

Theoretische Grundlagen der Informatik

Übung

3. Übungstermin · 21. November
Torsten Ueckerdt

INSTITUT FÜR THEORETISCHE INFORMATIK · LEHRSTUHL ALGORITHMIK

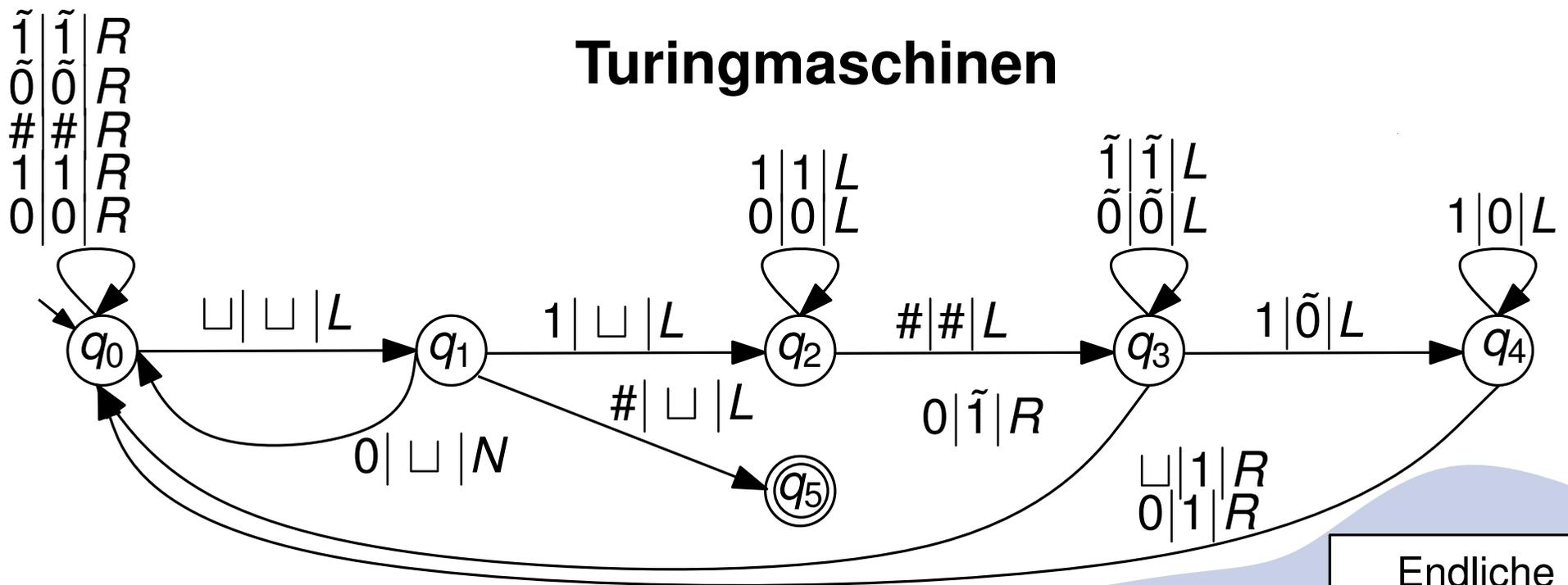
Gliederung

Inhalt

- **Turingmaschinen**
 - Erweiterungen
- **Entscheidbarkeit**
 - Beispiele
 - Werkzeugkasten

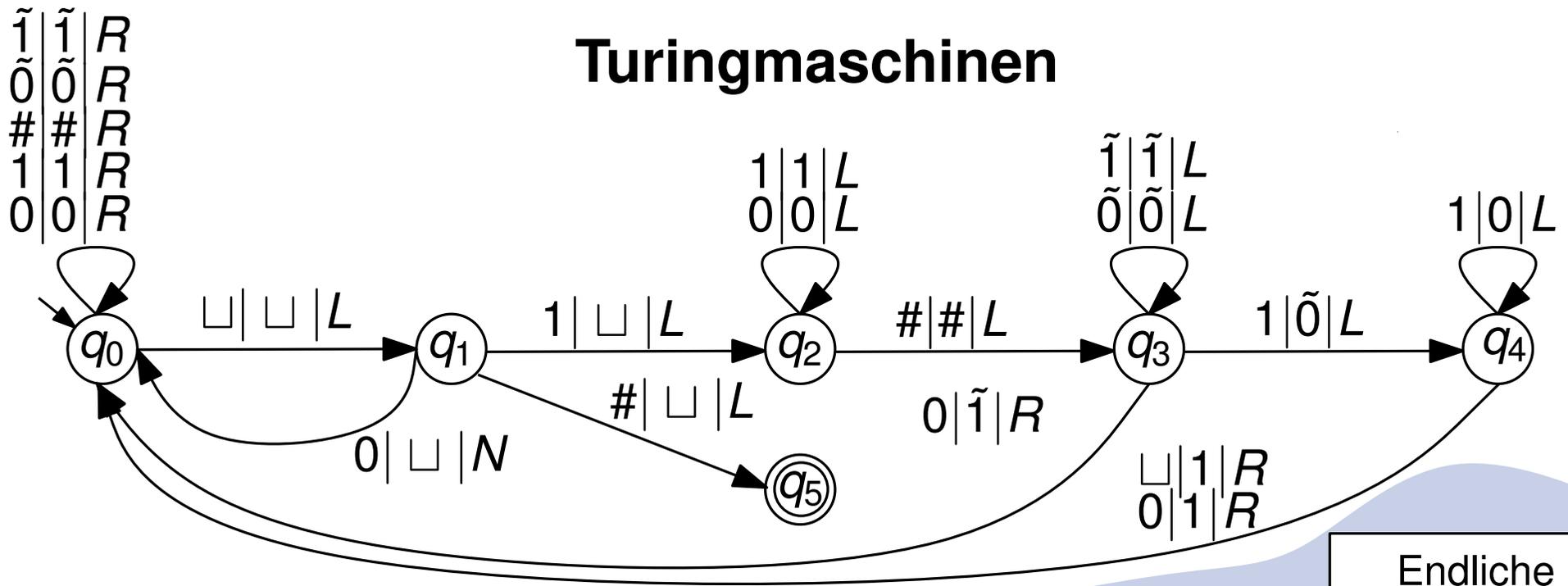
Turingmaschinen

Turingmaschinen



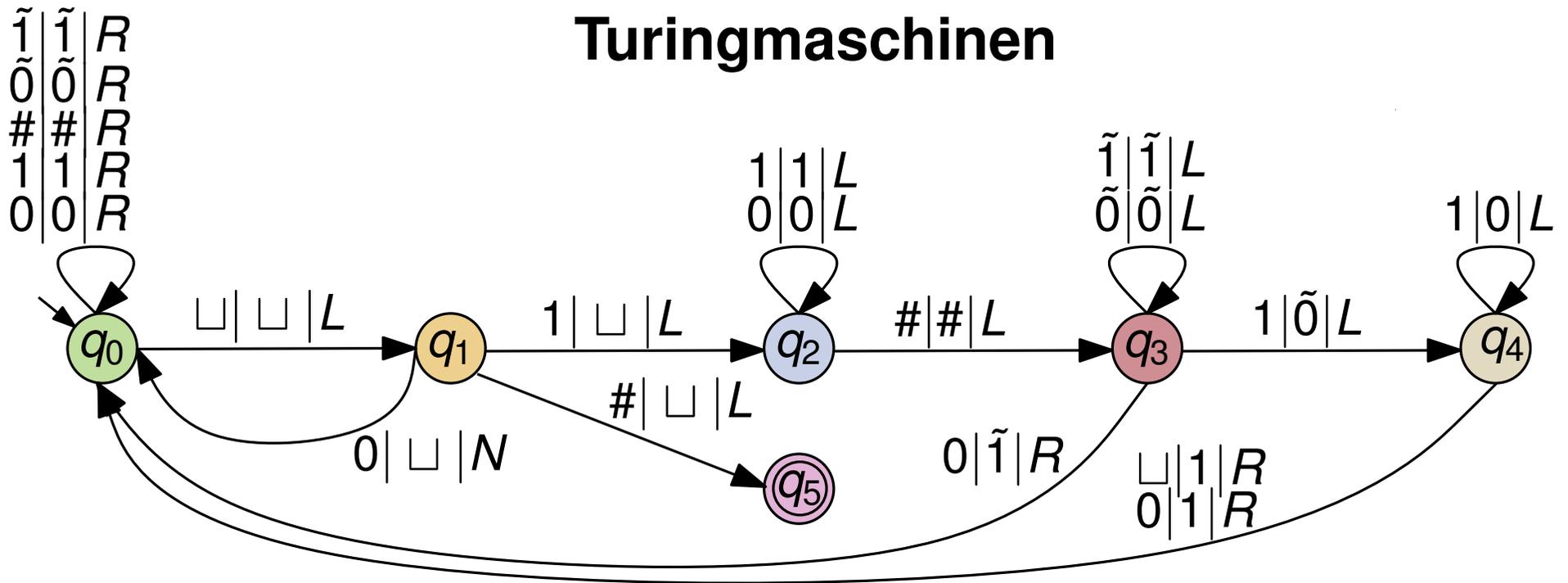
- Q , endliche Zustandsmenge,
- Σ , endliches Eingabealphabet,
- \sqcup , Blanksymbol mit $\sqcup \notin \Sigma$,
- Γ , endliches Bandalphabet mit $\Sigma \cup \{\sqcup\} \subseteq \Gamma$,
- $s \in Q$, Startzustand,
- $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, N\}$, Übergangsfunktion,
- $F \subseteq Q$, Menge von Endzuständen

Turingmaschinen



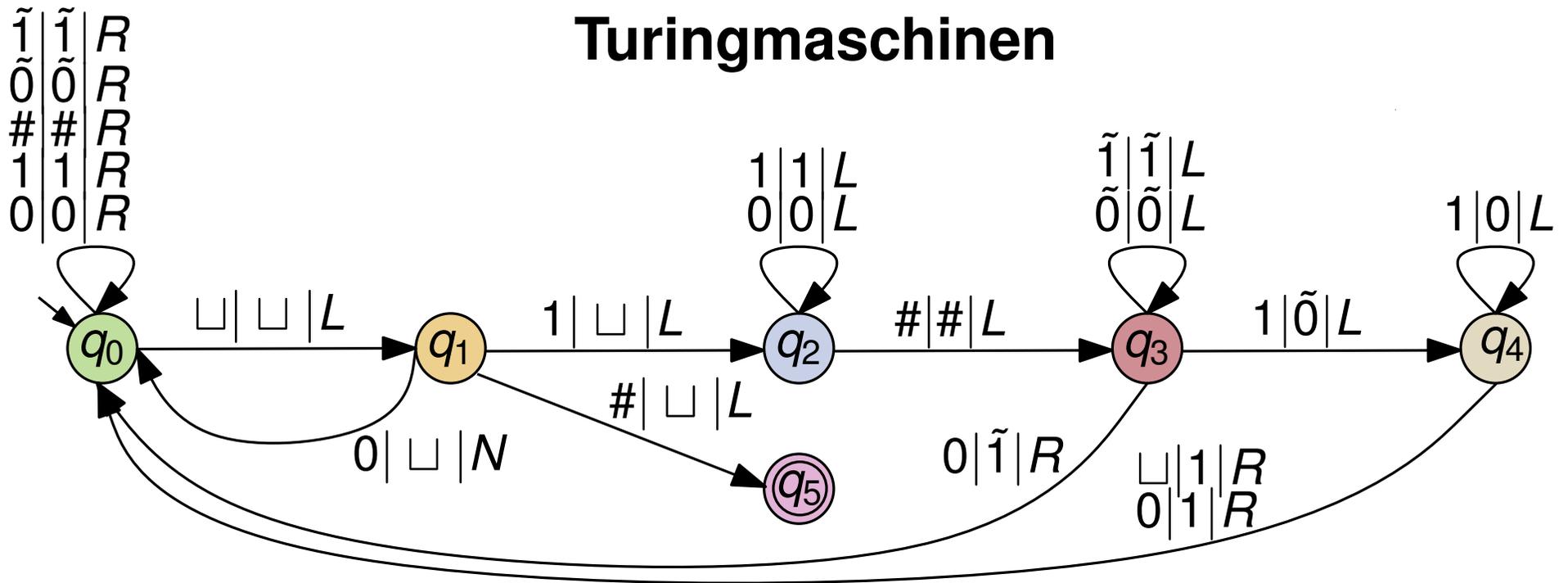
- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$,
- $\Sigma = \{0, 1, \#\}$,
- $\Gamma = \Sigma \cup \{\sqcup, \tilde{0}, \tilde{1}\}$,
- Startzustand q_0 ,
- Übergangsfunktion $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, N\}$
- Endzustand $q_5 \subseteq Q$

Turingmaschinen



$\square(q_0)111\#111\square$

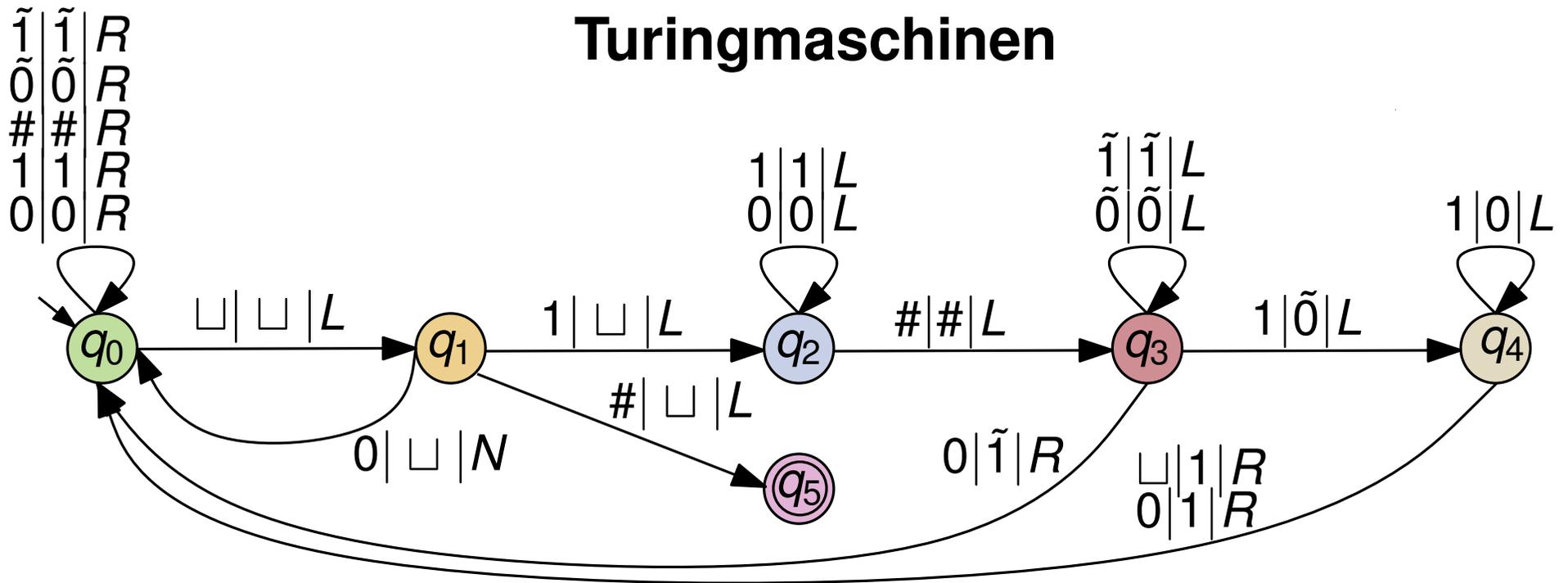
Turingmaschinen



$\square(q_0)111\#111\square$

$\square 1(q_0)11\#111\square$

Turingmaschinen

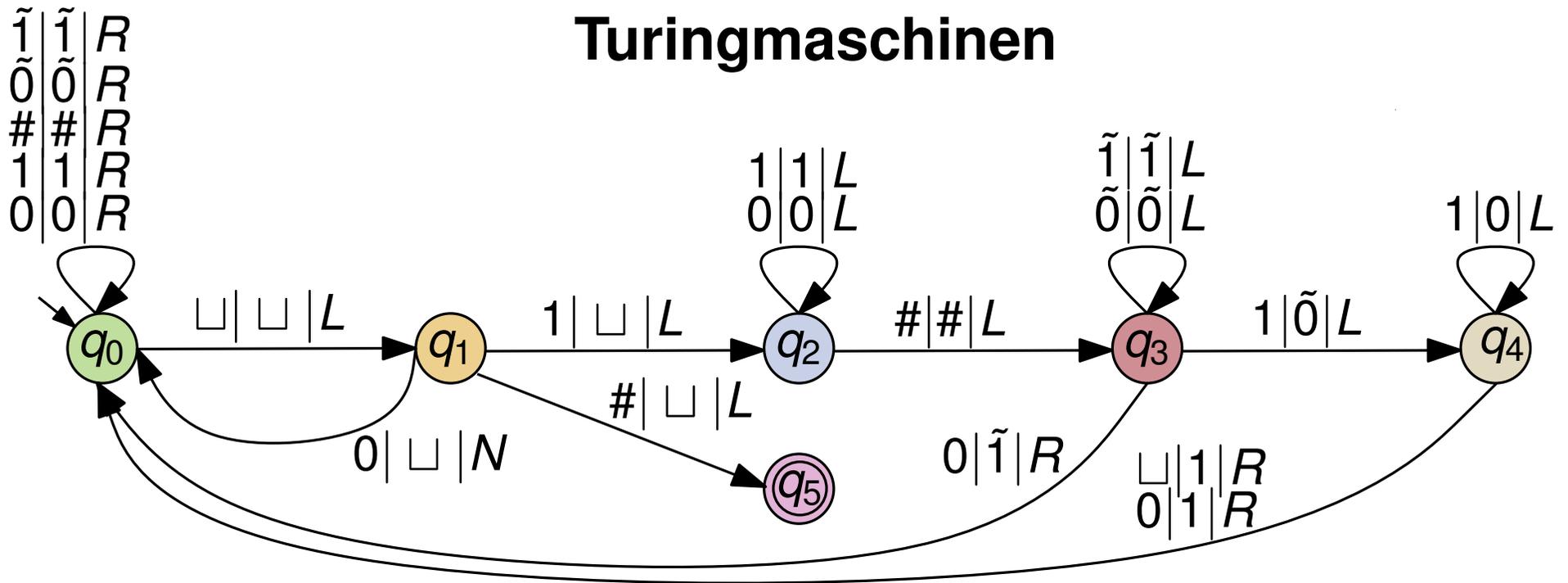


$\square(q_0)111\#111\square$

$\square 1(q_0)11\#111\square$

$\square 11(q_0)1\#111\square$

Turingmaschinen



$\square(q_0)111\#111\square$

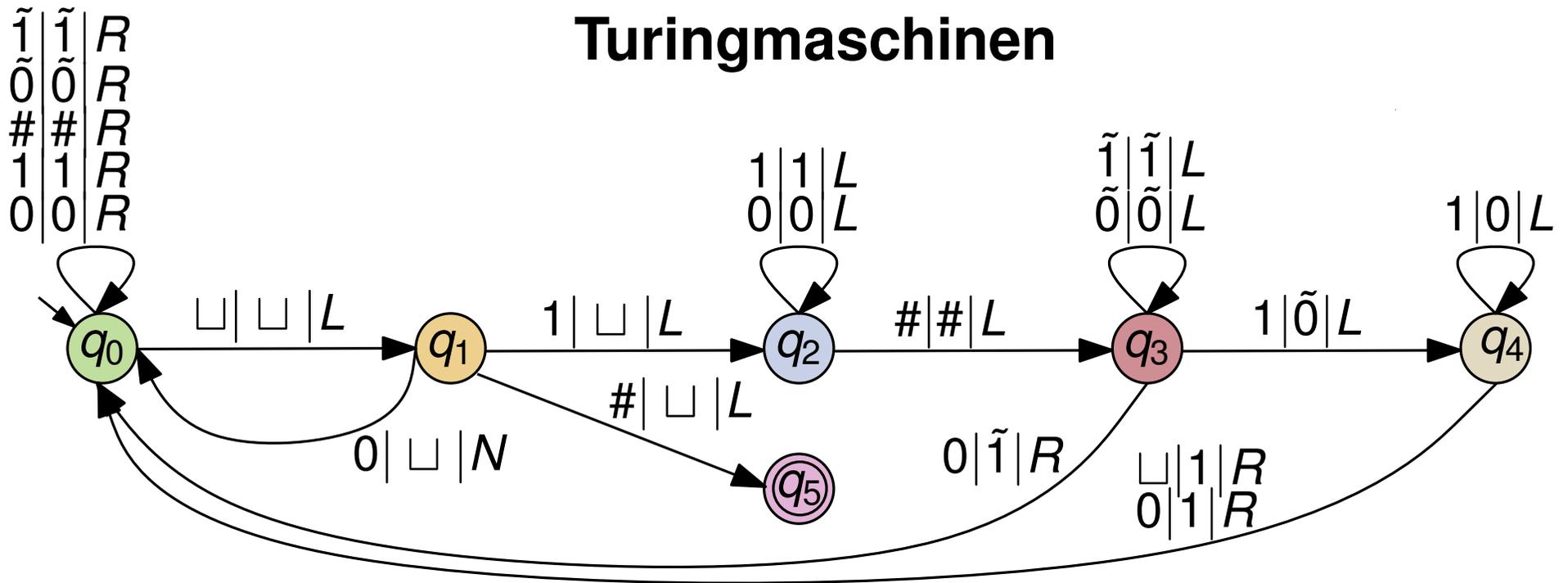
$\square 1(q_0)11\#111\square$

$\square 11(q_0)1\#111\square$

...

