Theoretische Informatik 1

Teil 4

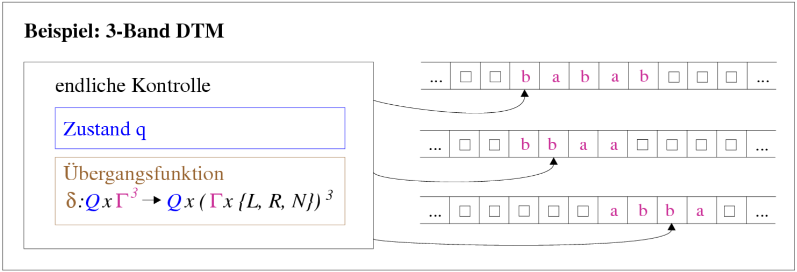
Bernhard Nessler

Institut für Grundlagen der Informationsverabeitung
TU Graz

SS 2008

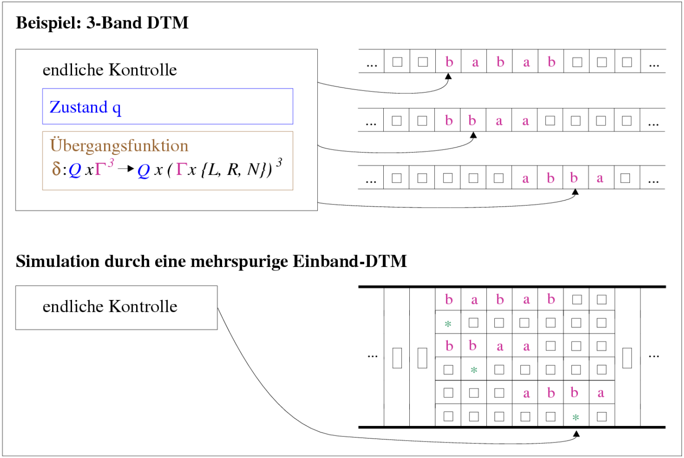
Übersicht

Mehrband-TM



- k Köpfe, also pro Schritt k Symbole veränderbar
- 1 Schritt = jeder Kopf bewegt sich
- Köpfe bewegen sich unanbhängig voneinander!
- Übergangsfunktion δ wird aufgeblasen
- k-Band DTM ist also schneller als 1-Band DTM

Äquivalenz k-DTM $\mathcal{T}'$ und 1-DTM $\mathcal{T}'$



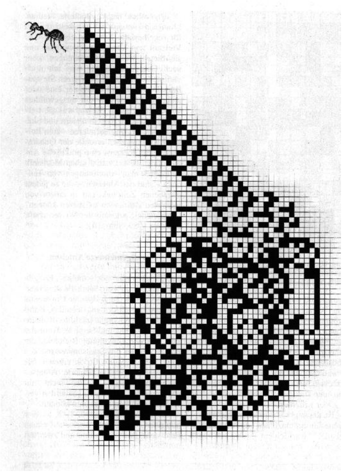
Äquivalenz k-DTM $\mathcal{T}$ und 1-DTM $\mathcal{T}'$

$\mathcal{T}' = (Q', \Sigma, \Gamma', \delta', q'_0, \square, F')$
$\Gamma' = \Gamma \cup (\Gamma \cup \{*\})^{2k}, \qquad * \notin \Gamma$
$q' \in Q'$: $q' = \langle q, b_1, \ldots, b_k, c, \ldots \rangle$, $q \in Q$, $b_i \in \Gamma \cup ?$
Start in q'_0: Eingabe umcodieren, dann $q' = \langle q_0, ?, \ldots, ?, 1, \ldots \rangle$
1 Schritt der k-DTM wird simuliert durch:

- startet in $q' = \langle q, ?, \ldots, ?, 1, \ldots \rangle$
- 1 Durchlauf von links nach rechts, solange $c = 1$
- dabei $\langle q, \ldots, ?, \ldots, 1, \ldots \rangle \ \rightarrow \langle q, \ldots, b_i, \ldots, 1, \ldots \rangle$, bis alle k Kopfsymbole in $q' = \langle q, b_1, \ldots, b_k, 1, \ldots \rangle$ gespeichert, dann $c = 2$
- $\delta(q, b_1, \ldots, b_k) = (p, \langle c_1, X_1 \rangle, \ldots, \langle c_k, x_k \rangle) = f(q')$
- schrittweises Bandupdate, dann $q' = \langle q, \ldots, c = 3, \ldots \rangle$
- rewind nach links, dann $q' = \langle p, ?, \ldots, ?, 1 \rangle$

