Theoretische Grundlagen der Informatik Übung

3. Übungstermin · 15. November

Marcel Radermacher

INSTITUT FÜR THEORETISCHE INFORMATIK · PROF. DR. DOROTHEA WAGNER

Gliederung

Organisatorisches

- Tutorium 16, gehalten von Jan Ellmers, findet am 16.11 in Seminarraum **SR -108** statt (einmalige Verlegung).
  Uhrzeit wie immer: 09:45 - 11:15 Uhr
- Bitte Deckblattgenerator für Übungsblätter benutzen:

  https://webinscribe.ira.uka.de/deckblatt/index.php?course=10588

Inhalt

- Verallgemeinertes Pumping-Lemma
- Nerode-Relation
- Turing-Maschinen
- Entscheidbarkeit

Verallgemeinertes Pumping-Lemma

Pumping-Lemma

Sei L eine reguläre Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, so dass für jedes Wort $w \in L$ mit $|w| > n$ eine Darstellung

$$w = uvx \text{ mit } |uv| \leq n \text{ und } v \neq \varepsilon,$$

existiert, bei der auch $uv^ix \in L$ ist für alle $i \in \mathbb{N}_0$.

Pumping-Lemma

Zeigen Sie für die Sprache $L = \{a^i b^{j^2} \in \{a, b\}^\star \mid i, j \in \mathbb{N}\}$, dass die Sprache nicht regulär ist.

Sei L eine reguläre Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, so dass für jedes Wort $w \in L$ mit $|w| > n$ eine Darstellung

$$w = uvx \text{ mit } |uv| \leq n \text{ und } v \neq \varepsilon,$$

existiert, bei der auch $uv^i x \in L$ ist für alle $i \in \mathbb{N}_0$.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Pumping-Lemma

Zeigen Sie für die Sprache $L = \{a^i b^{j^2} \in \{a, b\}^\star \mid i, j \in \mathbb{N}\}$, dass die Sprache nicht regulär ist.

Pumping-Lemma ist erfüllt (siehe Vorlesung)

Sei L eine reguläre Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, so dass für jedes Wort $w \in L$ mit $|w| > n$ eine Darstellung

$$w = uvx \text{ mit } |uv| \leq n \text{ und } v \neq \varepsilon,$$

existiert, bei der auch $uv^ix \in L$ ist für alle $i \in \mathbb{N}_0$.

Verallgemeinertes Pumping-Lemma

Zeigen Sie für die Sprache $L = \{a^i b^{j^2} \in \{a,b\}^\star \mid i,j \in \mathbb{N}\}$, dass die Sprache nicht regulär ist.

Verallgemeinertes Pumping-Lemma

Zeigen Sie für die Sprache $L = \{a^i b^{j^2} \in \{a, b\}^\star \mid i, j \in \mathbb{N}\}$, dass die Sprache nicht regulär ist.

Sei L eine reguläre Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, so dass für jedes Wort $w \in L$ mit $|w| \geq n$ und jede Darstellung $w = tyx$ mit $|y| = n$ gilt:

Für jedes Teilwort y existiert eine Darstellung $y = uvz$ mit $v \neq \varepsilon$ bei der auch $tuv^izx \in L$ für alle $i \in \mathbb{N}_0$.

Verallgemeinertes Pumping-Lemma

Zeigen Sie für die Sprache $L = \{a^i b^{j^2} \in \{a, b\}^\star \mid i, j \in \mathbb{N}\}$, dass die Sprache nicht regulär ist.

Annahme: L ist regulär.

Zeige für $w = a^n b^{n^2}$, dass das VPL nicht erfüllt ist.

n beliebig

Sei L eine reguläre Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, so dass für jedes Wort $w \in L$ mit $|w| \geq n$ und jede Darstellung $w = tyx$ mit $|y| = n$ gilt:

Für jedes Teilwort y existiert eine Darstellung $y = uvz$ mit $v \neq \varepsilon$ bei der auch $tuv^izx \in L$ für alle $i \in \mathbb{N}_0$.

Verallgemeinertes Pumping-Lemma

Zeigen Sie für die Sprache $L = \{a^i b^{j^2} \in \{a,b\}^\star \mid i,j \in \mathbb{N}\}$, dass die Sprache nicht regulär ist.

Annahme: L ist regulär.

Zeige für $w = a^n b^{n^2}$, dass das VPL nicht erfüllt ist.

n beliebig

Wähle $t = a^n$, $yx = b^{n^2}$, also $y = b^n$, $x = b^{n^2-n}$

eine Darstellung

Sei L eine reguläre Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, so dass für jedes Wort $w \in L$ mit $|w| \geq n$ und jede Darstellung $w = tyx$ mit $|y| = n$ gilt:

Für jedes Teilwort y existiert eine Darstellung $y = uvz$ mit $v \neq \varepsilon$ bei der auch $tuv^izx \in L$ für alle $i \in \mathbb{N}_0$.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Verallgemeinertes Pumping-Lemma

Zeigen Sie für die Sprache $L = \{a^i b^{i^2} \in \{a,b\}^\star \mid i,j \in \mathbb{N}\}$, dass die Sprache nicht regulär ist.

Wähle $t = a^n$, $y = b^n$, $x = b^{n^2-n}$

$y = uvz$, $u = b^j$, $v = b^k$, $z = b^{n-j-k}$, $k > 0$

beliebige Darstellung

Sei L eine reguläre Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, so dass für jedes Wort $w \in L$ mit $|w| \geq n$ und jede Darstellung $w = tyx$ mit $|y| = n$ gilt:

Für jedes Teilwort y existiert eine Darstellung $y = uvz$ mit $v \neq \varepsilon$ bei der auch $tuv^i zx \in L$ für alle $i \in \mathbb{N}_0$.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Verallgemeinertes Pumping-Lemma

Zeigen Sie für die Sprache $L = \{a^i b^{j^2} \in \{a, b\}^\star \mid i, j \in \mathbb{N}\}$, dass die Sprache nicht regulär ist.

Wähle $t = a^n$, $y = b^n$, $x = b^{n^2-n}$

$y = uvz$, $u = b^j$, $v = b^k$, $z = b^{n-j-k}$, $k > 0$

beliebige Darstellung

$w_k = tuv^0zx = a^n b^j (b^k)^0 b^{n-j-k} b^{n^2-n} = a^n b^{n^2-k}$

ein i, $i = 0$

Sei L eine reguläre Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, so dass für jedes Wort $w \in L$ mit $|w| \geq n$ und jede Darstellung $w = tyx$ mit $|y| = n$ gilt:

Für jedes Teilwort y existiert eine Darstellung $y = uvz$ mit $v \neq \varepsilon$ bei der auch $tuv^izx \in L$ für alle $i \in \mathbb{N}_0$.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Verallgemeinertes Pumping-Lemma

Zeigen Sie für die Sprache $L = \{a^i b^{j^2} \in \{a, b\}^\star \mid i, j \in \mathbb{N}\}$, dass die Sprache nicht regulär ist.

Wähle $t = a^n$, $y = b^n$, $x = b^{n^2 - n}$

$y = uvz$, $u = b^j$, $v = b^k$, $z = b^{n-j-k}$, $k > 0$

beliebige Darstellung

$w_k = tuv^0zx = a^n b^j (b^k)^0 b^{n-j-k} b^{n^2-n} = a^n b^{n^2-k}$

ein i, $i = 0$

w_k ist für jedes $k > 0$ kein Element von L. ↯

Sei L eine reguläre Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, so dass für jedes Wort $w \in L$ mit $|w| \geq n$ und jede Darstellung $w = tyx$ mit $|y| = n$ gilt:

Für jedes Teilwort y existiert eine Darstellung $y = uvz$ mit $v \neq \varepsilon$ bei der auch $tuv^izx \in L$ für alle $i \in \mathbb{N}_0$.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Nerode-Relation

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Rechtsinvariante Relationen

Was sind rechtsinvariante Relationen?

Rechtsinvariante Relationen

Was sind rechtsinvariante Relationen?

Eine Äquivalenzrelation R über Σ^* heißt *rechtsinvariante Relation*, wenn

für alle $x, y \in \Sigma^*$ gilt: falls $x \; R \; y$ so gilt auch $xz \; R \; yz$ für alle $z \in \Sigma^*$.

Den *Index* von R bezeichnen wir mit *ind(R)*; er ist die Anzahl der Äquivalenzklassen von Σ^* bezüglich R.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Nerode-Relation

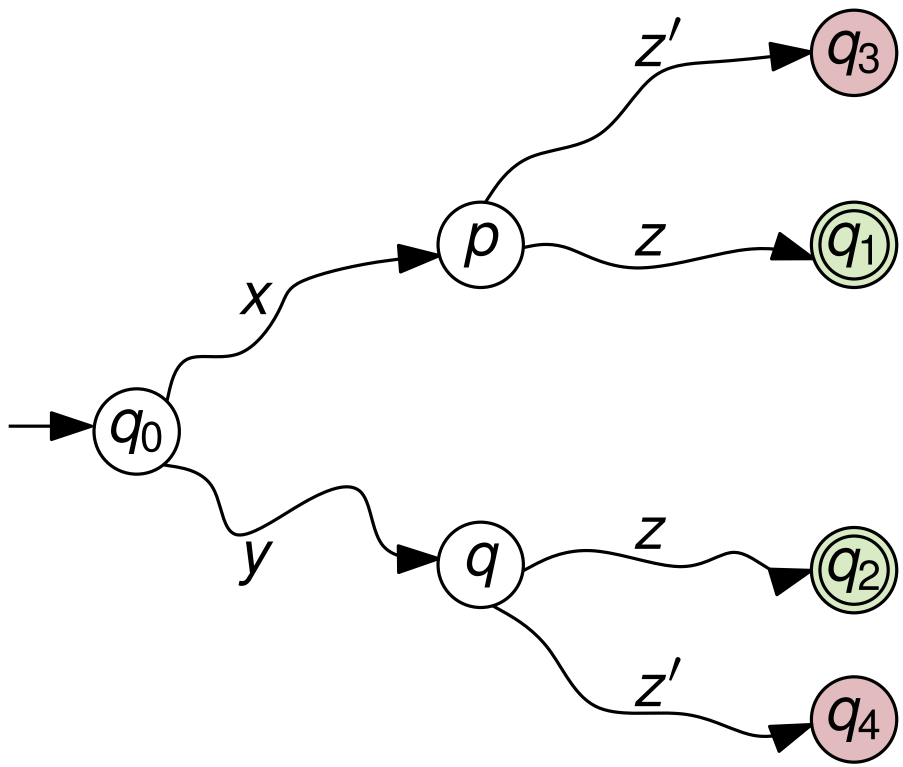
Für eine Sprache $L \subseteq \Sigma^*$ ist die *Nerode–Relation* R_L definiert durch: für

$$x, y \in \Sigma^* \text{ ist } x \; R_L \; y$$

genau dann wenn

$$(xz \in L \Leftrightarrow yz \in L) \text{ für alle } z \in \Sigma^* \text{ gilt.}$$

Sei $x R_L y$ für $x, y \in \Sigma^\star$ und L regulär.



p und q sind äquivalent
$\rightarrow$ selbes Akzeptanzverhalten.

Nerode-Relation

Für eine Sprache $L \subseteq \Sigma^*$ ist die *Nerode–Relation* R_L definiert durch: für

$x, y \in \Sigma^*$ ist $x \; R_L \; y$

genau dann wenn

$(xz \in L \Leftrightarrow yz \in L)$ für alle $z \in \Sigma^*$ gilt.

Satz von Nerode

Die folgenden Aussagen sind äquivalent:

1. $L \subseteq \Sigma^*$ wird von einem deterministischen endlichen Automaten erkannt bzw. akzeptiert.
2. L ist die Vereinigung von (einigen) Äquivalenzklassen einer rechtsinvarianten Äquivalenzrelation mit endlichem Index.
3. Die Nerode–Relation hat endlichen Index.

Äquivalenzklassen

Geben Sie für die folgenden Sprachen jeweils die Äquivalenzklassen bezüglich der Nerode-Relation R_L an. Geben Sie außerdem jeweils den Minimalautomaten an.

Äquivalenzklassen

Geben Sie für die folgenden Sprachen jeweils die Äquivalenzklassen bezüglich der Nerode-Relation R_L an. Geben Sie außerdem jeweils den Minimalautomaten an.

$L_1 = \{w \in \{0,1\}^\star \mid |w|_1 \text{ ist gerade.}\}$

Äquivalenzklassen

Geben Sie für die folgenden Sprachen jeweils die Äquivalenzklassen bezüglich der Nerode-Relation R_L an. Geben Sie außerdem jeweils den Minimalautomaten an.

$L_1 = \{w \in \{0,1\}^\star \mid |w|_1 \text{ ist gerade.}\}$

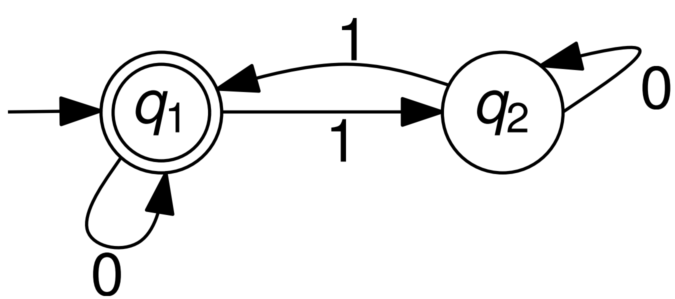
$$[\varepsilon] = \{w \in \{0,1\}^\star \mid |w|_1 \text{ ist gerade.}\}$$
$$[1] = \{w \in \{0,1\}^\star \mid |w|_1 \text{ ist ungerade.}\}$$

Äquivalenzklassen

Geben Sie für die folgenden Sprachen jeweils die Äquivalenzklassen bezüglich der Nerode-Relation R_L an. Geben Sie außerdem jeweils den Minimalautomaten an.

$L_1 = \{w \in \{0,1\}^\star \mid |w|_1 \text{ ist gerade.}\}$

$$[\varepsilon] = \{w \in \{0,1\}^\star \mid |w|_1 \text{ ist gerade.}\}$$
$$[1] = \{w \in \{0,1\}^\star \mid |w|_1 \text{ ist ungerade.}\}$$



Nicht-Regulär

Zeigen Sie für die Sprache $L = \{a^n b^{n^2} \in \{a,b\}^\star \mid n \in \mathbb{N}\}$, dass die Sprache nicht regulär ist.

Nicht-Regulär

Zeigen Sie für die Sprache $L = \{a^n b^{n^2} \in \{a,b\}^\star \mid n \in \mathbb{N}\}$, dass die Sprache nicht regulär ist.

Betrachte $w_i = a^i$ und $w_j = a^j$ mit $i < j$.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Nicht-Regulär

Zeigen Sie für die Sprache $L = \{a^n b^{n^2} \in \{a,b\}^\star \mid n \in \mathbb{N}\}$, dass die Sprache nicht regulär ist.

Betrachte $w_i = a^i$ und $w_j = a^j$ mit $i < j$.

Zeige: w_i und w_j liegen in unterschiedlichen Äquivalenzklassen bzgl. R_L

$\Rightarrow$ Nerode-Relation hat unendlich viele Äquivalenzklassen.

$\Rightarrow$ L ist nicht regulär.

Satz von Nerode

Die folgenden Aussagen sind äquivalent:

1. $L \subseteq \Sigma^*$ wird von einem deterministischen endlichen Automaten erkannt bzw. akzeptiert.
2. L ist die Vereinigung von (einigen) Äquivalenzklassen einer rechtsinvarianten Äquivalenzrelation mit endlichem Index.
3. Die Nerode–Relation hat endlichen Index.

Nicht-Regulär

Zeigen Sie für die Sprache $L = \{a^n b^{n^2} \in \{a, b\}^\star \mid n \in \mathbb{N}\}$, dass die Sprache nicht regulär ist.

Betrachte $w_i = a^i$ und $w_j = a^j$ mit $i < j$.

Zeige: w_i und w_j liegen in unterschiedlichen Äquivalenzklassen bzgl. R_L

Annahme: w_i und w_j liegen in derselben Äquivalenzklasse.

Nicht-Regulär

Zeigen Sie für die Sprache $L = \{a^n b^{n^2} \in \{a,b\}^\star \mid n \in \mathbb{N}\}$, dass die Sprache nicht regulär ist.

Betrachte $w_i = a^i$ und $w_j = a^j$ mit $i < j$.

Zeige: w_i und w_j liegen in unterschiedlichen Äquivalenzklassen bzgl. R_L

Annahme: w_i und w_j liegen in derselben Äquivalenzklasse.

$$w_i z \in L \Leftrightarrow w_j z \in L \text{ für alle } z \in \Sigma^\star$$

Für eine Sprache $L \subseteq \Sigma^*$ ist die *Nerode–Relation* R_L definiert durch: für

$x, y \in \Sigma^*$ ist $x \; R_L \; y$

genau dann wenn

$(xz \in L \Leftrightarrow yz \in L)$ für alle $z \in \Sigma^*$ gilt.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Nicht-Regulär

Zeigen Sie für die Sprache $L = \{a^n b^{n^2} \in \{a,b\}^\star \mid n \in \mathbb{N}\}$, dass die Sprache nicht regulär ist.