$\square 111\#111(q_0)\square$

Turingmaschinen



$\square(q_0)111\#111\square$

$\square 1(q_0)11\#111\square$

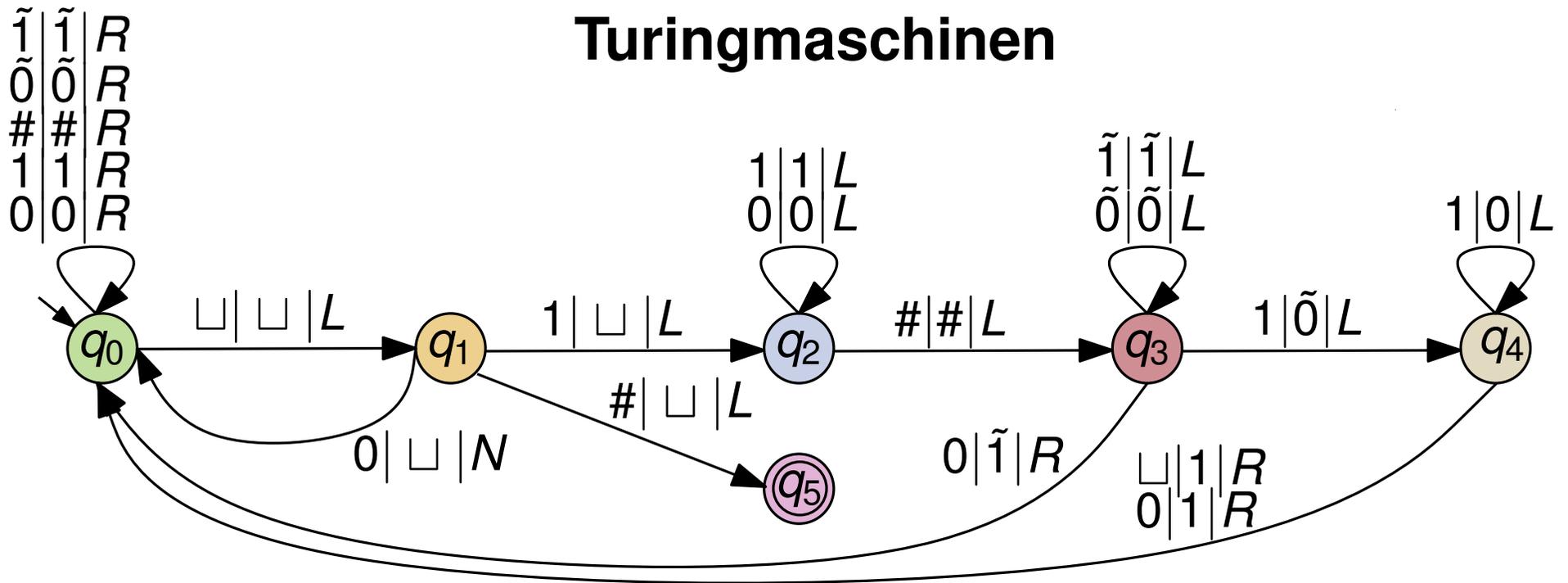
$\square 11(q_0)1\#111\square$

...

$\square 111\#111(q_0)\square$

$\square 111\#11(q_1)1\square$

Turingmaschinen



$\square(q_0)111\#111\square$

$\square 1(q_0)11\#111\square$

$\square 11(q_0)1\#111\square$

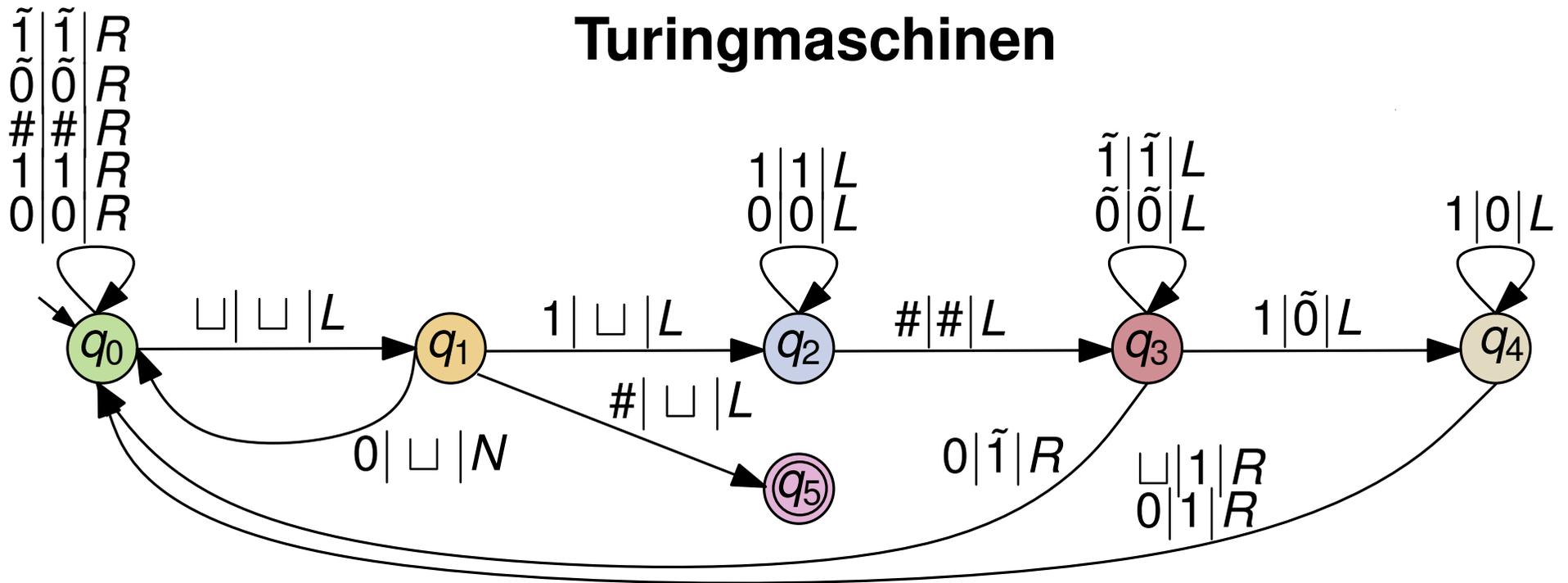
...

$\square 111\#111(q_0)\square$

$\square 111\#11(q_1)1\square$

$\square 111\#1(q_2)1\square$

Turingmaschinen



$\square(q_0)111\#111\square$ $\square111\#(q_2)11\square$

$\square1(q_0)11\#111\square$

$\square11(q_0)1\#111\square$

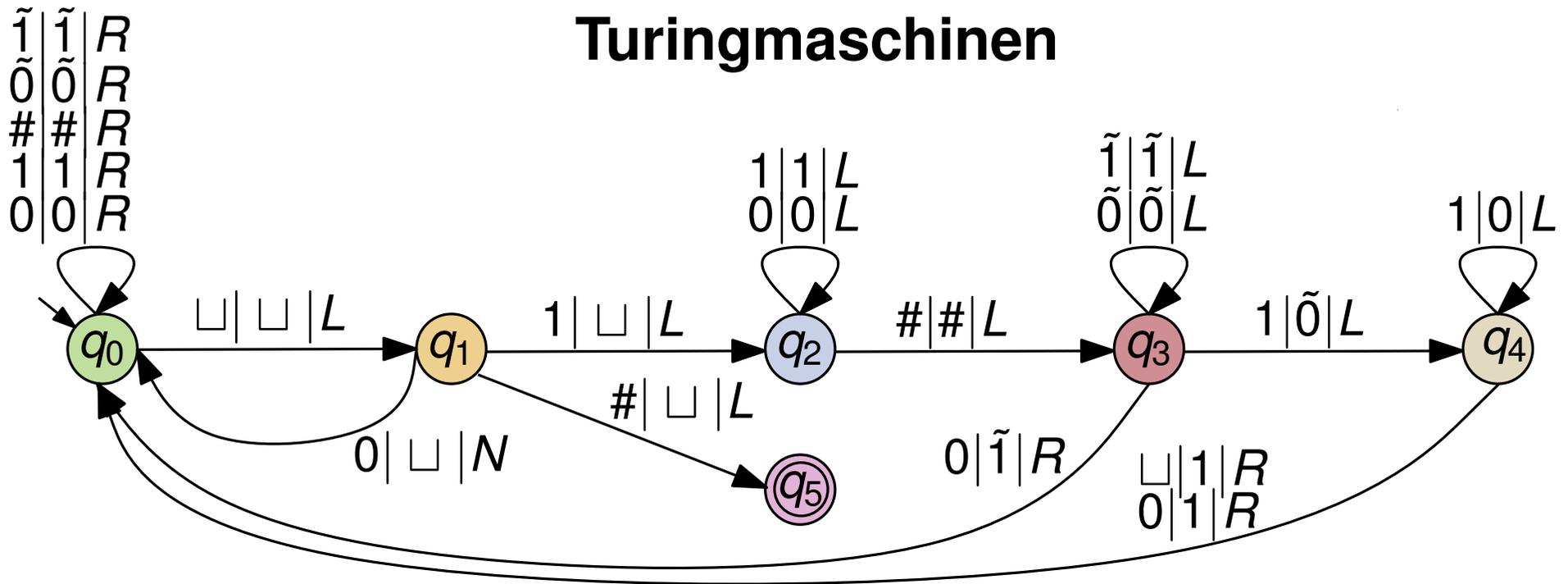
...

$\square111\#111(q_0)\square$

$\square111\#11(q_1)1\square$

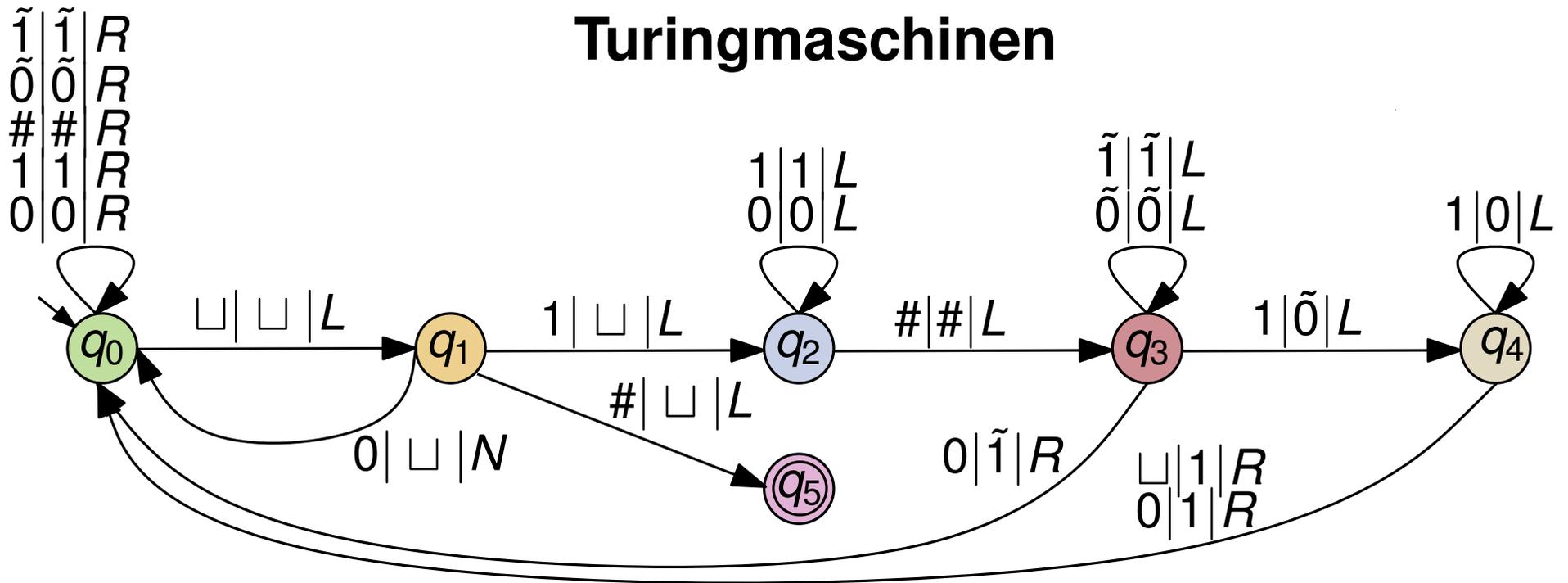
$\square111\#1(q_2)1\square$

Turingmaschinen



- $\square(q_0)111\#111\square$ $\square111\#(q_2)11\square$
- $\square1(q_0)11\#111\square$ $\square111(q_2)\#11\square$
- $\square11(q_0)1\#111\square$
- ...
- $\square111\#111(q_0)\square$
- $\square111\#11(q_1)1\square$
- $\square111\#1(q_2)1\square$

Turingmaschinen



$\square(q_0)111\#111\square$ $\square111\#(q_2)11\square$

$\square1(q_0)11\#111\square$ $\square111(q_2)\#11\square$

$\square11(q_0)1\#111\square$ $\square11(q_3)1\#11\square$

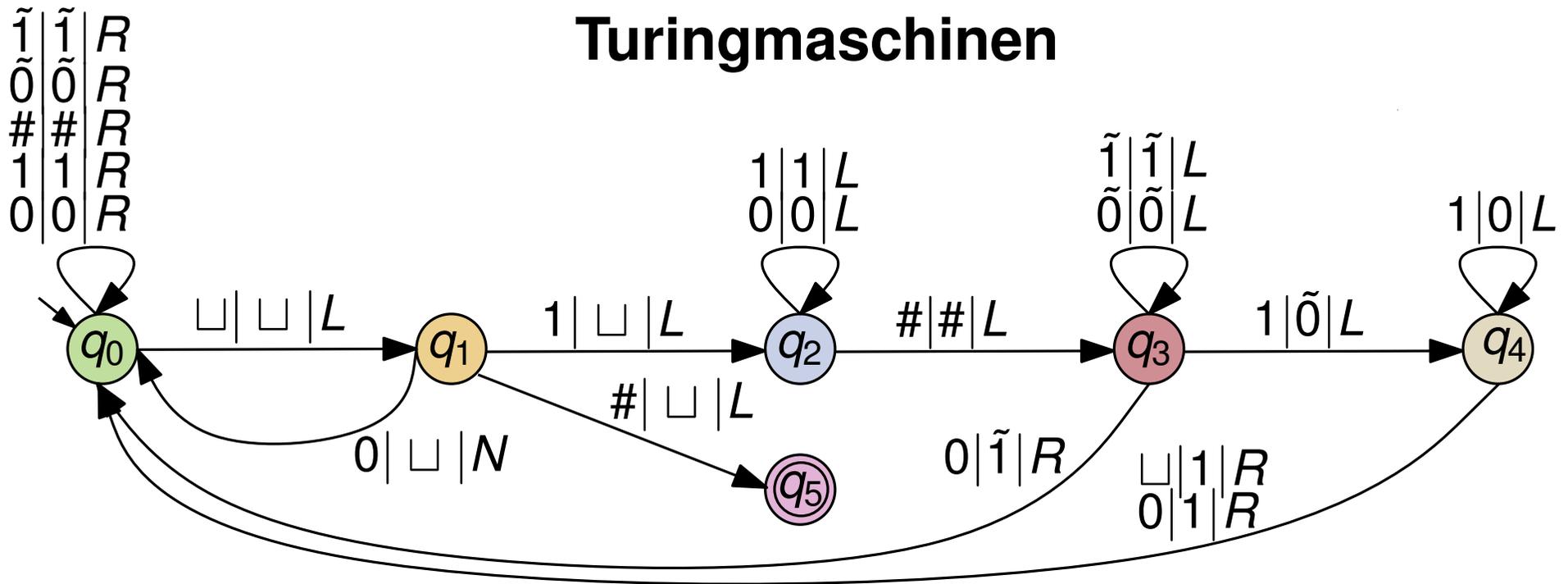
...

