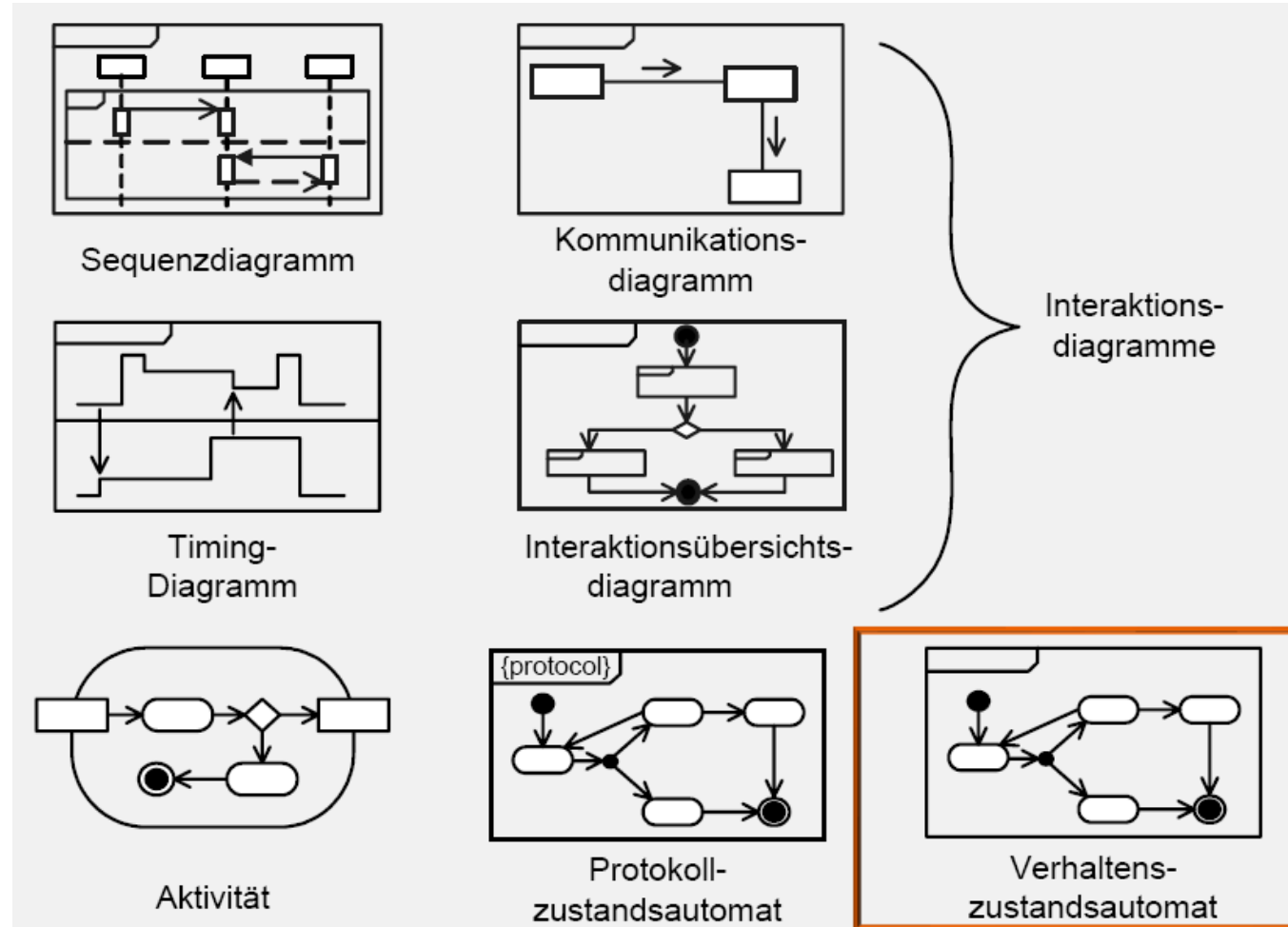


Softwaretechnik 1(A)

UML-Zustandsautomaten

Autorin: Prof. Dr. Sabine Sachweh

Verhaltens-Diagramme



Begriff - Zustand

Zu·stand

Substantiv, maskulin [der]

- a) augenblickliches Beschaffen-, Geartetsein;
Art und Weise des Vorhandenseins von jemandem, einer Sache in einem bestimmten Augenblick;
Verfassung, Beschaffenheit
"ein normaler, ungewohnter Zustand"
- b) augenblicklich bestehende Lage, Situation, Verhältnisse
"ein gesetzloser, chaotischer Zustand"

