Informatik II: Modellierung

Prof. Dr. Martin Glinz

Kapitel 4

Verhaltensmodelle

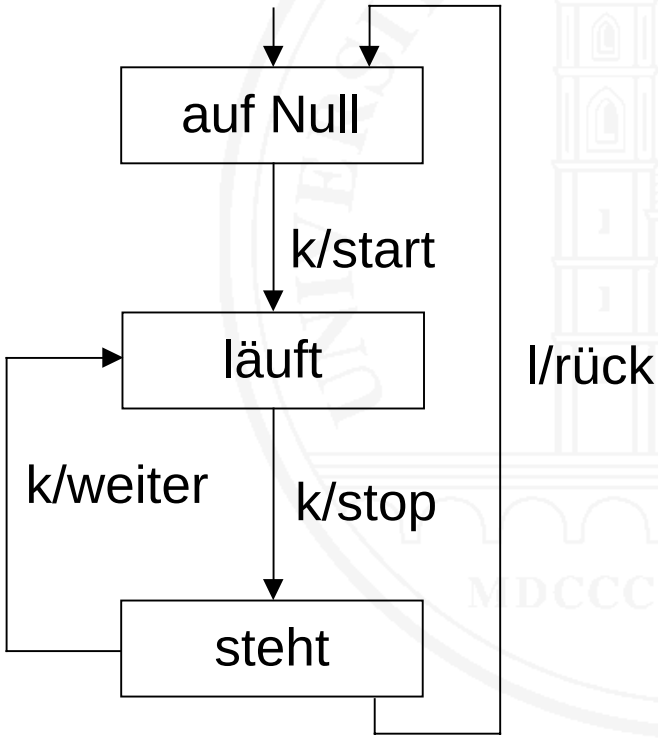
Universität Zürich
Institut für Informatik

4.1 Grundlagen

- Modellierung des zeitlich/dynamischen Verhaltens eines Systems, insbesondere für
  - Dialoge
  - Parallele und verteilte Systeme
  - Echtzeit-Systeme
  - „Eingebettete“ Informationssysteme
- Interessierende Fragestellungen:
  - Wann geschieht etwas?
  - Wie lange geschieht etwas bzw. in welchem Zeitraum kann / muss etwas stattfinden?
  - Welche Funktionen stehen wann zur Verfügung?
  - Wie koordinieren parallele und verteilte Komponenten den zeitlichen Ablauf ihrer Tätigkeiten?

4.2 Zustandsautomaten

Eine einfache Stoppuhr mit einem Bedienknopf:



Ereignisse:

k	Bedienknopf kurz gedrückt
l	Bedienknopf lang gedrückt

Aktionen:

start	Stoppuhr starten
stop	Stoppuhr anhalten
weiter	Stoppuhr weiterlaufen lassen
rück	Stoppuhr auf Null stellen

Zustandsautomaten: Definition, Grundlagen

- Ein Zustandsautomat besteht aus
  - einer endlichen Menge Z von Zuständen
  - einer Menge U von Zustandsübergängen zwischen Zuständen aus Z
  - einem oder mehreren Ereignissen zu jedem Zustandsübergang aus U, die den Zustandsübergang auslösen
  - Null bis n Aktionen zu jedem Zustandsübergang aus U, die durch den Zustandsübergang ausgelöst werden
  - der Kennzeichnung genau eines Zustands aus Z als Startzustand
  - optional der Kennzeichnung von Endzuständen
- Formale Grundlage: Theorie der Endlichen Automaten

Zustandsautomaten: Definitionen – 2

- **Zustandsautomat** (state automaton) – Endliche Menge von Zuständen, Zustandsübergängen, Ereignissen und Aktionen.
- **Zustand** (state) – repräsentiert einen genau abgegrenzten *Zeitabschnitt*, in dem das System ein bestimmtes Verhalten zeigt und auf eine bestimmte Menge von Eingaben reagiert.
- **Zustandsübergang** (state transition) – Zeitpunkt, an dem ein System einen gegebenen Zustand verlässt und in einen neuen Zustand übergeht
- **Ereignis** (event) – Zeitpunkt, an dem eine Handlung wirksam wird oder eine Veränderung festgestellt wird.
- **Aktion** (action) – eine Operation, die bei einem Zustandsübergang angestoßen wird oder ein Ereignis, das bei einem Zustandsübergang erzeugt wird.

Interpretation

- Ausgehend vom gegebenen Startzustand befindet sich ein Zustandsautomat immer in genau einem Zustand.
- Im Zustand Z reagiert der Automat auf alle Ereignisse, die Zustandsübergänge von Z in einen anderen Zustand auslösen. Alle übrigen Ereignisse werden ignoriert.
- Tritt ein Ereignis ein, das einen Zustandsübergang auslöst, werden die zugehörigen Aktionen angestoßen und der Automat geht in den durch den Zustandsübergang spezifizierten Folgezustand.
- Zustandsübergänge sind zeitfrei.

Notation

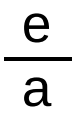
- Zustandsautomaten können tabellarisch oder graphisch dargestellt werden durch
  - Zustandsmatrizen
  - Zustandsdiagramme
- In der Regel werden Zustandsdiagramme verwendet

Zustandsdiagramm (state diagram) – Graphische Darstellung eines Zustandsautomaten.

Notation – 2

Zustand (state)

Zustandsübergang (state transition)



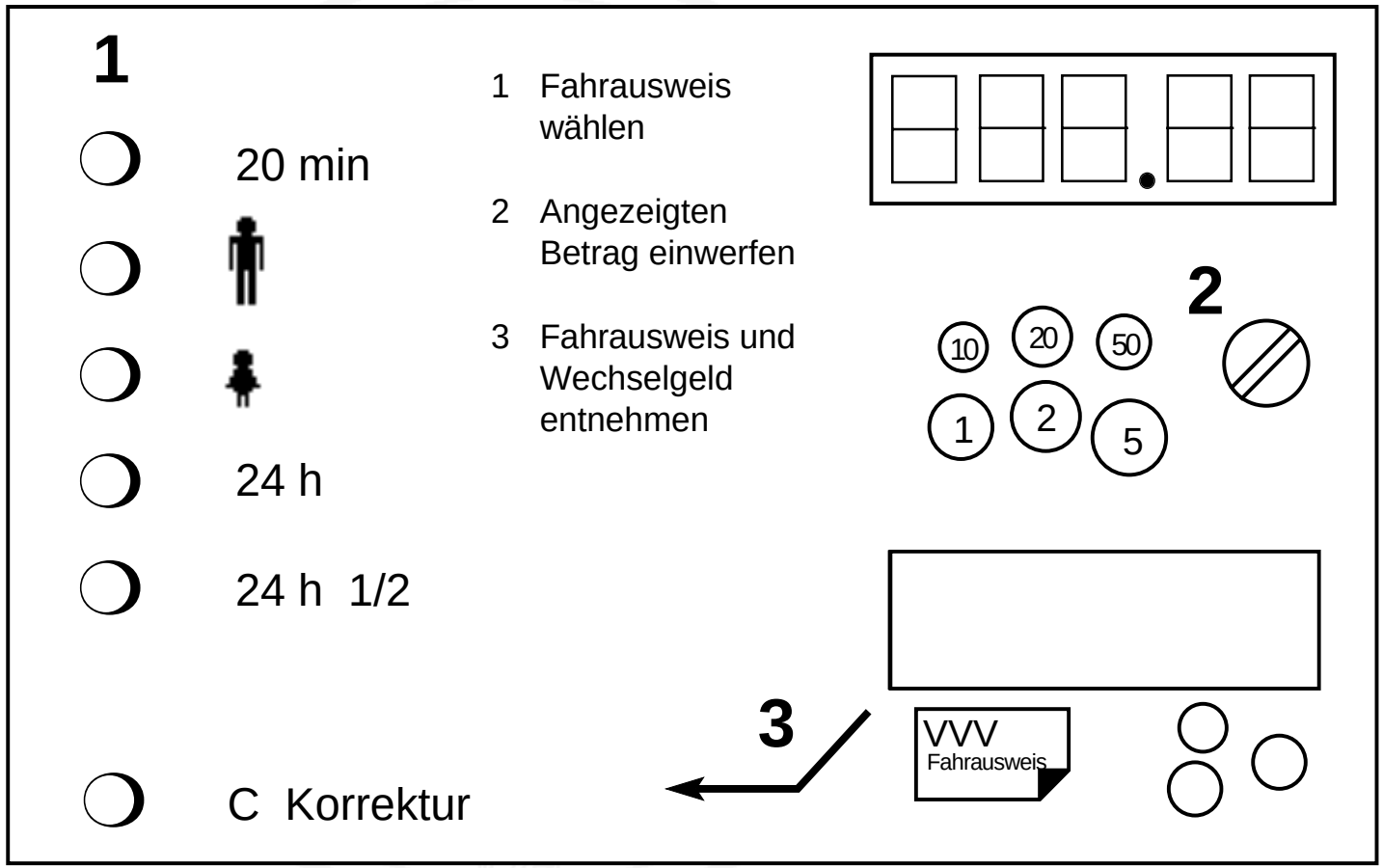
e/a

Auslösendes Ereignis / ausgelöste Aktion (event / action)

