Zwei einfache Programmiertechniken

1. **Merken im Zustand**
    - **Nutzen Zustände als endlichen Speicher.**

2. **Markieren von Symbolen**
    - **Nutzen Bandalphabet zur Markierung von Positionen des Bands.**

DTM M_3 für Palindrome über {0,1}

δ	0	1	$\sqcup$	$\triangleright$
q_0	$(q_1,\sqcup,R)$	$(q_2,\sqcup,R)$	$(q_6,\sqcup,R)$	$(q_0,\triangleright,R)$
q_1	$(q_1,0,R)$	$(q_1,1,R)$	$(q_3,\sqcup,L)$	$(q_7,\triangleright,R)$
q_2	$(q_2,0,R)$	$(q_2,1,R)$	$(q_4,\sqcup,L)$	$(q_7,\triangleright,R)$
q_3	$(q_5,\sqcup,L)$	$(q_7,1,L)$	$(q_6,\sqcup,L)$	$(q_7,\triangleright,R)$
q_4	$(q_7,0,L)$	$(q_5,\sqcup,L)$	$(q_6,\sqcup,L)$	$(q_7,\triangleright,R)$
q_5	$(q_5,0,L)$	$(q_5,1,L)$	$(q_0,\sqcup,R)$	$(q_7,\triangleright,R)$

$q_6 = q_{accept}, q_7 = q_{reject}$

Im Zustand merken - Beispiel

$L := \{w \in \Sigma^* \mid w = w_1 \ldots w_n, \exists i, 2 \leq i \leq n: w_i = w_1\}$

1)	$\delta(q_0, \triangleright)$	=	$(q_0, \triangleright, R)$	
2)	$\delta(q_0, \sqcup)$	=	$(q_2, \sqcup, L)$	
3)	$\delta(q_0, a)$	=	$([q_0,a], a, R)$	$\forall a \in \Sigma$
4)	$\delta([q_0,a], a)$	=	(q_1, a, L)	$\forall a \in \Sigma$
5)	$\delta([q_0,a], b)$	=	$([q_0,a], b, R)$	$\forall a,b \in \Sigma, a \neq b$
6)	$\delta([q_0,a], \sqcup)$	=	$(q_2, \sqcup, L)$	
7)	$\delta([q_0,a], \triangleright)$	=	$(q_2, \triangleright, R)$	

$q_{accept} = q_1$, $q_{reject} = q_2$

DTM M_3' für Palindrome über {0,1}

δ	**0**	**1**	$\sqcup$	$\triangleright$
q_0	$([q_0,0],\sqcup,R)$	$([q_0,1],\sqcup,R)$	$(q_6,\sqcup,R)$	$(q_0,\triangleright,R)$
$[q_0,0]$	$([q_0,0],0,R)$	$([q_0,0],1,R)$	$([q_1,0],\sqcup,L)$	$(q_7,\triangleright,R)$
$[q_0,1]$	$([q_0,1],0,R)$	$([q_0,1],1,R)$	$([q_1,1],\sqcup,L)$	$(q_7,\triangleright,R)$
$[q_1,0]$	$(q_5,\sqcup,L)$	$(q_7,1,L)$	$(q_6,\sqcup,L)$	$(q_7,\triangleright,R)$
$[q_1,1]$	$(q_7,0,L)$	$(q_5,\sqcup,L)$	$(q_6,\sqcup,L)$	$(q_7,\triangleright,R)$
q_5	$(q_5,0,L)$	$(q_5,1,L)$	$(q_0,\sqcup,R)$	$(q_7,\triangleright,R)$

$q_6 = q_{accept}, q_7 = q_{reject}$

Markieren von Symbolen

Element Distinctness:

$L:=\{\#w_1\#\ldots\ \#w_n \mid w_i \in \{0,1\}^*, w_i \neq w_j \text{ für alle } i \neq j\}$

Beispiele

- $\#001\#111\#10\#11 \in L$
- $\#011\#11\#01\#10\#01 \notin L$

DTM für Element-Distinctness

1. **Falls das erste Eingabesymbol nicht # ist, lehne ab, sonst ersetze # durch #´ (markiere #).**

2. **Finde das nächste # rechts vom ersten # und ersetze es durch #´. Wird kein weiteres # gefunden, akzeptiere.**

3. **Teste, ob die beiden Folgen w_i, w_j rechts der Symbole #´ gleich sind. Wenn ja, lehne ab.**

4. **Verschiebe Markierungen für den Vergleich des nächsten Paares von Folgen. Falls dieses Paar nicht mehr existiert, akzeptiere. Sonst gehe zu 3.**

Element-Distinctness - Markierte Positionen

Markierte # bei Vergleich w_1 und w_2.

