Prof. Dr. Jasmin Blanchette
Jannis Limperg

Ludwig-Maximilians-Universität München
Institut für Informatik
Abgabe bis 17. Mai 2023, 10:00 Uhr

Lösungsvorschlag zur Übung 4 zur Vorlesung
Formale Sprachen und Komplexität

FSK4-1 ***Reguläre Ausdrücke*** (2 Punkte)

In dieser Aufgabe sind reguläre Ausdrücke entsprechend der Definition und Syntax im Vorlesungsskript, Definition 4.7.1, anzugeben.

a) Geben Sie einen regulären Ausdruck an, der genau die Sprache

$$L = \{w \mid i \in \Sigma, w \in \Sigma^* \text{ und } \#_i(w) = i\}$$

erzeugt, wobei $\Sigma = \{1, 2, 3\}$. (Siehe auch FSK3-1b.)

LÖSUNGSVORSCHLAG:

$$\big(\underbrace{((2\,|\,3)^*1(2\,|\,3)^*)}_{\text{genau eine 1}} \mid \underbrace{((1\,|\,3)^*2(1\,|\,3)^*2(1\,|\,3)^*)}_{\text{genau zwei 2en}} \mid \underbrace{((1\,|\,2)^*3(1\,|\,2)^*3(1\,|\,2)^*3(1\,|\,2)^*)}_{\text{genau drei 3en}}\big)$$

b) Für Autokennzeichen in München gibt es folgende Regeln:

- Sie beginnen mit einem M, danach folgt ein Buchstabenfeld, danach folgt ein Ziffernfeld. Anschließend können noch ein H für Oldtimer oder ein E für Elektroautos kommen, aber nicht beides gleichzeitig.
- Das Buchstabenfeld besteht aus 1 oder 2 Buchstaben aus $\{A, \ldots, Z\}$.
- Das Ziffernfeld besteht aus 3-4 Ziffern $\{0, \ldots, 9\}$, die erste Ziffer darf nicht 0 sein.
- Gehört das Fahrzeug allerdings einer Behörde an, so ist das Buchstabenfeld leer und der Ziffernblock 1-5 Ziffern lang. Dann darf der Ziffernblock ebenfalls nicht mit einer 0 anfangen und falls er 3 oder mehr Ziffern lang ist, auch nicht mit einer 4.
- Anmerkung: Es gibt weitere Sonderregelungen, welche in dieser Aufgabe ignoriert werden.

Geben Sie einen regulären Ausdruck für die gültigen Münchner Autokennzeichen an.

LÖSUNGSVORSCHLAG:

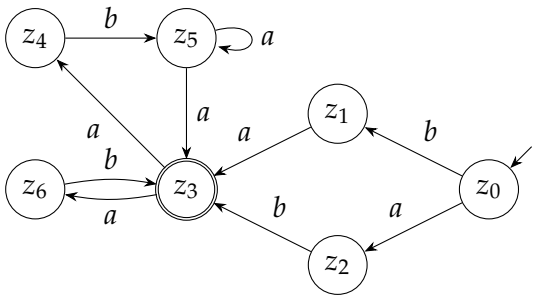
Teile des Ausdrucks:

- $X_1 := A \mid \cdots \mid Z$ sind die möglichen einbuchstabigen Autokennzeichen
- $X_2 := (A \mid \cdots \mid Z)(A \mid \cdots \mid Z)$ sind die möglichen zweibuchstabigen Autokennzeichen
- $X_3 := (1 \mid \cdots \mid 9)(0 \mid \cdots \mid 9)(0 \mid \cdots \mid 9)(0 \mid \cdots \mid 9 \mid \varepsilon)$ sind die 3-4-stelligen Zahlen
- $X_4 := (1 \mid 2 \mid 3 \mid 5 \mid \cdots \mid 9)(0 \mid \cdots \mid 9 \mid \varepsilon)(0 \mid \cdots \mid 9 \mid \varepsilon)(0 \mid \cdots \mid 9 \mid \varepsilon)(0 \mid \cdots \mid 9 \mid \varepsilon)$ sind die 1-5-stelligen Zahlen, die nicht mit einer 4 anfangen
- $X_5 := 4(0 \mid \cdots \mid 9 \mid \varepsilon)$ sind die 1-2-stelligen Zahlen, die mit einer 4 anfangen

$M((X_1 \mid X_2)X_3 \mid X_4 \mid X_5)(E \mid H \mid \varepsilon)$ ist damit der Gesamtausdruck.

