

Name: Vorname: Matr.Nr:

Technische Universität München
 Institut für Informatik
 Prof. Manfred Broy

WS 2010/2011
 09.02.2011

Semestralklausur zu Modellierung verteilter Systeme

Allgemeine Hinweise:

- Schreiben Sie als Erstes Ihren **Namen, Vornamen** und Ihre **Matrikelnummer** in die dafür vorgesehenen Felder auf **diesem Deckblatt**. Vergessen Sie nicht auf **jedem Blatt**, das Sie bearbeitet haben, nochmals Ihre Matrikelnummer im dafür vorgesehenen Feld (oben, rechts) einzutragen.
- Die **Gesamtpunktzahl** beträgt **90 Punkte**. Zum Bestehen der Klausur sind **40%** der maximal erreichbaren Punkte ausreichend.
- Die **Bearbeitungszeit** beträgt **90 Minuten**.
- Es sind keine elektronischen Hilfsmittel zugelassen! Folien, Skriptum und Mitschrift aus Vorlesung und Übung sowie jegliche andere Literatur in Papierform sind erlaubt.
- Bei Täuschungsversuchen wird die Klausur sofort eingezogen und als "Nicht bestanden" gewertet.
- Die Klausur umfasst einschließlich dieses Deckblattes 8 Seiten. Überprüfen Sie zu Beginn der Bearbeitungszeit, ob alle Seiten vorhanden sind.

Erst nach Aufforderung umblättern!

Viel Erfolg!

Aufgabe	1	2	3	4	5	6	
Punkte	10	15	13	20	12	20	
Korrektur							Σ

Note: _____

Name: Vorname: Matr.-Nr.:

Aufgabe 1 **Multiple Choice**

(10 Punkte)

Entscheiden Sie, welche Antworten richtig sind und kreuzen Sie diese an. Jeder Aussage entspricht nur **eine** korrekte Antwort. Für jede richtige Angabe erhalten Sie 1 Punkt.

Achtung: Für jede falsche Antwort wird 1 Punkt abgezogen! Insgesamt können jedoch nicht weniger als 0 Punkte für die gesamte Aufgabe erreicht werden.

- (1) Eine Zustandsmaschine heißt *deterministisch*, wenn es von jedem Zustand einen Pfad zu einem Endzustand gibt.
 - Ja.
 - Nein.
- (2) *Temporallogische Formeln* (z.B. CTL oder LTL) spezifizieren ...
 - ...die syntaktische Schnittstelle eines Systems.
 - ...Eigenschaften des Ablaufs.
 - ...die Systemarchitektur.
- (3) Wenn ein *stabiles Prädikat* in allen Startzuständen gilt, gilt es auch für alle erreichbaren Zustände.
 - Ja.
 - Nein.
- (4) Eine *Invariante* spezifiziert ...
 - ...Lebendigkeitseigenschaften.
 - ...Eigenschaften, die in allen erreichbaren Zuständen gelten müssen.
 - ...die Menge aller internen Zustände eines Systems.
- (5) Zwei Programme heißen *disjunkt*, wenn ...
 - ...sie gemeinsame Variablen haben.
 - ...sie auf keine gemeinsamen Variablen schreibend zugreifen.
 - ...bei Zugriff auf gemeinsame Variablen gegenseitiger Ausschluss garantiert ist.
- (6) Greifen zwei Prozesse ohne *wechselseitigen Ausschluss* auf eine gemeinsame Variable zu, führt das möglicherweise zu ...
 - Verklemmung
 - Nichtdeterminismus
 - Verhungern

- (7) *Schnittstellenabstraktion* dient zum Information Hiding bezüglich ...
- ... Zuständen.
 - ... Nachrichten.
 - ... Folgen von ausgetauschten Nachrichten.
- (8) *Stromverarbeitende Funktionen* werden verwendet zur ...
- ... Spezifikation des Schnittstellenverhalten eines Systems.
 - ... Beschreibung der internen Zustände eines Systems.
- (9) Werden zwei Zustandsautomaten parallel komponiert, so kann es unerreichbare Zustände geben.
- Nein, wenn zwei Systeme syntaktisch kompatibel sind, tritt das nicht auf.
 - Nein, wenn zwei Systeme gemeinsame Kanäle haben, tritt das nicht auf.
 - Ja, bei paralleler Komposition können manche Zustandskombinationen unerreichbar sein.
- (10) Starke Kausalität führt zu ...
- ... Problemen bei Rückkopplung.
 - ... Zustandsexplosion bei paralleler Komposition.
 - ... Extra-Zeitverzögerungen bei sequentieller Komposition.
 - ... Nichtdeterminismus bei sequentieller Komposition.