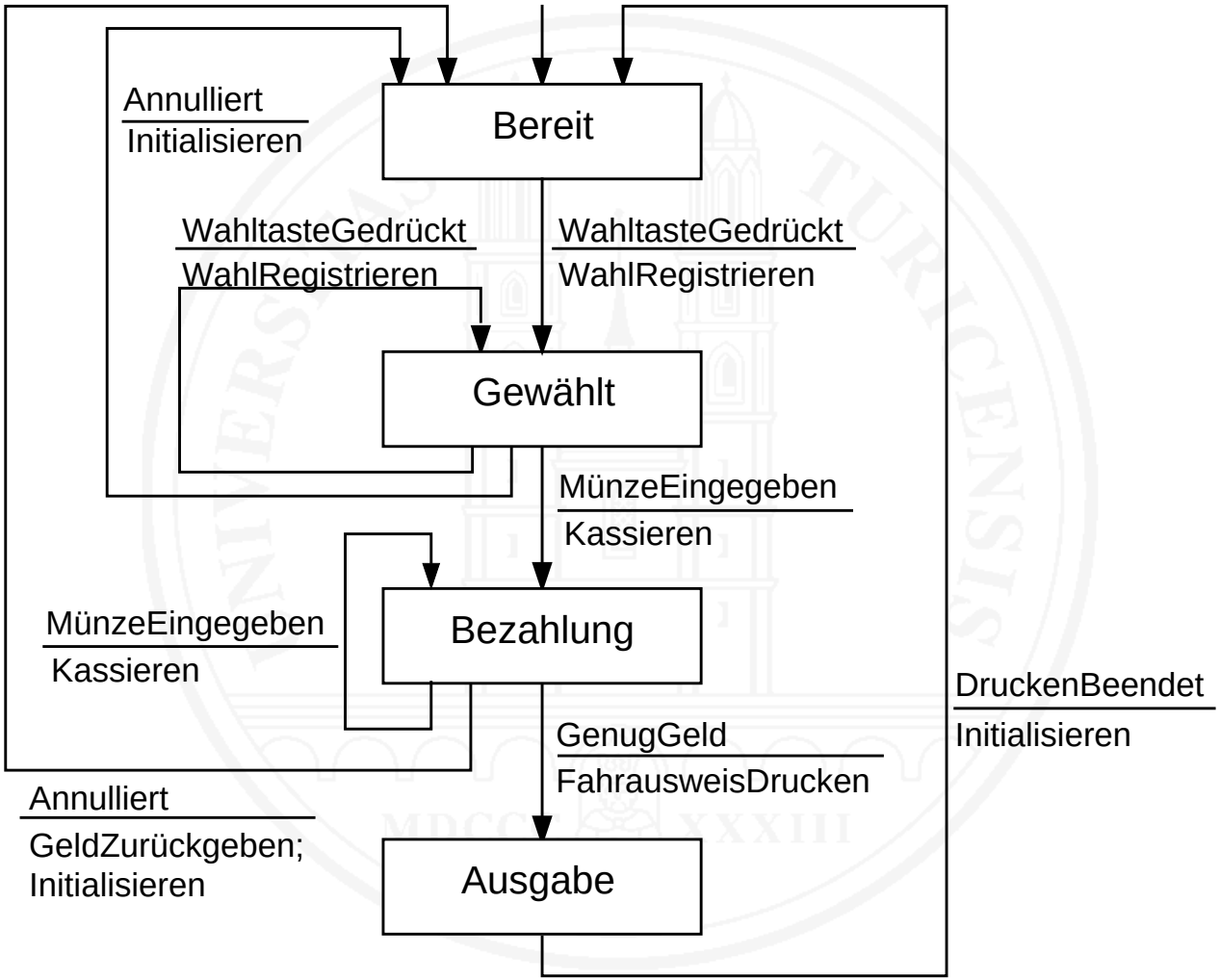
Startzustand (initial state)

Beispiel: Ein einfacher Fahrausweis-Automat

Bedienung:



Beispiel – 2: Verhaltensmodell



Aufgabe 4.1

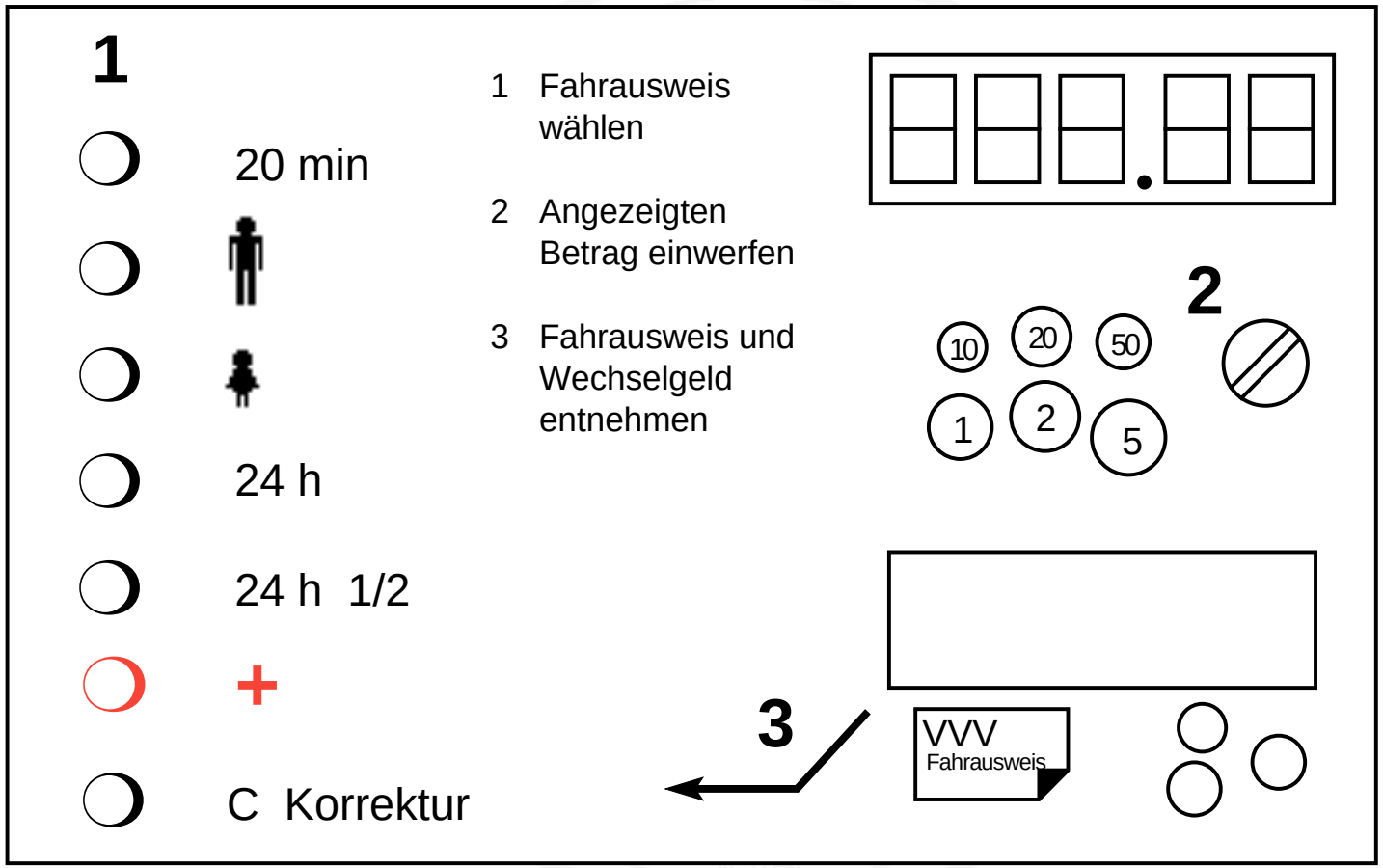
Der oben beschriebene Fahrausweis-Automat soll wie folgt erweitert werden:

Es gibt eine zusätzliche Taste, die PLUS-Taste, welche es ermöglicht, mehrere Auswahlen hintereinander zu treffen und gesammelt zu bezahlen. Nach dem Drücken einer Wahltaste kann also entweder bezahlt oder die PLUS-Taste gedrückt werden. Im letzteren Fall muss anschließend eine weitere Wahltaste gedrückt werden.