Betrachte $w_i = a^i$ und $w_j = a^j$ mit $i < j$.

Zeige: w_i und w_j liegen in unterschiedlichen Äquivalenzklassen bzgl. R_L

Annahme: w_i und w_j liegen in derselben Äquivalenzklasse.

$$w_i z \in L \Leftrightarrow w_j z \in L \text{ für alle } z \in \Sigma^\star$$

Insbesondere muss dann für $z = b^{i^2}$ gelten

$$w_i b^{i^2} \in L \Leftrightarrow w_j b^{i^2} \in L$$

Nicht-Regulär

Zeigen Sie für die Sprache $L = \{a^n b^{n^2} \in \{a,b\}^\star \mid n \in \mathbb{N}\}$, dass die Sprache nicht regulär ist.

Betrachte $w_i = a^i$ und $w_j = a^j$ mit $i < j$.

Zeige: w_i und w_j liegen in unterschiedlichen Äquivalenzklassen bzgl. R_L

Annahme: w_i und w_j liegen in derselben Äquivalenzklasse.

$$w_i z \in L \Leftrightarrow w_j z \in L \text{ für alle } z \in \Sigma^\star$$

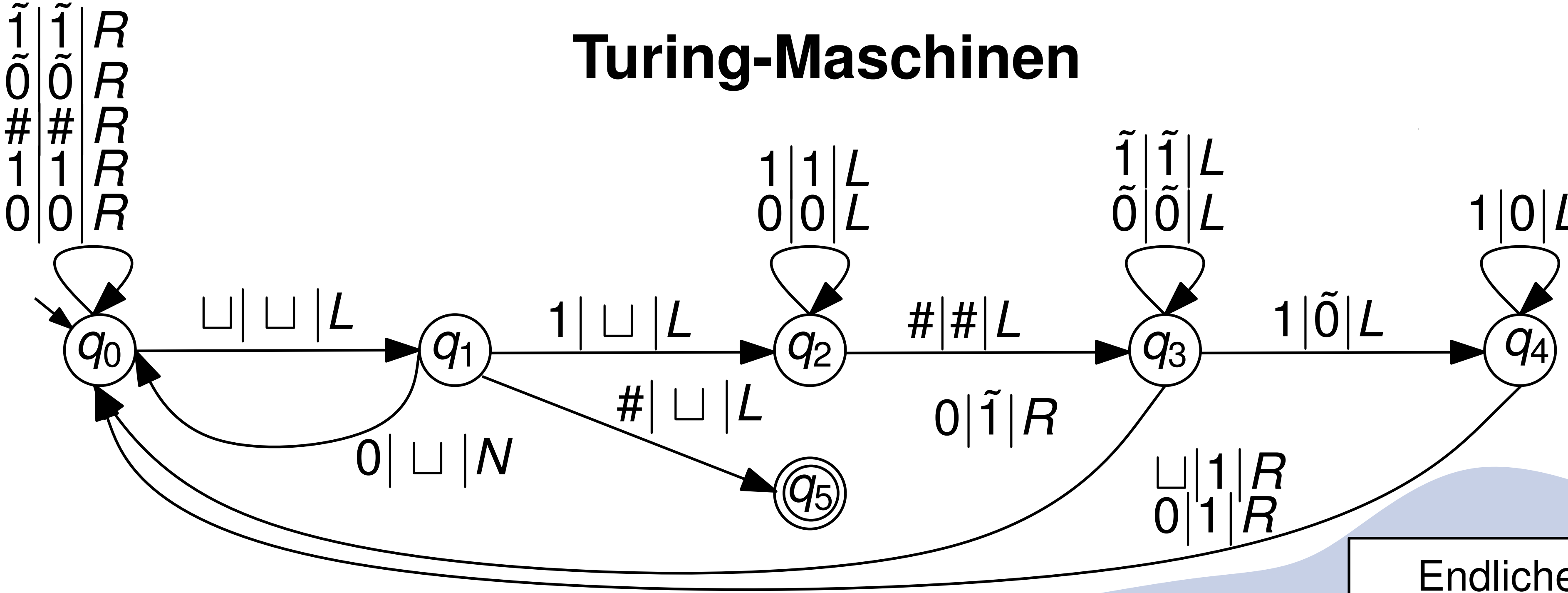
Insbesondere muss dann für $z = b^{i^2}$ gelten

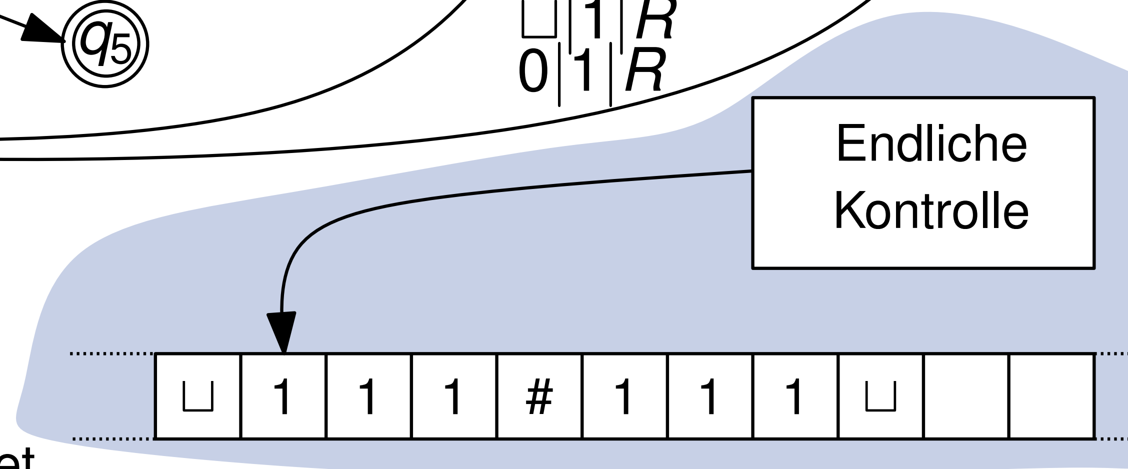
$$w_i b^{i^2} \in L \Leftrightarrow w_j b^{i^2} \in L$$

Es gilt $w_i b^{i^2} \in L$ und somit $w_j b^{i^2} \in L$. Wegen $j > i$ gilt aber $w_j b^{i^2} \notin L$. ↯

Turing- Maschinen

Turing-Maschinen

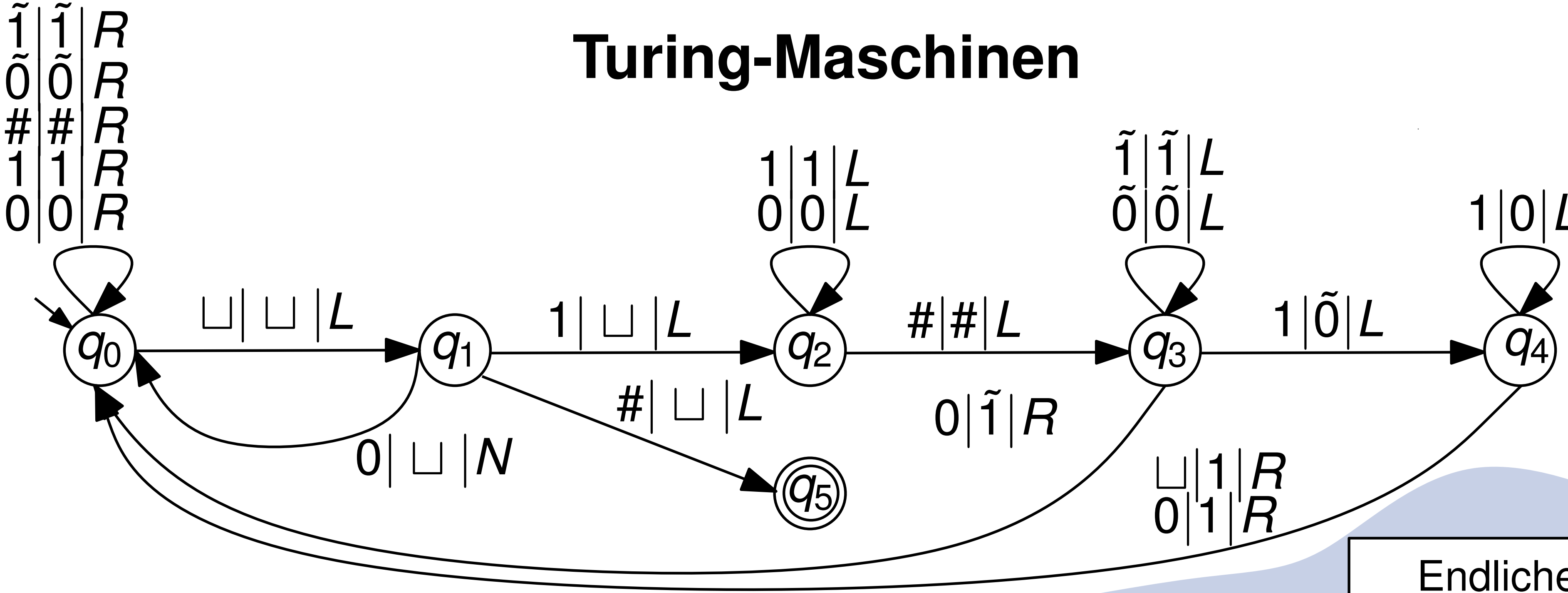


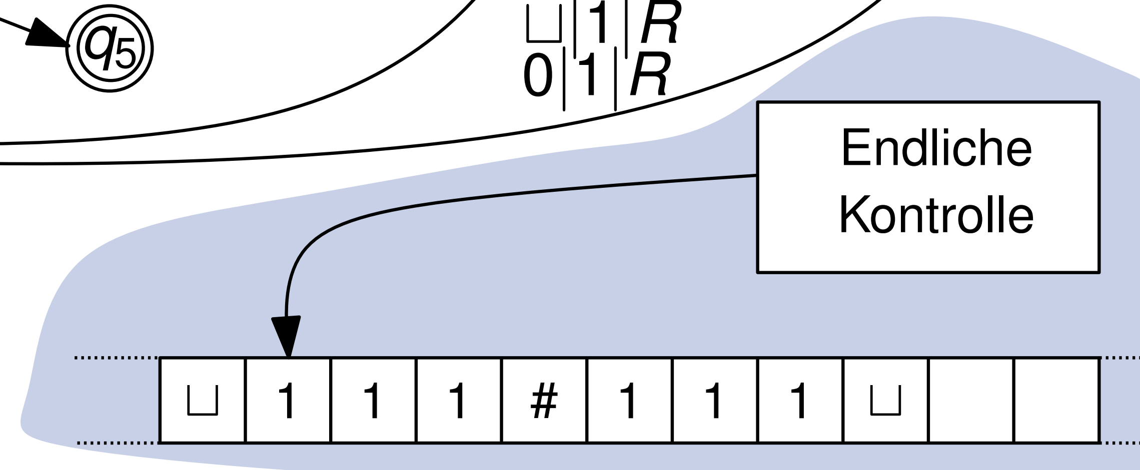


- Q, einer endlicher Zustandsmenge,
- Σ, einem endlichen Eingabealphabet,
- $\sqcup$, einem Blanksymbol mit $\sqcup \notin \Sigma$,
- Γ, einem endlichen Bandalphabet mit $\Sigma \cup \{\sqcup\} \subseteq \Gamma$,
- $s \in Q$, einem Startzustand,
- $\delta\colon Q \times \Gamma \to Q \times \Gamma \times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

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Turing-Maschinen

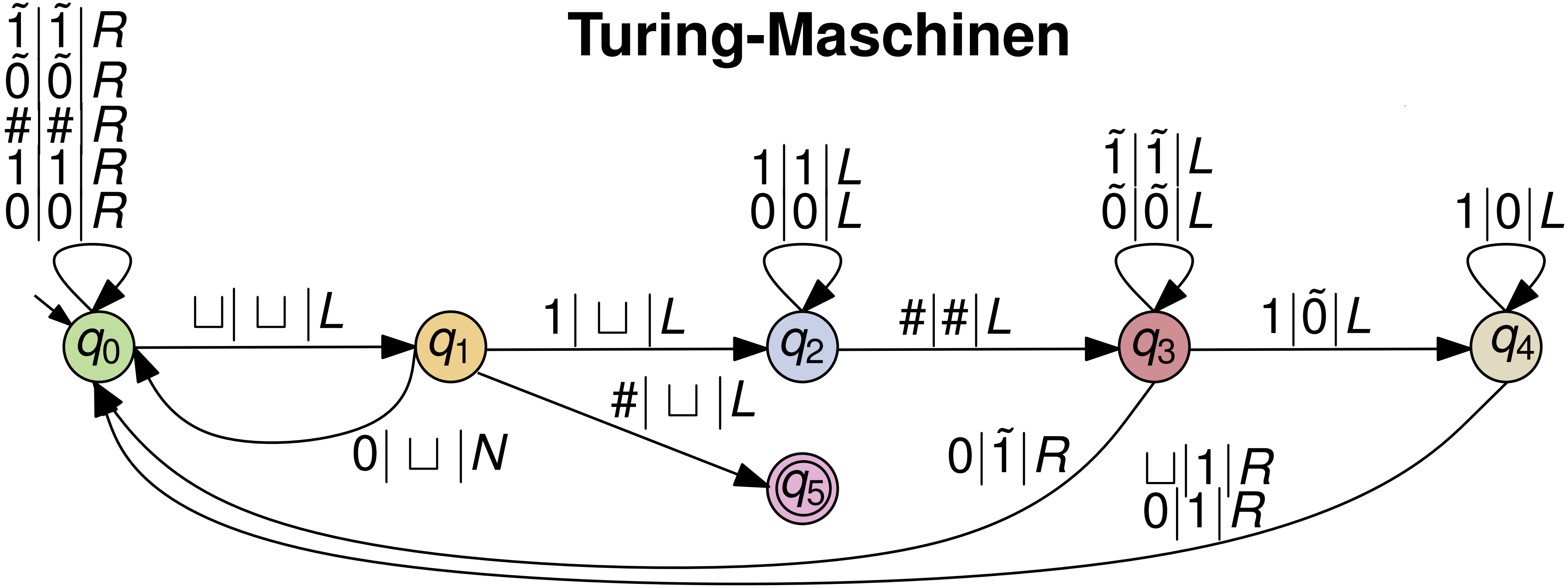




- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$,
- $\Sigma = \{0, 1, \#\}$,
- $\Gamma = \Sigma \cup \{\sqcup, \tilde{0}, \tilde{1}\}$,
- Startzustand q_0,
- Übergangsfunktion $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R, N\}$
- Endzustand $q_5 \subseteq Q$

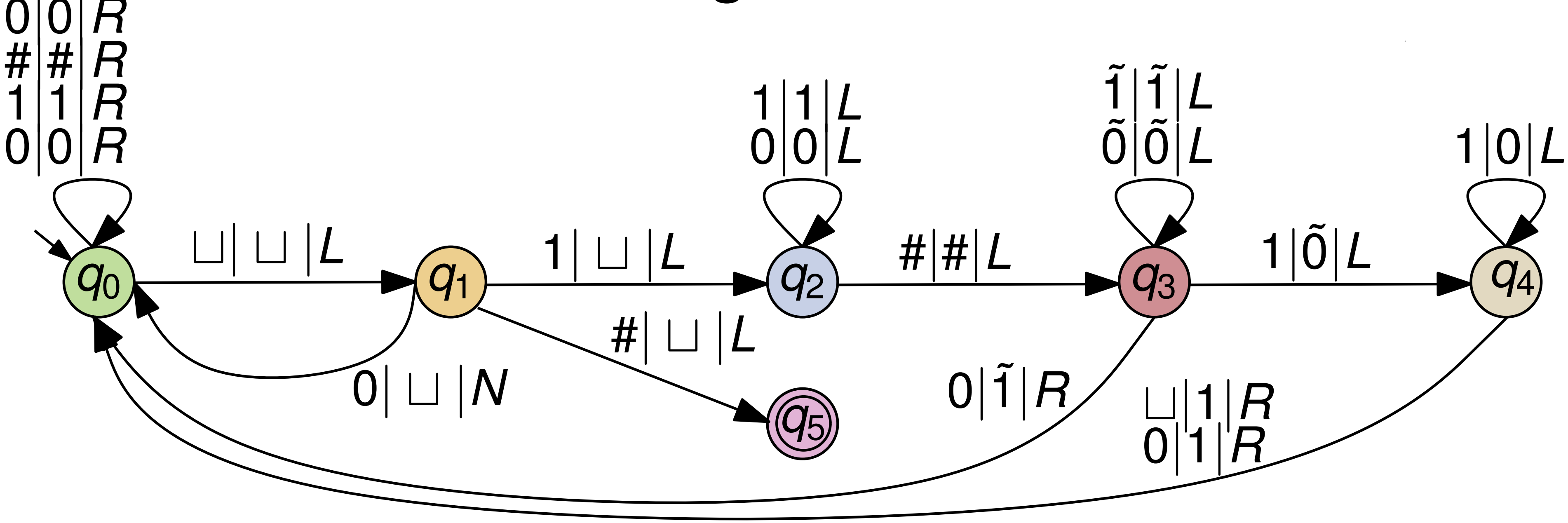
Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Turing-Maschinen



