Technische Universität München
School of Computation, Information and Technology
Lehrstuhl für Algorithmen und Komplexität
Prof. Dr. Susanne Albers
Tobias Forner

Sommersemester 2025
Lösungsblatt 2
2. Mai 2025

Einführung in die Theoretische Informatik

Das Übungsblatt besteht neben der Vorbereitungsaufgabe aus zwei Teilen. (1) Die Tutoraufgaben werden in den Tutorien erarbeitet. (2) Die Hausaufgaben werden von den Studierenden in Abgabeteams gelöst. Korrigiert werden Hausaufgabe 1 (a), Hausaufgabe 2 (b) und Hausaufgabe 4.

Vorbereitungsaufgabe 1

Überprüfen Sie, dass Sie die folgenden Begriffe und Notationen korrekt definieren können.

- DFA, NFA
- Zustandsgraph
- akzeptierte Sprache
- Akeptanzbedingung von DFAs/NFAs

Tutoraufgabe 1

Wir betrachten die folgenden vier Sprachen über einem Alphabet Σ. Geben Sie jeweils einen DFA (graphisch) an, der genau die Sprache erkennt, die

(a) am Anfang jedes Wortes eine bestimmte Sequenz von Buchstaben fordert,

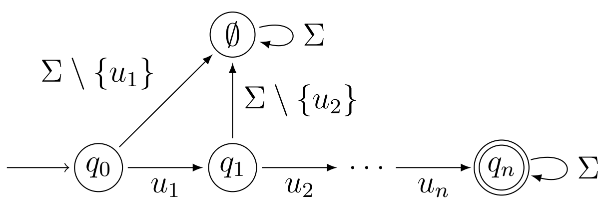
(b) alle Wörter gerader/ungerader Länge enthält,

(c) alle Wörter einer fixierten Länge n enthält,

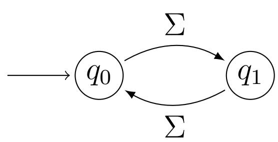
(d) alle Wörter enthält, die an einer fixierten Position einen bestimmten Buchstaben enthalten.

Lösungsvorschlag

(a) Sei $u_1, \ldots, u_n$ die gewünschte Sequenz. Wir konstruieren den folgenden DFA:

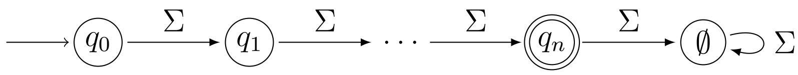


(b)

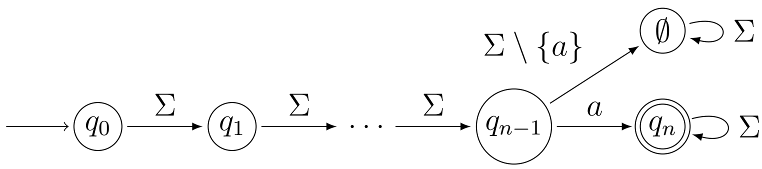


Wenn q_0 zu einem Endzustand gemacht wird, akzeptiert dieser DFA genau alle Wörter gerader Länge. Wird hingegen q_1 zum Endzustand gemacht, so akzeptiert er genau alle Wörter ungerader Länge.

(c) Der folgende DFA akzeptiert genau alle Wörter der Länge n:



(d) Sei a der gewünschte Buchstabe an Stelle n.



Tutoraufgabe 2

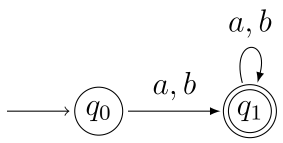
Sei $\Sigma = \{a, b\}$ und $B_n \coloneqq \{w \in \Sigma^* \mid \exists i : w_i = w_{i+n}\}$ die Sprache aller Wörter über Σ, in denen an irgendeiner Stelle der selbe Buchstabe im Abstand n vorkommt. Insbesondere ist B_0 die Menge aller nicht leeren Wörter, und B_1 die Menge aller Wörter in denen ein Buchstabe zweimal hintereinander vorkommt. *Versuchen Sie in beiden Aufgabenteilen NFAs und DFAs mit möglichst wenigen Zuständen anzugeben.*

(a) Geben Sie jeweils einen NFA für B_0, B_1 und B_2 an.

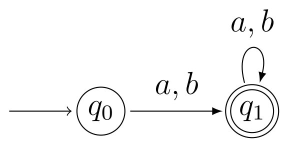
(b) Geben Sie jeweils einen DFA für B_0, B_1 und B_2 an.