Wie muss das oben gegebene Zustandsdiagramm ergänzt/geändert werden, wenn diese Erweiterung modelliert werden soll?

Wo stoßen Sie beim Modellieren auf Unklarheiten in der Aufgaben-stellung?

Aufgabe 4.1: Neues Bedienfeld



Vor- und Nachteile von Zustandsautomaten

- Anschaulich
- Symbolisch ausführbar

- Ignoriert Daten und Funktionalität
- Zustandsexplosion bei Modellen mit vielen parallelen Abläufen
- keine Mittel zur Dekomposition großer Modelle

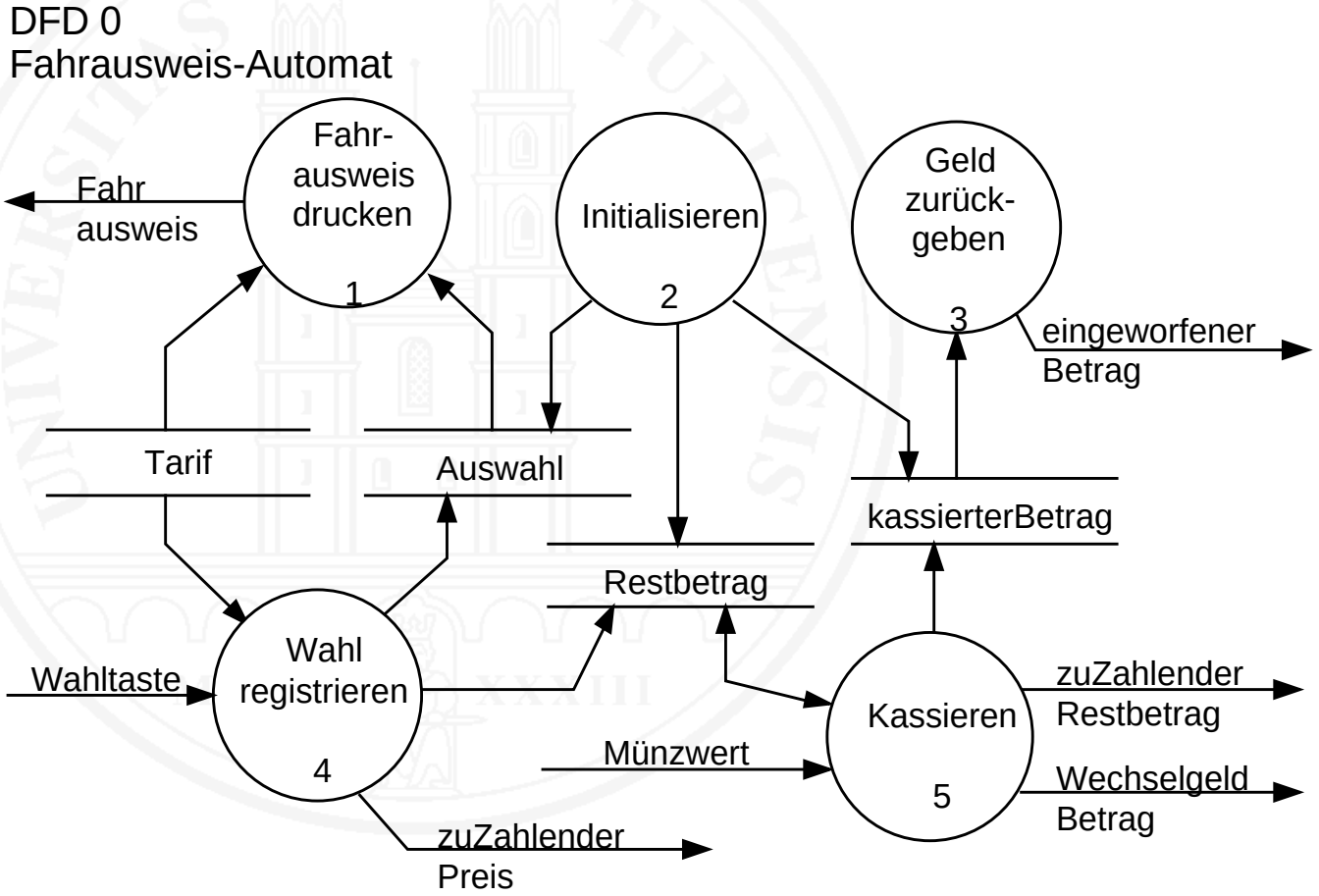
- Die Probleme von Zustandsexplosion und fehlender Dekomposition werden durch Statecharts (siehe nächster Abschnitt) gelöst

Modellierung zusätzlicher Aspekte

- Zustandsautomaten ignorieren Daten und Funktionalität
- Kombination von Verhaltensmodellen mit anderen Modellen, zum Beispiel
  - Klassenmodelle (vgl. Kapitel 5)
  - Datenflussdiagramme

Beispiel: Funktionalität des Fahrausweis-Automaten

Ergänzung des Verhaltensmodells des Fahrausweis-Automaten durch ein Datenflussmodell:



Methodik der Erstellung von Zustandsautomaten

0. Problemstellung analysieren.
1. Startzustand Z_0 identifizieren und benennen.
2. Sei Z_a der aktuelle Zustand (zu Beginn ist dies der Startzustand). Alle Arten von Ereignissen, auf die im Zustand Z_a reagiert werden soll, identifizieren und die zugehörigen auszuführenden Aktionen bestimmen.
3. Für jede Ereignisart wird ein Zustandsübergang $ZÜ_i$ von Z_a in einen potenziellen Folgezustand Z_i modelliert.
4. Für jeden potenziellen Folgezustand Z_i:
   4a. Alle Arten von Ereignissen, auf die in Z_i reagiert werden soll, identifizieren und die zugehörigen auszuführenden Aktionen bestimmen.

Methodik der Erstellung von Zustandsautomaten – 2

4b. Sind diese Ereignisarten und Aktionen für zwei potenzielle Folgezustände Z_i und Z_k gleich, so führen die Zustandsübergänge $ZÜ_i$ und $ZÜ_k$ in einen gemeinsamen Folgezustand.