Beispiele



Wechselschalter
Zustände: ein, aus

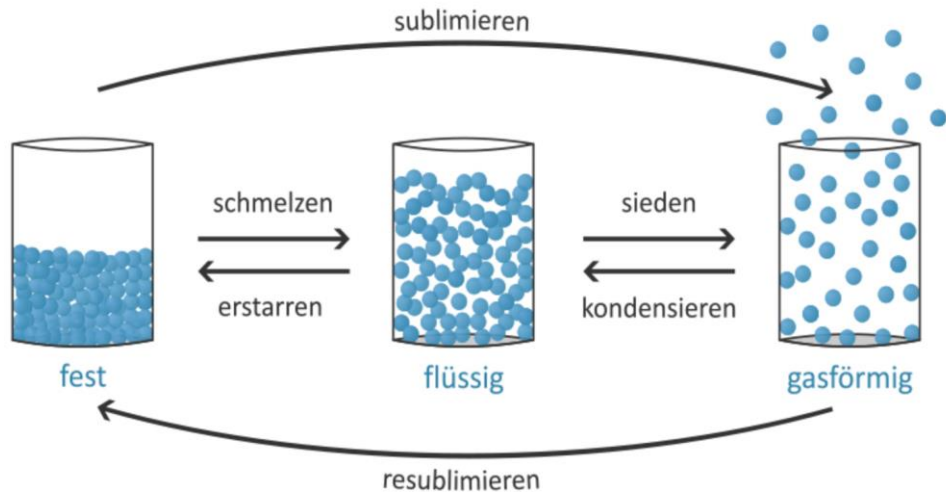


Geschäfte
Zustände: offen,
geschlossen

Ampel
Zustände: rot, grün



Schranke
Zustände: offen,
geschlossen

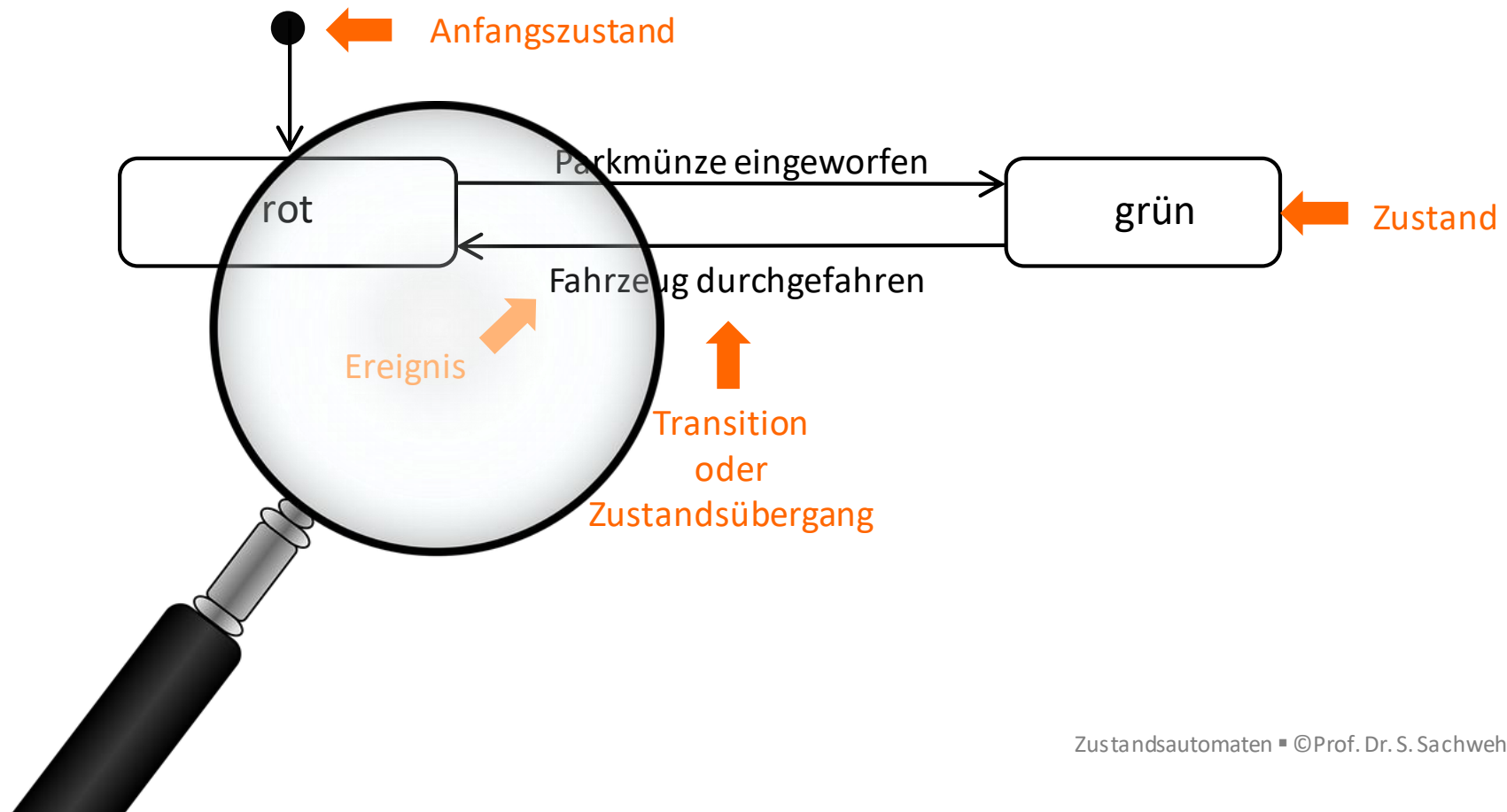


Wasser
(Aggregat-)Zustände: fest, flüssig, gasförmig



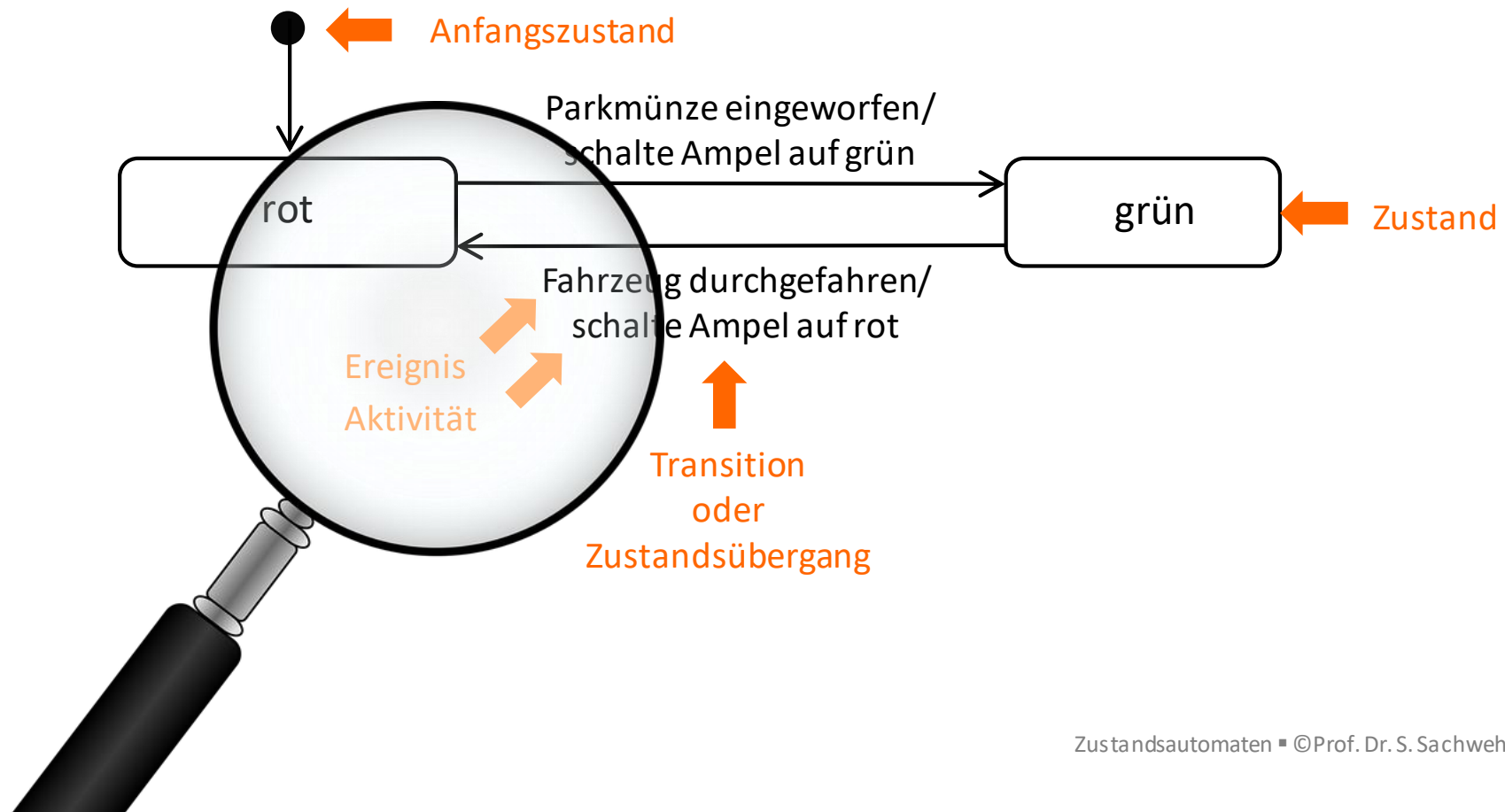
Zustandsautomat

Bsp: Ampel an der Ausfahrt eines Parkhauses



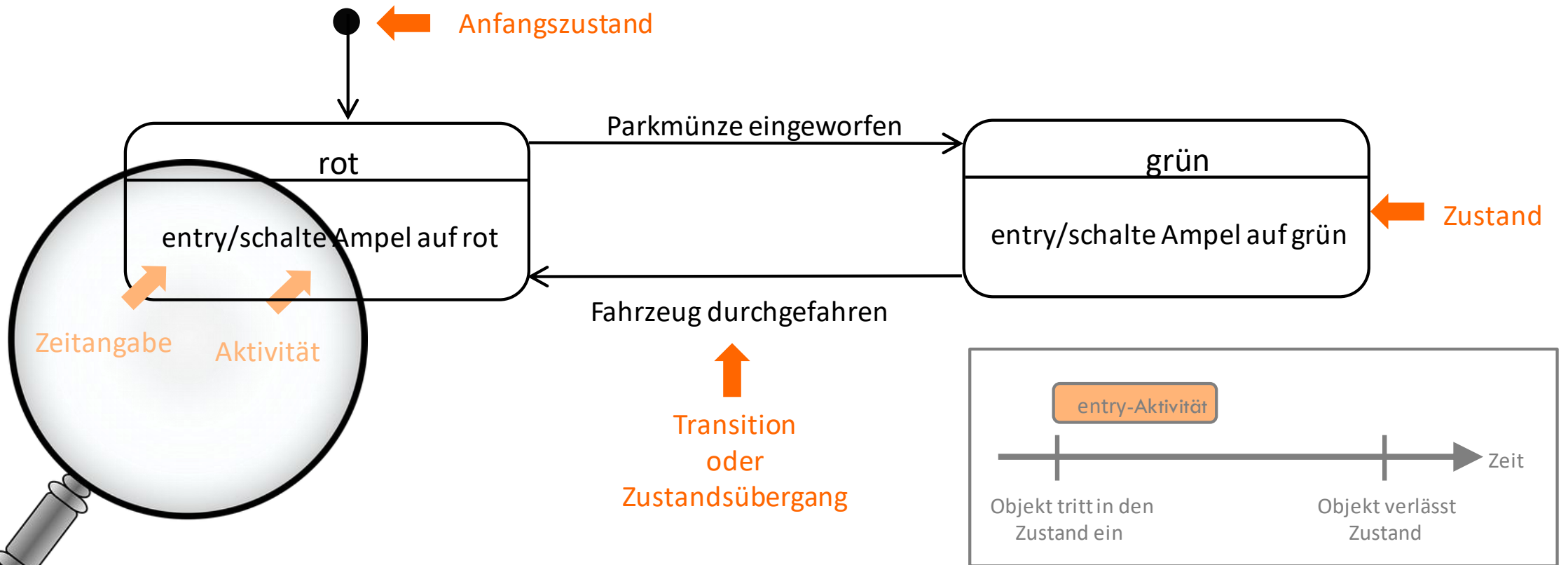
Zustandsautomat (mit Aktivitäten an Transitionen)

Bsp: Ampel an der Ausfahrt eines Parkhauses



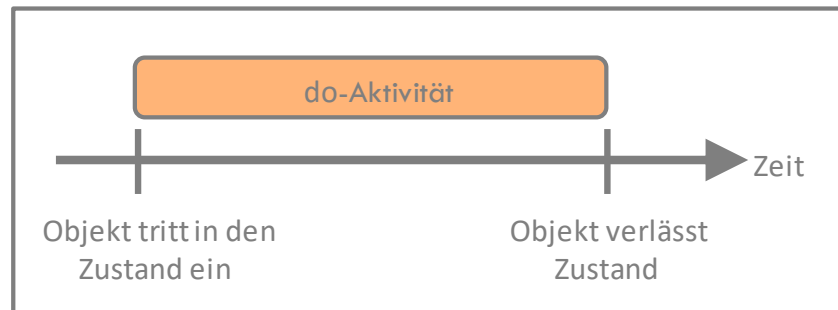
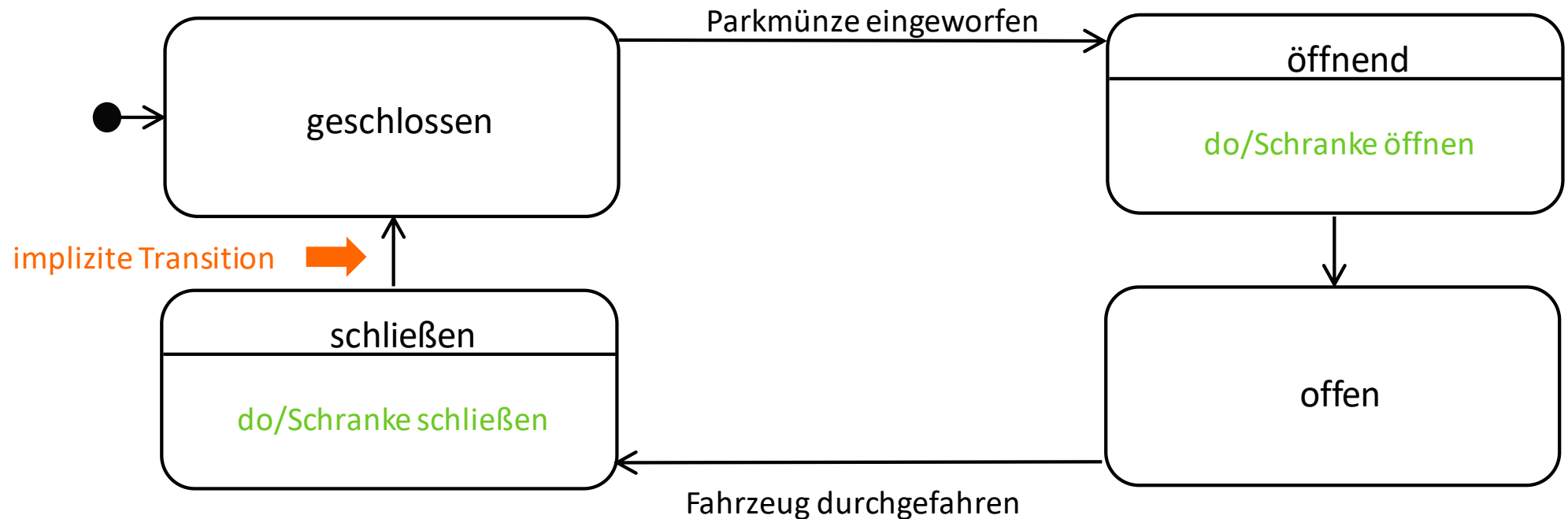
Zustandsautomat (mit Entry-Aktivitäten in Zuständen)

Bsp: Ampel an der Ausfahrt eines Parkhauses



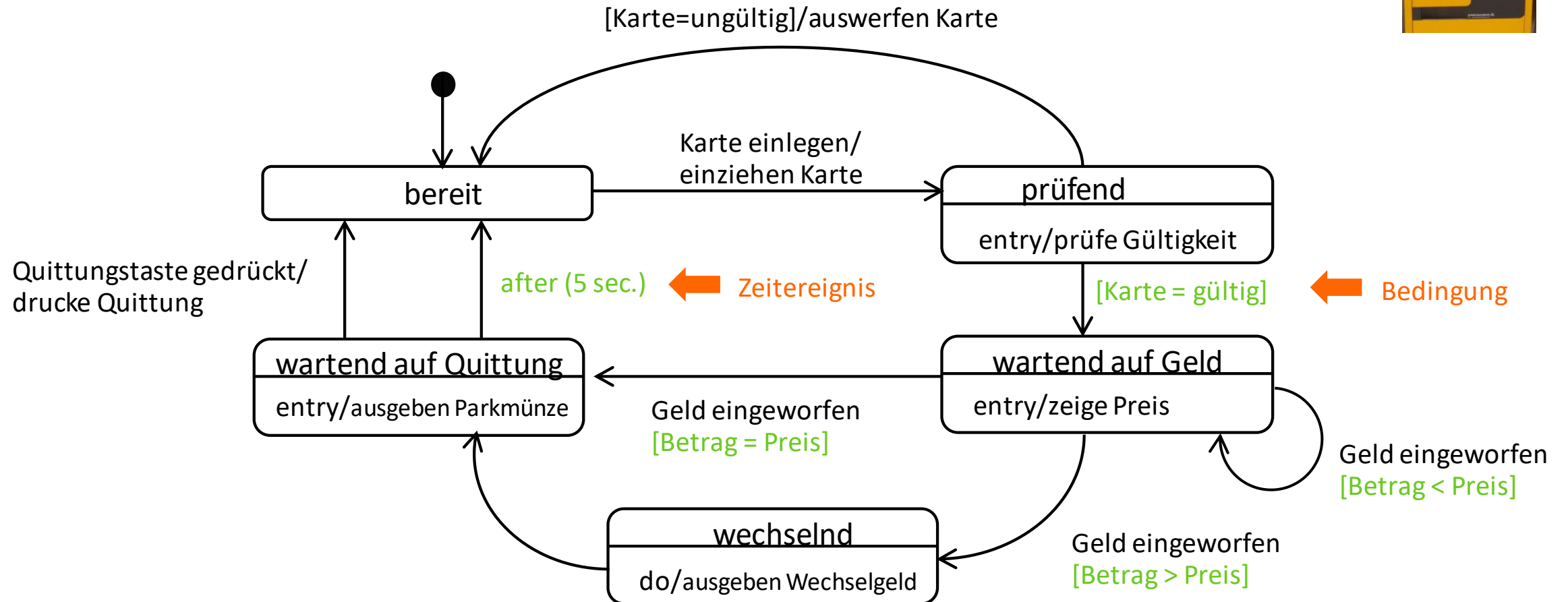
Zustandsautomat (mit Do-Aktivitäten in Zuständen)

