Prof. J. Esparza
S. Sickert, J. Krämer

Einführung in die theoretische Informatik

Sommersemester 2017 – Übungsblatt Lösungsskizze 6

Übungsblatt

Wir unterscheiden zwischen Übungs- und Abgabeblättern. Auf diesem *Übungsblatt* finden Sie eine Übersicht über die Kernaspekte, die Sie in Kalenderwoche 23 in den Tutorien diskutieren, üben und vertiefen. Die Aufgaben auf diesem Blatt dienen dem Üben und Verstehen des Vorlesungsstoffes, sowie dem *eigenständigen Erarbeiten* der Kernaspekte. Außerdem sollen Ihnen diese Aufgaben auch helfen, ein Gefühl dafür zu bekommen, was Sie inhaltlich in der Klausur erwartet. Klausuraufgaben können jedoch deutlich von den hier gestellten Aufgaben abweichen. Abschreiben und Auswendiglernen von Lösungen wird Ihnen daher keinen dauerhaften Erfolg in der Vorlesung bringen. Fragen zu den Übungsblättern können Sie montags bis donnerstags von 12 Uhr bis 14 Uhr in der *THEO-Sprechstunde* in Raum 03.11.034 stellen.

Kernaspekte

K6.1 korrektes Wiedergeben der folgenden Definitionen

- Syntaxbaum
- Rechts- und Linksableitung

K6.2 zu einer natürlich sprachlich gegebenen Sprache eine formale Definition angeben
K6.3 zu einer gegebenen Sprache L eine Grammatik G angeben, so dass $\mathsf{L} = \mathsf{L}(\mathsf{G})$ und die Korrektheit der Grammatik beweisen
K6.4 für eine gegebene Sprache entscheiden, ob diese regulär, kontextfrei oder kontextsensitiv ist
K6.5 für ein Wort w, das von einer Grammatik G erzeugt wird, die Links-/Rechtsableitung angeben

Die Abgabeblätter sind die beste Gelegenheit, Feedback zu Ihrem aktuellen Wissens- und Fähigkeitenstand zu bekommen. Gleichzeitig sehen wir auch, wo es noch Verständnisschwierigkeiten gibt. *Die folgenden Aufgaben sollen Ihnen dabei helfen, Fehler, die wir während der Korrektur häufig gesehen haben, zu vermeiden – auch im Hinblick auf die Klausur!* Ein Teil der folgenden Aufgaben umfasst inhaltliche Probleme. Darüber hinaus zählt in dieser Vorlesung aber nicht nur, dass Sie intuitiv verstanden haben, wie etwas funktioniert, sondern vor allem dass Sie etwas in angemessen *formaler Art und Weise präzise aufschreiben, begründen und beweisen* können.

AUFGABE 1. (*Warum?*)
Sie fragen sich vielleicht, *warum* es uns so wichtig ist, dass Sie lernen, in formaler Art und Weise präzise und unmissverständlich zu formulieren. Stellen Sie sich vor, Sie sind Tutor*in für die theoretische Informatik und müssen Ihre Übungsgruppe davon überzeugen, dass eine formale und präzise Sprache wichtig ist.
Sammeln Sie auf einer halben Seite stichpunktartig die Argumente, mit denen Sie Ihre Übungsgruppe überzeugen möchten

Hinweise:

- Suchen Sie nach realen, informatischen Beispielen, in denen das Festhalten und Abstimmen von Definitionen bis ins kleinste Detail kostenspielige oder lebensgefährliche Situationen vermieden hätte. *Beispiele:*
  - `https://de.wikipedia.org/wiki/Mars_Climate_Orbiter`
  - viele "bekannte" Software-Bugs
- Schauen Sie sich die Klausuranforderungen der letzten Jahre an.

Die folgenden Fragestellungen sind alle im Anforderungsniveau B anzusiedeln und sollen Ihnen dabei helfen, unser Feedback zu Ihrer Abgabe so zu verwerten, dass Sie bei ähnlichen Aufgaben in Zukunft mehr Punkte erreichen. Wenn die Beantwortung der Fragen Ihnen Probleme bereitet, empfehlen wir Ihnen *nachdrücklich* die Sprechstunde aufzusuchen oder Ihren Tutoren auf die offenen Fragen anzusprechen.

AUFGABE 2. (*Formale Notation*)
Schauen Sie sich Ihre korrigierte Abgabe genau an.

