

# Aufgabenblatt 2 (Theorie)

(11 Punkte)

**Hinweis:** Abgabe (einzeln oder in Zweiergruppen) bis zum 22.10.2018 um 10:00 durch Einwurf in den Briefkasten (Erdgeschoss OH16, gegenüber von Raum E16). Eine Abgabe per E-Mail ist *nicht* möglich. Besprechung: 24-26.10.2018.

## 1 Spezifikations- und Modellerungssprachen (2 Punkte)

Nennen Sie mindestens zwei Anforderungen an Spezifikations- und Modellierungssprachen für eingebettete Systeme.

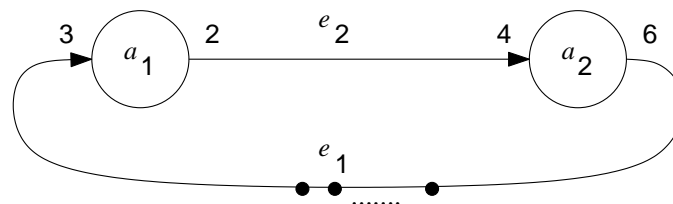
## 2 StateCharts (3 Punkte)

Mr. Smart hat eine Bierbraumaschine entwickelt, die verschiedene Konfigurationsmöglichkeiten bietet. Es ist möglich, entweder Bier (Standardeinstellung) oder Radler herzustellen. Wird Bier gebraut, so kann dieses entweder hopfig (Standardeinstellung) oder malzig schmecken. Zwischen beiden Einstellungen kann mittels einer *Geschmackstaste* gewechselt werden. Darüber hinaus lässt sich der Alkoholgehalt des Bieres auswählen, der entweder weniger als 4%, 4-5% (Standardeinstellung) oder mehr als 5% betragen kann und mittels einer *Alkoholtaste* gewechselt werden kann (Auswahl von mehr als 5% durch einmaliges, von weniger als 4% durch zweimaliges, von 4-5% durch dreimaliges Drücken etc.). Wird die Maschine aus und wieder eingeschaltet, so wird vorherige Auswahl des Alkoholgehaltes wiederhergestellt. Sowohl für Bier als auch für Radler muss die Größe der Flaschen, in die das jeweilige Getränk abgefüllt wird, festgelegt werden. Es können entweder große (Standardeinstellung) oder kleine Flaschen verwendet werden, wobei die Einstellung durch eine *Flaschentaste* verändert werden kann.

Stellen Sie die Bierbraumaschine anhand eines StateCharts dar!

## 3 SDF (2 Punkte)

Das folgende SDF-Modell sei gegeben:



Nehmen Sie an, dass die SDF-Akteure eine Zeiteinheit zur Ausführung benötigen. Erzeugen Sie einen möglichen Ausführungsschedule unter der Annahme, dass Kante  $e_1$  initial 6 Tokens besitzt und nur ein Akteur gleichzeitig aktiv sein kann (keine parallele Ausführung möglich):

Zeit	Tokens auf Kanten		Nächste Akteur-Ausführung
	$e_1$	$e_2$	
0	6	0	$a_1$ or $a_2$
1			
2			
3			
4			

Nehmen Sie nun an, dass Kante  $e_1$  initial 9 Tokens besitzt. Geben Sie ein Beispiel für parallele Ausführung, d. h., ein Beispiel, in dem beide Akteure aktiv sind. Notieren Sie die Namen der gleichzeitig aktiven Akteure in der rechten Spalte!

Zeit	Tokens auf Kanten		Akteur-Ausführung(en)
	$e_1$	$e_2$	
0	9	0	$a_1, a_2$ or ( $a_1$ and $a_2$ )
1			
2			
3			
4			

#### 4 KPN (4 Punkte)

Die Fibonacci-Zahlen sind folgendermaßen definiert: 
$$F(n) = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ F(n-2) + F(n-1) & \text{if } n > 1 \end{cases}$$

Erstellen Sie die folgenden Prozessprimitive:

- **Process Init1(input A, output B):** Sendet zu Beginn einmal den ganzzahligen Wert "1" auf seinem Outputchannel und führt danach eine unendliche Schleife aus: Lese einen Wert vom Inputchannel und schreibe ihn in den Outputchannel.
- **Process Init0(input A, output B):** Sendet zu Beginn einmal den ganzzahligen Wert "0" auf seinem Outputchannel und verhält sich danach wie Prozess Init1.
- **Process Dup(input A, output B, C):** Führt eine unendliche Schleife aus: Lese einen Wert vom Inputchannel und schreibe ihn in beide Outputchannel.
- **Process Add(input A, B, output C):** Führt eine unendliche Schleife aus: Lese einen Wert von jedem Inputchannel, addiere die beiden Werte, schreibe das Ergebnis in den Outputchannel.
- **Process Sink(input A):** Führt eine unendliche Schleife aus: Lese einen Wert pro Zyklus vom Inputchannel.

Entwickeln Sie ein Prozessnetzwerk, das ausschließlich diese Komponenten verwendet, um die Reihe der Fibonacci-zahlen zu berechnen.

**Allgemeine Hinweise:** Alle Übungstermine und weitere Informationen zur Veröffentlichung und Abgabe der Übungszettel sowie zum Erreichen der Studienleistung finden Sie unter

<https://ls12-www.cs.tu-dortmund.de/daes/de/lehre/lehrveranstaltungen/wintersemester-2018/es-1819.html>.