Bsp: Schranke an der Ausfahrt eines Parkhauses



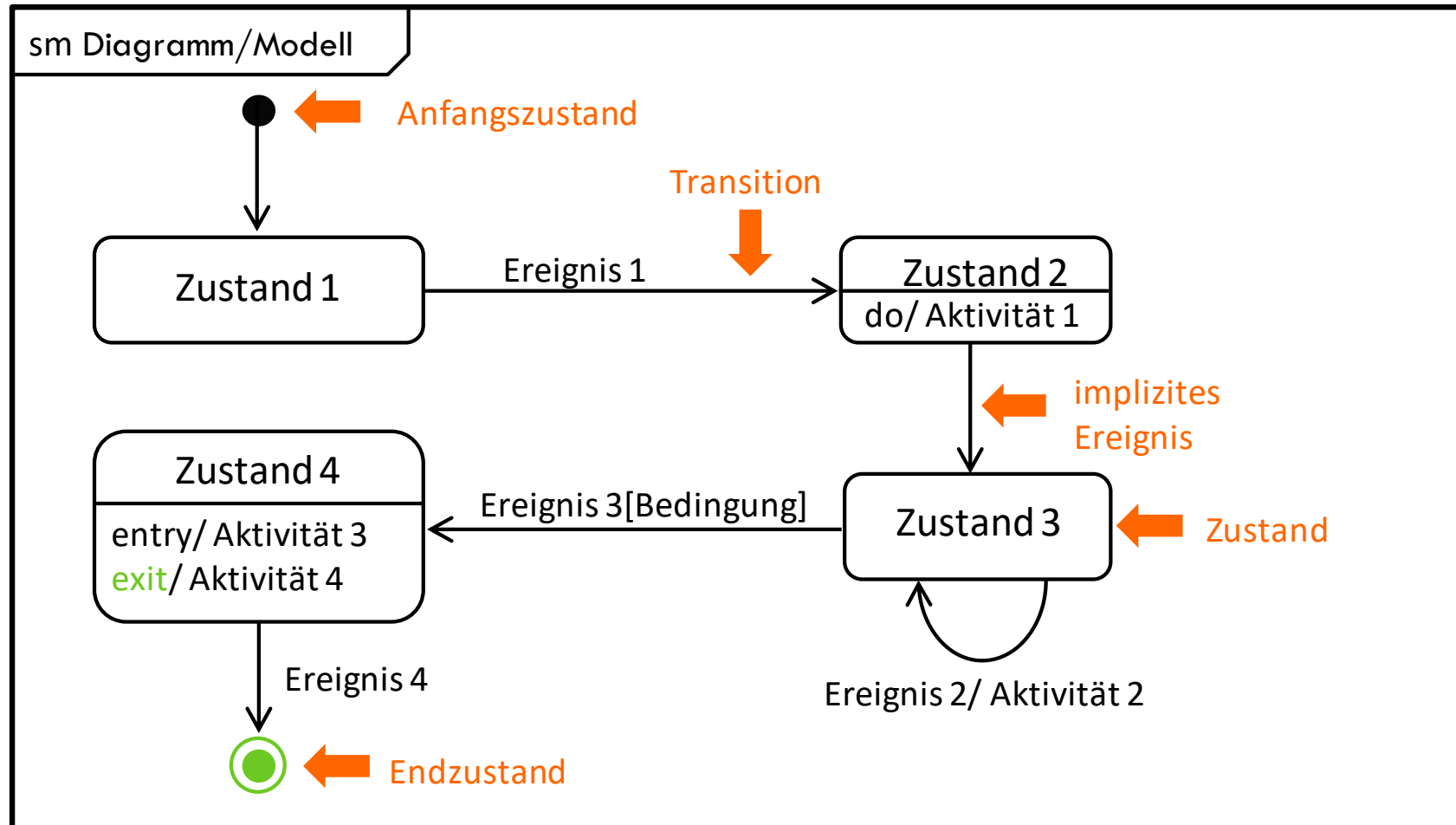
Zustandsautomat (mit Bedingungen und Zeitereignis)

Bsp: Parkhaus mit Kassenautomat



Zustandsautomat - Notation

State machine



Anfangszustand
ist notwendig

—
Endzustand
ist optional

Zustandsautomat - Begriffsdefinition

Definition



Ein **Zustandsautomat** (finite state machine) besteht aus Zuständen und Zustandsübergängen (Transitionen).

- Ein Zustand ist eine Zeitspanne, in der ein Objekt auf ein Ereignis wartet
- Zustandsübergänge werden durch Ereignisse ausgelöst
- Ein Ereignis tritt immer zu einem Zeitpunkt auf und besitzt keine Dauer
- Ein Objekt kann – nacheinander – mehrere Zustände durchlaufen.
- Zu einem Zeitpunkt befindet es sich in genau einem Zustand.
- Dient zur Beschreibung des Lebenszyklusses (dynamisches Verhalten) von Objekten
- Zustandsautomaten werden durch Zustandsdiagramme (statechart diagrams) dargestellt.

Beschreibung des Lebenszyklusses (dynamisches Verhalten) von Objekten

Zustände modellieren

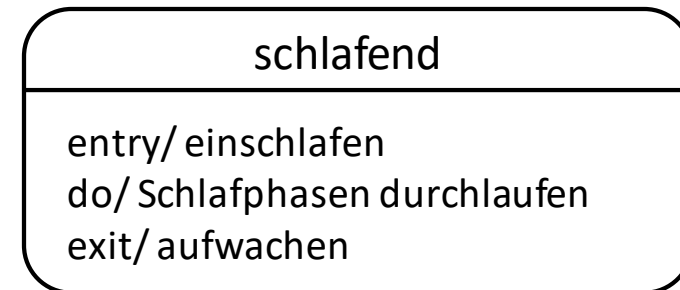


- Zustandsname ist optional
- Zustände ohne Namen heißen anonyme Zustände und sind alle voneinander verschieden
- Zustandsname soll kein Verb sein z.B. ausgeliehen statt ausleihen
- Innerhalb eines Zustandsautomaten muss jeder Zustandsname eindeutig sein
- Benannte Zustände können aus Gründen der besseren Lesbarkeit mehrmals in ein Diagramm eingetragen werden.
- Anfangszustand 
Pseudozustand, der mit einem echten Zustand durch eine Transition verbunden ist
- Endzustand 
Kein weiteres Ereignis kann mehr folgen \Rightarrow Objekt hört auf zu existieren

Verarbeitung (durch Aktivitäten) modellieren

Aktivität

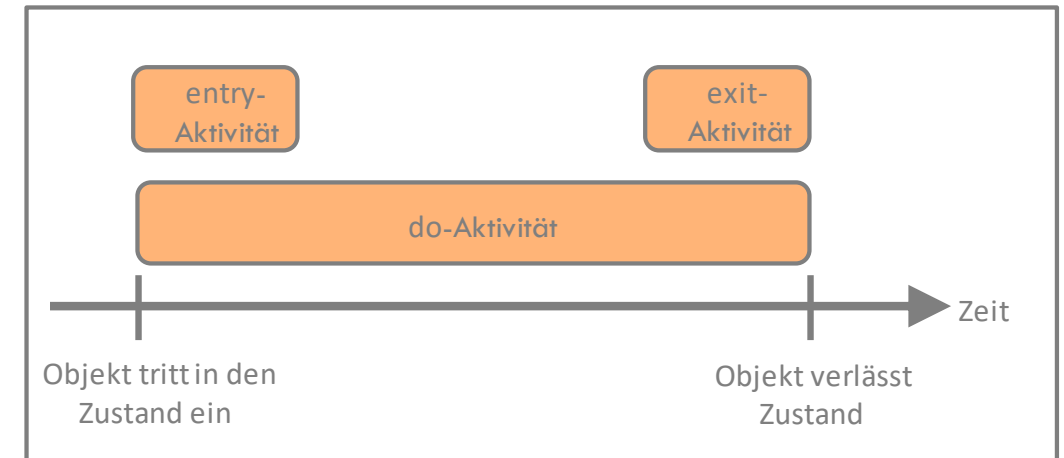
- Mit einer Transition kann eine Verarbeitung verbunden sein
⇒ modelliert durch Aktivität im Zustandsautomaten
- an einer Transition
besitzen Aktivitäten theoretisch keine Dauer, d.h. für die Modellierung nicht relevant
- in einem Zustand besitzen eine Dauer
 - entry-Aktivität
 - exit-Aktivität
 - do-Aktivität



Verarbeitung mit Dauer (im Zustand) modellieren

Aktivitäten

- in einem Zustand besitzen eine Dauer
 - entry-Aktivität
 - löst automatisch beim Eintritt in den Zustand aus
 - exit-Aktivität
 - löst automatisch beim Verlassen des Zustandes aus
⇒ terminiert selbständig
 - do-Aktivität
 - beginnt, wenn Objekt einen Zustand einnimmt und endet, wenn es ihn verlässt
⇒ ermöglicht implizite Übergänge

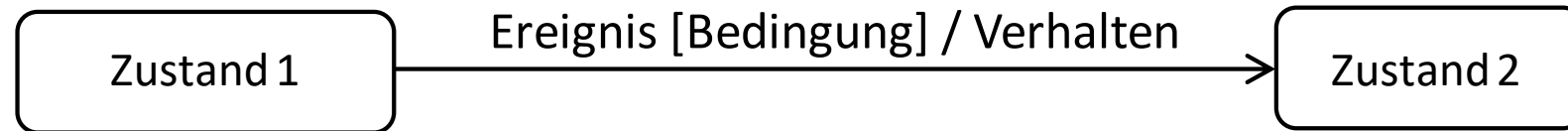


Jeder Zustand besitzt maximal eine entry-Aktivität bzw. eine exit-Aktivität!

Zustandsübergänge (Transitionen) modellieren

Zustandsübergang (Transition)

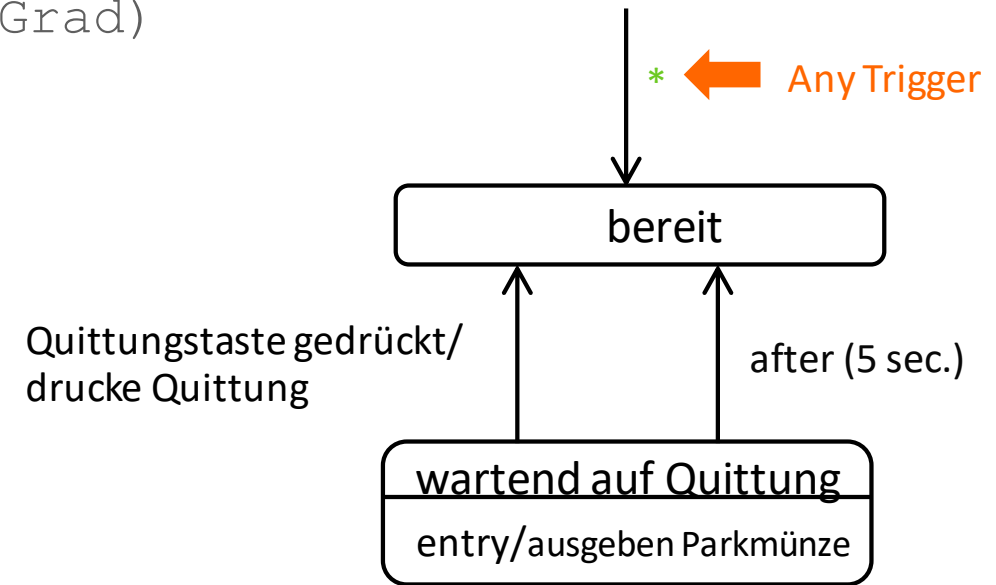
- Verbindet zwei Zustände
- Wird durch ein Ereignis ausgelöst
- Kann mit einer Bedingung verknüpft sein
- Kann mit einer Aktivität verbunden sein



Zustandsübergänge modellieren - Ereignis

Ein Ereignis (trigger) kann sein

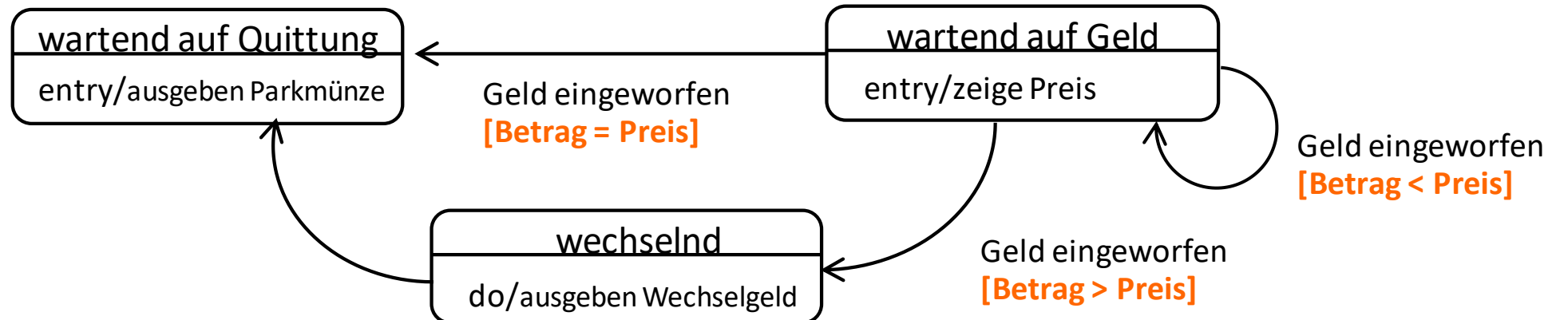
- Bedingung, die wahr wird, z.B. `when (Temperatur > 100 Grad)`
- Signal, z.B. `Quittungstaste gedrückt`
- Botschaft (Operationsaufruf), z.B. `MWST_berechnen()`
- zeitliches Ereignis
 - Zeitpunkt: `at (31.12.2000 24:00)`
 - Zeitspanne: `after (10 sec)`
- implizites Ereignis
 - Beendigung der Aktivität in dem jeweiligen Zustand
- AnyTrigger „all“
 - Platzhalter für alle Ereignisse, die an keiner anderen ausgehenden Transition des Zustands angetragen sind



Zustandsübergänge modellieren - Bedingung

Eine Bedingung/Wächter (guard) kann ein Ereignis ergänzen

- Die Transition feuert nur dann, wenn ...
 - das zugehörige Ereignis eintritt und
 - die spezifizierte Bedingung erfüllt ist
- z.B. Ampel wird grün [Strasse ist frei]

