

ingo

CHRISTIAN HUEMER

MARION SCHOLZ

Objektorientierte Modellierung mit UML
Teil III - Zustandsdiagramm

Zustandsdiagramm

Das Zustandsdiagramm

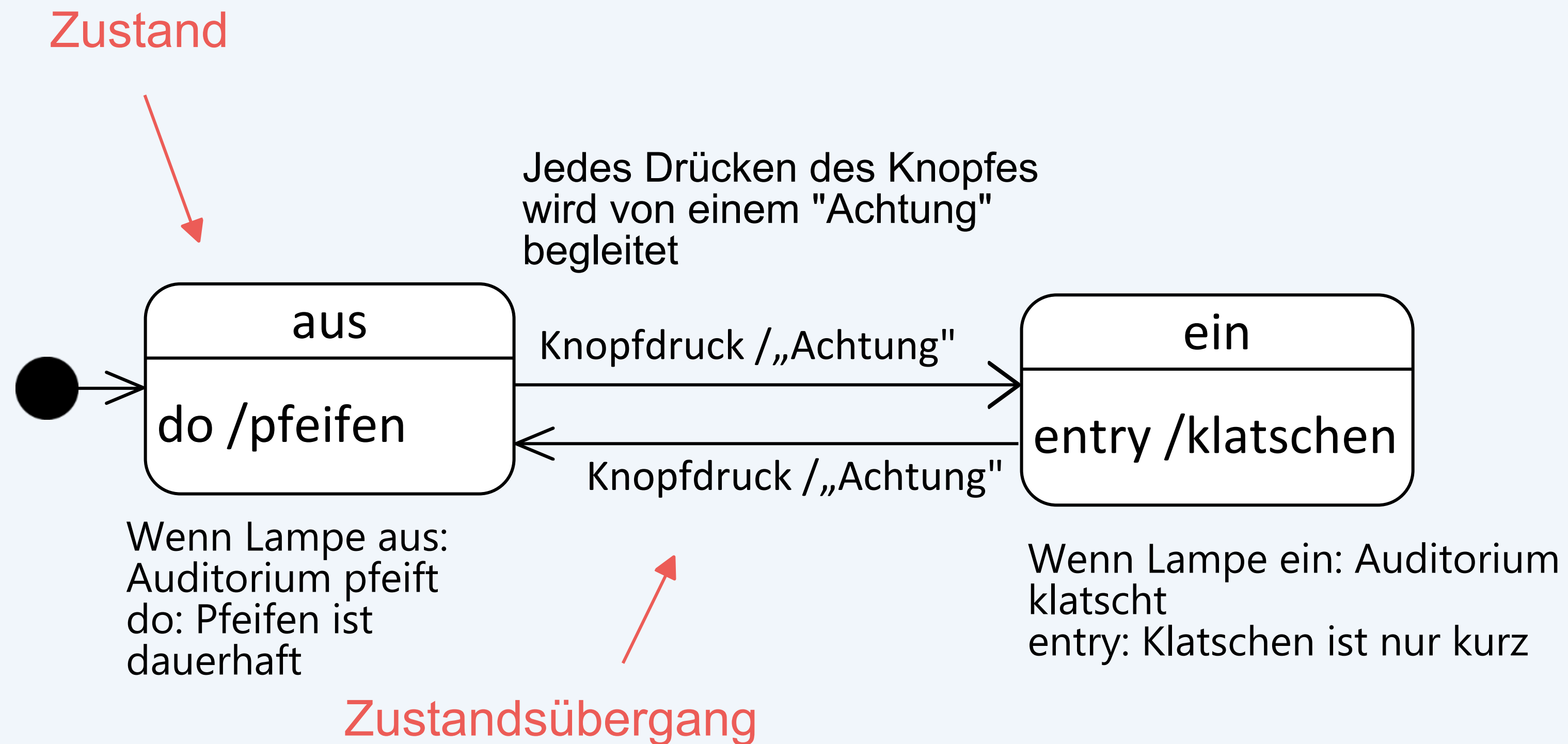


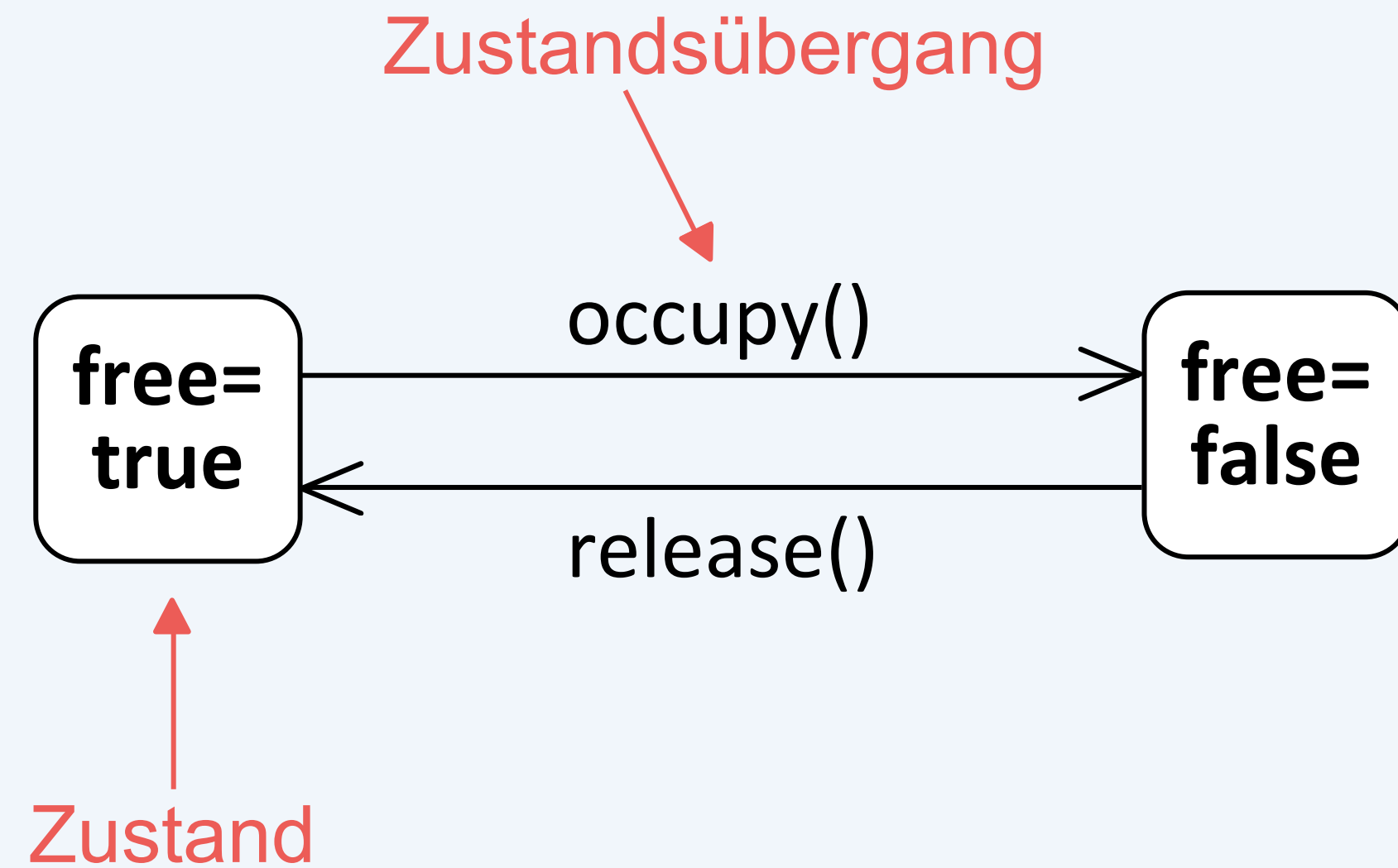
Christian Huemer und Marion Scholz

- Ein Zustandsdiagramm beschreibt die möglichen **Folgen von Zuständen** eines **Modell-Elements**, i.A. von Objekten einer bestimmten Klasse
 - Während seines **Lebenslaufs** (vom Erzeugen bis zum Löschen)
 - Während der **Ausführung** einer **Operation** oder **Interaktion**