Markierte # bei Vergleich w_1 und w_3.

⋮

⋮

Markierte # bei Vergleich w_2 und w_5.

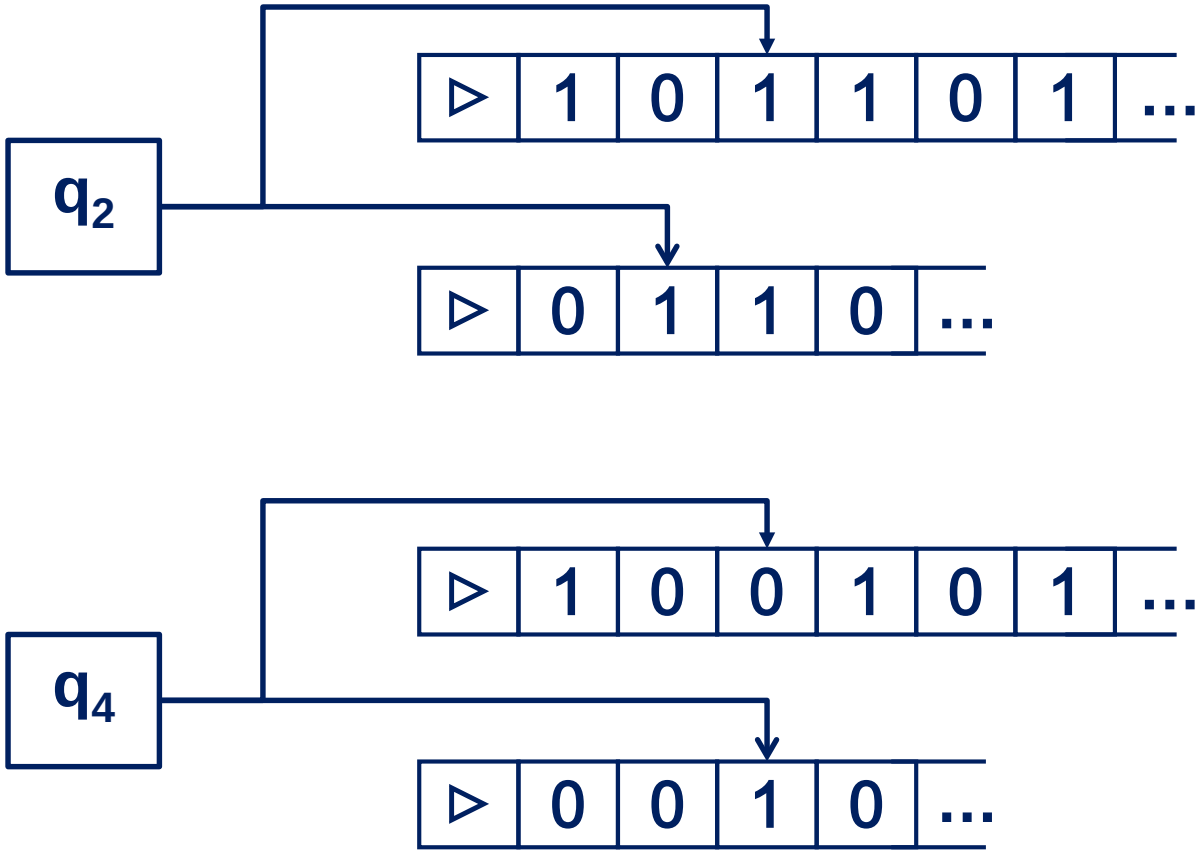
Mehrband-Turingmaschinen - Definition

Definition 2.7 **Eine deterministische k-Band Turingmaschine (k-DTM) ist ein 4-Tupel (Q,Σ,Γ,δ). Für Q,Σ,Γ gelten die Anforderungen einer 1-Band DTM. Für die Übergangsfunktion δ gilt**