(a) Sammeln Sie alle Stellen, an denen formale Fehler zu Punktabzug geführt haben. Formale Fehler erkennen Sie vor allem an folgenden Kommentaren:

- zu ungenau
- Begründung fehlt
- kein formaler Beweis
- Idee ok, aber...
- Wieso gilt das?
- vom Pumping-Lemma-Schema abgewichen

(b) Korrigieren Sie die formalen Fehler, indem Sie auf die Kommentare eingehen.[1]

AUFGABE 3. (*Pumping-Lemma*)
Beim Pumping-Lemma mussten wir bei vielen Abgaben Punkte abziehen, da die formal notwendigen Schritte des Pumping-Lemmas nicht eingehalten wurden. Da das Pumping-Lemma aber eine "fragile" Aussage ist, erhält man keinen wasserdichten Beweis, wenn man sich nicht an die formalen Vorgaben hält. Wir möchten Sie in dieser Aufgabe für die formal notwendigen Schritte sensibilisieren.

(a) Schauen Sie sich die Übungsaufgaben zum Pumping-Lemma von Übungsblatt 4 nochmals an. Markieren Sie in den Lösungen alle Aussagen, die die fünf Beweise wortwörtlich miteinander gemein haben.
(b) Erklären Sie in eigenen Worten, was eine Pumping-Lemma-Zahl für eine Sprache L ist. Diskutieren Sie, warum es notwendig ist, dass n eine Pumping-Lemma-Zahl ist (*und es NICHT* geht $n \in \mathbb{N}$ beliebig zu setzen).[2]
(c) Listen Sie auf, was Sie beim Pumping-Lemma wählen dürfen und was nicht. Erklären Sie Ihre Auswahl in eigenen Worten.
(d) Erklären Sie, warum in Aufgabe 1.4 des Abgabeblattes $z = \mathsf{a}^n \mathsf{b}^{\frac{n}{2}}$ eine ungültige/falsche Wahl für z ist.

AUFGABE 4. (*Sprachen darstellen*)
Wenn Sie *beweisen* sollen, dass eine Sprache regulär ist, dann sollen Sie im Allgemeinen eine rechtslineare Grammatik, einen regulären Ausdruck oder einen DFA, einen NFA oder einen ε-NFA angeben, der genau die Sprache erzeugt/beschreibt akzeptiert. Ihr Beweis ist aber *nur dann vollständig*, wenn Sie auch beweisen, dass Ihre Darstellung korrekt ist.

(a) Betrachten Sie Aufgabe 1.3 des Abgabeblattes. Geben Sie an, was Sie beweisen müssen, um zu zeigen, dass die Darstellung der Sprache als DFA korrekt ist.
(b) Stellen Sie sich vor, Sie hätten in Aufgabe 1.3 keinen DFA angeben müssen, sondern einen...
    (i) NFA
    (ii) ε-NFA
    (iii) reguläre Ausdruck oder
    (iv) eine rechtslineare Grammatik.

    Geben Sie erneut an, was Sie beweisen müssen, um zu zeigen, dass die Darstellung der Sprache korrekt ist.

AUFGABE 5. (*Induktive Beweise*)
Sei $\mathsf{N} = (\mathsf{Q}, \Sigma, \delta, \mathsf{q}_0, \mathsf{F})$ ein endlicher Automat. Um zu begründen bzw. zu beweisen, dass ein Wort $w \in \Sigma^*$ von einem Automaten akzeptiert wird, müssen Sie eine Aussage über $\hat{\delta}(\mathsf{q}_0, w)$ treffen.

(a) Geben Sie an, was Sie in Aufgabe 1.3 hätten beweisen müssen, um zu zeigen, dass die Sprache L_X und die Sprache $\mathsf{L}(\mathsf{D})$ äquivalent sind. Beschreiben Sie außerdem in eigenen Worten, wie und mit welchem Beweisverfahren man diese Aussage beweisen kann.
(b) Beschreiben Sie, wie Sie vorgehen können, wenn Sie ähnliche Aussagen für reguläre Ausdrücke, rechtslineare Grammatiken oder nicht-deterministische endliche Automaten beweisen sollen.
(c) Listen Sie alle Beweise der Vorlesung auf, bei denen wir mithilfe von Induktion Gleichheiten verschiedener Darstellungen von Sprachen beweisen. Identifizieren Sie Gemeinsamkeiten und Unterschiede und geben Sie weitere ähnliche Aussagen an, bei denen Sie Induktion als Beweistechnik einsetzen würden.

