Aufgabe 1: Formale Sprachen

a) Berechnen Sie das Ergebnis der folgenden Ausdrücke.

(i) $\{\mathtt{0},\mathtt{2}\} \cdot \big(\{\mathtt{0},\mathtt{01}\} \cdot \{\mathtt{1}\} \cup \{\varepsilon,\mathtt{11},\mathtt{22}\} \cdot \{\varepsilon\}\big)$

(ii) $\{\}^2 \cdot \{\mathtt{0},\mathtt{12}\}^3$

(iii) $\{\}^* \cdot \{\mathtt{0},\mathtt{12}\}^3$

(iv) $\{\}^* \cup \{\mathtt{0},\mathtt{12}\}^2$

(v) $(\{\mathtt{0}\}^+ \cup \{\}^*)^* \cdot \{\mathtt{0}\}^+$

b) Überlegen Sie für jede der folgenden Aussagen, welche Sprachen L sie erfüllen. Wenn die Aussage von keiner oder allen Sprachen erfüllt wird, argumentieren Sie diesen Sachverhalt. Gilt eine Aussage für manche Sprachen, aber nicht für andere, geben Sie je eine Beispielsprache an, d.h., geben Sie eine Sprache an, für die die Aussage gilt, und eine, für die die Aussage nicht gilt.

(i) $L \cdot \{\mathtt{0}\} = L \cup \{\mathtt{0}\}$

(ii) $L \cdot \{\varepsilon\} = L \cup \{\varepsilon\}$

(iii) $L^2 = L$, wobei $L \neq \{\}$ und $L \neq \{\varepsilon\}$

(iv) $(L^+)^+ = L^+$

Aufgabe 2: Reguläre Sprachen und endliche Automaten 1

Sei Σ das Alphabet $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ und L die Menge aller Wörter über Σ, die entweder mit `abac` oder mit `babc` enden. Beispiele für solche Wörter sind `abac` und `babc` selber, aber auch die Wörter `ababc` und `bbabac` liegen in L.

a) Geben Sie einen regulären Ausdruck in der Notation der Posix Extended Regular Expressions an, der alle Zeilen beschreibt, die ein Wort aus L (und sonst nichts) enthalten.

b) Geben Sie einen nichtdeterministischen Automaten an, der die Sprache L akzeptiert. Der Automat soll der Definition der Sprache direkt entsprechen, sodass die Korrektheit der Modellierung unmittelbar einsichtig ist.

c) Konstruieren Sie mit Hilfe des in der Vorlesung besprochenen Determinisierungsverfahrens zu Ihrem nichtdeterministischen Automaten einen äquivalenten deterministischen.

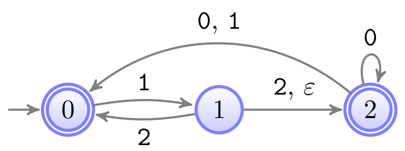
Aufgabe 3: Reguläre Sprachen und endliche Automaten 2

a) Geben Sie endliche Automaten an, die dieselben Sprachen beschreiben wie die folgenden regulären Ausdrücke in algebraischer Notation.

   (i) $\mathtt{0(12)^*1 + 0^*}$

   (ii) $\mathtt{(0 + 21^*)2^*}$

b) Konstruieren Sie zu folgendem endlichen Automaten einen regulären Ausdruck. Orientieren Sie sich am Algorithmus, der in der Vorlesung besprochen wurde und geben Sie den Automaten nach jeder Zustandselimination an.



Aufgabe 4: Reguläre Sprachen und Syntaxdiagramme

Die Modellnummer eines Panasonic-Geräts gibt Aufschluss über dessen grundlegende Kenndaten. Für Fernsehgeräte besitzt sie folgenden Aufbau:[1]

Beschreiben Sie den Aufbau derartiger Modellnummern für Fernseher mit den folgenden Methoden. Treffen Sie sinnvolle Annahmen, wenn Ihnen Informationen fehlen.

a) Geben Sie einen regulären Ausdruck in algebraischer Notation an. Nutzen Sie dabei die Möglichkeit Abkürzungen einzuführen, um einen übersichtlichen und gut nachvollziehbaren Ausdruck zu erhalten.

b) Geben Sie einen regulären Ausdruck in der Notation der POSIX Extended Regular Expressions an, der alle Zeilen beschreibt, die *ausschließlich* eine derartige Modellnummer-Zeile enthalten.

c) Zeichnen Sie das Syntaxdiagramm, das Ihrem regulären Ausdruck aus Teil a) entspricht.