- Modelliert werden
 - Die **Zustände**, in denen sich die Objekte einer Klasse befinden können
 - Die möglichen **Zustandsübergänge** (Transitionen) von einem Zustand zum anderen
 - Die **Ereignisse**, die Transitionen auslösen
 - **Aktivitäten**, die in Zuständen bzw. im Zuge von Transitionen ausgeführt werden

Beispiel Lampe





LectureHall

– free: boolean

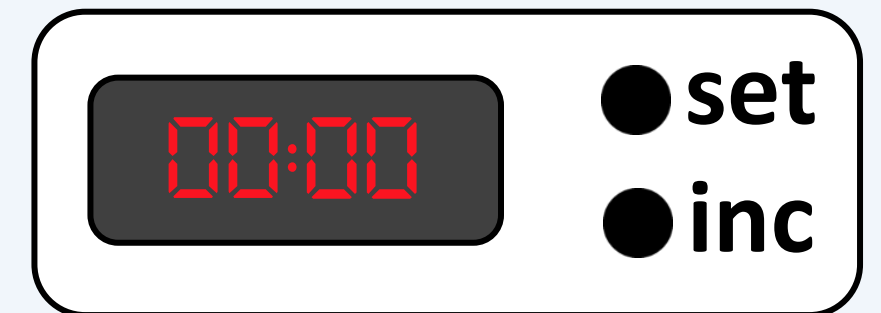
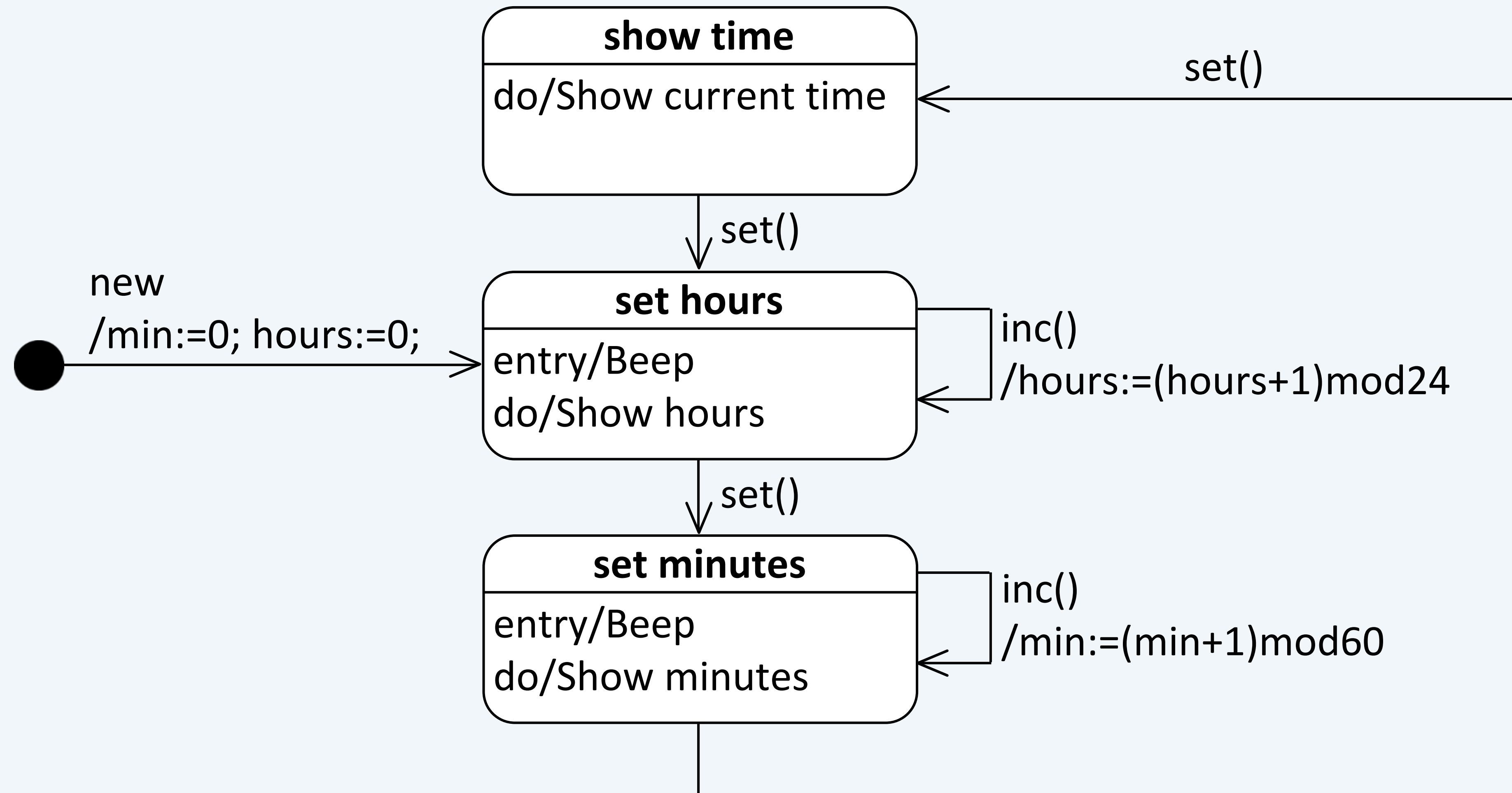
+ occupy()
+ release()

```
class LectureHall {
    private boolean free;

    public void occupy() {
        free=false;
    }
    public void release() {
        free=true;
    }
}
```

Beispiel Digitaluhr

- Modelliert werden sollen die Zustände, die eine Digitaluhr beim Stellen der Uhr einnehmen kann.