Ausgeschrieben: $M(((A \mid \cdots \mid Z) \mid ((A \mid \cdots \mid Z)(A \mid \cdots \mid Z)))((1 \mid \cdots \mid 9)(0 \mid \cdots \mid 9)(0 \mid \cdots \mid 9)(0 \mid \cdots \mid 9 \mid \varepsilon)) \mid ((1 \mid 2 \mid 3 \mid 5 \mid \cdots \mid 9)(0 \mid \cdots \mid 9 \mid \varepsilon)(0 \mid \cdots \mid 9 \mid \varepsilon)(0 \mid \cdots \mid 9 \mid \varepsilon)(0 \mid \cdots \mid 9 \mid \varepsilon)) \mid (4(0 \mid \cdots \mid 9 \mid \varepsilon)))(E \mid H \mid \varepsilon)$

c) Betrachten Sie folgenden NFA A:



Geben Sie einen regulären Ausdruck an, der die von A erkannte Sprache $L(A)$ erzeugt.

Hinweis: Sie müssen nicht das Verfahren aus der Vorlesung befolgen. Ein Vorgehen ist es, die vom NFA erkannte Sprache zunächst in Wörtern zu beschreiben und sich im Anschluss daran einen regulären Ausdruck dafür zu überlegen.

LÖSUNGSVORSCHLAG:

Zunächst muss vom Startzustand z_0 der Zustand z_3 erreicht werden. Daher beginnt jedes akzeptierte Wort mit ab oder mit ba. Um von z_3 wieder in (den einzigen akzeptierenden Zustand) z_3 zu gelangen, gibt es zwei Pfade (Kreise): Entweder über z_6, dies ist mit ab möglich, oder über $z_4 \to z_5 \to z_3$, wobei man in z_5 noch beliebig lange durch Lesen von a verbleiben kann, d.h. dieser Weg entspricht den Teilwörten dargestellt durch aba^*a. Man kann über die beiden Kreise z_3 immer wieder besuchen, daher ergibt sich ingesamt der reguläre Ausdruck

$$(ab \mid ba)(ab \mid aba^*a)^*$$

FSK4-2 ***Reguläre Ausdrücke: Geldscheine wechseln*** (0 Punkte)

Ein Geldscheinwechselautomat nimmt 10- und 20-Euro-Scheine als Eingabe und wechselt diese in 50-, 100- und 200-Euro-Scheine. Er erhält als Eingabe ein Wort über $\Sigma = \{\texttt{10}, \texttt{20}\}$ und akzeptiert genau dann, wenn die Summe des Worts größer 0 und durch 50 teilbar ist. Zum Beispiel akzeptiert er `10 10 20 10 20 20 10` (da die Summe 100 ist), aber er akzeptiert `10 10 20 20` nicht (da die Summe 60 ist).

Geben Sie einen regulären Ausdruck an, der genau die durch den Geldscheinwechselautomaten akzeptierte Sprache erzeugt.

Hinweis: Es kann sinnvoll sein, zuerst einen endlichen Automaten zu konstruieren und dann den regulären Ausdruck zu entwerfen.

LÖSUNGSVORSCHLAG:

Folgen, die nicht mit einem durch 50 teilbaren Betrag enden, können entweder noch passend aufgefüllt werden oder enden mit 10 zu viel (da der letzte Schein ein Zwanziger war). Im zweiten Fall muss eine 40er-Folge (zusätzlich zu beliebig vielen 50er-Folgen) kommen.