Ausgabe umcodieren, wenn $q' = (q, \ldots), q \in F$

Turmiten, Ameisen



- Chris Langton, 1986
- 4 Zustände = aktuelle Richtung
- Feldfarbe wird immer invertiert
- Bewegungsrichtung rotiert

Turmiten haben dieselbe Berechnungsstärke wie TM.

Kostenmaße bei Eingabe w

Definition (Zeitkosten bei Eingabe w)

Die Funktion $t_{\mathcal{T}}(w) : \Sigma^* \mapsto \mathbb{N} \cup \{\infty\}$ gibt die Länge des (endlichen) Berechnungspfades der TM $\mathcal{T}$ bei der Eingabe w an, oder ∞, wenn dieser Berechnungspfad unendlich ist.

Definition (Platzkosten bei Eingabe w)

Die Funktion $s_{\mathcal{T}}(w) : \Sigma^* \mapsto \mathbb{N} \cup \{\infty\}$ gibt die Anzahl der Bandquadrate an, die während der Berechnug der TM $\mathcal{T}$ bei Eingabe von w besucht werden.

Satz (Platzkosten sind beschränkt durch Zeitkosten)

$$\forall \mathcal{T} : \forall w \in \Sigma^* : s_{\mathcal{T}}(w) \leq t_{\mathcal{T}}(w) + 1$$

Komplexität einer DTM

Definition (Zeitkomplexität)

Die Zeitkomplexität einer DTM $\mathcal{T}$ (in Abhängigkeit der Länge der Eingabe) ist definiert als $T_{\mathcal{T}}(n) = \max_{w \in \Sigma^* : |w| \leq n} t_{\mathcal{T}}(w)$

Definition (Platzkomplexität)

Die Platzkomplexität einer DTM $\mathcal{T}$ (in Abhängigkeit der Länge der Eingabe) ist definiert als $S_{\mathcal{T}}(n) = \max_{w \in \Sigma^* : |w| \leq n} s_{\mathcal{T}}(w)$

Satz (Platzkomplexität ist kleiner als Zeitkomplexität)

$$\forall \mathcal{T} : \forall n \in \mathbb{N} : S_{\mathcal{T}}(n) \leq T_{\mathcal{T}}(n) + 1$$

Problemarten

- **Konstruktionsprobleme (Optimierungsprobleme)**
  Zu einer Eingabe x (der Probleminstanz) soll die optimale Lösung, soferne sie existiert, bestimmt werden.
- **Funktionsberechnungen**
  Eingabe x, berechne $f(x)$. Lösung ist eindeutig.
- **Entscheidungsprobleme**
  Eingabe x, Ausgabe JA/NEIN bzw 1/0

Größte Bedeutung für Komplexitätstheorie haben Entscheidungsprobleme.
Anstelle von Konstruktionsproblemen werden die zugehörigen Entscheidungsprobleme betrachtet.

Beachte: Eingabecodierung ist Teil der Problemdefinition!!

Sprachprobleme (=Entscheidungsprobleme)

geg: Sprache $L \subseteq \Sigma^*$ und ein Wort $w \in \Sigma^*$
ges: Ist $w \in L$

characteristische Funktion:

$$f_L : \Sigma^* \mapsto \{0, 1\} : f_L(w) = \begin{cases} 1 & w \in L \\ 0 & w \notin L \end{cases}$$

Sprache einer Entscheidungsfunktion:

$$L = \{w \in \Sigma^* | f_L(w) = 1\}$$

Eine TM $\mathcal{T}$ ***entscheidet*** L, wenn $f_{\mathcal{T}} = f_L$. $\mathcal{T}$ hält immer nach endlich vielen Schritten. (L heißt *rekursiv*).
Eine TM $\mathcal{T}$ ***akzeptiert*** L, wenn $f_L(w) = 1 \Leftrightarrow f_{\mathcal{T}}(w) = 1$. $\mathcal{T}$ hält zumindest dann, wenn $f_L(w) = 1$. (L heißt *rekursiv aufzählbar*)

Sprachprobleme vs. Konstruktionsprobleme

Aus mehreren Ergebnissen eines Sprachproblems kann effizient auf die Lösung des zugrundeliegenden Konstruktionsproblems geschlossen werden.

geg: ungerichteter Graph $G = (V, E)$ und $k \geq 1$
ges 1: Enthält G eine Clique der Größe k?
ges 2: Knotenmenge der größten Clique aus G.