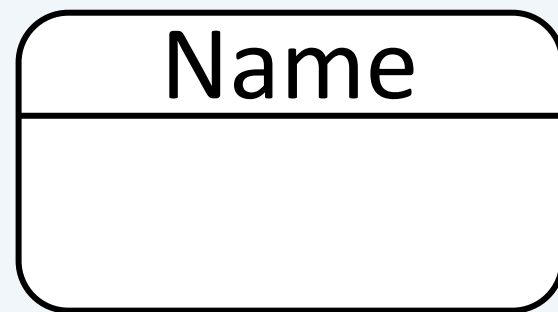


DigitalClock
– min: int – hours: int
+ set(): void + inc(): void

Zustandsdiagramm Der Zustand



Christian Huemer und Marion Scholz



■ Zustand

- System kann sich dauerhaft im Zustand befinden
- „Echter“ Zustand
- Endzustand

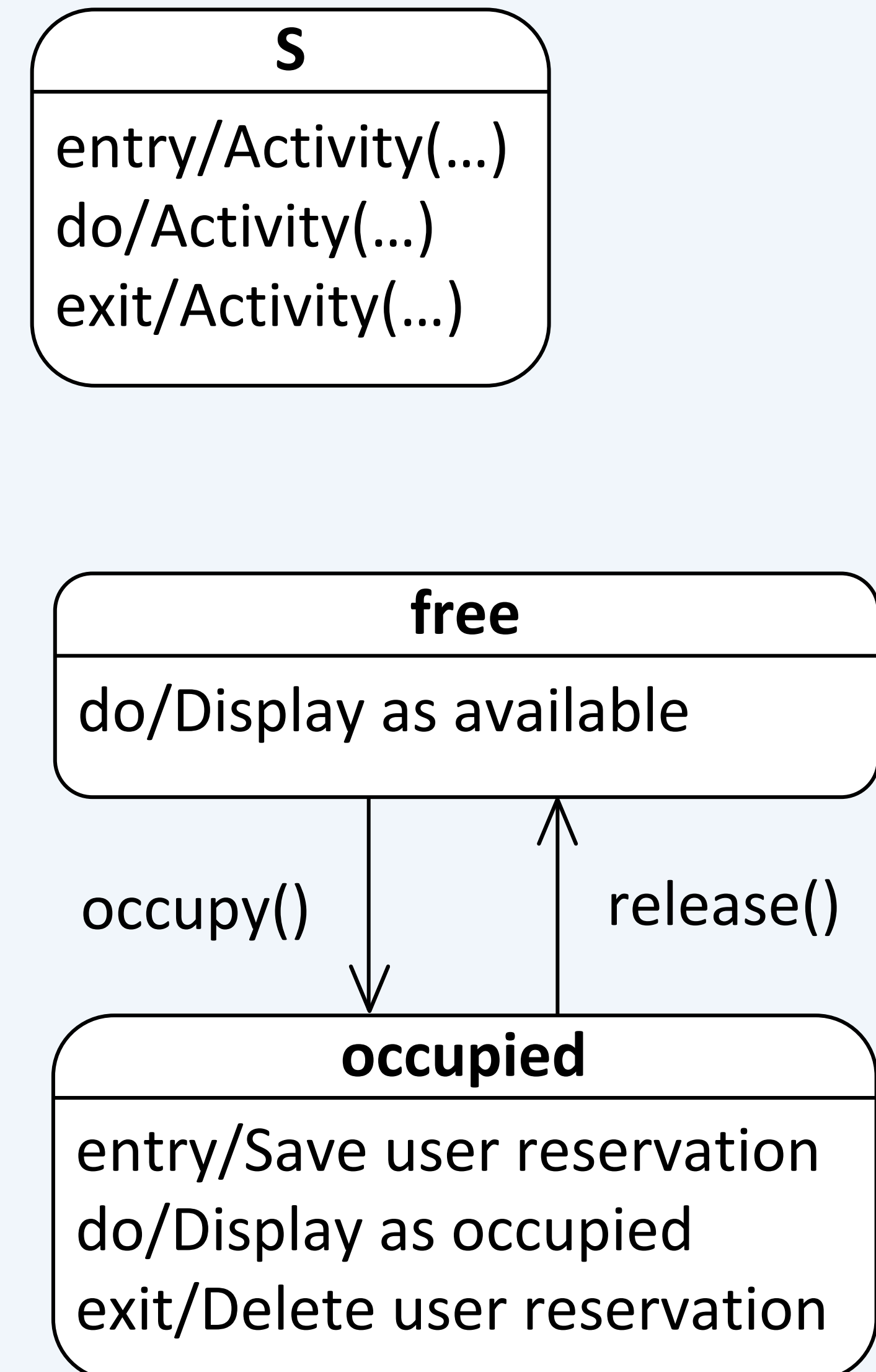
■ Pseudozustand

System kann sich nicht dauerhaft im Zustand befinden

- Startzustand
 - Flacher/Tiefer History-Zustand
 - Parallelisierungsknoten & Synchronisierungsknoten
 - Terminierungsknoten
 - Entscheidungsknoten
- Die zwei wichtigsten
- Von Verwendung wird abgeraten

Aktivitäten innerhalb eines Zustands

- **entry** / Aktivität
 - Wird beim Eingang in den Zustand ausgeführt
- **exit** / Aktivität
 - Wird beim Verlassen des Zustands ausgeführt
- **do** / Aktivität
 - Wird ausgeführt, wenn das System sich in dem Zustand befindet
 - Parameter sind erlaubt
- **event** / Aktivität event Beliebig definierbar
Bsp: Knopfdruck
Aktivität behandelt Ereignis innerhalb des Zustands
 - Wird ausgeführt, wenn sich das System in dem Zustand befindet und das Ereignis eintritt



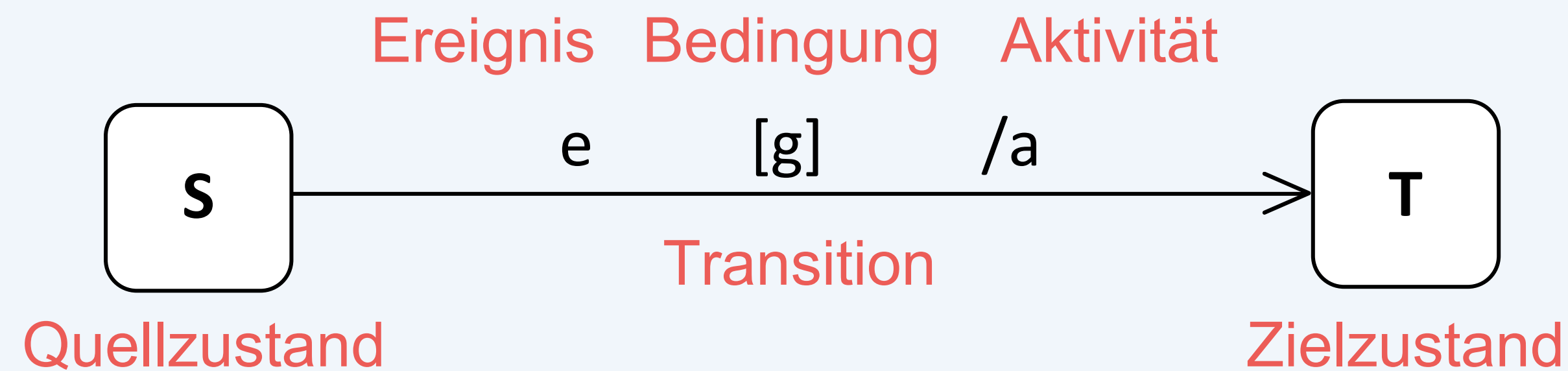
Zustandsdiagramm Die Zustandsübergänge



Christian Huemer und Marion Scholz

Zustandsübergang (Transition) (1/2)

- Übergang vom Quellzustand in den Zielzustand
- Erfolgt, wenn das Ereignis eintritt und die Bedingung (sofern vorhanden) erfüllt ist



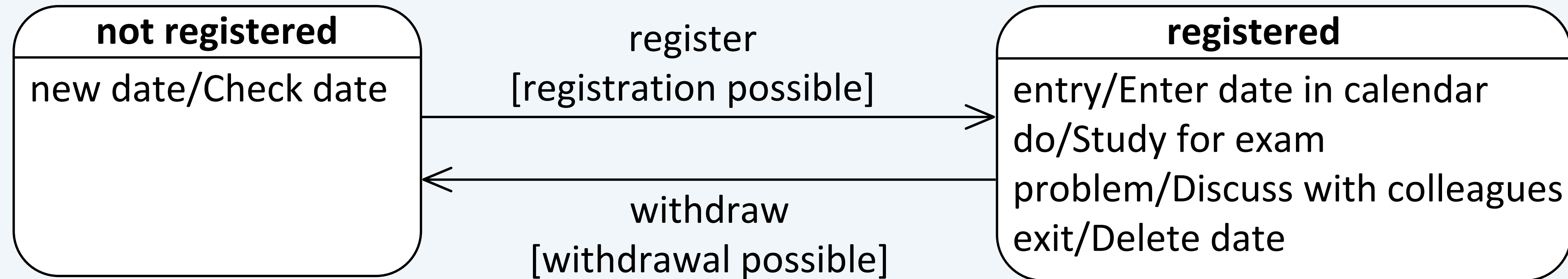
Mit Bedingungen können mehrere Transitionen für das selbe Ereignis festgelegt werden.
Bedingungen müssen mutually exclusive sein

Zustandsübergang (Transition) (2/2)

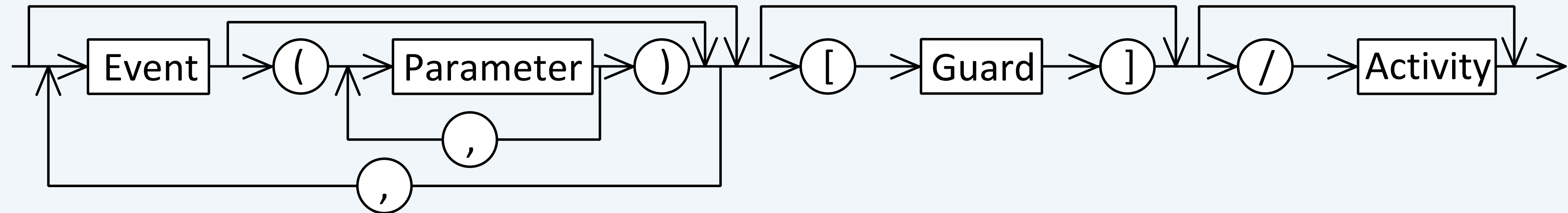


- Ereignis (Event, Trigger)
 - Externer Stimulus
 - Kann Zustandsübergang auslösen
- Bedingung (Guard)
 - Boolescher Ausdruck
 - Wird geprüft, wenn das zugehörige Ereignis eintritt
 - Bedingung wahr: Aktivitäten im aktuellen Zustand werden abgebrochen, exit-Aktivitäten werden ausgeführt und der Zustandsübergang findet statt
 - Bedingung falsch: System bleibt im aktuellen Zustand, das Ereignis geht verloren
- Aktivität (Effect)
 - Wird während der Transition ausgeführt
 - Kann eine Reihe von Aktionen beinhalten