(c) Beschreiben Sie kurz, wie der NFA B_n und der DFA B_n für beliebige $n \in \mathbb{N}$ aussehen.

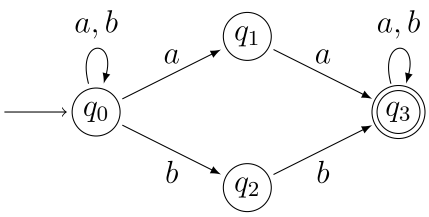
Lösungsvorschlag



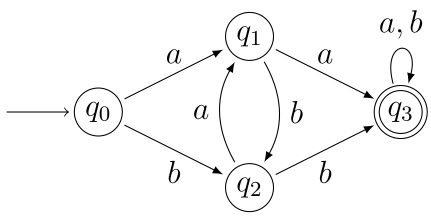
NFA für B_0



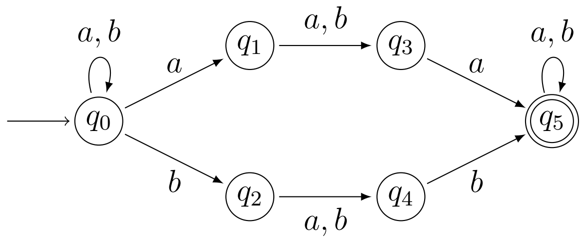
DFA für B_0



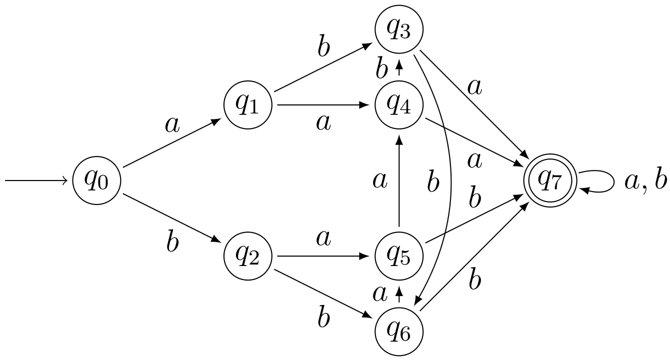
NFA für B_1



DFA für B_1



NFA für B_2



DFA für B_2

(c) Der NFA rät, ob jetzt das i-te Zeichen gelesen wird, welches die Bedingung erfüllt. Der DFA muss sich hingegen immer die letzten n gelesenen Zeichen merken, um zu überprüfen, ob die Bedingung $w_i = w_{i+n}$ erfüllt ist. Genauer besteht der DFA besteht aus einem Start- und einem Endzustand, sowie n "Spalten" zwischen diesen Zuständen. Zunächst liest der DFA n Zeichen ein und bewegt sich dabei in die jeweils nächste Spalte. Dabei merkt sich der DFA immer die letzten n Zeichen, die er gelesen hat, in seinem Zustand. In der n-ten Spalte verbleibt der Automat dann, bis entweder das Wort zu Ende ist (Wort wird abgelehnt) oder der nächste Buchstabe den ersten im Zustand gespeicherten Buchstaben entspricht (Wort wird akzeptiert).

Tutoraufgabe 3

Sei $M = (Q, \Sigma, \delta, q_0, F)$ ein deterministischer endlicher Automat. Wir betrachten Zustände $q_1, q_2, \ldots, q_{n+1} \in Q$ und Zeichen $a_1, a_2, \ldots, a_n \in \Sigma$. Nach Vorlesung, Folie 30, ist die relationale Notation $p \overset{a}{\to} q$ gleichbedeutend mit $\delta(p, a) = q$.
Zeigen Sie:

$$q_1 \overset{a_1}{\to} q_2 \overset{a_2}{\to} \ldots \overset{a_n}{\to} q_{n+1} \quad \Longrightarrow \quad \hat{\delta}(q_1, a_1 a_2 \ldots a_n) = q_{n+1}\,.$$

Lösungsvorschlag

Beweis per Induktion:

$\underline{n = 1:}$
$q_1 \overset{a_1}{\to} q_2$ bedeutet nach Definition $\delta(q_1, a_1) = q_2$. Damit folgt $\hat{\delta}(q_1, a_1) = \delta(q_1, a_1) = q_2$.

