Turingmaschinen

Wir betrachten hier folgende Variante von deterministischen Turingmaschinen (kurz: TM):

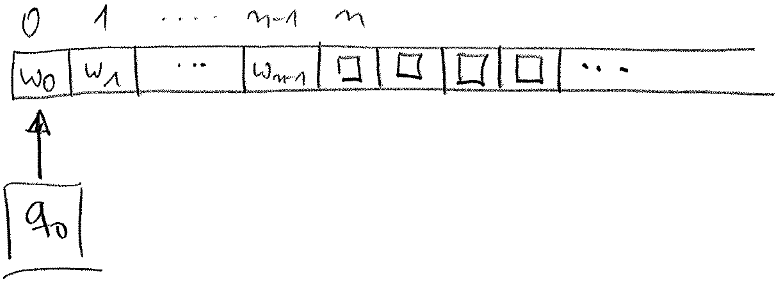
Eine TM $M = (Q, \Sigma, \Gamma, \delta, q_0)$ hat

- Zustandsmenge Q
- Startzustand $q_0 \in Q$
- Eingabealphabet $\Sigma \neq \emptyset$
- Arbeitsalphabet $\Gamma \supseteq \Sigma \cup \{\Box\}$, wobei $\Box \notin \Sigma$ das Blanksymbol (Leerzeichen) ist
- Übergangsfunktion
  $\delta : Q \times \Gamma \to \hat{Q} \times \Gamma \times \{-1, 0, 1\}$
  mit $\hat{Q} := Q \,\dot{\cup}\, \{\text{halt}, \text{akzeptiere}, \text{verwerfe}\}$

M besitzt genau ein Band; dessen Zellen haben die Adressen $0, 1, 2, 3, \ldots$

Die Startkonfiguration von M bei Eingabe
$w = w_0 \cdots w_{n-1} \in \Sigma^*$ mit $n := |w| \geq 1$ und $w_0, \ldots, w_{n-1} \in \Sigma$ ist die Konfiguration, bei der der Kopf von M auf Position 0 steht, M im Zustand q_0 ist, die Bandposition $p \in \{0, \ldots, n-1\}$ mit dem Buchstaben w_p beschriftet ist und jede Bandposition $p \geq n$ mit dem Blanksymbol $\Box$ beschriftet ist.

Skizze:



Startkonfiguration von M bei Eingabe $w = w_0 w_1 \cdots w_{n-1}$.

Datalog zur Simulation von Turingmaschinen:

Satz

Für jede TM $M = (Q, \Sigma, \Gamma, \delta, q_0)$ und jede Zahl $k \in \mathbb{N}_{\geq 1}$ gibt es ein Datalog-Programm $P_{M,k}$ mit $\mathrm{edb}(P_{M,k}) = S_\Sigma$ und $\mathrm{idb}(P_{M,k}) \ni Ans$, wobei Ans ein 0-stelliger Relationsname ist, so dass für die Datalog-Anfrage $Q_{M,k} := (P_{M,k}, Ans)$ und für jedes Wort $w \in \Sigma^*$ der Länge ≥ 2 gilt:

$[\![Q_{M,k}]\!](\mathbb{I}_w) =$ "ja" $\iff$ Bei Eingabe w landet M nach höchstens $|w|^k - 1$ Schritten im Zustand "akzeptiere".

Beweis:

Bei Eingabe eines Worts $w \in \Sigma^*$, der Länge $n := |w| \geq 2$ wird M bis zum Berechnungsschritt $n^k - 1$ nur Bandpositionen zwischen 0 und $n^k - 1$ betreten.

Zahlen aus $\{0, \ldots, n^k - 1\}$ repräsentieren wir in ihrer Darstellung zur Basis n, also durch k-Tupel von Elementen aus $[n] := \{0, \ldots, n-1\} = \mathrm{adom}(I_w$

Für $\bar{x} = (x_{k-1}, \ldots, x_0) \in [n]^k$ ist

$$\mathrm{Zahl}(\bar{x}) := \sum_{i=0}^{k-1} x_i \cdot n^i .$$

Umgekehrt schreiben wir für jede Zahl $z \in \{0, \ldots, n^k - 1\}$

$\mathrm{Tupel}(z)$,

um dasjenige k-Tupel $\bar{x} \in [n]^k$ zu bezeichnen, für das $\mathrm{Zahl}(\bar{x}) = z$ ist.