Beispiel Anmeldestatus bei einer Prüfung



Syntax von Transitionen



- Die Aktivität kann aus mehreren Aktionen bestehen

- Bsp.:

Ereignis

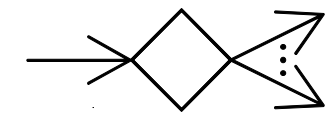
Bedingung

```
right-mouse-button-down (loc) [loc in window]  
/ obj := pick-obj (loc); send obj.highlight()
```

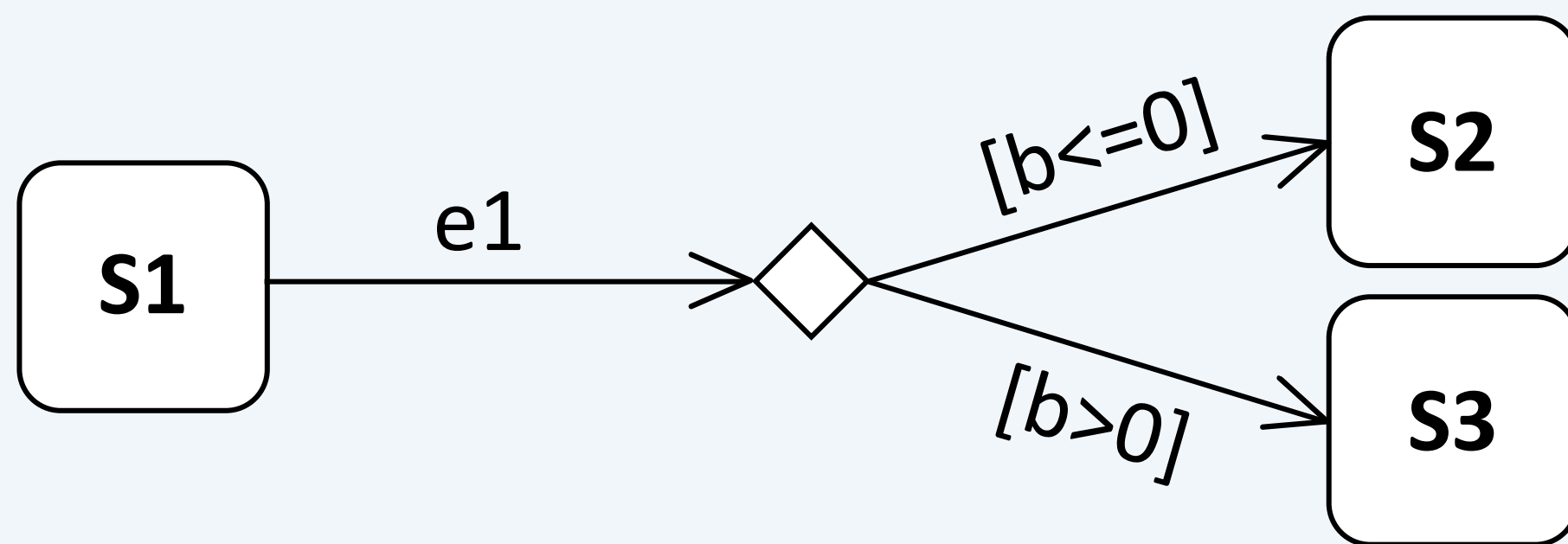
Aktion 1

Aktion 2

Modellierung mit Entscheidungsknoten (1/2)

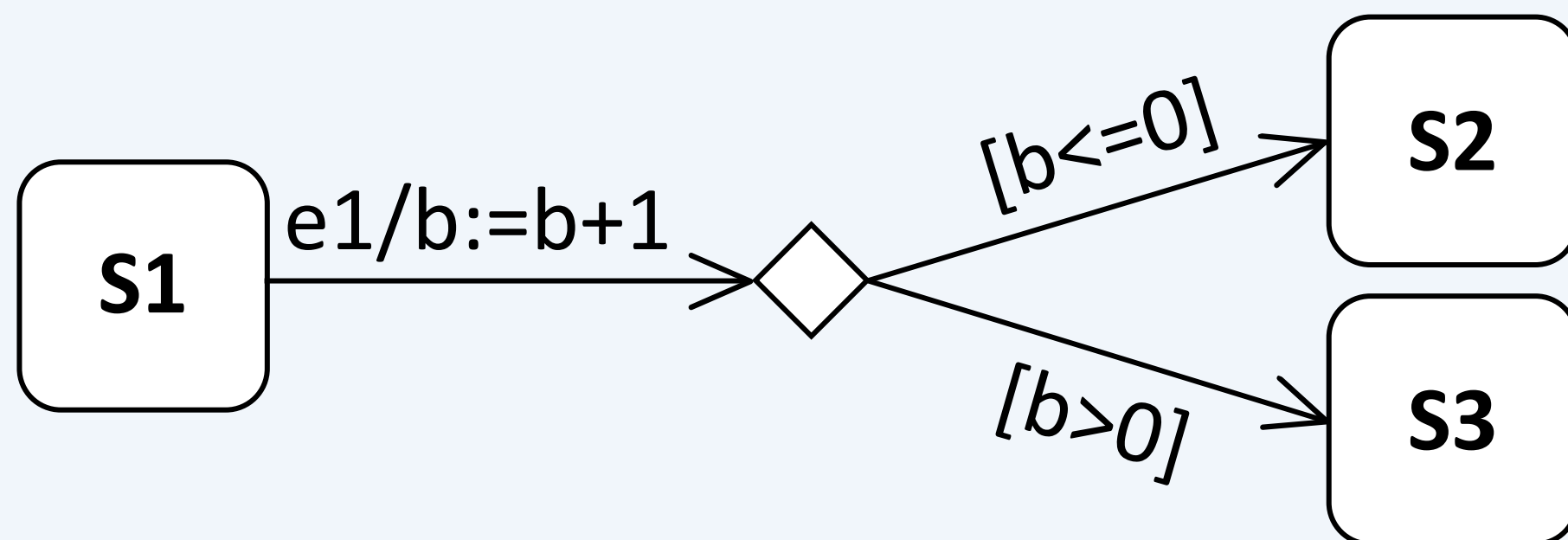
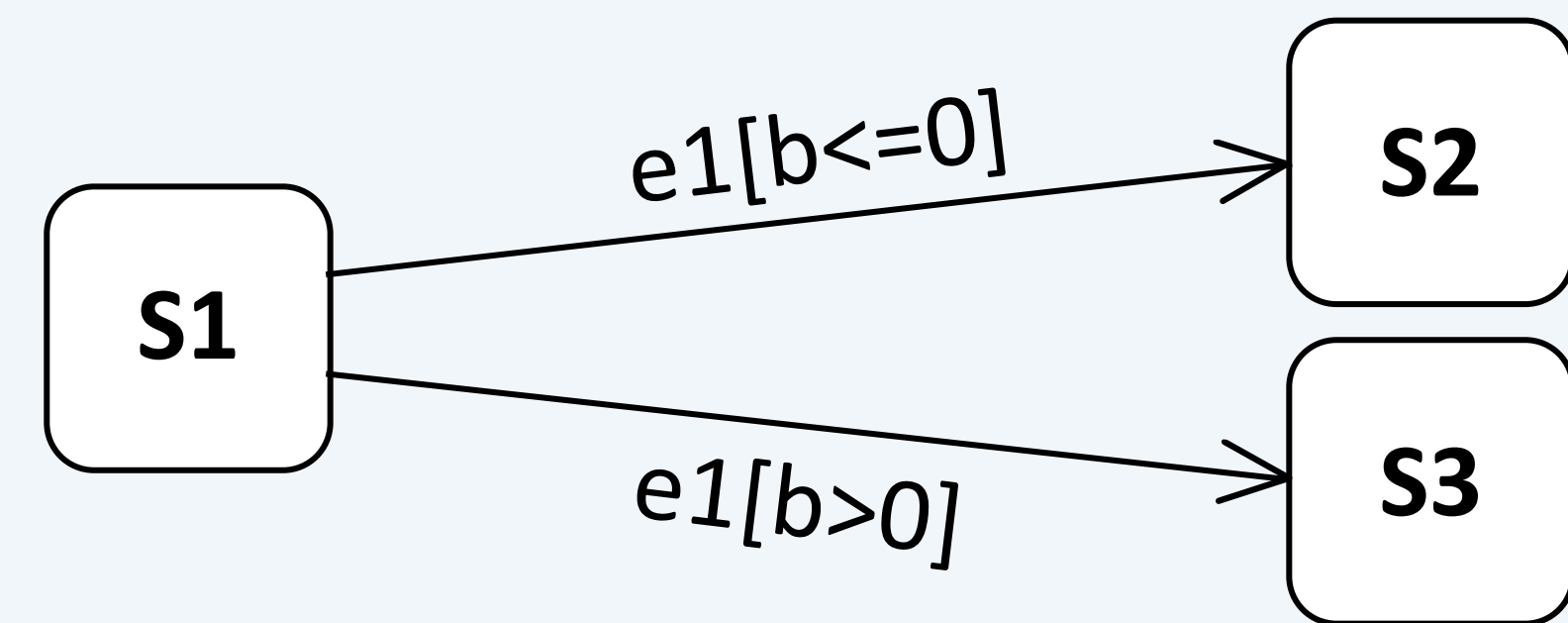