AUFGABE 6. (*Falsche Schlüsse*)
Wir haben (fehlerhafte) Begründungen der folgenden Art auf den Abgaben gesehen. Ziel dieser Aufgabe ist es, zu zeigen, dass diese Schlüsse *falsch* sind. Geben Sie dazu für jede der folgenden Aussagen ein Gegenbeispiel an:

(a) Die *unendliche* Vereinigung regulärer Mengen ist regulär.
(b) Die Vereinigung einer regulären und einer nicht regulären Sprache ist nicht regulär.

[1] Schreiben Sie *wirklich* auf, wie Sie die Korrektur vornehmen. Dann können wir Sie leichter bei Problemen beraten!
[2] Für die Logiker: Wenn wir annehmen, dass eine Sprache regulär ist und wir damit das Pumping-Lemma anwenden können, nehmen wir ebenfalls an, dass es eine natürliche Zahl *gibt*, die die Eigenschaften der Pumping-Lemma-Zahl erfüllt. Das n ergibt sich damit als Instantiierung eines Existenz-Quantors (und nicht als Auflösung eines Allquantors, wie man es von Beweisen von Allaussagen kennt).

AUFGABE 6.1. Stufe B + D

Mit $\mathsf{G} = (\{\mathsf{S}, E, O, A, B, X\}, \{\mathsf{a}, \mathsf{b}\}, \mathsf{P}, \mathsf{S})$ sei die CFG mit folgenden Produktionen P bezeichnet:

$$\begin{array}{lcl} \mathsf{S} & \to & E \mid O \\ E & \to & AB \mid BA \\ A & \to & XAX \mid \mathsf{a} \\ B & \to & XBX \mid \mathsf{b} \\ O & \to & XXO \mid X \\ X & \to & \mathsf{a} \mid \mathsf{b} \end{array}$$

(a) Geben Sie für jedes der folgenden Wörter jeweils eine Linksableitung und eine Rechtsableitung zzgl. des entsprechenden Syntaxbaums an:

(i) abaaaa (ii) babab (iii) aabbaaba

Diskutieren Sie Gemeinsamkeiten und Unterschiede der Link- und Rechtssableitungen.

(b) Bestimmen Sie die von $\mathsf{L}(\mathsf{G})$ erzeugte Sprache und beweisen Sie, dass die Grammatik tatsächlich die angegebene Sprache erzeugt.

Lösungsskizze

(a) Linksableitung für (i): $\mathsf{S} \to E \to BA \to X\mathsf{b}XA \to \mathsf{ab}XA \to \mathsf{aba}A \to \mathsf{aba}X\mathsf{a}X \to \mathsf{abaaa}X \to \mathsf{abaaaa}$
Rechtsableitung für (i): $\mathsf{S} \to E \to BA \to BX\mathsf{a}X \to BX\mathsf{aa} \to B\mathsf{aaa} \to X\mathsf{b}X\mathsf{aaa} \to X\mathsf{baaaa} \to \mathsf{abaaaa}$

(b) Für jedes Nicht-Terminal Z bezeichnen wir mit $\mathsf{L}_Z(\mathsf{G})$ die von G erzeugte Sprache, wenn man Z als Startsymbol (Axiom) wählt.
Sei nun $\mathsf{L}' = \bigcup_{k \geq 0} \{u\mathsf{a}v \mid u, v \in \Sigma^k\}$. Wir zeigen $\mathsf{L}_A(\mathsf{G}) = \mathsf{L}'$.

(i) $\mathsf{L}_A(\mathsf{G}) \subseteq \mathsf{L}'$.
Strukturelle/wohlfundierte Induktion nach den Ableitungsregeln $A \to XAX$, $A \to a$. Sei $w \in \mathsf{L}_A(\mathsf{G})$. Dann gibt es ein k, so dass w mit $3k+1$ Ableitungsschritten abgeleitet worden ist. Dann $w \in \{uav \mid u, v \in \Sigma^k\}$.
$k = 0$: Dann wird w durch genau eine Produktionsanwendung erzeugt, d.h. w wird durch $A \to a$ erzeugt. Somit $w = a \in \{uav \mid u, v \in \Sigma^0\}$.
$k \to k+1$: Es gilt $A \to XAX \to_{\mathsf{G}}^{*} w = x_1 w' x_2$ mit $x_1, x_2 \in \{a, b\}$. Da gilt $A \to_{\mathsf{G}}^{3k+1} w'$ und nach Induktion gilt $w' \in \{uav \mid u, v \in \Sigma^k\}$, so dass $w \in \{uav \mid u, v \in \Sigma^{k+1}\}$ folgt.