$n \to n+1$, für alle $n \geq 1$:

Wir nehmen an, dass die Implikation für n, mit $n \geq 1$, und alle möglichen Eingaben a_i und Zustände q_i gilt.

Sei $q_1 \overset{a_1}{\to} q_2 \overset{a_2}{\to} \dots \overset{a_{n+1}}{\to} q_{n+2}$.

Nun zeigen wir $\hat{\delta}(q_1, a_1 a_2 \dots a_{n+1}) = q_{n+2}$ wie folgt.

$$\begin{aligned} & q_1 \overset{a_1}{\to} q_2 \overset{a_2}{\to} \dots \overset{a_{n+1}}{\to} q_{n+2} \\ \Longrightarrow\ & q_1 \overset{a_1}{\to} q_2 \quad \wedge \quad q_2 \overset{a_2}{\to} \dots \overset{a_{n+1}}{\to} q_{n+2} \\ \Longrightarrow\ & \delta(q_1, a_1) = q_2 \quad \wedge \quad \hat{\delta}(q_2, a_2 \dots a_{n+1}) = q_{n+2} \\ \Longrightarrow\ & \hat{\delta}(\delta(q_1, a_1), a_2 \dots a_{n+1}) = q_{n+2} \\ \Longrightarrow\ & \hat{\delta}(q_1, a_1 a_2 \dots a_{n+1}) = q_{n+2}\,. \end{aligned}$$

Tutoraufgabe 4

Entscheiden Sie, ob die folgenden Aussagen korrekt sind, und begründen Sie Ihre Behauptung, indem Sie entweder ein Gegenbeispiel oder eine passende Konstruktion angeben.

Für jeden NFA $N = (Q, \Sigma, \delta, q_0, F)$ gibt es einen NFA $N' = (Q', \Sigma, \delta', q_0', F')$ mit $L(N) = L(N')$ und ...

(a) der Startzustand hat keine eingehenden Kanten.

(b) kein Endzustand hat eine ausgehende Kante.

(c) für jeden Zustand q gilt: alle eingehenden Kanten von q sind mit demselben Zeichen beschriftet.

(d) für jeden Zustand q gilt: alle ausgehenden Kanten von q sind mit demselben Zeichen beschriftet.

Lösungsvorschlag

(a) Ja: Man erstellt vom Startzustand q_0 eine Kopie q_0', die nur die ausgehenden Kanten von q_0 erbt. Danach wird q_0 zu einem normalen Zustand, während q_0' zu einem Startzustand wird. Formal: $N' = (Q', \Sigma, \delta', q_0', F')$, wobei

- $Q' = Q \cup \{q_0'\}$,
- $\delta' = \delta \cup \{(q_0', x, q) : (q_0, x, q) \in \delta\}$
- $F' = F \cup \{q_0' : q_0 \in F\}$

(b) Nein: Betrachte einen NFA N für $L = \{\epsilon, a\}$.

(c) Ja: man spaltet jeden Zustand $q \in Q$ nach dem Buchstaben der eingehenden Kanten auf, d.h. aus q macht man die Zustände (q, x) für jedes $x \in \Sigma$.

Danach setzt man $\delta'((q, x), y, (q', y)) := \delta(q, y, q')$.

Wird dabei der Startzustand aufgespalten, wählt man einen beliebigen daraus hervorgehenden Zustand als neuen Startzustand.

(d) Nein: Betrachte einen NFA N mit $L(N) = \{a, b\} = \Sigma$. Dann gilt $\delta(q_0, a) \cap F \neq \emptyset$ und $\delta(q_0, b) \cap F \neq \emptyset$.

Tutoraufgabe 5

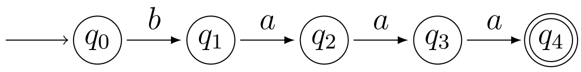
Sei $\Sigma = \{a, b\}$ und $L = \{ba^n\}$ für ein beliebiges aber festes $n \in \mathbb{N}_0$.