- Pseudozustand
- Kann zur Modellierung alternativer Transitionen benutzt werden



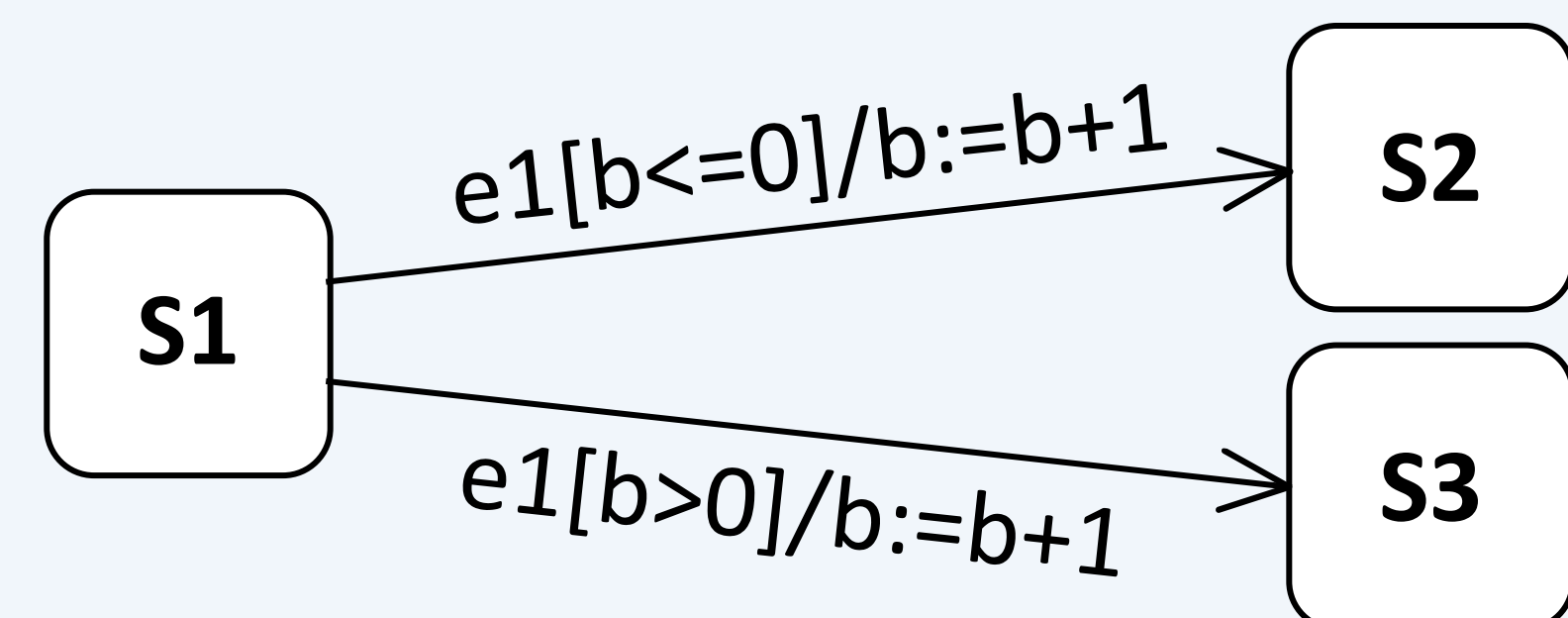
äquivalent?

=



äquivalent?