(ii) $\mathsf{L}_A(\mathsf{G}) \supseteq \mathsf{L}'$
Mittels Induktion nach k zeigen wir, dass jedes $w \in \{uav \mid u, v \in \Sigma^k\}$ eine Ableitung mit A als Startsymbol besitzt.
$k = 0$: Dann $w = a$ mit $A \to a = w$.
$k \to k+1$: Dann $w = xw'y$ mit $x, y \in \{a, b\}$ und $w' \in \{uav \mid u, v \in \Sigma^k\}$. Nach Induktion besitzt w' eine Ableitung $A \to^* w'$, somit $A \to XAX \to Xw'X \to xw'X \to xw'y = w$.

Entsprechend folgt $\mathsf{L}_B(\mathsf{G}) = \bigcup_{k \geq 0} \{ubv \mid u, v \in \Sigma^k\}$ und $\mathsf{L}_O(\mathsf{G}) = \Sigma(\Sigma\Sigma)^*$, d.h. O produziert alle Wörter ungerader Länge.
Für die restlichen Nichtterminale kann man nun direkt die erzeugten Sprachen angeben, da diese nur von Nichtterminalen abhängen, für welche die Sprachen schon bekannt sind:

$$\mathsf{L}_E(\mathsf{G}) = \mathsf{L}_A(\mathsf{G})\mathsf{L}_B(\mathsf{G}) \cup \mathsf{L}_B(x)\mathsf{L}_A(\mathsf{G}) = \bigcup_{i,j \geq 0} \mathsf{L}((a|b)^i a (a|b)^{i+j} b (a|b)^j) \cup \bigcup_{i,j \geq 0} \mathsf{L}((a|b)^i b (a|b)^{i+j} a (a|b)^j)$$

also die Sprache aller Wörter der Länge $2(i+j+1)$, die sich zumindest an den Positionen $i+1$ und $i+1+(i+j+1)$ unterscheiden. Da sich jedes Wort in $w \in \mathsf{L}$ in einen der Fälle $|w|$ ungerade oder $|w| = 2n$ mit $w_k \neq w_{n+k}$ unterteilen lässt, folgt insgesamt, dass G die Sprache $\Sigma^* \setminus \{ww \mid w \in \Sigma^*\}$ erzeugt.

AUFGABE 6.2. (*Lernen von regulären Sprachen*) Stufe C

Bilden Sie Gruppen zu jeweils vier Studierenden. Zwei Studierende nehmen die Rolle des **Teachers** ein und zwei die Rolle des **Learners** ein. Die Aufgabe des **Teachers** ist es, sich eine reguläre Sprache L auszudenken und Fragen zu dieser zu beantworten. Die **Learner** müssen diese Sprache lernen, d.h. einen Automaten für diese Sprache konstruieren. Hierbei dürfen die Learner aber nur die folgenden zwei Fragen stellen:

(a) **Membership:** Ist das Wort w in der Sprache L?

(b) **Equivalence:** Erkennt der gegebene Automat (DFA, NFA, ε-NFA) A die Sprache L?

Die Teacher dürfen nur mit **Ja** oder **Nein** antworten. Zusätzlich geben sie bei (2) bei der Antwort **Nein** ein unterscheidendes Wort w an. Ein unterscheidendes Wort hat die folgenden Eigenschaften: $w \in \mathsf{L} \wedge w \notin \mathsf{L}(A)$ *oder* $w \notin \mathsf{L} \wedge w \in \mathsf{L}(A)$.

(a) Diskutieren Sie zunächst, wie man ein unterscheidendes Wort für zwei Automaten finden kann bzw. wie man entscheiden kann, dass zwei Automaten gleich sind.

(b) Wenden Sie dann das Teacher-Learner-Verfahren so an, dass jedes Gruppenmitglied jede Rolle zweimal inne hatte.

Lösungsskizze

Hinweise:

- Die Learner sollten versuchen einen DFA zu konstruieren.
- Die Teacher sollten, falls sie nicht sofort ein unterscheidendes Wort finden, die Äquivalenz mit Determinisierung und Minimierung prüfen.

AUFGABE 6.3.