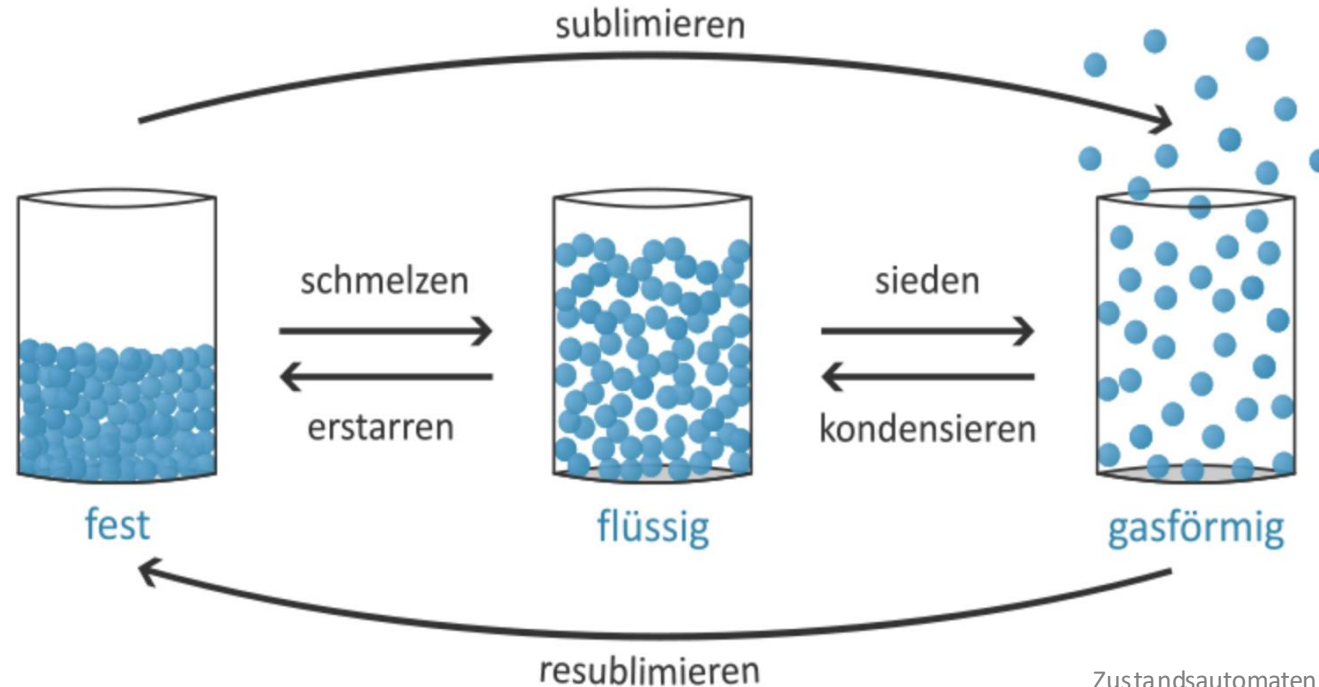




Aufgabe: Aggregatzustände des Wassers (10 Minuten)

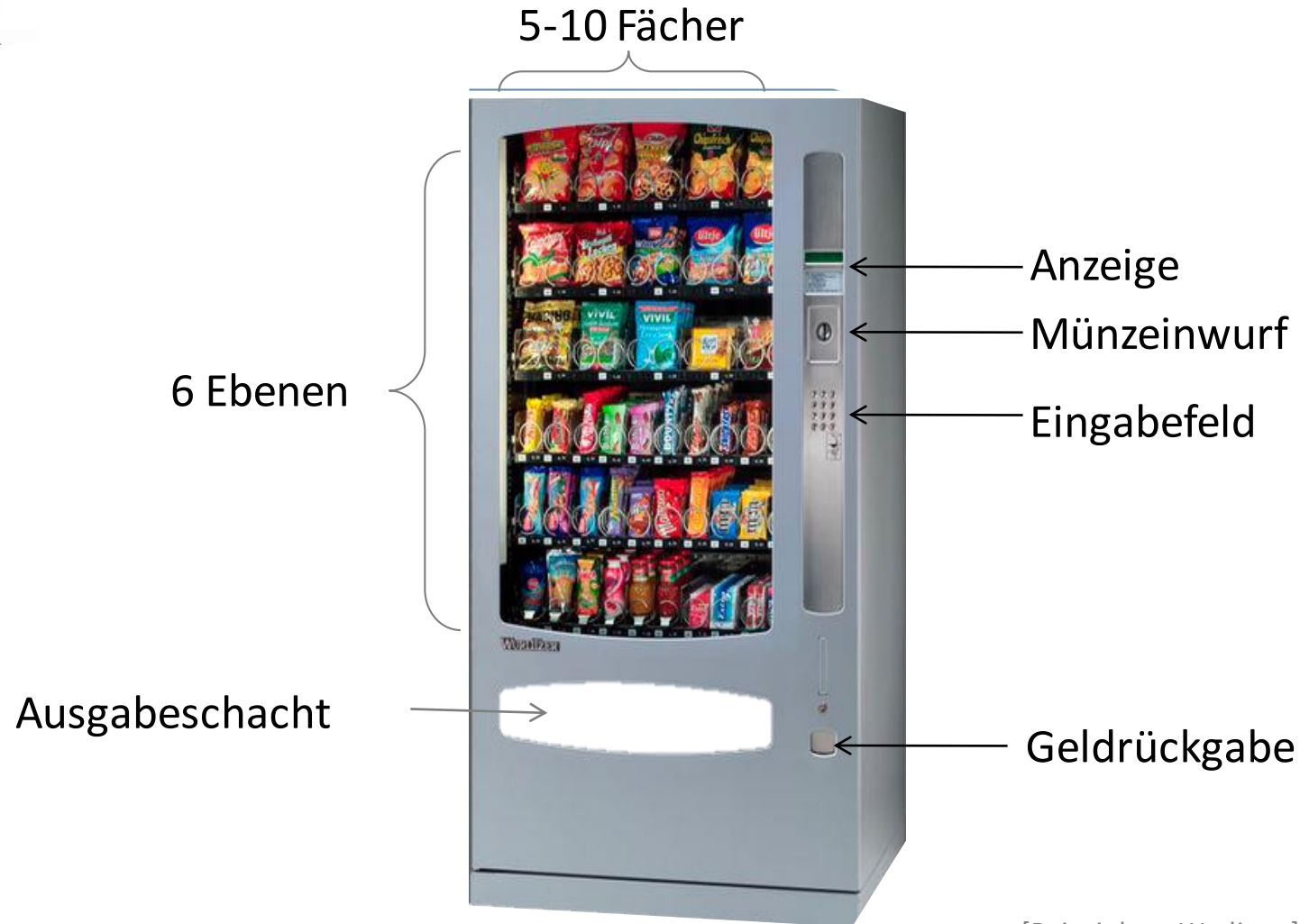
Aggregatzustände des Wassers

Modellieren Sie die Aggregatzustände des Wassers in UML durch ein (Verhaltens-)Zustandsdiagramm. Verwenden Sie u.a. Temperaturangaben, um die Transitionen zu beschreiben.





Aufgabe: Snackautomat (10 Minuten)





Aufgabe: Snackautomat (10 Minuten)

Snackautomat

Zu Beginn **wartet der Automat auf die Auswahl des Produktes** durch den Kunden. Die Produktauswahl findet in zwei Schritten statt. Zunächst **wählt der Kunde die Ebene**, in welcher sich das gewünschte Produkt befindet. Wählt der Kunde eine **Ebene aus, die nicht existiert**, wartet der Automat weiter auf die Produktauswahl. Ist die Ebene gewählt, gibt der Kunde das **Fach des gewünschten Produktes** an. Ist das gewählte **Produktfach ausverkauft**, bricht der Automat den Kaufvorgang ab und wartet erneut auf die Produktauswahl. Nach erfolgreicher Produktauswahl wirft der Kunde so lange Münzen ein, bis der eingeworfene **Betrag gleich oder größer dem Preis** des ausgewählten Produktes ist. Solange der Kunde nicht ausreichend Geld in den Automaten eingeworfen hat, **wartet der Automat auf den Einwurf des fehlenden Geldbetrages**. Hat der Kunde ausreichend Geld eingeworfen, befördert der Automat das gewählte **Produkt in den Ausgabeschacht**. Danach entnimmt der Kunde das Produkt. Hat der Kunde **genau so viel Geld eingeworfen, wie das Produkt kostet**, wartet der Automat auf die nächste Produktauswahl. Hat der Kunde das Produkt entnommen und **mehr Geld eingeworfen**, als das ausgewählte Produkt kostet, so gibt der **Automat das Rückgeld** in den Ausgabeschacht aus. Nachdem der Kunde das Rückgeld entnommen hat, wartet der Automat wieder auf die **nächste Produktauswahl**.

Parallele Unterzustände

Beispiel: Sekt

Wie lässt sich der Zustand
von Sekt beschreiben?

Temperatur ?

Blubber-
effekt ?

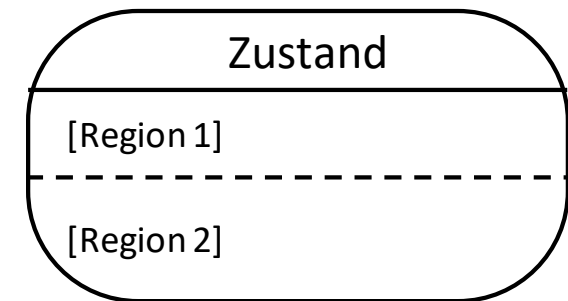
Trinkbarkeit?



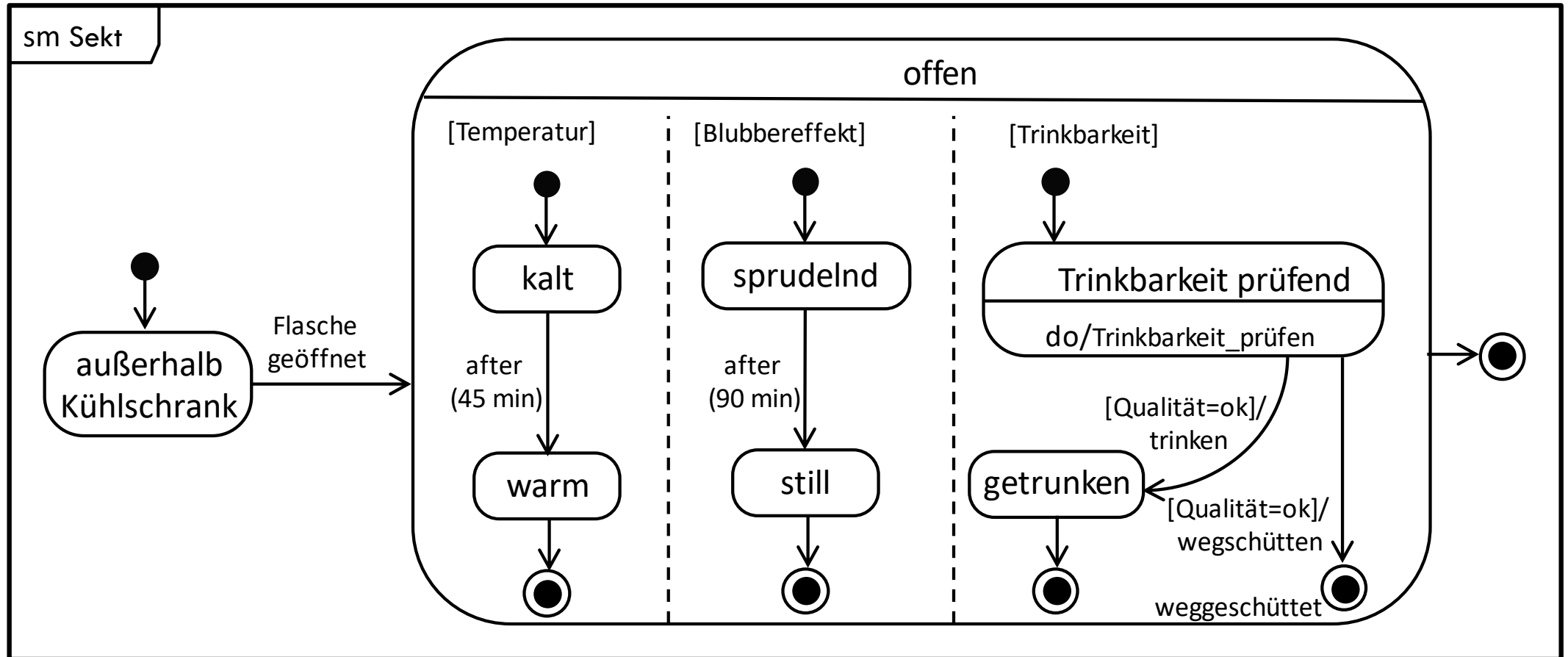
Regionen

Pseudozustand: Region

- Beschreibt zueinander parallele Zustandsfolgen durch Pseudozustände: die sogenannten Regionen.
- Ein zusammengesetzter Zustand kann so durch parallele Regionen strukturiert werden.
- Eine Region definiert einen eigenen Bereich für eine Zustandsfolge.
- Die für den zusammengesetzten Zustand definierten Ereignisse gelten gleichzeitig für alle Regionen.

