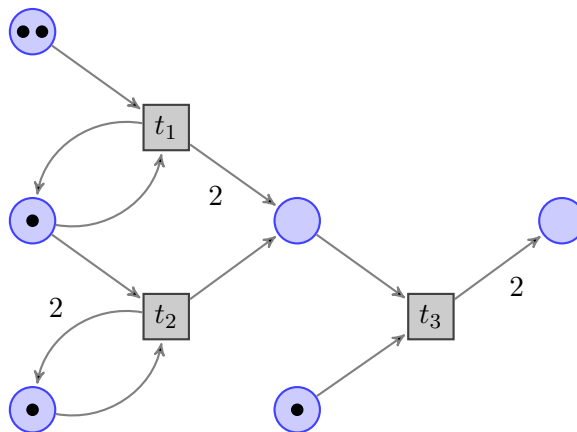


3.0/2.0 VU Formale Modellierung			
185.A06		WS 2011	14. März 2012
Matrikelnummer	Familiename	Vorname	Gruppe A

Aufgabe 1 (10 Punkte) Bestimmen Sie eine Folge von Transitionen, die nacheinander feuern können. Die Folge soll jede Transition mindestens einmal enthalten. Nach dem Feuern der letzten Transition soll keine Transition mehr aktiviert sein. Geben Sie die Transitionen dieser Folge sowie die erreichte Endmarkierung an.



Aufgabe 2 (15 Punkte) Ein elektronisches Tresorschloss besteht aus einem zweistelligen Display sowie den Tasten +, L, R, Ok und Reset. Jede Stelle des Displays kann eine der drei Ziffern 0, 1 oder 2 anzeigen. Mit jedem Drücken der +-Taste ändert sich die Anzeige der aktiven Stelle von 0 auf 1, von 1 auf 2 bzw. von 2 auf 0. Welche der beiden Stellen aktiv ist, lässt sich durch die L- und R-Taste kontrollieren: Ein- oder mehrmaliges Drücken der L- bzw. R-Taste aktiviert die linke bzw. rechte Stelle. Im Anfangszustand zeigt das Display die Zahl 00 an und die linke Stelle ist aktiviert. Wird die Zahl 21 eingestellt und anschließend die Ok-Taste gedrückt, öffnet das Schloss; bei jeder anderen Zahl geht das Schloss in einen Fehlerzustand. Sowohl im geöffneten Zustand als auch im Fehlerzustand werden alle weiteren Tasten ausgenommen Reset ignoriert, d.h., sie beeinflussen den Zustand des Schlosses nicht. Wird zu einem beliebigen Zeitpunkt die Reset-Taste gedrückt, geht das Schloss wieder in den Anfangszustand über. Das Schloss lässt sich also z.B. mit folgenden Tastenfolgen öffnen:

+ - + - R - + - Ok
+ - Reset - + - L - R - + - L - + - Ok - Ok

- Überlegen Sie, welche Informationen notwendig sind, um den Zustand des Schlosses zu beschreiben. Wieviele Zustände kann das Schloss annehmen? Wieviele Zustände sind es im Allgemeinen, wenn das Schloss n Ziffern (statt 3) pro Stelle sowie k Stellen (statt 2) besitzt?
- Legen Sie die möglichen Aktionen fest, die zu einem Zustandswechsel führen.

- Geben Sie einen endlichen Automaten an, der das Verhalten des beschriebenen Schlosses vollständig beschreibt. Der Endzustand ist erreicht, wenn das Schloss öffnet. Spezifizieren Sie die Übergangsfunktion des Automaten mittels einer Tabelle.

Aufgabe 3 (15 Punkte) DATALOG-Programme besitzen folgenden Aufbau.