Mit dieser Vorüberlegung ergibt sich als Idee:

$$(\mathit{Fünzig} \mid \mathit{Sechzig}\,\mathit{Fünzig}^*\,\mathit{Vierzig})(\mathit{Fünzig} \mid \mathit{Sechzig}\,\mathit{Fünzig}^*\,\mathit{Vierzig})^*$$

wobei $\mathit{Vierzig}, \mathit{Fünzig}, \mathit{Sechzig}$ Abkürzungen für reguläre Ausdrücke sind, die Folgen mit den entsprechenden Summen erzeugen.

Dabei genügt es, wenn *Sechzig* nur solche Folgen erzeugt, die als Präfix nicht die Summe 50 erzeugen können. Daher sei

$$\mathit{Sechzig} := \mathit{Vierzig}\,\texttt{20}$$

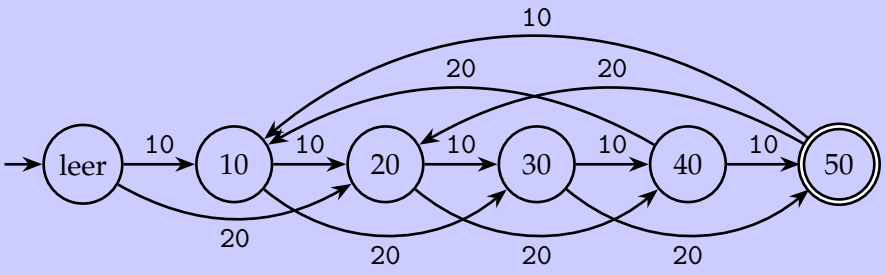
Für *Vierzig* und *Fünzig* müssen alle Varianten aufgezählt werden:

$$
\begin{array}{lcl}
\mathit{Vierzig} & := & (10\,10\,10\,10 \mid 10\,10\,20 \mid 10\,20\,10 \mid 20\,10\,10 \mid 20\,20) \\
\mathit{Fünzig} & := & (\mathit{Vierzig}\,10 \mid 10\,\mathit{Vierzig} \mid 20\,10\,20)
\end{array}
$$

Insgesamt ergibt dies:

$$
\begin{array}{l}
(((\ 10\,10\,10\,10 \mid 10\,10\,20 \mid 10\,20\,10 \mid 20\,10\,10 \mid 20\,20)\,10 \\
\quad \mid 10\,(10\,10\,10\,10 \mid 10\,10\,20 \mid 10\,20\,10 \mid 20\,10\,10 \mid 20\,20) \\
\quad \mid 20\,10\,20 \\
\ ) \\
\ \mid (10\,10\,10\,10 \mid 10\,10\,20 \mid 10\,20\,10 \mid 20\,10\,10 \mid 20\,20)\,20 \\
\quad (\ (10\,10\,10\,10 \mid 10\,10\,20 \mid 10\,20\,10 \mid 20\,10\,10 \mid 20\,20)\,10 \\
\quad \mid 10\,(10\,10\,10\,10 \mid 10\,10\,20 \mid 10\,20\,10 \mid 20\,10\,10 \mid 20\,20) \\
\quad \mid 20\,10\,20 \\
\quad)^* \\
\quad (10\,10\,10\,10 \mid 10\,10\,20 \mid 10\,20\,10 \mid 20\,10\,10 \mid 20\,20 \\
\quad) \\
) \\
(((\ 10\,10\,10\,10 \mid 10\,10\,20 \mid 10\,20\,10 \mid 20\,10\,10 \mid 20\,20)\,10 \\
\quad \mid 10\,(10\,10\,10\,10 \mid 10\,10\,20 \mid 10\,20\,10 \mid 20\,10\,10 \mid 20\,20) \\
\quad \mid 20\,10\,20 \\
\ ) \\
\ \mid (10\,10\,10\,10 \mid 10\,10\,20 \mid 10\,20\,10 \mid 20\,10\,10 \mid 20\,20)\,20 \\
\quad (\ (10\,10\,10\,10 \mid 10\,10\,20 \mid 10\,20\,10 \mid 20\,10\,10 \mid 20\,20)\,10 \\
\quad \mid 10\,(10\,10\,10\,10 \mid 10\,10\,20 \mid 10\,20\,10 \mid 20\,10\,10 \mid 20\,20) \\
\quad \mid 20\,10\,20 \\
\quad)^* \\
\quad (10\,10\,10\,10 \mid 10\,10\,20 \mid 10\,20\,10 \mid 20\,10\,10 \mid 20\,20 \\
\quad) \\
)^*
\end{array}
$$