[1] `https://de.tab-tv.com/?p=12177`

Aufgabe 5: Grammatik und Sprache

Songtexte enthalten gelegentlich Füllworte wie „Schubidubidu“ oder „Schubischubidubiduaaah“. Die Grammatik $G = \langle V, T, P, A \rangle$ erzeugt solche Füllwörter, wobei

$$
\begin{aligned}
V &= \{A, B, C, D\} \\
T &= \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{h}, \mathtt{i}, \mathtt{s}, \mathtt{u}\} \\
P &= \{\, A \to B\, D \;, \\
&\qquad B \to \texttt{"schu"}\, C\, B\, \texttt{"du"}\, C \mid \varepsilon \;, \\
&\qquad C \to \texttt{"bi"} \mid \texttt{"aaah"} \;, \\
&\qquad D \to \texttt{"du"} \mid \varepsilon \;\}
\end{aligned}
$$

Überprüfen Sie für die nachfolgenden Wörter, ob sie in der von der Grammatik G spezifizierten Sprache $\mathcal{L}(G)$ liegen. Falls ja, geben Sie eine Ableitung an. Falls nein, argumentieren Sie, warum nicht.

a) `schubidubidu`

b) `schubischubidubidu`

c) `schubischubidubiduaaah`

Sind folgende Aussagen wahr oder falsch? Begründen Sie Ihre Antwort!

d) Jedes Wort der Sprache besitzt eine gerade Länge.

e) Die Anzahl der `u`’s in jedem Wort der Sprache ist ungerade.

Beantworten Sie folgende Fragen.

f) Wie lautet das kürzeste Wort der Sprache?

g) Ist es möglich, die Sprache $\mathcal{L}(G)$ auch durch einen endlichen Automaten zu beschreiben? Falls ja, geben Sie einen derartigen Automaten an. Fall nein, begründen Sie, warum das nicht geht.

Aufgabe 6: Kontextfreie Produktionen

Ein *Dokument* in einer vereinfachten Version der Sprache HTML beginnt mit der Zeichenfolge „`<html>`“, dann folgen Kopf und Hauptteil des Dokumentes sowie die Zeichenfolge „`</html>`“. Der *Kopf* eines Dokumentes besteht aus dem Dokumenttitel, dem „`<head>`“ vorangeht und „`</head>`“ folgt. Der *Dokumenttitel* besteht aus einem Text eingeschlossen zwischen „`<title>`“ und „`</title>`“. Der *Hauptteil* des Dokumentes beginnt mit „`<body>`“ und endet mit „`</body>`“; dazwischen liegen Texte, geordnete und ungeordnete Listen in beliebiger Reihenfolge und Anzahl (auch gar nichts ist erlaubt). Ein *Text* ist eine nicht-leere Folge von Buchstaben, Ziffern, Leerzeichen und den Sonderzeichen „`,`“, „`;`“, „`:`“, „`.`“, „`!`“ und „`?`“. Eine *geordnete Liste* besteht aus „`<ol>`“, einer nicht-leeren Folge von Listeneinträgen und der Zeichenfolge „`</ol>`“. Ein *Listeneintrag* besteht aus einer möglicherweise leere Folge von Texte, geordneten und ungeordneten Listen, die zwischen „`<li>`“ und „`</li>`“ eingeschlossen ist. Eine *ungeordnete Liste* sieht ebenso aus wie eine geordnete, nur werden „`<ul>`“ und „`</ul>`“ an Stelle von „`<ol>`“ und „`</ol>`“ verwendet. Beispiel eines solchen Dokuments:

```
<html><head><title>Beispiel</title></head>
      <body>Ich bin ein Text. Ungeordnete Liste:
            <ul><li>Listeneintrag</li>
                <li>Geordnete Liste in einem Listeneintrag:
                    <ol><li>Schon wieder ein Eintrag.</li></ol>
                </li>
            </ul>
      </body>
</html>
```

a) Geben Sie kontextfreie Produktionen an, die derartige einfache HTML-Dokumente generieren. Verwenden Sie EBNF-Notationen, um die Grammatik übersichtlich zu strukturieren. Achten Sie auf eine eindeutige Kennzeichnung von Terminalen und Nonterminalen.