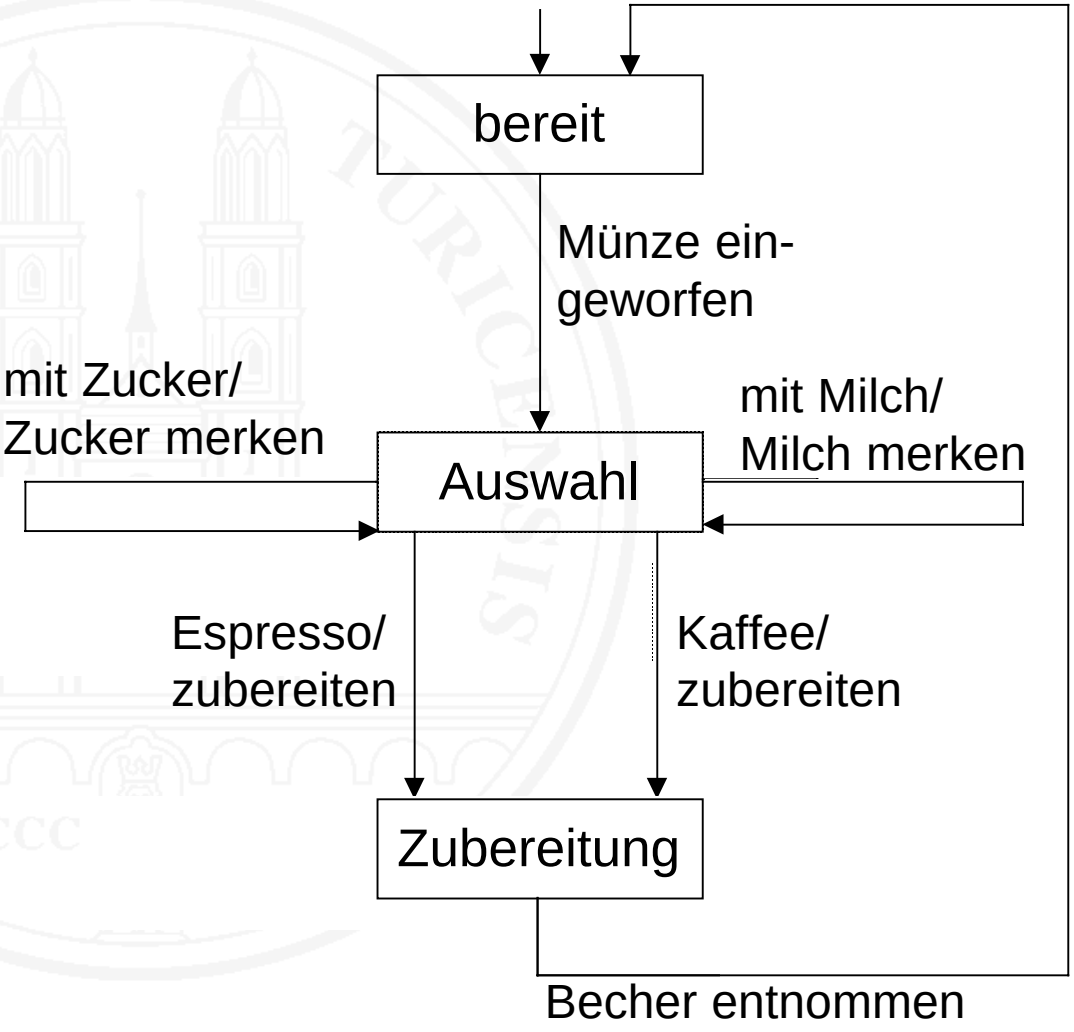
4c. Sind die Ereignisarten und Aktionen, auf die in Z_i reagiert wird, gleich wie diejenigen, auf die in Z_a reagiert wird, so ist $Z_i = Z_a$ und $ZÜ_i$ ist ein Zustandsübergang vom aktuellen Zustand Z_a auf sich selbst.

4d. Die auf diese Weise verbleibenden, voneinander verschiedenen Folgezustände benennen.

5. Schritte 2, 3 und 4 rekursiv für jeden neu modellierten Zustand wiederholen.

Aufgabe 4.2

Modellieren Sie einen Kaffeeautomaten, der wie folgt funktioniert:
Als erstes wird eine Wertmünze eingeworfen.
Als nächstes können Zusätze gewählt werden: mit Milch, mit Zucker.
Als drittes wird die Art des Kaffees gewählt: Normal oder Espresso.
Dann wird das gewählte Getränk zubereitet.
Sobald der gefüllte Becher entnommen ist, ist das Gerät wieder bereit.



4.3 Statecharts

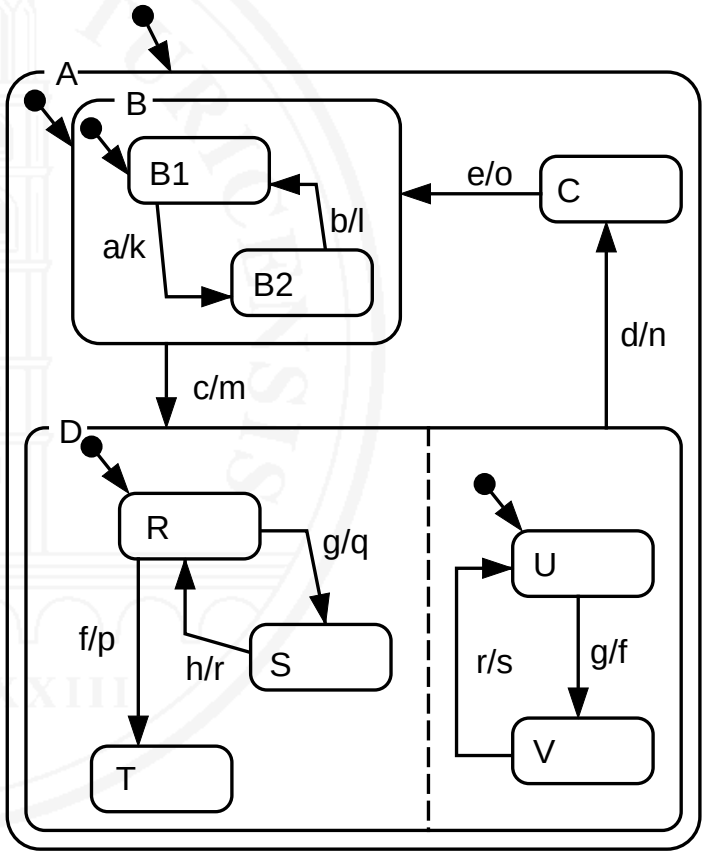
Statechart – Ein Zustandsdiagramm, dessen Zustände hierarchisch und parallel zerlegbar sind.

- Statecharts =

  Zustandsdiagramme

  + Hierarchische Zerlegung
  + Parallelität

  (Harel 1987)



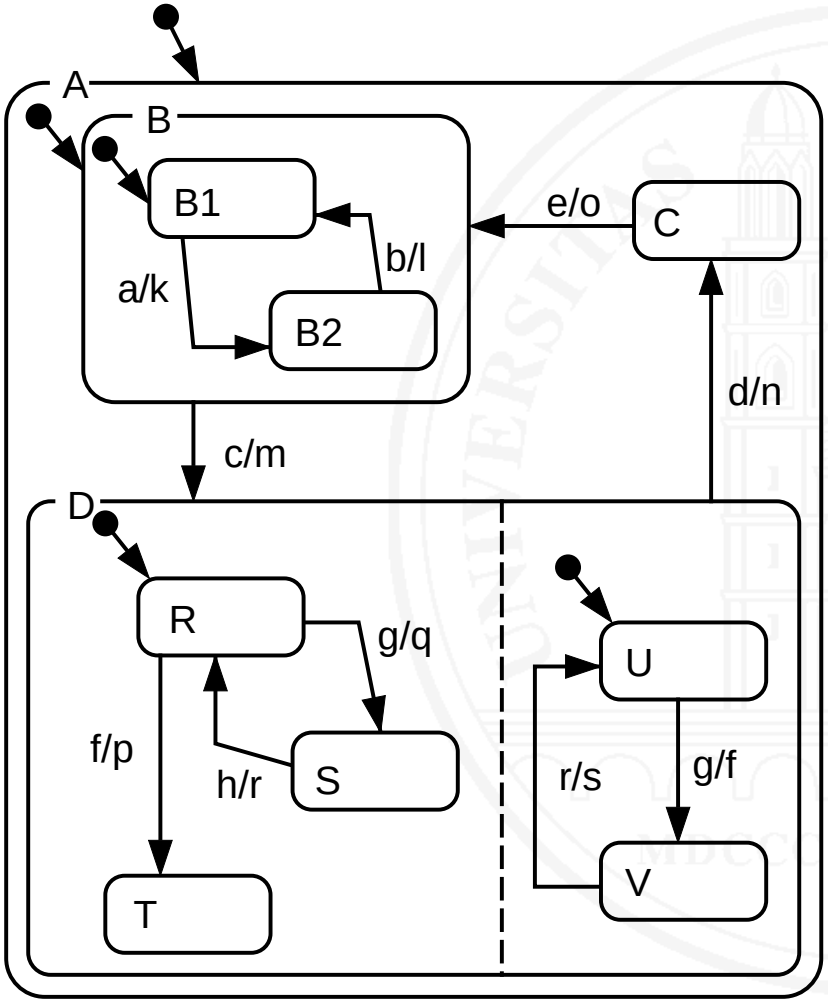
Statecharts – 2

Jeder Zustand in einem Statechart ist

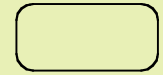
- entweder elementar
- oder hierarchisch zerlegt durch ein anderes Statechart
- oder parallel zerlegt in mehrere parallele Statecharts

- Vermeiden die Probleme von Zustandsautomaten
- Theoretisch äquivalent mit Zustandsautomaten

Notation und Interpretation



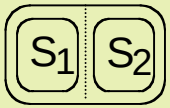
Notation:



Zustand oder Statechart

Zustandsübergang

Initialzustand

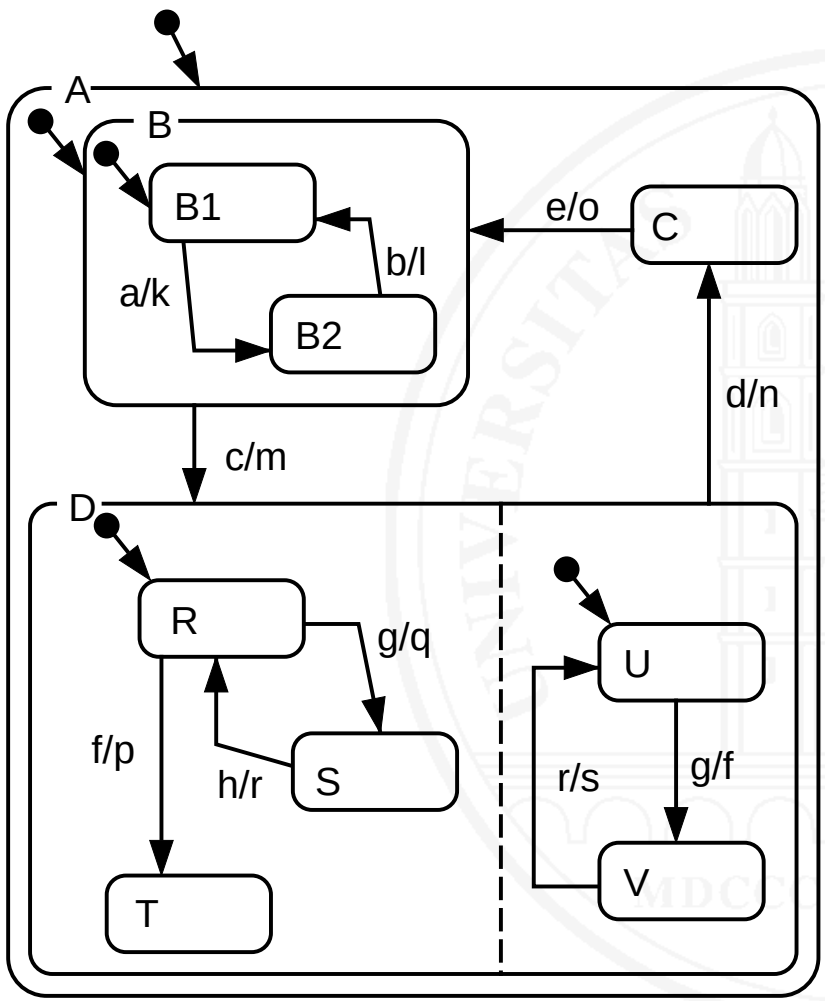


Zustand besteht aus parallelen Statecharts

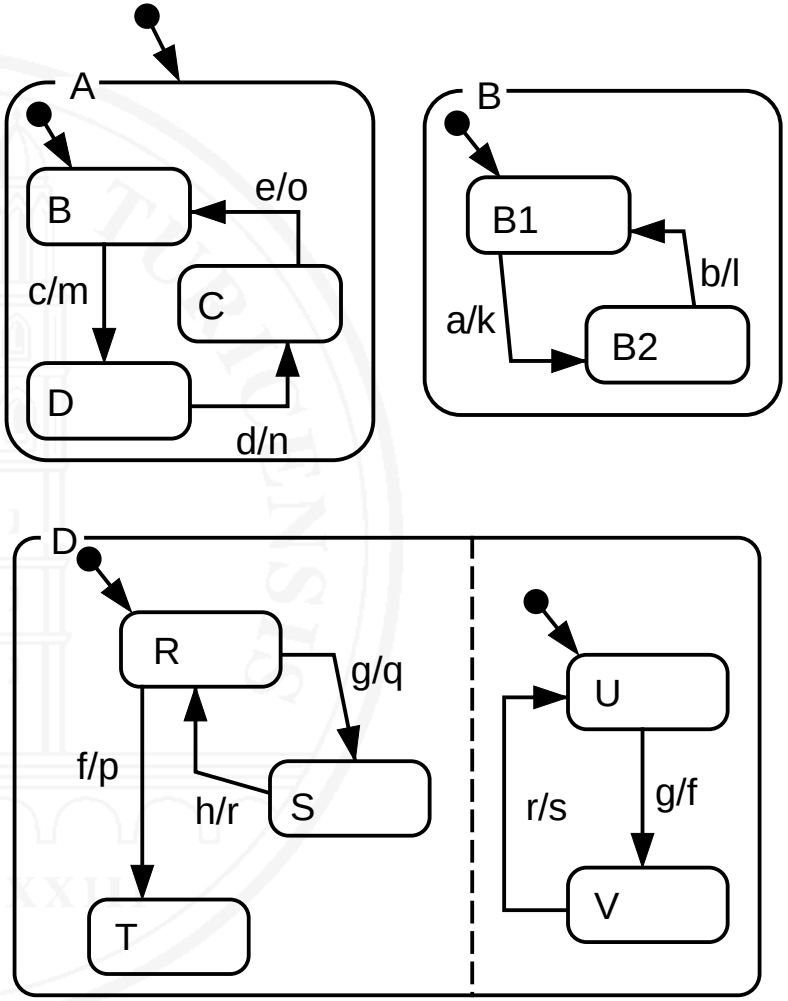
e/a

auslösendes Ereignis e und ausgelöste Aktion a

Verschachtelte oder flache Darstellung möglich



verschachtelt



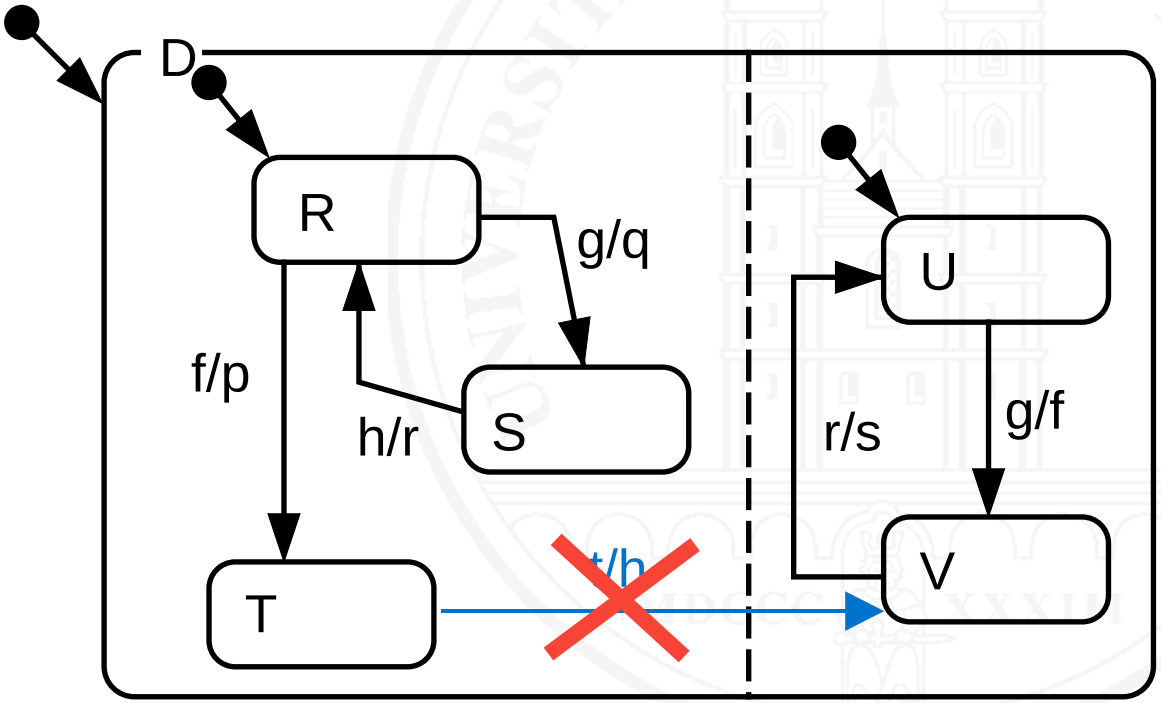
flach

Interpretation hierarchischer Statecharts (Details)

- Zustandsübergang in einen elementaren Zustand:
  wie bei Zustandsautomaten
- Verlassen eines elementaren Zustands:
  wie bei Zustandsautomaten
- Zustandsübergang in einen Zustand, der aus einem Statechart besteht:
  dieses Statechart geht in den Initialzustand; bei mehrstufiger Hierarchie gilt diese Regel rekursiv
- Verlassen eines Zustands, der aus einem Statechart besteht:
  alle Zustände dieses und aller darin hierarchisch enthaltenen Statecharts werden verlassen

Interpretation paralleler Statecharts

- Bei der Initialisierung geht dieses Statechart in den Zustand (R,U)
- Die Ereignis-Sequenz g h f



führt zur folgenden Zustandssequenz:

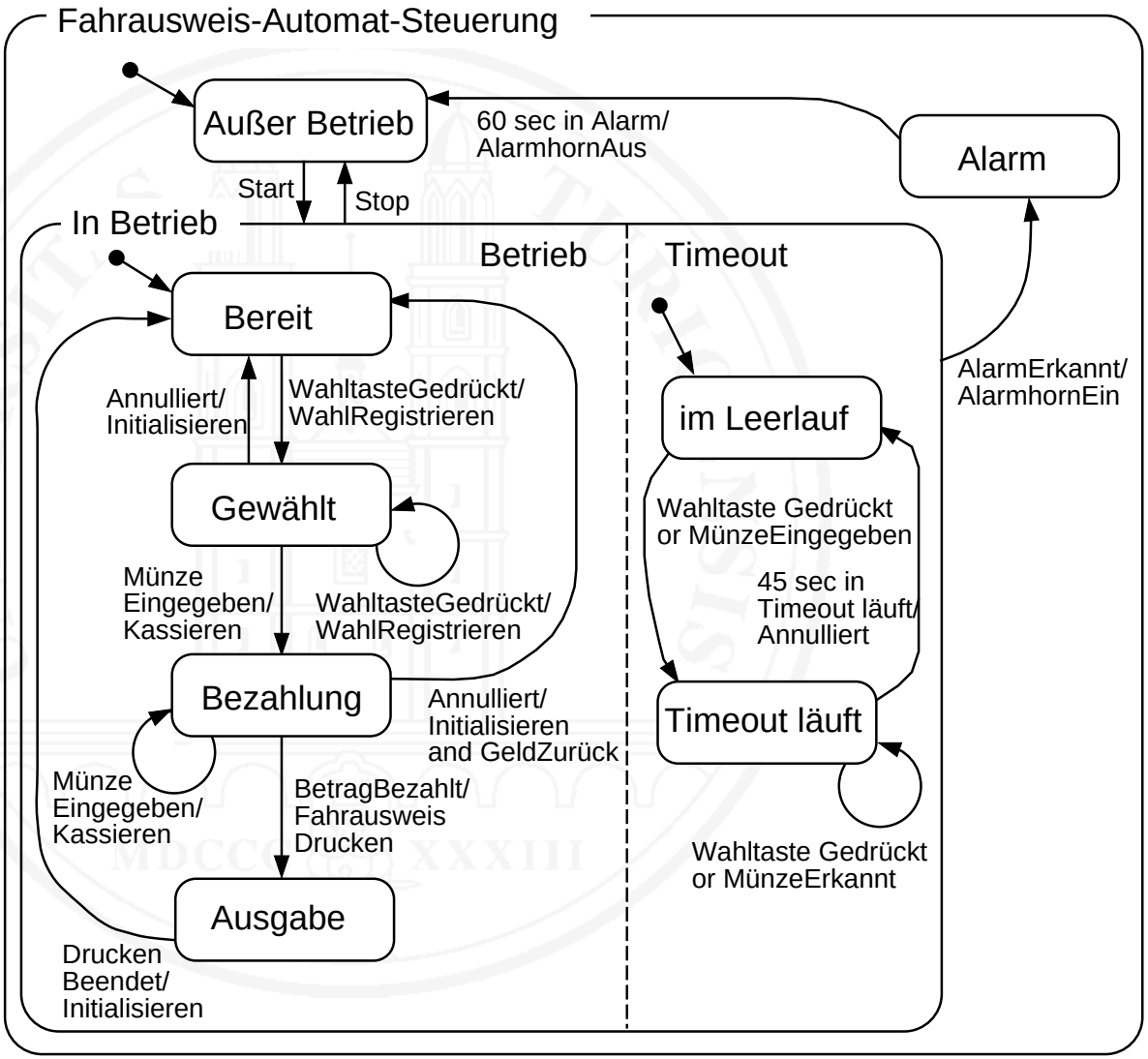
(S,V) (R,U) (T,U)

Wichtig: Zustandsübergänge zwischen parallelen Statecharts sind verboten!

Interpretation paralleler Statecharts (Details)

- Ein Zustand Z kann aus mehreren parallelen Statecharts $S_1,...S_n$ bestehen (Statechart D in Beispiel oben). Dabei ...
  - bilden die Zustände $S_1,...S_n$ zusammen den Zustand Z
  - können in parallelen Statecharts parallel und unabhängig voneinander Zustandsübergänge stattfinden
  - sind Zustandsübergänge zwischen parallelen Statecharts verboten (graphisch: die gestrichelten Trennlinien dürfen nie von einem Zustandsübergangspfeil gekreuzt werden)
  - können sich parallele Statecharts über Ereignisse gegenseitig beeinflussen
- Beim Zustandsübergang nach Z gehen die Statecharts $S_1,...S_n$ in ihren jeweiligen Initialzustand
- Beim Verlassen des Zustands Z werden alle Zustände aller Statecharts $S_1,...S_n$ verlassen

Beispiel: Erweitertes Fahrausweis-Automat-Modell



Aufgabe 4.3

Die Haustür eines Gebäudes ist aus Sicherheitsgründen nur über eine automatische Tür zugänglich.

Diese Tür hat folgende Eigenschaften: Die Tür öffnet automatisch, wenn auf der Außenseite eine Schlüsselkarte mit einprogrammierter Zutrittsberechtigung gesteckt wird, wenn ein Annäherungssensor auf der Innenseite anspricht oder wenn ein Öffnungsknopf auf der Innenseite gedrückt wird. Vier Sekunden nachdem die Tür sich vollständig geöffnet hat, schließt sie automatisch wieder.

An der Tür befinden sich Sensoren, welche folgende Ereignisse melden: Tür vollständig geschlossen, Tür vollständig geöffnet, Hindernis in der Türöffnung. Wird beim Schließen der Tür ein Hindernis in der Türöffnung registriert, so wird der Schließvorgang unterbrochen und die Tür öffnet wieder.

Aufgabe 4.3 (Fortsetzung)

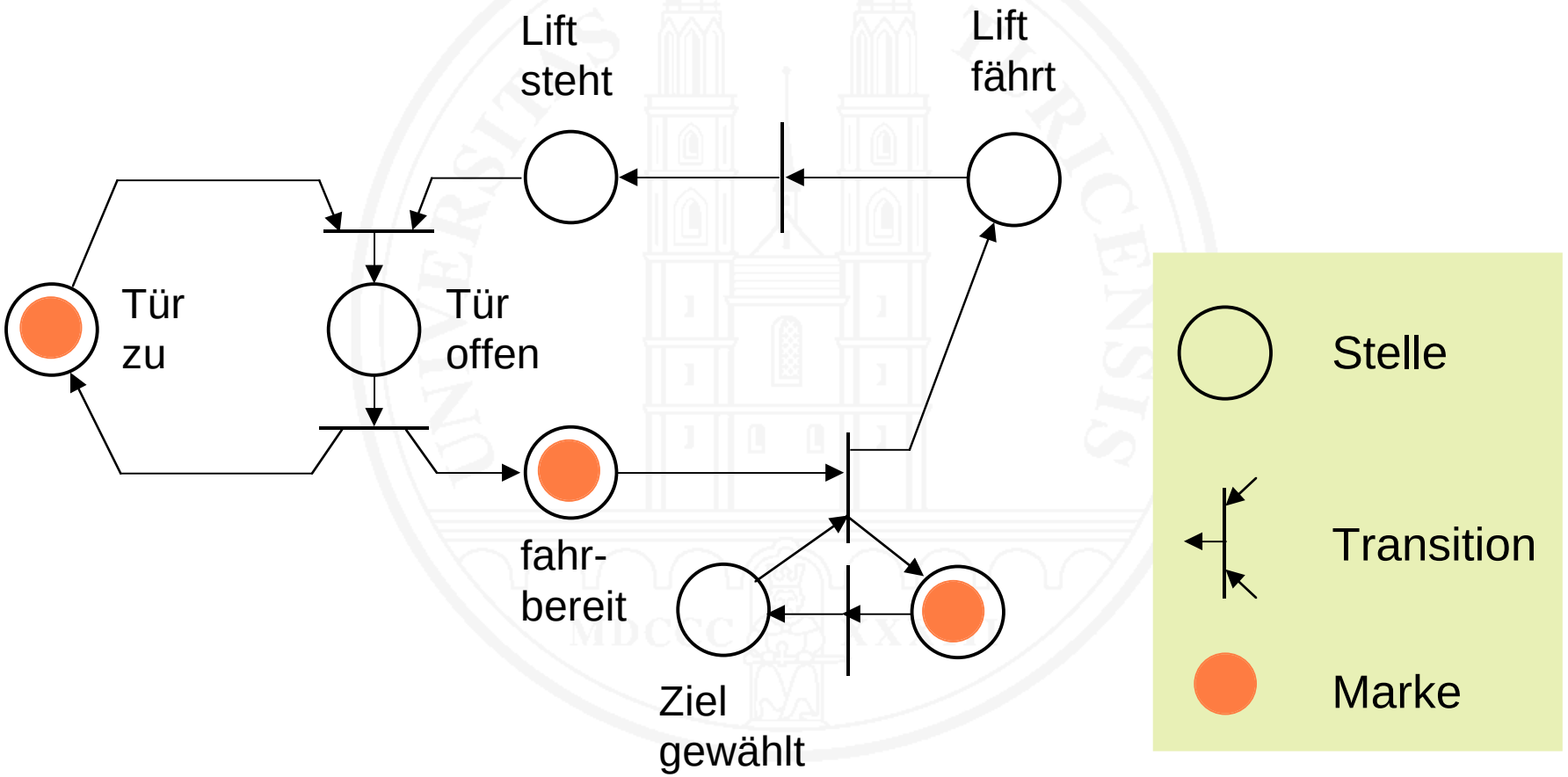
An der Innenseite der Tür befindet sich ein plombierter Notöffnungsknopf. Wird dieser gedrückt, so muss die Tür sofort öffnen und offen bleiben.