$\sqcup(q_0)1 1 1\#1 1 1\sqcup$

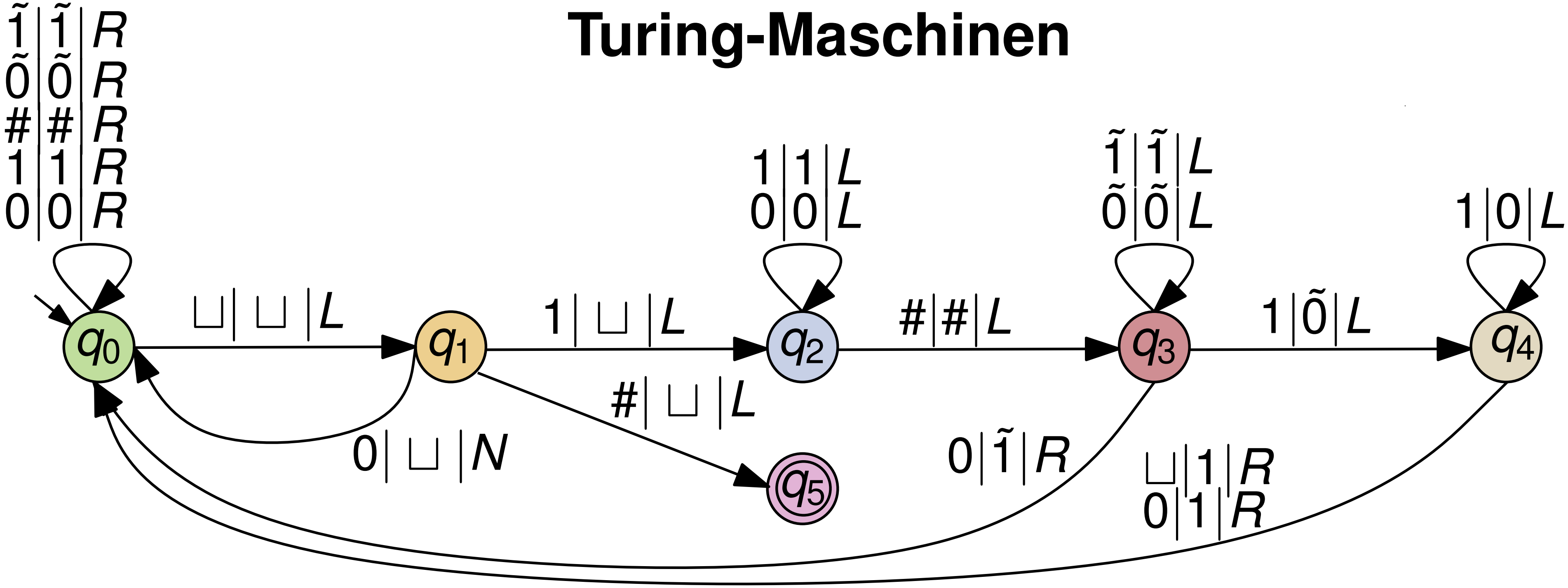
Turing-Maschinen



$\sqcup(q_0)111\#111\sqcup$

$\sqcup1(q_0)11\#111\sqcup$

Turing-Maschinen



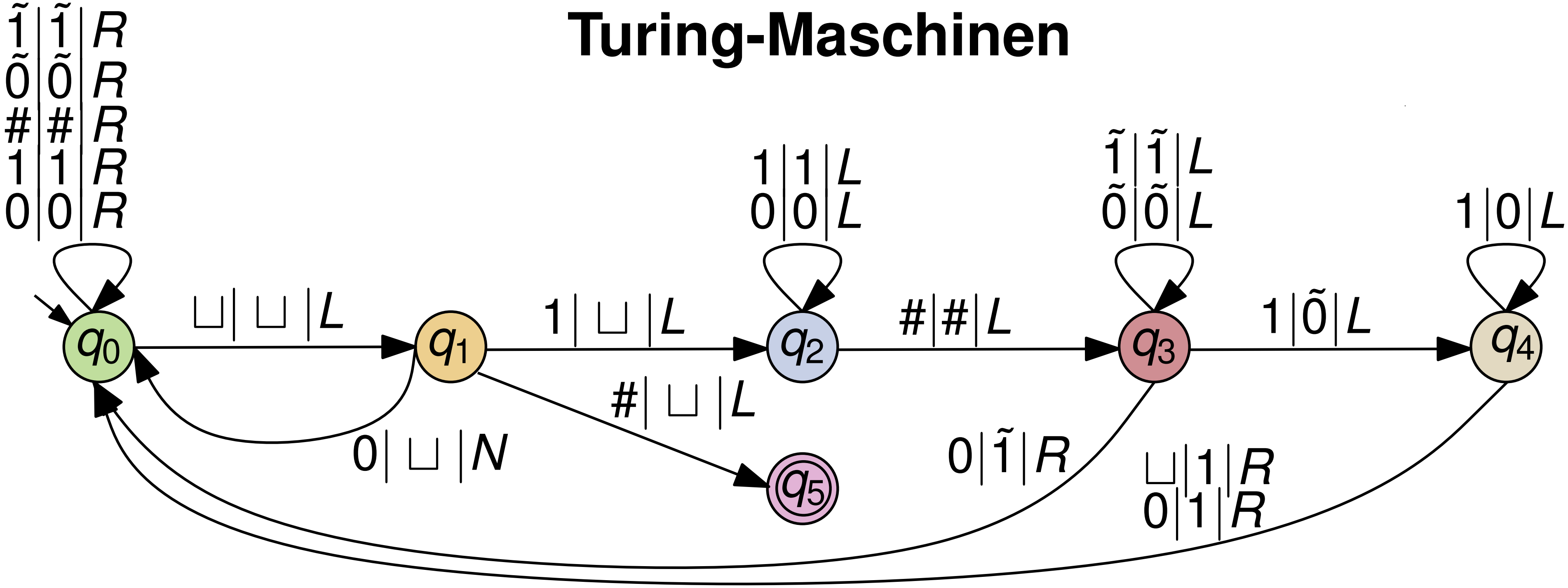
⊔(q_0)111#111⊔

⊔1(q_0)11#111⊔

⊔11(q_0)1#111⊔

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Turing-Maschinen



$\sqcup(q_0)111\#111\sqcup$

$\sqcup1(q_0)11\#111\sqcup$

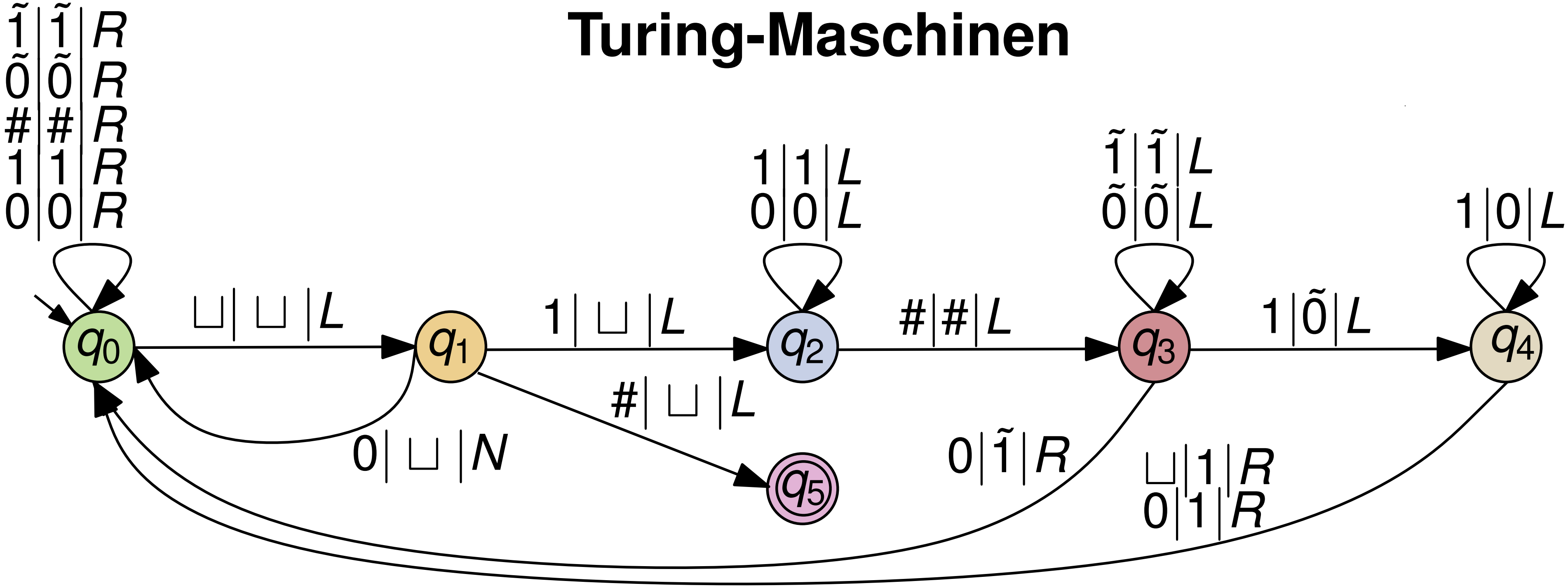
$\sqcup11(q_0)1\#111\sqcup$

...

$\sqcup111\#111(q_0)\sqcup$

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Turing-Maschinen



$\sqcup(q_0)1 1 1\#1 1 1\sqcup$

$\sqcup 1(q_0)1 1\#1 1 1\sqcup$

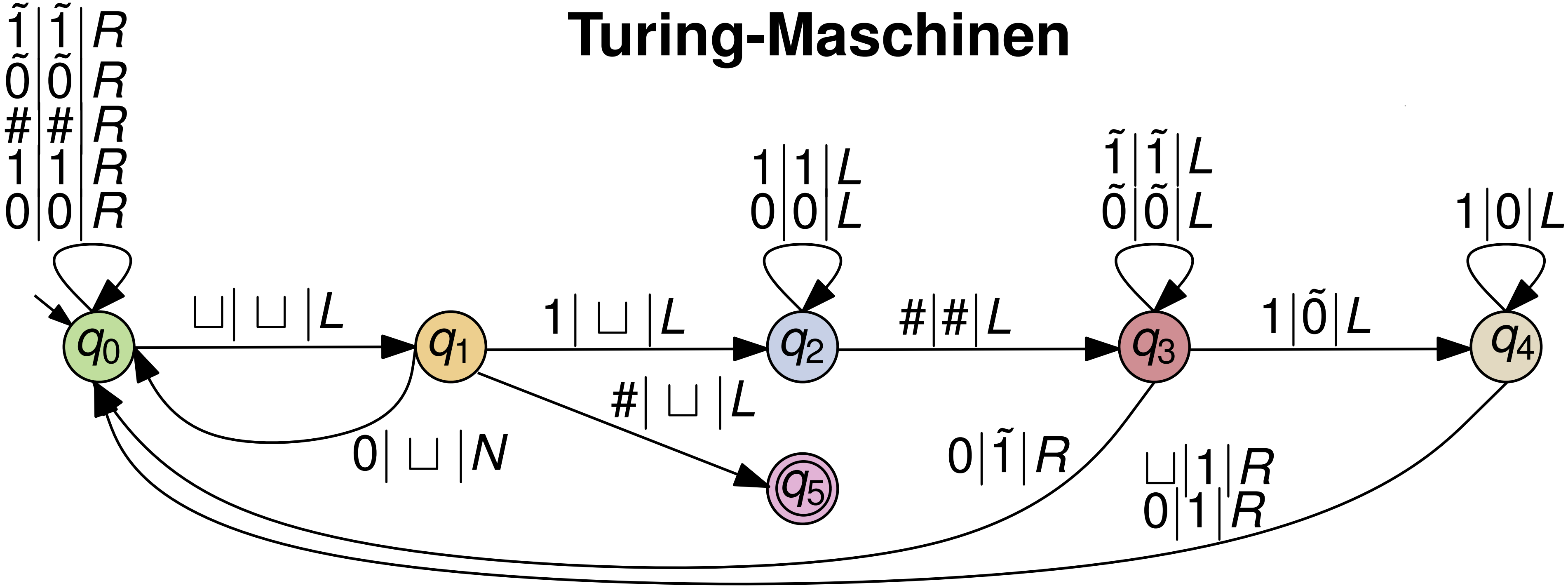
$\sqcup 1 1(q_0)1\#1 1 1\sqcup$

...

$\sqcup 1 1 1\#1 1 1(q_0)\sqcup$

$\sqcup 1 1 1\#1 1(q_1)1\sqcup$

Turing-Maschinen



$\sqcup(q_0)111\#111\sqcup$

$\sqcup1(q_0)11\#111\sqcup$

$\sqcup11(q_0)1\#111\sqcup$

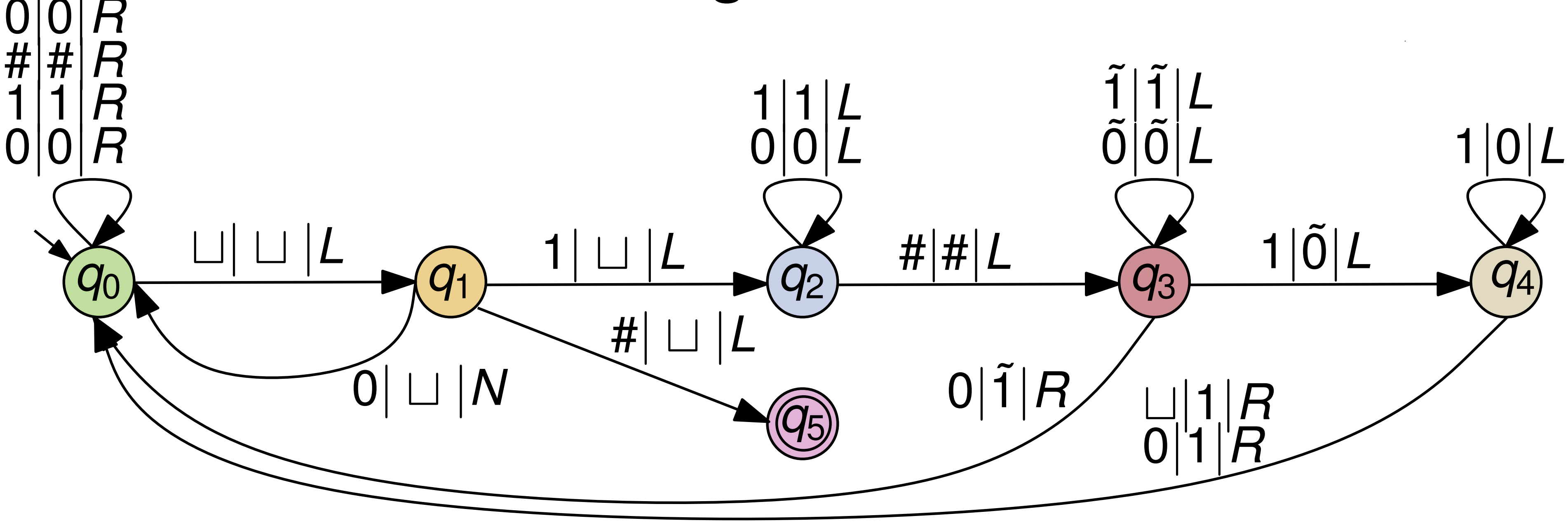
...

$\sqcup111\#111(q_0)\sqcup$

$\sqcup111\#11(q_1)1\sqcup$

$\sqcup111\#1(q_2)1\sqcup$

Turing-Maschinen



⊔(q_0)111#111⊔

⊔1(q_0)11#111⊔

⊔11(q_0)1#111⊔

...