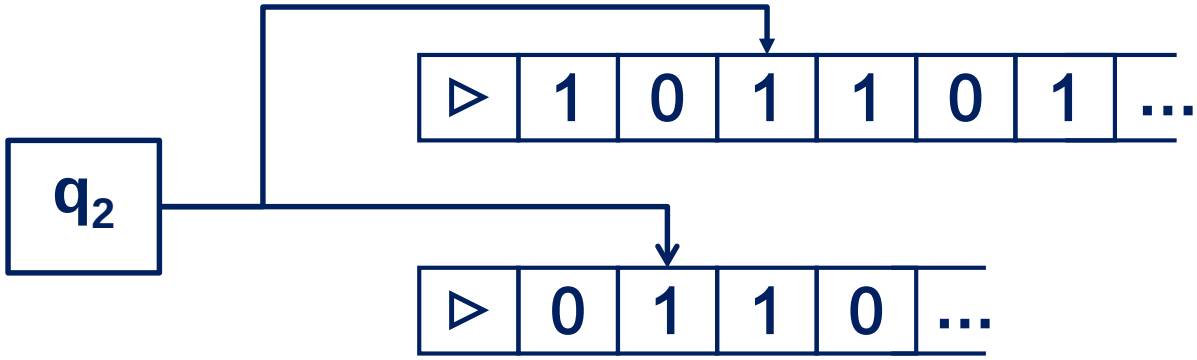
$$\delta: Q\setminus\{q_{accept},q_{reject}\}\times\Gamma^k \rightarrow Q\times\Gamma^k\times \{L,R,S\}^k.$$

Dabei steht S für stationär, d.h., der Lesekopf wird nicht bewegt. Für das Startsymbol gelten dieselben Einschränkungen wie bei 1-Band DTMs.

Rechenschritt bei $\delta(q_2,1,1)=(q_4,0,0,S,R)$



Konfiguration einer 2-DTM



Konfiguration besteht aus

- Zustand,
- Inhalt der Bänder
- Position der Leseköpfe.

Konfiguration einer k-DTM

Übliche Kodierung einer Konfiguration:

$$(q,(v_1,w_1),\ldots,(v_k,w_k)) \in Q\times((\Gamma^*)^2)^k$$

wobei v_i,w_i die Zeichenketten vor und nach der Kopfposition auf Band i sind (der Kopf steht auf dem ersten Zeichen von w_i).

Startkonfiguration bei Eingabe $w \in \Sigma^*$ besteht aus

- Startzustand q_0,
- erstes Band enthält $\triangleright w$,
- alle anderen Bänder $\triangleright\sqcup\ldots$ (d.h. $v_i = \varepsilon$ für alle i),
- alle Leseköpfe stehen auf $\triangleright$ (d.h. $w_i = \triangleright$ für alle i).

Akzeptierende/ablehnende Konfiguration

- Zustand ist q_{accept}, q_{reject}.

Berechnung, Akzeptieren, Entscheiden, …

wie bei 1-Band DTMs.

Beispiel: Addition zweier Binärzahlen

Eingabe: $w_1\#w_2$ für zwei Binärzahlen w_1 und w_2, wobei die erste Binärzahl jeweils die niedrigste Binärstelle angibt **(Beispiel:** 01#001 entspricht den Zahlen 2 und 4**)**
Ausgabe: das Ergebnis von w_1+w_2 auf irgendeinem Band

Strategie (Übung):
- Wir verwenden 3-Band Turingmaschine
- w_2 wird zunächst auf Band 2 verschoben
- Dann werden w_1 und w_2 auf Band 3 bitweise zu w_1+w_2 zusammenaddiert. Zum Schluss gehen wir in q_{accept}.

Mächtigkeit von k-DTMs

Satz 2.8 **Wird die Sprache L von einer k-DTM entschieden (akzeptiert), so gibt es auch eine 1-Band Turingmaschine, die L entscheidet (akzeptiert).**