Im Hinblick auf eine zu entwerfende automatische Steuerung soll das Verhalten der Tür präzise modelliert werden.

Verwenden Sie zur Modellierung Statecharts.

4.4 Petrinetze

Eine (stark vereinfachte) Liftsteuerung:



Petrinetze: Definitionen, Grundlagen

- Petrinetze (Petri nets) modellieren ereignisgesteuerte, parallele Abläufe (Petri 1962, Reisig 1982, 1985)
- Petrinetze basieren auf einer fundierten mathematischen Theorie (Graphentheorie).

- Ein **Petrinetz** ist ein sogenannter bipartiter gerichteter Graph, d.h. ein Netz aus Knoten und Kanten (gerichteten Verbindungen zwischen Knoten) mit zwei sich abwechselnden Sorten von Knoten.
- Jeder Knoten ist entweder eine Stelle oder eine Transition.
- Stellen und Transitionen wechseln einander im Netz ab: es dürfen nie zwei Stellen oder zwei Transitionen direkt benachbart sein.

Petrinetze: Definitionen, Grundlagen – 2

- **Stellen** (places) speichern Information mit Marken und sind Träger des Zustands des modellierten Systems
- **Transitionen** (transitions) modellieren Verarbeitungen und Zustandsveränderungen: Sie entnehmen Information (d.h. Marken) aus Stellen, transformieren diese und legen sie in neuen Stellen ab.
- **Marken** (tokens), die auf den Stellen des Netzes abgelegt sind, modellieren den aktuellen Systemzustand.
- Durch die Benennung von Stellen, Transitionen und Marken werden begriffliche Bezüge zum modellierten Original hergestellt.
- Einfache Petrinetze verwenden anonyme Marken, lassen pro Stelle höchstens eine Marke zu und modellieren Transitionen ohne Bedingungen.

Notation

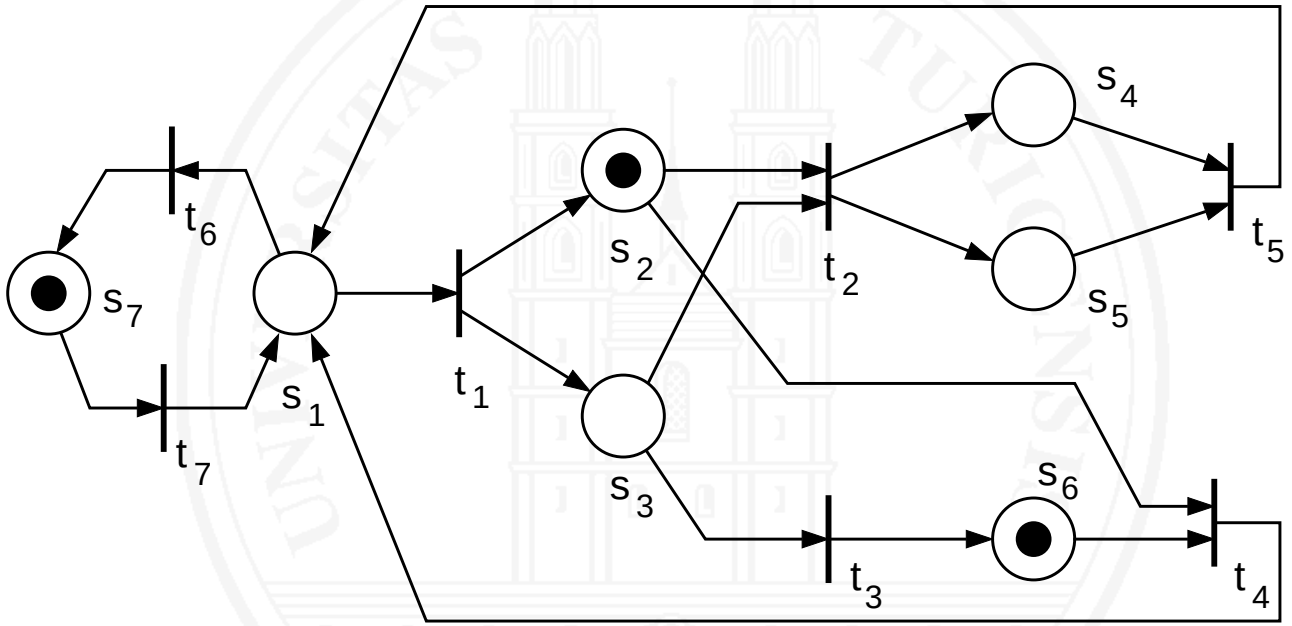
- Stelle — Kreis
- Transition — Strich oder schmales Rechteck, Verbindung mit Ein- und Ausgangsstellen durch Pfeile
- Marke — schwarzes oder farbiges Plättchen

Interpretation einfacher Petrinetze

- Eine Initialbelegung mit Marken stellt den Startzustand des Netzes dar.
- Jede Transition, deren Eingangsstellen alle mit Marken belegt sind, und deren Ausgangsstellen alle frei sind, kann „feuern“.
- Eine Transition feuert, indem sie alle Marken von ihren Eingangsstellen entnimmt und jede ihrer Ausgangsstellen mit je einer Marke belegt.
- Verschiedene Transitionen können gleichzeitig feuern.
- Sind zwei konkurrierende Transitionen (d.h. solche mit mindestens einer gemeinsamen Eingangsstelle) feuerbereit, so feuert nur eine. Welche, bestimmt der Zufall.

Aufgabe 4.4

Gegeben sei das folgende einfache Petrinetz:



a) Spielen Sie mögliche Markierungen durch.

b) Können die Stellen s4, s5 und s6 jemals gleichzeitig markiert sein? Begründen Sie Ihre Aussage.

Aufgabe 4.4 (Fortsetzung)

c) Zeigen Sie, dass Sie von der gegebenen Initialbelegung zu einer Markenbelegung gelangen können, in der zwei Transitionen gleichzeitig feuern können.