1. Sei $n = 3$. Zeichnen Sie einen NFA, der L erkennt.

2. Zeigen sie, dass jeder NFA, der die Sprache L erkennt, mindestens $n + 2$ Zustände hat.

Lösungsvorschlag

(a) Skizze:



(b) *Beweis.* Annahme zum Widerspruch: Es gibt einen Automaten $M = (Q, \Sigma, \delta, q_0, F)$ mit weniger als $n+2$ Zuständen, der L akzeptiert. Da M das Wort $w = ba^n$ akzeptiert und $|ba^n| = n + 1$, muss es einen akzeptierenden Lauf $L = s_0 \xrightarrow{b} s_1 \xrightarrow{a} \ldots \xrightarrow{a} s_{n+1} \in F$ geben wobei $s_0 = q_0$. Es gibt aber höchstens $n + 1$ Zustände, daher folgt nach Schubfachprinzip, dass mindestens ein Zustand s_i durch den Lauf zweimal besucht wird. Dadurch gibt es einen Kreis $s_i \xrightarrow{w_i} s_{i+1} \xrightarrow{w_{i+1}} \cdots \xrightarrow{w_{j-1}} s_j$ in L, wobei $s_i = s_j$. Wenn wir den Kreis abkürzen, d.h. den Lauf $s_0 \xrightarrow{w_0} \ldots \xrightarrow{w_{i-1}} s_i \xrightarrow{w_j} s_{j+1} \xrightarrow{w_{j+1}} \ldots \xrightarrow{w_n} s_{n+1} \in F$ betrachten, dann ist dies auch ein akzeptierender Lauf. Im Spezialfall $j = n + 1$ ist $s_0 \xrightarrow{w_0} \ldots \xrightarrow{w_{i-1}} s_i = s_{n+1} \in F$ ein akzeptierender Lauf. Deshalb akzeptiert M ein Wort w' mit $|w'| < n + 1$, was ein Widerspruch zur Annahme ist, dass M die Sprache L akzeptiert. □

Hausaufgabe 1 (5 Punkte)

Geben Sie einen deterministischen endlichen Automaten (als Graph und tabellarischer Übergangsfunktion) an, der über dem Alphabet $\Sigma = \{0, 1\}$ die folgende Sprache akzeptiert.

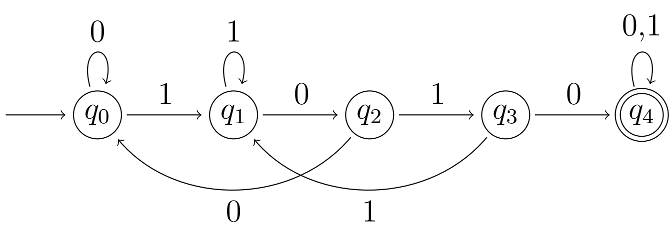
(a) Die Menge aller Wörter, die das Teilwort 1010 enthalten.

(b) Die Menge aller Wörter, bei denen die Anzahl der Einsen durch 3 teilbar ist.

Lösungsvorschlag

(a) Die Menge aller Wörter, die das Teilwort 1010 enthalten, werden akzeptiert vom Automaten $M_1 = (\{q_0, q_1, q_2, q_3, q_4\}, \Sigma, \delta, q_0, \{q_4\})$.

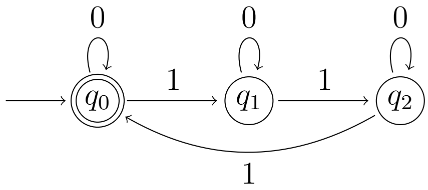
Graph und Übergangsfunktion:



q_i	$\delta(q_i, 0)$	$\delta(q_i, 1)$
q_0	q_0	q_1
q_1	q_2	q_1
q_2	q_0	q_3
q_3	q_4	q_1
q_4	q_4	q_4

(b) Die Menge aller Wörter, bei denen die Anzahl der Einsen durch 3 teilbar ist, werden akzeptiert vom Automaten $M_2 = (\{q_0, q_1, q_2\}, \Sigma, \delta, q_0, \{q_0\})$.

Graph und Übergangsfunktion:



q_i	$\delta(q_i, 0)$	$\delta(q_i, 1)$
q_0	q_0	q_1
q_1	q_1	q_2
q_2	q_2	q_0

Hausaufgabe 2 (5 Punkte)

Sei $M = (Q, \Sigma, \delta, q_0, F)$ ein DFA. Zeigen Sie

$$\hat{\delta}(q, uv) = \hat{\delta}(\hat{\delta}(q, u), v) \quad \text{für alle} \quad u, v \in \Sigma^*$$

auf zweierlei Weise:

(a) Direkt mittels Induktion über $|u|$.

(b) Mit Hilfe der relationalen Schreibweise aus Tutoraufgabe 3 dieses Übungsblatts.

