

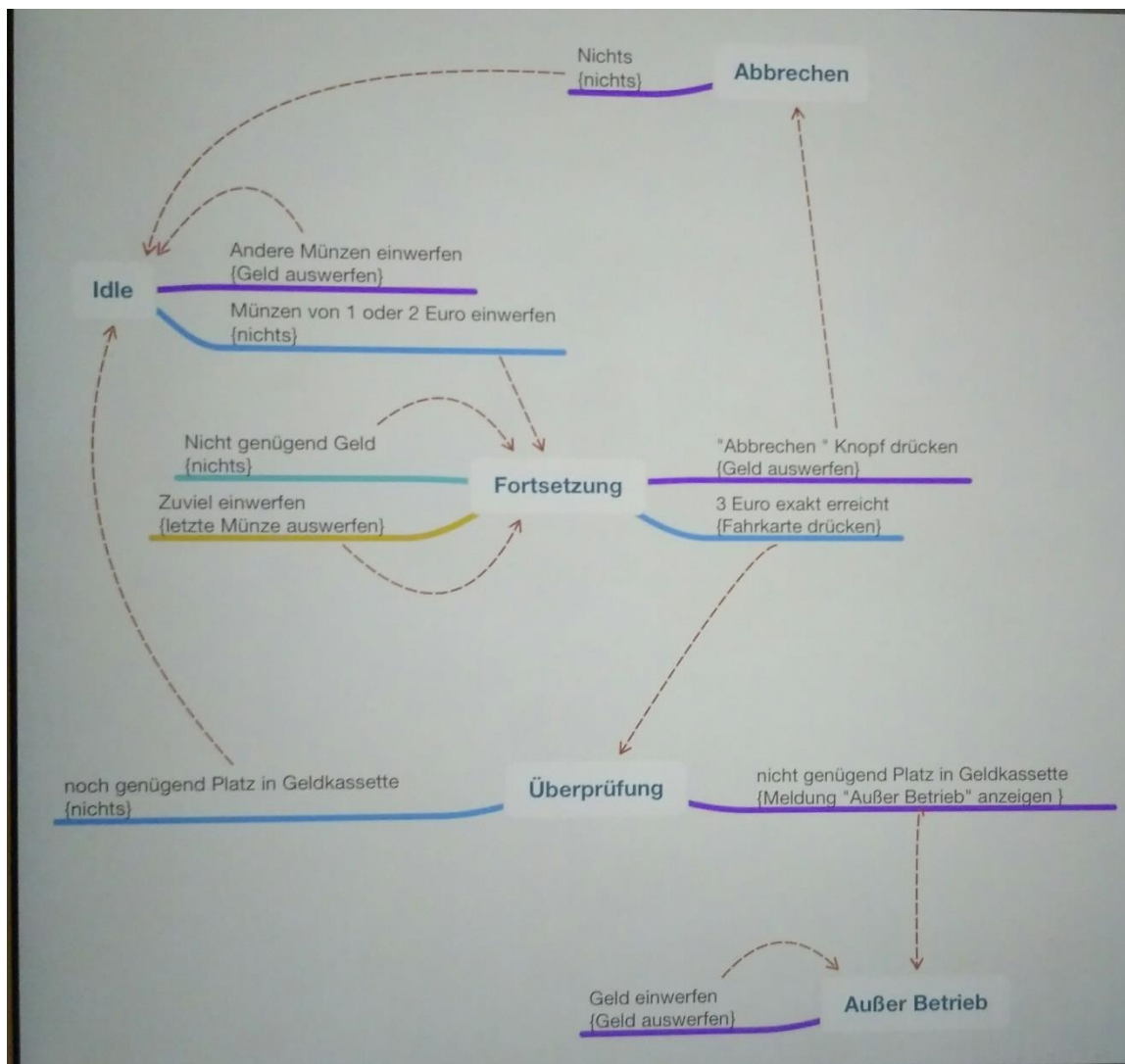
Aufgabe 1.1

Ereignisse: Münzen von 1 oder 2 Euro einwerfen, Andere Münzen einwerfen, "Abbrechen" Knopf drücken, 3 Euro exakt erreicht, nicht genügend Geld, Zuviel einwerfen, noch genügend Platz in Geldkassette, nicht genügend Platz in Geldkassette

Zustände: Idle, Fortsetzung, Überprüfung, Abbrechen, Außer Betrieb

Aktion: Geld auswerfen, Fahrkarte drücken, Meldung "Außer Betrieb" anzeigen

Warum kann ein endlicher Automat keine Warteliste von Vorbestellungen unterstützen?
Weil sie unendlich lang sein könnte und weil er sich nicht merken kann, wer zuerst vorbestellt hat.