b) Geben Sie die Grammatik formal als 4-Tupel an, wobei die kontextfreien Produktionen aus der vorigen Teilaufgabe eine der vier Komponenten bilden.

c) Zeigen Sie mit Hilfe einer Parallelableitung, dass das folgende Dokument in Ihrer Grammatik ableitbar ist.

```
<html><head><title>Beispiel</title></head>
      <body>Liste:
            <ul><li>Eintrag 1</li>
                <li>Eintrag 2</li>
            </ul>
      </body>
</html>
```

Hinweis: Wenn die rechte Seite einer Produktion EBNF-Notationen enthält, handelt es sich um eine Menge an gewöhnlichen Produktionen, deren rechte Seite jeweils ein Wort aus der durch den EBNF-Ausdruck beschriebenen Menge ist. Um eine derartige Produktion anzuwenden, wählen Sie zuerst eine passende gewöhnliche Produktion aus der Menge und wenden diese dann an.

Beispiel: Die Produktion $A \to \{B\}C[D]$ repräsentiert die folgende unendliche Menge von gewöhnlichen Produktionen:

$$\begin{array}{ll} A \to C & A \to CD \\ A \to BC & A \to BCD \\ A \to BBC & A \to BBCD \\ A \to BBBC & A \to BBBCD \\ A \to \dots & A \to \dots \end{array}$$

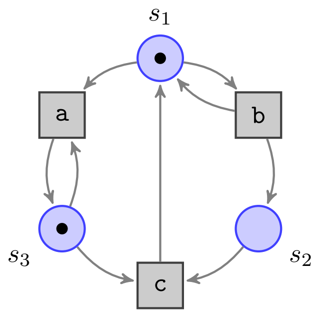
Die Produktion $A \to \{B\}C[D]$ auf ein Wort anzuwenden bedeutet, sich aus obiger Liste eine passende Produktion, etwa $A \to BBCD$, auszusuchen und danach im Wort ein Vorkommen des Nonterminals A durch $BBCD$ zu ersetzen.

d) Handelt es sich bei der Menge der einfachen HTML-Dokumente um eine reguläre Sprache, d.h., lassen sich die Zeichenketten, die ein zulässiges HTML-Dokument darstellen, im Prinzip auch durch einen (komplizierten) regulären Ausdruck spezifizieren? Begründen Sie Ihre Antwort.

e) Ändert sich etwas an der Antwort auf die Frage aus Teilaufgabe d), wenn die Länge der Eingabe auf 1024 Zeichen begrenzt wird? Begründen Sie Ihre Antwort.

Aufgabe 7: Funktionsweise von Petri-Netzen

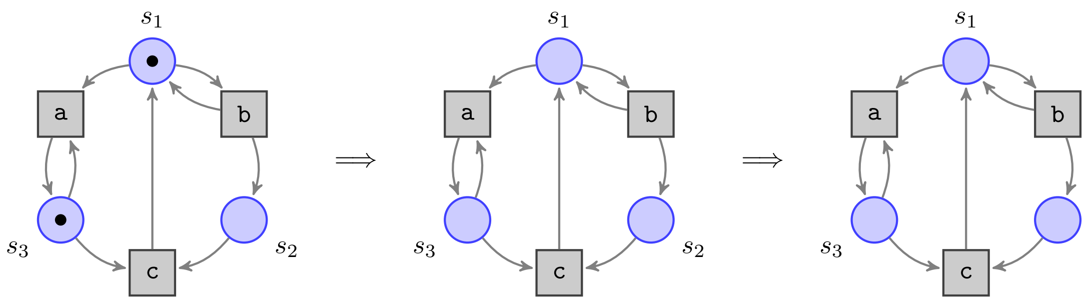
Gegeben sei das folgende markierte Petri-Netz.