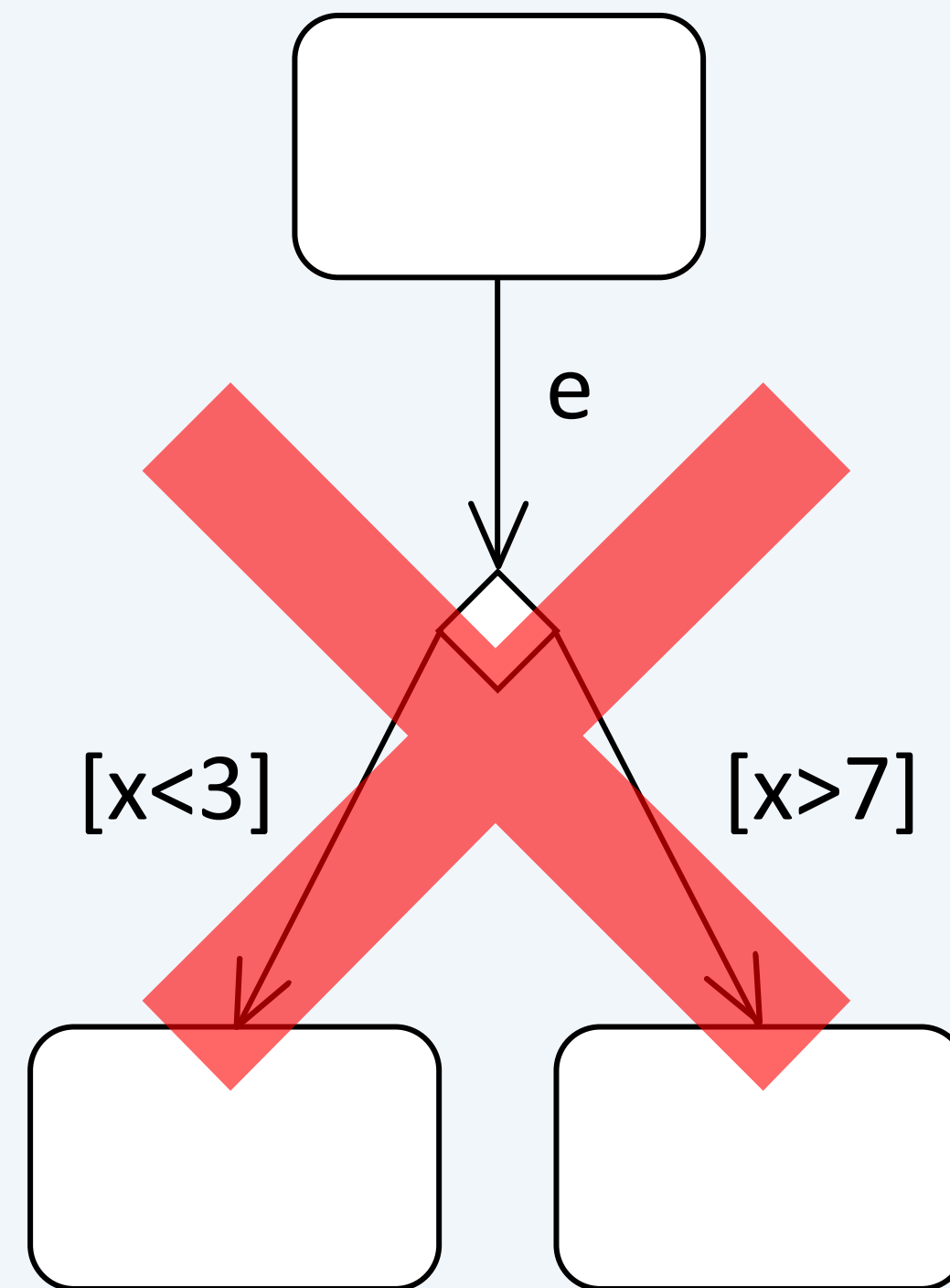
≠



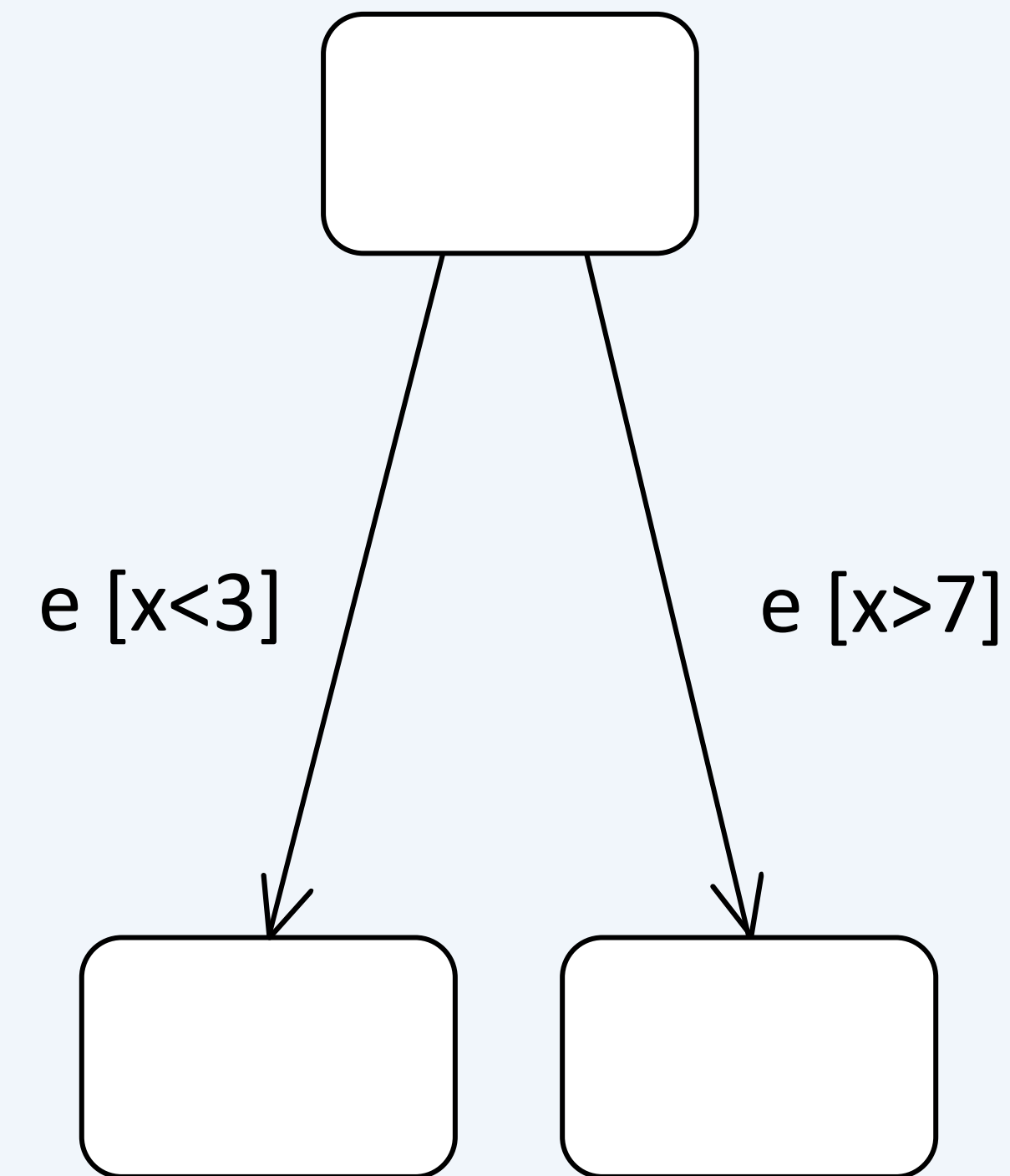
b+=1 wird vor Entscheidung ausgeführt

b+=1 wird nach Guard Check ausgeführt

Modellierung mit Entscheidungsknoten (2/2)



Deckt nicht den gesamten Wertebereich ab

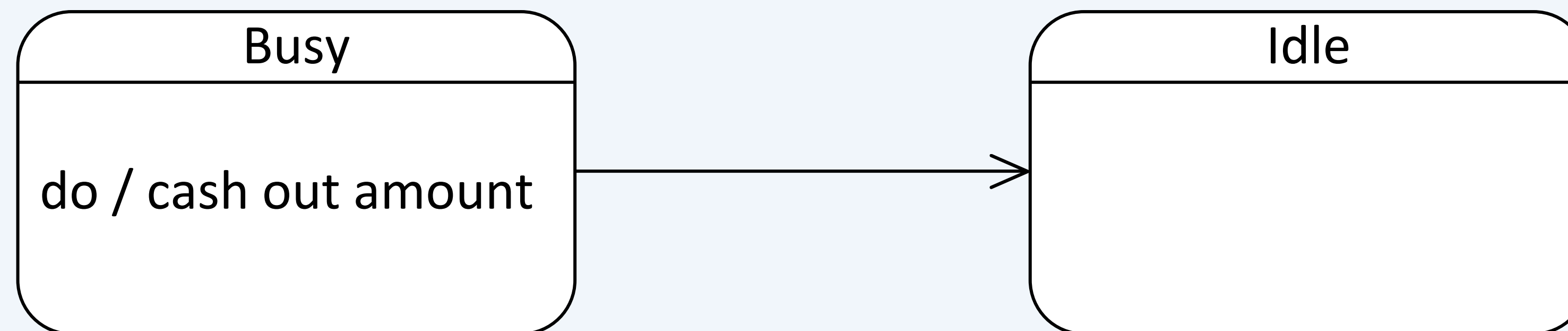


Default-Werte für Zustandsübergänge

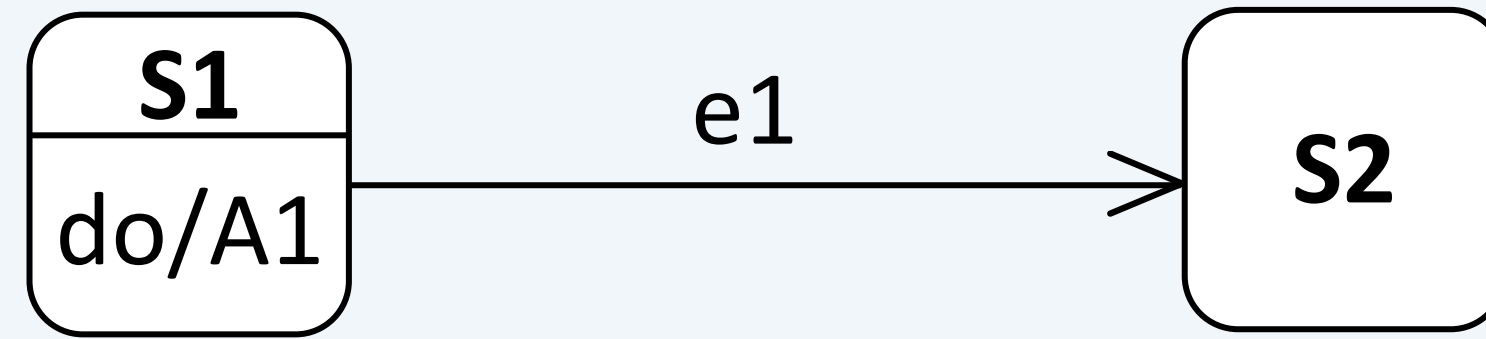
- Default-Werte

- Fehlendes Ereignis entspricht dem Ereignis »Aktivität ist abgeschlossen«
- Fehlende Bedingung entspricht der Bedingung [**true**]