Name:

Vorname:

Matr.-Nr.:

Aufgabe 2 **Modellierung von Prozessen**

(15 Punkte)

Ein Getränkeautomat bietet Wasser und Bier an. Wasser kostet 1€, Bier 2€. Der Automat erwartet erst die Auswahl des gewünschten Getränkes und danach den passenden Geldbetrag. Anschließend wird das jeweilige Getränk ausgegeben. Rückgeld oder Abbruch werden nicht modelliert.

Der Automat besitzt die Eingabekanäle:

wahl : Channel Getränk
geld : Channel Geld

sowie den Ausgabekanal:

ware : Channel Getränk

mit den Typen $Getränk = \{WASSER, BIER\}$ und $Geld = \{1€, 2€\}$

- (a) Modellieren Sie den Getränkeautomaten als CSP Prozess.
- (b) Geben Sie einen CSP Prozess für einen “durstigen” Studenten an, der immer abwechselnd Wasser und Bier an dem Automaten kauft. Achten Sie dabei darauf, dass eine parallele Komposition der beiden Prozesse nicht zu einer Verklemmung führt.

Name: Vorname: Matr.-Nr.:

Aufgabe 3 Modellierung

(13 Punkte)

Modellieren Sie einen Kühlschrank als Zustandsmaschine. Der Kühlschrank besteht aus den folgenden drei Komponenten, die folgende Zustände annehmen können:

Tür	0 (geschlossen)	1 (offen)
Licht	0 (aus)	1 (an)
Kühlung	0 (inaktiv)	1 (kühlend)

Folgende Aktionen stehen zur Verfügung, um das System zu beeinflussen:

$$A = \{Tür_auf, Tür_zu, Kühlung_an, Kühlung_aus, anschließen\}$$

Dabei gelten folgende Regeln:

- Wenn die Tür des Kühlschranks offen ist, ist das Licht an, und die Kühlung ist inaktiv.
- Wird die Tür geschlossen, ist die Kühlung an.
- Bei geschlossener Tür kann die Kühlung inaktiv oder kühlend sein, das Licht ist aus.
- Im Auslieferungszustand ist die Tür offen, die Kühlung und das Licht sind aus.
- Nach dem Anschließen des Kühlschranks ist die Tür geschlossen, und die Kühlung kühlt.

Der Kühlschrank soll als Gesamtsystem modelliert werden.

- (a) Geben Sie die Zustandsmenge Σ an.
- (b) Modellieren Sie das System als aktionsmarkierte Zustandsmaschine. Stellen Sie diese grafisch dar.
- (c) Geben Sie die Menge der erreichbaren Zustände R_∞ an.
- (d) Formulieren Sie folgende Aussagen als Prädikate auf der Zustandsmenge Σ . Entscheiden Sie, ob das Prädikat eine Invariante und/oder ein stabiles Prädikat ist.
 - i) Wenn die Tür offen ist, ist die Kühlung aus.
 - ii) Das Licht ist genau dann an, wenn die Kühlung aus und die Tür offen ist.
 - iii) Wenn das Licht an ist und die Kühlung aus, dann ist die Tür offen.

Name:

Vorname:

Matr.-Nr.:

Aufgabe 4 **Parallele Programme****(20 Punkte)**

Gegeben sei das folgende Protokoll das den gegenseitigen Ausschluss von 2 Prozessen realisiert. Dabei ist zu beachten, dass die Variablen i und $turn$ vom Typ *Bit* sind, und es gilt: $\neg i = 1$, wenn $i = 0$ und $\neg i = 0$, wenn $i = 1$.

```

proc  $P_i$  :

    do true then

        //nichtkritischerBereich

        do  $turn \neq i$  then nop od

        //kritischerBereich

         $turn := \neg i$ 

    od

```

Das folgende Programm initialisiert die Variablen und startet 2 parallele Prozessinstanzen:

```

begin
     $turn := 0$ 
    [ $P_0 \parallel P_1$ ]
end

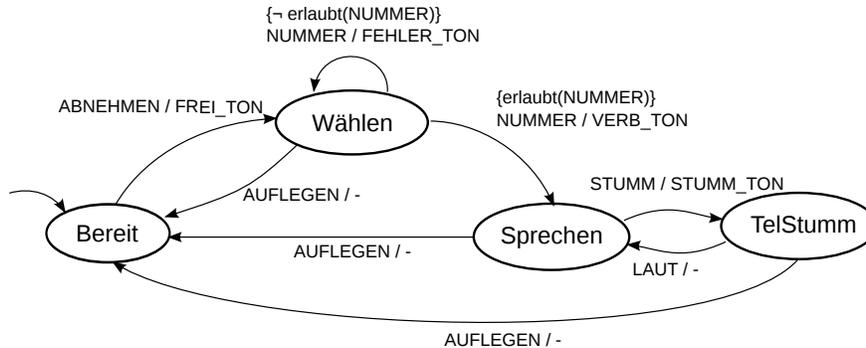
```

