



## Berechenbarkeit und Komplexität – WS 2022/2023

### Heimübung 10

**Abgabe: 9. Januar 2023 – 13:00 Uhr**

(Dieser Übungszettel besteht aus 4 Aufgaben mit insgesamt 24 Punkten)

#### Aufgabe 1 (NP-Vollständigkeit)

(6 Punkte)

Alle Jahre wieder sollen zu Weihnachten alle Menschen der Welt beschenkt werden. Die beschenkende Person will das möglichst energieeffizient gestalten und jedes der  $n$  Häuser genau einmal besuchen. Hierbei werden die Häuser als Knoten eines Graphen und die Verbindungen zwischen den Häusern als dessen Kanten betrachtet. In Aufgabe 2 der Präsenzübung 10 aufgepasst, weiß die Person, dass das Problem einen Hamiltonkreis zu finden NP-vollständig ist. Da sie von jedem Punkt der Welt zurück zum Nordpol gelangen kann, verzichtet sie darauf am Startpunkt zu enden. Beweisen Sie, dass auch das Finden von *Hamiltonpfaden* NP-vollständig ist.

$$\text{HAMILTONPATH} = \left\{ \langle G \rangle \mid \begin{array}{l} G \text{ ist ein ungerichteter Graph} \\ \text{in dem es einen Hamiltonpfad gibt.} \end{array} \right\}$$

#### Lösung:

Wir zeigen;

1.  $\text{HAMILTONPATH} \in NP$
2.  $\text{HAMILTONKREIS} \leq_p \text{HAMILTONPATH}$

Beweis von 1.

Betrachte folgende DTM  $V$

$V$  bei Eingabe  $x \in \{0, 1\}^*$

1. (Formcheck) Prüfe, ob  $x = \langle G, P \rangle$ , wobei  $G$  die Kodierung eines ungerichteten Graphen ist und  $P$  eine Permutation der Knotenmenge von  $G$  ist.
2. Überprüfe in Permutationsreihenfolge, ob die Kante  $\{i, i + 1\}$  in der Kantenmenge enthalten sind. Falls dies für eine der Kanten nicht der Fall ist lehne ab.
3. Akzeptiere

**Laufzeit von  $V$ :**

1. Zu prüfen, ob  $G$  in Graph ist ist trivialerweise in polynomieller Zeit möglich. Das selbe gilt für  $P$ , da hierbei maximal  $|V|^2$  Vergleich durchgeführt werden müssen.
2. In polynomieller Zeit möglich, da hierbei nur maximal  $|V|$  Kanten betrachtet werden.

### Polynomielle Größe von $P$ :

Da  $P$  eine Permutation der Knotenmenge ist hat diese eine feste Länge von  $|V|$  und ist damit trivialerweise polynomiell

Behauptung:  **$V$  ist ein Verifizierer der Sprache HamiltonPath.**

Angenommen  $x \in \text{HamiltonPath}$

$\implies x = \langle G \rangle$  mit  $G$  besitzt einen Hamiltonpfad  $P$  (in Form einer Permutation ansonsten wandel diese in einen um).

$\implies \langle G, P \rangle \in L(V)$ , da nach der Definition eines Hamiltonpfads alle Kanten der Permutation in der Kantenmenge des Graphen befinden (und die Eingabe das richtige Format hat).

Angenommen  $x \notin \text{HamiltonPath}$

a)  $x$  von der falschen Form, dann wird  $x$  in 1 abgelehnt.

b)  $x = \langle G \rangle$  enthält, aber keinen Hamiltonpfad.

Angenommen es würde eine Permutation  $F$  geben, s.d:  $\langle G, F \rangle \in L(V)$

Dann ist nach 2) jede auf einander folgende Kante der Permutation in  $G$  vorhanden.

Folglich ist die Permutation  $F$  zu einem Pfad in  $G$  isomorph, welche jede Knoten genau einmal besucht  $\nexists x \notin \text{HamiltonPath}$ .

$\implies V$  ist ein Verifizierer der Sprache HamiltonPath.