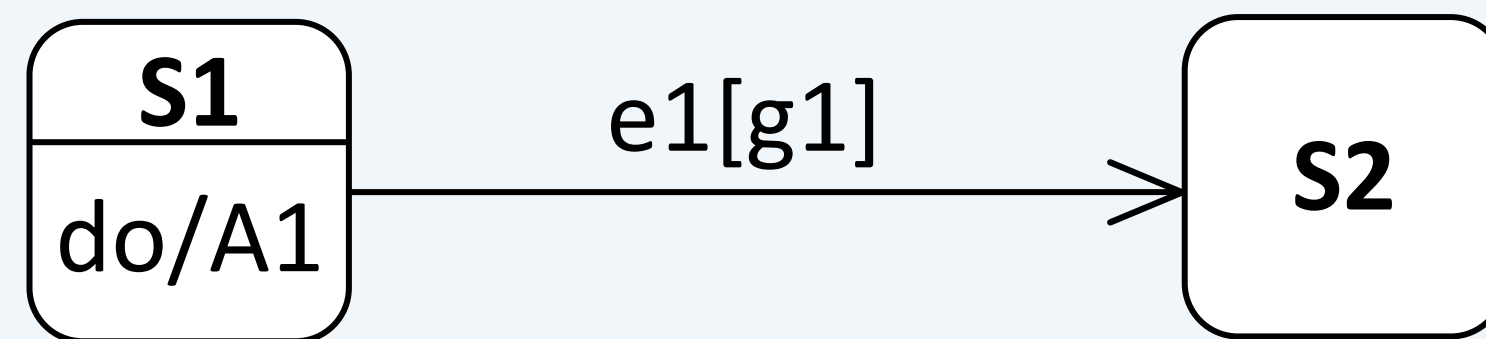
- Beispiel: Bankomat



Arten von Transitionen



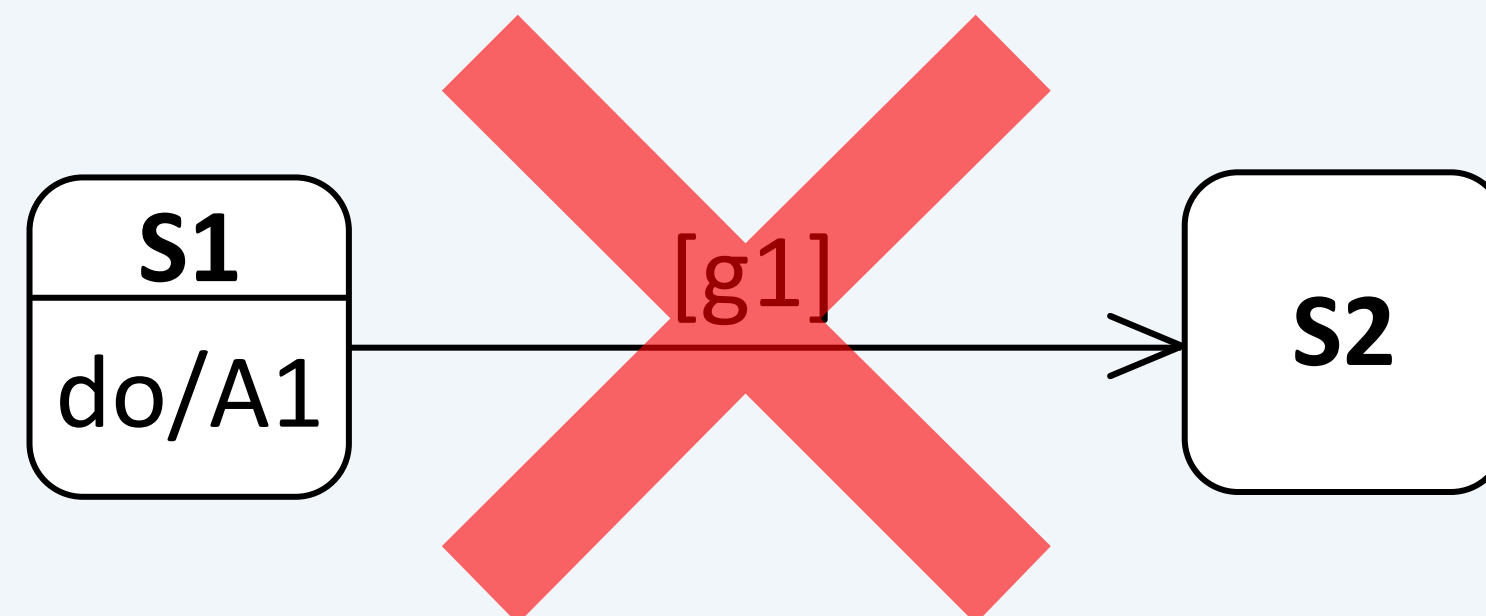
Wenn **e1** eintritt, wird **A1** abgebrochen und das Objekt wechselt in den Zustand **S2**



Wenn **e1** eintritt und **g1** wahr ist, wird **A1** abgebrochen und das Objekt wechselt in den Zustand **S2**



Sobald **A1** beendet ist, wird ein Beendigungsereignis erzeugt das den Übergang zu **S2** auslöst



Sobald **A1** beendet ist, wird ein Beendigungsereignis erzeugt und **g1** wird überprüft. Ist **g1** wahr, erfolgt der Übergang zu **S2**. Ist **g1** falsch, kann der Übergang nie stattfinden

modelliert selten den
gewünschten Sachverhalt

Wenn das wollen Leute meistens Change-Event angeben

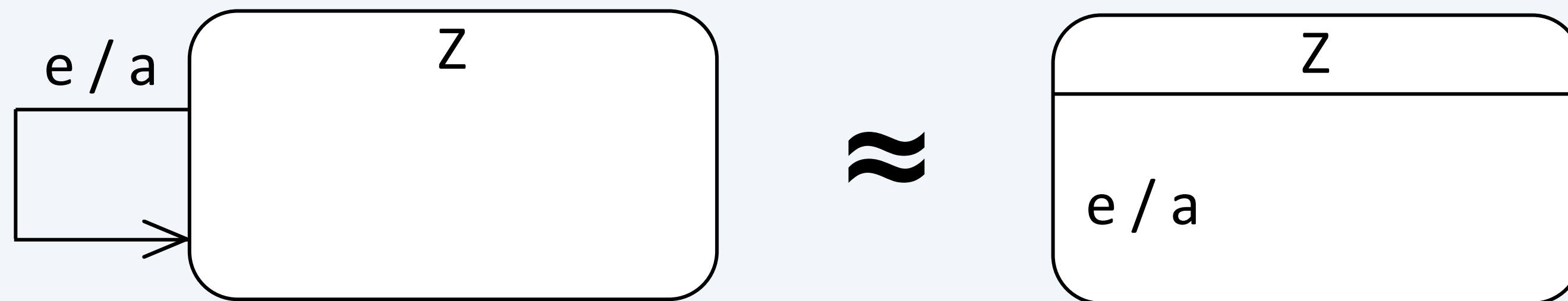
Zustandsdiagramm Die inneren Übergänge



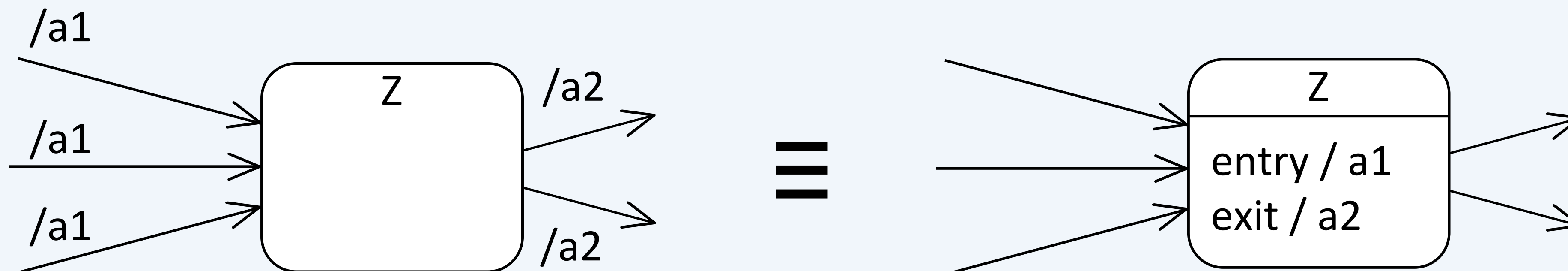
Christian Huemer und Marion Scholz

Zustandsübergang: Innere Transitionen (1/2)