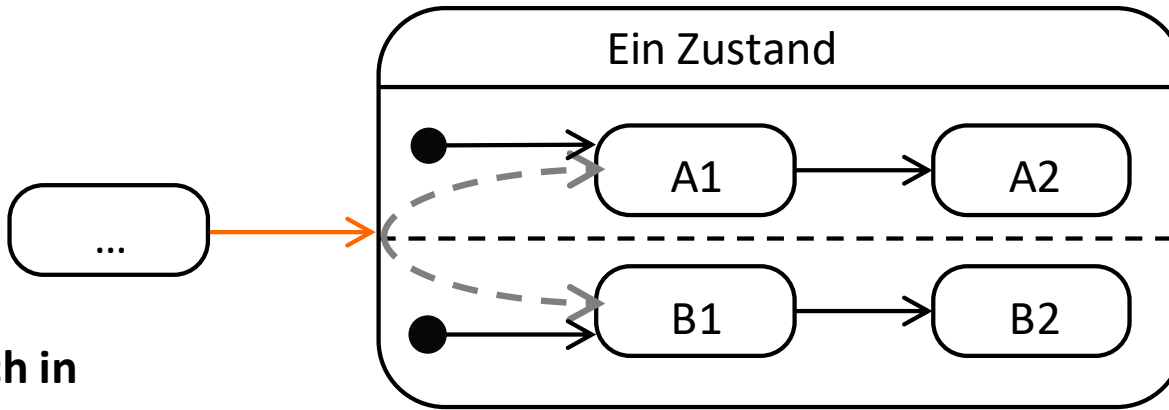


Regionen – Beispiel: Sekt



Bereiche – Start der Ausführung

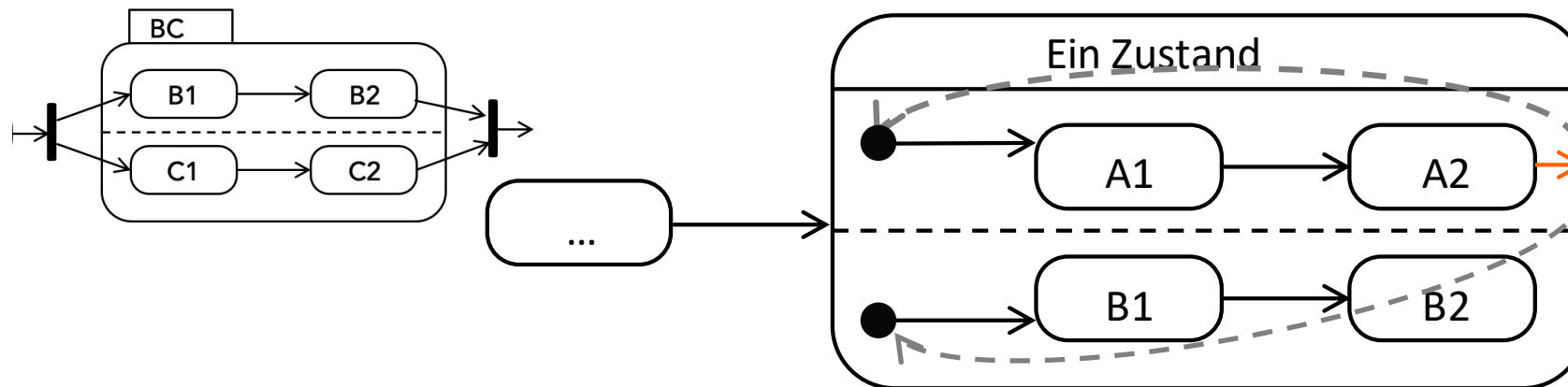
Durch Synchronisation
und Gabelung kann auch in
andere Zustände gesprungen
werden!



Zustandsübergang von
außen



Die Startzustände aller
Regionen werden aktiv

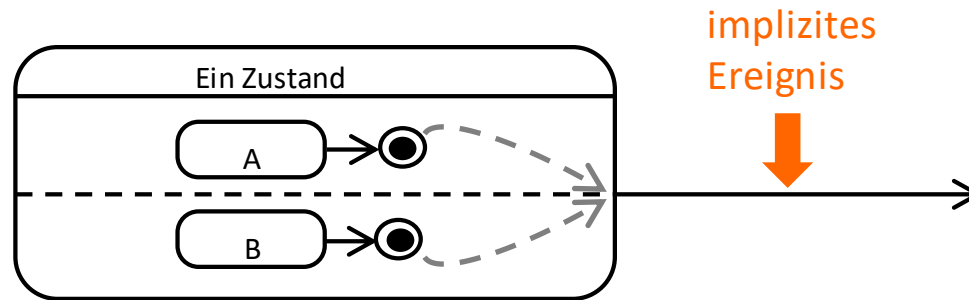


Zustandsübergang von innen
an den Rand

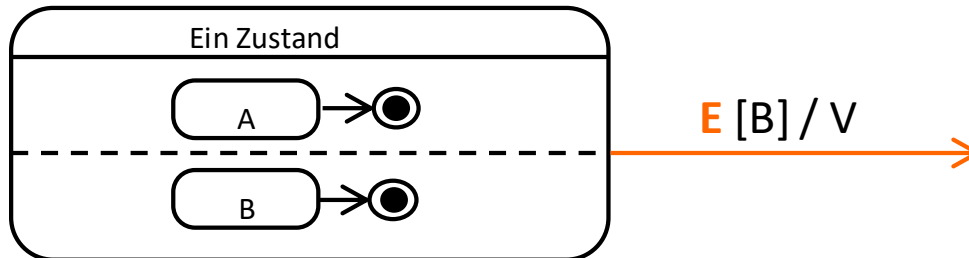


Warten Bis alle Regionen
bereit sind, um gemeinsam
mit den Startzuständen zu
beginnen

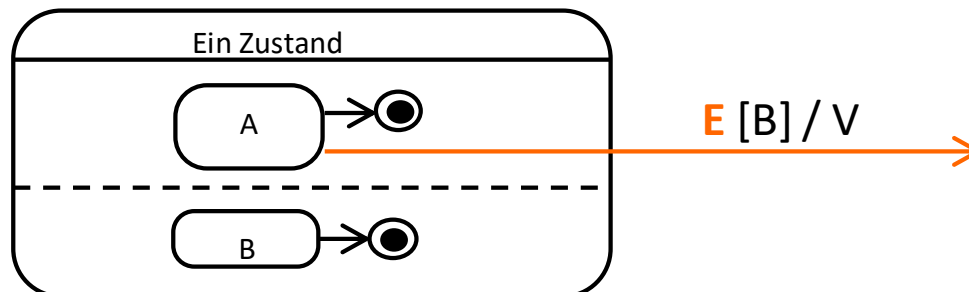
Bereiche – Beenden der Ausführung



Bei einem **impliziten Ereignis** wird gewartet bis die Endzustände in allen betroffenen Regionen erreicht wurden.



Bei einem **expliziten Ereignis** des **zusammengesetzten Zustands**, werden bei Eintritt des Ereignisses alle Regionen (des zusammengesetzten Zustands) **sofort beendet/verlassen**.



Bei einem **expliziten Ereignis** eines **Zustands in einem Bereich**, werden bei Eintritt des Ereignisses alle Regionen (des zusammengesetzten Zustands) **sofort beendet/verlassen**.

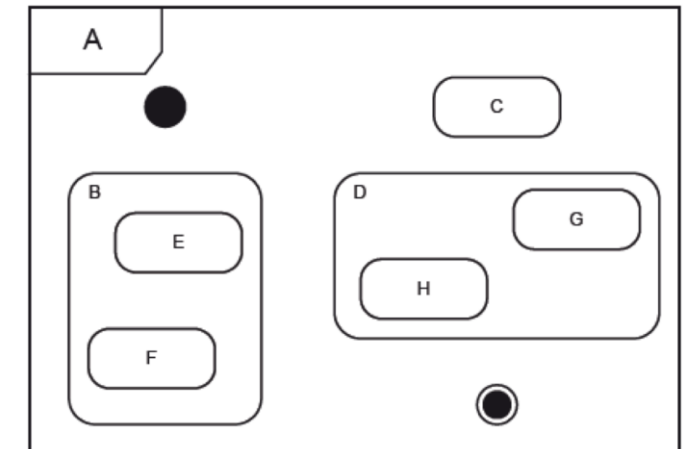
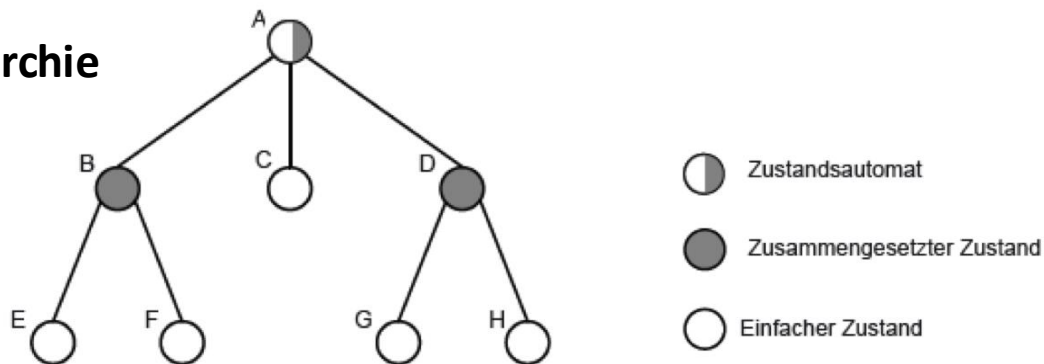
Verfeinerung von Zuständen

Zusammengesetzte Zustände

Verfeinerung von Zuständen

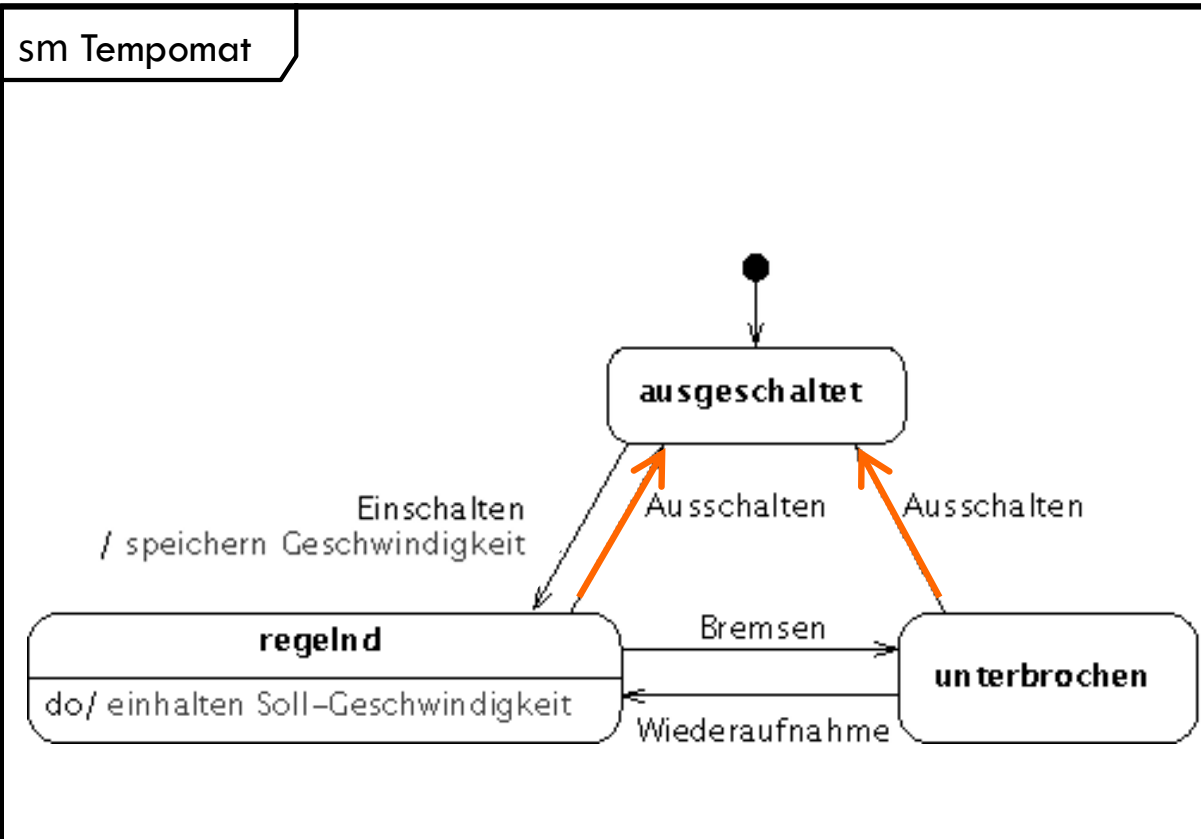
- Ein Zustand, der verfeinert wird, heißt auch zusammengesetzter Zustand
- Ein Zustand kann durch Unterzustände (substates) verfeinert werden
- Alle Unterzustände schließen sich gegenseitig aus
- Mehrere Unterzustände können auch parallel ablaufen

