2. Übungsblatt

3.0 VU Formale Modellierung

Marion Scholz, Gernot Salzer

7. November 2014

Allgemeines

Dieses Übungsblatt enthält Aufgaben zu den endlichen Automaten, zu kontextfreien Grammatiken, zur Prädikatenlogik und zu Petri-Netzen. Bitte beachten Sie:

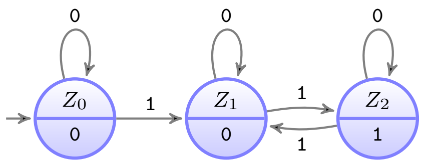
- Geben Sie Ihre Lösung als **ein einziges Pdf-Dokument** in TUWEL ab.
- Beachten Sie die **Abgabefrist**: Donnerstag, 20. November 2014, 23:55
- Verabsäumen Sie nicht, sich für das **Tutoren- und das Abgabegespräch** (2 Termine) in TUWEL anzumelden. Die Anmeldung wird am Tag nach der Abgabefrist für das Übungsblatt geöffnet.
- Gönnen Sie sich eine **Zeitreserve**, d.h., warten Sie nicht bis zum letzten Augenblick mit dem Hochladen der Lösungen. Mögliche Probleme in letzter Sekunde (aus den letzten Semestern):
  - Wie erzeuge ich eine PDF-Datei aus meinem Word-Dokument oder aus meinen Zetteln?
  - Wie bekomme ich alle Seiten in eine einzige PDF-Datei?
  - Das Internet/TUWEL/der Computer funktioniert plötzlich nicht.
  - Eine falsche Datei wird hochgeladen oder die richtige unvollständig, und es ist keine Zeit mehr für eine Wiederholung.
  - Krankheit
- Verständigen Sie uns im Falle eines **Notfalls** ehestmöglich per E-Mail an

  `fmod14w@logic.at` .

  In der Regel erwarten wir einen Nachweis des Notfalls (etwa ärztliche Bestätigung im Fall einer Erkrankung).
- Lösen Sie die Beispiele selbständig. Wir weisen darauf hin, dass **Plagiate** mit **null Punkten** beurteilt werden. Ein Plagiat entsteht sowohl dann, wenn Sie abschreiben, als auch dann, wenn Sie abschreiben lassen.

Aufgaben

Aufgabe 1 (0.3 Punkte) Sei $\mathcal{A}$ der folgende Moore-Automat.



(a) Geben Sie die Ausgaben zu folgenden Eingaben an: `00111`, `01111`, `10100`.

(b) Berechnen Sie schrittweise $\delta^*(Z_0, \texttt{01101})$ und $\gamma^*(Z_0, \texttt{01101})$.

(c) Beschreiben Sie die Übersetzungsfunktion $[\mathcal{A}]$.

Aufgabe 2 (0.3 Punkte) Sei L die Sprache

$$\{\, w \in \{\texttt{a}, \texttt{i}, \texttt{l}, \texttt{x}\}^* \mid w \text{ enthält das Teilwort } \texttt{lila} \text{ oder } \texttt{ill} \,\} \ .$$

Geben Sie einen *deterministischen* Automaten für L an. Gehen Sie folgendermaßen vor:

1. Konstruieren Sie einen indeterministischen Automaten für diese Sprache.
2. Wandeln Sie den indeterministischen Automaten mit Hilfe des in der Vorlesung besprochenen Verfahrens in einen deterministischen um.

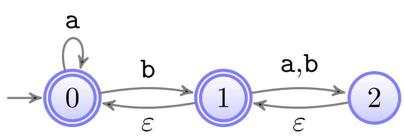
Ist der so gewonnene Automat minimal, d.h., ist er unter den deterministischen Automaten für L jener mit der kleinstmöglichen Zahl von Zuständen?

Aufgabe 3 (0.3 Punkte) Geben Sie endliche Automaten an, die dieselbe Sprache beschreiben wie die folgenden regulären Ausdrücke.