Insbes. gilt:

Tupel	Zahl
$(0, \ldots, 0, 0)$	0
$(0, \ldots, 0, 1)$	1
$(0, \ldots, 0, n-1)$	$n-1$
$(0, \ldots, 1, 0)$	n
$(0, \ldots, 1, 1)$	$n+1$
$(0, \ldots, 2, 0)$	$2n$
$(n-1, \ldots, n-1, n-1)$	$n^k - 1$

Im Folgenden schreiben wir "Zeitpunkt 0", um uns auf die Startkonfiguration von M bei Eingabe w zu beziehen.

Für $t \geq 1$ bezieht sich "Zeitpunkt t" auf die Konfiguration von M direkt nach dem t-ten Berechnungsschritt.

<u>Idee zur Konstruktion von $P_{M,k}$:</u>

Um Zeitpunkte und Bandpositionen aus $\{0,\dots,n^k-1\}$ zu repräsentieren, nutzen wir k-Tupel $\bar{x} = (x_{k-1},\dots,x_0)$ und $\bar{y} = (y_{k-1},\dots,y_0)$, bei denen jedes x_i und jedes y_j mit einem Element aus $\{0,\dots,n-1\} = adom(A_w)$ interpretiert wird.

Unser Datalog-Programm $P_{M,k}$ nutzt u.a. folgende idb-Prädikate, um die Konfigurationen von M bei Eingabe w zu den Zeitpunkten $0,1,\dots,n^k-1$ zu repräsentieren:

"Ziele" (*)

- Für jedes $q \in \hat{Q} = Q \cup \{halt, akzeptiere, verwerfe\}$ ein k-stelliges Prädikat $Zustand_q$
  <u>Ziel</u>: $Zustand_q(\bar{x})$ $\hat{=}$ zum Zeitpunkt $Zahl(\bar{x})$ ist M in Zustand q
- Ein $2k$-stelliges Prädikat $Kopf$
  <u>Ziel</u>: $Kopf(\bar{x},\bar{y})$ $\hat{=}$ zum Zeitpunkt $Zahl(\bar{x})$ steht der Kopf von M auf Bandposition $Zahl(\bar{y})$
- Für jedes $\gamma \in \Gamma$ ein $2k$-stelliges Prädikat $Band_\gamma$
  <u>Ziel</u>: $Band_\gamma(\bar{x},\bar{y})$ $\hat{=}$ zum Zeitpunkt $Zahl(\bar{x})$ steht auf Bandposition $Zahl(\bar{y})$ der Buchstabe γ.

Um das Datalog-Programm $P_{M,k}$ zu konstruieren, starten wir mit $\boxed{P_{M,k} := \emptyset}$ und fügen nach und nach Regeln hinzu:

Schritt 1: Regeln zum Sicherstellen, dass die Ziele (*) zum Zeitpunkt 0 erfüllt sind.

Beachte: $\bar{x} = (x_{k-1}, \dots, x_0)$ repräsentiert den Zeitpunkt 0

$\iff x_{k-1} = \dots = x_0 = 0$

$\iff$ es gilt: $Min(x_{k-1}), \dots, Min(x_0)$

Wir fügen folgende Regeln zu $P_{M,k}$ hinzu:

- "zum Zeitpunkt 0 ist M in Zustand q_0":

$$\boxed{Zustand_{q_0}(x_{k-1}, \dots, x_0) \leftarrow Min(x_{k-1}), \dots, Min(x_0)}$$

- "zum Zeitpunkt 0 steht der Kopf von M auf Bandposition 0":

$$\boxed{\begin{array}{l} Kopf(x_{k-1}, \dots, x_0, y_{k-1}, \dots, y_0) \leftarrow Min(x_{k-1}), \dots, Min(x_0), \\ \qquad Min(y_{k-1}), \dots, Min(y_0) \end{array}}$$

- "zum Zeitpunkt 0 sind die Bandpositionen $0, \dots, n-1$ mit den Buchstaben $w_0, \dots, w_{n-1}$ beschriftet"

Beachte: Bandposition $p \in \{0, \dots, n-1\}$ wird durch das k-Tupel $(0, \dots, 0, p)$ repräsentiert.