Zustandshierarchie

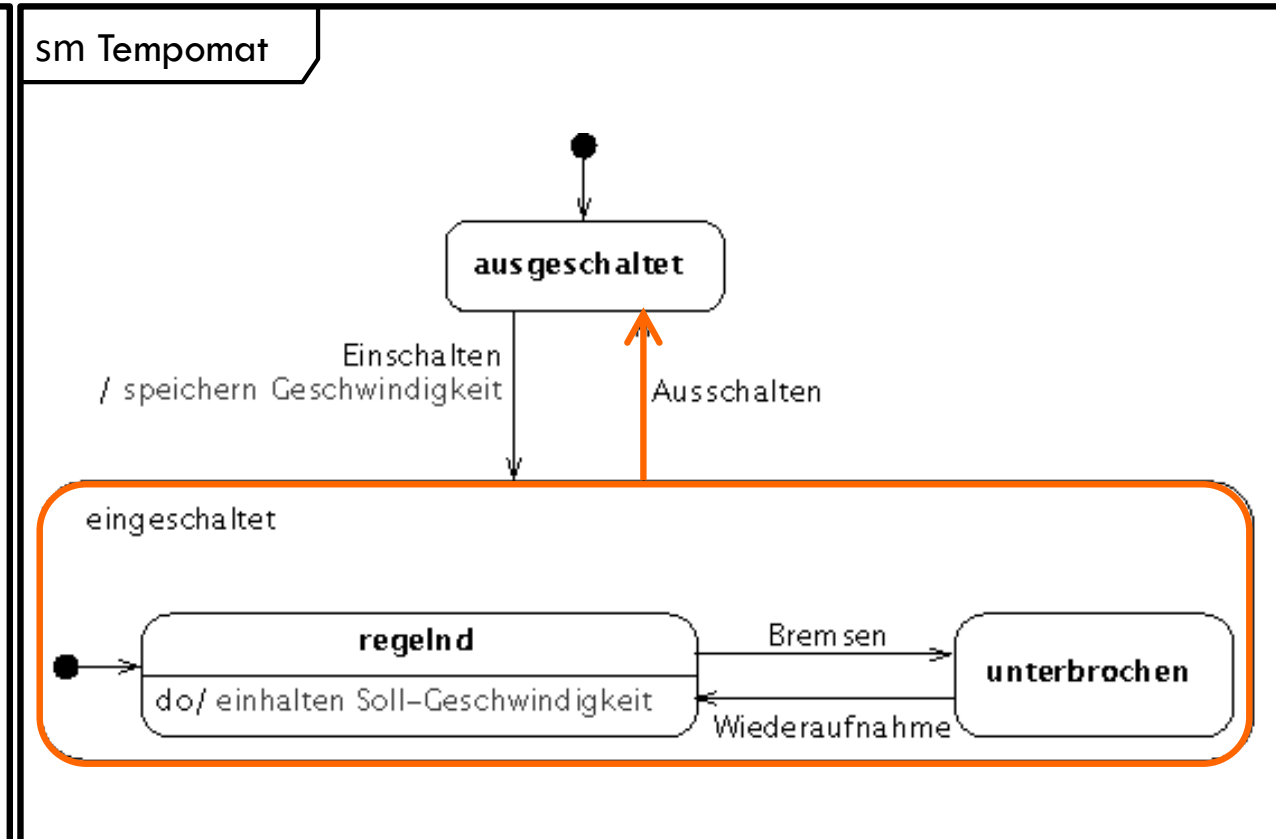


**Zustandsautomat A
ohne Transitionen**

Zusammengesetzte Zustände – Beispiel: Tempomat



ohne Verfeinerung



mit Verfeinerung

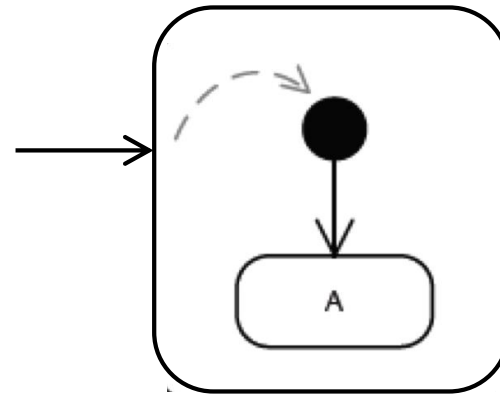
Zusammengesetzte Zustände – Start der Ausführung

- Zusammengesetzter Zustand enthält eine oder mehrere Regionen, die jeweils Unterzustände enthalten
 - orthogonale Regionen können parallel ausgeführt werden

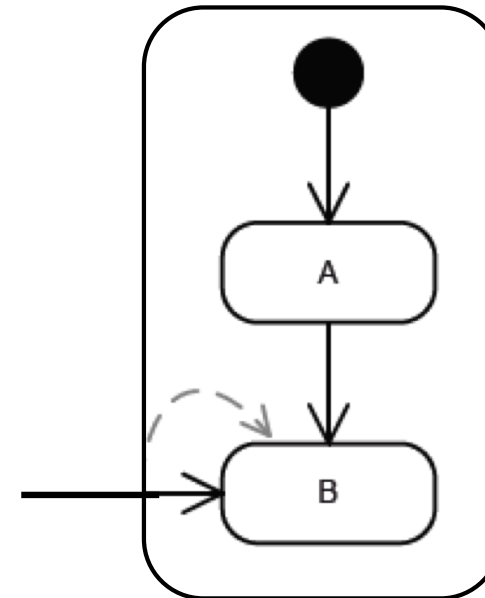
- Start der Ausführung des Verhaltens in einer Region
 - Standardaktivierung des Startzustands durch Aktivierung des umschließenden Zustands bzw. des Zustandsautomats (auf der höchsten Ebene)
 - Explizite Aktivierung durch Transition zu einem Zustand oder Pseudozustand in der Region
⇒ Standardaktivierung aller zu dieser orthogonalen Regionen

Auch möglich: explizite Aktivierung mehrerer Regionen durch Parallelisierung

Zusammengesetzte Zustände – Start der Ausführung

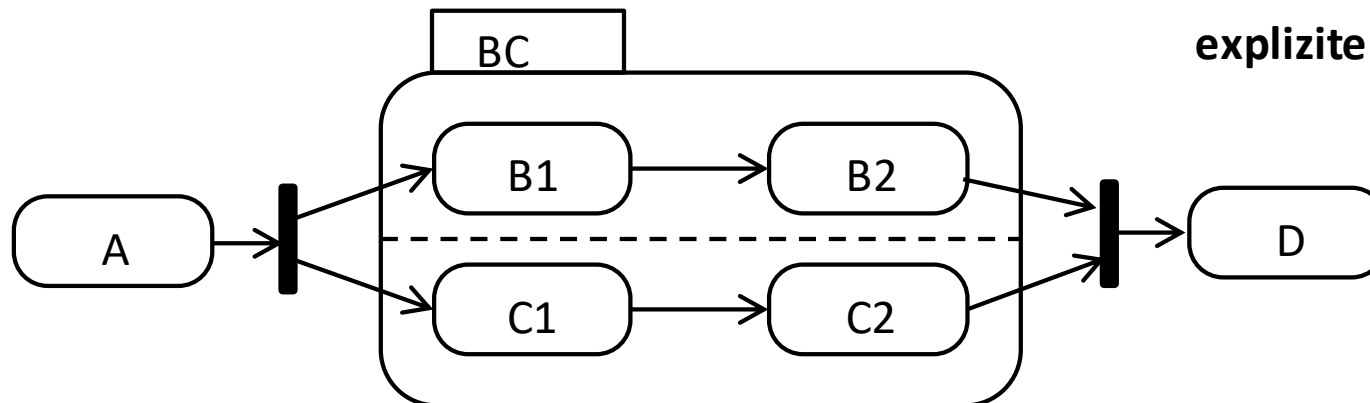


Standardaktivierung



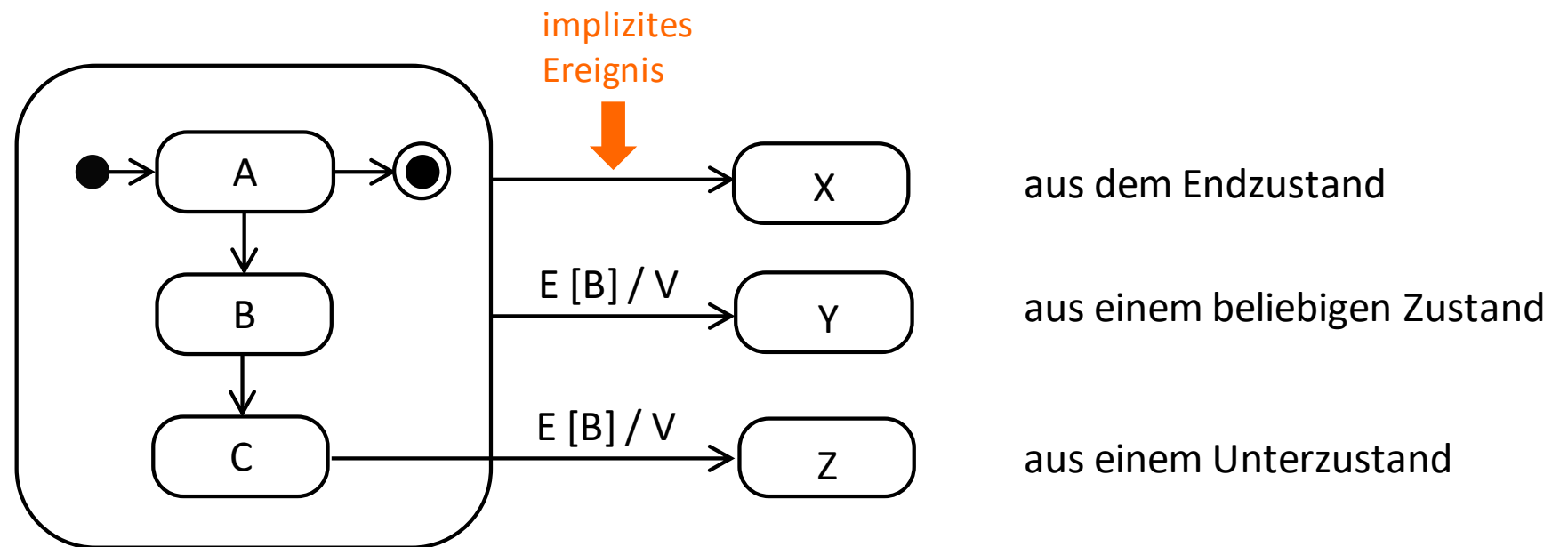
explizite Aktivierung

Parallelisierung
und
Synchronisation



[UML 2 glasklar]

