

Einführung in die Anwendungsorientierte Informatik - Übung 1

Robin Heinemann

4. Juni 2017

1 Aufgabe 1

notwendige Zustände:

- idle
- 1€ bezahlt
- 2€ bezahlt
- Außer Betrieb
- Münzkassette überprüfen

notwendige Ereignisse:

- 1€ eingeworfen
- 2€ eingeworfen
- Münze $\neq 1\text{€} \vee 2\text{€}$
- Abbrechen gedrückt
- Münzkassette voll
- Münzkassette nicht voll

notwendige Aktionen

- Münze auswerfen
- Fahrkarte drucken
- Meldung „Außer Betrieb“ anzeigen



2 Aufgabe 2a

Zustand \ Ereignis	Buch trifft ein	Buch wird ausgeliehen	Buch wird zurückgegeben	Timeout	schlechter Zustand	guter Zustand
Grundzustand	Katalogisieren, Timer starten 8 Wochen, \Rightarrow {ausleihbar}	%	%	%	%	%
ausleihbar	%	Timer reset, Timer starten 4 Wochen, \Rightarrow {verliehen}	%	\Rightarrow {Buch überprüfen}	%	%
Buch überprüfen	%	%	%	%	Buch aus Katalog streichen, Buch wegwerfen, \Rightarrow {Grundzustand}	Timer starten 8 Wochen, \Rightarrow {ausleihbar}
verliehen	%	Fehlermeldung "Buch verliehen" ist {verliehen}	Timer reset, \Rightarrow {ausleihbar}	erste Mahnung versenden, Timer auf 2 Wochen, \Rightarrow {erste Mahnung}	%	%
erste Mahnung	%	Fehlermeldung "Buch verliehen" ist {verliehen}	Timer reset, 5€ einfordern, \Rightarrow {ausleihbar}	zweite Mahnung versenden, Timer auf 1 Woche, \Rightarrow {zweite Mahnung}	%	%
zweite Mahnung	%	Fehlermeldung "Buch verliehen" ist {verliehen}	20€ einfordern, \Rightarrow {ausleihbar}	Buch aus Katalog streichen, Rechtsanwalt einschalten, \Rightarrow {Grundzustand}	%	%

3 Aufgabe 2b

Zustand \ Ereignis	Buch trifft ein	Buch wird ausgeliehen	Buch wird zurückgegeben	Timeout	schlechter Zustand	guter Zustand	Buch vorbestellt	wird
Grundzustand	Katalogisieren, Timer starten 8 Wochen, \Rightarrow {ausleihbar} %	%	%	%	%	%	%	
ausleihbar		Timer reset, Timer starten 4 Wochen, \Rightarrow {verliehen} %	%	\Rightarrow {Buch überprüfen} %	%	%	%	
Buch überprüfen	%	%	%	%	Buch aus Katalog streichen, Buch wegwerfen, \Rightarrow {Grundzustand} %	Timer starten 8 Wochen, \Rightarrow {ausleihbar} %	%	
verliehen	%	Fehlermeldung ist "Buch verliehen" \Rightarrow {verliehen} Fehlermeldung ist "Buch verliehen" \Rightarrow {erste Mahnung} Fehlermeldung ist "Buch verliehen" \Rightarrow {zweite Mahnung}	Timer reset, \Rightarrow {ausleihbar}	erste Mahnung versenden, Timer auf 2 Wochen, \Rightarrow {erste Mahnung} zweite Mahnung versenden, Timer auf 1 Woche, \Rightarrow {zweite Mahnung}	%	%	Buch vorbestellen, \Rightarrow {verliehen (vorbestellt)}	
erste Mahnung	%	%	Timer reset, 5€ einfordern, \Rightarrow {ausleihbar}	%	%	%	Buch vorbestellen, \Rightarrow {verliehen (vorbestellt)}	
zweite Mahnung	%	Fehlermeldung ist "Buch verliehen" \Rightarrow {zweite Mahnung}	20€ einfordern, \Rightarrow {ausleihbar}	Buch aus Katalog streichen, Rechtsanwalt einschalten, \Rightarrow {Grundzustand}	%	%	Buch vorbestellen, \Rightarrow {verliehen (vorbestellt)}	
verliehen (vorbestellt)	%	Fehlermeldung ist "Buch verliehen" \Rightarrow {verliehen (vorbestellt)}	Timer 4 Wochen, Vorbesteller informieren, \Rightarrow {verliehen}	erste Mahnung versenden, Timer auf 2 Wochen, \Rightarrow {erste Mahnung (verliehen)}	%	%	%	
erste Mahnung (vorbestellt)	%	Fehlermeldung ist "Buch verliehen" \Rightarrow {erste Mahnung (vorbestellt)}	Timer 4 Wochen, Vorbesteller informieren, 5€ einfordern, \Rightarrow {verliehen}	zweite Mahnung versenden, Timer auf 1 Woche, \Rightarrow {zweite Mahnung (verliehen)}	%	%	%	
zweite Mahnung (vorbestellt)	%	Fehlermeldung ist "Buch verliehen" \Rightarrow {zweite Mahnung (vorbestellt)}	Timer 4 Wochen, Vorbesteller informieren, 20€ einfordern, \Rightarrow {verliehen}	Buch aus Katalog streichen, Rechtsanwalt einschalten, \Rightarrow {Grundzustand}	%	%	%	

