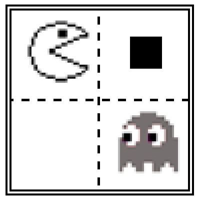
3.0/2.0 VU Formale Modellierung 185.A06 WS 2016/SS 2017 31. Mai 2017			
Matrikelnummer	Nachname	Vorname	Gruppe **A**

Aufgabe 1 (10 Punkte) Bei Pac-Man, einem Videospiel aus den 80ern, muss die Spielfigur Pac-Man Punkte in einem Labyrinth fressen, während sie von Geistern verfolgt wird. Pac-Man und jeder Geist kann pro Zug ein Feld nach oben, unten, links oder rechts bewegt werden, vorausgesetzt, es ist keine Mauer im Weg. Bewegt Pac-Man sich auf ein Feld mit einem Punkt, so frisst er diesen Punkt. Sind alle Punkte gefressen, gilt das Level als gewonnen. Befinden sich Pac-Man und ein Geist auf dem gleichen Feld, so wird Pac-Man vom Geist gefressen und das Spiel ist beendet. Befinden sich der letzte Punkt, Pac-Man und ein Geist auf demselben Feld, hat der Geist Vorrang, d.h., Pac-Man verliert auch in diesem Fall.

Nehmen Sie an, dass das aktuelle Level, das Pac-Man bewältigen muss, so aussieht wie rechts skizziert. Zu Beginn befindet sich Pac-Man im linken oberen Feld, im rechten unteren Feld ist ein Geist und im rechten oberen Feld ist ein Punkt (schwarzes Quadrat). Rundherum ist eine Mauer (doppelte Linie). Pac-Man gewinnt das Level, wenn er den einen Punkt frisst. Pac-Man verliert, falls er von dem Geist gefressen wird.



- Überlegen Sie, welche Informationen notwendig sind, um den Zustand des Systems zu beschreiben.
- Legen Sie die möglichen Aktionen fest, die zu einem Zustandswechsel führen können.
- Geben Sie einen endlichen Automaten an, der das Systemverhalten vollständig beschreibt. Gehen Sie dabei davon aus, dass Pac-Man und der Geist abwechselnd je einen Zug machen, wobei Pac-Man beginnt.

Aufgabe 2 (10 Punkte) Thomas hat sich ein Haus mit Garten gekauft. Er hat zum ersten Mal einen Garten und hat keine Ahnung, was er alles im Frühling machen muss, damit sein Rasen gut gedeiht. Auch eine intensive Recherche im Internet hat ihn mehr verwirrt als zu helfen. Thomas hat alle Hinweise, die er gefunden hat zusammengeschrieben:

- Häufiges Gießen des Rasens führt zu schnellerem Wachstum. Dann muss der Rasen allerdings auch oft gemäht werden.
- Immer wenn der Rasen vertikutiert wird, sollten auch Samen gestreut werden, um den Rasen schnell wieder instandzusetzen. Ebenso macht Samen streuen nur Sinn, wenn zuvor vertikutiert wurde.
- Man sollte Samen streuen oder häufig den Rasen mähen. Beides gemeinsam darf nicht durchgeführt werden, sonst würde der Rasenmäher die frischen Keime zerstören.
- Wenn man vertikutiert, dann sollte man gießen aber nicht den Rasen mähen.

Er bittet seinen Freund Markus um Hilfe, der sich mit Aussagenlogik auskennt. Er weist ihn noch darauf hin, dass er nicht sehr viel Zeit hat und daher nur zwei Tipps befolgen kann.

a) Formalisieren Sie die beschriebene Situation inklusive aller Anhaltspunkte mittels aussagenlogischer Formeln. Geben Sie die Bedeutung der von Ihnen verwendeten Aussagenvariablen an.

b) Kann Markus Thomas helfen und einen Rat geben? Begründen Sie die Antwort mit Hilfe Ihrer aussagenlogischen Modellierung.