Stufe B

Geben Sie eine CFG G an, die genau die Terme über einer Kleene Algebra $\langle K, +, \cdot, ^*, 0, 1\rangle$ erzeugt, die mittels der Konstanten $0, 1, a, b$ gebildet werden können. Die Multiplikation $\cdot$ soll dabei ausgeschrieben werden. Zum Beispiel sollte Ihre Grammatik $a, a+0, a+b+a, a \cdot a + a^* \cdot a + b \cdot a^* + (a^*)$ erzeugen können, jedoch nicht $+a()$, ^{*1} und auch nicht ab.

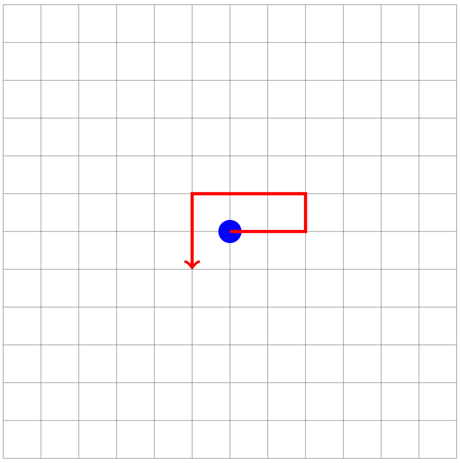
Lösungsskizze

$$T \to 0 \mid 1 \mid a \mid b \mid T+T \mid T \cdot T \mid T^* \mid (T)$$

AUFGABE 6.4. (*Pfeilsprachen*)

Stufe B – D

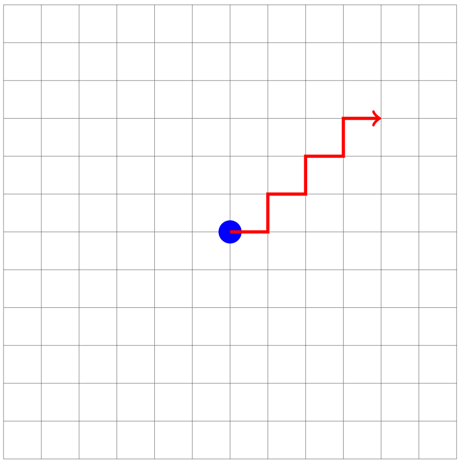
In dieser Aufgabe betrachten wir Sprachen, deren Worte Linienzüge in einem unendlichen zweidimensionalen Gitter von einem fixen Startpunkt aus beschreiben. Die folgende Grafik zeigt einen Ausschnitt aus dem Gitter:

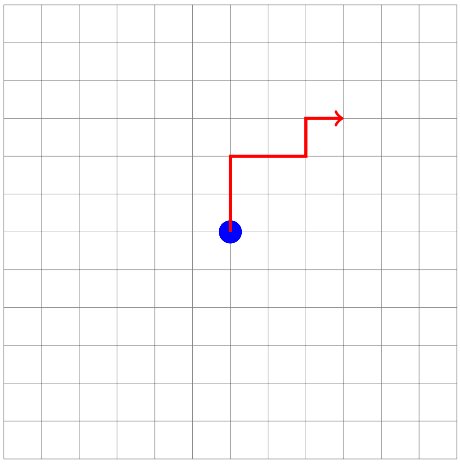


Wir haben Startpunkt blau markiert. Linienzüge beschreiben wir im Folgenden als eine Sequenz von Pfeilen, d.h. als Worte über dem Alphabet $\Sigma = \{\rightarrow, \leftarrow, \uparrow, \downarrow\}$. Die Pfeile beschreiben dabei (vom Startpunkt aus gesehen) einen ein Kästchen langen Schritt entlang des Gitters. Wir stellen daher den im Bild rot eingezeichnete Linienzug durch das Wort $w = \rightarrow\rightarrow\uparrow\leftarrow\leftarrow\leftarrow\downarrow\downarrow$ dar.