$\square111\#111(q_0)\square$

$\square111\#11(q_1)1\square$

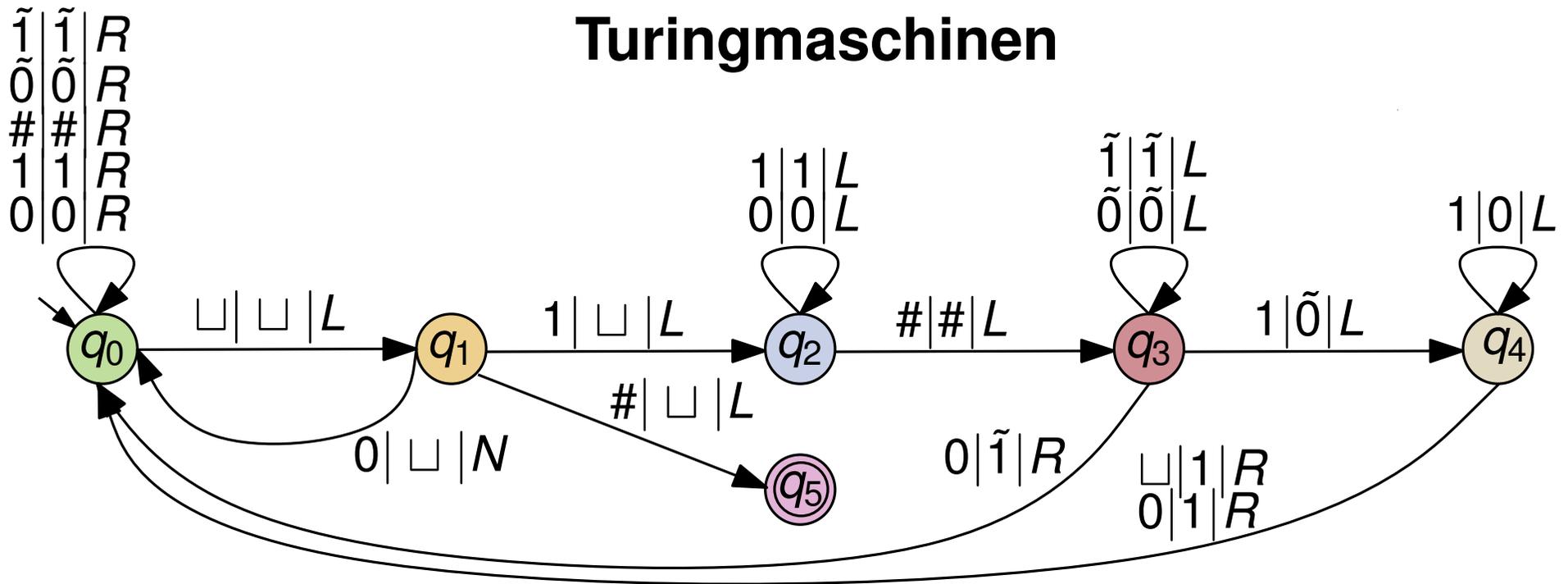
$\square111\#1(q_2)1\square$

Turingmaschinen



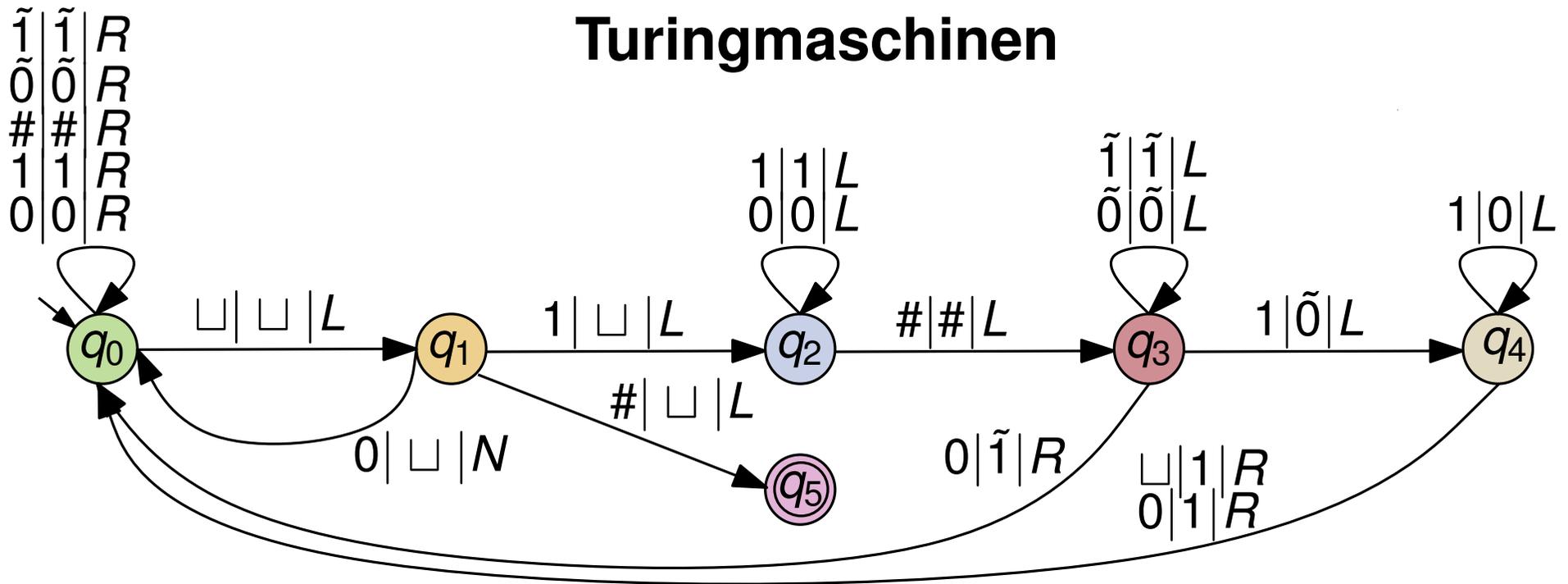
- $\square(q_0)111\#111\square$ $\square111\#(q_2)11\square$
- $\square1(q_0)11\#111\square$ $\square111(q_2)\#11\square$
- $\square11(q_0)1\#111\square$ $\square11(q_3)1\#11\square$
- ...
- $\square1(q_4)1\tilde{0}\#11\square$
- $\square111\#111(q_0)\square$
- $\square111\#11(q_1)1\square$
- $\square111\#1(q_2)1\square$

Turingmaschinen



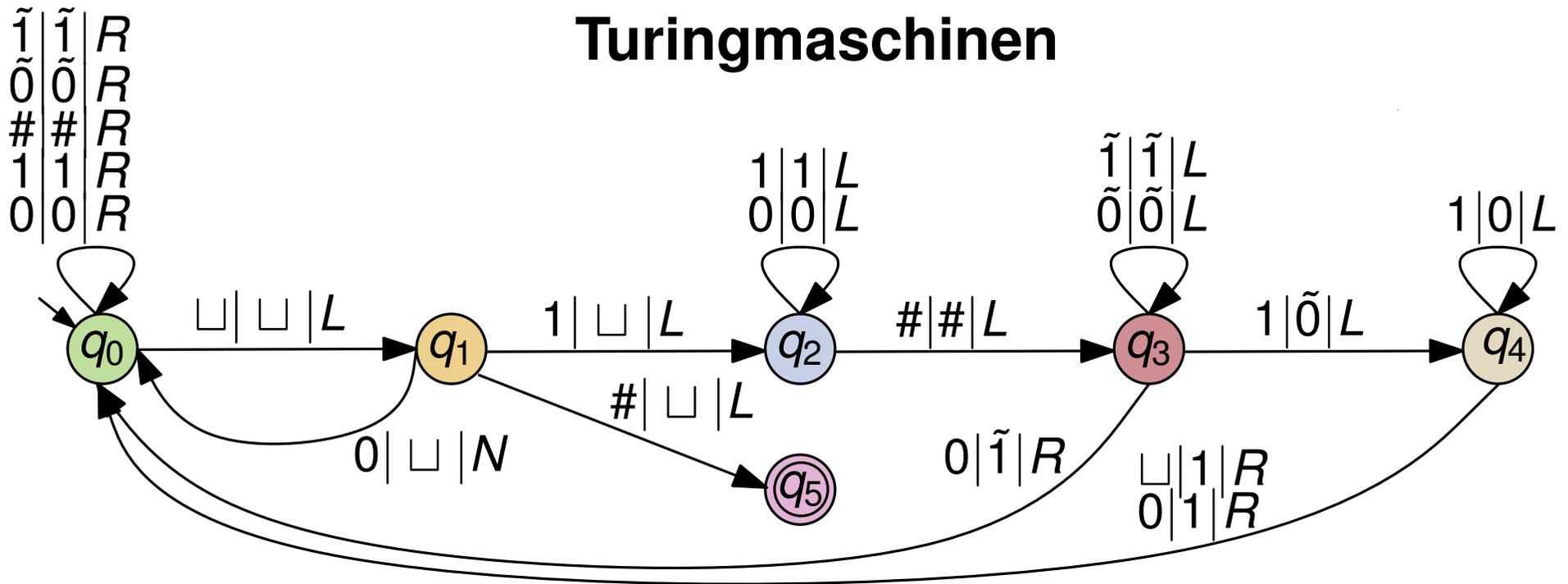
$\square(q_0)111\#111\square$ $\square111\#(q_2)11\square$
 $\square1(q_0)11\#111\square$ $\square111(q_2)\#11\square$
 $\square11(q_0)1\#111\square$ $\square11(q_3)1\#11\square$
 ... $\square1(q_4)1\tilde{0}\#11\square$
 $\square111\#111(q_0)\square$ $\square(q_4)10\tilde{0}\#11\square$
 $\square111\#11(q_1)1\square$
 $\square111\#1(q_2)1\square$

Turingmaschinen



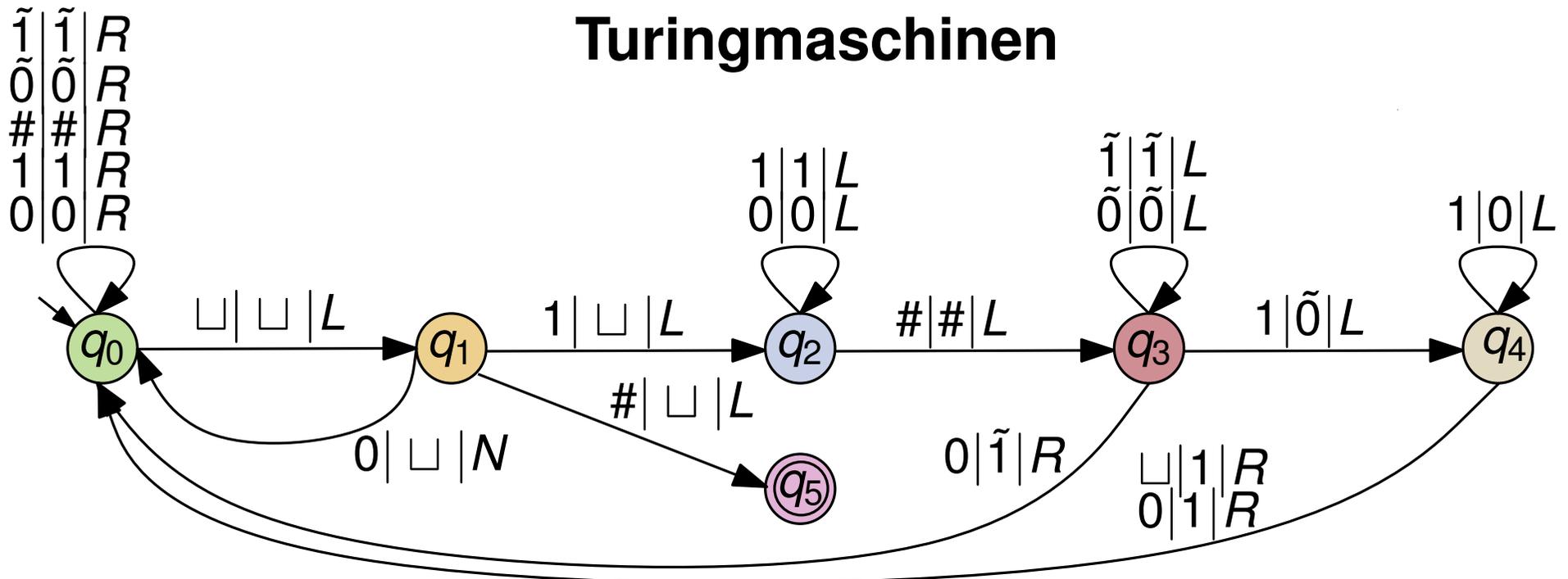
- $\square(q_0)111\#111\square$ $\square111\#(q_2)11\square$
- $\square1(q_0)11\#111\square$ $\square111(q_2)\#11\square$
- $\square11(q_0)1\#111\square$ $\square11(q_3)1\#11\square$
- ...
- $\square111\#111(q_0)\square$ $\square(q_4)10\tilde{0}\#11\square$
- $\square111\#11(q_1)1\square$ $\square(q_4)\square00\tilde{0}\#11\square$
- $\square111\#1(q_2)1\square$

Turingmaschinen



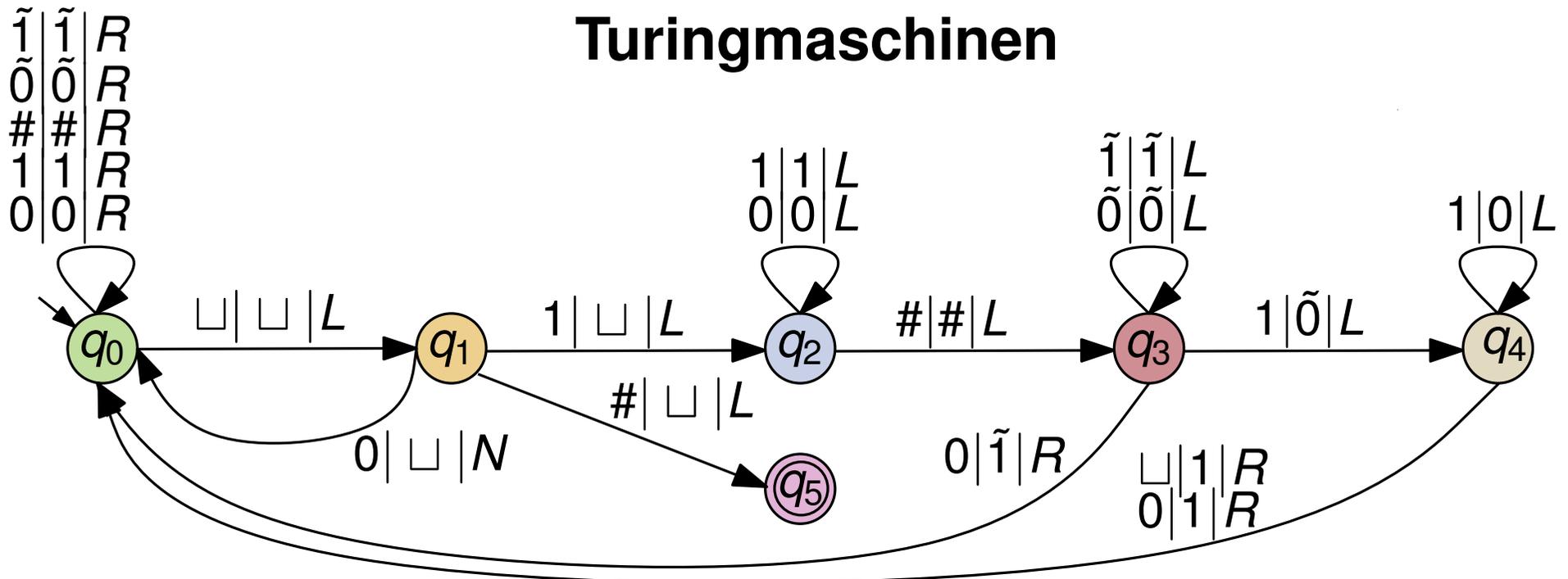
- $\square (q_0) 111 \# 111 \square$ $\square 111 \# (q_2) 11 \square$
- $\square 1 (q_0) 11 \# 111 \square$ $\square 111 (q_2) \# 11 \square$
- $\square 11 (q_0) 1 \# 111 \square$ $\square 11 (q_3) 1 \# 11 \square$
- ...
- $\square 111 \# 111 (q_0) \square$ $\square (q_4) 10\tilde{0} \# 11 \square$
- $\square 111 \# 11 (q_1) 1 \square$ $\square (q_4) \square 00\tilde{0} \# 11 \square$
- $\square 111 \# 1 (q_2) 1 \square$ $\square 1 (q_0) 00\tilde{0} \# 11 \square$

Turingmaschinen



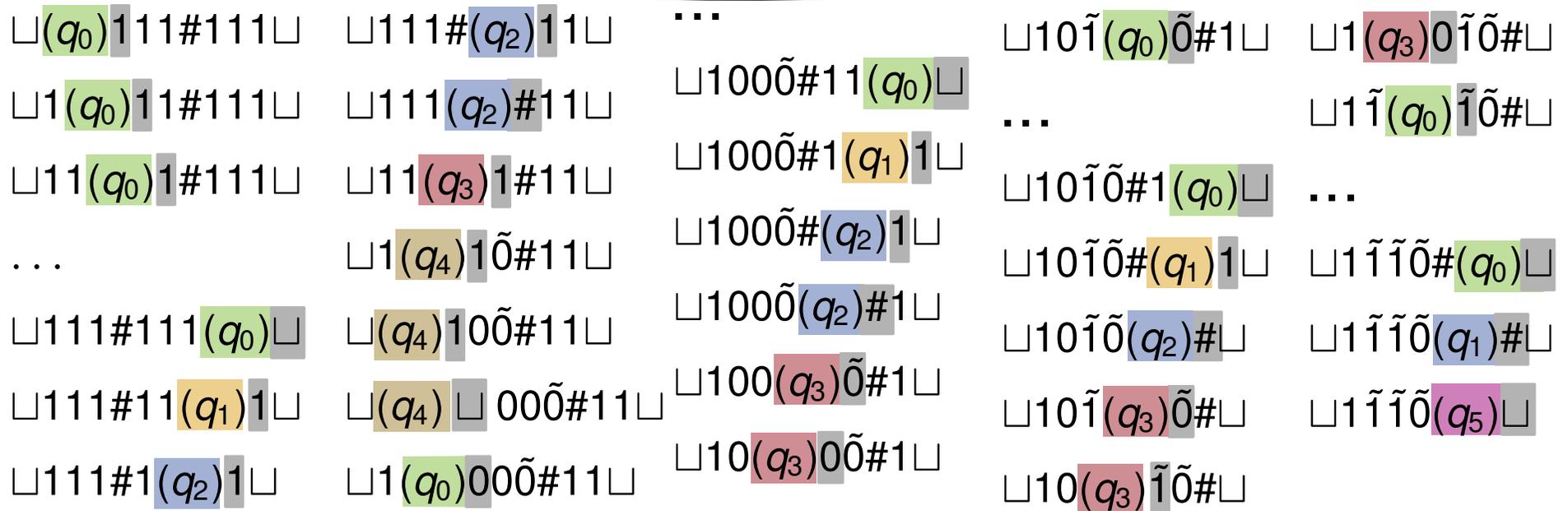
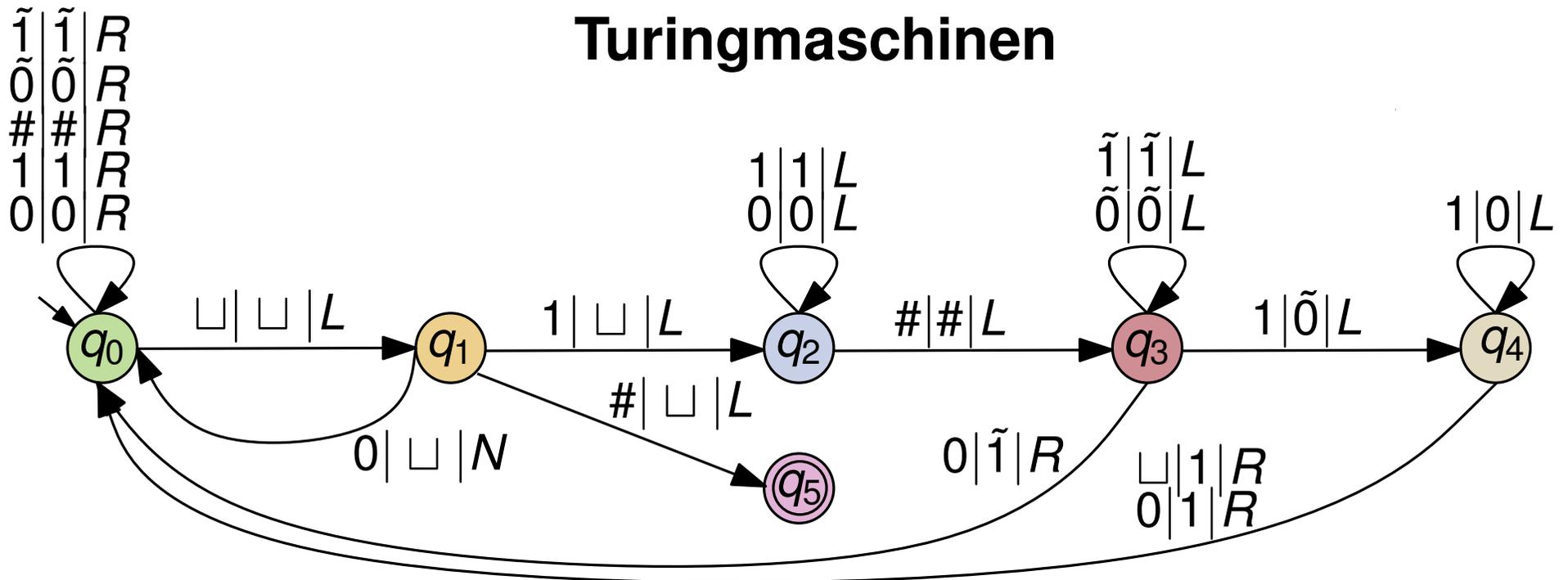
$\square(q_0)111\#111\square$	$\square111\#(q_2)11\square$...
$\square1(q_0)11\#111\square$	$\square111(q_2)\#11\square$	$\square100\tilde{0}\#11(q_0)\square$
$\square11(q_0)1\#111\square$	$\square11(q_3)1\#11\square$	$\square100\tilde{0}\#1(q_1)1\square$
...	$\square1(q_4)1\tilde{0}\#11\square$	$\square100\tilde{0}\#(q_2)1\square$
$\square111\#111(q_0)\square$	$\square(q_4)10\tilde{0}\#11\square$	$\square100\tilde{0}(q_2)\#\#1\square$
$\square111\#11(q_1)1\square$	$\square(q_4)\square00\tilde{0}\#11\square$	$\square100(q_3)\tilde{0}\#\#1\square$
$\square111\#1(q_2)1\square$	$\square1(q_0)00\tilde{0}\#11\square$	$\square10(q_3)0\tilde{0}\#\#1\square$