Zur Anschauung ein DFA für die Sprache:



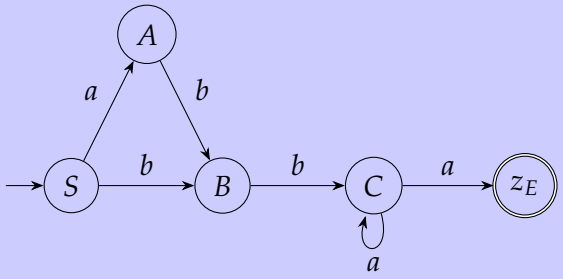
FSK4-3 ***Grammatik über Automaten zu Grammatik*** (2 Punkte)

Gegeben sei die Grammatik

$$G = (\{S, A, B, C\}, \{a, b\}, \{S \to aA \mid bB,\ A \to bB,\ B \to bC,\ C \to aC \mid a\}, S)$$

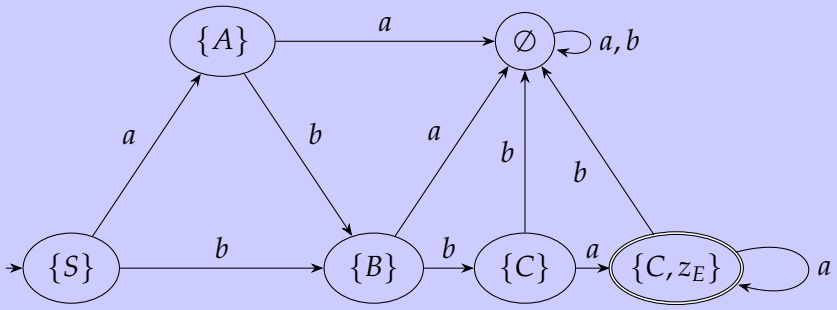
a) Erzeugen Sie gemäß der Konstruktion aus der Vorlesung aus G einen NFA A mit $L(G) = L(A)$. Zeichnen Sie den Zustandsgraph von A.

LÖSUNGSVORSCHLAG:



b) Erzeugen Sie mit der Potenzmengenkonstruktion aus A einen DFA B mit $L(B) = L(A)$. Zeichnen Sie den vom Startzustand erreichbaren Teil des Zustandsgraphen von B.

LÖSUNGSVORSCHLAG:



c) Erzeugen Sie gemäß der Konstruktion aus der Vorlesung aus B eine Grammatik H mit $L(B) = L(H)$.

LÖSUNGSVORSCHLAG:

Neubenennung der Zustände (um sie klarer erkenntlich als Variablennamen schreiben zu können):

$V_S := \{S\}, V_A := \{A\}, V_B := \{B\}, V_C := \{C\}, V_{CE} := \{C, z_E\}, V_\emptyset = \emptyset$
$H = (\{V_S, V_A, V_B, V_C, V_{CE}, V_\emptyset\}, \{a, b\}, \{$
$V_S \to aV_A \mid bV_B,$
$V_A \to aV_\emptyset \mid bV_B,$
$V_B \to aV_\emptyset \mid bV_C,$
$V_C \to aV_{CE} \mid a \mid bV_\emptyset,$
$V_{CE} \to aV_{CE} \mid a \mid bV_\emptyset,$
$V_\emptyset \to aV_\emptyset \mid bV_\emptyset,$
$\}, V_S)$

d) Vergleichen Sie die Grammatiken G und H. Beschreiben Sie die Gemeinsamkeiten dieser Grammatiken, sowie ihre Unterschiede.