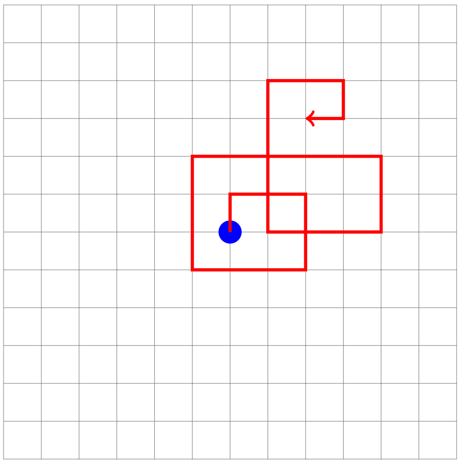
(a) Betrachten Sie die folgenden natürlich sprachlichen Beschreibungen zusammen mit jeweils einem Beispiel, welches in der Sprache liegt (auf der linken Seite), und einem Beispiel, das kein Element der Sprache ist (auf der rechten Seite). *Geben Sie für jede der Sprachen eine formale Definition der Form* $\{w \in \Sigma^* \mid \ldots\}$ *an.*[3]

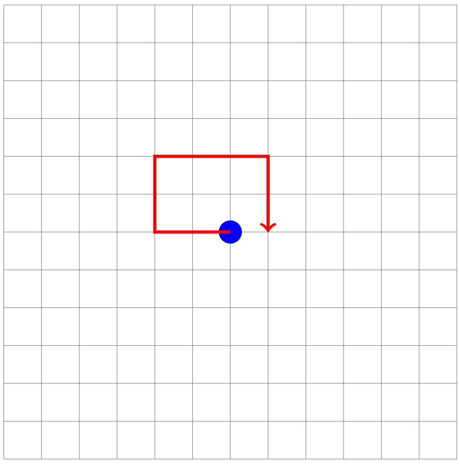
(i) die Sprache aller Treppen über dem Alphabet $\Sigma' = \{\rightarrow, \uparrow\}$



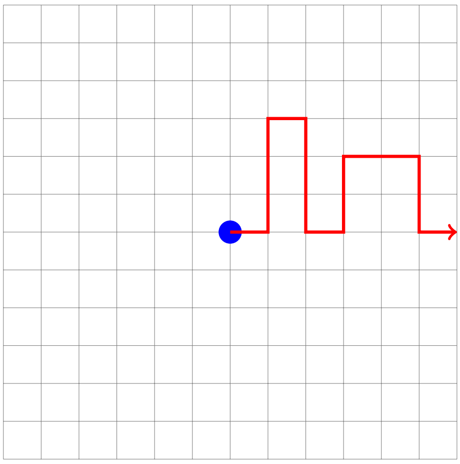


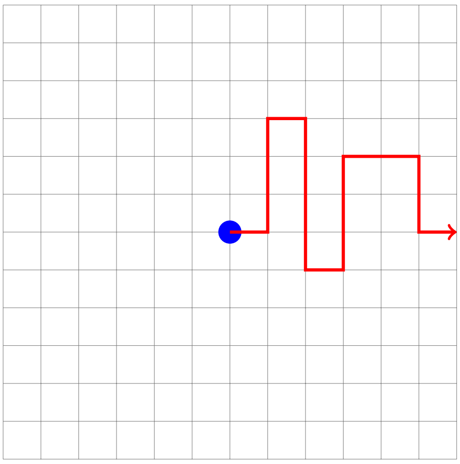
(ii) die Sprache aller im Uhrzeigersinn laufenden Spiralen über dem Alphabet Σ, die vom Startpunkt aus zuerst nach oben laufen





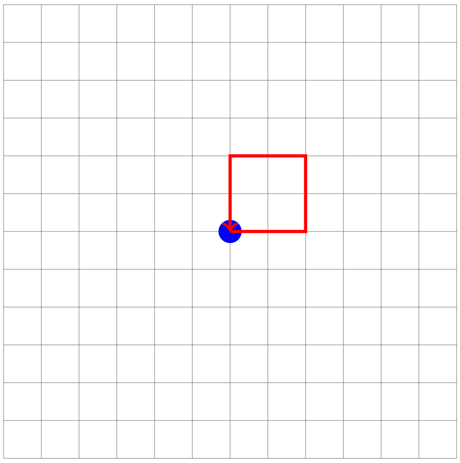
(iii) die Sprache aller "Skylines" über dem Alphabet $\Sigma'' = \{\rightarrow, \uparrow, \downarrow\}$

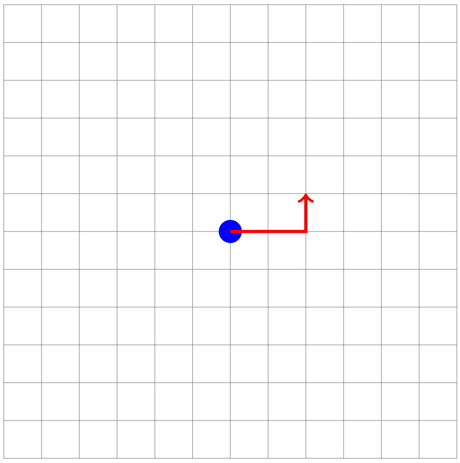




[3] Das heißt insbesondere, dass Sie in diesem Aufgabenteil keinen Automaten, keinen regulären Ausdruck, keine Grammatik oder ähnliches angeben sollen, die die Sprache beschreiben.