(a) $(\texttt{a} + \texttt{b})^+(\texttt{ac})^*\texttt{b}$

(b) $(\texttt{ab} + \texttt{b})(\texttt{ab})^+ + \texttt{a}$

Aufgabe 4 (0.4 Punkte) Geben Sie einen regulären Ausdruck an, der dieselbe Sprache beschreibt wie der folgende endliche Automat. Verwenden Sie dabei das in der Vorlesung beschriebene Verfahren und geben Sie den Automaten nach jeder Zustandselimination an.



Aufgabe 5 (0.3 Punkte) An der TU beschäftigte Personen erhalten Mailadressen der Form

$$\mathit{Vorname}\,\texttt{.}\,\mathit{Mittelteil}\,\texttt{.}\,\mathit{Zuname}\texttt{+}\mathit{OrgEinheit}\texttt{@tuwien.ac.at}$$

wobei die Teile „`.` *Mittelteil*" und „`+`*OrgEinheit*" optional sind. Die Namensteile bestehen aus Groß- und Kleinbuchstaben, wobei Umlaute und scharfes S durch ae, oe, ue, ss etc. zu ersetzen sind. *OrgEinheit* ist die Kennung des Instituts oder der Serviceeinrichtung, der die Person angehört; sie besteht aus dem Zeichen `E` gefolgt von drei Ziffern und einem optionalen Großbuchstaben. Beispiele:

`Hikaru.Sulu@tuwien.ac.at`
`James.Tiberius.Kirk@tuwien.ac.at`
`Beverly.Crusher+E010C@tuwien.ac.at`
`Nyota.Penda.Uhura+E006@tuwien.ac.at`[1]

(a) Geben Sie einen regulären Ausdruck in algebraischer Notation an, der die Menge derartiger Emailadressen beschreibt.

(b) Geben Sie einen POSIX Extended Regular Expression an, der alle Zeilen beschreibt, die ausschließlich eine derartige Emailadresse enthalten.

(c) Geben Sie das Syntaxdiagramm an, das Ihrem regulären Ausdruck aus Teilaufgabe a entspricht.

Aufgabe 6 (0.3 Punkte) Bertram der Biber friert im Winter regelmäßig, er zittert und bibbert. Seine Mitbiber geben ihm daher Namen wie bibberbibberbibbernderbiber, bibbernderbibberbibberbibberbibberbiber, oder einfach nur bibberbiber. Einem Biberforscher gelingt es, diese Bibbersprache durch die Grammatik $G = \langle N, T, P, A\rangle$ zu beschreiben, wobei

$$\begin{aligned}
N &= \{A, B, C, D, E\}\\
T &= \{\texttt{b}, \texttt{d}, \texttt{e}, \texttt{i}, \texttt{n}, \texttt{r}\}\\
P &= \{\, A \to \texttt{bib}\, B\\
&\qquad B \to \texttt{ber}\, C \mid \texttt{er}\\
&\qquad C \to \texttt{bib}\, B \mid \texttt{nder}\, D\\
&\qquad D \to \texttt{bib}\, E\\
&\qquad E \to \texttt{ber}\, D \mid \texttt{er}\ \}
\end{aligned}$$

(a) 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 Parallelableitung an. Falls nein, argumentieren Sie, warum nicht.

(1) `bibberndebibberbibber`

(2) `bibberbibberndebiber`

[1] Wer von diesen vier Personen passt nicht zu den anderen?

(3) `bibbernderbibbernderbiber`

(b) Sind folgende Aussagen über die Sprache $\mathcal{L}(G)$ korrekt? Wenn ja, warum? Wenn nein, geben Sie ein Gegenbeispiel!

(1) Jedes Wort endet mit `biber`.

(2) Die Anzahl der `bibber` vor bzw. nach `nder` im Wort sind gleich.

(3) Jedes Wort beginnt mit `bib`.