Turingmaschinen



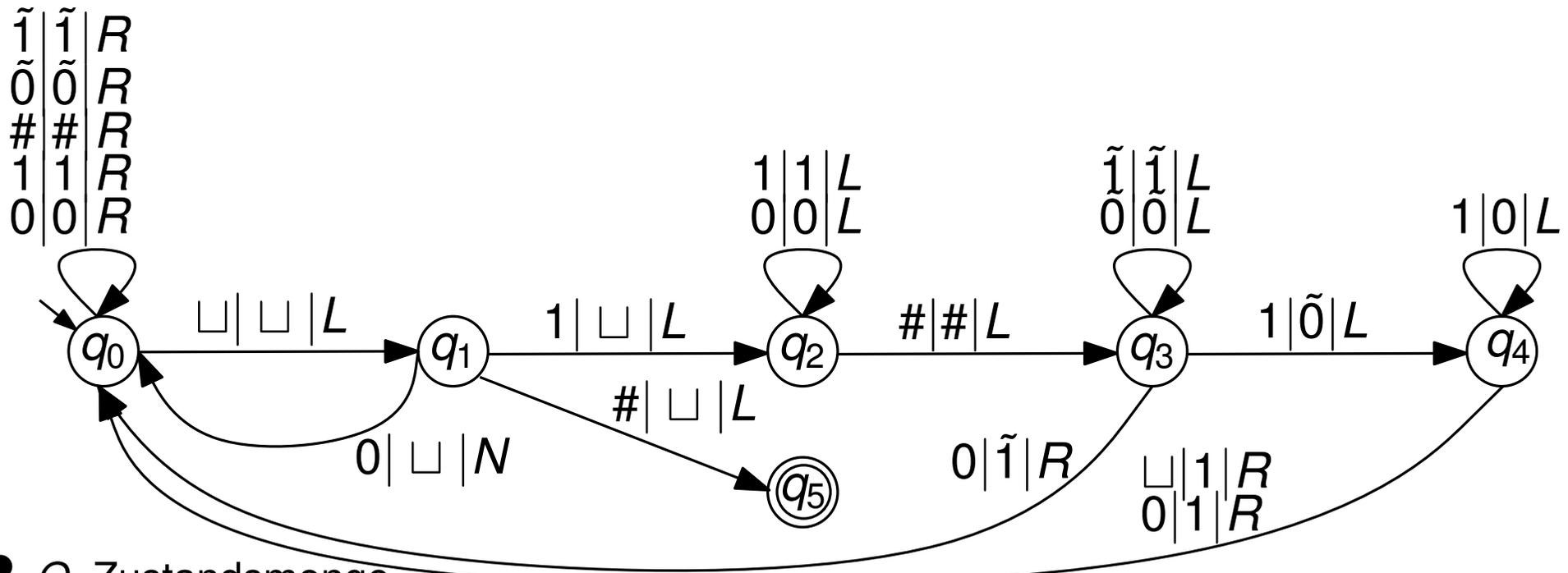
$\square(q_0)111\#111\square$	$\square111\#(q_2)11\square$...	$\square10\tilde{1}(q_0)\tilde{0}\#1\square$
$\square1(q_0)11\#111\square$	$\square111(q_2)\#11\square$	$\square100\tilde{0}\#11(q_0)\square$...
$\square11(q_0)1\#111\square$	$\square11(q_3)1\#11\square$	$\square100\tilde{0}\#1(q_1)1\square$	$\square10\tilde{1}\tilde{0}\#1(q_0)\square$
...	$\square1(q_4)1\tilde{0}\#11\square$	$\square100\tilde{0}\#(q_2)1\square$	$\square10\tilde{1}\tilde{0}\#(q_1)1\square$
$\square111\#111(q_0)\square$	$\square(q_4)10\tilde{0}\#11\square$	$\square100\tilde{0}(q_2)\#1\square$	$\square10\tilde{1}\tilde{0}(q_2)\#\square$
$\square111\#11(q_1)1\square$	$\square(q_4)\square00\tilde{0}\#11\square$	$\square100(q_3)\tilde{0}\#1\square$	$\square10\tilde{1}(q_3)\tilde{0}\#\square$
$\square111\#1(q_2)1\square$	$\square1(q_0)00\tilde{0}\#11\square$	$\square10(q_3)0\tilde{0}\#1\square$	$\square10(q_3)\tilde{1}\tilde{0}\#\square$

Turingmaschinen



Turingmaschinen

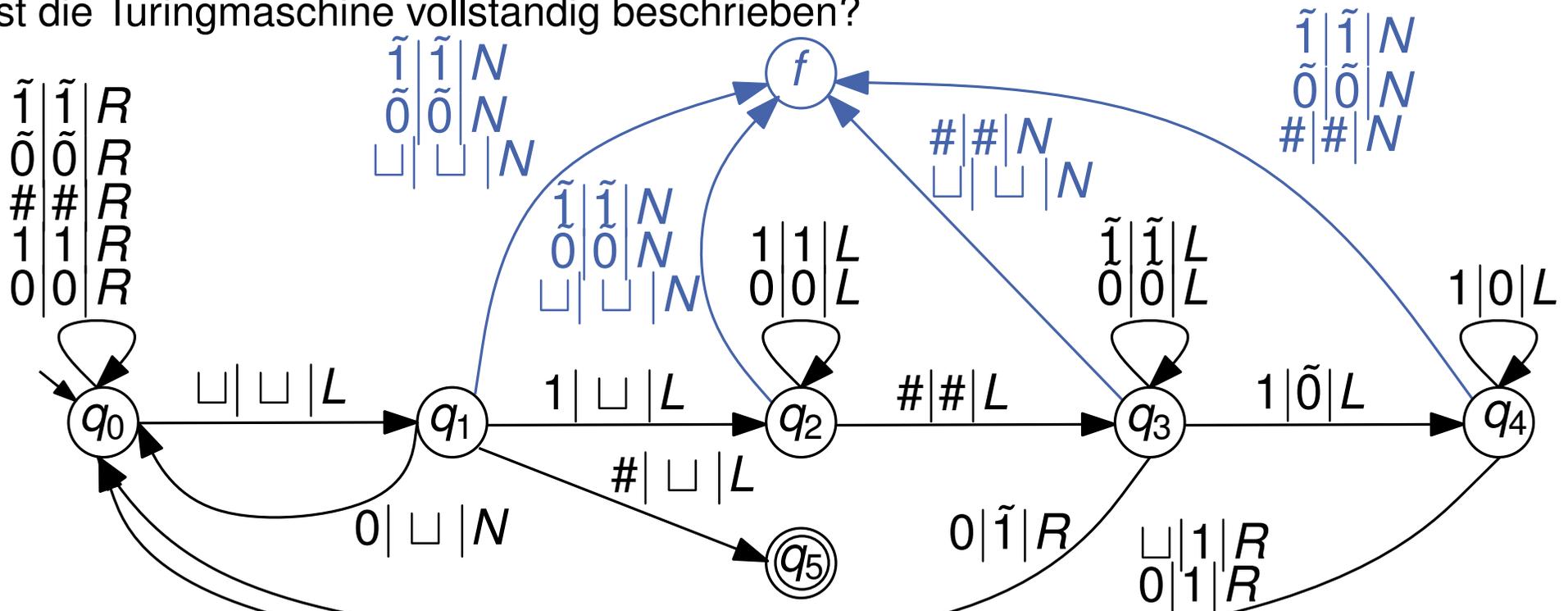
Ist die Turingmaschine vollständig beschrieben?



- Q , Zustandsmenge,
- Σ , einem endlichen Eingabealphabet,
- \sqcup , einem Blanksymbol mit $\sqcup \notin \Sigma$,
- Γ , einem endlichen Bandalphabet mit $\Sigma \cup \{\sqcup\} \subseteq \Gamma$,
- $s \in Q$, einem Startzustand,
- $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, N\}$, einer Übergangsfunktion.
- $F \subseteq Q$, einer Menge von Endzuständen.

Turingmaschinen

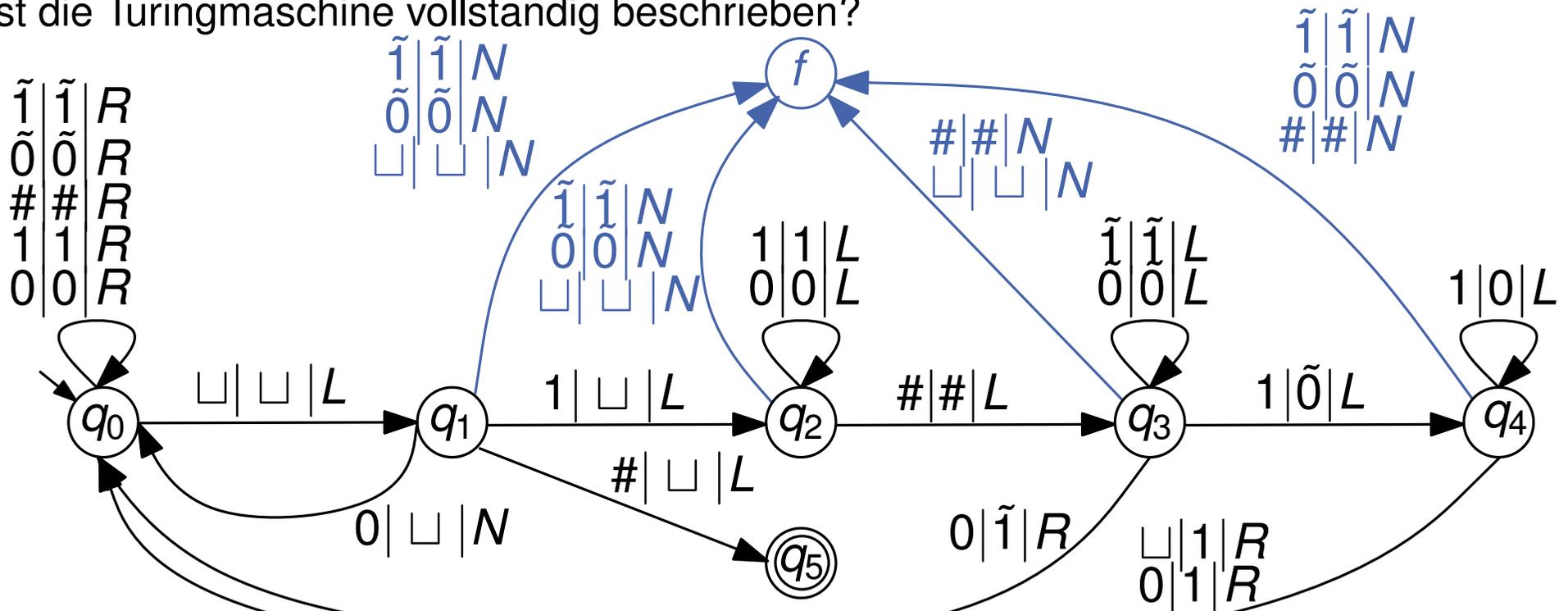
Ist die Turingmaschine vollständig beschrieben?



- Q , Zustandsmenge,
- Σ , einem endlichen Eingabealphabet,
- \sqcup , einem Blanksymbol mit $\sqcup \notin \Sigma$,
- Γ , einem endlichen Bandalphabet mit $\Sigma \cup \{\sqcup\} \subseteq \Gamma$,
- $s \in Q$, einem Startzustand,
- $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, N\}$, einer Übergangsfunktion.
- $F \subseteq Q$, einer Menge von Endzuständen.

Turingmaschinen

Ist die Turingmaschine vollständig beschrieben?



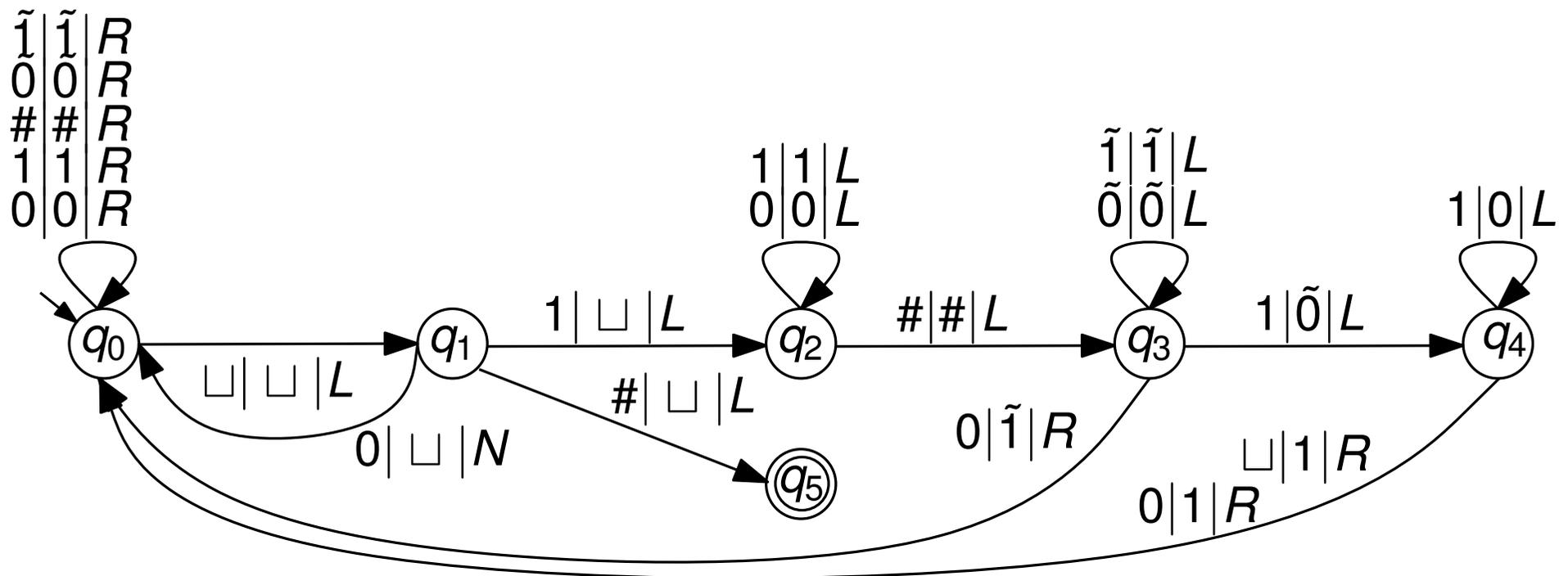
- Q , Zustandsmenge,
- Σ , einem en
- \square , einem Bl
- Γ , einem en
- $s \in Q$, einer
- $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, N\}$, einer Übergangsfunktion.
- $F \subseteq Q$, einer Menge von Endzuständen.

Was berechnet die Turingmaschine?

Turingmaschinen

Wie ändern, damit Addition von allgemeineren Eingaben funktioniert?

Beispiel: 100#110

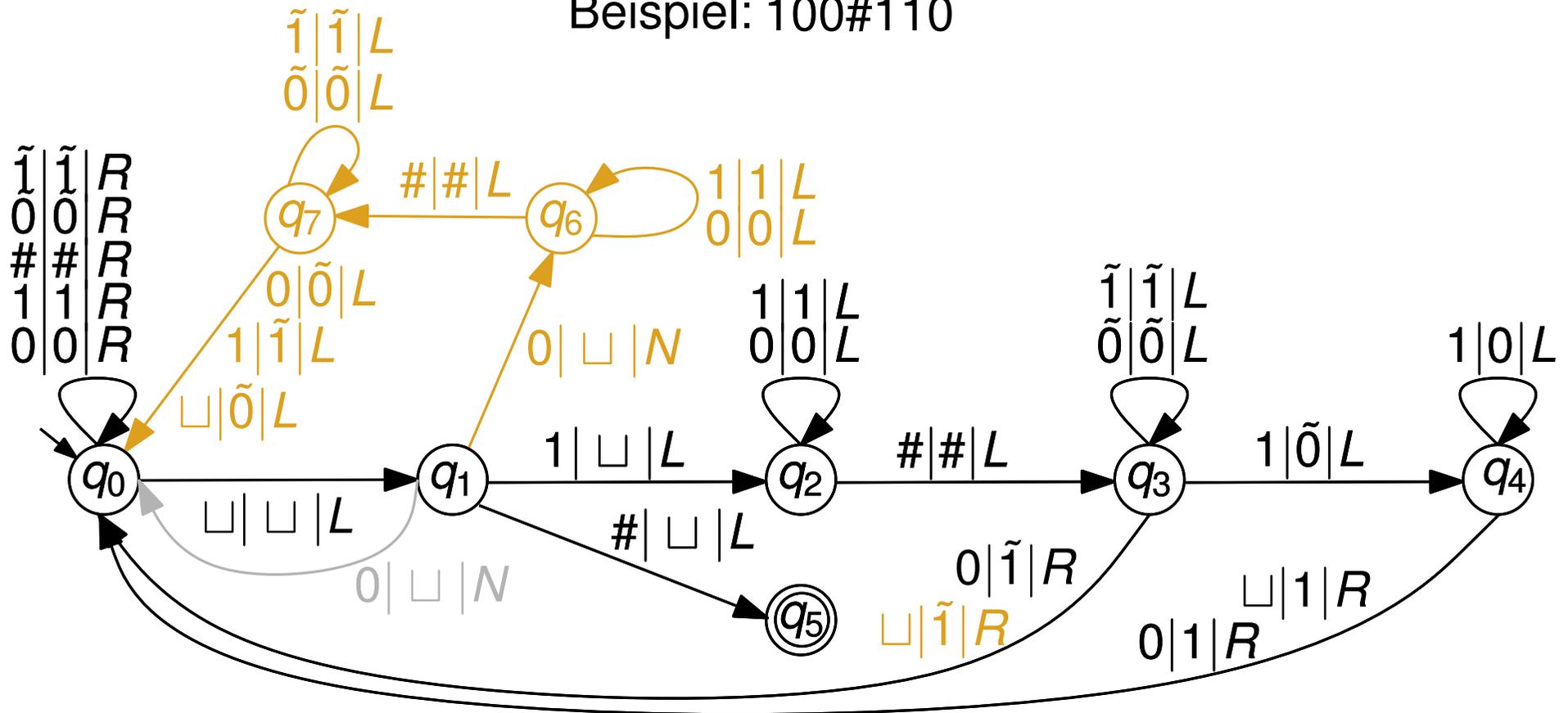


 3 min Zeit 

Turingmaschinen

Wie ändern, damit Addition von allgemeineren Eingaben funktioniert?