Wir fügen also $\boxed{\text{für jedes } \sigma \in \Sigma}$ die folgende Regel zu $P_{M,k}$ hinzu:

$$\boxed{\begin{array}{l} Band_\sigma(x_{k-1}, \dots, x_0, y_{k-1}, \dots, y_0) \leftarrow Min(x_{k-1}), \dots, Min(x_1), Min(x_0), \\ \qquad Min(y_{k-1}), \dots, Min(y_1), P_\sigma(y_0) \end{array}}$$

- "zum Zeitpunkt 0 sind die Bandpositionen $n, n+1, \dots, n^k-1$ mit dem Blanksymbol $\square$ beschriftet"

<u>Beachte</u>: Bandpositionen $p \in \{n, n+1, \dots, n^k-1\}$ werden durch k-Tupel $\bar{y} = (y_{k-1}, \dots, y_1, y_0)$ repräsentiert, für die es mindestens ein $i \in \{1, \dots, k-1\}$ gibt, so dass $y_i > 0$ ist.

Wir fügen also zu $P_{M,k}$ $\boxed{\text{für jedes } i \in \{1, \dots, k-1\}}$ die folgende Regel hinzu:

$$\boxed{\text{Band}_\square(x_{k-1}, \dots, x_0, y_{k-1}, \dots, y_0) \leftarrow \text{Min}(x_{k-1}), \dots, \text{Min}(x_0), <(x_0, y_i)}$$

- Außerdem fügen wir zu $P_{M,k}$ die beiden Regeln

$<(z, z')$	$\leftarrow$	$\text{Succ}(z, z')$
$<(z, z')$	$\leftarrow$	$\text{Succ}(z, z''), <(z'', z')$

hinzu, die den transitiven Abschluss der Relation Succ (also die durch Succ festgelegte strikte lineare Ordnung) definieren.

<u>Schritt 2</u>: Wir fügen zu $P_{M,k}$ Regeln hinzu, die sicherstellen, dass Folgendes gilt:
Wenn die Ziele (*) zu einem Zeitpunkt t erfüllt sind, dann auch zum Zeitpunkt $t' := t+1$

Zunächst fügen wir zu $P_{M,k}$ Regeln hinzu, um die Relationen "Succ" und "<" auch für Tupel der Länge k zu konstruieren: Füge zu $P_{M,k}$ die Regel

$$\boxed{Succ_1(z,z') \leftarrow Succ(z,z')}$$

und $\boxed{\text{für jedes } \ell \in \{1,\dots,k-1\}}$ die Regeln

$$\boxed{Succ_{\ell+1}(\underline{x_\ell}, x_{\ell-1},\dots,x_0, \underline{x_\ell}, y_{\ell-1},\dots,y_0) \leftarrow Succ_\ell(x_{\ell-1},\dots,x_0, y_{\ell-1},\dots,y_0)}$$

und

$$\boxed{\begin{array}{ll} Succ_{\ell+1}(x_\ell, x_{\ell-1},\dots,x_0, y_\ell, y_{\ell-1},\dots,y_0) \leftarrow & Succ_1(x_\ell, y_\ell), \\ & Max(x_{\ell-1}),\dots, Max(x_0), \\ & Min(y_{\ell-1}),\dots, Min(y_0) \end{array}}$$

hinzu.

