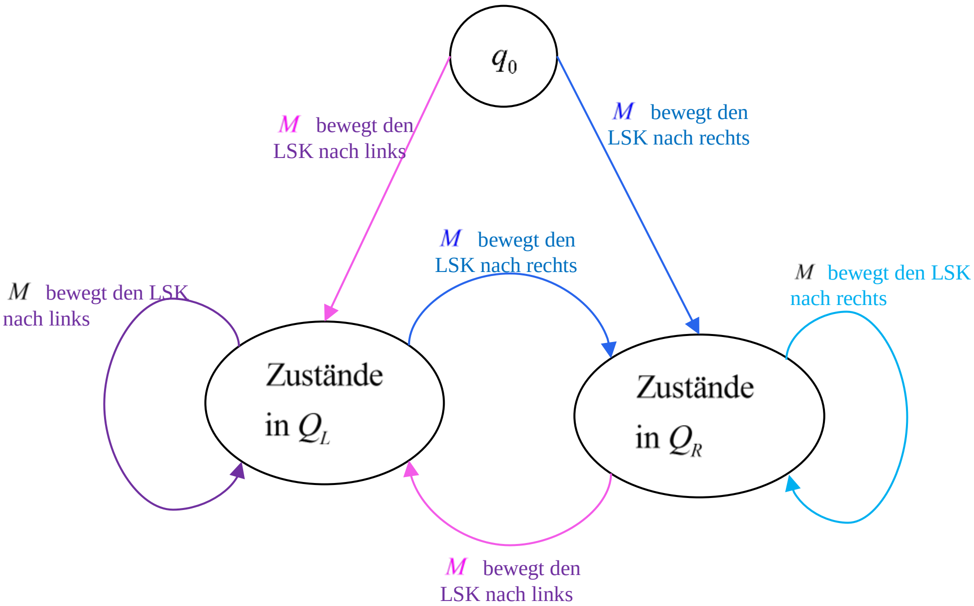
Geteilte Turing-Maschinen – Konstruktion einer äquivalenten geteilten Turing-Maschine

Es sei M_0 eine gegebene Turing-Maschine, deren Startzustand nicht zu den Endzuständen gehört, und die den LSK niemals unbewegt lässt. Dann gibt es eine äquivalente Turing-Maschine M mit dem gleichen Eingabe- und Bandalphabet wie bei M_0, die im folgenden Sinne „geteilt“ ist: Die Zustandsmenge Q von M hat eine disjunkte Zerlegung $Q = Q_L \cup \{q_0\} \cup Q_R$ mit zwei gleich großen „Hälften“ Q_L und Q_R, so dass M wie folgt zwischen den „Hälften“ hin und her wechselt:



Der Sinn solch einer Teilung liegt darin: Nach dem ersten Verarbeitungsschritt befindet sich die Turing-Maschine stets in einem Zustand, der zur Menge Q_L oder zur Menge Q_R gehört, und man kann dann am aktuellen Zustand erkennen, wie der LSK zuletzt bewegt wurde: gehört der aktuelle Zustand zu Q_L, so fand zuletzt eine Linksbewegung statt, gehört er zu Q_R, so fand zuletzt eine Rechtsbewegung statt. Damit kann man z.B. leichter entscheiden, ob der LSK über das linke oder das rechte Ende der Bandaufschrift hinausbewegt wurde.

Die Konstruktion der äquivalenten Maschine M ist z.B. wie folgt möglich:

Wir konstruieren zur gegeben Turing-Maschine $M_0 = (Q_0, \Sigma, \Gamma, \delta_0, q_0, B, F_0)$ wie folgt eine neue Turing-Maschine $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$: Die Alphabete Σ und Γ, das Leerzeichen B und der Startzustand q_0 werden für M unverändert von M_0 übernommen.