⊔111#111(q_0)⊔

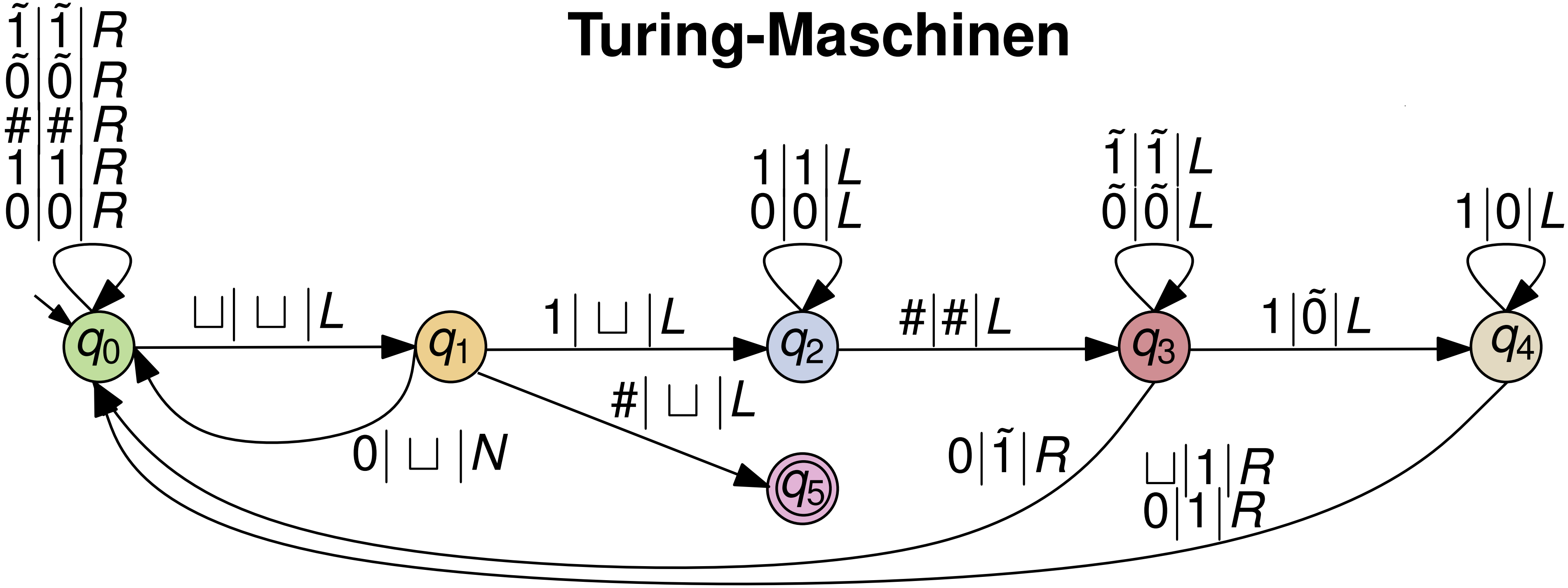
⊔111#11(q_1)1⊔

⊔111#1(q_2)1⊔

⊔111#(q_2)11⊔

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Turing-Maschinen



⊔(q_0)1 1 1#1 1 1⊔

⊔1(q_0)1 1#1 1 1⊔

⊔1 1(q_0)1#1 1 1⊔

. . .

⊔1 1 1#1 1 1(q_0)⊔

⊔1 1 1#1 1(q_1)1⊔

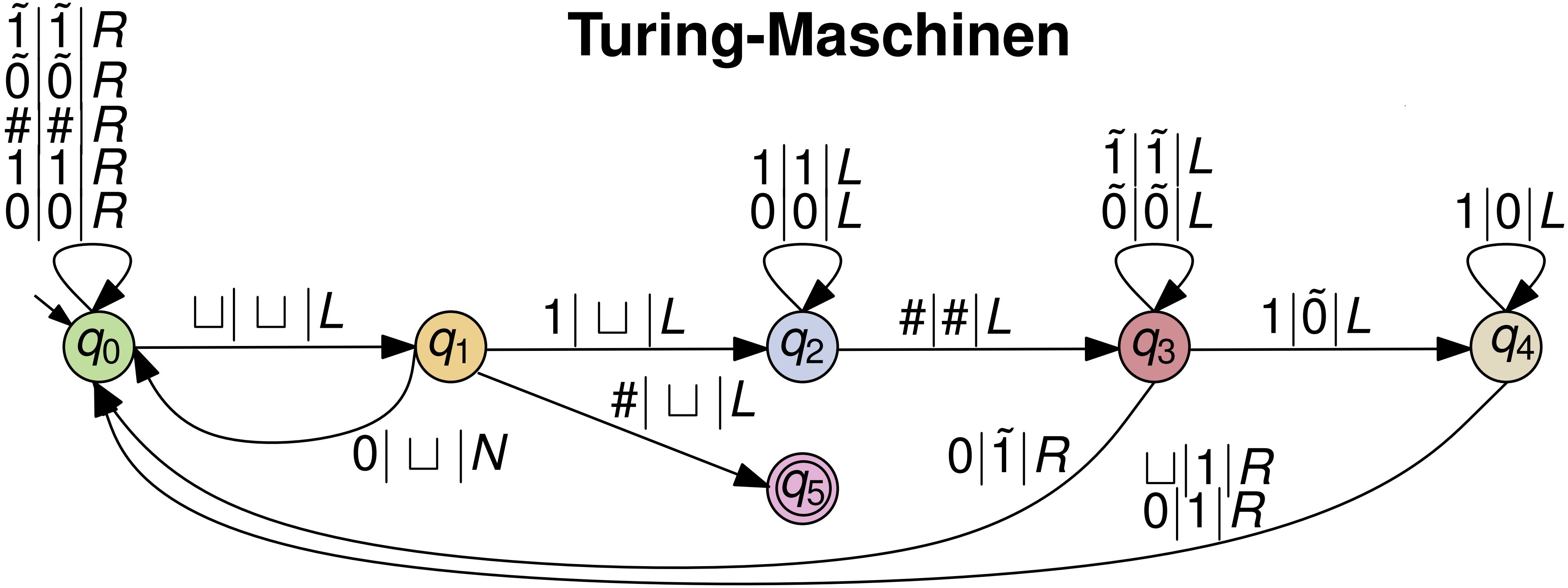
⊔1 1 1#1(q_2)1 1⊔

⊔1 1 1#(q_2)1 1⊔

⊔1 1 1(q_2)#1 1⊔

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Turing-Maschinen



$\sqcup(q_0)111\#111\sqcup$

$\sqcup1(q_0)11\#111\sqcup$

$\sqcup11(q_0)1\#111\sqcup$

...

$\sqcup111\#111(q_0)\sqcup$

$\sqcup111\#11(q_1)1\sqcup$

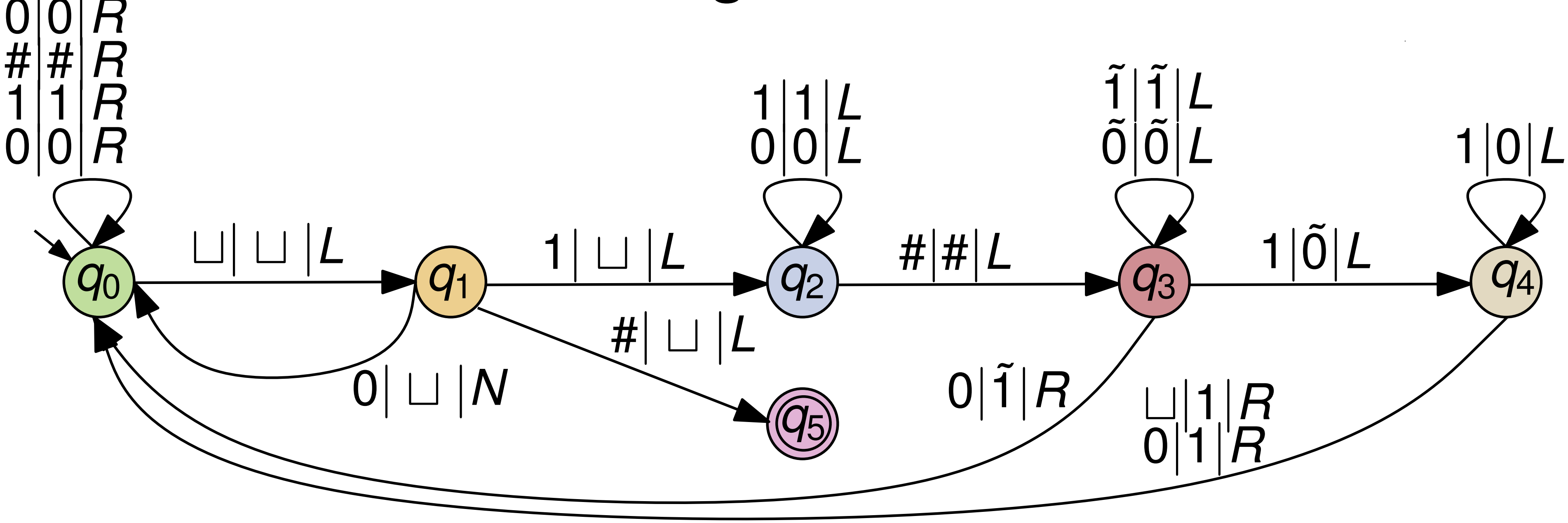
$\sqcup111\#1(q_2)1\sqcup$

$\sqcup111\#(q_2)11\sqcup$

$\sqcup111(q_2)\#11\sqcup$

$\sqcup11(q_3)1\#11\sqcup$

Turing-Maschinen



$\sqcup(q_0)111\#111\sqcup$

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

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

...

$\sqcup 111\#111(q_0)\sqcup$

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

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

$\sqcup 111\#(q_2)11\sqcup$

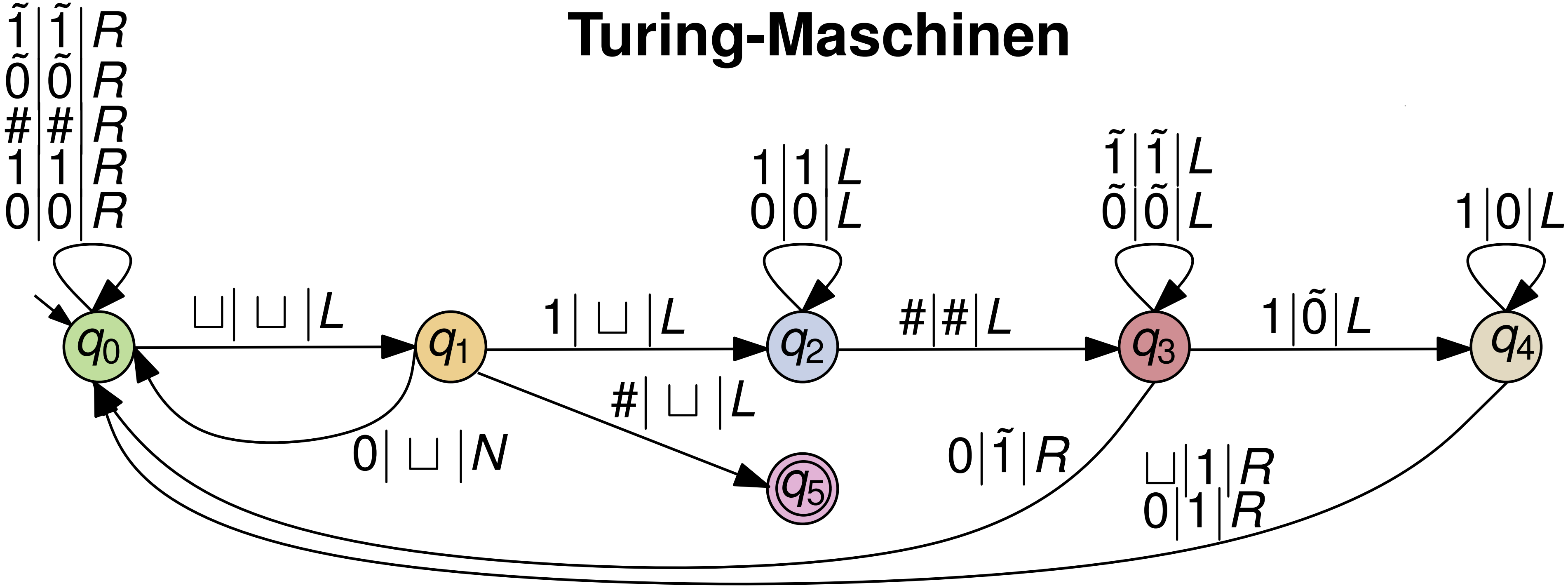
$\sqcup 111(q_2)\#11\sqcup$

$\sqcup 11(q_3)1\#11\sqcup$

$\sqcup 1(q_4)1\tilde{0}\#11\sqcup$

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Turing-Maschinen



$\sqcup(q_0)111\#111\sqcup$

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

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

...

$\sqcup 111\#111(q_0)\sqcup$

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

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

$\sqcup 111\#(q_2)11\sqcup$

$\sqcup 111(q_2)\#11\sqcup$

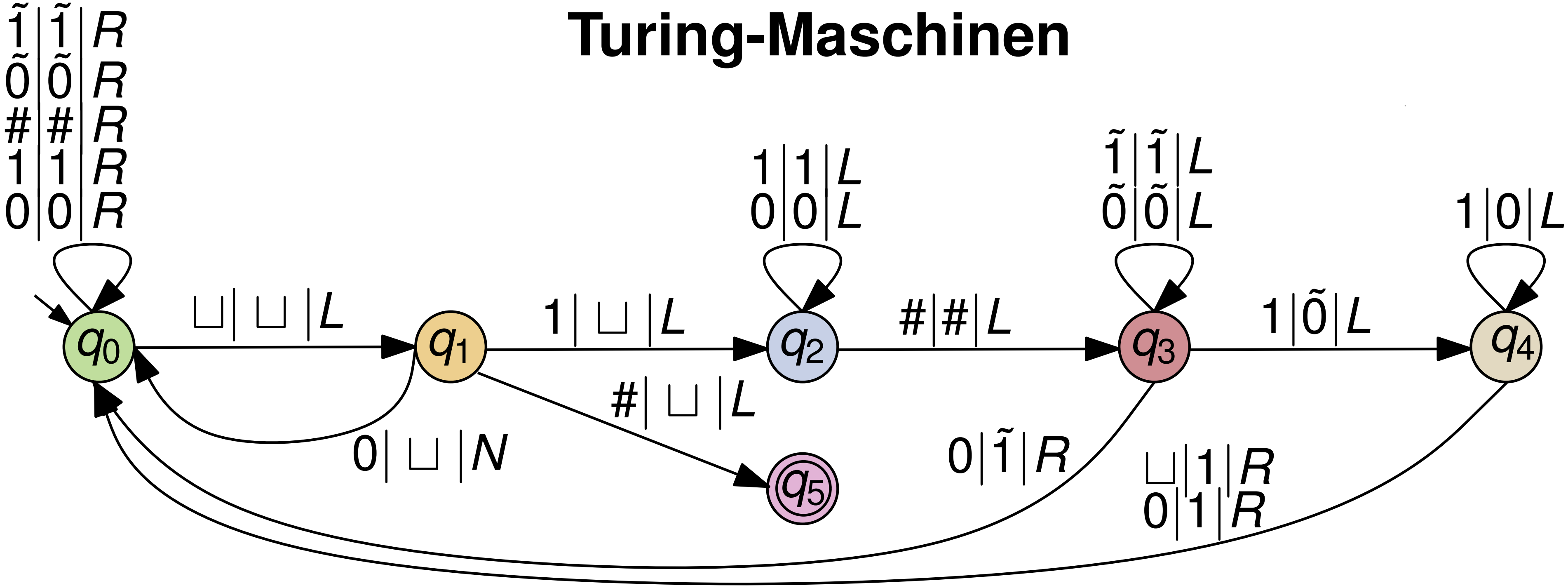
$\sqcup 11(q_3)1\#11\sqcup$

$\sqcup 1(q_4)1\tilde{0}\#11\sqcup$

$\sqcup(q_4)10\tilde{0}\#11\sqcup$

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

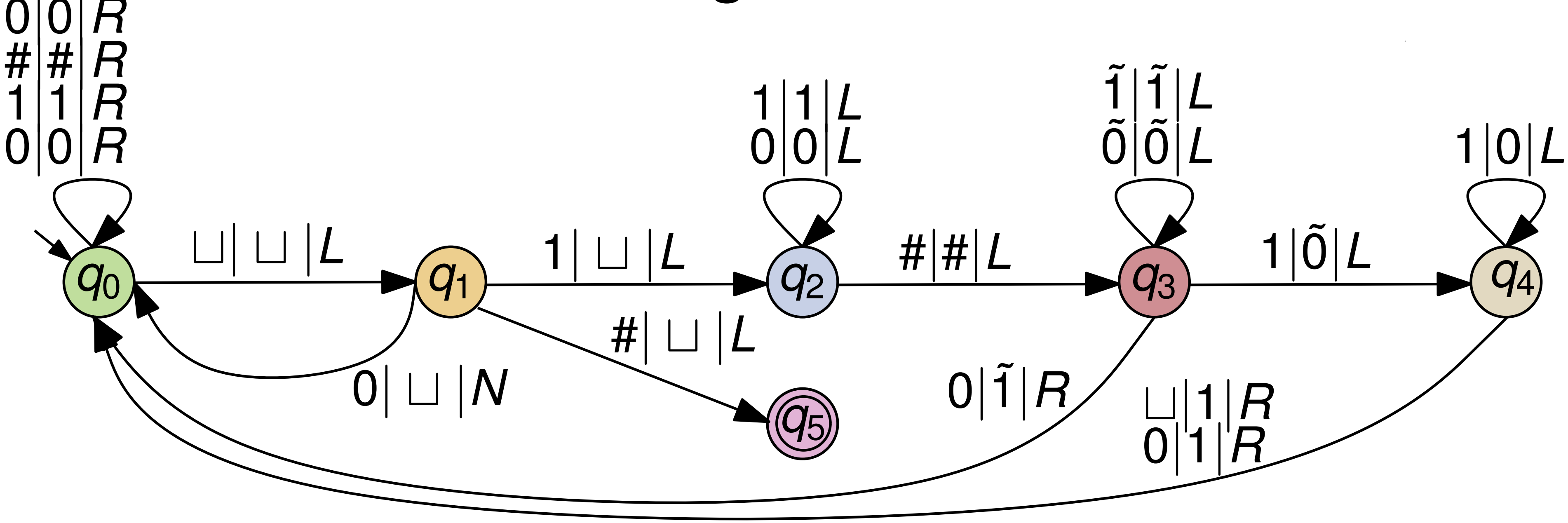
Turing-Maschinen



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

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Turing-Maschinen



$\sqcup(q_0)111\#111\sqcup$

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

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

. . .