Diese Regeln gewährleisten, dass für jedes $\ell \in \{1,\dots,k\}$ die Relation $Succ_\ell$ die "Nachfolger-Relation" auf ℓ-Tupeln ist. Insbes. gilt für $\ell = k$ und $\bar{x}, \bar{y} \in [n]^k$:

$$Succ_k(\bar{x}, \bar{y}) \quad \hat{=} \quad Zahl(\bar{y}) = Zahl(\bar{x}) + 1.$$

Außerdem fügen wir zu $P_{M,k}$ noch Regeln hinzu, die den transitiven Abschluss von $Succ_k$ erzeugen:

$<_k(\bar{x},\bar{y})$	$\leftarrow$	$\mathrm{Succ}_k(\bar{x},\bar{y})$
$<_k(\bar{x},\bar{y})$	$\leftarrow$	$\mathrm{Succ}_k(\bar{x},\bar{z}),\ <_k(\bar{z},\bar{y})$

und Regeln

$\mathrm{Ungleich}_k(\bar{x},\bar{y})$	$\leftarrow$	$<_k(\bar{x},\bar{y})$
$\mathrm{Ungleich}_k(\bar{x},\bar{y})$	$\leftarrow$	$<_k(\bar{y},\bar{x})$

die bewirken, dass für alle $\bar{x},\bar{y} \in [n]^k$ gilt:

$$\mathrm{Ungleich}_k(\bar{x},\bar{y}) \quad \hat{=} \quad \mathrm{Zahl}(\bar{x}) \neq \mathrm{Zahl}(\bar{y}).$$

Wir betrachten nun nacheinander jedes $q \in Q$ und jedes $\gamma \in \Gamma$, setzen $(q',\gamma',b) := \delta(q,\gamma)$ (wobei δ die Überführungsfunktion der TM M ist) und fügen zu $P_{M,k}$ die folgenden Regeln hinzu:

- $$\boxed{\text{Zustand}_{q'}(\bar{x}') \leftarrow \text{Succ}_k(\bar{x},\bar{x}'),\ \text{Zustand}_q(\bar{x}),\ \text{Kopf}(\bar{x},\bar{y}),\ \text{Band}_\gamma(\bar{x},\bar{y})}$$

Zum Zeitpunkt $t+1$ ist M in Zustand q' $\Leftarrow$ Zum Zeitpunkt $t := \text{Zahl}(\bar{x})$ ist M in Zustand q und liest den Buchstaben γ, und $\bar{x}'$ repräsentiert den Zeitpunkt $t+1$

- $$\boxed{\text{Band}_{\gamma'}(\bar{x}',\bar{y}) \leftarrow \text{Succ}_k(\bar{x},\bar{x}'),\ \text{Zustand}_q(\bar{x}),\ \text{Kopf}(\bar{x},\bar{y}),\ \text{Band}_\gamma(\bar{x},\bar{y})}$$

Zum Zeitpunkt $t+1$ steht auf der Bandposition $\text{Zahl}(\bar{y})$ das in Schritt t geschriebene Symbol γ'

... und auf allen anderen Bandpositionen steht zum Zeitpunkt $t+1$ dasselbe Symbol wie zum Zeitpunkt t:

$\boxed{\text{Für jedes } \gamma'' \in \Gamma}$ fügen wir zu $P_{M,k}$ die folgende Regel hinzu:

$$\boxed{\text{Band}_{\gamma''}(\bar{x}',\bar{y}') \leftarrow \text{Succ}_k(\bar{x},\bar{x}'),\ \text{Band}_{\gamma''}(\bar{x},\bar{y}'),\ \text{Kopf}(\bar{x},\bar{y}),\ \text{Ungleich}_k(\bar{y},\bar{y}')}$$

- Falls $b = 0$, so fügen wir zu $P_{M,k}$ die folgende Regel hinzu:

$$\boxed{\mathrm{Kopf}(\overline{x}', \overline{y}) \leftarrow \mathrm{Succ}_k(\overline{x}, \overline{x}'),\ \mathrm{Zustand}_q(\overline{x}),\ \mathrm{Kopf}(\overline{x}, \overline{y}),\ \mathrm{Band}_\gamma(\overline{x}, \overline{y})}$$