(iv) die Sprache aller Quadrate über dem Alphabet Σ





Hinweis: Die Sprachen sind mithilfe der Beispiele nicht eindeutig bestimmt! Ziel der Aufgabe ist es, die intuitive Beschreibung (z.B. "Sprache aller Quadrate") zusammen mit den Beispielen in eine möglichst allgemeine Sprachdefinition zu bringen.

(b) *Stellen Sie Vermutungen auf, ob die obigen Sprachen regulär, kontextfrei bzw. kontextsensitiv sind.* Begründen Sie Ihre Antwort möglichst anschaulich anhand des Beispiels. Überlegen Sie dazu, was es für Linienzüge heißt, regulär, kontextfrei bzw. kontextsensitiv zu sein.

(c) *Geben Sie zu jeder der Sprachen* L *aus Aufgabenteil (a) eine Grammatik* G *an.*

(d) Verallgemeinern Sie Aufgabenteil (a), iv) zu der Sprache aller Rechtecke, indem Sie zunächst Beispiele von Wörtern angeben, die in der Sprache liegen bzw. nicht in der Sprache liegen, und dann die Aufgabenteile (a) bis (c) auch für diese Sprache lösen.

Lösungsskizze

(a) (i) $\mathsf{L} = \{(\rightarrow\uparrow)^i \mid i \in \mathbb{N}_0\}\{\varepsilon, \rightarrow\}$

(ii) $\mathsf{L} = \{u \in \Sigma^* \mid \exists v \in \Sigma^*.\ uv \in \mathsf{L}((\uparrow^+\rightarrow^+\downarrow^+\leftarrow^+)^*)\}$

(iii) $\mathsf{L} = \{w \in \Sigma^* \mid (\forall u, v \in \Sigma^*.\ w = uv \rightarrow |u|_\uparrow \geq |u|_\downarrow) \wedge (\forall u, v \in \Sigma^*.\ \forall x, y \in \Sigma.\ w = uxyv \rightarrow xy \notin \{\uparrow\downarrow, \downarrow\uparrow\})\}$

(iv) $\mathsf{L}_1 = \{\mathsf{a}^i\mathsf{b}^i\mathsf{c}^i\mathsf{d}^i \mid i \in \mathbb{N}\}$

Wir definieren vier Homomorphismen auf Σ (mit ihrer natürlichen Erweiterung auf Σ^*) um L_1 in Σ^* einzubetten:

$$h_1: \begin{array}{lll} \mathsf{a} & \mapsto & \uparrow \\ \mathsf{b} & \mapsto & \rightarrow \\ \mathsf{c} & \mapsto & \downarrow \\ \mathsf{d} & \mapsto & \leftarrow \end{array} \qquad h_2: \begin{array}{lll} \mathsf{a} & \mapsto & \rightarrow \\ \mathsf{b} & \mapsto & \downarrow \\ \mathsf{c} & \mapsto & \leftarrow \\ \mathsf{d} & \mapsto & \uparrow \end{array} \qquad h_3: \begin{array}{lll} \mathsf{a} & \mapsto & \downarrow \\ \mathsf{b} & \mapsto & \leftarrow \\ \mathsf{c} & \mapsto & \uparrow \\ \mathsf{d} & \mapsto & \rightarrow \end{array} \qquad h_4: \begin{array}{lll} \mathsf{a} & \mapsto & \leftarrow \\ \mathsf{b} & \mapsto & \uparrow \\ \mathsf{c} & \mapsto & \rightarrow \\ \mathsf{d} & \mapsto & \downarrow \end{array}$$

$\mathsf{L}_2 = (h_1(\mathsf{L}_1) \cup h_2(\mathsf{L}_1) \cup h_3(\mathsf{L}_1) \cup h_4(\mathsf{L}_1))$
$\mathsf{L} = \mathsf{L}_2 \cup \mathsf{L}_2^R$

(b) (i) regulär, da nur eine Alternierung von konstanter Pfade verlangt wird.