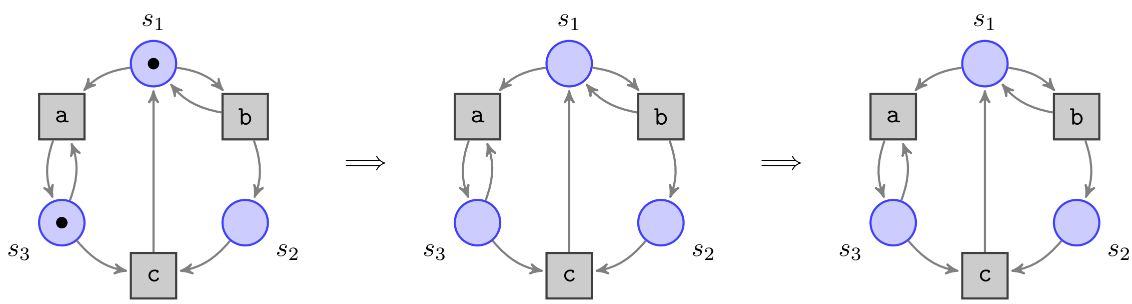
a) Beschreiben Sie das Petri-Netz durch ein 5-Tupel.

b) Welche Transitionen sind anfänglich aktiviert?

c) Lassen Sie ausgehend von der ursprünglichen Markierung zwei verschiedene Transitionen hintereinander feuern. Geben Sie sowohl die Transitionen, die Sie feuern lassen, als auch die Markierung nach jedem Feuern an.



d) Lassen Sie ausgehend von der ursprünglichen Markierung so lange Transitionen hintereinander feuern, bis ein Deadlock erreicht wird, d.h., bis eine Markierung erreicht wird, in der keine Transition aktiviert ist. Erklären Sie, warum es sich um einen Deadlock handelt.



Aufgabe 8: Modellieren mit Petri-Netzen

Der Great Dragon Summit (kurs GDS) ist ein beliebtes touristisches Ziel, allerdings hat es der Berg in sich. Aus Sicherheitsgründen ist die Besteigung nur geführt möglich. Die Touristen reisen aus dem Tal an und treffen im Basislager eine der zertifizierten Führerinnen. Jede Führerin nimmt zwei Kunden mit auf die Tour. Der Weg führt vom Basislager über einen Wasserfall auf den Gipfel und dann denselben Weg wieder zurück zum Lager. Der Ehrenkodex der Führerinnen und die geltenden Gesetze verbieten es, dass sich das Team unterwegs trennt (niemand wird zurückgelassen). Zurück im Lager bezahlen die Kunden, ehe sie ins Tal absteigen, während die Führerin auf die nächsten Kunden wartet.

a) Modellieren Sie den GDS-Tourismus mit Hilfe eines Petri-Netzes. Nehmen Sie an, dass drei Führerinnen zur Verfügung stehen. Gestalten Sie das Petri-Netz offen, sodass ständig Touristen nachkommen und das Netz wieder verlassen können (Transition ohne Eingangs- bzw. Ausgangsstellen). Beschriften Sie Transitionen und Stellen geeignet, um Ihr Modell verständlich zu halten.

b) Wenn sich in der Engstelle beim Wasserfall zwei Teams treffen, kommt es aufgrund des Gedränges immer wieder zu Unfällen. Daher wird vereinbart, dass der Wasserfall immer nur von einem Team besucht werden darf, während die anderen darunter oder darüber warten müssen. Ergänzen Sie Ihr Petri-Netz entsprechend.