TM $\mathcal{T}_1$ löst 1. Problem. Wie kann unter mithilfe von $\mathcal{T}_1$ das 2. Problem effizient gelöst werden?

Lösungsidee: Kanten aus G entsprechend den Entscheidungen von T schrittweise entfernen.

Maximal soviele Aufrufe von $\mathcal{T}_1$ wie Kanten, also $O(poly(n))$.

Äquivalenz RM und DTM

Satz

Zu jeder Registermaschine $\mathcal{R}$ gibt es eine Turingmaschine $\mathcal{T}$, sodaß für die jeweils berechneten (partiellen) Funktionen gilt:

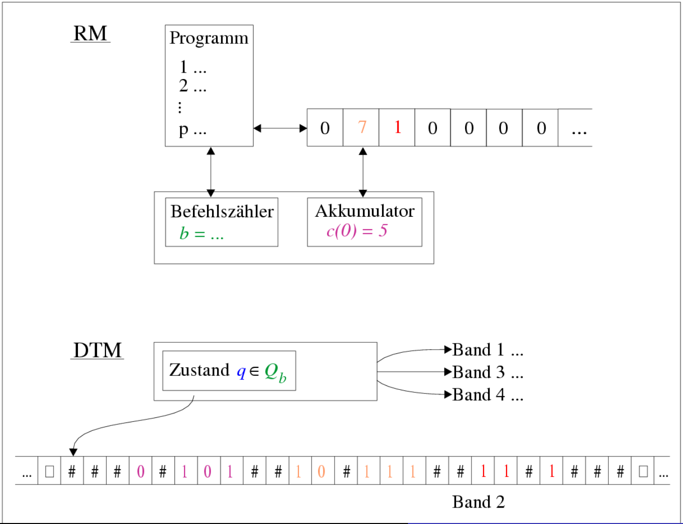
$$f_{\mathcal{T}}(bin(x_1)\#bin(x_2)\#\ldots\#bin(x_k)) = bin(f_{\mathcal{R}}(x_1,\ldots,x_k)).$$

Satz

Zu jeder Turingmaschine $\mathcal{T}$ (mit Ausgabealphabet $\{0,1\}$) gibt es eine Registermaschine $\mathcal{R}$, sodaß für die jeweils berechneten (partiellen) Funktionen gilt:

$$f_{\mathcal{R}}(w_1,w_2,\ldots,w_{|w|}) = bin^{-1}(f_{\mathcal{T}}(w))$$

Simulation RM durch DTM



Simulation RM durch DTM

RM: Programm mit p Befehlen, Speicher $r_0, \ldots, r_m$
4-Band DTM: $\Sigma = \{0, 1, \#\}, \Gamma = \{0, 1, \#, \Box\}$

- Band 1: Eingabe, Band 3: Ausgabe
- Band 4: Nebenrechnungen
- Band2: $\#\#\#0\#r_0\#\# \ldots \#\#bin(m)\#bin(r_m)\#\#\#$
- $Q = Q_0 \cup Q_1 \cup \ldots \cup Q_p \cup Q_{p+1}$
- Q_0-Zustände übersetzen Eingabe in Registerstruktur
- Q_{p+1}-Zustände übertragen Ergebnis r_0 auf Band 3
- $Q_i, 1 \leq i \leq p$ simuliert Befehl i

Simulation RM durch DTM

`274:   ADD  *24` $\Longrightarrow Q_{274}$

1. suche `##11000#`x`#` auf Band 2 (9q's)
2. kopiere x von Band 2 bis `#` auf Band 4 (1q)
3. suche `##`x`#`y`#` auf Band 2 (4q's)
4. kopiere y von Band 2 bis `#` auf Band 4 (1q)
5. Rewind Band 2 zum Ende des Accumulators (6q's)
6. Addiere bitweise Accumulator zu Band 4 (2q)
7. Ersetzte Accumulator durch Inhalt von Band 4 (4q's)