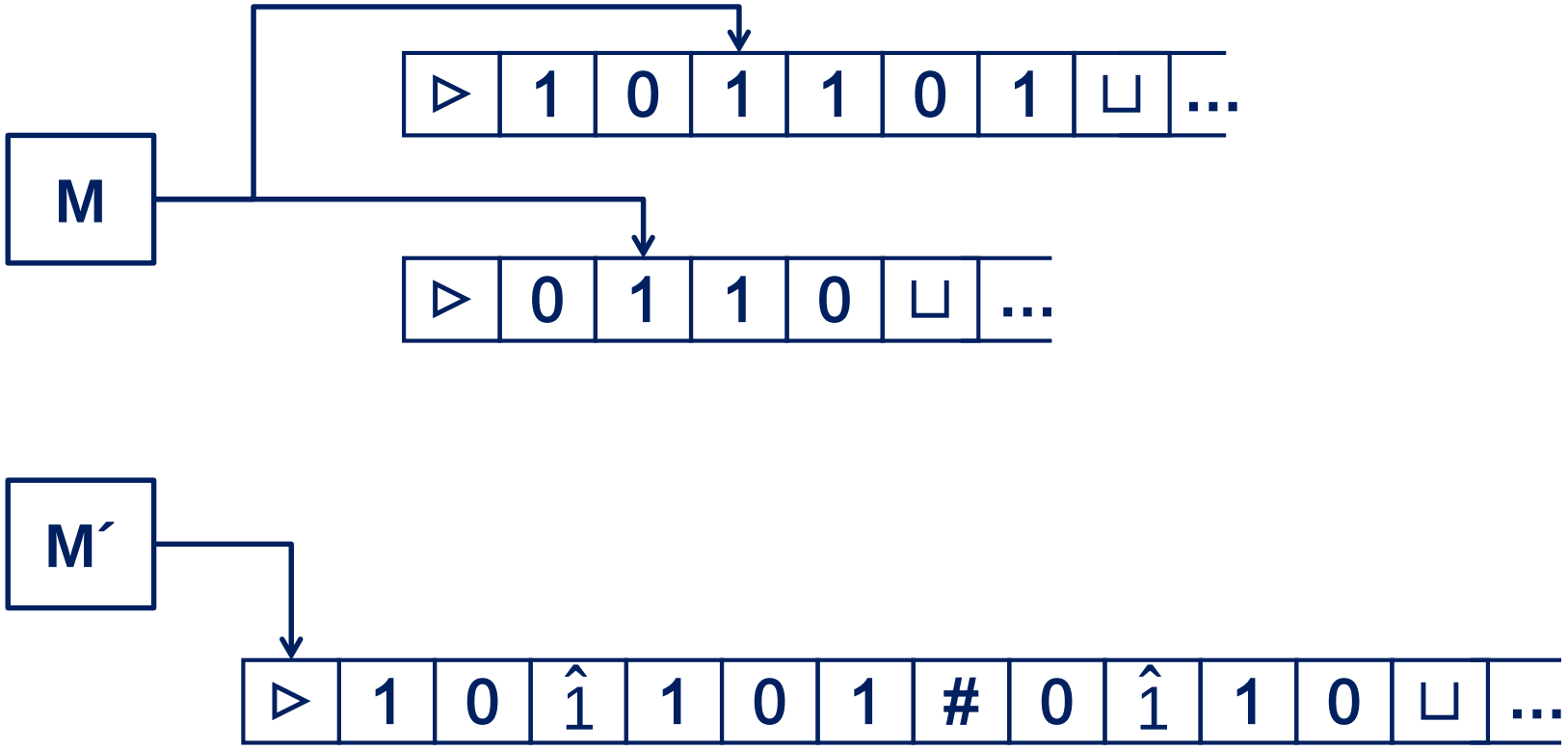
Beweis beruht auf Simulation einer k-DTM durch DTM.

Simulation von k-DTMs durch DTMs

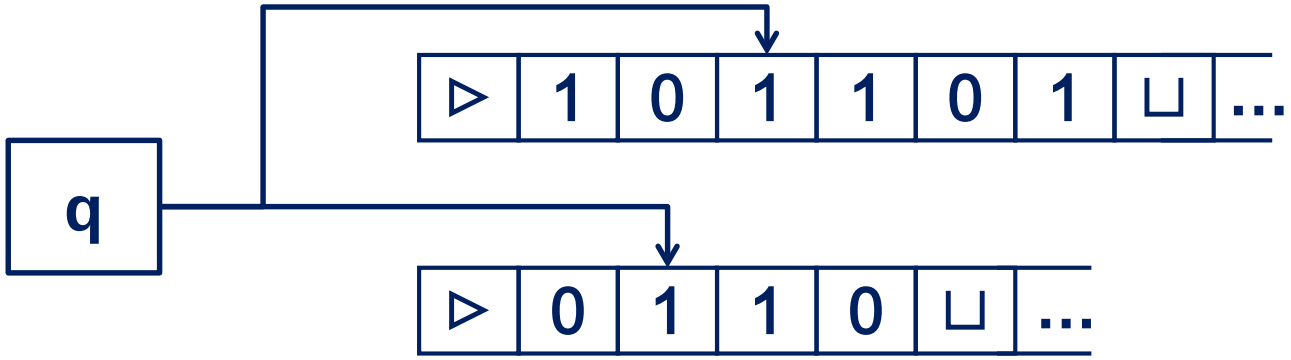
Simulation einer k-DTM durch DTM beruht auf zwei Techniken:

- **Kodierung der Konfigurationen einer k-DTM M auf 1-Band DTM M´.**
- **Berechnung der Kodierung der Nachfolgekonfiguration aus Kodierung der Konfiguration.**

Speicherung von 2 Bändern durch 1 Band



Speicherung der Bandinhalte im Kopf



Details: Tafel

Abstrahierung von TM zum Algorithmus

- **Algorithmus verwendet v Variablen:**

  verwende dafür v zusätzliche Bänder in der TM

- **Grundlegende Rechenoperationen:**

  einfach in TM umsetzbar

- **If <Condition> then <Block1> else <Block2>:**

	$(a_1,\ldots,a_k)$	$(b_1,\ldots,b_k)$	…
q-Block für <Condition>	Nach Test von <Condition> Sprung zu <Block1> oder <Block2>		
q-Block für <Block1>			
q-Block für <Block2>			

Abstrahierung von TM zum Algorithmus

- **for i=1 to n do <Block>:**

	$(a_1,\ldots,a_k)$	$(b_1,\ldots,b_k)$	…
q-Block für i=1			
q-Block für <Block>			
q-Block für i=i+1, i>n?	i≤n: springe zurück zum q-Block für <Block>		

- **while <Condition> do <Block>:**

	$(a_1,\ldots,a_k)$	$(b_1,\ldots,b_k)$	…
q-Block für <Condition>	<Condition> wahr: springe zu <Block>		
q-Block für <Block>	am Ende zurück zu <Condition>		

Churchsche These (1936)

Die im intuitiven Sinne berechenbaren Funktionen und Sprachen sind genau die, die durch Turingmaschinen berechenbar sind.

Warum sind Turingmaschinen tatsächlich ein geeignetes Modell?

- **Menschliche Wahrnehmung ist endlich (endliche Anzahl an Rezeptoren)**
- **Jeder realisierbare Rechner muss endlicher Natur sein und den physikalischen Gesetzen folgen**

Abschlusseigenschaften

$\overline{L}$: Komplementsprache zu L, d.h. $\overline{L} = \Sigma^* \setminus L$

Satz 2.10 Seien L_1, L_2 entscheidbare Sprachen, dann gilt:

1. $\overline{L_1}$ ist entscheidbar.
2. $L_1 \cap L_2$ ist entscheidbar.
3. $L_1 \cup L_2$ ist entscheidbar.

Satz 2.11 Seien L_1, L_2 rekursiv aufzählbare Sprachen, dann gilt:

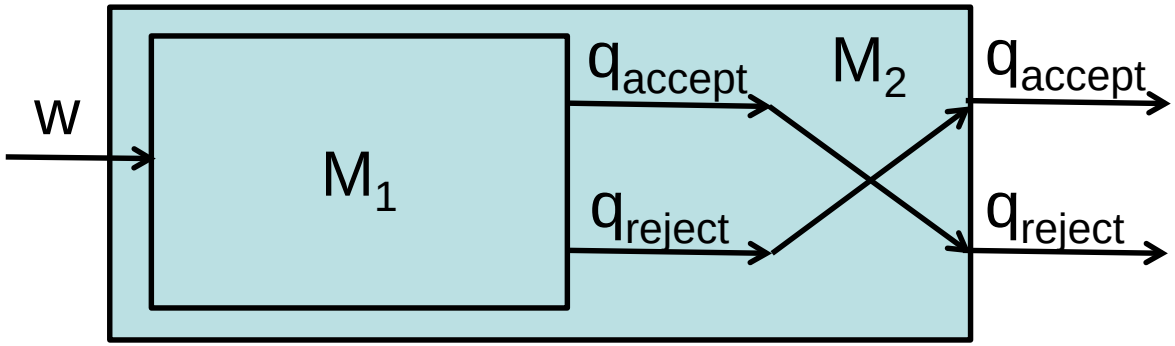
1. $L_1 \cap L_2$ ist rekursiv aufzählbar.
2. $L_1 \cup L_2$ ist rekursiv aufzählbar.