$\sqcup 111\#111(q_0)\sqcup$

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

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

$\sqcup 111\#(q_2)11\sqcup$

$\sqcup 111(q_2)\#11\sqcup$

$\sqcup 11(q_3)1\#11\sqcup$

$\sqcup 1(q_4)1\tilde{0}\#11\sqcup$

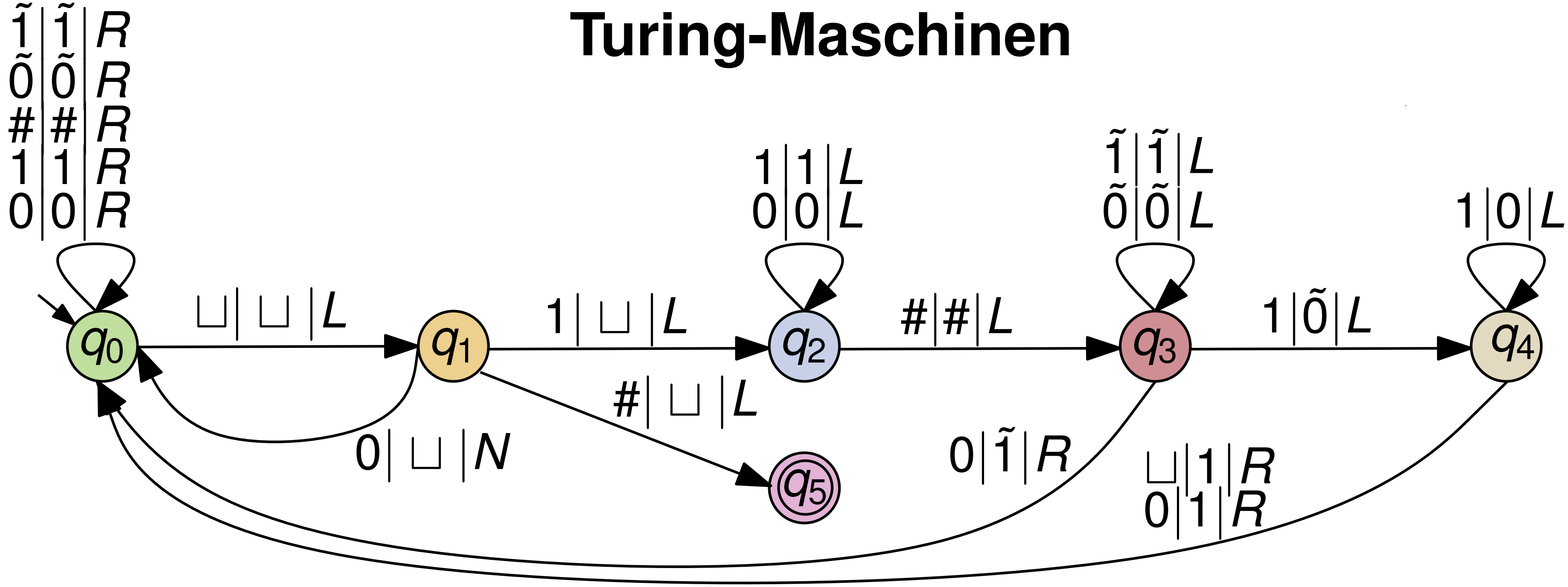
$\sqcup(q_4)10\tilde{0}\#11\sqcup$

$\sqcup(q_4)\sqcup 00\tilde{0}\#11\sqcup$

$\sqcup 1(q_0)00\tilde{0}\#11\sqcup$

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Turing-Maschinen



$\sqcup(q_0)111\#111\sqcup$

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

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

...

$\sqcup 111\#111(q_0)\sqcup$

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

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

$\sqcup 111\#(q_2)11\sqcup$

$\sqcup 111(q_2)\#11\sqcup$

$\sqcup 11(q_3)1\#11\sqcup$

$\sqcup 1(q_4)1\tilde{0}\#11\sqcup$

$\sqcup(q_4)10\tilde{0}\#11\sqcup$

$\sqcup(q_4)\sqcup 00\tilde{0}\#11\sqcup$

$\sqcup 1(q_0)00\tilde{0}\#11\sqcup$

...

$\sqcup 100\tilde{0}\#11(q_0)\sqcup$

$\sqcup 100\tilde{0}\#1(q_1)1\sqcup$

$\sqcup 100\tilde{0}\#(q_2)1\sqcup$

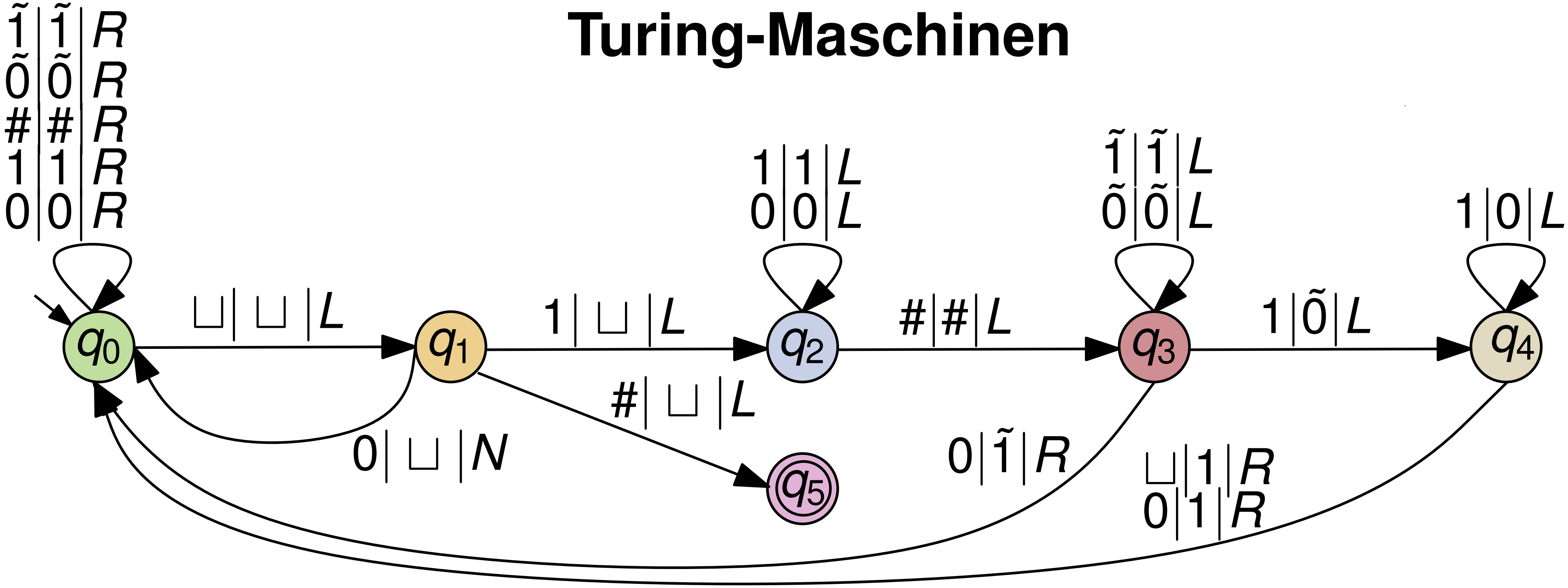
$\sqcup 100\tilde{0}(q_2)\#1\sqcup$

$\sqcup 100(q_3)\tilde{0}\#1\sqcup$

$\sqcup 10(q_3)0\tilde{0}\#1\sqcup$

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Turing-Maschinen



$\sqcup(q_0)111\#111\sqcup$

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

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

...

$\sqcup 111\#111(q_0)\sqcup$

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

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

$\sqcup 111\#(q_2)11\sqcup$

$\sqcup 111(q_2)\#11\sqcup$

$\sqcup 11(q_3)1\#11\sqcup$

$\sqcup 1(q_4)1\tilde{0}\#11\sqcup$

$\sqcup(q_4)10\tilde{0}\#11\sqcup$

$\sqcup(q_4)\sqcup 00\tilde{0}\#11\sqcup$

$\sqcup 1(q_0)00\tilde{0}\#11\sqcup$

...

$\sqcup 100\tilde{0}\#11(q_0)\sqcup$

$\sqcup 100\tilde{0}\#1(q_1)1\sqcup$

$\sqcup 100\tilde{0}\#(q_2)1\sqcup$

$\sqcup 100\tilde{0}(q_2)\#1\sqcup$

$\sqcup 100(q_3)\tilde{0}\#1\sqcup$

$\sqcup 10(q_3)0\tilde{0}\#1\sqcup$

$\sqcup 10\tilde{1}(q_0)\tilde{0}\#1\sqcup$

...

$\sqcup 10\tilde{1}\tilde{0}\#1(q_0)\sqcup$

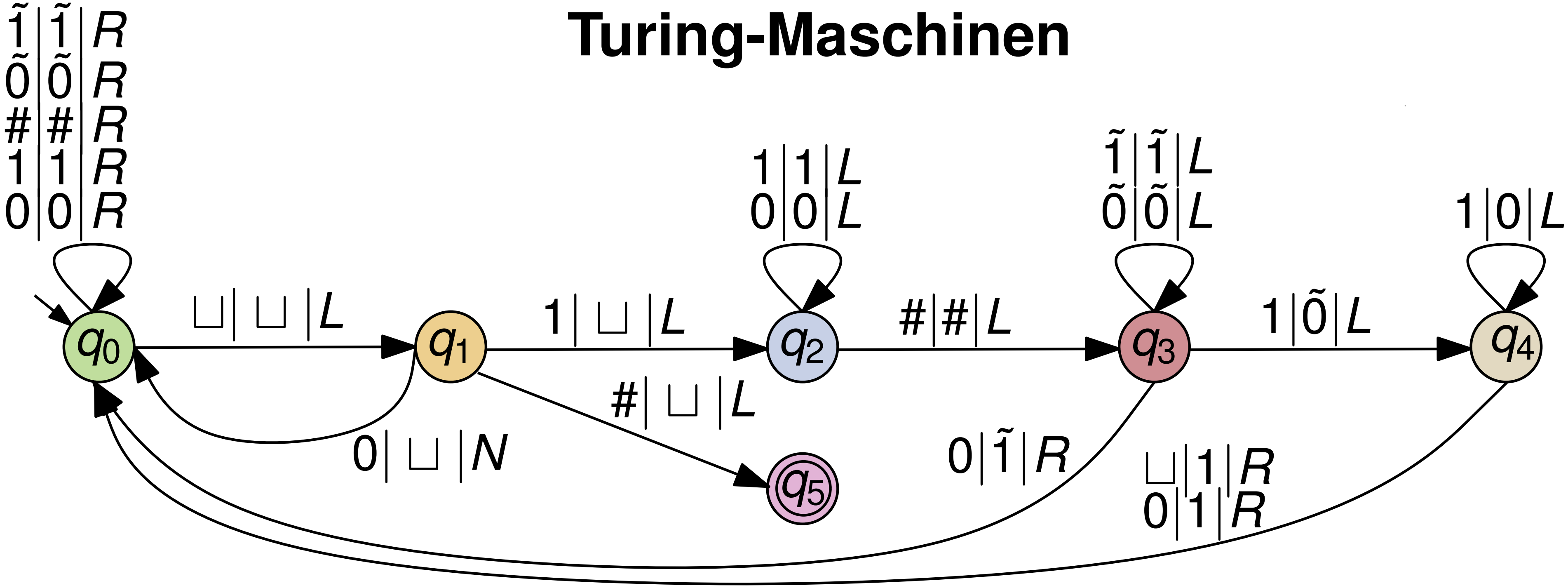
$\sqcup 10\tilde{1}\tilde{0}\#(q_1)1\sqcup$

$\sqcup 10\tilde{1}\tilde{0}(q_2)\#\sqcup$

$\sqcup 10\tilde{1}(q_3)\tilde{0}\#\sqcup$

$\sqcup 10(q_3)\tilde{1}\tilde{0}\#\sqcup$

Turing-Maschinen



⊔(q_0)111#111⊔
⊔1(q_0)11#111⊔
⊔11(q_0)1#111⊔
. . .
⊔111#111(q_0)⊔
⊔111#11(q_1)1⊔
⊔111#1(q_2)1⊔

⊔111#(q_2)11⊔
⊔111(q_2)#11⊔
⊔11(q_3)1#11⊔
⊔1(q_4)10̃#11⊔
⊔(q_4)100̃#11⊔
⊔(q_4)⊔ 000̃#11⊔
⊔1(q_0)000̃#11⊔

...
⊔1000̃#11(q_0)⊔
⊔1000̃#1(q_1)1⊔
⊔1000̃#(q_2)1⊔
⊔1000̃(q_2)#1⊔
⊔100(q_3)0̃#1⊔
⊔10(q_3)00̃#1⊔

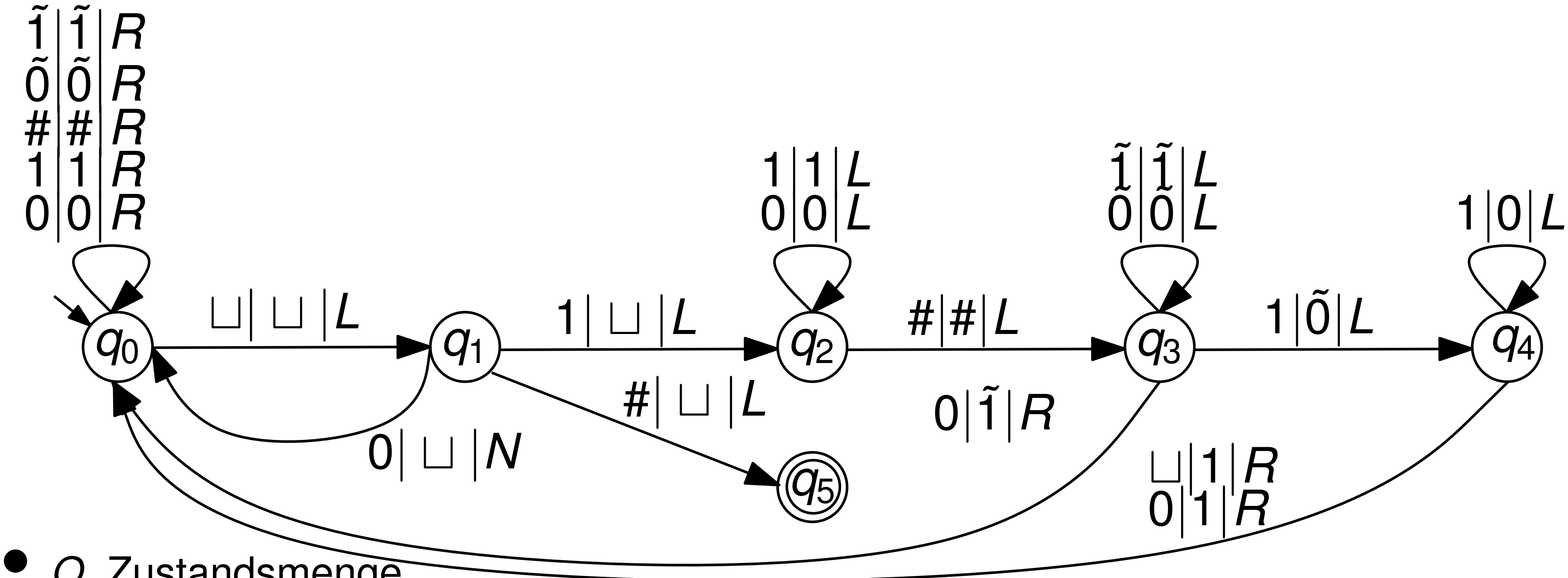
⊔101̃(q_0)0̃#1⊔
...
⊔101̃0̃#1(q_0)⊔
⊔101̃0̃#(q_1)1⊔
⊔101̃0̃(q_2)#⊔
⊔101̃(q_3)0̃#⊔
⊔10(q_3)1̃0̃#⊔

⊔1(q_3)01̃0̃#⊔
⊔11̃(q_0)1̃0̃#⊔
...
⊔11̃1̃0̃#(q_0)⊔
⊔11̃1̃0̃(q_1)#⊔
⊔11̃1̃0̃(q_5)⊔

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Turing-Maschinen

Ist die Turing-Maschine vollständig beschrieben?