Q_{274} umfasst also 27 Einzelzustände

siehe dazu Übungsaufgabe

Simulation DTM durch RM, Variante 1

Zentrale Frage: Wie wird das Band dargestellt?
Variante 1: Ein Register pro Bandfeld, startend bei r_{10}:

- Bandfelder nummerieren: $\ldots, F_{-2}, F_{-1}, F_0, F_1, F_2, \ldots$
- Kopfposition: r_1, Inhalt: r_{r_1}, Zustand: r_2
- $F_{-k} \rightarrow R_{10+2k},\ F_0 \rightarrow r_{10},\ F_k \rightarrow r_{10+2k-1}$
- Rechts: $\Delta = 2$; Links:$\Delta = -2$
  `IF` r_1 `mod 2=1 THEN` r_1=r_1+Δ `ELSE` r_1=r_1−Δ
  `IF` r_1`=8 THEN` r_1`=11 ELSEIF` r_1`=9 THEN` r_1`=10`
- Δ, r_2, r_{r_1} über `IF`-Tabelle bestimmen
- Nachteil: indirekte Adressierung notwendig

Simulation DTM durch RM, Variante 2

Variante 2: 2 Stacks in je einem Register
Stack L in r_3 ist Bandinhalt links vom Kopf
Stack R in r_4 ist Bandinhalt rechts vom Kopf
Symbol unter dem Kopf in r_2, aktueller Zustand in r_1
Koniguration: $\beta_m \dots \beta_0 \mathbf{q} \alpha_0 \dots \alpha_n$

- $r_3 = \beta_0 + \beta_1 * |\Gamma| + \dots + \beta_m * |\Gamma|^m$
  $r_4 = \alpha_1 + \alpha_2 * |\Gamma| + \dots + \alpha_n * |\Gamma|^{n-1}$
  $r_2 = \alpha_0 \qquad r_1 = \mathbf{q}$
- `moveR:` $r_3 = r_3 * |\Gamma| + r_2$; $r_2 = r_4 \;\mathtt{mod}\; |\Gamma|$; $r_4 = r_4 \;\mathtt{div}\; |\Gamma|$
- `moveL:` $r_4 = r_4 * |\Gamma| + r_2$; $r_2 = r_3 \;\mathtt{mod}\; |\Gamma|$; $r_3 = r_3 \;\mathtt{div}\; |\Gamma|$

Simulation DTM durch RM, Variante 2

$$\delta(q_1, \alpha_1) = (q'_1, \alpha'_1, R)$$
$$\delta(q_2, \alpha_2) = (q'_2, \alpha'_2, L)$$
$$\delta(q_3, \alpha_3) = (q'_3, \alpha'_3, L), \qquad F = \{q_{F1}, \ldots, q_{Fk}\}$$

Loop:
IF $r_1 = q_1 \wedge r_2 = \alpha_1$ THEN $r_1 = q'_1$; $r_2 = \alpha'_1$;MoveR;
ELSE IF $r_1 = q_2 \wedge r_2 = \alpha_2$ THEN $r_1 = q'_2$; $r_2 = \alpha'_2$;MoveL;
ELSE IF $r_1 = q_3 \wedge r_2 = \alpha_2$ THEN $r_1 = q'_3$; $r_2 = \alpha'_3$;MoveL;
ELSE IF ...
ELSE GOTO Rejected

IF $r_1 = q_{F1} \vee \ldots \vee r_1 = q_{Fk}$ THEN GOTO ACCEPTED;
GOTO Loop

Kosten der Simulation

Gegeben: RM mit Komplexität $T_{\mathcal{R}}(n)$
Gesucht: $T_{\mathcal{T}}(n)$ der äquivalenten DTM

Satz

$$T_{\mathcal{T}}(n) = \mathcal{O}\left(T_{\mathcal{R}}^2(n)\right)$$

Gegeben: DTM mit Komplexität $T_{\mathcal{T}}(n)$
Gesucht: $T_{\mathcal{R}}(n)$ der simulierenden RM

Satz

$$T_{\mathcal{R}}(n) = \mathcal{O}\left(T_{\mathcal{T}}^2(n)\right)$$