Überlegen Sie sich, wodurch diese Effekte zustande kommen.

LÖSUNGSVORSCHLAG:

Grund für diese Effekte ist, dass G direkt einem nichtdeterministischen Automaten entspricht und H direkt einem deterministischen Automaten. Darum übertragen sich die Eigenschaften der entsprechenden Automaten und Automatenmodelle auf die Grammatiken.

Gemeinsamkeiten:

- Sie akzeptieren die gleiche Sprache (aufgrund der Konstruktion).
- Es gibt sowohl in G als auch in H Variablen, die ein Terminalsymbol und sich selbst erzeugen.

Unterschiede:

- H enthält mehr Variablen und deutlich mehr Regeln als G (da die Automaten unterschiedlich groß sind).
- H enthält die Variable $V_\emptyset$, über die beliebig („unendlich“) lange abgeleitet werden kann, ohne dass je ein Wort produziert würde. Dadurch kann man für jedes Teilwort $w \in \Sigma^*$ eine Satzform wV_x (wobei V_x eine Variable ist) ableiten. Wendet man die Regeln von H „blind“ an, kommt man also leicht in eine Sackgasse. G hat dieses Problem nicht.

FSK4-4 ***DNA-Analyse mit NFA*** (0 Punkte)

Diese Aufgabe handelt von der Analyse von Desoxyribonukleinsäure (DNS/DNA) mithilfe von NFA. DNA ist eine Abfolge der Basen Adenin, Thymin, Guanin und Cytosin, typischerweise mit A, T, G und C abgekürzt. Dementsprechend ist das Alphabet aller Automaten in dieser Aufgabe $\Sigma = \{A, C, G, T\}$.

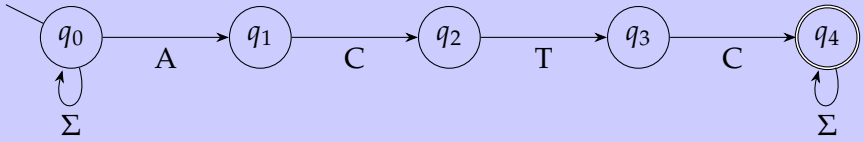
a) Um das Vorkommen einer Basensequenz zu finden, wird aus dieser Sequenz ein NFA erzeugt, der alle Wörter akzeptiert, in denen diese Sequenz als Teilwort vorkommt.

Zeichnen Sie den Zustandsgraph eines NFA B, der genau diejenigen Wörter akzeptiert, in denen $ACTC$ als Teilwort vorkommt.

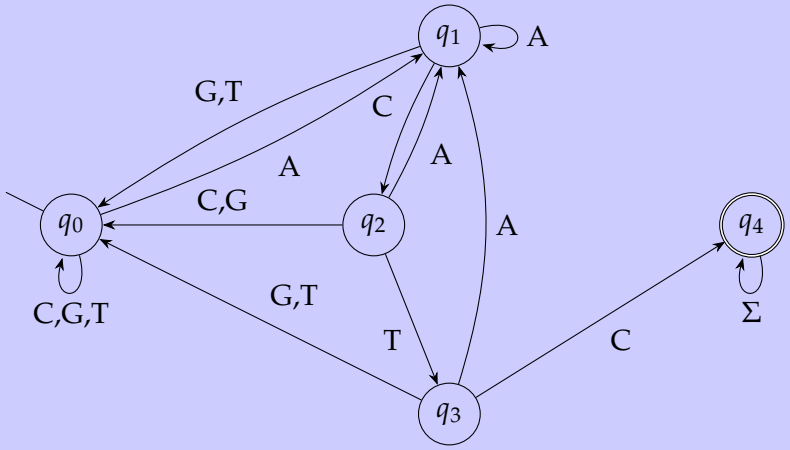
Hinweis: Sie können (müsssen aber nicht) dazu den regulären Ausdruck $(A|C|G|T)^*ACTC(A|C|G|T)^*$ verwenden, der genau diese Sprache akzeptiert.