$\implies \text{HamiltonPath} \in NP$

Zu 2)

Behauptung **Hamiltonkreis  $\leq_p$  HamiltonPath**

Betrachte folgende Reduktionsfunktion:

$$f(w) = \begin{cases} \langle G_w \rangle & \text{wenn } w = \langle G \rangle, \text{ wobei } \langle G \rangle \text{ die Kodierung eines Graphen ist} \\ \langle G_0 \rangle & \text{wenn } w \neq \langle G \rangle \text{ sonst} \end{cases}$$

mit  $G = (V, E)$  und  $u_o \in V$ ,  $x, s, t \notin V$

Sei  $G_w = (V_w, E_w)$  mit  $V_w = V \cup \{u_0, s, t\}$

und  $E_w = E \cup \{\{u_0, v\} \mid \{u, v\} \in E\} \cup \{\{u_0, t\}, \{s, u\}, \{u, u_0\}\}$

Sei  $G_0 = (V_0, E_0)$  mit  $V_0 = V \cup \{x\}$  und  $E_0 = E$

Behauptung **f ist polynomiell**:

Zu Prüfen, ob eine Eingabe der Kodierung eines Graphen entspricht ist trivialerweise in polynomieller Zeit berechenbar möglich.

Falls  $w = \langle G \rangle$ , so werden 3 Knoten und maximal  $|V| + 3$  Kanten hinzugefügt. Dies ist offensichtlich in polynomieller Zeit möglich und berechenbar.

Falls  $w \neq \langle G \rangle$ , so wird ein neuer Knoten hinzugefügt. Dies ist auch in polynomieller Zeit möglich und berechenbar.

$\Rightarrow f$  ist in polynomieller Zeit berechenbar.

Korrektheit:

Angenommen  $w \in \text{Hamiltonkreis}$ .

Folglich wird  $w$  unter  $f$  auf den Graphen  $G_w$  abgebildet.

Da  $G$  einen Hamiltonkreis besitzt, kann in  $G$  ein Pfad gefunden werden, welcher von  $u$  nach  $u$  geht und jeden Knoten des Graphen genau einmal enthält. Sei nun  $P = (u, \dots, v)$  dieser Pfad in der Form eines Tupels. Betrachte nun  $P_w = (s, u, \dots, v, u_0, t)$ . Dieser ist ein Hamiltonpfad in  $G_w$ , da nach Konstruktion die zusätzlichen Kanten, welche für die Konstruktion dieses Pfades nötig sind, in  $E_w$  enthalten sind.

$\Rightarrow f(w) \in \text{HamiltonPath}$ .

Angenommen  $f(w) \in \text{HamiltonPath}$ .

Sei  $P_w$  dieser Pfad in der Form einer Permutation.

$\Rightarrow P_w = (s, u, \dots, v, u_0, t)$  oder  $P_w = (t, u_0, \dots, v, u, s)$ , da  $\deg(s) = 1$  und  $\deg(t) = 1$ . (o.B.d.A.  $P_w = (s, u, \dots, v, u_0, t)$ )

Betrachte  $P = (u, \dots, v)$  in  $G$ . Dieser ist, zum einen ein Pfad in  $G$ , da die Kanten dieses Pfades unter der Abbildung in  $G_w$  nicht abgeändert wurden. Da  $u_0$  nach Konstruktion genau die selben Kanten besitzt, wie  $u$ , muss  $u$  auch eine Kante zu  $v$  besitzen. Folglich ist  $P$  also auch ein Kreis in  $G$ . Da  $P_w$  darüberhinaus auch

ein Hamiltonpfad ist, muss  $P$  als Teilmenge dessen auch ein Hamiltonpfad auf der verringerten Knotenmenge sein. Da diese, aber genau die Knotenmenge von  $V$  ist und  $P$  ein Kreis ist, folgt hiraus, dass  $P$  ein Hamiltkreis ist.

$\implies w \in \text{Hamiltonkreis}.$

$\implies \text{Hamiltonkreis} \leq_p \text{HamiltonPath}$

$\xrightarrow{(1),(2)} \text{HamiltonPath}$  ist NP-Vollständig  $\square$

## Aufgabe 2 (NP-Vollständigkeit)

(6 Punkte)

In der Vorlesung haben Sie einen relativ aufwendigen Beweis für die NP-Vollständigkeit der Sprache INDEPENDENTSET durch polynomielle Reduktion von 3SAT kennengelernt. Machen Sie Ihrem Professor ein Weihnachtsgeschenk, indem Sie einen einfacheren (aber korrekten) Beweis der NP-vollständigkeit von INDEPENDENTSET erstellen. Nutzen Sie hierfür eine andere, besser geeignete, in der Vorlesung als NP-vollständig bewiesene Sprache.