Aufgabe 3 (10 Punkte) Seien *Bewohnt*/2, *Orden*/1, *Planet*/1 und *Alt*/1 Prädikatensymbole sowie *naboo* und *alderaan* Konstantensymbole mit folgender Bedeutung:

$Orden(x)$	… x ist ein Orden	$Bewohnt(x, y)$	… x bewohnt y
$Planet(x)$	… x ist ein Planet	*naboo*	… Naboo
$Alt(x)$	… x ist alt	*alderaan*	… Alderaan

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

a) Jeder alte Orden bewohnt entweder Naboo oder Alderaan.

b) Es gibt alte Planeten, die von allen Orden bewohnt werden.

Sei weiters folgende Interpretation I gegeben:

$$\begin{aligned}
\mathcal{U} &= \{\text{Jedi-Ritter}, \text{Sith}, \text{Matukai}, \text{Watto}, \text{Mustafar}, \text{Zeison-Sha}, \text{Naboo}, \text{Alderaan},\\
&\qquad \text{Bespin}, \text{Endor}\}\\
I(\mathit{Orden}) &= \{\text{Jedi-Ritter}, \text{Sith}, \text{Matukai}, \text{Zeison-Sha}\}\\
I(\mathit{Planet}) &= \{\text{Naboo}, \text{Alderaan}, \text{Bespin}, \text{Endor}\}\\
I(\mathit{Alt}) &= \{\text{Naboo}, \text{Alderaan}, \text{Bespin}, \text{Endor}, \text{Watto}\}\\
I(\mathit{Bewohnt}) &= \{(\text{Jedi-Ritter}, \text{Bespin}), (\text{Jedi-Ritter}, \text{Endor}), (\text{Sith}, \text{Endor}),\\
&\qquad (\text{Zeison-Sha}, \text{Endor}), (\text{Zeison-Sha}, \text{Bespin}),\\
&\qquad (\text{Matukai}, \text{Naboo}), (\text{Matukai}, \text{Alderaan})\}\\
I(\mathit{naboo}) &= \text{Naboo} \qquad I(\mathit{alderaan}) = \text{Alderaan}
\end{aligned}$$

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

c) $\forall x\,(\mathit{Alt}(x) \supset \mathit{Planet}(x))$

d) $\exists x\,(\mathit{Bewohnt}(x, \mathit{naboo}) \lor \mathit{Bewohnt}(x, \mathit{alderaan}))$

e) $\forall x\,(\mathit{Orden}(x) \lor \mathit{Planet}(x) \lor \mathit{Alt}(x))$

f) $\exists x \exists y\,(\mathit{Orden}(x) \land \mathit{Planet}(y) \land \mathit{Bewohnt}(y, x))$

Aufgabe 4 (10 Punkte) *Graphen* treten in vielen technischen Bereichen auf und müssen daher oft durch Programme verarbeitet werden. DOT ist eine Sprache, um Graphen zu spezifizieren, also um die *Knoten* und die sie verbindenden *Kanten* festzulegen ebenso wie die *Eigenschaften* (Attribute) dieser beiden Objektarten. Wir beschreiben im Folgenden eine vereinfachte Version von DOT.
Jede Beschreibung eines Graphen in DOT beginnt mit dem Schlüsselwort `graph`, dem optional eine Graphbezeichnung folgt sowie in jedem Fall eine Anweisungsliste. Graphbezeichnungen sind beliebige Folgen von Buchstaben und Ziffern, die nicht mit einer Ziffer beginnen.

Anweisungslisten sind durch geschwungene Klammern (`{`,`}`) begrenzt und bestehen aus einer Folge von Anweisungen, die durch Strichpunkte (`;`) getrennt sind. Es gibt drei Arten von Anweisungen: Knotenanweisungen, Kantenanweisungen und Teilgraphen.
Eine Knotenanweisung besteht nur aus einem Knotennamen, dem optional eine Attributliste folgen kann. Attributlisten sind durch eckige Klammern (`[`,`]`) begrenzt und bestehen aus einer Folge von Attributnamen, die durch ein einzelnes Komma (`,`) voneinander getrennt sind. Für Knoten- und Attributnamen gelten dieselben Regeln wie für Graphbezeichnungen.
Kantenanweisungen sind Folgen von mindestens zwei Knotennamen oder Teilgraphen (Knoten und Graphen können beliebig abwechseln), die jeweils entweder durch die Zeichen `--` für ungerichtete Kanten bzw. durch `->` für gerichtete Kanten getrennt sind.
Die Beschreibung von Teilgraphen sieht genauso aus wie jene von Graphen, nur dass statt `graph` das Schlüsselwort `subgraph` verwendet wird. Wird keine Graphbezeichnung nach `subgraph` angegeben, kann auch das Schlüsselwort fehlen, sodass der Teilgraph gleich mit der geschwungenen Klammer beginnt.