(ii) regulär, da nur die Laufrichtung wichtig ist, aber nicht die Länge.

(iii) kontextfrei, da zwei Längen verglichen werden müssen.

(iv) kontextsensitiv, da vier Längen verglichen werden müssen.

(c) (i) $\mathsf{S} \mapsto\rightarrow\uparrow S \mid\rightarrow\mid \varepsilon$

(ii) $\mathsf{S} \mapsto \uparrow S \mid \uparrow T \mid \varepsilon \qquad T \mapsto \rightarrow T \mid \rightarrow U \mid \varepsilon \qquad U \mapsto \downarrow U \mid \downarrow V \mid \varepsilon \qquad V \mapsto \leftarrow V \mid \leftarrow S \mid \varepsilon$

(iii) $\mathsf{S} \mapsto T_? \mid T_? \rightarrow S \qquad T_? \mapsto \varepsilon \mid T \qquad T \mapsto \uparrow T \downarrow \mid T_? \rightarrow T_?$

(iv) Anstatt R in der Grammatik auszudrücken, verwenden wir vier weitere Homomorphismen:

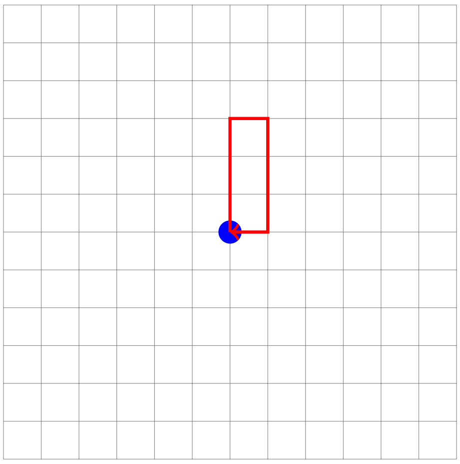
$$\begin{array}{rcll} \mathsf{S} & \mapsto & H_iS' \quad (i \in [1, 8]) & \\ S' & \mapsto & ABCDS' \mid ABCD & \\ BA & \mapsto & AB \qquad CA \mapsto AC & DA \mapsto AD \\ CB & \mapsto & BC \qquad DB \mapsto BD & \\ DC & \mapsto & AB & \end{array}$$

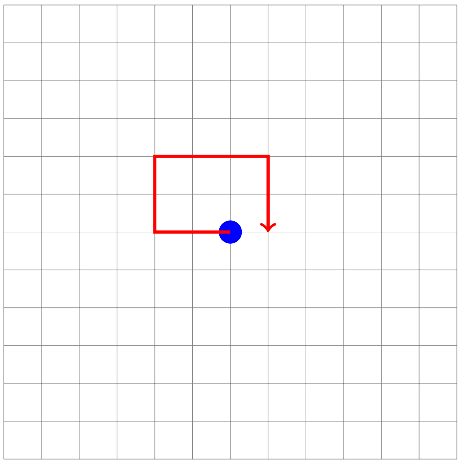
Die Homomorphismen h_i werden in der Variable H_i encodiert, z.B. h_1:

$$H_1A \mapsto\uparrow H_1 \qquad H_1B \mapsto \rightarrow H_1 \qquad H_1C \mapsto\downarrow H_1 \qquad H_1D \mapsto \leftarrow H_1 \qquad H_1 \mapsto \varepsilon$$

Diese Grammatik ist zwar nicht kontext-sensitiv, kann aber mit dem üblichen Tricks in diese Form gebracht werden.

(d) Beispiele eines Wortes in der Sprache (links) bzw. nicht der Sprache (rechts):





Die formale Definition der Sprache lautet wie folgt: $\mathsf{L} = \mathsf{L}_2 \cup \mathsf{L}_2^R$, wobei $\mathsf{L}_2 = (h_1(\mathsf{L}_1) \cup h_2(\mathsf{L}_1) \cup h_3(\mathsf{L}_1) \cup h_4(\mathsf{L}_1))$ mit den vier Homomorphismen aus Aufgabenteil a,iv) und $\mathsf{L}_1 = \{\mathsf{a}^i\mathsf{b}^j\mathsf{c}^i\mathsf{d}^j \mid i, j \in \mathbb{N}\}$
Die Grammatik ergibt sich dann wie in Aufgabenteil c,iv)