Zusammengesetzte Zustände – Beenden der Ausführung

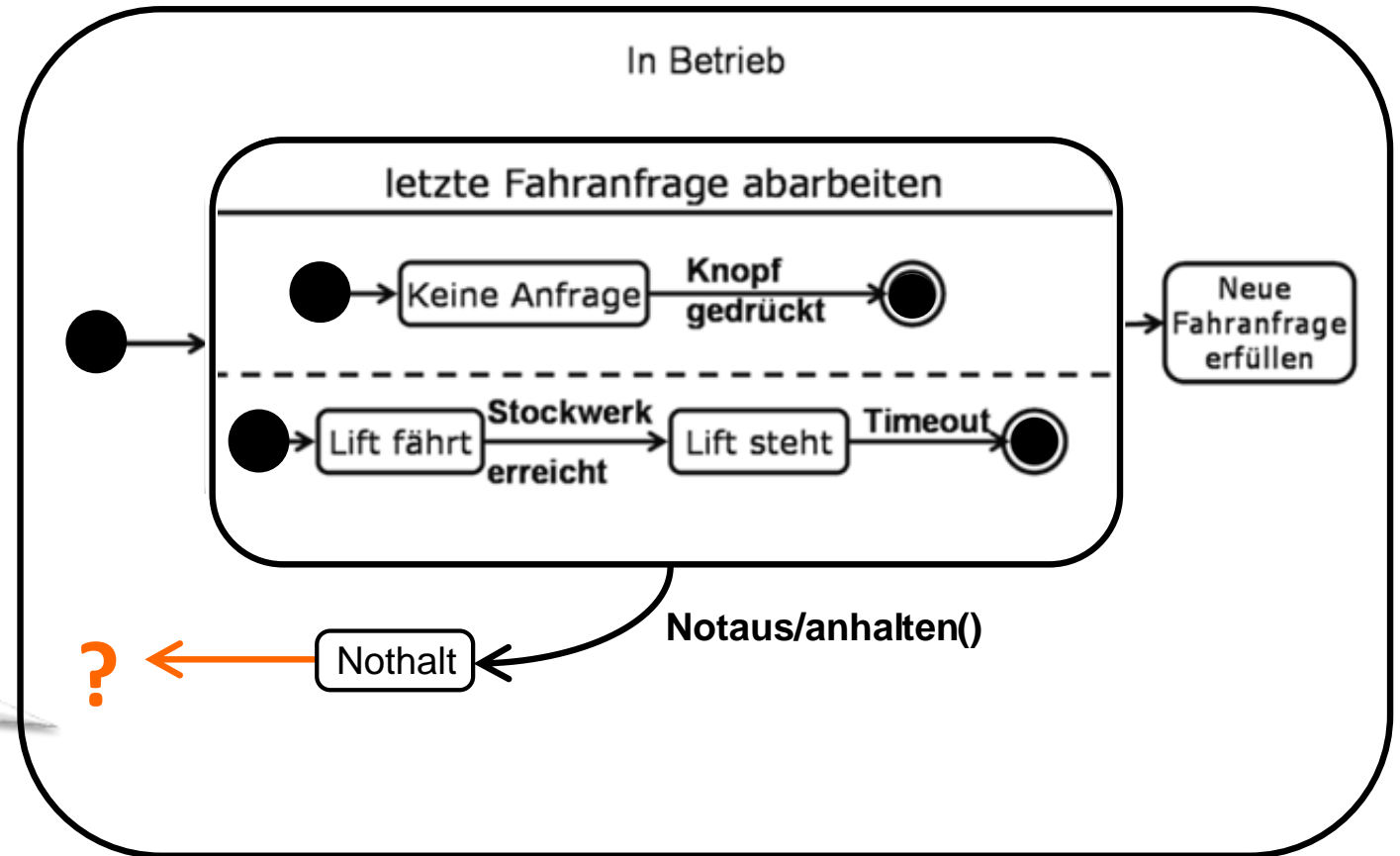


Historienzustand

Beispiel: Lift



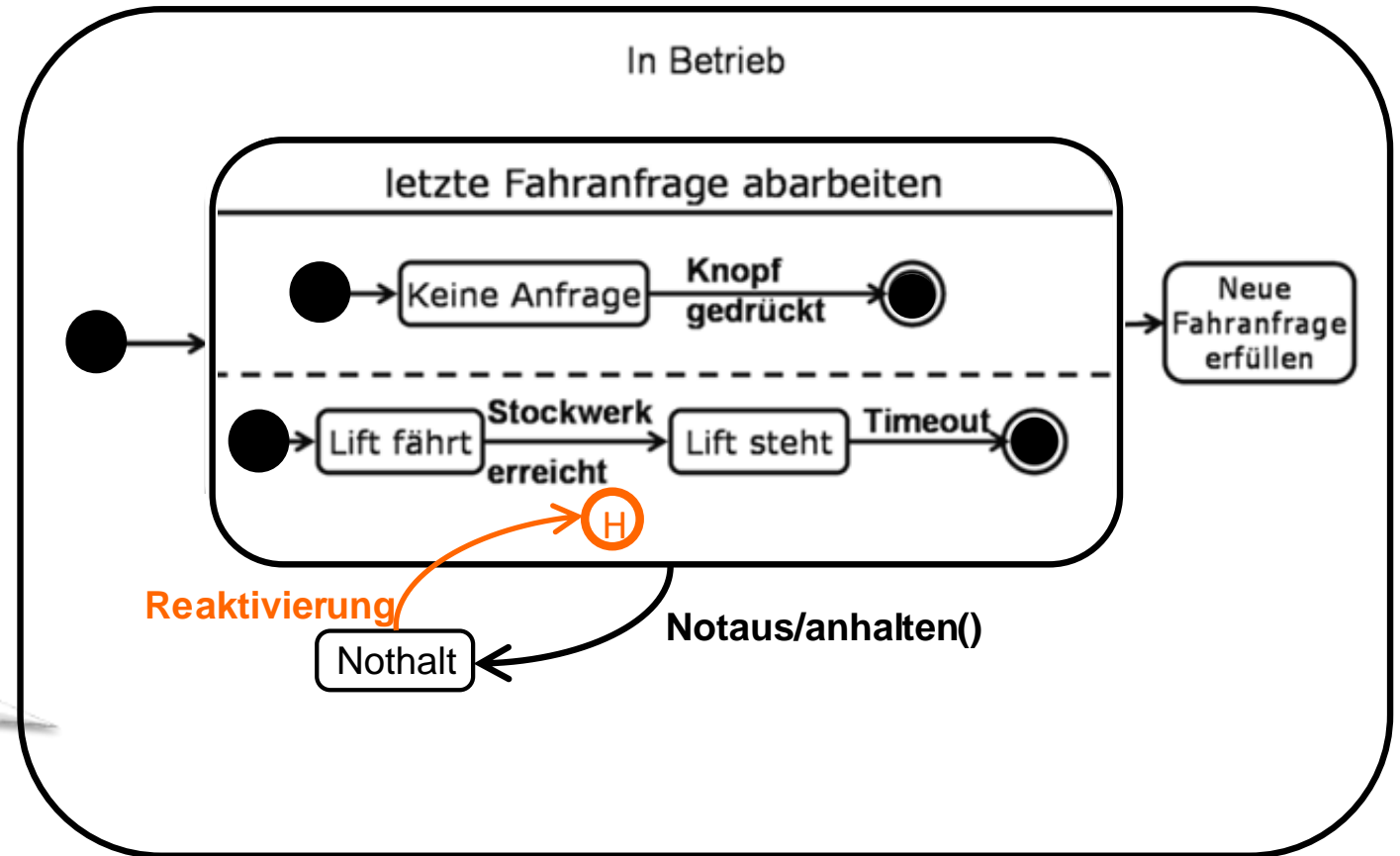
Wie lässt modellieren,
dass der Lift nach der
Unterbrechung (Notaus)
seine **Aktivität wieder
aufnimmt**?



Beispiel: Lift



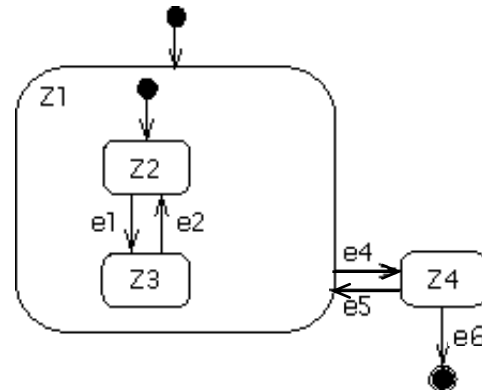
Wie lässt modellieren,
dass der Lift nach der
Unterbrechung (Notaus)
seine **Aktivität wieder
aufnimmt**?



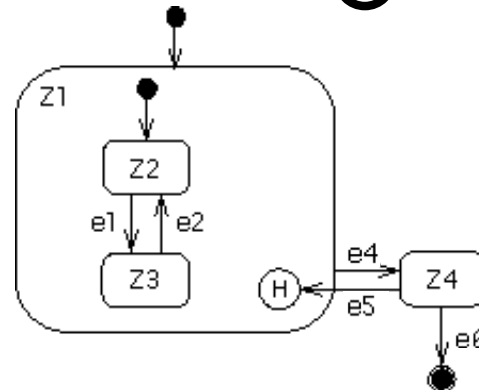
Historienzustand

- Mithilfe von Historienzuständen **kann sich ein Zustandsautomat merken**, welcher Unterzustand zuletzt eingenommen wurde
- **Spezieller Anfangszustand** in einem zusammengesetzten Zustand
- „**Gedächtnis**“, welcher Unterzustand zuletzt eingenommen wurde
- Bei Wiedereintritt in zusammengesetzten Zustand automatisch Übergang in letzten Unterzustand

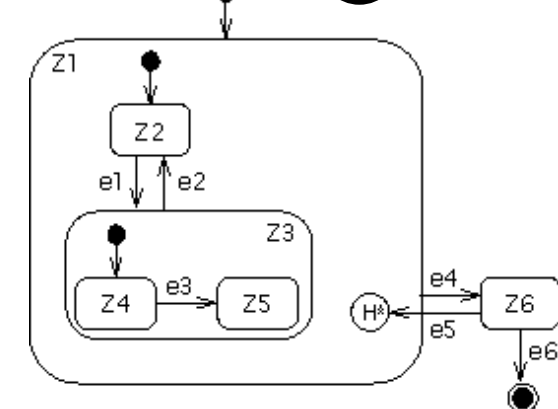
ohne Historie



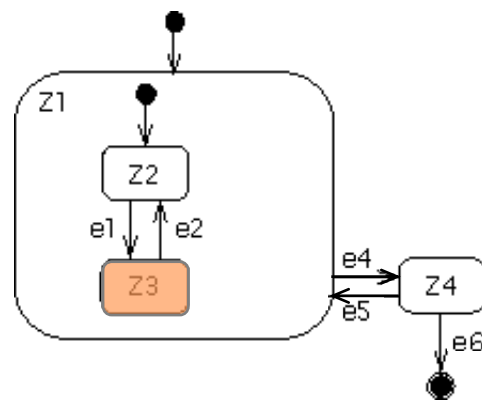
flache Historie (H)



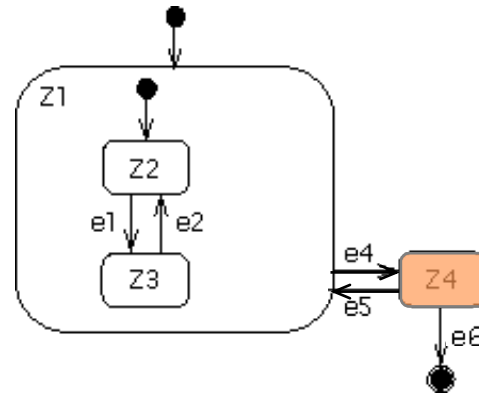
tiefe Historie (H*)