### Lösung:

Es ist zu zeigen, dass IndependentSet NP vollständig ist. Dies führen wir in den folgende 2 Schritten durch:

1. INDEPENDANTSET  $\in NP$
2. KNOTENÜBERDECKUNG  $\leq_p$  INDEPENDENTSET

Zu 1) (Dies wurde in der Vorlesung als trivial vorausgesetzt und deswegen nicht bewiesen. Um sicher zugehen, dass wir keine Punkte verlieren, beweisen wir das gerade noch vorab).

Betrachte folgende DTM  $V$

$V$  bei Eingabe  $x \in \{0, 1\}^*$

1. (Formcheck) Prüfe, ob  $x = \langle G, k, U \rangle$ , wobei  $G$  die Kodierung eines ungerichteten Graphen ist,  $k \in \mathbb{N}$  und  $U \subset V$  mit  $|U| = k$
2. Überprüfe, ob  $\forall u, v \in U: \{u, v\} \notin E$ . Falls dies nicht der Fall ist so lehne ab.
3. Akzeptiere.

### Laufzeit von $V$ :

1. Zu prüfen, ob  $G$  in Graph ist ist trivialerweise in polynomieller Zeit möglich. Selbiges gilt für  $k$  und  $U$ , da  $k$  eine natürliche Zahl ist und  $U$  eine endliche Teilmenge einer endlichen Menge ist.

2. In polynomieller Zeit möglich, da hierbei nur maximal  $|U|^2$  Kanten betrachtet werden.

### Polynomielle GrösSe von $U$ :

Da  $U$  eine Teilmenge der Knotenmenge ist hat diese eine maximale GrösSe von  $|V|$  und ist damit trivialerweise polynomiell.

Behauptung: **V ist ein Verifizierer der Sprache IndependentSet.**

Angenommen  $x \in \text{IndependentSet}$

$\implies x = \langle G, k \rangle$  mit  $G$  besitzt eine unabhängige Menge  $U$  mit  $|U| = k$

$\implies \langle G, k, U \rangle \in L(V)$ , da nach der Definition einer unabhängigen Menge die Bedingung 2 erfüllt ist, also in 3. akzeptiert wird.

Angenommen  $x \notin \text{IndependentSet}$

a)  $x$  von der falschen Form, dann wird  $x$  in 1 abgelehnt.

b)  $x = \langle G, k \rangle$  enthält, aber keinen unabhängigen Menge mit Kardinalität  $k$ .

Angenommen es würde eine Menge  $U$  geben, s.d:  $\langle G, k, U \rangle \in L(V)$

Dann gilt nach 2, dass  $U$  eine unabhängige Menge mit  $|U| = k$  ist (nach Def).  $\nexists x \notin \text{IndependentSet}$ .

$\implies V$  ist ein Verifizierer der Sprache IndependentSet.

$\implies \text{IndependentSet} \in NP$

Zu 2)

Behauptung: **Knotenüberdeckung  $\leq_p$  IndependentSet.**

Betrachte folgende Reduktionsfunktion:

$$f(w) = \begin{cases} \langle G, |V| - k \rangle & \text{wenn } w = \langle G \rangle, \text{ mit } \langle G, k \rangle \text{ ist die Kodierung eines Graphen und } k \in \mathbb{N} \\ \langle G_0 \rangle & \text{wenn } w \neq \langle G \rangle \text{ sonst} \end{cases}$$

Sei  $G_0 = (V_0, E_0)$  mit  $V_0 = V$  und  $E_0 = \{u, v\}$  mit  $u, v \in V$

Die Laufzeit von  $f(w)$  ist trivialerweise polynomiell.

Behauptung:  $w \in \text{Knotenüberdeckung} \iff f(w) \in \text{IndependentSet}$

Angenommen  $w \in \text{Knotenüberdeckung}$ .

$\implies \exists U \subset V$  s.d.  $U$  eine Knotenüberdeckung des Graphen ist und darüberhinaus gilt:  $|U| = k$ .

**Behauptung:**  $U^C := V \setminus U$  ist eine unabhängige Menge.

Beweis durch Widerspruch. Angenommen  $U^C$  wäre keine unabhängige Menge, d.h.  $\exists u, v \in U^C$  s.d.  $\{u, v\} \in E$ .