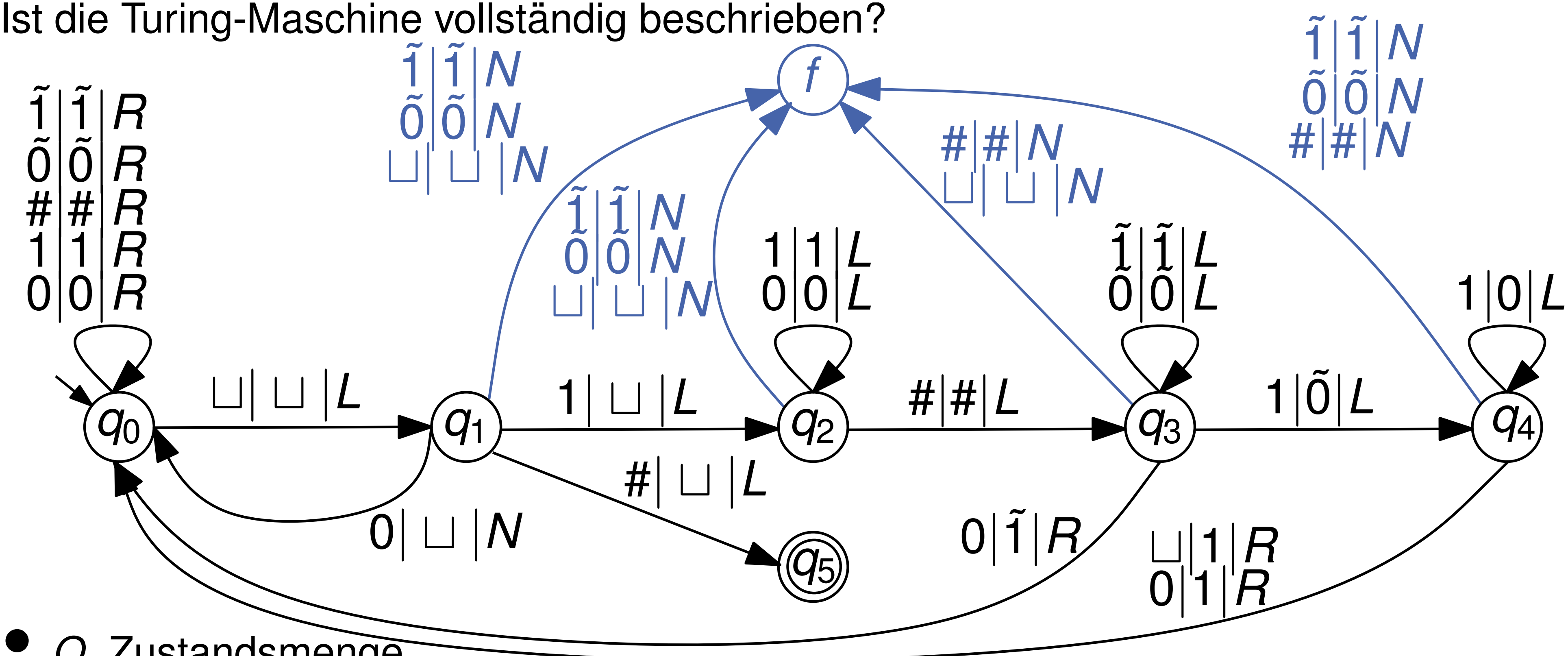


- Q, Zustandsmenge,
- Σ, einem endlichen Eingabealphabet,
- $\sqcup$, einem Blanksymbol mit $\sqcup \notin \Sigma$,
- Γ, einem endlichen Bandalphabet mit $\Sigma \cup \{\sqcup\} \subseteq \Gamma$,
- $s \in Q$, einem Startzustand,
- $\delta\colon Q \times \Gamma \to Q \times \Gamma \times \{L, R, N\}$,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

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Turing-Maschinen

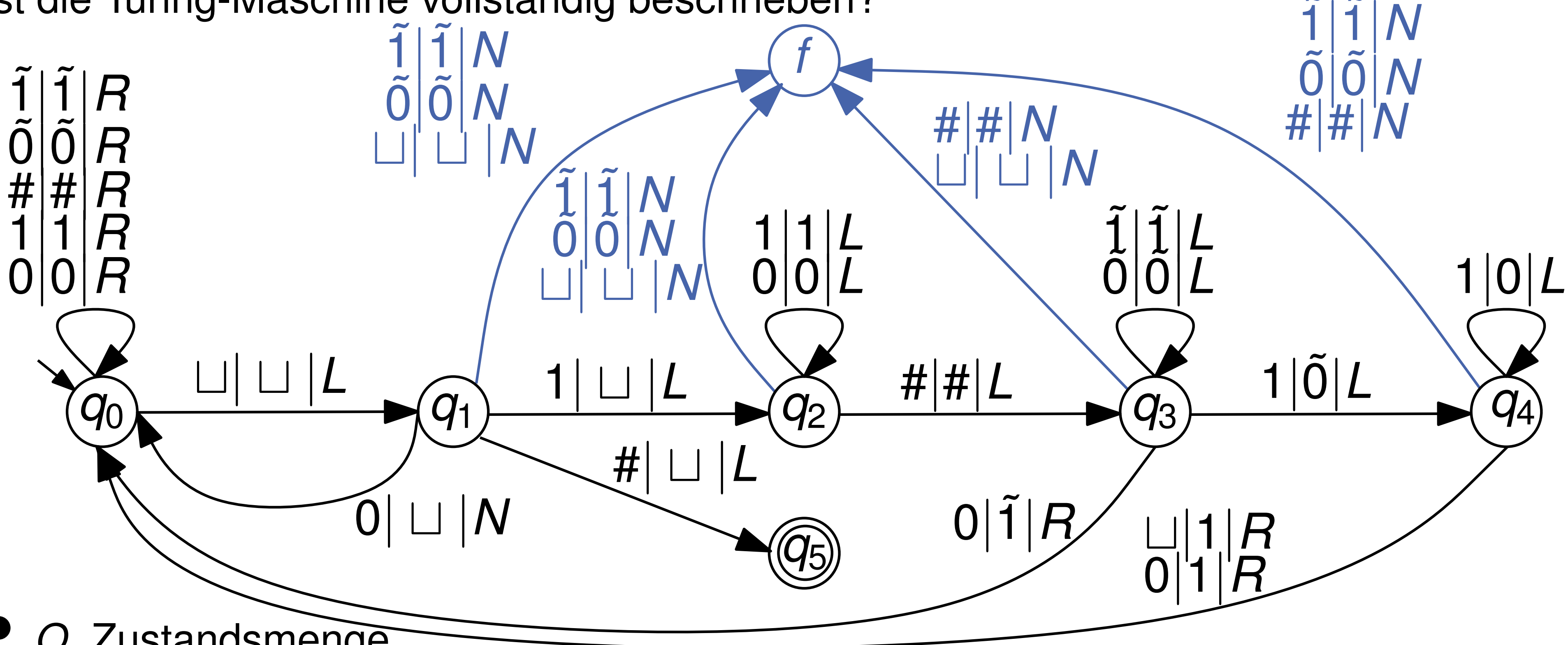
Ist die Turing-Maschine vollständig beschrieben?



- 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,
- $\delta\colon Q \times \Gamma \to Q \times \Gamma \times \{L, R, N\}$, einer Übergangsfunktion.
- $F \subseteq Q$, einer Menge von Endzuständen.

Turing-Maschinen

Ist die Turing-Maschine vollständig beschrieben?



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

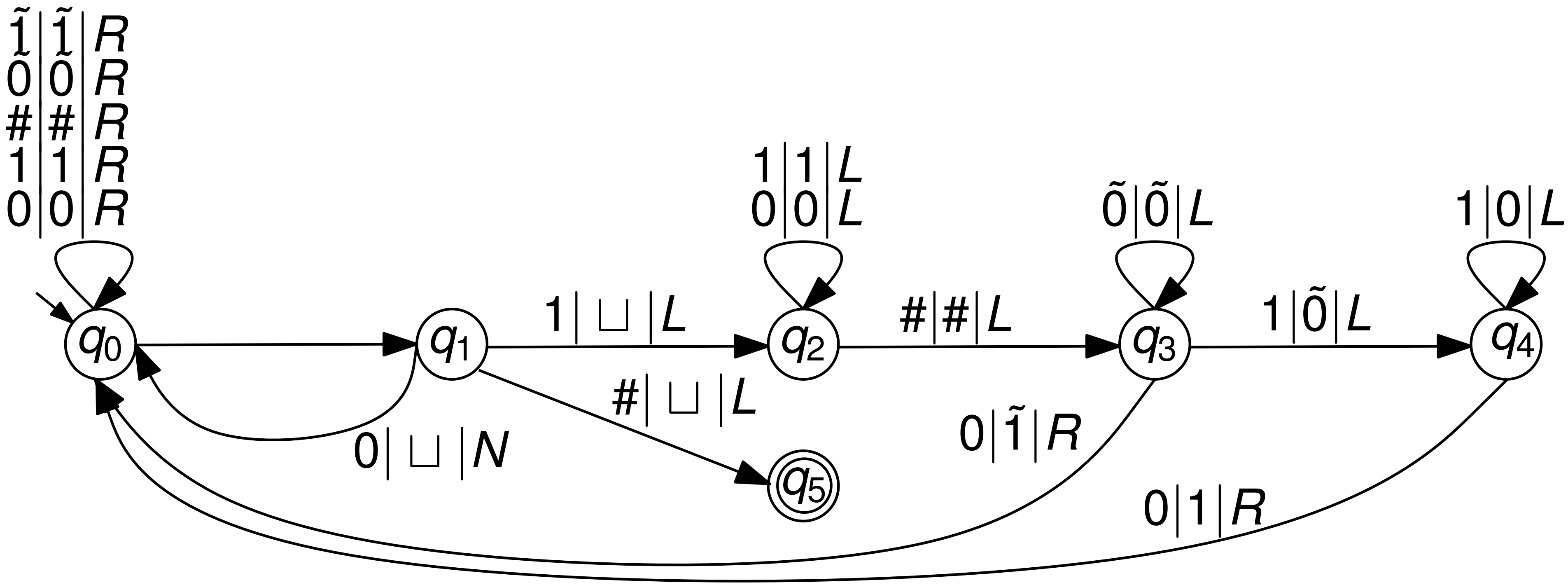
Was berechnet die Turingmaschine?

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Turing-Maschinen

Wie ändern, damit Addition von allgemeineren Eingaben funktioniert?

Beispiel: 100#110



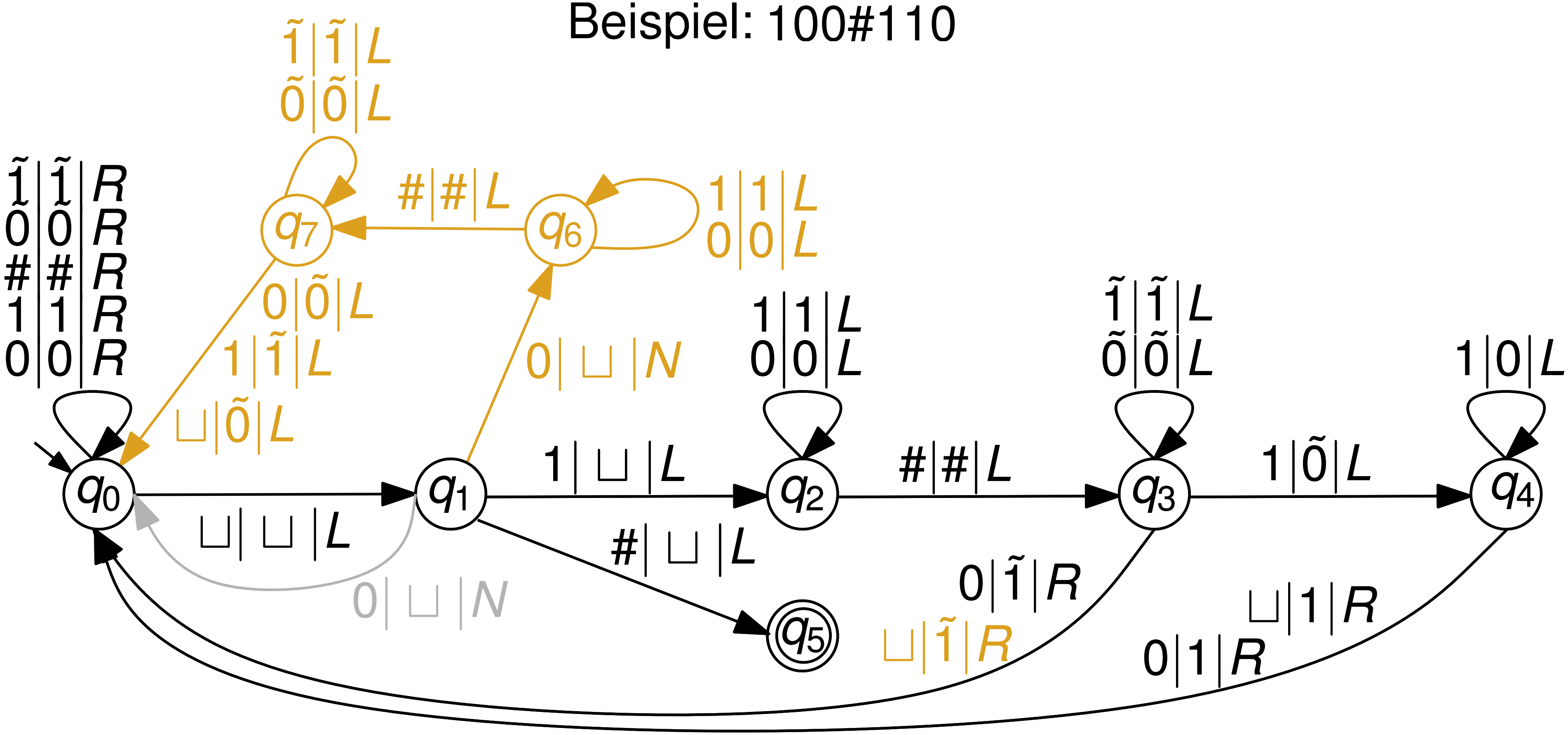
3 min Zeit

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Turing-Maschinen

Wie ändern, damit Addition von allgemeineren Eingaben funktioniert?

Beispiel: 100#110



Hinzugefügt

Entfernt

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Erweiterungen von Turing-Maschinen

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Mehrere Spuren

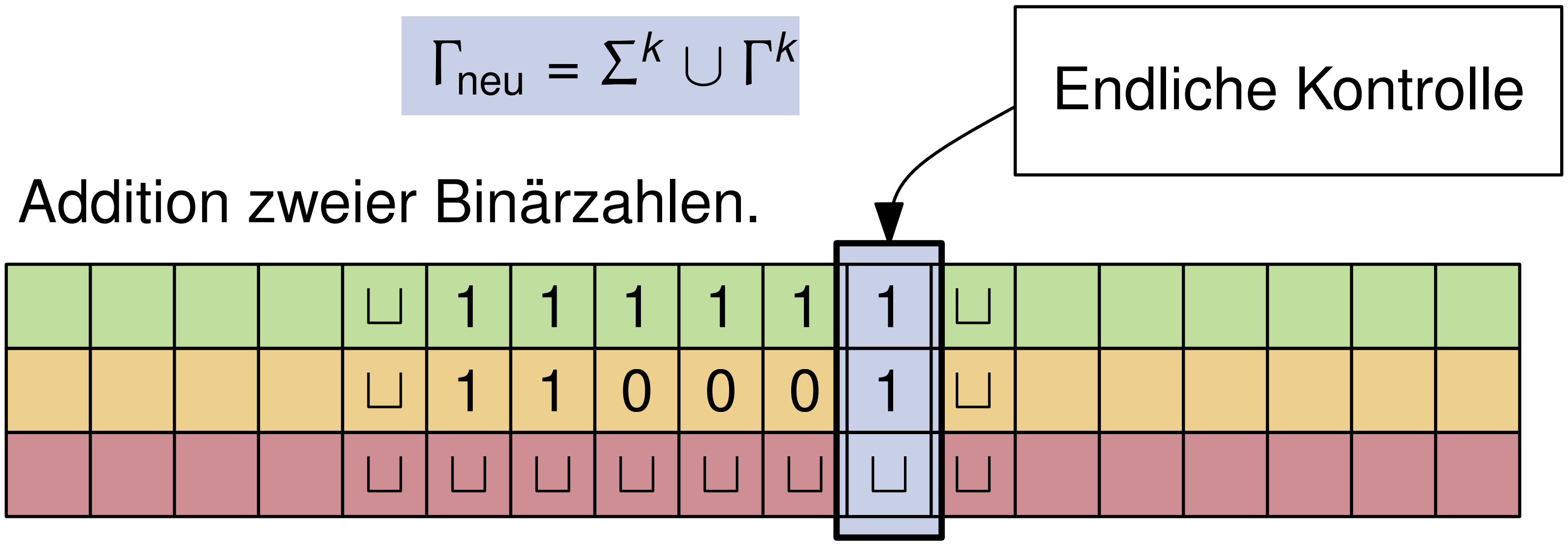
Turingmaschine mit k Spuren:

Idee: Teile das Eingabeband in k Spuren ein.