Abschlusseigenschaften

Satz 2.10 Seien L_1, L_2 entscheidbare Sprachen, dann gilt:

1. $\overline{L_1}$ ist entscheidbar.
2. $L_1 \cap L_2$ ist entscheidbar.
3. $L_1 \cup L_2$ ist entscheidbar.

Beweis von 1.: Sei M_1 die TM zu L_1. TM M_2 zu $\overline{L_1}$:



Schnitt/Vereinigung entscheidbarer Sprachen

Sei M_1 die DTM zu L_1 und M_2 die DTM zu L_2.
M bei Eingabe w:

1. Kopiere die Eingabe auf Band 2.
2. Arbeite parallel wie M_1 auf Band 1 und wie M_2 auf Band 2.

Parallel arbeiten:

$\delta([q_1,q_2],a_1,a_2)=([p_1,p_2],b_1,b_2,D_1,D_2)$, falls

$\delta(q_1,a_1)=(p_1,b_1,D_1)$ und $\delta(q_2,a_2)=(p_2,b_2,D_2)$

M für $L_1 \cap L_2$:
- Akzeptiere, sobald M_1 und M_2 akzeptieren
- Lehne Eingabe ab, wenn M_1 oder M_2 Eingabe ablehnen

Schnitt/Vereinigung entscheidbarer Sprachen

M für $L_1 \cap L_2$:

- **Akzeptiere, sobald M_1 und M_2 akzeptieren**
- **Lehne Eingabe ab, wenn M_1 oder M_2 Eingabe ablehnen**

$q_1 \notin \{q^1_{accept}, q^1_{reject}\}$ und $q_2 \notin \{q^2_{accept}, q^2_{reject}\}$:

$\delta([q_1,q_2],a_1,a_2) = ([p_1,p_2],b_1,b_2,D_1,D_2)$, falls
$\delta(q_1,a_1) = (p_1,b_1,D_1)$ und $\delta(q_2,a_2)=(p_2,b_2,D_2)$

$q_2 \notin \{q^2_{accept}, q^2_{reject}\}$:

$\delta([q^1_{accept},q_2],a_1,a_2) = ([q^1_{accept},p_2],a_1,b_2,S,D_2)$, falls $\delta(q_2,a_2)=(p_2,b_2,D_2)$

Genauso für $[q_1,q^2_{accept}]$. Wir setzen $[q^1_{accept},q^2_{accept}]$ als q_{accept} und $[q_1,q_2]$ als q_{reject} für alle $[q_1,q_2]$ mit $q_1=q^1_{reject}$ oder $q_2=q^2_{reject}$.

Durchschnitt entscheidbarer Sprachen

Sei M_1 die DTM zu L_1 und M_2 die DTM zu L_2.

Alternativstrategie für M für $L_1 \cap L_2$ bei Eingabe w:

1. Simuliere M_1 gestartet auf w. Falls M_1 w ablehnt, dann lehne w ab.

2. Falls M_1 w akzeptiert, dann simuliere M_2 gestartet mit w. Falls M_2 w akzeptiert, dann akzeptiere w, sonst lehne w ab.

Zu zeigen: M entscheidet $L_1 \cap L_2$.

- $w \in L_1 \cap L_2$: M_1 und M_2 akzeptieren w $\Rightarrow$ M akzeptiert w.
- $w \notin L_1 \cap L_2$: M_1 oder M_2 verwerfen w $\Rightarrow$ M verwirft w.

Durchschnitt entscheidbarer Sprachen

Anschaulich:

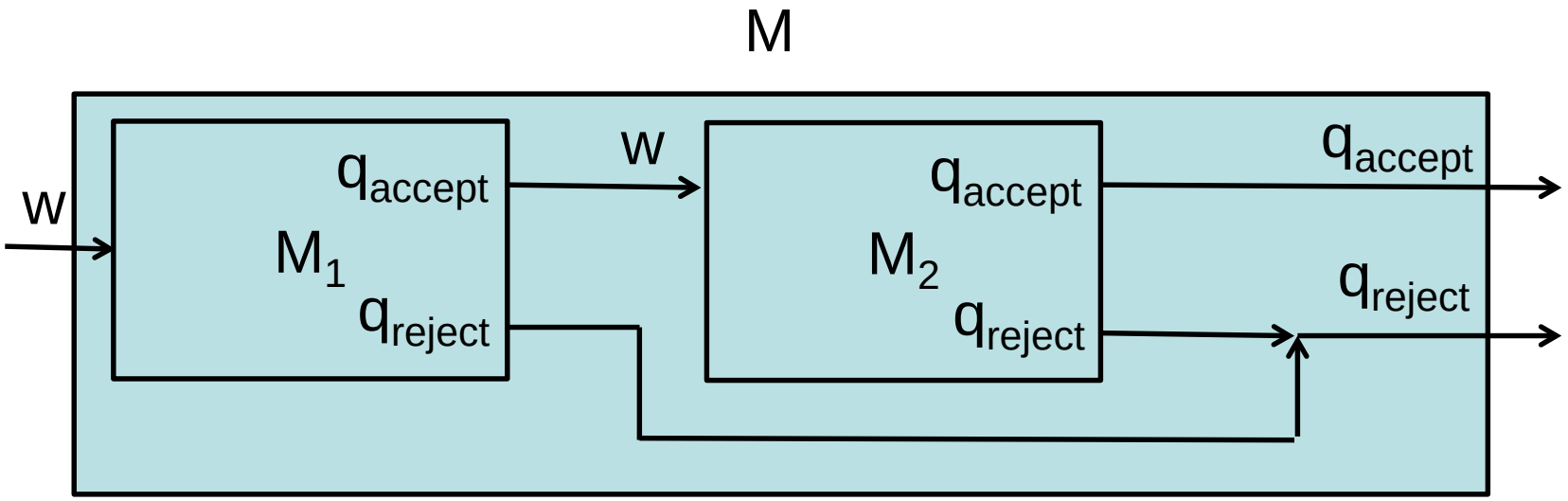


Schaubild in Beweisen zur Illustration hilfreich, aber trotzdem ist eine formale (algorithmische) Beschreibung zu leisten. Darüber hinaus ist formale Argumentation notwendig, dass DTM M tatsächlich die gewünschte Sprache entscheidet.

Schnitt/Vereinigung entscheidbarer Sprachen

Sei M_1 die DTM zu L_1 und M_2 die DTM zu L_2.
M bei Eingabe w:

1. **Kopiere die Eingabe auf Band 2.**
2. **Arbeite parallel wie M_1 auf Band 1 und wie M_2 auf Band 2.**

Parallel arbeiten:

$\delta([q_1,q_2],a_1,a_2)=([p_1,p_2],b_1,b_2,D_1,D_2)$, **falls**

$\delta(q_1,a_1)=(p_1,b_1,D_1)$ **und** $\delta(q_2,a_2)=(p_2,b_2,D_2)$

M für $L_1 \cup L_2$:
- **Akzeptiere, sobald M_1 oder M_2 akzeptieren**
- **Lehne Eingabe ab, wenn M_1 und M_2 Eingabe ablehnen**

Abschlusseigenschaften

Satz 2.11 Seien L_1, L_2 rekursiv aufzählbare Sprachen, dann gilt:

1. $L_1 \cap L_2$ ist rekursiv aufzählbar.
2. $L_1 \cup L_2$ ist rekursiv aufzählbar.

Beweis:
1. und 2. wie mit der Parallelmethode auf Folie 22

- Warum kann sie sequentielle Methode auf Folie 24 nicht angewandt werden?
- Warum ist $\overline{L_1}$ nicht notwendigerweise rekursiv aufzählbar?

Abschlusseigenschaften

Satz 2.12 L ist genau dann entscheidbar, wenn L und $\overline{L}$ rekursiv aufzählbar sind.

Beweis:

- L und $\overline{L}$ rekursiv aufzählbar: es gibt DTMs M und $\overline{M}$, die L und $\overline{L}$ akzeptieren
- M´, die L entscheidet: lass M und $\overline{M}$ parallel auf Eingabe w laufen
- akzeptiere, falls M w akzeptiert
- verwerfe, falls $\overline{M}$ w akzeptiert
- $w \in L$: M akzeptiert w $\Rightarrow$ M´ akzeptiert w
- $w \notin L$: $\overline{M}$ akzeptiert w $\Rightarrow$ M´ verwirft w