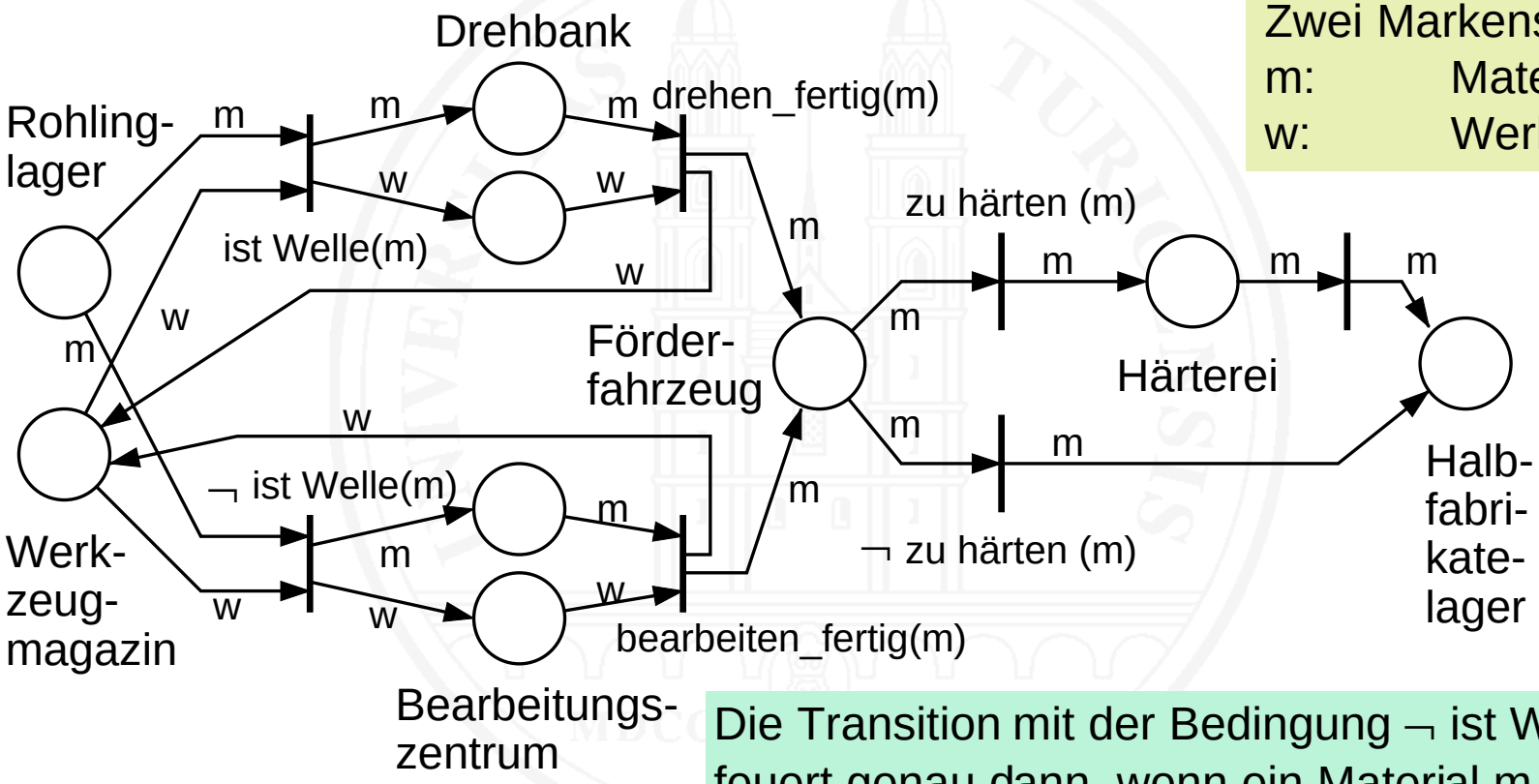
d) Welche Transitionen konkurrieren miteinander?

Prädikat-Transitionsnetze

- Neben einfachen Petrinetzen gibt es verschiedenste Formen von Petrinetzen mit erweiterten Modellierungsmöglichkeiten
- Ein wichtiger Vertreter sind die Prädikat-Transitionsnetze
- Ein Prädikat-Transitionsnetz ist ein Petrinetz mit folgenden zusätzlichen Eigenschaften:
  - Marken können benannt werden
  - Es sind mehrere Marken auf einer Stelle möglich
  - Transitionen können mit Bedingungen versehen werden
  - Namen von Marken an den Transitionspfeilen geben an, auf welche Marken über diesen Pfad zu transportieren sind
- Eine Transition in einem Prädikat-Transitionsnetz feuert genau dann, wenn alle ihre Eingangsstellen mit Marken belegt sind und diese Marken die mit der Transition verbundene Bedingung erfüllen

Beispiel eines Prädikat-Transitionsnetzes

Stark vereinfachtes Modell einer Produktionsstraße



Zwei Markensorten:
m: Material
w: Werkzeug

Die Transition mit der Bedingung ¬ ist Welle(m) feuert genau dann, wenn ein Material m im Rohlinglager und ein Werkzeug w im Werkzeugmagazin vorliegen und m keine Welle ist

Aufgabe 4.5

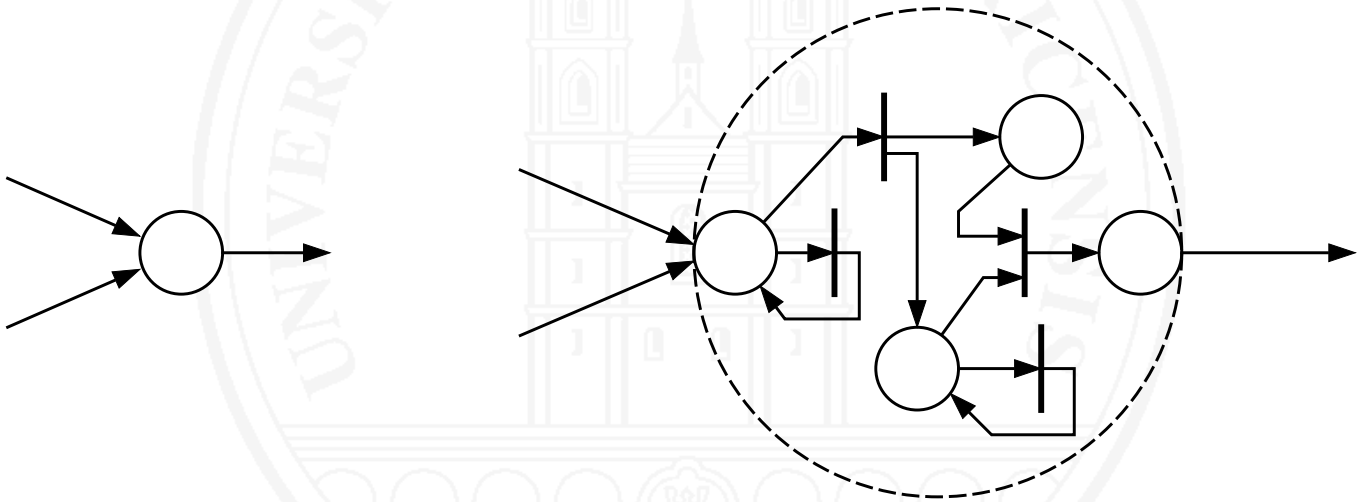
In einer Bankfiliale ist die Beratung von Laufkundschaft (d.h. von Leuten, die sich nicht angemeldet haben) wie folgt organisiert:

Es gibt fünf Bankangestellte, die an zwei Beratungsplätzen Beratungen durchführen. Daneben führen sie auch noch Arbeiten an anderen Arbeitsplätzen in der Filiale aus. Wenn ein Kunde beraten werden will, so wird er einem Beratungsplatz zugewiesen, wenn ein Platz und ein Berater frei sind. Andernfalls muss der Kunde warten.

Modellieren Sie diesen Ablauf mit einem Petrinetz (Prädikat-Transitions-netz).

Verfeinerung von Petrinetzen

- Petrinetze können hierarchisch verfeinert werden, indem eine Stelle oder eine Transition durch ein Subnetz ersetzt wird
- Beispiel: Verfeinerung einer Stelle



- Dabei sind Konsistenzbedingungen zu beachten, beispielsweise darf ein Subnetz, welches eine Stelle verfeinert, die Zahl und Art der erhaltenen Marken nicht verändern

Abstraktion von Petrinetzen

- Petrinetze können vergröbert werden, indem man
  - auf eine Belegung mit Marken verzichtet
  - und nur noch Netze modelliert mit
    - Kanälen (aktiv, transportierend/transformierend)
    - Instanzen (passiv, speichernd)
- Kanal-Instanzennetze eignen sich als Übersichtsmodelle
- Sie sind nicht ausführbar

Literatur

Harel, D. (1987). Statecharts: A Visual Formalism for Complex Systems. *Science of Computer Programming* **8** (1987). 231-274.

Harel, D. (1988). On Visual Formalisms. *Communications of the ACM* **31**, 5 (May 1988). 514-530.

Petri, C.A. (1962). *Kommunikation mit Automaten*. Dissertation, Universität Bonn.

Reisig, W. (1982). *Petrinetze - Eine Einführung*. Berlin-Heidelberg-New York-Tokyo: Springer.

Reisig, W. (1985). *Systementwurf mit Netzen*. Berlin-Heidelberg-New York-Tokyo: Springer.

Yourdon, E. (1989). *Modern Structured Analysis*. Englewood Cliffs, N.J.: Prentice-Hall.