Lösungsvorschlag

(a) Induktion über $|u|$:

Induktionsanfang: $u = \epsilon$. $\hat{\delta}(q, \epsilon v) = \hat{\delta}(q, v) = \hat{\delta}(\hat{\delta}(q, \epsilon), v)$

Induktionsschritt: $u = aw$ mit $a \in \Sigma$ und $w \in \Sigma^*$.

Wir haben die Induktionshypothese $\hat{\delta}(q, wv) = \hat{\delta}(\hat{\delta}(q, w), v) \quad \forall q \in Q$.

$\hat{\delta}(q, awv) = \hat{\delta}(\delta(q, a), wv) \stackrel{\text{(IH)}}{=} \hat{\delta}(\hat{\delta}(\delta(q, a), w), v) = \hat{\delta}(\hat{\delta}(q, aw), v)$

(b) Seien zunächst $|u| \geq 1$ und $|v| \geq 1$. Dann seien $u = a_1 \dots a_n$ und $v = b_1 \dots b_m$. Wir betrachten die Folgezustände $q_1, \dots, q_n$ so dass $q \xrightarrow{a_1} q_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} q_n$. Weiter erhalten wir ausgehend von q_n die Übergänge $q_n \xrightarrow{b_1} q'_1 \xrightarrow{b_2} \cdots \xrightarrow{b_m} q'_m$. Insgesamt ergibt sich daraus die Kette

$$q \xrightarrow{a_1} q_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} q_n \xrightarrow{b_1} q'_1 \xrightarrow{b_2} \cdots \xrightarrow{b_m} q'_m \; .$$

Aus diesen drei Ketten folgen nun mit Tutoraufgabe 3 die Gleichungen $\hat{\delta}(q, u) = q_n$, $\hat{\delta}(q_n, v) = q'_m$ und $\hat{\delta}(q, uv) = q'_m$.

Durch Einsetzen ergibt sich hieraus die geforderte Gleichung.

Ist $u = \varepsilon$, so besteht die erste Zustandssequenz nur aus $q = q_n$ und $\hat{\delta}(q, u) = q = q_n$. Ist $v = \varepsilon$, besteht die zweite Zustandssequenz nur aus $q_n = q'_m$ und $\hat{\delta}(q_n, v) = q_n = q'_m$. Das Einsetzen ist analog.

Hausaufgabe 3 (5 Punkte)

Endliche Automaten können auch in der Kodierung bzw. Dekodierung eingesetzt werden. Sei $\Sigma := \{\mathtt{0}, ..., \mathtt{9}\}$. Wir können eine Liste von Wörtern $w_1, ..., w_k$, mit $k \in \mathbb{N}$, über ein einziges Wort kodieren, indem wir vor jedes Wort w_i dessen Länge $n_i := |w_i|$ schreiben: $w := n_1 w_1\, n_2 w_2 ... n_k w_k$. Das funktioniert nur, wenn $w_1, ..., w_k$ höchstens Länge 9 haben. Ein so erzeugtes Wort w nennen wir *Präfix-Kodierung*. Zum Beispiel können wir $w_1 = \mathtt{512}, w_2 = \varepsilon, w_3 = \mathtt{2024}$ über $w = \mathtt{3512042024}$ kodieren. Weitere Präfix-Kodierungen sind etwa $\mathtt{123456}$, ε, oder $\mathtt{2020}$. Die Wörter $\mathtt{1234}$, $\mathtt{007}$ und $\mathtt{2024}$ sind dagegen keine Präfix-Kodierungen.
Wir betrachten nun die Sprache L aller Präfix-Kodierungen.

(a) Geben Sie einen deterministischen endlichen Automaten (DFA) an, der L akzeptiert. Erstellen Sie dafür sowohl eine graphische Darstellung als auch eine formale Beschreibung, in der Sie das Tupel $(Q, \Sigma, \delta, q_0, F)$ angeben. Ihr DFA darf höchstens 20 Zustände haben.

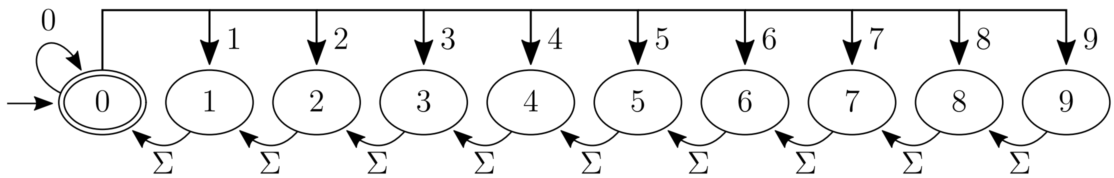
Hinweis: Beim Zeichnen von DFAs können Sie eine Kante mit Σ beschriften, statt jedes Zeichen des Alphabets einzeln aufzulisten.