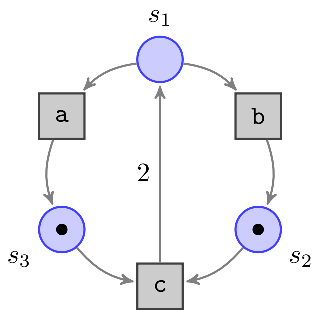
c) Das unvergessliche Erlebnis mit GDS spricht sich herum, sodass die Zahl der Führerinnen auf sechs erhöht wird. Die große Zahl an Besuchern macht die Tour weniger attraktiv und gefährlicher. Daher wird festgelegt, dass sich nie mehr als vier Teams auf Tour befinden dürfen. Ergänzen Sie Ihr Petri-Netz entsprechend.

d) Manche Führerinnen haben wenig Disziplin und drängen sich vor, wenn ihr Team im Lager warten muss. Daher führt der Lagerleiter eine Liste ein, damit die Teams in der Reihenfolge, in der sie sich eintragen, auf den Berg gehen können. Ergänzen Sie Ihr Petri-Netz entsprechend. Es genügt, zwei Listenplätze vorzusehen, da vier Teams ja ohnehin ohne Warten auf den Berg dürfen.

Aufgabe 9: Abläufe im Petri-Netz

Jedes Petri-Netz kann als Automat aufgefasst werden, dessen momentaner Zustand durch die Markierung des Petri-Netzes festgelegt ist. Feuert eine Transition, führt das zu einer neuen Markierung und damit zu einem neuen Zustand. Die Bezeichnungen der Transitionen spielen dabei die Rolle eines Alphabets, dessen Symbole zu den Zustandsänderungen führen. Wir können alle Folgen von Transitionen, die von der Anfangsmarkierung ausgehend hintereinander feuern können, als die Sprache des Petri-Netzes auffassen. Aus Sicht des Automaten bedeutet das, dass alle Zustände auch Endzustände sind.

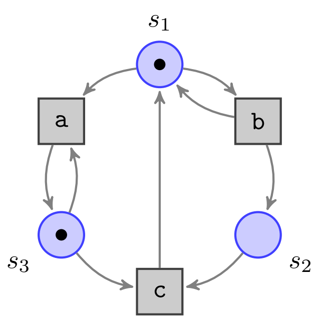
Gegeben sei folgendes Petri-Netz samt Anfangsmarkierung:



a) Zählen Sie die unterschiedlichen Markierungen auf, die das Petri-Netz von der angegebenen Markierung aus durch das Feuern von Transitionen erreichen kann. (Es sollten sechs sein.)

b) Fassen Sie die Markierungen als die Zustände eines endlichen Automaten auf und beschreiben Sie das Verhalten des Petri-Netzes durch einen Automaten, mit den Bezeichnungen der feuernden Transitionen als Alphabet.

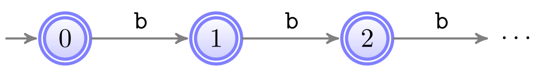
c) Betrachten Sie nun das Petri-Netz, das wir schon aus Aufgabe 7 kennen:

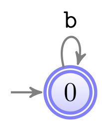


Warum führt derselbe Ansatz wie oben zu keinem endlichen Automaten? Skizzieren Sie den Automaten, der sich ergibt.

d) Geben Sie einen *endlichen* Automaten für die Sprache des Petri-Netzes aus Teilaufgabe c) an. Betrachten Sie dafür die auftretenden Transitionsfolgen und beschreiben Sie diese direkt, ohne jeder Markierung einen eigenen Zustand zuzuordnen.

Alternativ können Sie auch von Ihrer Skizze des Automaten in Teilaufgabe c) ausgehen und solange ununterscheidbare Zustände zusammenfassen, bis Sie einen endlichen Automaten erhalten. Zwei Zustände heißen ununterscheidbar, wenn die Wörter, die von dort aus akzeptiert werden, dieselben sind. Beispielsweise sind die Zustände im unendlichen Automaten unten links ununterscheidbar: Von jedem Zustand aus werden alle Wörter akzeptiert, die aus beliebig vielen `b`s bestehen. Fasst man die Zustände zusammen, erhält man den endlichen Automaten rechts.





e) *Bonusaufgabe:* Erklären Sie informell, warum sich die Sprache des folgenden Petri-Netzes nicht durch einen endlichen Automaten beschreiben lässt.