(c) Konstruieren Sie einen endlichen Automat, der die Sprache $\mathcal{L}(G)$ akzeptiert.

Aufgabe 7 (0.4 Punkte) In Büchern über formale Sprachen ist folgende Definition zu finden:

> Eine *reguläre Grammatik* wird durch ein 4-Tupel $G = \langle V, T, P, S\rangle$ festgelegt, wobei
>
> - V und T endliche, disjunkte Mengen von Symbolen sind ($V \cap T = \{\}$),
> - S ein Symbol aus V ist ($S \in V$) und
> - $P \subseteq V \times (T \cdot V \cup \{\varepsilon\})$ eine endliche Menge von Paaren ist.
>
> Die Elemente von P werden Produktionen genannt; statt $(x, y) \in P$ wird auch $x \to y$ geschrieben. Die Notation $x \to y_1 \mid \cdots \mid y_n$ ist eine Abkürzung für die Produktionen $x \to y_1, \ldots, x \to y_n$.
>
> Das Wort $u\,y\,v$ ist aus dem Wort $u\,x\,v$ in einem Schritt ableitbar, geschrieben $u\,x\,v \Rightarrow u\,y\,v$, wenn $x \to y$ gilt. Die von G generierte Sprache $\mathcal{L}(G)$ ist definiert als die Menge $\{\, w \in T^* \mid S \overset{*}{\Rightarrow} w \,\}$, wobei $\overset{*}{\Rightarrow}$ den reflexiven und transitiven Abschluss von $\Rightarrow$ bezeichnet.[2]

(a) Geben Sie an, welche der folgenden Tupeln eine reguläre Grammatik gemäß der obigen Definition darstellt. Begründen Sie Ihre Antwort, falls es sich um keine reguläre Grammatik handelt. Entspricht das Tupel der Definition, geben Sie die Sprache an, die durch die Grammatik generiert wird.

i. $\langle \{\mathtt{X}\}, \{\mathtt{a}, \mathtt{b}\}, \{\mathtt{X} \to \mathtt{aXb} \mid \varepsilon\}, \mathtt{X}\rangle$

ii. $\langle \{\mathtt{X}, \mathtt{Y}\}, \{\mathtt{a}, \mathtt{b}\}, \{\mathtt{X} \to \mathtt{aX} \mid \mathtt{bY} \mid \varepsilon, \mathtt{Y} \to \mathtt{bY}\}, \mathtt{X}\rangle$

iii. $\langle \{\mathtt{a}, \mathtt{b}\}, \{\mathtt{X}, \mathtt{Y}\}, \{\mathtt{a} \to \mathtt{Xa} \mid \mathtt{Yb} \mid \varepsilon, \mathtt{b} \to \mathtt{Yb}\}, \mathtt{b}\rangle$

iv. $\langle \{\mathtt{X}, \mathtt{Y}\}, \{\mathtt{a}, \mathtt{b}\}, \{\mathtt{X} \to \mathtt{Xa} \mid \mathtt{Yb} \mid \varepsilon, \mathtt{Y} \to \mathtt{Yb} \mid \varepsilon\}, \mathtt{X}\rangle$

---

[2] Das heißt, dass $\overset{*}{\Rightarrow}$ die kleinste Relation mit folgenden Eigenschaften ist:

- Aus $u \Rightarrow v$ folgt $u \overset{*}{\Rightarrow} v$.
- Es gilt $u \overset{*}{\Rightarrow} u$ für alle Wörter $u \in T^*$.
- Aus $u \overset{*}{\Rightarrow} v$ und $v \overset{*}{\Rightarrow} w$ folgt $u \overset{*}{\Rightarrow} w$.

Anschaulich gesprochen steht $\overset{*}{\Rightarrow}$ für die Ableitbarkeit in beliebig vielen Schritten.

(b) Beschreiben Sie ein Verfahren, das zu einer regulären Grammatik einen äquivalenten endlichen Automaten liefert.