3.1 Warteliste

Ein endlicher Automat kann keine beliebig große Warteliste implementieren, denn für jeden Eintrag wäre (mindestens) ein weiterer Zustand nötig. Außerdem gäbe es Probleme einer Position in der Warteliste einen Namen zuzuordnen und somit den richtigen Vorbesteller zu informieren.

4 Aufgabe 3

4.1 Teilaufgabe a

4.1.1 $A \rightarrow B$

Zustand \ Ereignis	Fahrzeug bei C	C frei
heranfahren	\Rightarrow {warten}	\Rightarrow {weiterfahren}
warten	\Rightarrow {warten}	\Rightarrow {weiterfahren}
weiterfahren	%	%

4.2 Teilaufgabe b

4.2.1 $B \rightarrow A$

Zustand \ Ereignis	Fahrzeug bei C	C frei
heranfahren	\Rightarrow {weiterfahren}	\Rightarrow {weiterfahren}
weiterfahren	%	%

4.2.2 $A \rightarrow C$

Zustand \ Ereignis	Fahrzeug bei C	C frei
heranfahren	\Rightarrow {weiterfahren}	\Rightarrow {weiterfahren}
weiterfahren	%	%

4.2.3 $C \rightarrow A$

Zustand \ Ereignis	Fahrzeug bei B	C frei
heranfahren	\Rightarrow {warten}	\Rightarrow {weiterfahren}
warten	\Rightarrow {warten}	\Rightarrow {weiterfahren}
weiterfahren	%	%

4.2.4 $B \rightarrow C$

Zustand \ Ereignis	Fahrzeug bei A	A frei
heranfahren	\Rightarrow {warten}	\Rightarrow {weiterfahren}
warten	\Rightarrow {warten}	\Rightarrow {weiterfahren}
weiterfahren	%	%

4.2.5 $C \rightarrow B$

Zustand \ Ereignis	Fahrzeug bei B	B frei
heranfahren	\Rightarrow {warten}	\Rightarrow {weiterfahren}
weiterfahren	%	%

4.3 Teilaufgabe c

Ein *deadlock* würde zum Beispiel bei folgenden Bedingungen entstehen:

- $C \rightarrow A$
- $A \rightarrow B$
- $B \rightarrow C$

Die könnte im echten Straßenverkehr entstehen, wenn gleichzeitig drei Fahrzeuge, jeweils eins von jeder Seite der Kreuzung, an die Kreuzung herankommen und entsprechend der Bedienungen abbiegen.