Hinweis: Sie können auch einen DFA angeben, aber ein NFA ist übersichtlicher.

LÖSUNGSVORSCHLAG:



Ein dazu äquivalenter DFA ist



b) Beim Kopieren von DNA kann es vorkommen, dass Fehler auftreten. Zum Beispiel kann eine Base durch eine andere ersetzt werden; es kann eine Base ausgelassen werden; es kann eine zusätzliche Base eingefügt werden; und es können auch komplexere Fehler auftreten. Zur Vereinfachung behandeln wir hier nur den Fall, dass eine Base durch eine andere ersetzt wird.

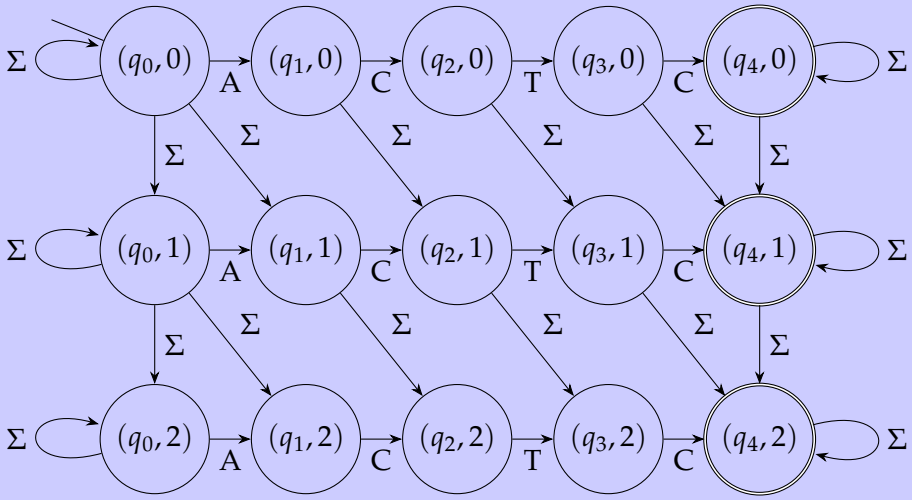
Aus einem NFA $D = (Z, \Sigma, \delta, S, E)$ kann ein NFA $F = (Z', \Sigma, \delta', S', E')$ erzeugt werden, der alle Wörter akzeptiert, die durch höchstens k fehlerhafte Ersetzungen aus D entsteht.

Dabei sind

- $Z' = Z \times \{0, \ldots, k\}$
- $\delta'((q,i),a) = \{(q',i) \mid q' \in \delta(q,a)\}$
  $\cup \{(q',i+1) \mid (\exists b \in \Sigma.\ q' \in \delta(q,b) \wedge i+1 \leq k)\}$
- $S' = S \times \{0\} = \{(s,0) \mid s \in S\}$
- $E' = E \times \{0, \ldots, k\}$

Berechnen Sie mit der obigen Konstruktion einen NFA H aus B, der Wörter mit bis zu 2 Fehlern akzeptiert. Zeichnen Sie den Zustandsgraphen von H.

LÖSUNGSVORSCHLAG:



c) Geben Sie an und begründen Sie, welche der folgenden Wörter von H akzeptiert werden. Prüfen Sie, ob Ihr Ergebnis korrekt ist, also ob die erkannten Wörter tatsächlich diejenigen sind, bei denen bis auf höchstens 2 Fehler das Wort $ACTC$ als Teilwort vorkommt.

