-2}, q_3, 1)$

$\vdash^* (\varepsilon, q, B1^{n-1})$

$\vdash (\varepsilon, q_e, 1^{n-1})$

LOGIK

Programmverifikation ist mühsam!

aber nicht zu automatisiere

a

1.7 Simulation eine TM die (!)

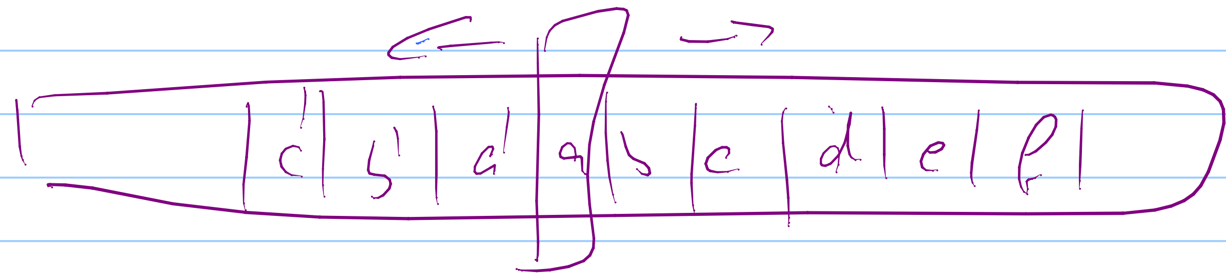
2k S(|w|) Zellen braucht, durch

eine TM die S(|w|) braucht (!)

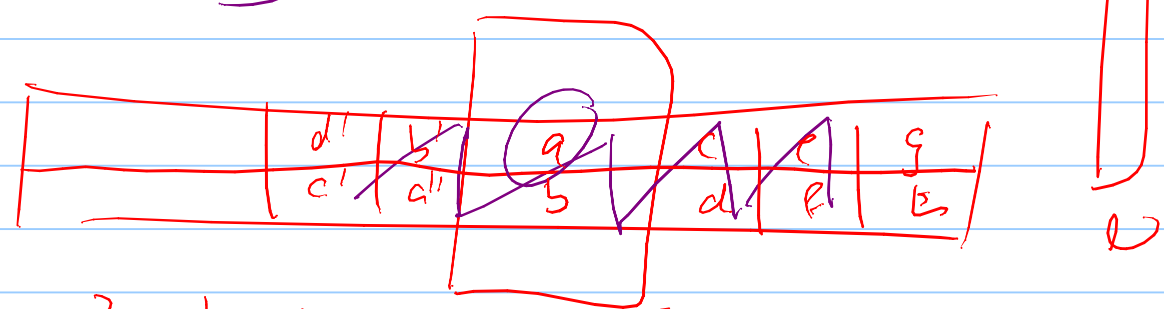
Tip 2-Spur Maschine

Sinn: Konstante Faktoren bei Platzkalkulation sind unbedeutend

Idee Bandzelle von $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$



falten



verwalte im Zustand, welche Spur ich lesen muß

$M' = (Q', \Sigma, \Gamma', \delta', q_0', (B,B), F')$

$\Gamma' = \Gamma \times \Gamma$ Paare von Bandzelle für M

Q' $Q + Q$ (disjunkte Vereinigung)

$= \{ q^{o} \mid q \in Q \} \cup \{ q^{u} \mid q \in Q \} \cup Q$ ← Zwischen Zustand

ich lese Spur 1 ich lese Spur 2

$q_0' = q_0'$ Starte in q_0, lese erste Spur

δ': wenn $\delta(q, X) = (q_j, Y, R)$

dann $\delta'(q', (X, Z)) = (q_j, (Y, Z), R)$

$\delta'(q_j, (Y, Z)) = (q_j'', (Y, Z), L)$

$\delta(q, X) = (q_j, Y, L)$

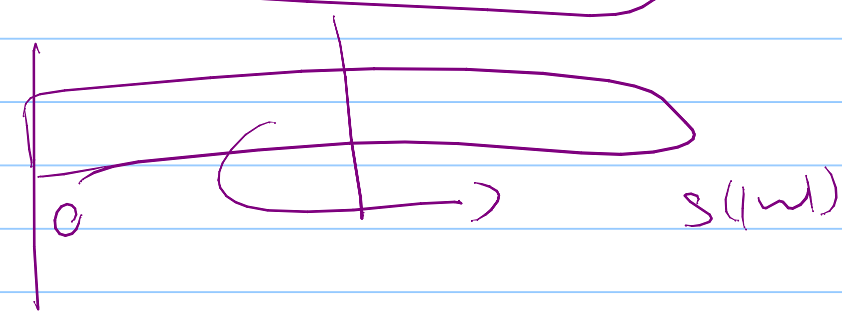
dann $\delta'(q', (X, Z)) = (q_j'', (Y, Z), L)$

analog für q''

mit weitere Kopie von Q

$$\boxed{Q + Q \triangleq Q \times \{0, 1\}}$$

bei halbseitige Band



prim-rek Funktionen:

Zeigen Sie, dass .. p.r. ist (und geben Sie Programm an)

3.2.a

$$f_>(m, n) = \begin{cases} 1 & m > n \\ 0 & \text{sonst} \end{cases}$$

Idee

$$m > n \iff m - n > 0$$
$$\iff m \dot{-} n \neq 0$$
$$\iff \mathrm{sign}(m \dot{-} n) = 1$$

$$m \leq n \iff \mathrm{sign}(m \dot{-} n) = 0$$

$$f_>(m,n) = \mathrm{sign}(m \dot{-} n)$$

weil $\dot{-}$, sign pr. ist, ist $f_>$ p.r.

$$\boxed{f_> = \mathrm{sign} \circ \mathrm{sub}}$$

VL

e) $$f_{prim}(n) = \begin{cases} 1 & n \text{ primzahl} \\ 0 & \text{sonst} \end{cases}$$

n primzahl wenn keine Zahl y zwische 1 und n Teile von n

Wenn kleinste Zahl $y > 1$, die n teilt gleich n ist

$$\min\{1 < y \leq n \mid y \text{ teilt } n\} = n$$

$f_{divides}(y,n) = 1$

Beschränkte Minimierung:

$$\min\{1 < y \leq n \mid 1 - f_{divides}(y,n) = 0\} = n$$

verwende beschränkte Minimierung, definiert durch:

VL

$$Mn_g[f] = \begin{cases} \min\{y \leq g(n) \mid f(n,y) = 0\} & \text{falls dieses existiert} \\ g(n)+1 & \text{sonst} \end{cases}$$

als Begründung würde jetzt ausreichen, daß f_{prim} zusammengesetzt ist durch $f_{divides}$, Subtraktion, Test auf ≥ 1 und beschränkte Minimierung. Weil diese Komponenten alle prim. rek. sind, gilt dies auch für f_{prim}

Das 'Programm' kann man auch ansehen, muß aber aufpassen

daß die beschränkte Minimierung bei $g(n)+1$ endet

Hierzu müssen wir g und f bestimmen

- wir wissen, daß n Teiler von n ist
- wenn n Primzahl ist, dann ist

$$\min\{ y \leq n-1 \mid 1 \dot{-} f_{divides}(y,n) = 0 \}$$

nicht definiert, also gibt die beschränkte Minimierung für $g(n) := n \dot{-} 1 = p(n)$ genau den Wunschwert n

- jetzt müssen wir noch sicherstellen, daß die Suche erst bei zwei beginnt, also

$$\min\{ y \leq n-1 \mid 1 \dot{-} f_{divides}(y,n) = 0 \ \underline{\text{und}}\ f_{\geq}(1,n) = 0 \}$$

undefiniert ist.

Die Konjunktion läßt sich durch eine Addition beschreiben, d.h. es muß

$$\left(1 \dot{-} f_{divides}(y,n) + f_{\geq}(1,n)\right) = 0 \quad \text{sein}$$

Damit ergibt sich

$$f(n,y) = 1 \dot{-} f_{divides}(y,n) + f_{\geq}(1,n)$$

umgeschrieben für Argumente (n,y)

$$=$$

$$add \circ \left(sub \circ \left(c_1^2, f_{divides} \circ (pr_2^2, pr_1^2)\right), f_{\geq} \circ (c_1^2, pr_2^2)\right)(y,n)$$

Und damit ist f_{prim} programmierbar als

$$f_{prim} = Mn_p\left[add \circ \left(sub \circ \left(c_1^2, f_{divides} \circ \left(pr_2^2, pr_1^2\right)\right), f_{>} \circ \left(c_1^2, pr_2^2\right)\right)\right]$$

Wie gesagt, für den Beweis, daß f_{prim} prim. rek. ist, ist die explizite Angabe des Programms nicht erforderlich

---

Heute war nicht mein Tag. Es tut mir Leid, daß so viele Details fehlen hatten. Aber hoffentlich haben Sie daraus auch gelernt.