Beispiel: 100#110



Hinzugefügt

Entfernt

Erweiterungen von Turingmaschinen

Mehrere Spuren

Turingmaschine mit k Spuren:

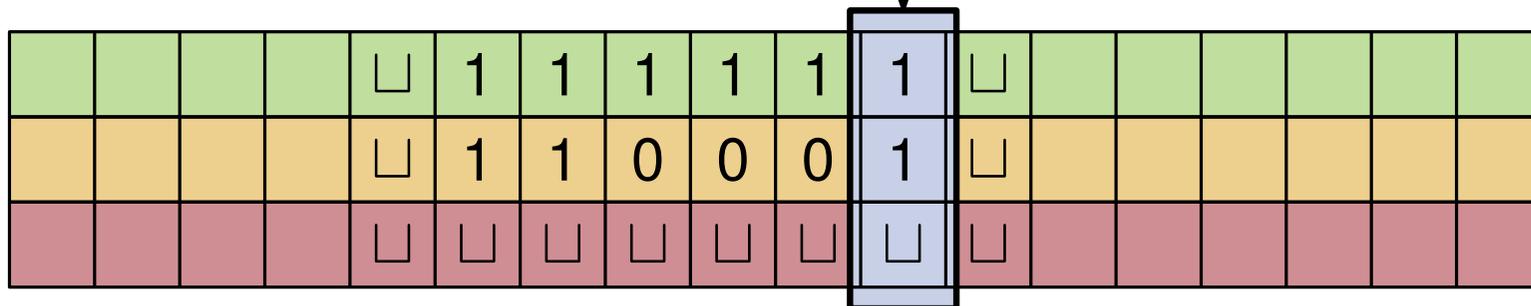
Idee: Teile das Eingabeband in k Spuren ein.

Formal: Erweitere das Bandalphabet um k -dimensionale Vektoren:

$$\Gamma_{\text{neu}} = \Sigma^k \cup \Gamma^k$$

Endliche Kontrolle

Beispiel: Addition zweier Binärzahlen.



Mehrere Spuren

Turingmaschine mit k Spuren:

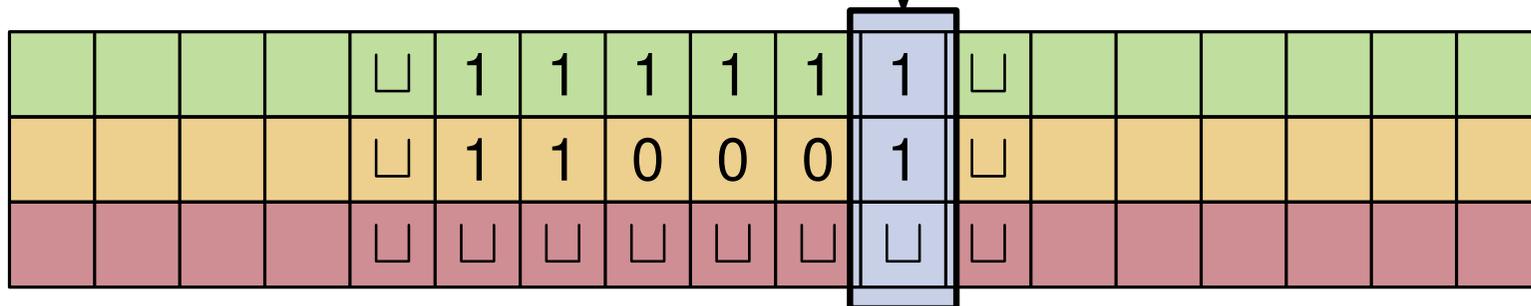
Idee: Teile das Eingabeband in k Spuren ein.

Formal: Erweitere das Bandalphabet um k -dimensionale Vektoren:

$$\Gamma_{\text{neu}} = \Sigma^k \cup \Gamma^k$$

Endliche Kontrolle

Beispiel: Addition zweier Binärzahlen.



- Verwende **erste Spur** für ersten Summand, **zweite Spur** für zweiten Summand und **dritte Spur** für Ergebnis.

Mehrere Spuren

Turingmaschine mit k Spuren:

Idee: Teile das Eingabeband in k Spuren ein.

Formal: Erweitere das Bandalphabet um k -dimensionale Vektoren:

$$\Gamma_{\text{neu}} = \Sigma^k \cup \Gamma^k$$

Endliche Kontrolle

Beispiel: Addition zweier Binärzahlen.

				□	1	1	1	1	1	1	□						
				□	1	1	0	0	0	1	□						
				□	□	□	□	□	□	□	□						

- Verwende **erste Spur** für ersten Summand, **zweite Spur** für zweiten Summand und **dritte Spur** für Ergebnis.
- Verwende *Schulmethode* für Addition: Kopf läuft von links nach rechts.

Mehrere Spuren

Turingmaschine mit k Spuren:

Idee: Teile das Eingabeband in k Spuren ein.

Formal: Erweitere das Bandalphabet um k -dimensionale Vektoren:

$$\Gamma_{\text{neu}} = \Sigma^k \cup \Gamma^k$$

Endliche Kontrolle

Beispiel: Addition zweier Binärzahlen.

				□	1	1	1	1	1	1	□						
				□	1	1	0	0	0	1	□						
				□	□	□	□	□	□	□	□						

- Verwende **erste Spur** für ersten Summand, **zweite Spur** für zweiten Summand und **dritte Spur** für Ergebnis.
- Verwende *Schulmethode* für Addition: Kopf läuft von links nach rechts.
- Speichere Übertrag in Zustand.

Mehrere Spuren

Turingmaschine mit k Spuren:

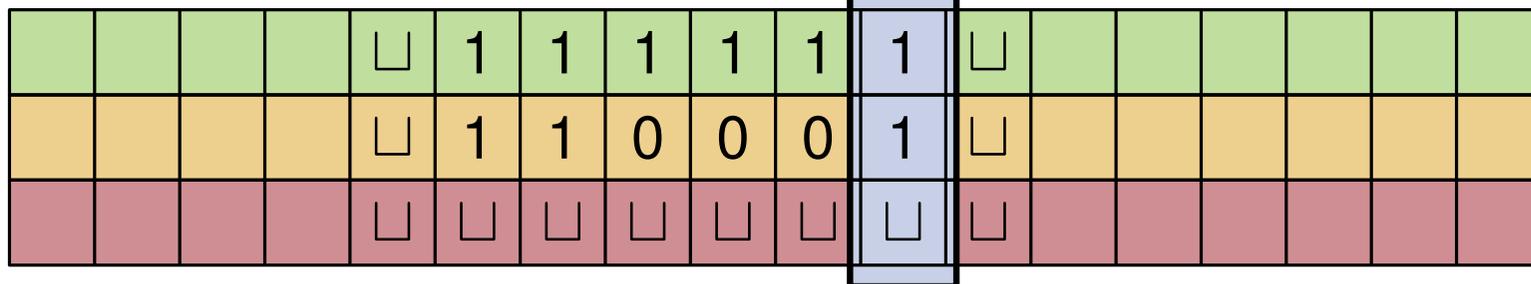
Idee: Teile das Eingabeband in k Spuren ein.

Formal: Erweitere das Bandalphabet um k -dimensionale Vektoren:

$$\Gamma_{\text{neu}} = \Sigma^k \cup \Gamma^k$$

Endliche Kontrolle

Beispiel: Addition zweier Binärzahlen.



- Verwende **erste Spur** für ersten Summand, **zweite Spur** für zweiten
- Ist eine Turingmaschine mit $k \geq 2$ mächtiger, als eine Turing-
- maschine mit einer Spur? ts.



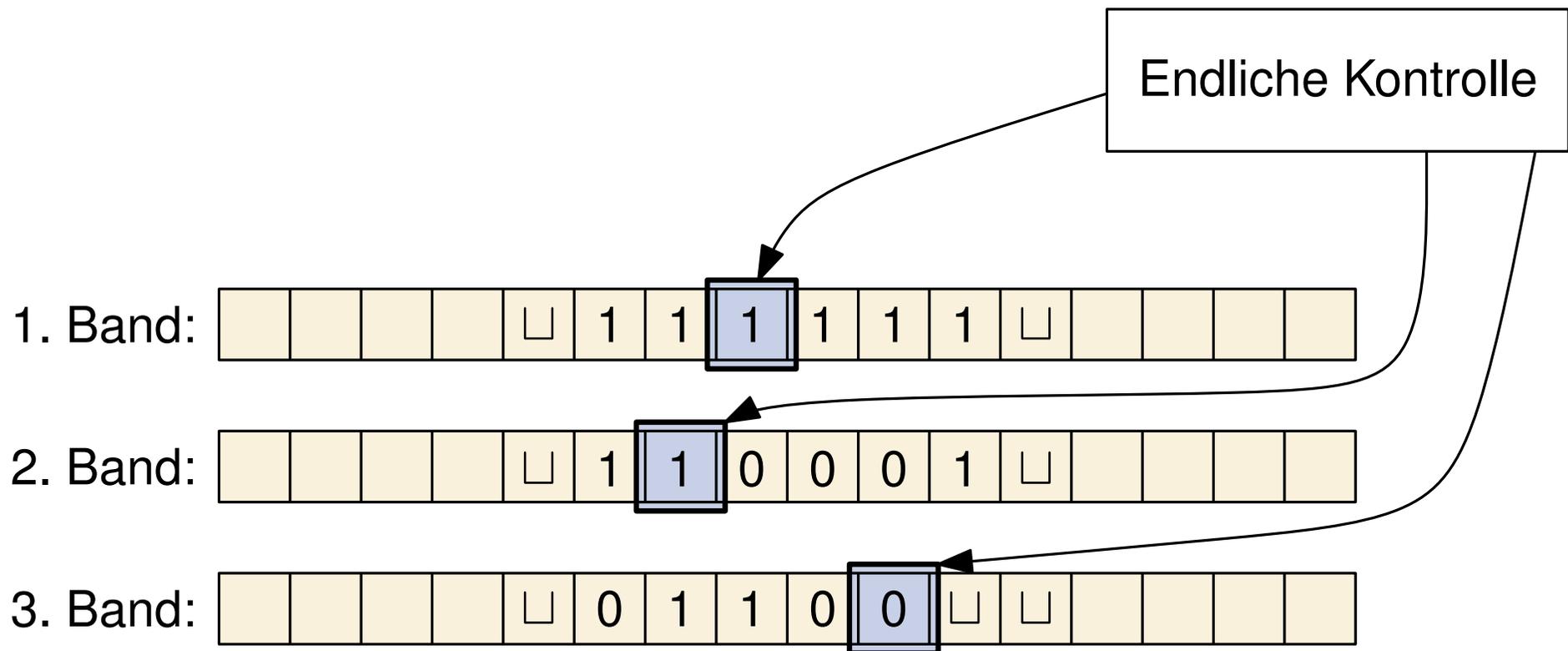
Mehrere Bänder

Turingmaschine mit k Bändern:

Idee: Es gibt k Bänder und auf jedem Band arbeitet eigener Kopf.

Formal: Passe Übergangsfunktion δ an.

$$\delta: Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{R, L, N\}^k$$



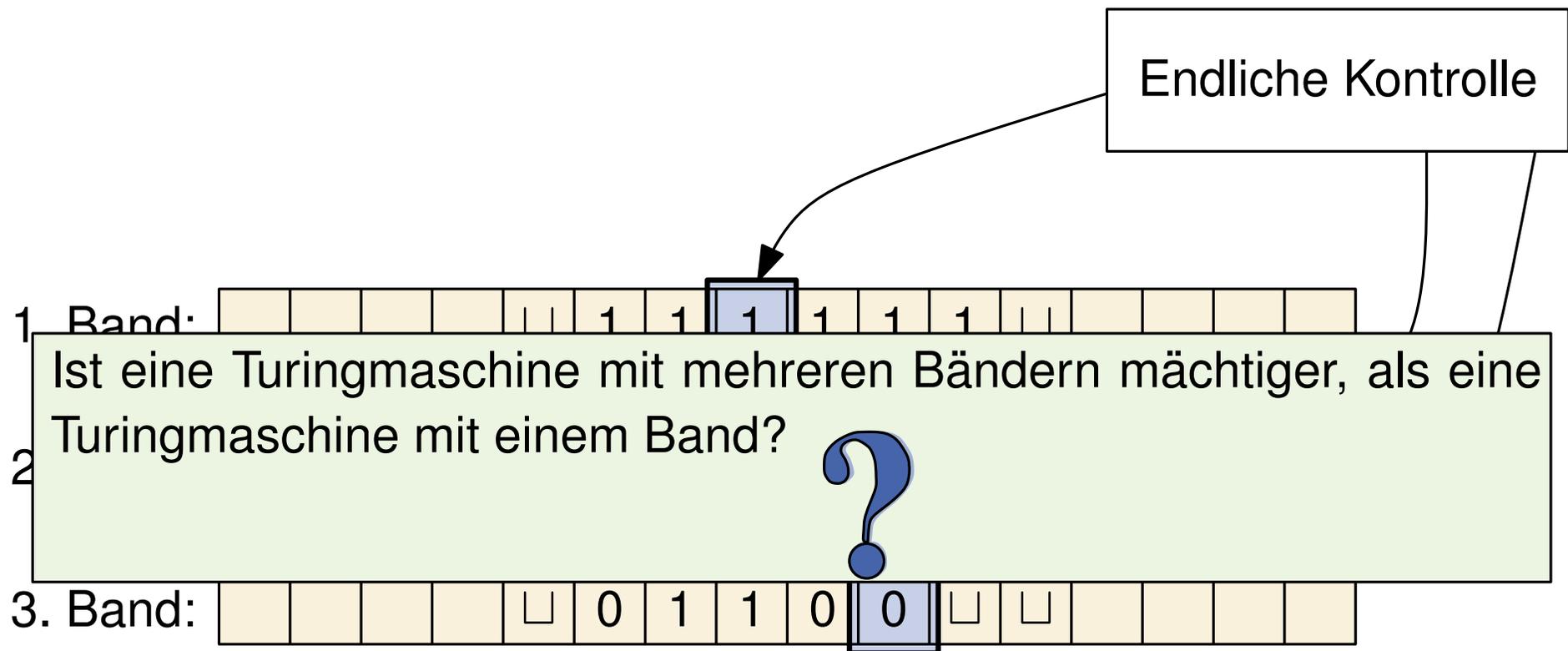
Mehrere Bänder

Turingmaschine mit k Bändern:

Idee: Es gibt k Bänder und auf jedem Band arbeitet eigener Kopf.

Formal: Passe Übergangsfunktion δ an.

$$\delta: Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{R, L, N\}^k$$



Erweiterte Turingmaschinen

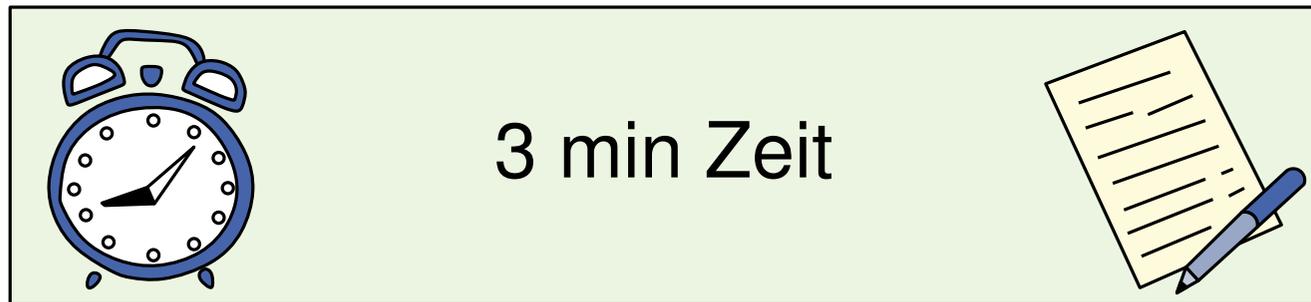
Satz: Eine k -Band Turingmaschine \mathcal{M} , die mit Rechenzeit $t(n)$ und Platz $s(n)$ auskommt, kann von einer Turingmaschine \mathcal{M}' mit Zeitbedarf $O(t^2(n))$ und Platzbedarf $O(s(n))$ simuliert werden.

Beweis:

Erweiterte Turingmaschinen

Satz: Eine k -Band Turingmaschine \mathcal{M} , die mit Rechenzeit $t(n)$ und Platz $s(n)$ auskommt, kann von einer Turingmaschine \mathcal{M}' mit Zeitbedarf $O(t^2(n))$ und Platzbedarf $O(s(n))$ simuliert werden.

Beweis:



Erweiterte Turingmaschinen

Satz: Eine k -Band Turingmaschine \mathcal{M} , die mit Rechenzeit $t(n)$ und Platz $s(n)$ auskommt, kann von einer Turingmaschine \mathcal{M}' mit Zeitbedarf $O(t^2(n))$ und Platzbedarf $O(s(n))$ simuliert werden.

Beweis: Die Turingmaschine \mathcal{M}' verwendet $2k$ Spuren
Nach Simulation des t -en Schritts für $1 \leq t \leq t(n)$ gilt:

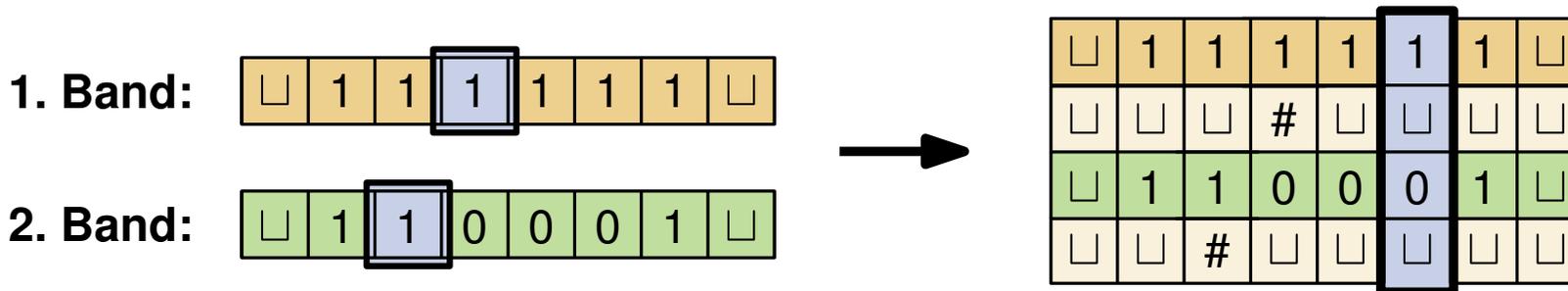
Erweiterte Turingmaschinen

Satz: Eine k -Band Turingmaschine \mathcal{M} , die mit Rechenzeit $t(n)$ und Platz $s(n)$ auskommt, kann von einer Turingmaschine \mathcal{M}' mit Zeitbedarf $O(t^2(n))$ und Platzbedarf $O(s(n))$ simuliert werden.