Wir setzen $Q_R := Q_0 \setminus \{q_0\}$. Wir nehmen für jedes Element $q \in Q_R$ ein neues Element hinzu, das wir mit q^* bezeichnen (solch ein neues Element darf natürlich nicht mit q_0 übereinstimmen); die Menge aller neuen Elemente bildet dann die Menge Q_L. Damit gibt es eine umkehrbare Abbildung $Q_R \to Q_L, q \mapsto q^*$, und die beiden Mengen enthalten gleich viele Elemente. Die

Zustandsmenge für M sei nun $Q := \underbrace{Q_R \cup \{q_0\}}_{Q_0} \cup Q_L$. Die Menge der Endzustände für M sei $F := F_0 \cup \{q^* | q \in F_0\}$; eine Hälfte der Endzustände liegt also in Q_R, die andere in Q_L (hier wird übrigens die Voraussetzung $q_0 \notin F$ verwendet).

Wir definieren die neue Zustandsübergangsfunktion $\delta : Q \times \Gamma \to Q \times \Gamma \times \{-1,1\}$ wie folgt: bei den Bewegungsrichtungen steht dabei -1 für links und 1 für rechts, und die in den folgenden Fallunterscheidungen verwendeten Farben entsprechen den Pfeilfarben im obigen Schaubild:

Für $(q, X) \in Q_R \times \Gamma$:

$$\delta(q,X) := \begin{cases} (q'^*, Y, -1) & \text{falls} \quad \delta_0(q,X) = (q',Y,-1) \\ (q',Y,1) & \text{falls} \quad \delta_0(q,X) = (q',Y,1) \\ \text{undefiniert} & \text{falls} \quad \delta_0(q,X) \text{ undefiniert ist} \end{cases}$$

Für $(q^*, X) \in Q_L \times \Gamma$ (beachte: zu $q^* \in Q_L$ gehört genau ein Zustand $q \in Q_R \subset Q_0$):

$$\delta(q^*,X) := \begin{cases} (q'^*, Y, -1) & \text{falls} \quad \delta_0(q,X) = (q',Y,-1) \\ (q',Y,1) & \text{falls} \quad \delta_0(q,X) = (q',Y,1) \\ \text{undefiniert} & \text{falls} \quad \delta_0(q,X) \text{ undefiniert ist} \end{cases}$$

und schließlich noch für $X \in \Gamma$:

$$\delta(q_0,X) := \begin{cases} (q'^*, Y, -1) & \text{falls} \quad \delta_0(q_0,X) = (q',Y,-1) \\ (q',Y,1) & \text{falls} \quad \delta_0(q_0,X) = (q',Y,1) \\ \text{undefiniert} & \text{falls} \quad \delta_0(q_0,X) \text{ undefiniert ist} \end{cases}$$

Offensichtlich wechselt dann M - wie in der Skizze oben angedeutet - zwischen Q_L und Q_R hin und her, denn die Definition von δ zeigt, dass eine Rechtsbewegung des LSK immer mit einem Zielzustand aus Q_R und eine Linksbewegung immer mit einem Zielzustand aus Q_L verknüpft ist.

<u>Zeige:</u> M akzeptiert die gleiche Sprache wie M_0 (das ist also die Äquivalenz!)

Zunächst noch zur Sprech- und Schreibweise:

- Mit „Konfiguration" ist im Folgenden die Zusammenfassung all *dieser* Informationen gemeint: aktueller Zustand q der Turing-Maschine, das Teilwort $\xi \in \Gamma^*$, das sich links vom LSK befindet, das Teilwort $\zeta \in \Gamma^*$, dessen erstes Zeichen sich unter dem LSK befindet, und dessen restliche Zeichen rechts vom LSK stehen. Diese Teilworte dürfen auch leer sein. Wir schreiben dafür kurz: $\xi q \zeta$.

- Mit $\xi q\zeta \vdash \xi' q' \zeta'$ ist gemeint, dass die Turing-Maschine in <u>einem einzigen</u> Verarbeitungsschritt von der Konfiguration $\xi q\zeta$ zur Konfiguration $\xi' q' \zeta'$ gelangen kann.
- Mit $\xi q\zeta \overset{*}{\vdash} \xi' q' \zeta'$ ist gemeint, dass die Turing-Maschine in <u>endlich vielen</u> (0,1,2,…,n) unmittelbar aufeinander folgenden Verarbeitungsschritten von der Konfiguration $\xi q\zeta$ zur Konfiguration $\xi' q' \zeta'$ gelangen kann.