- Nennen Sie die gemeinsam genutzten Variablen der beiden parallelen Prozesse.
- Erweitern Sie den Prozess P_i um eine Geistervariable a_i vom Typ *Boolean*, die genau dann den Wert *true* hat, wenn sich der Prozess im kritischen Bereich befindet. Sie können davon ausgehen, dass a_i initial den Wert *false* hat.
- Finden Sie eine Invariante, die den Zusammenhang zwischen $turn$ und a_i beschreibt.
- Formalisieren Sie die Bedingung des gegenseitigen Ausschlusses mit Hilfe der Geistervariablen a_0 und a_1 . Zeigen Sie, dass das Protokoll diese Bedingung erfüllt.

Name: Vorname: Matr.-Nr.: Aufgabe 5 Schnittstellenabstraktion

(12 Punkte)

Im Folgenden ist ein stark vereinfachtes Telefon modelliert. Wenn der Hörer abgehoben wird, gibt das Telefon das Freizeichen aus und erwartet die Eingabe einer Telefonnummer. Manche Telefonnummern sind gesperrt und können nicht gewählt werden. Es wird davon ausgegangen, dass das Gespräch stets zu Stande kommt. Ein laufendes Gespräch kann stumm geschaltet werden. Zu jeder Zeit kann der Hörer wieder aufgelegt werden.



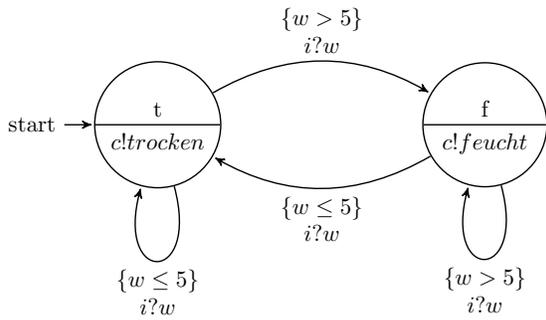
- Geben Sie die Zustandsmenge Σ , die Menge *Input* der Eingabesignale und die Menge *Output* der Ausgabesignale an.
- Wenden Sie das Verfahren zur Schnittstellenabstraktion auf den Automaten an. Spezifizieren Sie dazu für jeden Zustand eine stromverarbeitende Funktion, die das Ein-/Ausgabe-Verhalten der Maschine in diesem Zustand beschreibt, indem Sie jeweils charakteristische Gleichungen für die Funktionen angeben.

Aufgabe 6 **Komposition von Zustandsmaschinen**

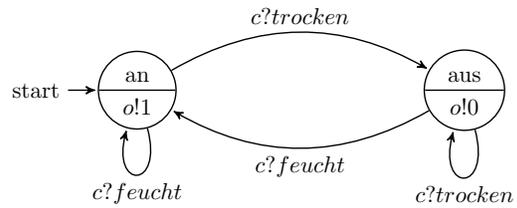
(20 Punkte)

Es seien die folgenden zwei Automaten *Sensor* und *Schalter* gegeben:

Sensor:



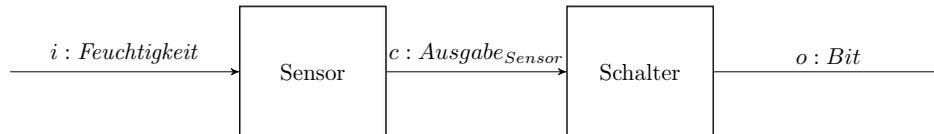
Schalter:



Dabei sind die Ein- und Ausgabemengen definiert als:

$$\begin{aligned}
 \textit{Feuchtigkeit} &= \{0 \dots 10\} \\
 \textit{Ausgabe}_{\textit{Sensor}} &= \{\textit{trocken}, \textit{feucht}\} \\
 \textit{Bit} &= \{0, 1\}
 \end{aligned}$$

Die zwei Systeme sollen wie nachstehend dargestellt komponiert werden:



Geben Sie den Automaten der parallelen Komposition als Zustandsübergangsdiagramm an.

Hinweis: Nutzen Sie bei Bedarf eine White-Box-Sicht als Zwischenschritt.