Aufgabe 1.2

a)

| | | | | | | |
|------------------|-------------------------------------|--|---|--|---|--|
| | neu gekaufte Buch katalogisieren | Buch ausleihen | Überprüfung | (Timer 1 zur Leihfrist) Time out | (Timer 2 zur ersten Mahnung) Time out | (Timer 3 zur zweiten Mahnung) Time out |
| Grundzustand | => ausleihbar {nichts} | | | | | |
| Ausleihbar | | => nicht ausleihbar {Timer 1 startet} | Falls ein Buch in schlechten Zustand ist => verliehen {Buch auswerfen, aus dem Katalog austragen} | | | |
| | | Falls nicht so => Grundzustand {nichts} | | Falls Buch zurückgeben => ausleihbar {Timer 1 stoppt} | Falls Buch zurückgeben => ausleihbar {Timer 2 stoppt, Verzugsgebühr von 5 Euro fordern} | Falls Buch zurückgeben => ausleihbar {Timer 3 stoppt, Verzugsgebühr von 20 Euro fordern} |
| Nicht ausleihbar | | | | Falls Buch nicht zurückgeben => nicht ausleihbar {Timer 1 stoppt, Timer 2 startet, Erste Mahnung senden} | Falls Buch nicht zurückgeben => nicht ausleihbar {Timer 2 stoppt, Timer 3 startet, Zweite Mahnung senden} | Falls Buch nicht zurückgeben => verliehen => Grundzustand {Timer 3 stoppt, das Buch aus dem Katalog gestrichen} |
| verliehen | | => Grundzustand {Fehlermeldung "Buch ist verliehen" senden} | | | | |

| | | | | | | | |
|------------------|-------------------------------------|--|---|----------------------------|--|---|---|
| | neu gekaufte Buch katalogisieren | Buch ausleihen | Überprüfung | Buch vorbestellen | (Timer 1 zur Leihfrist) Time out | (Timer 2 zur ersten Mahnung) Time out | (Timer 3 zur zweiten Mahnung) Time out |
| Grundzustand | => ausleihbar {nichts} | | | | | | |
| Ausleihbar | | => nicht ausleihbar {Timer 1 startet} | Falls ein Buch in schlechten Zustand ist => verliehen {Buch auswerfen, aus dem Katalog austragen} Falls nicht so => Grundzustand {nichts} | | | | |
| Nicht ausleihbar | | | | => vorbestellt {nichts} | Falls Buch zurückgeben => ausleihbar {Timer 1 stoppt} | Falls Buch zurückgeben => ausleihbar {Timer 2 stoppt, Verzugsgebühr von 5 Euro fordern} | Falls Buch zurückgeben => ausleihbar {Timer 3 stoppt, Verzugsgebühr von 20 Euro fordern} |
| | | | | | Falls Buch nicht zurückgeben => nicht ausleihbar {Timer 1 stoppt, Timer 2 startet, Erste Mahnung senden} | Falls Buch nicht zurückgeben => nicht ausleihbar {Timer 2 stoppt, Timer 3 startet, Zweite Mahnung senden} | Falls Buch nicht zurückgeben => verliehen => Grundzustand {Timer 3 stoppt, das Buch aus dem Katalog gestrichen} |
| verliehen | | => Grundzustand {Fehlermeldung "Buch ist verliehen" senden} | | | | | |
| vorbestellt | | | | | Falls Buch zurückgeben => verliehen {Vorbesteller informieren, Buch aus dem Katalog austragen, Timer 1 startet} | Falls Buch zurückgeben => verliehen {Vorbesteller informieren, Buch aus dem Katalog austragen, Timer 1 startet} | Falls Buch zurückgeben => verliehen {Vorbesteller informieren, Buch aus dem Katalog austragen, Timer 1 startet} |

b)

Aufgabe 1.3

a. Kommt von A, will nach B

| | | |
|---------------------------------|--|-----------------------------|
| | Fahrzeug bei C | C ist frei |
| (an die Kreuzung) heranfahen | => warten {Fahrzeug bei C weiterfahren} | => weiterfahren {nichts} |
| warten | => warten {nichts} | => weiterfahren {nichts} |
| weiterfahren | | |

b. Kommt von A, will nach C

| | | |
|---------------------------------|--|-----------------------------|
| | Ein Fahrradfahrer geradeaus durchfahren will | Kein Fahrradfahrer |
| (an die Kreuzung) heranfahen | => warten {Fahrrad weiterfahren} | => weiterfahren {nichts} |
| warten | => warten {nichts} | => weiterfahren {nichts} |
| weiterfahren | | |

Kommt von C, will nach B

| | | |
|---------------------------------|--|-----------------------------|
| | Ein Fahrradfahrer geradeaus durchfahren will | Kein Fahrradfahrer |
| (an die Kreuzung) heranfahen | => warten {Fahrrad weiterfahren} | => weiterfahren {nichts} |
| warten | => warten {nichts} | => weiterfahren {nichts} |
| weiterfahren | | |

Kommt von C, will nach A

| | | |
|---------------------------------|--|-----------------------------|
| | "Fahrzeug bei A, das nach B will" oder/und "Fahrzeug bei B, das nach A will" | A und B ist frei |
| (an die Kreuzung) heranfahen | => warten {Fahrzeug bei A oder/und B weiterfahren} | => weiterfahren {nichts} |
| warten | => warten {nichts} | => weiterfahren {nichts} |
| weiterfahren | | |

Kommt von B, will nach A

| | |
|---------------------------------|-----------------------------|
| | |
| (an die Kreuzung) heranfahen | => weiterfahren {nichts} |
| warten | |
| weiterfahren | |

Kommt von B, will nach C

| | Fahrzeug bei A | A ist frei |
|----------------------------------|--|-----------------------------|
| (an die Kreuzung) heranfahren | => warten {Fahrzeug bei A weiterfahren} | => weiterfahren {nichts} |
| warten | => warten {nichts} | => weiterfahren {nichts} |
| weiterfahren | | |

c. Fahrzeug bei A, das nach B will, Fahrzeug bei B, das nach C will, und Fahrzeug bei C, das nach A will