$\implies u, v \in U$ , da  $U$  eine Knotenüberdeckung ist.  $\nmid u, v \in U^C \quad \square$

$\implies f(w) \in \text{IndependentSet}$ , da  $U^C$  eine Kardinalität von  $|V| - k$  hat.

Angenommen  $f(w) \in \text{IndependentSet}$ .

$\implies \exists U \subset V$  s.d.  $U$  eine IndependentSet des Graphen ist und darüberhinaus gilt:  $|U| = |V| - k$ .

**Behauptung:**  $U^C := V \setminus U$  ist eine  $k$ -Knotenüberdeckung.

Beweis durch Widerspruch. Angenommen  $U^C$  wäre keine Knotenüberdeckung.

$\implies \exists e \in E$  s.d.  $e \cap U^C = \emptyset$ . Sei  $\{u, v\}$  mit  $u, v \in V$  diese Kante.

$\implies$  Nach der Definition von einer Unabhängigen Menge können  $u, v$  nicht in  $U$  sein

$\implies u, v \in U^C$  also gilt  $e \cap U^C \neq \emptyset \quad \square$

$\implies w \in \text{Knotenüberdeckung}$ , da  $U^C$  eine Kardinalität von  $k$  hat.

$\implies w \in \text{Knotenüberdeckung} \iff f(w) \in \text{IndependentSet}$

$\xrightarrow{(1),(2)} \text{IndependentSet}$  ist NP-Vollständig  $\quad \square$

### Aufgabe 3 (Rekursive Aufzählbarkeit)

(6 Punkte)

Einem Weihnachtself, der seine Hausaufgaben nicht erledigt hat, wird eine FleiSSaufgabe gestellt: Er soll für eine der beiden folgenden Sprachen ein Programm schreiben, welches alle enthaltenen Worte nacheinander aufzählt. Helfen Sie bei der Entscheidungsfindung, indem Sie für jede der beiden folgenden Sprachen beweisen, ob sie rekursiv aufzählbar ist, oder nicht.

$L_1 = \{\langle M \rangle \mid M \text{ lehnt mindestens eine Eingabe nicht ab.}\}$

$L_2 = \{\langle M \rangle \mid M \text{ lehnt mindestens eine Eingabe ab.}\}$

**Lösung:**

Behauptung  $L_1$  ist nicht rekursiv aufzählbar:

Wir definieren die Reduktionsfunktion  $f : \{0, 1, \}^* \rightarrow \{0, 1, \}^*$ :

$$f(w) = \begin{cases} \langle M^{(reject)} \rangle & \text{wenn } w \neq \langle M \rangle x \\ \langle M^{(x)} \rangle & \text{wenn } w = \langle M \rangle x \end{cases}$$

Mit  $\langle M^{(reject)} \rangle$  sei die Turingmaschine die jede Eingabe ablehnt. und  $\langle M^{(x)} \rangle$  sei eine leicht abgeänderte Turingmaschine aus Satz 2.10.1, welche wie folgt definiert ist:

$\langle M^{(x)} \rangle$  bei Eingabe  $z \in \{0, 1\}^*$

1. Berechne  $|z|$ .
2. Simuliere  $M$  bei Eingabe  $x$  für  $|z|$  Schritte.
3. Falls  $M$  in  $|z|$  Schritten hält, so akzeptiere  $x$ .
4. Sonst lehne  $z$  ab.

Folglich ist  $f(w)$  nach diesem Satz 2.10.1 auch berechenbar, da hierbei nur die Kontraposition der Annahmebedingung gebildet wurde.

Darüber hinaus gilt die Kontraposition der Äquivalenz 2.2:

$M$  hält bei Eingabe  $x$  nicht  $\iff M^{(x)}$  hält bei keiner Eingabe  $z \in \{0, 1\}^*$  (1)

Z.z.  $(w \in \overline{H} \iff f(w) \in L)$

Richtung  $\implies$ : Angenommen  $w \in \overline{H}$ .

Das heißt,  $w = \langle M \rangle x$  mit  $M$  hält bei Eingabe  $x$  nicht.

Also wird  $w$  unter  $f$  nach  $\langle M^{(x)} \rangle$  abgebildet.

Da  $M$  bei keiner Eingabe hält, gilt nach (1), dass  $\langle M^{(x)} \rangle$  keine Eingabe akzeptiert, was eine endliche Anzahl ist.

Damit ist  $f(w) = \langle M^{(x)} \rangle \in L$ .