(b) Geben Sie einen mengentheoretischen Ausdruck für L an. Sie dürfen nur endliche Mengen verwenden und diese mit Hilfe von endlichen Vereinigungen, endlichen Schnitten, sowie den Operationen auf Sprachen aus Definition 2.3 kombinieren.

Lösungsvorschlag

(a): $Q := \Sigma$, $q_0 := \mathtt{0}$, $F := \{\mathtt{0}\}$,

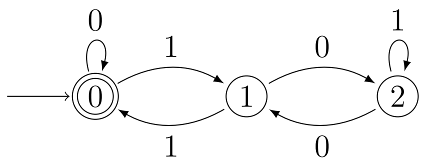
$$\delta(q, c) := \begin{cases} c & \text{falls } q = 0 \\ q - 1 & \text{sonst} \end{cases}$$



(b): $L = \left(\bigcup_{i \in \Sigma} \{i\}\Sigma^i\right)^*$

Hausaufgabe 4 (5 Punkte)

Gegeben sei der DFA $M = (\{0, 1, 2\}, \{0, 1\}, \delta, 0, \{0\})$ (Beispiel 3.5 aus der Vorlesung):



Zeigen Sie, dass der DFA genau die Wörter $w \in \{0, 1\}^*$ akzeptiert, für die $\#w$ ein Vielfaches von 3 ist. Hierbei definieren wir $\#\varepsilon = 0$ und erlauben führende Nullen. Die Intuition ist, dass sich der Automat genau dann im Zustand n befindet, wenn $\#w \bmod 3 = n$ für das bisher gelesene Wort w gilt.

(a) Zeigen Sie zuerst unter der Annahme, dass sich der Automat nach dem Einlesen eines Wortes w mit $\#w \bmod 3 = n$ im Zustand n befindet, die Aussage $\delta(n, b) = \#(wb) \bmod 3$ für ein beliebiges Zeichen $b \in \{0, 1\}$ gilt.

(b) Zeigen Sie dann $\hat{\delta}(0, w) = \#w \bmod 3$.

Lösungsvorschlag

(a) Für die binäre Repräsentation gilt $\#(wb) = 2\#w + b$. Damit können wir $\#(wb) \bmod 3$ wie folgt umformen:

$$\begin{aligned}
&\#(wb) \bmod 3 \\
&= (2\#w + b) \bmod 3 \\
&= (2(\#w \bmod 3) + b) \bmod 3 \\
&= (2n + b) \bmod 3
\end{aligned}$$

Damit genügt es zu zeigen, dass $\delta(n, b) = (2n + b) \bmod 3$ gilt. Einsetzen aller Kombinationen von n und b ergibt folgende Tabelle:

n	b	$\delta(n, b)$	$(2n + b) \bmod 3$
0	0	0	0
0	1	1	1
1	0	2	2
1	1	0	0
2	0	1	1
2	1	2	2

(b) Wir zeigen $\hat{\delta}(0, w) = \#w \bmod 3$ mittels Induktion über die Länge von w.

IBasis:

- $|w| = 0$: Dann ist $w = \varepsilon$ und damit $\#w = 0$. Wegen $\hat{\delta}(0, \varepsilon) = 0$ gilt die Behauptung.

ISchritt: Sei $w \in \{0, 1\}^*$ mit $|w| = n + 1$ beliebig aber fixiert.

- IAnnahme: Für alle w mit $|w| \leq n$ gilt $\hat{\delta}(0, w) = \#w \bmod 3$.
- IBehauptung: Für alle w mit $|w| = n + 1$ gilt $\hat{\delta}(0, w) = \#w \bmod 3$.
- IBeweis: Sei $w = w'b$ mit $b \in \{0, 1\}$. Dann gilt:

$$\begin{aligned}
&\hat{\delta}(0, w'b) \\
&= \delta(\hat{\delta}(0, w'), b) && \text{Vorlesung Folie 29} \\
&= \delta(\#w' \bmod 3, b) && \text{IA} \\
&= \#(w'b) \bmod 3 && \text{Teilaufgabe (a)}
\end{aligned}$$