Beweis: Die Turingmaschine \mathcal{M}' verwendet $2k$ Spuren

Nach Simulation des t -en Schritts für $1 \leq t \leq t(n)$ gilt:

- Die ungeraden Spuren $1, 3, 5, \dots, 2k - 1$ enthalten Inhalt der Bänder $1 \dots, k$ von M .
- Auf den geraden Spuren $1, 2, \dots, 2k$ sind die Kopfposition auf diesen Bändern mit dem Zeichen $\#$ markiert.



Erweiterte Turingmaschinen

Satz: Eine k -Band Turingmaschine \mathcal{M} , die mit Rechenzeit $t(n)$ und Platz $s(n)$ auskommt, kann von einer Turingmaschine \mathcal{M}' mit Zeitbedarf $O(t^2(n))$ und Platzbedarf $O(s(n))$ simuliert werden.

Beweis: Der Kopf von \mathcal{M}' steht auf dem linkesten #.

Jeder Rechenschritt von \mathcal{M} wird durch \mathcal{M}' wie folgt simuliert:

Erweiterte Turingmaschinen

Satz: Eine k -Band Turingmaschine \mathcal{M} , die mit Rechenzeit $t(n)$ und Platz $s(n)$ auskommt, kann von einer Turingmaschine \mathcal{M}' mit Zeitbedarf $O(t^2(n))$ und Platzbedarf $O(s(n))$ simuliert werden.

Beweis: Der Kopf von \mathcal{M}' steht auf dem linkesten #.

Jeder Rechenschritt von \mathcal{M} wird durch \mathcal{M}' wie folgt simuliert:

- 1. Schritt:** Der Kopf von \mathcal{M}' läuft nach rechts bis zum rechtesten # und merkt sich in einem *endlichen Speicher*, was die k Köpfe von \mathcal{M} lesen würden.

Erweiterte Turingmaschinen

Satz: Eine k -Band Turingmaschine \mathcal{M} , die mit Rechenzeit $t(n)$ und Platz $s(n)$ auskommt, kann von einer Turingmaschine \mathcal{M}' mit Zeitbedarf $O(t^2(n))$ und Platzbedarf $O(s(n))$ simuliert werden.

Beweis: Der Kopf von \mathcal{M}' steht auf dem linkesten #.

Jeder Rechenschritt von \mathcal{M} wird durch \mathcal{M}' wie folgt simuliert:

- 1. Schritt:** Der Kopf von \mathcal{M}' läuft nach rechts bis zum rechtesten # und merkt sich in einem *endlichen Speicher*, was die k Köpfe von \mathcal{M} lesen würden.
- 2. Schritt:** Der Kopf von \mathcal{M}' läuft nach links und an den entsprechenden Markierungen werden die Banschriiten so verändert, wie es \mathcal{M} tun würde.

Die Positionsmarkierungen werden ggf. nach links bzw. rechts verschoben.

Merke: Zustand, in dem sich nun \mathcal{M} befindet.

Erweiterte Turingmaschinen

Satz: Eine k -Band Turingmaschine \mathcal{M} , die mit Rechenzeit $t(n)$ und Platz $s(n)$ auskommt, kann von einer Turingmaschine \mathcal{M}' mit Zeitbedarf $O(t^2(n))$ und Platzbedarf $O(s(n))$ simuliert werden.

Beweis: Der Kopf von \mathcal{M}' steht auf dem linkesten #.

Jeder Rechenschritt von \mathcal{M} wird durch \mathcal{M}' wie folgt simuliert:

1. Schritt: Der Kopf von \mathcal{M}' läuft nach rechts bis zum rechtesten # und merkt sich in einem *endlichen Speicher*, was die k Köpfe von \mathcal{M} lesen würden.

2. Schritt: Der Kopf von \mathcal{M}' läuft nach links und an den entsprechenden Markierungen werden die Bandschriften so verändert, wie es \mathcal{M} tun würde.

Die Positionsmarkierungen werden ggf. nach links bzw. rechts verschoben.

Merke: Zustand, in dem sich nun \mathcal{M} befindet.

3. Schritt: Der Kopf von \mathcal{M}' läuft zum linkesten #

⇒ Ausgangssituation für nächsten Berechnungsschritt $t + 1$.

Erweiterte Turingmaschinen

Satz: Eine k -Band Turingmaschine \mathcal{M} , die mit Rechenzeit $t(n)$ und Platz $s(n)$ auskommt, kann von einer Turingmaschine \mathcal{M}' mit Zeitbedarf $O(t^2(n))$ und Platzbedarf $O(s(n))$ simuliert werden.

Erweiterte Turingmaschinen

Satz: Eine k -Band Turingmaschine \mathcal{M} , die mit Rechenzeit $t(n)$ und Platz $s(n)$ auskommt, kann von einer Turingmaschine \mathcal{M}' mit Zeitbedarf $O(t^2(n))$ und Platzbedarf $O(s(n))$ simuliert werden.

- Laufzeit:**
- $t(n)$ beschränkt die Breite des betrachteten Bandes von \mathcal{M}' .
 - Jede Rechenschritt von $t(n)$ wird in $O(t(n))$ Zeit simuliert.
- $\Rightarrow \mathcal{M}'$ simuliert \mathcal{M} in $O(t^2(n))$ Zeit.

Erweiterte Turingmaschinen

Satz: Eine k -Band Turingmaschine \mathcal{M} , die mit Rechenzeit $t(n)$ und Platz $s(n)$ auskommt, kann von einer Turingmaschine \mathcal{M}' mit Zeitbedarf $O(t^2(n))$ und Platzbedarf $O(s(n))$ simuliert werden.

Laufzeit:

- $t(n)$ beschränkt die Breite des betrachteten Bandes von \mathcal{M}' .
- Jede Rechenschritt von $t(n)$ wird in $O(t(n))$ Zeit simuliert.

$\Rightarrow \mathcal{M}'$ simuliert \mathcal{M} in $O(t^2(n))$ Zeit.

Speicherverbrauch: \mathcal{M}' braucht nur um einen konstanten Faktor mehr Speicher:

- Anzahl Spuren von $\mathcal{M}' = 2 \cdot$ Anzahl Bänder von \mathcal{M} .
- Speicher für Simulation der Zustände von \mathcal{M} .

Entscheidbarkeit

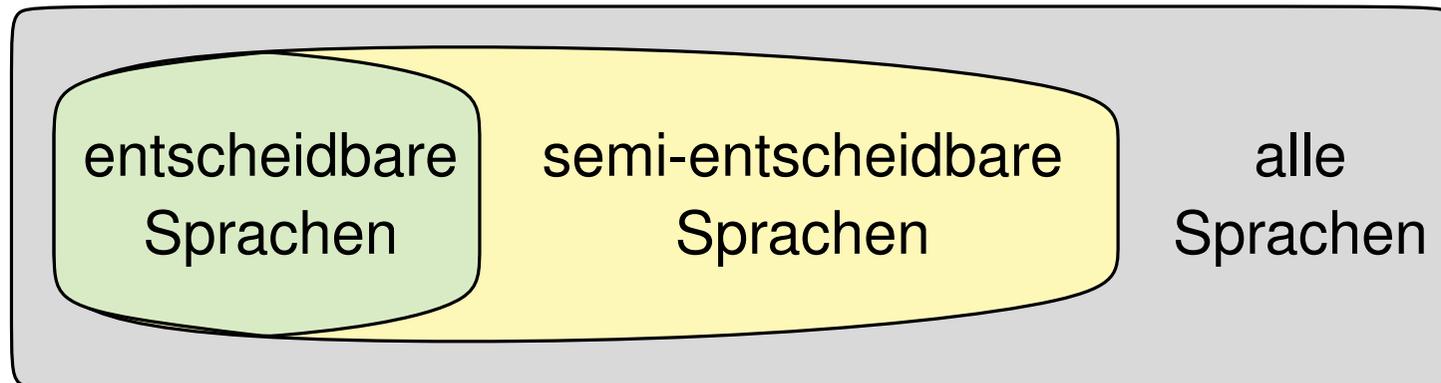
Entscheidbarkeit – Definitionen

Eine Sprache $L \subseteq \Sigma^*$ heißt **semi-entscheidbar**, wenn es eine Turingmaschine gibt, die eine Eingabe w genau dann akzeptiert, wenn $w \in L$ gilt.

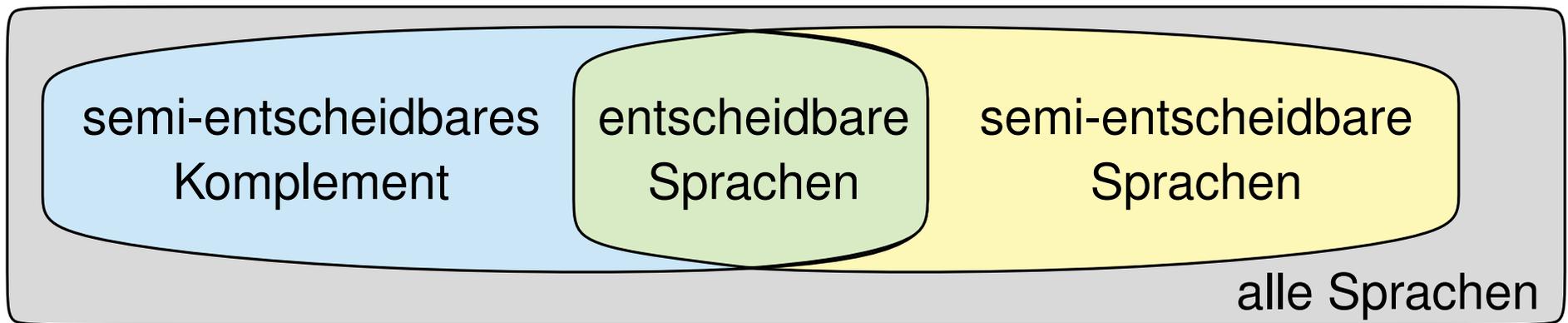
L wird
“erkannt”

Eine Sprache $L \subseteq \Sigma^*$ heißt **entscheidbar**, wenn es eine Turingmaschine gibt, die auf allen Eingaben stoppt und eine Eingabe w genau dann akzeptiert, wenn $w \in L$ gilt.

L wird
“entschieden”

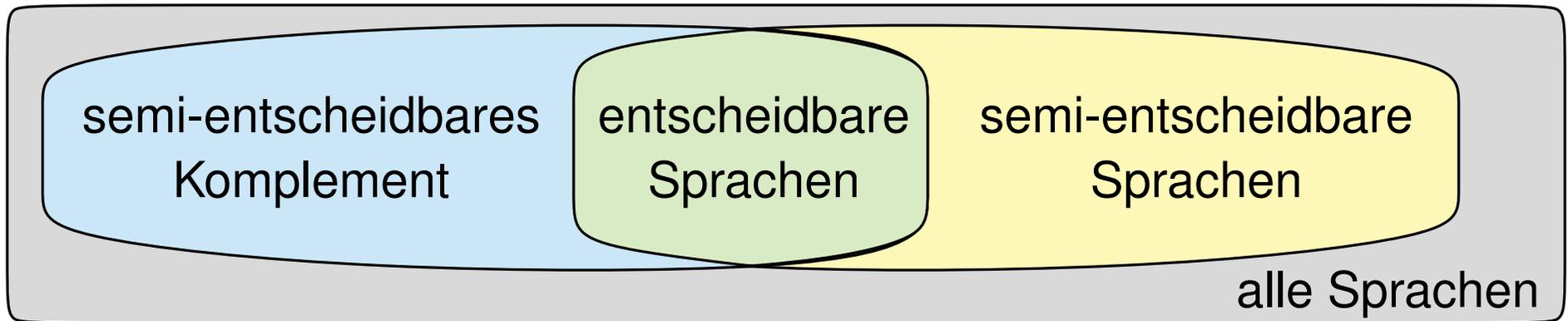


Entscheidbarkeit – Zusammenhänge



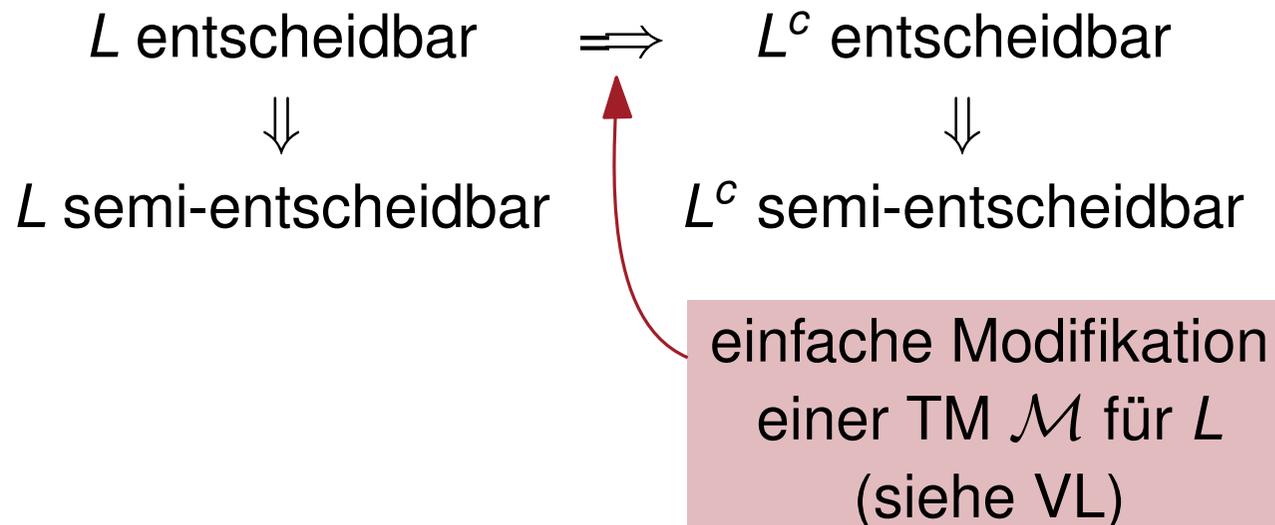
Satz: L ist entscheidbar $\iff L$ und L^c sind semi-entscheidbar

Entscheidbarkeit – Zusammenhänge

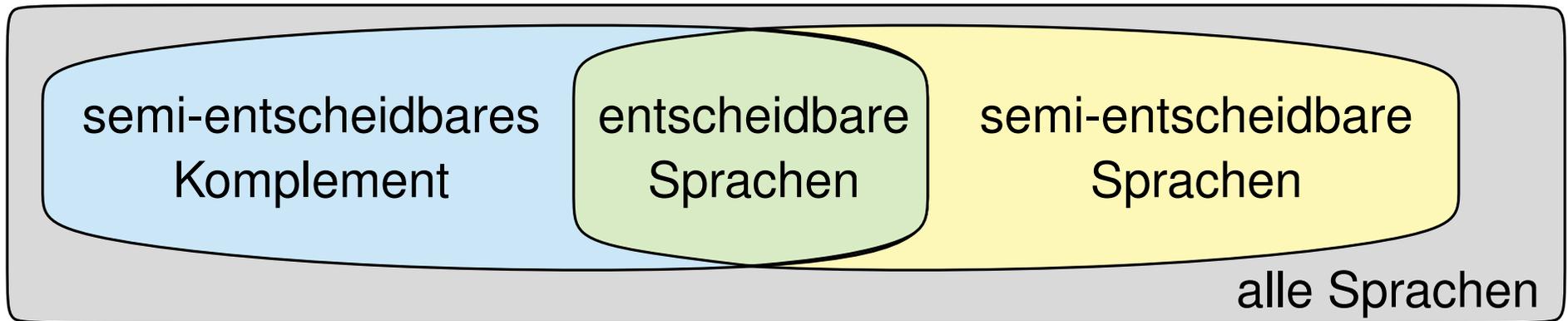


Satz: L ist entscheidbar $\iff L$ und L^c sind semi-entscheidbar

Beweis, links nach rechts:



Entscheidbarkeit – Zusammenhänge



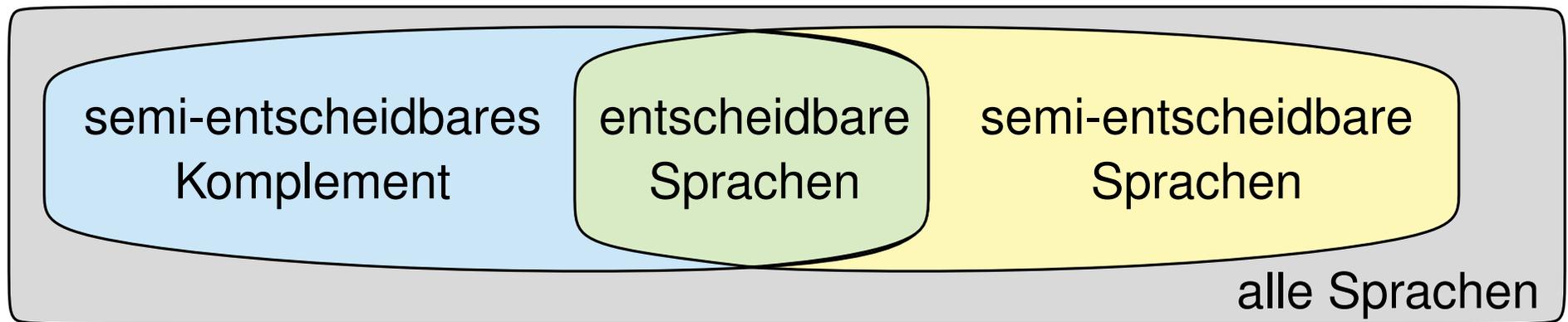
Satz: L ist entscheidbar $\iff L$ und L^c sind semi-entscheidbar

Beweis, rechts nach links:

L semi-entscheidbar \rightsquigarrow Turingmaschine \mathcal{M}_1 akzeptiert bei $w \in L$

L^c semi-entscheidbar \rightsquigarrow Turingmaschine \mathcal{M}_2 akzeptiert bei $w \notin L$

Entscheidbarkeit – Zusammenhänge



Satz: L ist entscheidbar $\iff L$ und L^c sind semi-entscheidbar

Beweis, rechts nach links:

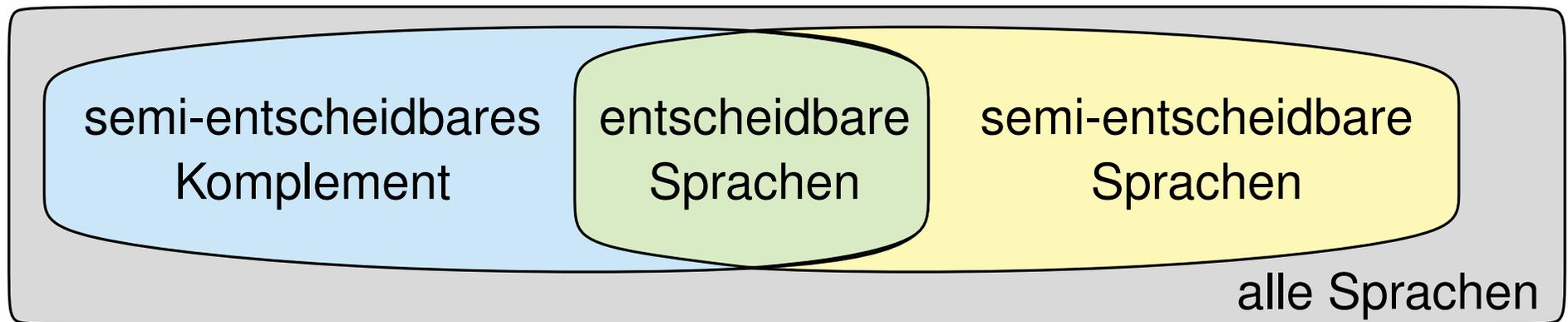
L semi-entscheidbar \rightsquigarrow Turingmaschine \mathcal{M}_1 akzeptiert bei $w \in L$

L^c semi-entscheidbar \rightsquigarrow Turingmaschine \mathcal{M}_2 akzeptiert bei $w \notin L$

Konstruiere 2-Band Turingmaschine \mathcal{M} :

- ▶ Schreibe w auf beide Bänder
- ▶ Führe \mathcal{M}_1 auf Band 1 und \mathcal{M}_2 auf Band 2 aus
- ▶ Akzeptiere wenn \mathcal{M}_1 akzeptiert, halte wenn \mathcal{M}_2 akzeptiert

Entscheidbarkeit – Zusammenhänge



Satz: L ist entscheidbar $\iff L$ und L^c sind semi-entscheidbar

Beweis, rechts nach links:

L semi-entscheidbar \rightsquigarrow Turingmaschine \mathcal{M}_1 akzeptiert bei $w \in L$

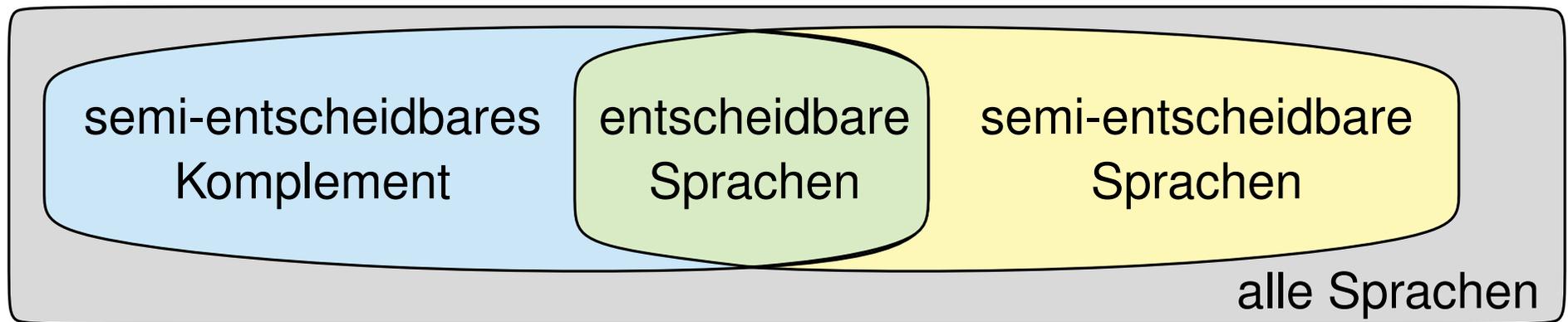
L^c semi-entscheidbar \rightsquigarrow Turingmaschine \mathcal{M}_2 akzeptiert bei $w \notin L$

Konstruiere 2-Band Turingmaschine \mathcal{M} :

- ▶ Schreibe w auf beide Bänder
- ▶ Führe \mathcal{M}_1 auf Band 1 und \mathcal{M}_2 auf Band 2 aus
- ▶ Akzeptiere wenn \mathcal{M}_1 akzeptiert, halte wenn \mathcal{M}_2 akzeptiert

\mathcal{M}
entscheidet
 L

Entscheidbarkeit – Überblick



Satz: L ist entscheidbar $\iff L$ und L^c sind semi-entscheidbar

Korollar: L ist entscheidbar $\iff L^c$ ist entscheidbar

Frage: L_1, L_2 entscheidbar $\implies L_1 \cup L_2, L_1 \cap L_2$ entscheidbar?

Entscheidbarkeit – Komplementbildung

Satz: L_1 und L_2 sind entscheidbar $\implies L_1 \setminus L_2$ ist entscheidbar

Entscheidbarkeit – Komplementbildung

Satz: L_1 und L_2 sind entscheidbar $\implies L_1 \setminus L_2$ ist entscheidbar

Beweis:

L_1 ist entscheidbar \rightsquigarrow Turingmaschine \mathcal{M}_1 akzeptiert bei $w \in L_1$
 L_2 ist entscheidbar \rightsquigarrow Turingmaschine \mathcal{M}_2 akzeptiert bei $w \in L_2$

Entscheidbarkeit – Komplementbildung

Satz: L_1 und L_2 sind entscheidbar $\implies L_1 \setminus L_2$ ist entscheidbar

Beweis:

L_1 ist entscheidbar \rightsquigarrow Turingmaschine \mathcal{M}_1 akzeptiert bei $w \in L_1$
 L_2 ist entscheidbar \rightsquigarrow Turingmaschine \mathcal{M}_2 akzeptiert bei $w \in L_2$

Konstruiere 2-Band Turingmaschine \mathcal{M} :

- ▶ Schreibe w auf beide Bänder
- ▶ Führe \mathcal{M}_1 auf Band 1 und \mathcal{M}_2 auf Band 2 aus
- ▶ Akzeptiere wenn \mathcal{M}_1 akzeptiert und \mathcal{M}_2 nicht akzeptiert

Entscheidbarkeit – Komplementbildung

Satz: L_1 und L_2 sind entscheidbar $\implies L_1 \setminus L_2$ ist entscheidbar

Beweis:

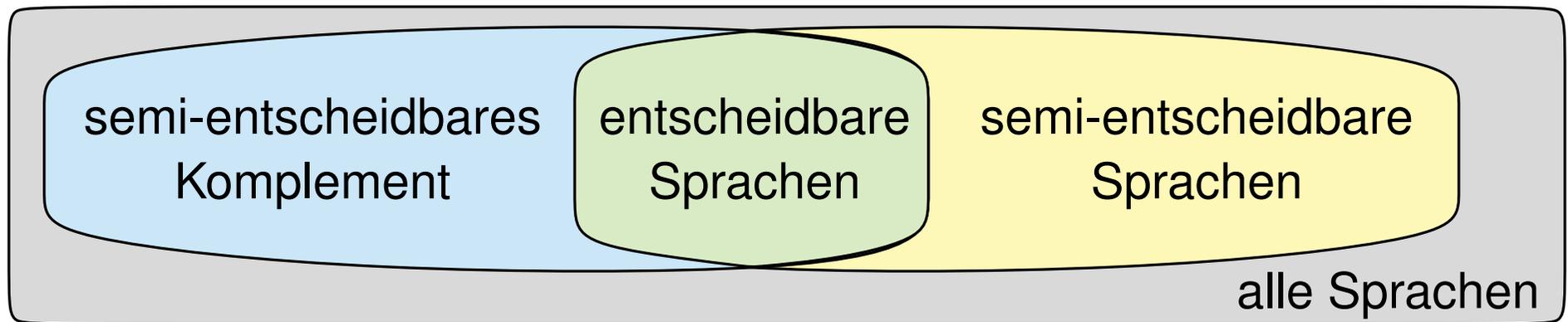
L_1 ist entscheidbar \rightsquigarrow Turingmaschine \mathcal{M}_1 akzeptiert bei $w \in L_1$
 L_2 ist entscheidbar \rightsquigarrow Turingmaschine \mathcal{M}_2 akzeptiert bei $w \in L_2$

Konstruiere 2-Band Turingmaschine \mathcal{M} :

- ▶ Schreibe w auf beide Bänder
- ▶ Führe \mathcal{M}_1 auf Band 1 und \mathcal{M}_2 auf Band 2 aus
- ▶ Akzeptiere wenn \mathcal{M}_1 akzeptiert und \mathcal{M}_2 nicht akzeptiert

\mathcal{M} entscheidet L

Entscheidbarkeit – Überblick



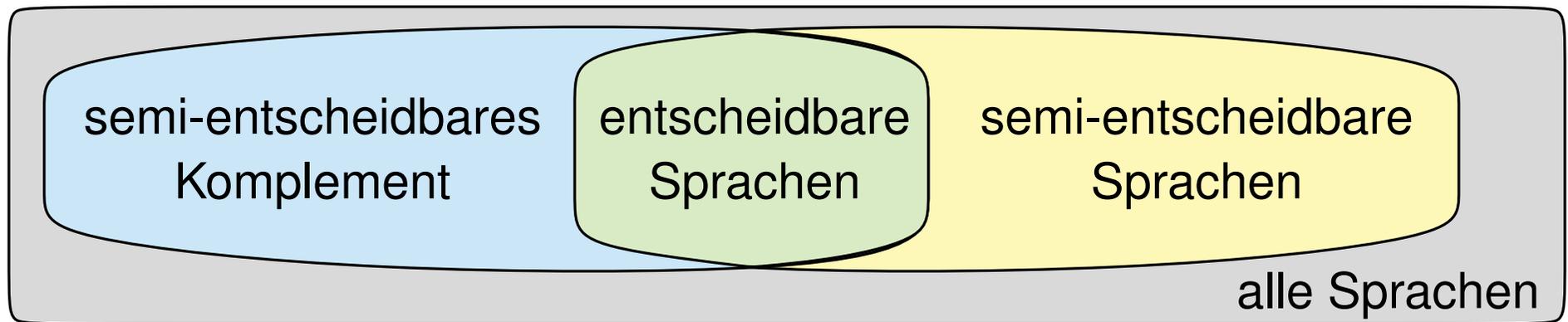
Satz: L ist entscheidbar $\iff L$ und L^c sind semi-entscheidbar

Korollar: L ist entscheidbar $\iff L^c$ ist entscheidbar

Satz: L_1 und L_2 sind entscheidbar $\implies L_1 \setminus L_2$ ist entscheidbar

Korollar: L_1, L_2 entscheidbar $\implies L_1 \cap L_2, L_1 \cup L_2$ entscheidbar

Entscheidbarkeit – Überblick



Satz: L ist entscheidbar $\iff L$ und L^c sind semi-entscheidbar

Korollar: L ist entscheidbar $\iff L^c$ ist entscheidbar

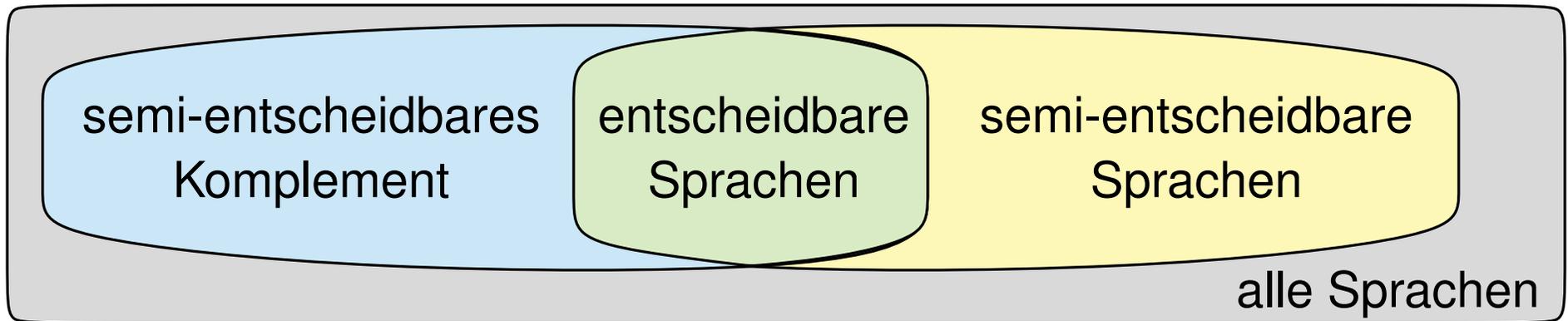
Satz: L_1 und L_2 sind entscheidbar $\implies L_1 \setminus L_2$ ist entscheidbar

Korollar: L_1, L_2 entscheidbar $\implies L_1 \cup L_2, L_1 \cap L_2$ entscheidbar

Frage: L_1, L_2 semi-entscheidbar $\implies L_1 \setminus L_2$ semi-entscheidbar?

Frage: L_1, L_2 semi-entscheidbar $\implies L_1 \cup L_2, L_1 \cap L_2$ semi-entsch.?

Entscheidbarkeit – Überblick



Satz: L ist entscheidbar $\iff L$ und L^c sind semi-entscheidbar

Korollar: L ist entscheidbar $\iff L^c$ ist entscheidbar

Satz: L_1 und L_2 sind entscheidbar $\implies L_1 \setminus L_2$ ist entscheidbar

Korollar: L_1, L_2 entscheidbar $\implies L_1 \cup L_2, L_1 \cap L_2$ entscheidbar

Frage: L_1, L_2 semi-entscheidbar  $L_1 \setminus L_2$ semi-entscheidbar?

Frage: L_1, L_2 semi-entscheidbar  $L_1 \cup L_2, L_1 \cap L_2$ semi-entsch.?

Entscheidbarkeit – Beispiele

Diagonalsprache

Gödelnummer:

injektive Abbildung von { alle Turingmaschinen } nach $\{0, 1\}^*$

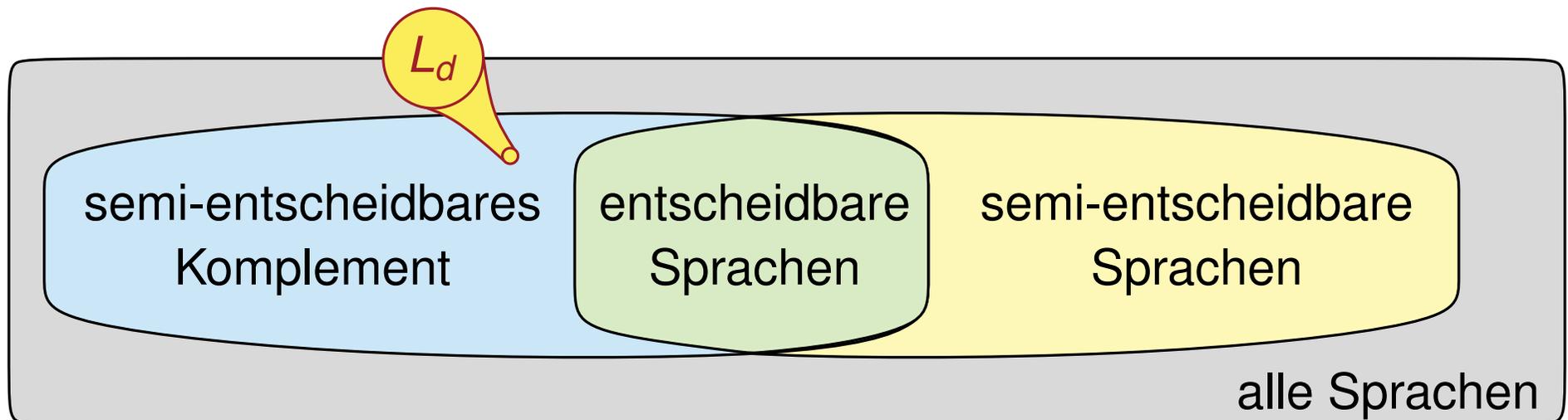
Notation: $\mathcal{M} \longrightarrow \langle \mathcal{M} \rangle \in \{0, 1\}^*$ $w \in \{0, 1\}^* \longrightarrow T_w$

Diagonalsprache:

$L_d = \{w \in \{0, 1\}^* \mid T_w \text{ akzeptiert } w \text{ nicht}\}$

VL: L_d ist nicht entscheidbar.

ÜB: L_d^c ist semi-entscheidbar.



Halteproblem

Sprache des Halteproblems:

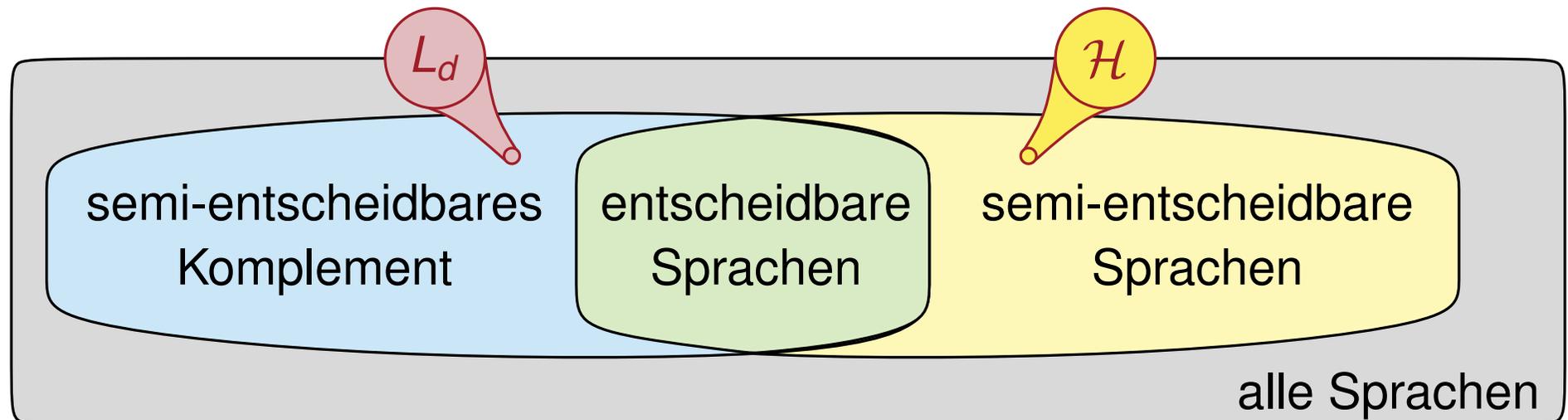
$$\mathcal{H} = \{wv \in \{0, 1\}^* \mid T_w \text{ hält bei Eingabe } v\}$$

universelle Turingmaschine:

bei Eingabe (w, v) wird T_w mit Eingabe v simuliert

VL: \mathcal{H} ist nicht entscheidbar. **ÜB:** \mathcal{H}^c ist nicht semi-entscheidbar.

universelle Turingmaschine: \mathcal{H} ist semi-entscheidbar.