- Falls $b = 1$, so füge zu $P_{M,k}$ die Regel

$$\boxed{\mathrm{Kopf}(\overline{x}', \overline{y}') \leftarrow \mathrm{Succ}_k(\overline{x}, \overline{x}'),\ \mathrm{Zustand}_q(\overline{x}),\ \mathrm{Kopf}(\overline{x}, \overline{y}),\ \mathrm{Band}_\gamma(\overline{x}, \overline{y}),\ \mathrm{Succ}_k(\overline{y}, \overline{y}')}$$

hinzu.

- Falls $b = -1$, so füge zu $P_{M,k}$ die folgende Regel hinzu:

$$\boxed{\mathrm{Kopf}(\overline{x}', \overline{y}') \leftarrow \mathrm{Succ}_k(\overline{x}, \overline{x}'),\ \mathrm{Zustand}_q(\overline{x}),\ \mathrm{Kopf}(\overline{x}, \overline{y}),\ \mathrm{Band}_\gamma(\overline{x}, \overline{y}),\ \mathrm{Succ}_k(\overline{y}', \overline{y})}$$

<u>Akzeptanzbedingung</u>:

Füge zu $P_{M,k}$ noch die folgende Regel hinzu:

$$\boxed{\mathrm{Akzeptiere}() \leftarrow \mathrm{Zustand}_{\mathrm{akzeptiere}}(\overline{x}).}$$

Dies beendet die Konstruktion des Programms $P_{M,k}$.

Man kann leicht nachprüfen (Details: Übung), dass die Ziele (*) erreicht werden. D.h. für jedes $n \geq 2$ und jedes $w \in \Sigma^n$ gilt:

(1) f.a. $\bar{x} \in [n]^k$, f.a. $q \in Q \cup \{halt, verwerfe, akzeptiere\}$ ist

$\bar{x} \in [\![P_{M,k}]\!](\mathbb{I}_w)(Zustand_q) \iff$ Bei Eingabe w ist M nach dem $Zahl(\bar{x})$-ten Berechnungsschritt in Zustand q

(2) f.a. $\bar{x} \in [n]^k$, f.a. $\bar{y} \in [n]^k$ ist

$(\bar{x}, \bar{y}) \in [\![P_{M,k}]\!](\mathbb{I}_w)(Kopf) \iff$ Bei Eingabe w zeigt der Kopf von M nach dem $Zahl(\bar{x})$-ten Berechnungsschritt auf Bandposition $Zahl(\bar{y})$.

(3) f.a. $\bar{x} \in [n]^k$, $\bar{y} \in [n]^k$, $\gamma \in \Gamma$ ist

$(\bar{x}, \bar{y}) \in [\![P_{M,k}]\!](\mathbb{I}_w)(Band_\gamma) \iff$

Bei Eingabe w steht nach dem $Zahl(\bar{x})$-ten Berechnungsschritt von M auf Bandposition $Zahl(\bar{y})$ das Symbol γ.

Insbesondere folgt aus (1) mit der zuletzt zu $P_{M,k}$ hinzugefügten Regel, dass

$() \in [\![P_{M,k}]\!](I_w)(\text{Akzeptiere}) \iff$

Bei Eingabe w gelangt M spätestens im (n^k-1)-ten Berechnungsschritt in den Zustand "akzeptiere".

Dies beendet den Beweis des Satzes.

$\square$

Bemerkung:
Man kann sich leicht davon überzeugen, dass bei Eingabe einer geeigneten Repräsentation $\langle M,k \rangle$ von M und k die Datalog-Anfrage $Q_{M,k}$ in Zeit polynomiell in k und der Größe von M konstruiert werden kann.

Außerdem gibt es einen logarithmisch Platz-beschränkten Algorithmus, der bei Eingabe von $w \in \Sigma^*$ mit $|w| \geq 2$ die Datenbank I_w erzeugt.