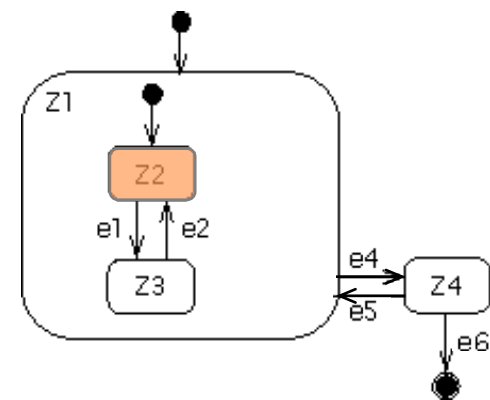
Zustandsautomat – ohne Historie



e1, e4, e5

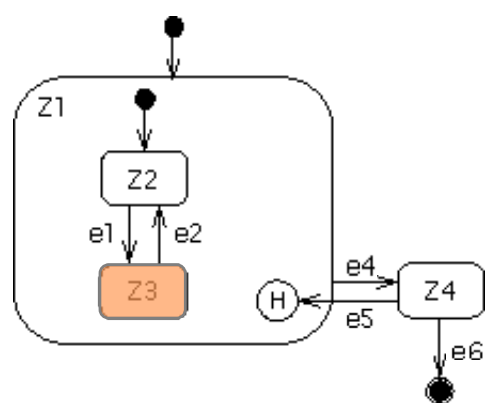


e1, e4, e5

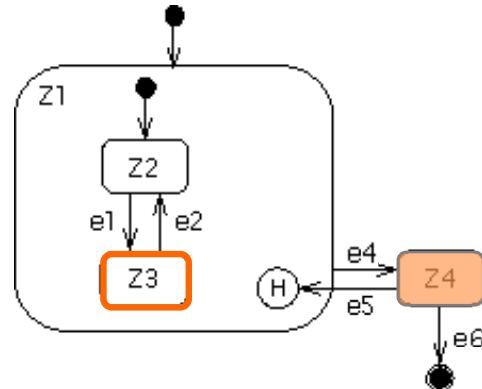


e1, e4, e5

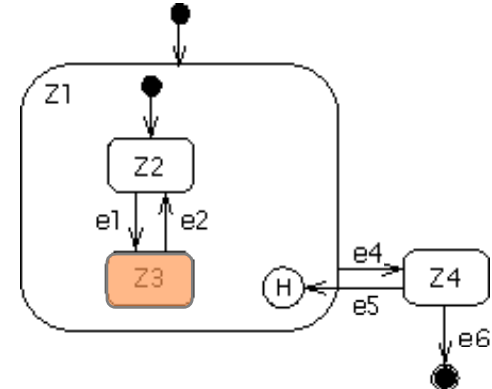
Zustandsautomat – flache Historie



e1, e4, e5

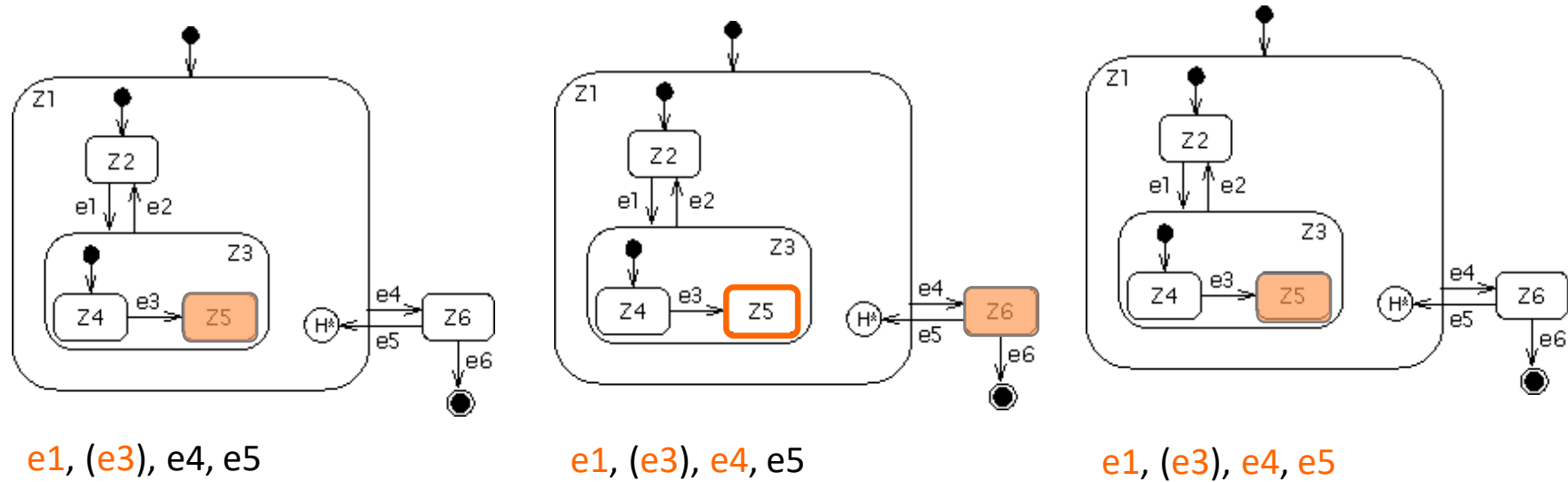


e1, e4, e5



e1, e4, e5

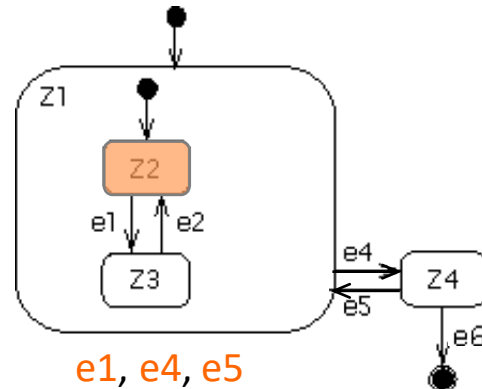
Zustandsautomat – tiefe Historie



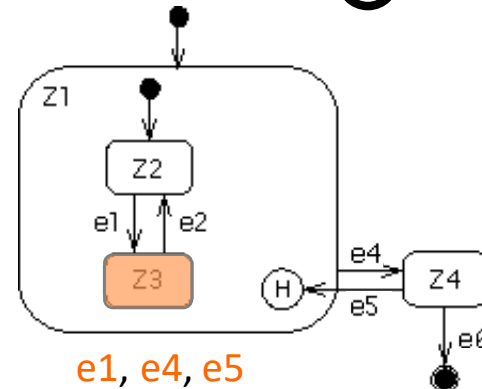
Historienzustand

- Mithilfe von Historienzuständen **kann sich ein Zustandsautomat merken**, welcher Unterzustand zuletzt eingenommen wurde
- **Spezieller Anfangszustand** in einem zusammengesetzten Zustand
- „**Gedächtnis**“, welcher Unterzustand zuletzt eingenommen wurde
- Bei Wiedereintritt in zusammengesetzten Zustand automatisch Übergang in letzten Unterzustand

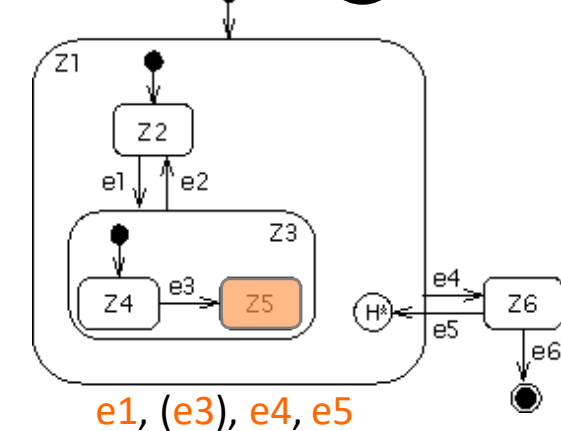
ohne Historie



flache Historie \textcircled{H}



tiefe Historie $\textcircled{H^*}$



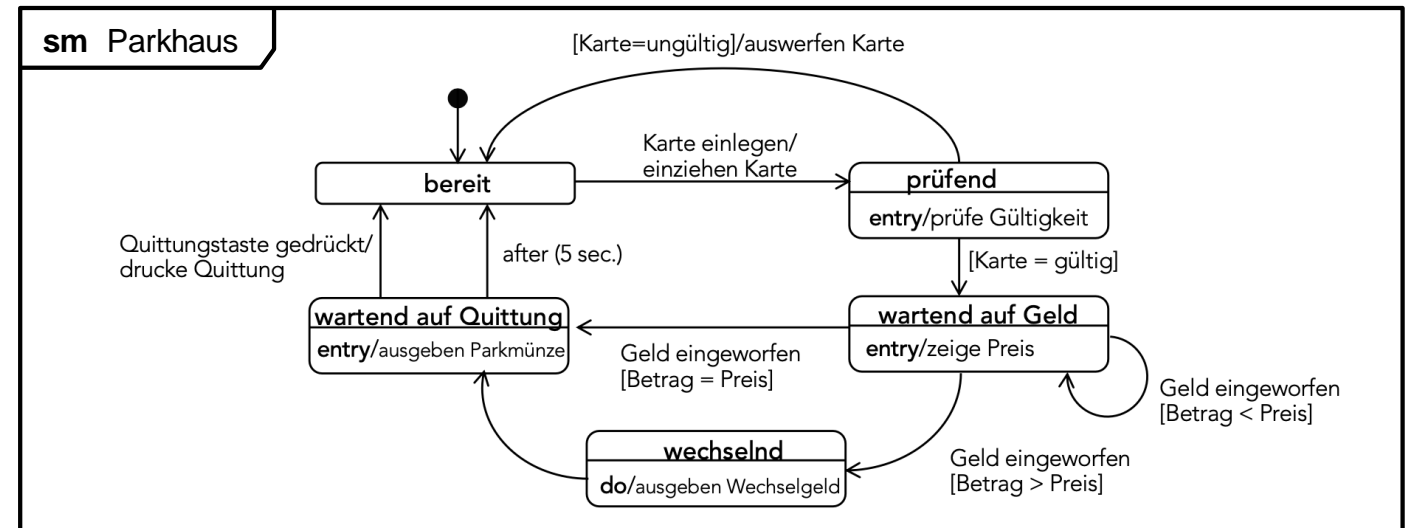
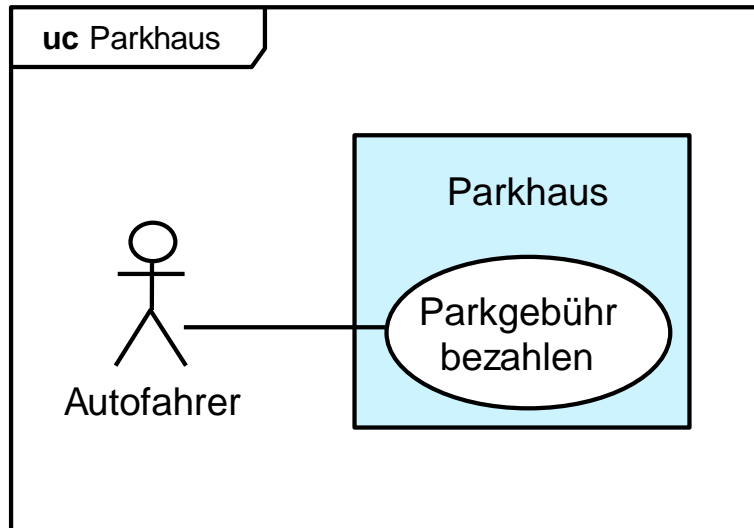
Methodische Vorgehensweise

Konsistenzregeln

- **Geeigneter Zustandsname**
Beschreibt eine bestimmte Zeitspanne und enthält kein Verb
- **Operationsnamen** werden in der Form **Operation()** bei Aktivitäten eingetragen.
- Wenn **eine Operation** in mehreren Zuständen aktiviert werden kann, so kann sie in Abhängigkeit vom jeweiligen Zustand eine **unterschiedliche Wirkung** besitzen.
- Erhält ein Objekt in einem Zustand **einen Operationsaufruf als Ereignis**, wobei diese **Operation nicht als Aktivität zur Verfügung** steht, dann besitzt die Botschaft **keine Wirkung**, d.h. das Objekt tut nichts.
- Jeder **Zustand muss erreichbar sein**
- Jeder **Zustand** - mit Ausnahme der Endzustände – **muss verlassen werden können**
- **Ereignisse** der Transitionen **müssen eindeutig sein**
- Die spezifizierten Ereignisse müssen alle Ereignisse abdecken, die auftreten können und etwas auslösen

Zustandsautomat und Use Cases

- Use-Case durch einen Zustandsautomat spezifizieren
 - Zustandsautomaten sind bei der Use-Case-Spezifikation eine Alternative zu Aktivitätsdiagrammen
 - Zustandsdiagramme sind zu wählen, wenn der Fokus nicht auf der Verarbeitung, sondern auf den eingenommen Zuständen liegt



Zustandsautomat - Methodische Vorgehensweise

Schritte zum Modellieren von Zustandsautomaten

- Brainstorming
 - Erstellen Sie in einer Brainstorming-Sitzung eine Tabelle mit folgenden Spalten:
 1. Spalte: alle Zustände
 2. Spalte: alle Ereignisse, die extern oder intern auftreten können
 3. Spalte: alle Verarbeitungsschritte, die ausgeführt werden müssen
- Welche Zustände enthält der Automat?
 - Ausgangsbasis ist der Anfangszustand
 - Durch welche Ereignisse wird ein Zustand verlassen?
 - Welche Folgezustände treten auf?
 - Wodurch wird der Zustand definiert (Attributwerte, Objektbeziehungen)?
- Benötigt der Zustandsautomat einen Endzustand?
 - Wird der Endzustand erreicht, endet die Bearbeitung des Zustandsautomaten
 - Beschreibt der Automat den Lebenszyklus, dann kann das Beenden des Zustandsautomaten gleichgesetzt werden mit dem Lebensende des Objekts
 - In einem Endzustand darf keine Verarbeitung durchgeführt werden und er darf keine Ausgabepfeile besitzen

Zustandsautomat - Methodische Vorgehensweise

Schritte zum Modellieren von Zustandsautomaten

- Welche Aktivitäten sind zu modellieren?
 - Ist mit einem Zustandsübergang eine Verarbeitung verbunden?
 - Besitzen alle eingehenden Transitionen eines Zustands die gleiche Aktivität? Modellieren Sie eine entry-Aktivität.
 - Besitzen alle ausgehenden Transitionen eines Zustands die gleiche Aktivität? Modellieren Sie eine exit-Aktivität.
 - Ist eine Verarbeitung an die Dauer des Zustands gekoppelt? Modellieren als do-Aktivität.
- Welche Ereignisse sind zu modellieren?
 - Externe Ereignisse: vom Benutzer, von anderen Objekten
 - Zeitliche Ereignisse: Zeitdauer, Zeitpunkt
 - Intern generierte Ereignisse des Use-Case