(c) Beschreiben Sie ein Verfahren, das zu einem endlichen Automaten ohne ε-Übergängen eine äquivalente reguläre Grammatik liefert. Lässt sich Ihr Verfahren auch für Automaten mit ε-Übergängen verwenden?

Aufgabe 8 (0.4 Punkte) Angenommen es sind zwei deterministische Automaten $\mathcal{A}$ und $\mathcal{B}$ gegeben. Geben Sie ein allgemeines Verfahren an um einen endlichen Automaten zu konstruieren, der genau jene Wörter akzeptiert, die von $\mathcal{A}$ aber nicht von $\mathcal{B}$ akzeptiert werden. Welche Eigenschaft regulärer Sprachen lässt sich daraus ablesen?

Aufgabe 9 (0.3 Punkte) Die *Planning Domain Definition Language* (PDDL) wird zur Beschreibung von Planungsproblemen verwendet. Das folgende PDDL-Beispiel beschreibt ein Problem namens „`Blocksworld-Instanz-1`“ aus dem Bereich „`Blocksworld`“, in dem es die Objekte „`A`“, „`B`“ und „`C`“ vom Typ „`Block`“ sowie das Objekt „`D`“ vom Typ „`Cube`“ gibt; der „`:init`“-Abschnitt beschreibt die gegebene Ausgangssituation, der „`:goal`“-Abschnitt die gewünschte Zielsituation.

```
(define (problem Blocksworld-Instanz-1)
        (:domain Blocksworld)
        (:objects A B C - Block D - Cube)
        (:init (Clear A) (Clear C) (On C B) (OnTable A) (OnTable B) (HandEmpty) )
        (:goal (OnTable C) (On A B) (On B C) )
)
```

Grundsätzlich bestehen Problembeschreibungen aus ineinander geschachtelten Listen, die jeweils in runden Klammern eingeschlossen sind. Die Listenelemente werden durch Leerzeichen getrennt. Es genügt, wenn Sie einzelne Leerzeichen zur Trennung vorsehen, mehrfache Leerzeichen und Zeilenumbrüche müssen nicht berücksichtigt werden.
Die oberste Liste besteht aus sechs Elementen, nämlich dem Schlüsselwort „`define`“ gefolgt von Listen für die Problembezeichnung, die Bereichsbezeichnung, die Objektdefinitionen, die Ausgangs- und die Zielsituation (wie im Beispiel oben).
„`define`“, „`problem`“, „`:domain`“, „`:objects`“, „`:init`“ und „`:goal`“ sind Schlüsselwörter. Alle anderen Bezeichnungen bestehen aus Buchstaben, Ziffern und Bindestrichen, beginnen aber immer mit einem Buchstaben.
Objekte werden im „`:objects`“-Abschnitt definiert. Der Liste von Objekten folgt getrennt durch einen Bindestrich ihr Typ. Darauf können weitere Objekte mit einer weitere Typangabe (im Beispiel oben „`Cube`“) folgen, und so weiter.
Der „`:init`“- und der „`:goal`“-Abschnitt besteht jeweils aus einer Folge von sogenannten Prädikaten. Jedes Prädikat ist eine Liste, die mit einem Prädikatnamen beginnt, dem Argumente folgen können. Etwa enthält das Prädikat „`(Clear A)`“ den Prädikatnamen „`Clear`“ und das Argument „`A`“. „`(HandEmpty)`“ enthält nur einen Prädikatnamen, aber keine Argumente. Im Prädikat „`(On A B)`“ folgen dem Prädikatnamen zwei Argumente.
Geben Sie eine strukturierte kontextfreie Grammatik in EBNF an, die den Aufbau derartiger Problembeschreibungen spezifiziert.