$AAAACCCAAA$, $GAGGCGT$, $TAGCA$, $TCTCA$

LÖSUNGSVORSCHLAG:

- $AAAACCCAAA \in L(H)$ mit akzeptierendem Lauf $(q_0,0) \xrightarrow{A} (q_0,0) \xrightarrow{A} (q_0,0) \xrightarrow{A} (q_0,0) \xrightarrow{A} (q_1,0) \xrightarrow{C} (q_2,0) \xrightarrow{C} (q_3,1) \xrightarrow{C} (q_4,1) \xrightarrow{A} (q_4,1) \xrightarrow{A} (q_4,1) \xrightarrow{A} (q_4,1)$.

Ist korrekt: $ACTC$ kommt mit 1 Fehler im unterstrichenen Teil vor: $AAA\underline{ACCC}AAA$

- $GAGGCGT \in L(H)$ mit akzeptierendem Lauf $(q_0,0) \xrightarrow{G} (q_0,0) \xrightarrow{A} (q_1,0) \xrightarrow{G} (q_2,1) \xrightarrow{G} (q_3,2) \xrightarrow{C} (q_4,2) \xrightarrow{G} (q_4,2) \xrightarrow{T} (q_4,2)$.
  Ist korrekt: $ACTC$ kommt mit 2 Fehlern im unterstrichenen Teil vor: $G\underline{AGGC}GT$
- $TAGCA \notin L(H)$. Das ist zu erkennen, indem für jede Wortposition die Menge der erreichbaren Zustände aufgeführt wird. (Dies entspricht auch dem Lauf auf dem zugehörigen DFA aus der Potenzmengenkonstruktion, der muss dafür aber nicht vollständig konstruiert werden.)
  $\{(q_0,0)\} \xrightarrow{T} \{(q_0,0),(q_0,1),(q_1,1)\} \xrightarrow{A}$
  $\{(q_0,0),(q_0,1),(q_0,2),(q_1,0),(q_1,1),(q_1,2),(q_2,2)\} \xrightarrow{G}$
  $\{(q_0,0),(q_0,1),(q_0,2),(q_1,1),(q_1,2),(q_2,1),(q_2,2)\} \xrightarrow{C}$
  $\{(q_0,0),(q_0,1),(q_0,2),(q_1,1),(q_1,2),(q_2,1),(q_2,2)\} \xrightarrow{A}$
  $\{(q_0,0),(q_0,1),(q_0,2),(q_1,0),(q_1,1),(q_1,2),(q_2,2)\}$
  Diese Mengen beinhalten je die mit diesem Wortanfang erreichbaren Zustände. Von einer Menge kommt man zur nächsten, indem man die Vereinigung aller $\delta(q,a)$ bildet. Beispiele aus dieser Berechnung:
  - $\{(q_0,0)\} \xrightarrow{T} \{(q_0,0),(q_0,1),(q_1,1)\}$:
    $\delta_H((q_0,0),T) = \{(q_0,0),(q_0,1),(q_1,1)\}$
  - $\{(q_0,0),(q_0,1),(q_1,1)\} \xrightarrow{A}$
    $\{(q_0,0),(q_0,1),(q_0,2),(q_1,0),(q_1,1),(q_1,2),(q_2,2)\}$:

    $$\begin{aligned} & \delta_H((q_0,0),A) \cup \delta_H((q_0,1),A) \cup \delta_H((q_1,1),A) \\ = \; & \{(q_0,0),(q_0,1),(q_1,0),(q_1,1)\} \cup \{(q_0,1),(q_0,2),(q_1,1),(q_1,2)\} \cup \\ & \{(q_2,2)\} \\ = \; & \{(q_0,0),(q_0,1),(q_0,2),(q_1,0),(q_1,1),(q_1,2),(q_2,2)\} \end{aligned}$$