Jetzt zum <u>Beweis</u> der Gleichheit der beiden Sprachen; wir müssen $L(M_0) = L(M)$ zeigen.

<u>Zeige</u> $L(M_0) \subseteq L(M)$

<u>Bew</u>.: Es sei x das Eingabewort und $q_0 x \overset{*}{\vdash} \xi q\zeta$ irgendeine Folge von Konfigurationen für M_0 ; wir zeigen durch Induktion über die Anzahl m der Schritte, dass diese Folge in eine für M gültige Folge übersetzt werden kann, wobei einander entsprechende Konfigurationen entweder gleich sind oder sich nur im Zustand unterscheiden ($q \in Q_R$ in der Konfiguration für M_0 entspricht in diesem Fall dem zugehörigen Zustand $q^* \in Q_L$ in der Konfiguration für M).

$m = 0$: Die Startkonfiguration $q_0 x$ ist unverändert auch für M gültig, weil q_0 auch der Startzustand von M ist.

$m-1 \to m$: Es sei $m > 0$. Wir spalten den letzten Schritt ab: $q_0 x \overset{*}{\vdash} \tilde{\xi} p \tilde{\zeta} \vdash \xi q\zeta$. Nach Induktionsannahme gibt es zu $q_0 x \overset{*}{\vdash} \tilde{\xi} p \tilde{\zeta}$ eine für M gültige Folge von Konfigurationen, die entweder mit $\tilde{\xi} p \tilde{\zeta}$ ($p \in Q_0 = Q_R \cup \{q_0\}$) oder auf $\tilde{\xi} p^* \tilde{\zeta}$ ($p^* \in Q_L$) endet. Den letzten Schritt übersetzen wir nun wie folgt:

- Geht der Schritt von $\tilde{\xi} p \tilde{\zeta}$ aus und ist bei M_0 der letzte Übergang $p \to q$ mit einer Rechtsbewegung des LSK verbunden, so ist $\tilde{\xi} p \tilde{\zeta} \to \xi q \zeta$ der letzte Schritt für M .
- Geht der Schritt von $\tilde{\xi} p \tilde{\zeta}$ aus und ist bei M_0 der letzte Übergang $p \to q$ mit einer Linksbewegung des LSK verbunden, so ist $\tilde{\xi} p \tilde{\zeta} \to \xi q^* \zeta$ der letzte Schritt für M .
- Geht der Schritt von $\tilde{\xi} p^* \tilde{\zeta}$ aus und ist bei M_0 der letzte Übergang $p \to q$ mit einer Rechtsbewegung des LSK verbunden, so ist $\tilde{\xi} p^* \tilde{\zeta} \to \xi q \zeta$ der letzte Schritt für M .
- Geht der Schritt von $\tilde{\xi} p^* \tilde{\zeta}$ aus und ist bei M_0 der letzte Übergang $p \to q$ mit einer Linksbewegung des LSK verbunden, so ist $\tilde{\xi} p^* \tilde{\zeta} \to \xi q^* \zeta$ der letzte Schritt für M .

Endet die für M_0 gegebene Konfigurationsfolge $q_0 x \overset{*}{\vdash} \xi q\zeta$ mit einem Zustand $q \in F_0$ (d.h. M_0 akzeptiert das Eingabewort x), so endet auch die entsprechende Folge für M mit einem akzeptierenden Zustand, weil wir $F := F_0 \cup \{ q^* | q \in F_0 \}$ gesetzt haben. Dies zeigt $L(M_0) \subseteq L(M)$.

<u>Zeige</u>: $L(M) \subseteq L(M_0)$

<u>Bew</u>.: analog, indem man in ähnlicher Weise eine für M gültige Konfigurationsfolge $q_0 x \overset{*}{\vdash} \xi q\zeta$ in eine für M_0 gültige zurückübersetzt.