Aufgabe 10 (0.3 Punkte) Sei $\mathcal{E}$ die Sprache der Posix Extended Regular Expressions, wobei nur die in der folgenden Tabelle angeführten Sprachelemente zugelassen sind.

regexp	trifft zu auf	*regexp*	trifft zu auf
$\texttt{\\}s$	Zeichen s	rr'	r gefolgt von r'
s	s, falls kein Sonderzeichen	$r\texttt{\|}r'$	r oder r'
.	alle Zeichen	$r\texttt{*}$	≥ 0 Mal r
$\texttt{[}s_1 \cdots s_n\texttt{]}$	ein Zeichen aus $\{s_1, \ldots, s_n\}$	$r\texttt{+}$	≥ 1 Mal r
$\texttt{(}r\texttt{)}$	r	$r\texttt{?}$	≤ 1 Mal r

Weiters ist das Alphabet auf die Zeichen `A`, `9`, `\`, `.`, `[`, `]`, `(`, `)`, `|`, `*`, `+`, `?` und `!` eingeschränkt, es besteht also nur aus den in der Tabelle auftretenden Sondersymbolen sowie aus den drei Symbolen `A`, `9` und `!`.
Beispiel: Das Wort `([A9.]*(!|\?)?)+` liegt in der Sprache $\mathcal{E}$.

Spezifizieren Sie die Sprache der Posix Extended Regular Expressions mit Hilfe einer kontextfreien Grammatik. Verwenden Sie so weit als möglich Ebnf-Notationen, um die Grammatik übersichtlich zu halten und rekursive Regeln zu vermeiden.

Aufgabe 11 (0.4 Punkte) Zeigen Sie mit Hilfe der algebraischen Gesetze für logische Operatoren und Quantoren, dass die beiden Formeln jeweils äquivalent zueinander sind. Übersetzen Sie jede Formel in natürliche Sprache, wobei die Prädikatensymbolen die durch ihren Namen nahegelegten Bedeutungen haben sollen.

(a) $\neg\forall x\,(\mathit{Mann}(x) \supset \exists y\,(\mathit{Frau}(y) \land \mathit{Liebt}(x,y)))$ und
$\exists x\,(\mathit{Mann}(x) \land \forall y\,(\mathit{Frau}(y) \supset \neg\mathit{Liebt}(x,y)))$

(b) $\forall x\,((\mathit{Frau}(x) \land \exists y \mathit{Mann}(y)) \supset \forall z\,\mathit{Liebt}(x,y)))$ und
$\forall x\,\neg\mathit{Mann}(x) \lor \forall z\,(\mathit{Frau}(z) \supset \mathit{Liebt}(z,y))$

Aufgabe 12 (0.5 Punkte) Seien *Person*/1, *Nass*/1, *Hobby*/1 und *Betreibt*/2 Prädikatensymbole sowie *surfen* und *tauchen* Konstantensymbole mit folgender Bedeutung:

$\mathit{Person}(x)$	… x ist eine Person	$\mathit{Betreibt}(x,y)$	… x betreibt y
$\mathit{Nass}(x)$	… x ist nass	*surfen*	… Surfen
$\mathit{Hobby}(x)$	… x ist ein Hobby	*tauchen*	… Tauchen

Verwenden Sie diese Symbole, um die folgenden Sätze in prädikatenlogische Formeln zu übersetzen.

(a) Alle Personen, die Surfen oder Tauchen betreiben, sind nass.

(b) Es gibt Personen, die alle nassen Hobbies betreiben.

Sei weiters folgende Interpretation I gegeben:

$$\begin{aligned}
\mathcal{U} &= \{\text{Tom}, \text{Anna}, \text{Lisa}, \text{Karin}, \text{Schwimmen}, \text{Surfen}, \text{Segeln}, \text{Eislaufen},\\
&\quad\ \text{Tauchen}, \text{Radfahren}, \text{Lesen}, \text{Laufen}\}\\
I(\mathit{Person}) &= \{\text{Tom}, \text{Lisa}, \text{Karin}\}\\
I(\mathit{Nass}) &= \{\text{Schwimmen}, \text{Surfen}, \text{Segeln}, \text{Eislaufen}\}\\
I(\mathit{Hobby}) &= \{\text{Schwimmen}, \text{Surfen}, \text{Tauchen}, \text{Radfahren}, \text{Lesen}, \text{Laufen}\}\\
I(\mathit{Betreibt}) &= \{(\text{Tom}, \text{Eislaufen}), (\text{Tom}, \text{Radfahren}), (\text{Tom}, \text{Segeln}),\\
&\quad\ (\text{Anna}, \text{Radfahren}), (\text{Anna}, \text{Tauchen}), (\text{Anna}, \text{Eislaufen}),\\
&\quad\ (\text{Lisa}, \text{Surfen}), (\text{Lisa}, \text{Schwimmen}), (\text{Lisa}, \text{Segeln}),\\
&\quad\ (\text{Karin}, \text{Radfahren}), (\text{Karin}, \text{Eislaufen})\}\\
I(\mathit{surfen}) &= \text{Surfen}\\
I(\mathit{radfahren}) &= \text{Radfahren}
\end{aligned}$$

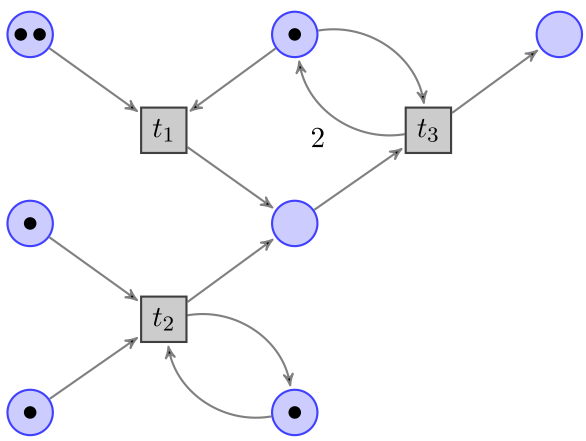
Übersetzen Sie die nachfolgenden Formeln in natürliche Sprache. Geben Sie an, ob die Formeln in der Interpretation I wahr oder falsch sind. Begründen Sie Ihre Antwort; es ist keine formale Auswertung erforderlich.

(c) $\exists x\,(\mathit{Betreibt}(x, \mathit{surfen}) \land \neg \mathit{Betreibt}(x, \mathit{radfahren}))$

(d) $\forall x\,(\mathit{Nass}(x) \supset (\mathit{Person}(x) \land \exists y\, \mathit{Betreibt}(x, y)))$

(e) $\exists x\,(\mathit{Person}(x) \land \exists y\,(\mathit{Hobby}(y) \land \mathit{Betreibt}(x, y)))$

Aufgabe 13 (0.4 Punkte) Geben Sie für das folgende Petri-Netz mit Anfangsmarkierung alle möglichen Reihenfolgen an, in denen die Transitionen feuern können, sowie die Endmarkierungen, die dadurch erreicht werden. (Endmarkierung bedeutet, dass keine Transition aktiviert ist.)



Aufgabe 14 (0.4 Punkte) Beim Boarding eines Flugzeugs werden zwei Schlangen gebildet, „regular“ und „premium“, wobei die Premiumschlange für Vielflieger gedacht ist. Jede der beiden Schlangen wird von einem Mitarbeiter der Fluglinie betreut. Der Prozess ist bei beiden Schlangen ident: Zuerst wird der Reisepass kontrolliert und dann der Barcode auf dem Flugticket gescannt.

(a) Modellieren Sie dieses System mit Hilfe eines Petri-Netzes. Geben Sie eine geeignete Anfangsmarkierung für fünf reguläre und zwei Vielflieger an.

(b) Da es meist mehr reguläre als Premiumgäste gibt, ist der Premiumbetreuer weniger ausgelastet als der andere. Erweitern Sie Ihr Modell dahingegend, dass der Premiumbetreuer, wenn er nicht gerade einen Passagier abfertigt, Fluggäste von der regulären Schlange in die Premiumschlange umreihen kann.