Richtung  $\impliedby$ : Angenommen  $w \notin \overline{H}$ .

Also ist entweder  $w$  von falscher Form oder  $M$  hält bei Eingabe  $x$ .

Ist  $w$  von falscher Form, dann ist  $f(w) = \langle M^{(reject)} \rangle \notin L$ , da  $\langle M^{(reject)} \rangle$  jede Eingabe ablehnt, dem entsprechend auch bei jeder Eingabe hält.

Ist  $w = \langle M \rangle x$  und  $M$  hält bei Eingabe  $x$ , dann:

Es existiert ein  $n$  mit  $M$  hält bei Eingabe  $x$  nach  $n$  Schritten.

Und für alle  $n' > n$  gilt:  $M$  hält bei Eingabe  $x$  nach  $n'$  Schritten.

Damit wird  $\langle M^{(x)} \rangle$  bei jeder Eingabe der Länge  $n' > n$  halten und die jeweilige Eingabe akzeptieren.

Also gilt  $L(\langle M^{(x)} \rangle) = \{z | z \in \{0, 1\}^* \text{ mit } |n| \leq |z|\}$

Damit ist  $f(w) = \langle M^{(x)} \rangle \notin L$ .

$\Rightarrow \overline{H} \leq L \quad \square$

---

Behauptung  $L_2$  ist rekursiv aufzählbar.

Betrachte folgende Turingmaschine  $M_{L_2}$

$M_{L_2}$  bei Eingabe  $x$  aus  $\{0, 1\}^*$

1. (Formcheck) Prüfe, ob es sich bei  $x = \langle M \rangle$  um eine Gödelnummer handelt
2. Für jede natürliche Zahl  $n$  (also bei 1 anfangen und hochzählen)
  - a) Für jede Eingabe  $w$  mit  $|w| \leq n$ :
    - i. Simuliere  $M$  bei Eingabe  $w$  für  $n$  Schritte.
    - ii. Falls  $M$  bei Eingabe  $w$  nach  $n$  Schritten hält und ablehnt, so akzeptiere.

Falls  $M \in L_2$ , dann existiert eine Eingabe  $w$ , sodass  $M$  bei Eingabe  $w$  mit  $|w| = m$  nach  $n$  Schritten hält und ablehnt.

Die Turingmaschine hält dann in der  $\max\{m, n\}$ -ten Iteration der  $n$ -Schleife.

Jede Iteration für ein bestimmtes  $n$  hat endlich viele Schritte, da einzelne Turingmaschinen maximal für  $n$  Schritte simuliert werden.

Ist  $M \notin L_2$ , dann ist  $M$  entweder nicht von der Form  $\langle M \rangle$  (Formcheck).

Oder  $M$  hält und lehnt bei keiner Eingabe ab. In dem Fall Terminiert  $M_{L_2}$  nicht, (und akzeptiert auch nicht).

Damit ist  $L_2$  rekursiv aufzählbar.



**Aufgabe 4** (2-Band-DTM formal)

(6 Punkte)

Entwerfen Sie eine *formelle* Beschreibung einer *2-Band-DTM*, welche für eine Eingabe  $w \in \{0, 1\}^*$  entscheidet, ob  $w = 1^n 0^{2^n}$  für ein  $n \in \mathbb{N}$  gilt. Beschreiben Sie hierfür zunächst intuitiv die einzelnen Schritte Ihrer 2-Band-DTM und nennen Sie die zugehörigen Zustände. Als ersten Schritt fordern wir, dass die DTM prüft ob die Eingabe ausschließlich aus Einsen gefolgt von Nullen besteht. Während dieses Schrittes sollen die Nullen vom ersten auf das zweite Band kopiert werden. Der unten vorgegebene Zustand  $q_0$  rückt den Lesekopf auf Band 1 bis zu ersten 0 vor. Für Ihre Lösung dürfen Sie *maximal 10 Zustände* verwenden, wobei beispielsweise 6 Zustände bereits ausreichen können. Beachten Sie folgende Punkte.