Formal: Erweitere das Bandalphabet um k-dimensionale Vektoren:

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

Beispiel: Addition zweier Binärzahlen.



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

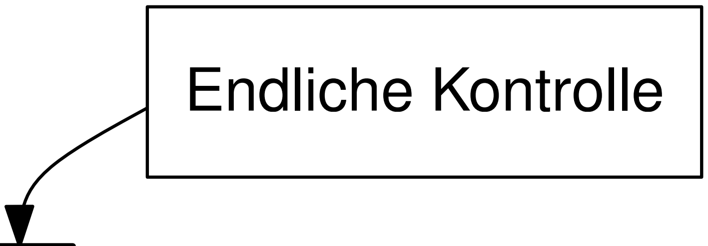
Mehrere Spuren

Turingmaschine mit k Spuren:

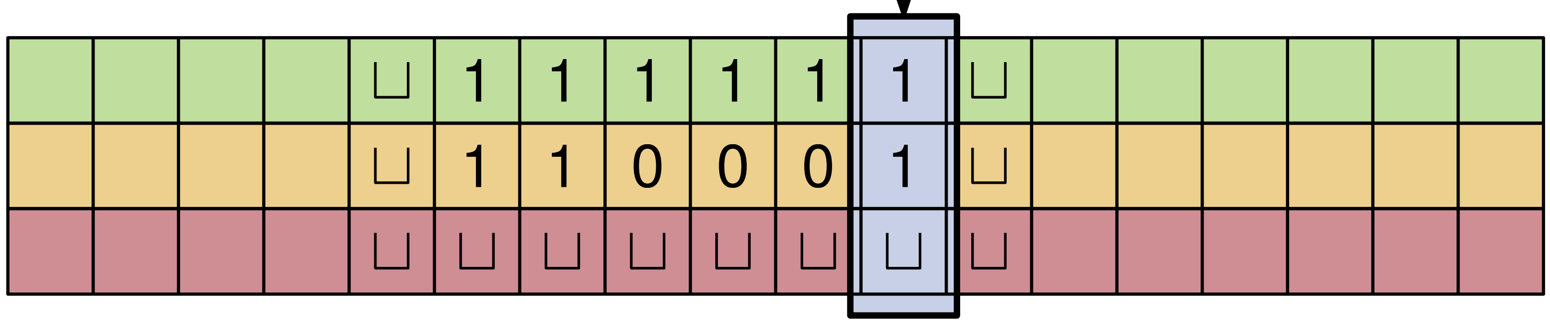
Idee: Teile das Eingabeband in k Spuren ein.

Formal: Erweitere das Bandalphabet um k-dimensionale Vektoren:

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



Beispiel: Addition zweier Binärzahlen.



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

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Mehrere Spuren

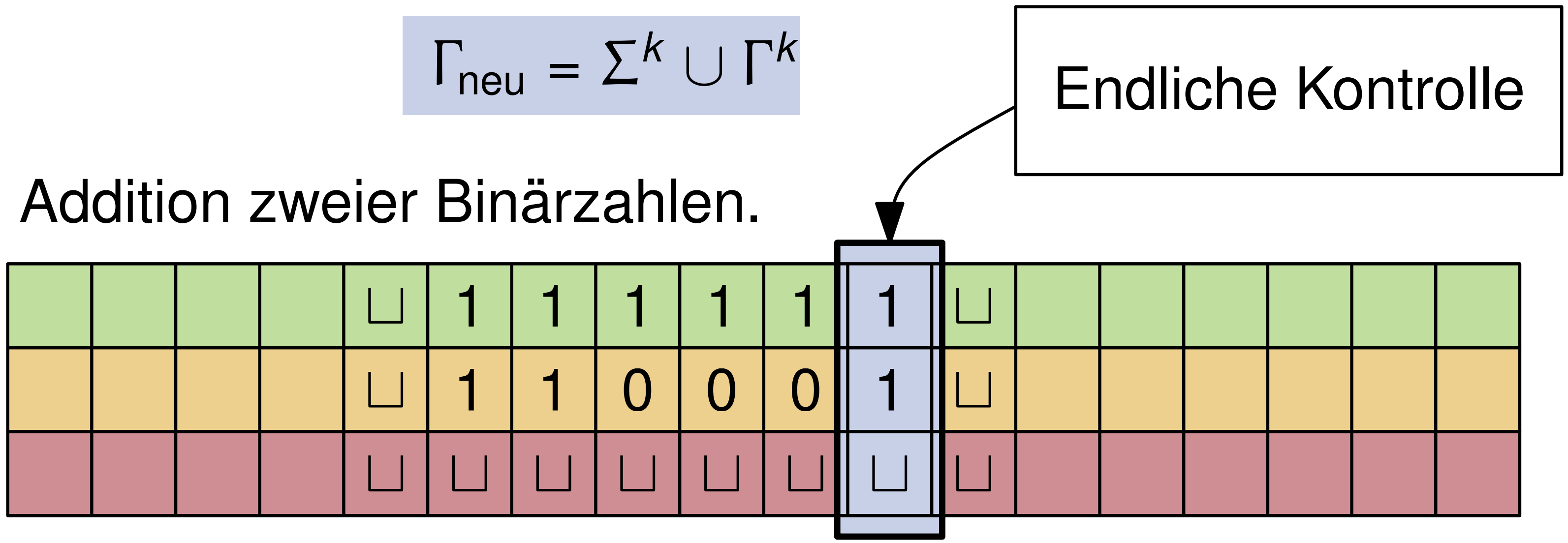
Turingmaschine mit k Spuren:

Idee: Teile das Eingabeband in k Spuren ein.

Formal: Erweitere das Bandalphabet um k-dimensionale Vektoren:

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

Beispiel: Addition zweier Binärzahlen.



- 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

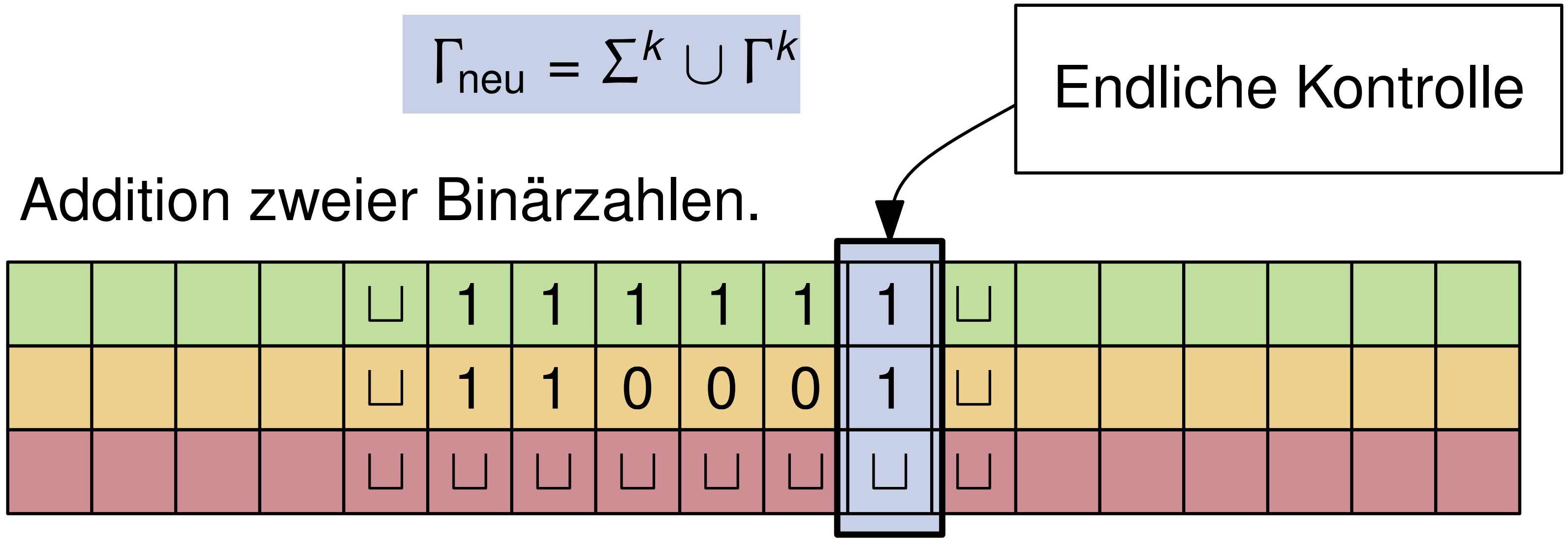
Turingmaschine mit k Spuren:

Idee: Teile das Eingabeband in k Spuren ein.

Formal: Erweitere das Bandalphabet um k-dimensionale Vektoren:

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

Beispiel: Addition zweier Binärzahlen.



- 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.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Mehrere Spuren

Turingmaschine mit k Spuren:

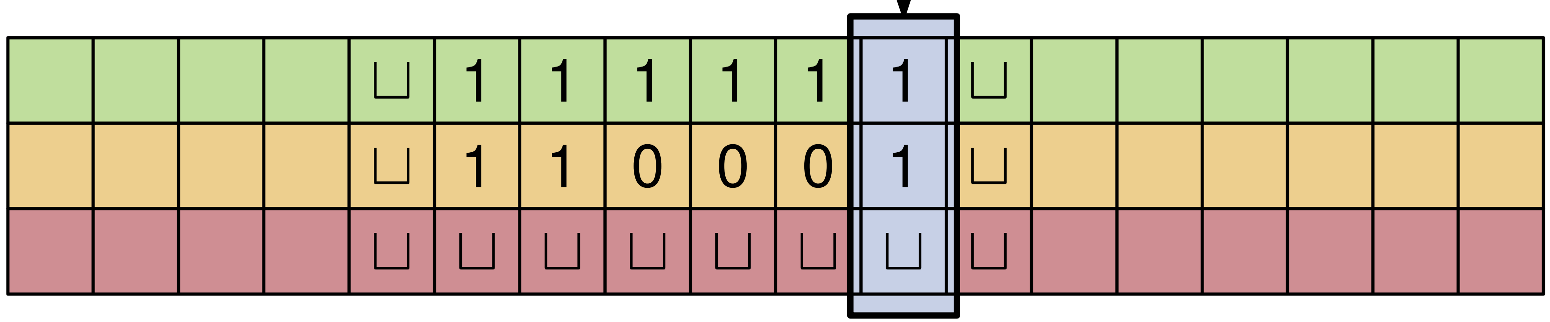
Idee: Teile das Eingabeband in *k* Spuren ein.

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
- ts.
-

Ist eine Turing-Maschine mit $k \geq 2$ mächtiger, als eine Turing-Maschine mit einer Spur?

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

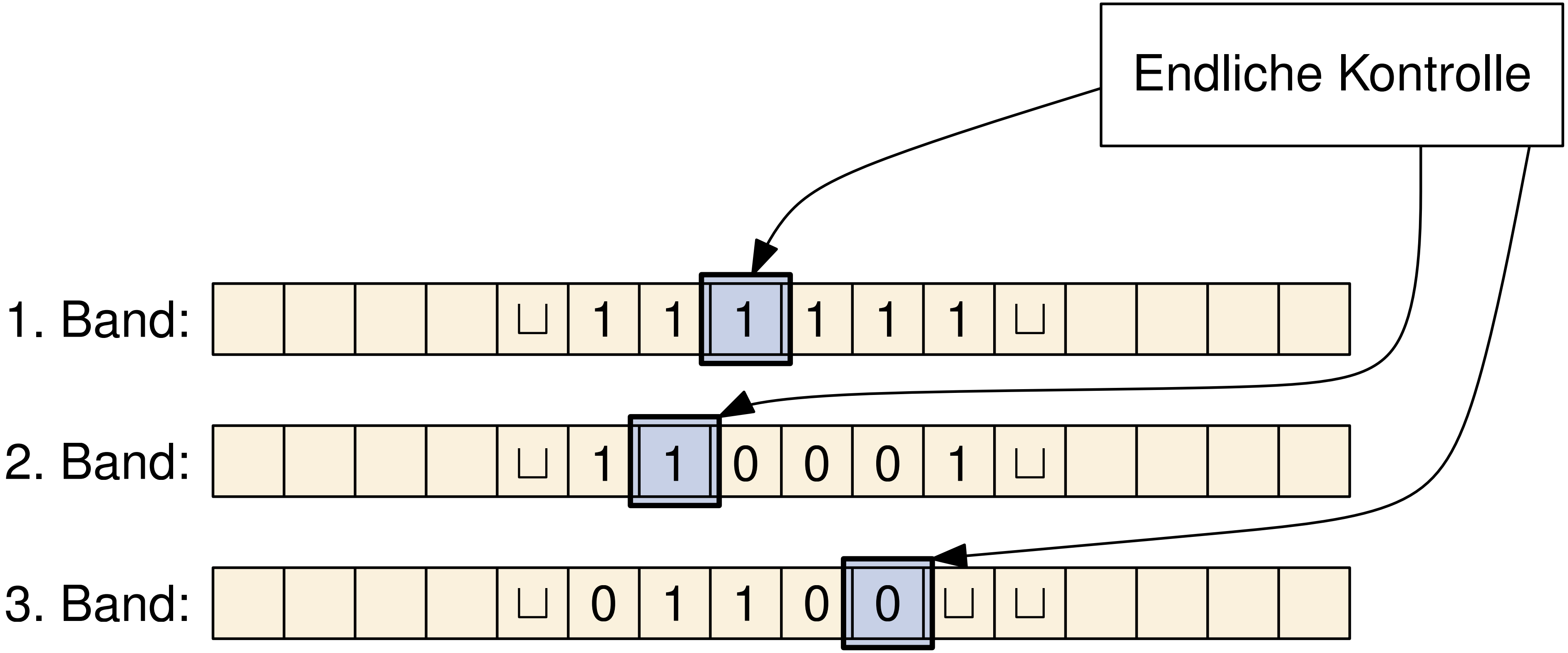
Mehrere Bänder

Turing-Maschine mit k Bändern:

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

Formal: Passe Übergangsfunktion δ an.

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



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

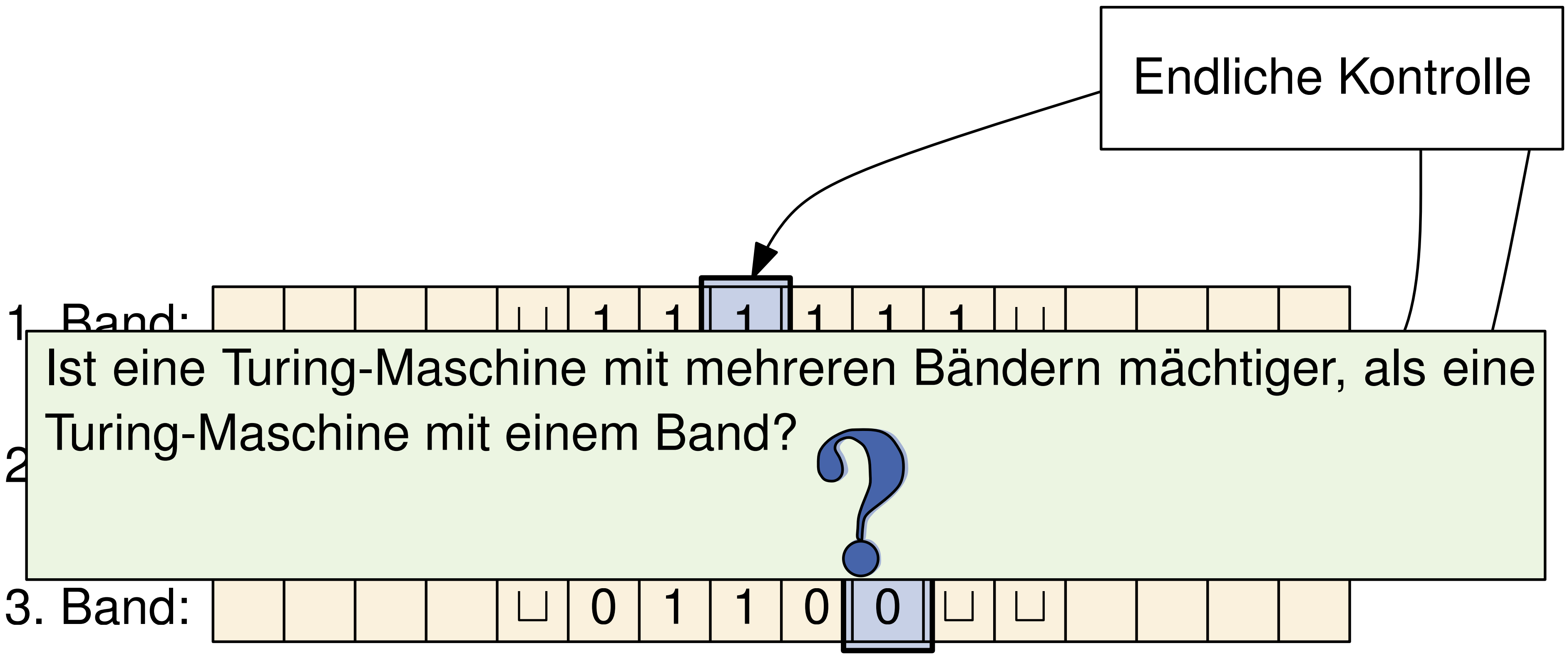
Mehrere Bänder

Turing-Maschine mit k Bändern:

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

Formal: Passe Übergangsfunktion δ an.

$$\delta\colon Q \times \Gamma^k \to Q \times \Gamma^k \times \{R, L, N\}^k$$



Ist eine Turing-Maschine mit mehreren Bändern mächtiger, als eine Turing-Maschine mit einem Band?