Universelle Sprache

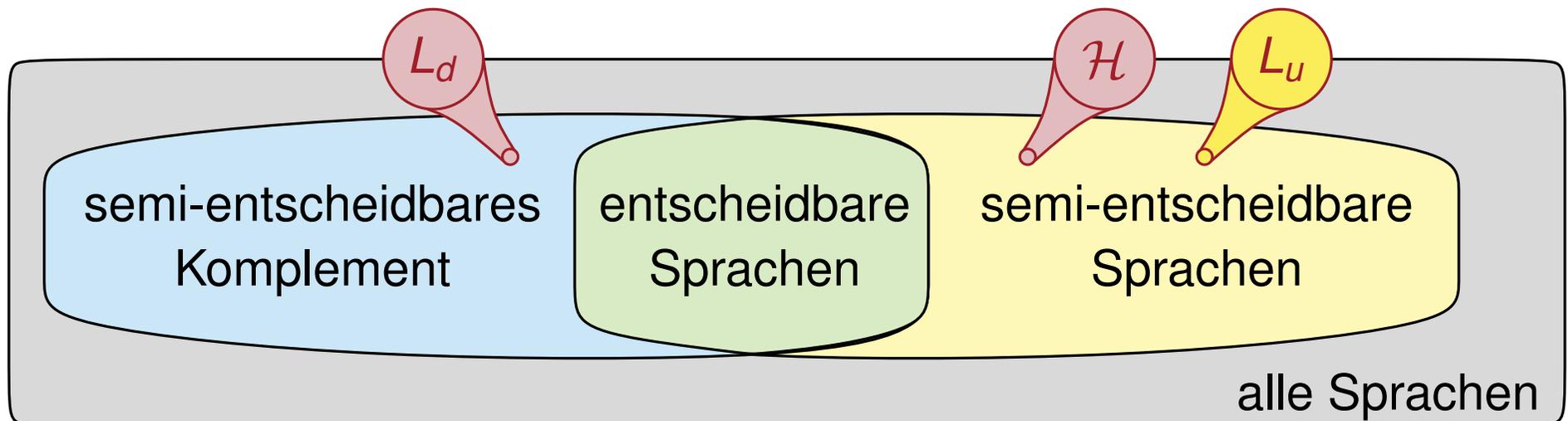
Die universelle Sprache:

$$L_u = \{wv \in \{0, 1\}^* \mid v \in L(T_w)\}$$
$$= \{wv \in \{0, 1\}^* \mid T_w \text{ hält und akzeptiert bei Eingabe } v\}$$

Vergleiche:

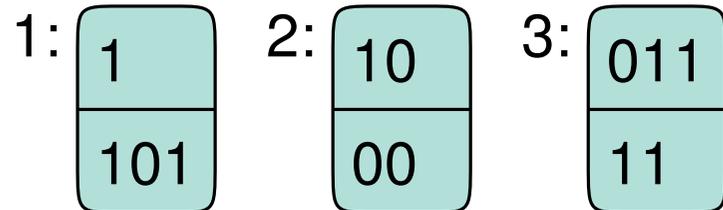
$$\mathcal{H} = \{wv \in \{0, 1\}^* \mid T_w \text{ hält bei Eingabe } v\}$$

VL: L_u ist nicht entscheidbar. **VL:** L_u ist semi-entscheidbar.



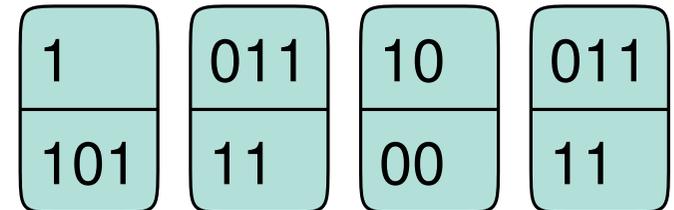
Post'sches Korrespondenzproblem

Das Problem Π :



Lösungen:

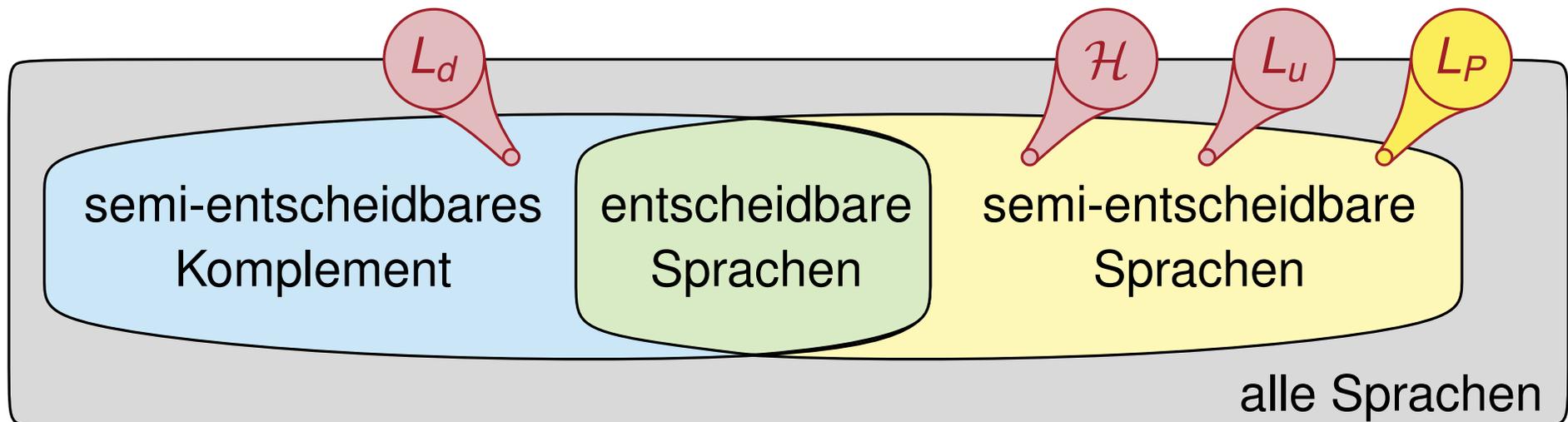
Indexfolge
1 3 2 3



Die zugehörige Sprache:

$$L_P := L[\Pi, s] = \{x \in \Sigma^* \mid x \text{ kodiert Ja-Beispiel von } \Pi \text{ unter } s\}$$

VL: L_P ist nicht entscheidbar, aber semi-entscheidbar.



Hilberts zehntes Problem

Frage von Hilbert: Sind die folgenden Sprachen entscheidbar?

$N = \{p \mid p \text{ ist ganzzahliges Polynom mit ganzzahliger Nullstelle}\}$

$N_1 = \{p \mid p \text{ ist ganzzahliges Polynom über } x \text{ mit ganzzahliger Nullstelle}\}$

Hilberts zehntes Problem

Frage von Hilbert: Sind die folgenden Sprachen entscheidbar?

$N = \{p \mid p \text{ ist ganzzahliges Polynom mit ganzzahliger Nullstelle}\}$

$N_1 = \{p \mid p \text{ ist ganzzahliges Polynom über } x \text{ mit ganzzahliger Nullstelle}\}$

Beobachtung: N und N' sind semi-entscheidbar.

Hilberts zehntes Problem

Frage von Hilbert: Sind die folgenden Sprachen entscheidbar?

$N = \{p \mid p \text{ ist ganzzahliges Polynom mit ganzzahliger Nullstelle}\}$

$N_1 = \{p \mid p \text{ ist ganzzahliges Polynom über } x \text{ mit ganzzahliger Nullstelle}\}$

Beobachtung: N und N' sind semi-entscheidbar.

Idee für N_1 : Überprüfe ob die Folge $0, 1, -1, 2, -2, \dots$ eine Nullstelle von p enthält.

Hilberts zehntes Problem

Frage von Hilbert: Sind die folgenden Sprachen entscheidbar?

$N = \{p \mid p \text{ ist ganzzahliges Polynom mit ganzzahliger Nullstelle}\}$

$N_1 = \{p \mid p \text{ ist ganzzahliges Polynom über } x \text{ mit ganzzahliger Nullstelle}\}$

Beobachtung: N und N' sind semi-entscheidbar.

Idee für N_1 : Überprüfe ob die Folge $0, 1, -1, 2, -2, \dots$ eine Nullstelle von p enthält.

- Verfahren terminiert nur, wenn p eine ganzzahlige Nullstelle hat.
- Einschränkung auf ein bestimmtes Intervall möglich

Hilberts zehntes Problem

Frage von Hilbert: Sind die folgenden Sprachen entscheidbar?

$N = \{p \mid p \text{ ist ganzzahliges Polynom mit ganzzahliger Nullstelle}\}$

$N_1 = \{p \mid p \text{ ist ganzzahliges Polynom über } x \text{ mit ganzzahliger Nullstelle}\}$

Beobachtung: N und N' sind semi-entscheidbar.

Idee für N_1 : Überprüfe ob die Folge $0, 1, -1, 2, -2, \dots$ eine Nullstelle von p enthält.

- Verfahren terminiert nur, wenn p eine ganzzahlige Nullstelle hat.
- Einschränkung auf ein bestimmtes Intervall möglich

$\rightsquigarrow N_1$ ist entscheidbar

Hilberts zehntes Problem

Frage von Hilbert: Sind die folgenden Sprachen entscheidbar?

$N = \{p \mid p \text{ ist ganzzahliges Polynom mit ganzzahliger Nullstelle}\}$

$N_1 = \{p \mid p \text{ ist ganzzahliges Polynom über } x \text{ mit ganzzahliger Nullstelle}\}$

Beobachtung: N und N' sind semi-entscheidbar.

Idee für N_1 : Überprüfe ob die Folge $0, 1, -1, 2, -2, \dots$ eine Nullstelle von p enthält.

- Verfahren terminiert nur, wenn p eine ganzzahlige Nullstelle hat.
- Einschränkung auf ein bestimmtes Intervall möglich

$\rightsquigarrow N_1$ ist entscheidbar

Dieses Intervall existiert für N nicht.

Hilberts zehntes Problem

Frage von Hilbert: Sind die folgenden Sprachen entscheidbar?

$N = \{p \mid p \text{ ist ganzzahliges Polynom mit ganzzahliger Nullstelle}\}$

$N_1 = \{p \mid p \text{ ist ganzzahliges Polynom über } x \text{ mit ganzzahliger Nullstelle}\}$

Beobachtung: N und N' sind semi-entscheidbar.

Idee für N_1 : Überprüfe ob die Folge $0, 1, -1, 2, -2, \dots$ eine Nullstelle von p enthält.

- Verfahren terminiert nur, wenn p eine ganzzahlige Nullstelle hat.
- Einschränkung auf ein bestimmtes Intervall möglich

$\rightsquigarrow N_1$ ist entscheidbar

Dieses Intervall existiert für N nicht.

Matijasevič: N ist nicht entscheidbar

Hilberts zehntes Problem

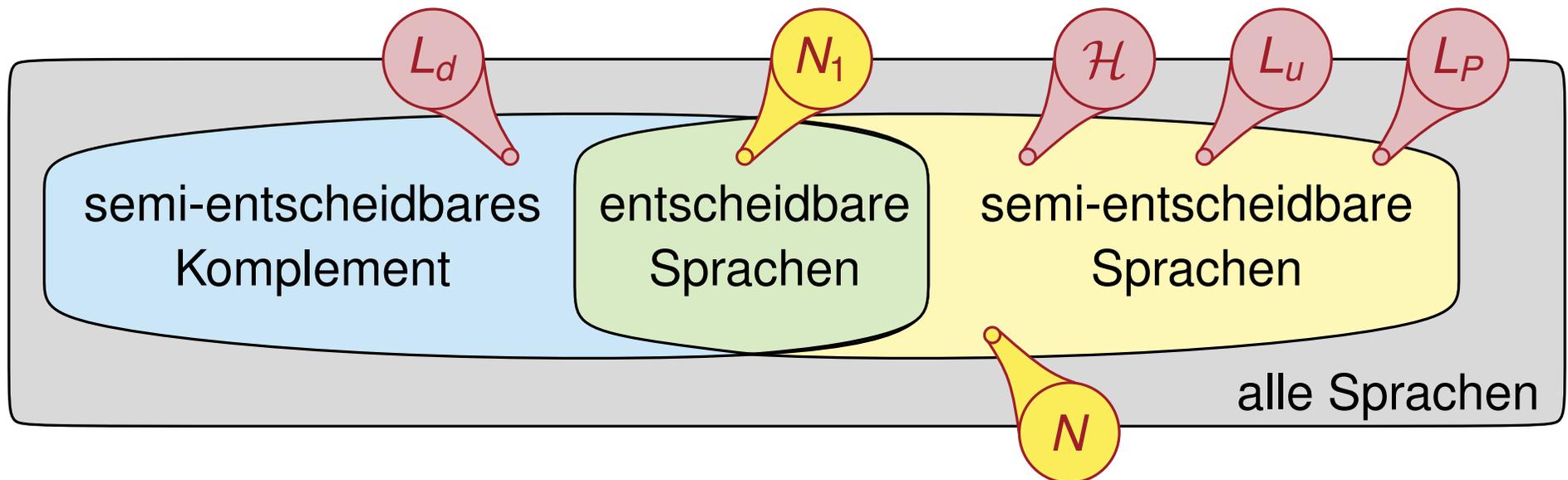
Polynome mit ganzzahligen Nullstellen:

$N = \{p \mid p \text{ ist ganzzahliges Polynom mit ganzzahliger Nullstelle}\}$

$N_1 = \{p \mid p \text{ ist ganzzahliges Polynom über } x \text{ mit ganzzahliger Nullstelle}\}$

letzte Folie: N_1 ist entscheidbar, N ist semi-entscheidbar.

Matijasevič: N ist nicht entscheidbar.

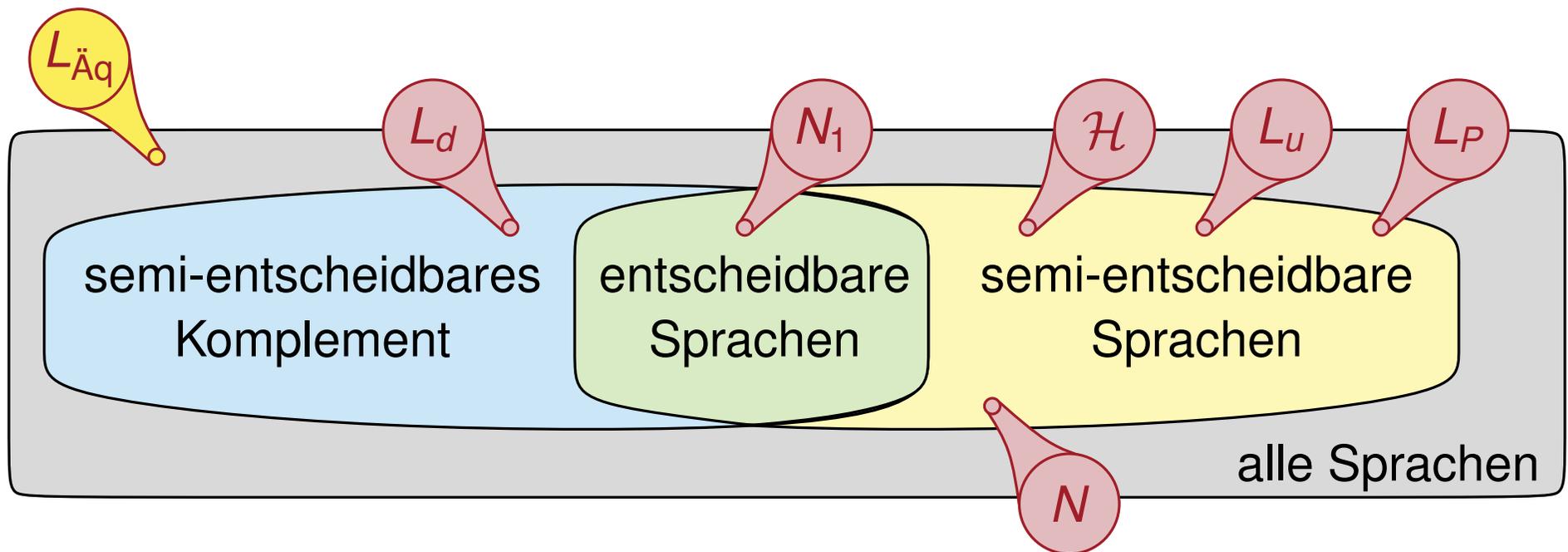


Äquivalenzproblem

Äquivalenz von Turingmaschinen:

$$L_{\text{Äq}} := \{w\#v \in \{0, 1, \#\}^* \mid L(T_w) = L(T_v)\}$$

Idee: Sowohl $L_{\text{Äq}}$ als auch $L_{\text{Äq}}^c$ sind Erweiterungen des Halteproblems.



Entscheidbarkeit – Werkzeugkasten

Satz von Rice

Satz von Rice: Sei S eine nicht-triviale Teilmenge von berechenbaren Funktionen. Dann ist

$$L(S) := \{w \in \{0, 1\}^* \mid T_w \text{ realisiert Funktion aus } S\}$$
nicht entscheidbar.

Beweisidee: Zeige das $L(S)$ Erweiterung von \mathcal{H}_ε ist.

Satz von Rice

Satz von Rice: Sei S eine nicht-triviale Teilmenge von berechenbaren Funktionen. Dann ist

$$L(S) := \{w \in \{0, 1\}^* \mid T_w \text{ realisiert Funktion aus } S\}$$
nicht entscheidbar.

Beweisidee: Zeige das $L(S)$ Erweiterung von \mathcal{H}_ε ist.

Beispiel 1: $L_1 = \{w \in \{0, 1\}^* \mid T_w \text{ berechnet } 17 \text{ bei Eingabe } 42\}$

- ▶ L_1 ist nicht berechenbar nach Satz von Rice.

Satz von Rice

Satz von Rice: Sei S eine nicht-triviale Teilmenge von berechenbaren Funktionen. Dann ist

$$L(S) := \{w \in \{0, 1\}^* \mid T_w \text{ realisiert Funktion aus } S\}$$
 nicht entscheidbar.

Beweisidee: Zeige das $L(S)$ Erweiterung von \mathcal{H}_ε ist.

Beispiel 1: $L_1 = \{w \in \{0, 1\}^* \mid T_w \text{ berechnet } 17 \text{ bei Eingabe } 42\}$

- ▶ L_1 ist nicht berechenbar nach Satz von Rice.

Beispiel 2: $L_2 = \{w \in \{0, 1\}^* \mid T_w \text{ macht } 17 \text{ Schritte bei Eingabe } 42\}$

- ▶ Keine Aussage über L_2 durch Satz von Rice.
- ▶ Ist L_2 berechenbar?

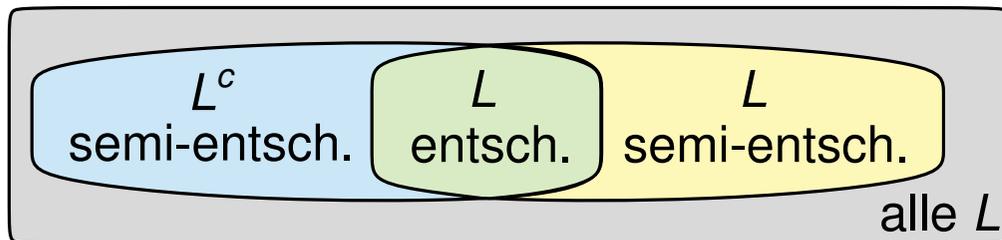
Zusammenfassung (Werkzeugkasten)

L ist entscheidbar

- ▶ Konstruiere TM oder NTM die L entscheidet.
- ▶ Zeige L, L^c semi-entscheidbar.

L ist semi-entscheidbar

- ▶ Konstruiere TM oder NTM die L erkennt.



L ist **nicht** entscheidbar

- ▶ Benutze Satz von Rice.
- ▶ Reduziere nicht entscheidbare Sprache N auf L .
- ▶ Konstruiere (N)TM für N aus (N)TM für L .

L ist **nicht** semi-entscheidbar

- ▶ Zeige L nicht entscheidbar und L^c semi-entscheidbar.
- ▶ Reduziere nicht semi-entscheidbare Sprache auf L .