  Es ist aber $\{(q_0,0),(q_0,1),(q_0,2),(q_1,0),(q_1,1),(q_1,2),(q_2,2)\} \cap E' = \varnothing$, also $TAGCA \notin L(H)$.
  Dieses Ergebnis ist auch korrekt, denn alle Teilwörter der Länge 4 haben mehr als 2 Zeichen anders als $ACTC$: $TAGC$ (Unterschied: 3) und $AGCA$ (Unterschied 3).
- $TCTCA \in L(H)$ mit akzeptierendem Lauf $(q_0,0) \xrightarrow{T} (q_1,1) \xrightarrow{C} (q_2,1) \xrightarrow{T} (q_3,1) \xrightarrow{C} (q_4,1) \xrightarrow{A} (q_4,1)$.
  Ist korrekt: $ACTC$ kommt mit 1 Fehler im unterstrichenen Teil vor: $\underline{TCTC}A$

d) Begründen Sie, dass die Konstruktion aus b) korrekt ist, also tatsächlich für jeden NFA D und jedes k einen NFA F liefert, der maximal k Fehler zulässt.

Hinweis: Sie können auch als Vorüberlegung dies erst für den NFA H zeigen.

LÖSUNGSVORSCHLAG:

Die Konstruktion ist korrekt, wenn folgende beiden Eigenschaften gelten, die wir getrennt zeigen:

- Wenn $w \in L(D)$ ist und u sich von w an maximal k Stellen unterscheidet, dann ist $u \in L(F)$.
- Wenn $w \in L(F)$ ist, dann gibt es ein $u \in L(D)$ sodass sich u von w an maximal k Stellen unterscheidet.

Gelten diese beiden Eigenschaften, dann akzeptiert F genau die Fehlersprache. Jetzt zeigen wir diese Eigenschaften:

- Da $w \in L(D)$, gibt es einen Lauf ϱ von D auf w.
  Wir definieren eine Folge, die angibt, wie viele Fehler wir bis zu einem bestimmten Punkt im Wort gesehen haben:

$$f_i = \begin{cases} 0 & \text{wenn } i = 0 \\ f_{i-1} & \text{wenn } u[i] = w[i] \\ f_{i-1} + 1 & \text{wenn } u[i] \neq w[i] \end{cases}$$

  Es ist $f_{|w|} \leq k$, da sich u und w an maximal k Stellen unterscheiden.
  Nun ist $(\varrho_0, f_0)(\varrho_1, f_1) \dots (\varrho_{|w|}, f_{|w|})$ ein akzeptierender Lauf von F auf u, da die Übergänge alle möglich sind. An jeder Stelle passt wahlweise der Übergang, da ϱ ein Lauf von D auf w ist und sich die Fehlerzahl f nicht ändert, oder die Fehlerzahl erhöht sich um 1 und der Buchstabe an dieser Position ist irrelevant. Zudem ist $(\varrho_{|w|}, f_{|w|})$ ein Endzustand in F.
- Da $w \in L(F)$, gibt es einen Lauf $(\varrho_0, f_0)(\varrho_1, f_1) \dots (\varrho_{|w|}, f_{|w|})$ von F auf w.
  Wähle nun ein Wort u mit $|u| = |w|$ und für alle $0 < i \leq |w|$:

$$u[i] = \begin{cases} w[i] & \text{wenn } f_{i-1} = f_i \\ \text{beliebig aus } \{a \mid \varrho_i \in \delta_D(\varrho_{i-1}, a)\} & \text{sonst} \end{cases}$$

  Dabei kann die Menge $\{a \mid \varrho_i \in \delta_D(\varrho_{i-1}, a)\}$ nicht leer sein, da es sonst in F keinen Übergang von (ϱ_{i-1}, f_{i-1}) nach (ϱ_i, f_i) gäbe.
  Da $(\varrho_{|w|}, f_{|w|}) \in E_F$ und damit $\varrho_{|w|} \in E_D$, ist $\varrho_0 \dots \varrho_{|w|}$ ein akzeptierender Lauf von D auf u, also $u \in L(D)$. Zudem unterscheidet sich u von w an maximal k Stellen.