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Erweiterte Turing-Maschinen

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

Beweis:

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Erweiterte Turing-Maschinen

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

Beweis:

3 min Zeit

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Erweiterte Turing-Maschinen

Satz: Eine k-Band Turing-Maschine $\mathcal{M}$, die mit Rechenzeit $t(n)$ und Platz $s(n)$ auskommt, kann von einer Turing-Maschine $\mathcal{M}'$ 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:

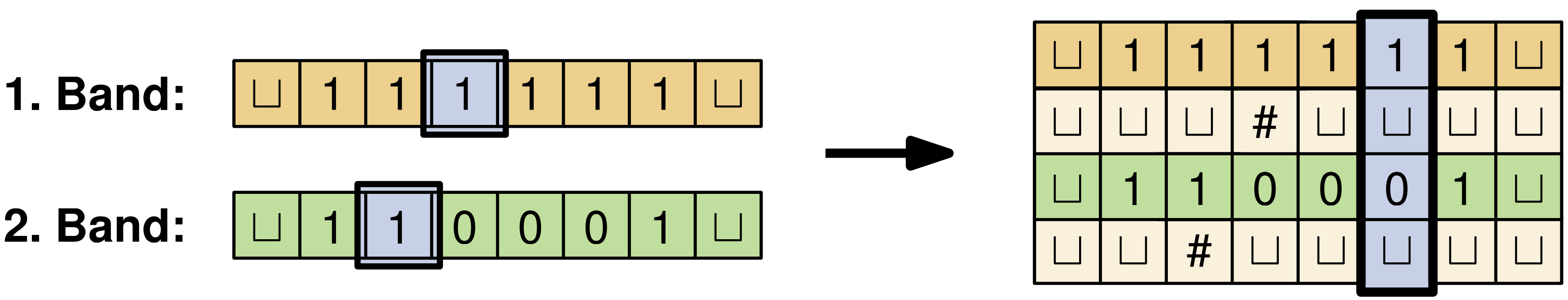
Erweiterte Turing-Maschinen

Satz: Eine k-Band Turing-Maschine $\mathcal{M}$, die mit Rechenzeit $t(n)$ und Platz $s(n)$ auskommt, kann von einer Turing-Maschine $\mathcal{M}'$ 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 $1, 3, 5, \ldots, 2k - 1$ enthalten Inhalt der Bänder $1 \ldots, k$ von M.
- Auf den geraden Spuren $1, 2, \ldots, 2k$ 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 Banschriften 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 Banschriften 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 #

$\Rightarrow$ Ausgangssituation für nächsten Berechnungsschritt $t + 1$.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

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.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

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 $\mathcal{M}$.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Entscheidbarkeit

Entscheidbarkeit

Semientscheidbarkeit einer Sprache:
Eine Sprache $L \subseteq \Sigma^*$ heißt *rekursiv-aufzählbar* oder *semi-entscheidbar*, wenn es eine Turing-Maschine gibt, die genau die Eingaben w akzeptiert für die $w \in L$.

Entscheidbarkeit einer Sprache:
Eine Sprache $L \subseteq \Sigma^*$ heißt *rekursiv* oder *entscheidbar*, wenn es eine Turing-Maschine gibt, die auf allen Eingaben stoppt und eine Eingabe w genau dann akzeptiert, wenn $w \in L$ gilt.

10. Hilbertsches Problem

Frage: Ist die Sprache $D = \{p \mid p$ ist ein Polynom mit ganzzahligen Koeffizienten und ganzzahliger Nullstelle $\}$ entscheidbar?

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

10. Hilbertsches Problem

Frage: Ist die Sprache $D = \{p \mid p \text{ ist ein Polynom mit ganzzahligen Koeffizienten und ganzzahliger Nullstelle}\}$ entscheidbar?

$D_1 = \{p \mid \text{Polynom über } x \text{ mit ganzzahliger Nullstelle}\}$

Beobachtung: D_1 ist semi-entscheidbar.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

10. Hilbertsches Problem

Frage: Ist die Sprache $D = \{p \mid p$ ist ein Polynom mit ganzzahligen Koeffizienten und ganzzahliger Nullstelle $\}$ entscheidbar?

$D_1 = \{p \mid$Polynom über x mit ganzzahliger Nullstelle$\}$

Beobachtung: D_1 ist semi-entscheidbar.

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

10. Hilbertsches Problem

Frage: Ist die Sprache $D = \{p \mid p$ ist ein Polynom mit ganzzahligen Koeffizienten und ganzzahliger Nullstelle $\}$ entscheidbar?

$D_1 = \{p \mid$Polynom über x mit ganzzahliger Nullstelle$\}$

Beobachtung: D_1 ist semi-entscheidbar.

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

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

10. Hilbertsches Problem

Frage: Ist die Sprache $D = \{p \mid p$ ist ein Polynom mit ganzzahligen Koeffizienten und ganzzahliger Nullstelle $\}$ entscheidbar?

$D_1 = \{p \mid$ Polynom über x mit ganzzahliger Nullstelle$\}$

Beobachtung: D_1 ist semi-entscheidbar.

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

- Verfahren terminiert nur, wenn p eine ganzzahlige Nullstelle hat.
- Einschränkung auf ein bestimmtes Interval möglich $\rightsquigarrow$ D_1 entscheidbar

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

10. Hilbertsches Problem

Frage: Ist die Sprache $D = \{p \mid p$ ist ein Polynom mit ganzzahligen Koeffizienten und ganzzahliger Nullstelle $\}$ entscheidbar?

$D_1 = \{p \mid$Polynom über x mit ganzzahliger Nullstelle$\}$

Beobachtung: D_1 ist semi-entscheidbar.

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

- Verfahren terminiert nur, wenn p eine ganzzahlige Nullstelle hat.
- Einschränkung auf ein bestimmtes Interval möglich $\rightsquigarrow D_1$ entscheidbar

Dieses Intervall existiert für D nicht.

10. Hilbertsches Problem

Frage: Ist die Sprache $D = \{p \mid p$ ist ein Polynom mit ganzzahligen Koeffizienten und ganzzahliger Nullstelle $\}$ entscheidbar?

$D_1 = \{p \mid$Polynom über x mit ganzzahliger Nullstelle$\}$

Beobachtung: D_1 ist semi-entscheidbar.

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

- Verfahren terminiert nur, wenn p eine ganzzahlige Nullstelle hat.
- Einschränkung auf ein bestimmtes Interval möglich $\rightsquigarrow$ D_1 entscheidbar

Dieses Intervall existiert für D nicht. $\rightsquigarrow$ D ist nicht entscheidbar

Entscheidbarkeit

Zeigen Sie: Sind die Sprachen L_1 und L_2 entscheidbar, dann ist auch die Sprache $L_1 \setminus L_2$ entscheidbar.

Entscheidbarkeit

Zeigen Sie: Sind die Sprachen L_1 und L_2 entscheidbar, dann ist auch die Sprache $L_1 \setminus L_2$ entscheidbar.

Beweis:

Da L_i entscheidbar, existiert eine TM $\mathcal{M}_i$ die L_i entscheidet.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Entscheidbarkeit

Zeigen Sie: Sind die Sprachen L_1 und L_2 entscheidbar, dann ist auch die Sprache $L_1 \setminus L_2$ entscheidbar.

Beweis:

Da L_i entscheidbar, existiert eine TM $\mathcal{M}_i$ die L_i entscheidet.

Konstruiere 2-Band Turing-Maschine:

Entscheidbarkeit

Zeigen Sie: Sind die Sprachen L_1 und L_2 entscheidbar, dann ist auch die Sprache $L_1 \setminus L_2$ entscheidbar.

Beweis:

Da L_i entscheidbar, existiert eine TM $\mathcal{M}_i$ die L_i entscheidet.

Konstruiere 2-Band Turing-Maschine:

- Schreibe w auf beide Bänder
- Führe $\mathcal{M}_1$ auf dem ersten Band aus
- Führe $\mathcal{M}_2$ auf dem zweiten Band aus

Entscheidbarkeit

Zeigen Sie: Sind die Sprachen L_1 und L_2 entscheidbar, dann ist auch die Sprache $L_1 \setminus L_2$ entscheidbar.

Beweis:

Da L_i entscheidbar, existiert eine TM $\mathcal{M}_i$ die L_i entscheidet.

Konstruiere 2-Band Turing-Maschine:

- Schreibe w auf beide Bänder
- Führe $\mathcal{M}_1$ auf dem ersten Band aus
- Führe $\mathcal{M}_2$ auf dem zweiten Band aus
- Akzeptiere w, wenn $\mathcal{M}_1$ w akzeptiert und $\mathcal{M}_2$ w nicht akzeptiert.

Exkurs: Cantor und die Diagonalisierung

Satz: Die Menge $\mathbb{R}$ der reelen Zahlen ist überabzählbar.

Menge M abzählbar $\Leftrightarrow$ $\exists$ Bijektion zwischen $\mathbb{N}$ und M

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Exkurs: Cantor und die Diagonalisierung

Satz: Die Menge $\mathbb{R}$ der reelen Zahlen ist überabzählbar.

Annahme: $(0, 1) \subset \mathbb{R}$ ist abzählbar, dann existiert eine (unendliche) Folge (z_i) mit $z_i \in (0, 1)$

Menge M abzählbar $\Leftrightarrow \exists$ Bijektion zwischen $\mathbb{N}$ und M

Exkurs: Cantor und die Diagonalisierung

Satz: Die Menge $\mathbb{R}$ der reelen Zahlen ist überabzählbar.

Menge M abzählbar $\Leftrightarrow \exists$ Bijektion zwischen $\mathbb{N}$ und M

Annahme: $(0,1) \subset \mathbb{R}$ ist abzählbar, dann existiert eine (unendliche) Folge (z_i) mit $z_i \in (0,1)$

	0	1	2	3	4	5	$\cdots$	k	$\cdots$
z_1	0,	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$	$a_{1,5}$	$\cdot$	$a_{1,k}$	$\cdot$
z_2	0,	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$a_{2,4}$	$a_{2,5}$	$\cdot$	$a_{2,k}$	$\cdot$
z_3	0,	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$	$a_{3,4}$	$a_{3,5}$	$\cdot$	$a_{3,k}$	$\cdot$
z_4	0,	$a_{4,1}$	$a_{4,2}$	$a_{4,3}$	$a_{4,4}$	$a_{4,5}$	$\cdot$	$a_{4,k}$	$\cdot$
z_5	0,	$a_{5,1}$	$a_{5,2}$	$a_{5,3}$	$a_{5,4}$	$a_{5,5}$	$\cdot$	$a_{5,k}$	$\cdot$
$\vdots$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$
z_k	0,	$a_{k,1}$	$a_{k,2}$	$a_{k,3}$	$a_{k,4}$	$a_{k,5}$	$\cdot$	$a_{k,k}$	$\cdot$
$\vdots$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$
x									

Exkurs: Cantor und die Diagonalisierung

Satz: Die Menge $\mathbb{R}$ der reelen Zahlen ist überabzählbar.

Menge M abzählbar $\Leftrightarrow \exists$ Bijektion zwischen $\mathbb{N}$ und M

Annahme: $(0,1) \subset \mathbb{R}$ ist abzählbar, dann existiert eine (unendliche) Folge (z_i) mit $z_i \in (0,1)$

	0	1	2	3	4	5	$\cdots$	k	$\cdots$
z_1	0,	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$	$a_{1,5}$	$\cdot$	$a_{1,k}$	$\cdot$
z_2	0,	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$a_{2,4}$	$a_{2,5}$	$\cdot$	$a_{2,k}$	$\cdot$
z_3	0,	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$	$a_{3,4}$	$a_{3,5}$	$\cdot$	$a_{3,k}$	$\cdot$
z_4	0,	$a_{4,1}$	$a_{4,2}$	$a_{4,3}$	$a_{4,4}$	$a_{4,5}$	$\cdot$	$a_{4,k}$	$\cdot$
z_5	0,	$a_{5,1}$	$a_{5,2}$	$a_{5,3}$	$a_{5,4}$	$a_{5,5}$	$\cdot$	$a_{5,k}$	$\cdot$
$\vdots$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$
z_k	0,	$a_{k,1}$	$a_{k,2}$	$a_{k,3}$	$a_{k,4}$	$a_{k,5}$	$\cdot$	$a_{k,k}$	$\cdot$
$\vdots$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$
x									

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Exkurs: Cantor und die Diagonalisierung

Satz: Die Menge $\mathbb{R}$ der reelen Zahlen ist überabzählbar.

Menge M abzählbar $\Leftrightarrow$ $\exists$ Bijektion zwischen $\mathbb{N}$ und M

Annahme: $(0, 1) \subset \mathbb{R}$ ist abzählbar, dann existiert eine (unendliche) Folge (z_i) mit $z_i \in (0, 1)$

	0	1	2	3	4	5	•••	k	•••
z_1	0,	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$	$a_{1,5}$	•	$a_{1,k}$	•
z_2	0,	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$a_{2,4}$	$a_{2,5}$	•	$a_{2,k}$	•
z_3	0,	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$	$a_{3,4}$	$a_{3,5}$	•	$a_{3,k}$	•
z_4	0,	$a_{4,1}$	$a_{4,2}$	$a_{4,3}$	$a_{4,4}$	$a_{4,5}$	•	$a_{4,k}$	•
z_5	0,	$a_{5,1}$	$a_{5,2}$	$a_{5,3}$	$a_{5,4}$	$a_{5,5}$	•	$a_{5,k}$	•
$\vdots$	•	•	•	•	•	•	•	•	•
z_k	0,	$a_{k,1}$	$a_{k,2}$	$a_{k,3}$	$a_{k,4}$	$a_{k,5}$	•	$a_{k,k}$	•
$\vdots$	•	•	•	•	•	•	•	•	•
x	0,	$\overline{a_{1,1}}$	$\overline{a_{2,2}}$	$\overline{a_{3,3}}$	$\overline{a_{4,4}}$	$\overline{a_{5,5}}$	•	$\overline{a_{k,k}}$	•

$\overline{a_{i,i}} \neq a_{i,i}$

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Exkurs: Cantor und die Diagonalisierung

Satz: Die Menge $\mathbb{R}$ der reelen Zahlen ist überabzählbar.

Menge M abzählbar $\Leftrightarrow \exists$ Bijektion zwischen $\mathbb{N}$ und M

Annahme: $(0,1) \subset \mathbb{R}$ ist abzählbar, dann existiert eine (unendliche) Folge (z_i) mit $z_i \in (0,1)$

	0	1	2	3	4	5	$\cdots$	k	$\cdots$
z_1	0,	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$	$a_{1,5}$	$\cdot$	$a_{1,k}$	$\cdot$
z_2	0,	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$a_{2,4}$	$a_{2,5}$	$\cdot$	$a_{2,k}$	$\cdot$
z_3	0,	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$	$a_{3,4}$	$a_{3,5}$	$\cdot$	$a_{3,k}$	$\cdot$
z_4	0,	$a_{4,1}$	$a_{4,2}$	$a_{4,3}$	$a_{4,4}$	$a_{4,5}$	$\cdot$	$a_{4,k}$	$\cdot$
z_5	0,	$a_{5,1}$	$a_{5,2}$	$a_{5,3}$	$a_{5,4}$	$a_{5,5}$	$\cdot$	$a_{5,k}$	$\cdot$
$\vdots$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$
z_k	0,	$a_{k,1}$	$a_{k,2}$	$a_{k,3}$	$a_{k,4}$	$a_{k,5}$	$\cdot$	$a_{k,k}$	$\cdot$
$\vdots$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$
x	0,	$\overline{a_{1,1}}$	$\overline{a_{2,2}}$	$\overline{a_{3,3}}$	$\overline{a_{4,4}}$	$\overline{a_{5,5}}$	$\cdot$	$\overline{a_{k,k}}$	$\cdot$

$x \in \mathbb{R},\ x \neq z_i$ ↯

$\overline{a_{i,i}} \neq a_{i,i}$