Beispiel:

```
graph fmod {
  a [rot, ende];
  b [blau];
  c;
  a -- {b;c} -> d [gruen];
  subgraph xxx { a -- d }
}
```

Es wird ein Graph namens `fmod` definiert, der aus drei Knotenanweisungen, einer Kantenanweisung und einem Teilgraphen besteht. Die ersten beiden Knoten- und die Kantenanweisung enthalten eine Attributliste. Die Kantenanweisung enthält den unbenannten Teilgraphen `{b;c}`. Die letzte Anweisung definiert einen Teilgraphen `xxx`, der aus einer einzigen Kantenanweisung besteht.

Sei $\mathcal{D}$ die Menge derartiger Graphbeschreibungen in Dot (ohne Berücksichtigung von Leerzeichen und Zeilenumbrüchen). Spezifizieren Sie die Sprache $\mathcal{D}$ mit Hilfe einer kontextfreien Grammatik. Verwenden Sie Ebnf-Notationen, um die Grammatik übersichtlich zu strukturieren.
Handelt es sich bei $\mathcal{D}$ um eine reguläre Sprache, d.h., lässt sich diese Sprache im Prinzip auch durch einen (komplizierten) regulären Ausdruck spezifizieren? Begründen Sie Ihre Antwort.

Aufgabe 5 (10 Punkte) In Büchern über formale Sprachen ist folgende Definition zu finden:

> Eine *monotone Grammatik* wird durch ein 4-Tupel $G = \langle V, T, S, P\rangle$ 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 eine endliche Menge von Paaren (x, y) ist, wobei $x, y \in (V \cup T)^+$ und $|x| \le |y|$ gilt.[1]
>
> 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

[1] $|x|$ bezeichnet die Anzahl der Zeichen in x. Die Bedingung $|x| \le |y|$ bedeutet also, dass y nicht kürzer sein darf als x.

als die Menge $\{\, w \in T^* \mid S \overset{*}{\Rightarrow} w \,\}$, wobei $\overset{*}{\Rightarrow}$ den reflexiven und transitiven Abschluss von $\Rightarrow$ bezeichnet.[2]

Überprüfen Sie, welche der folgenden Tupeln eine monotone Grammatik gemäß der obigen Definition darstellen. Begründen Sie Ihre Antwort, falls es sich um keine monotone Grammatik handelt. Entspricht das Tupel der Definition, geben Sie die Sprache an, die durch die Grammatik generiert wird.

a) $\langle \{\mathtt{S}\}, \{\mathtt{a}, \mathtt{b}\}, \mathtt{S}, \{\mathtt{S} \to \mathtt{ab}, \mathtt{ab} \to \mathtt{aba}\} \rangle$

b) $\langle \{\mathtt{S}, \mathtt{a}\}, \{\mathtt{b}\}, \mathtt{S}, \{\mathtt{S} \to \mathtt{ab}, \mathtt{ab} \to \mathtt{aba}\} \rangle$

c) $\langle \{\mathtt{S}\}, \{\mathtt{a}, \mathtt{b}\}, \mathtt{S}, \{\mathtt{S} \to \mathtt{a}\,\mathtt{S}\,\mathtt{b} \mid \varepsilon\} \rangle$

d) $\langle \{\mathtt{b}\}, \{\mathtt{S}, \mathtt{a}\}, \mathtt{b}, \{\mathtt{S} \to \mathtt{ab}, \mathtt{b} \to \mathtt{aba}\} \rangle$

[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.