- Eine Mehrband-DTM hat neben den üblichen Bewegungsrichtungen  $R, L$  die Option  $S$  für Stehenbleiben.
- Für alle Kombinationen aus Zeichen und Zuständen, für die kein Übergang angegeben ist, lehnt die DTM ab.
- Sie können für jeden Zustand eine eigene Tabelle angeben. Das kann sinnvoll sein, wenn für einen Zustand  $q_i$  andere Eingaben als für Zustand  $q_j$  relevant sind.
- In der Tabelle können Variablen verwendet werden, wobei definiert sein muss, welche Werte diese annehmen könnten. Ist zum Beispiel die Variable  $x$  für die Menge  $\{0, 1, \sqcup\}$  und für Zustand  $q_i$  ein Übergang von  $(x, 0)$  definiert, so gilt dieser Übergang für alle  $x \in \{0, 1, \sqcup\}$ .
- Das Eingabewort ist zu Beginn auf Band 1 direkt hinter dem Startsymbol abgelegt und das andere Band ist (abgesehen vom Startsymbol) leer.

Der folgende Startzustand  $q_0$  ist Ihnen vorgegeben:

**1. Schritt:** Gehe auf Band 1 bis zur ersten 0.

| $\delta$ | $(\triangleright, \triangleright)$          | $(1, \sqcup)$          | $(0, \sqcup)$          |
|----------|---|------------------------|------------------------|
| $q_0$    | $q_0, \triangleright, \triangleright, R, R$ | $q_0, 1, \sqcup, R, S$ | $q_1, \sqcup, 0, R, R$ |

**Lösung:**

Sei im folgenden  $M = (Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_{\text{accept}}, q_{\text{reject}}\}, \Sigma = \{0, 1\}, \Gamma = \{\triangleright, 0, 1, \#, \sqcup\}, \delta)$  eine DTM mit folgender Übergangsfunktion:

**1. Schritt:** Gehe auf Band 1 bis zur ersten 0.

| $\delta$ | $(\triangleright, \triangleright)$          | $(1, \sqcup)$          | $(0, \sqcup)$          |
|----------|---|------------------------|------------------------|
| $q_0$    | $q_0, \triangleright, \triangleright, R, R$ | $q_0, 1, \sqcup, R, S$ | $q_1, \sqcup, 0, R, R$ |

**2. Schritt:** Kopiere 0 auf Band 2 und entferne diese von Band 1.

| $\delta$ | $(0, \sqcup)$          | $(\sqcup, \sqcup)$          |
|----------|------------------------|-----------------------------|
| $q_1$    | $q_1, \sqcup, 0, R, R$ | $q_2, \sqcup, \sqcup, S, L$ |

**3. Schritt:** Gehe zum Start von Band 2 zurück und auf Band 1 zur letzten 1.

|          |   |                                |                         |
|----------|---|--------------------------------|-------------------------|
| $\delta$ | $(\triangleright, \triangleright)$          | $(\triangleright, 0)$          | $(1, 0)$                |
| $q_2$    | $q_5, \triangleright, \triangleright, R, R$ | $q_2, \triangleright, 0, S, L$ | $q_2, 1, 0, L, L$       |
| $\delta$ | $(\sqcup, 0)$                               | $(1, \#)$                      | $(1, \triangleright)$   |
| $q_2$    | $q_2, \sqcup, 0, L, L$                      | $q_2, 1, \#, S, L$             | $q_3, \sqcup, \#, L, R$ |

**5. Schritt:** Streiche jede 2 Null vom zweiten Band.

|          |                    |                    |                        |                             |
|----------|--------------------|--------------------|------------------------|-----------------------------|
| $\delta$ | $(1, 0)$           | $(1, \#)$          | $(1, \sqcup)$          | $(\sqcup, \sqcup)$          |
| $q_3$    | $q_4, 1, \#, S, R$ | $q_3, 1, \#, S, R$ | $q_3, 1, \sqcup, S, R$ | $q_3, \sqcup, \sqcup, L, L$ |
| $q_4$    | $q_3, 1, 0, S, R$  | $q_4, 1, \#, S, R$ | $q_3, 1, \sqcup, S, R$ | $q_3, \sqcup, \sqcup, L, L$ |

**6. Schritt:** Überprüfe, ob nur noch eine Null auf dem 2 Band ist.

|          |                        |                         |                               |
|----------|------------------------|-------------------------|-------------------------------|
| $\delta$ | $(\sqcup, 0)$          | $(\sqcup, \#)$          | $(\sqcup, \sqcup)$            |
| $q_5$    | $q_4, \sqcup, 0, S, R$ | $q_4, \sqcup, \#, S, R$ | $q_{accept}, \sqcup, 0, L, L$ |