- Ein *Programm* ist eine möglicherweise leere Folge von Klauseln. Eine *Klausel* ist entweder ein Faktum oder eine Regel.
- Ein *Faktum* besteht aus einer Atomformel gefolgt von einem Punkt.
- Eine *Regel* besteht aus einer Atomformel, gefolgt von den Zeichen `:-` sowie einer nicht-leeren Liste von Atomformeln, die durch Kommas (,) getrennt werden. Regeln enden ebenfalls mit einem Punkt.
- Eine *Atomformel* ist ein Name, dem optional eine in runden Klammern eingeschlossene Argumentliste folgen kann.
- Eine *Argumentliste* ist eine nicht-leere Folge von Namen und Variablen in beliebiger Reihenfolge, die voneinander durch Kommas getrennt werden.
- Ein *Name* ist eine nicht-leere Folge von Buchstaben und Ziffern, die mit einem Kleinbuchstaben beginnt.
- Eine *Variable* ist eine nicht-leere Folge von Buchstaben und Ziffern, die mit einem Großbuchstaben beginnt.

Das folgende Beispiel besteht aus zwei Fakten und drei Regeln; `adam`, `seth`, `istKindVon` usw. sind Namen, `X` und `Y` sind Variablen.

```
istKindVon(seth,adam).
istKindVon(enosh,seth).
istNachfahreVon(X,Y) :- istKindVon(X,Y).
istNachfahreVon(X,Z) :- istKindVon(X,Y), istNachfahreVon(Y,Z).
istMensch(X) :- istNachfahreVon(X,adam).
```

Beschreiben Sie die zulässigen DATALOG-Programme mittels einer kontextfreien Grammatik. Verwenden Sie so weit als möglich EBNF-Notationen, um die Grammatik übersichtlich zu halten und rekursive Regeln zu vermeiden.

Aufgabe 4 (15 Punkte) Lisa geht in ein Cafe und bestellt ein gemischtes Eis mit genau drei verschiedenen Sorten. Grundsätzlich kommen für sie nur die Sorten Mohn, Haselnuss, Erdbeere oder Schokolade in Frage.

- Lisa isst niemals Mohn und Schokolade gleichzeitig.
 - Ihr Eis soll auf jeden Fall entweder die Sorten Haselnuss und Erdbeer, oder die Sorten Haselnuss und Schokolade enthalten (aber nicht beides).
 - Immer wenn Lisa Mohn wählt, nimmt sie auch Haselnuss.
- a) Formalisieren Sie die beschriebene Situation inklusive aller Anhaltspunkte mittels aussagenlogischer Formeln. Geben Sie die Bedeutung jeder Aussagenvariablen an.
 - b) Kann sich Lisa für ein gemischtes Eis entscheiden? Wenn ja, mit welchen Sorten? Begründen Sie Ihre Antwort mit Hilfe Ihrer aussagenlogischen Modellierung.

Aufgabe 5 (15 Punkte) Formalisieren Sie die beiden nachfolgenden Sätze als prädikatenlogische Formeln. Verwenden Sie dabei die Prädikatensymbole *Verprügelt*, *Angriffslustig*, *Gallier* und *Römer* sowie die Konstantensymbole *obelix*, *lacmus* und *cäsar* mit folgender Bedeutung:

<i>Verprügelt</i> (x, y)	... x verprügelt y	<i>obelix</i>	... Obelix
<i>Angriffslustig</i> (x)	... x ist angriffslustig	<i>lacmus</i>	... Lacmus
<i>Gallier</i> (x)	... x ist ein Gallier	<i>cäsar</i>	... Cäsar
<i>Römer</i> (x)	... x ist ein Römer		

- Alle Römer werden von Obelix aber nicht von Cäsar verprügelt.
- Manche Gallier verprügeln alle angriffslustigen Römer.
- Bestimmen Sie unter Verwendung der Evaluierungsfunktion den Wahrheitswert der Formel

$$\forall x(Gallier(x) \supset Verprügelt(x, lacmus))$$

in folgender Interpretation:

$$\begin{aligned} \mathcal{U} &= \{Aerobus, Asterix, Bonus, Cäsar, Lacmus, Obelix, Troubadix, Verleihnix\} \\ I(Gallier) &= \{Asterix, Obelix, Verleihnix\} \\ I(Verprügelt) &= \{(Asterix, Aerobus), (Asterix, Lacmus), (Obelix, Aerobus), (Obelix, Bonus), \\ &\quad (Obelix, Lacmus), (Troubadix, Bonus), (Verleihnix, Lacmus)\} \\ I(lacmus) &= Lacmus \end{aligned}$$