- Werden wie »äußere« Transitionen von Ereignissen ausgelöst, **verlassen** aber den **aktuellen Zustand nicht**
- Äquivalent zu Selbsttransition, sofern keine **entry** / **exit**-Aktivitäten vorhanden
- Gleiche Aktivitäten können in den Zustand hineingezogen werden:

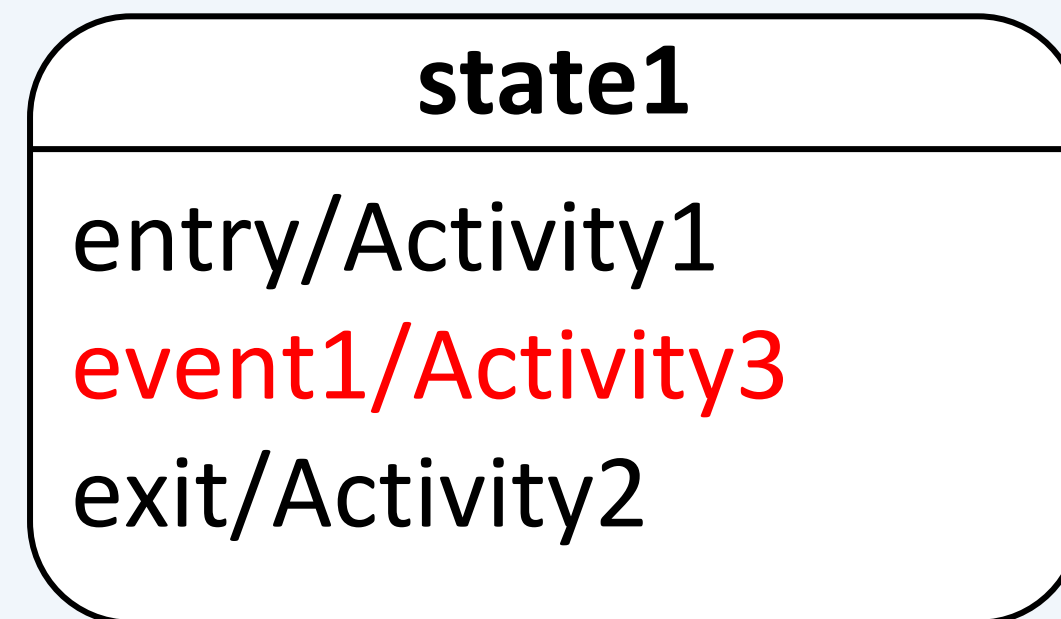


- Gleiche Aktivitäten können in den Zustand hineingezogen werden:



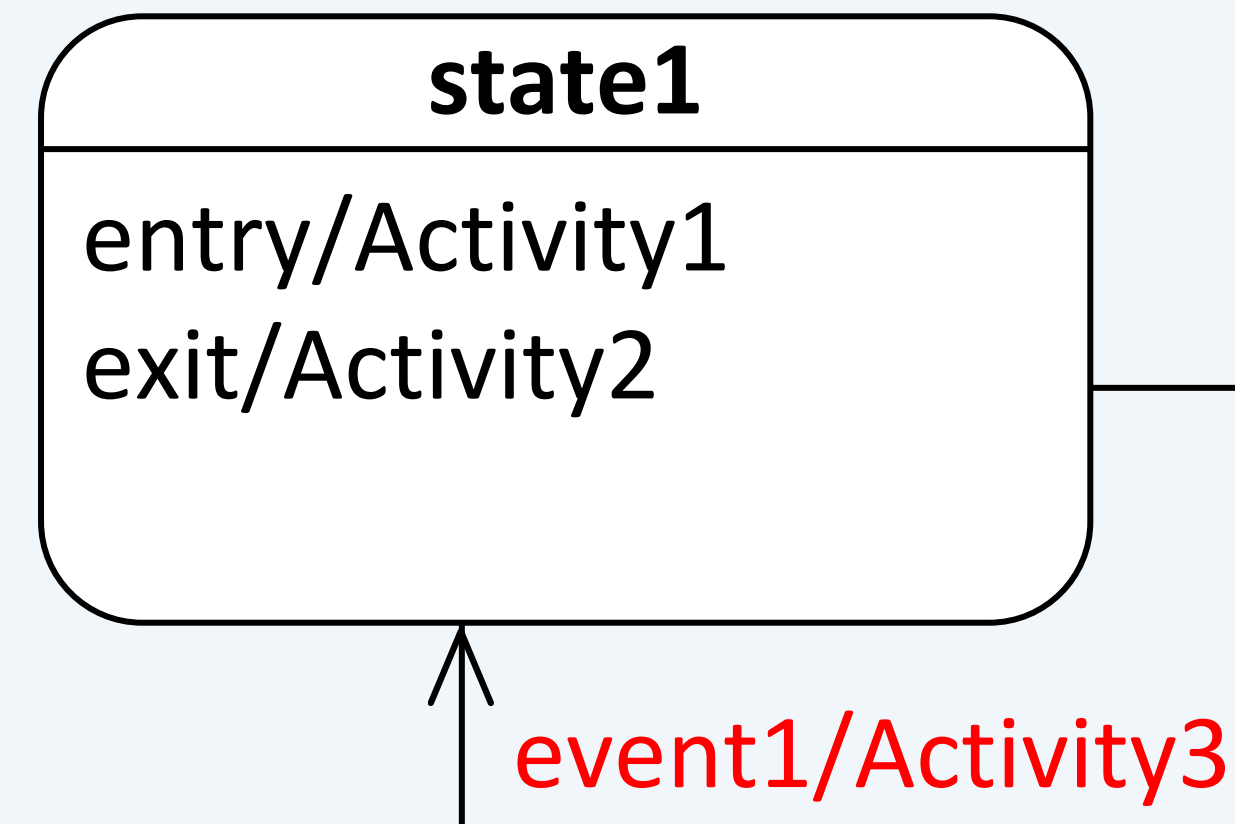
Zustandsübergang: Innere Transitionen (2/2)

Innere Transition



- Wenn **event1** eintritt
 - Objekt bleibt in **state1**
 - **Activity3** wird ausgeführt

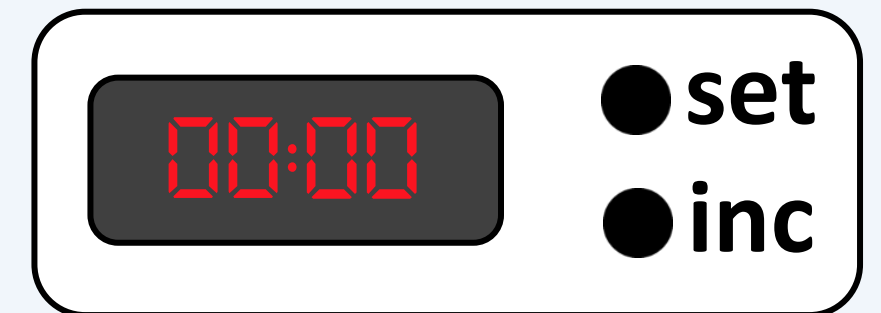
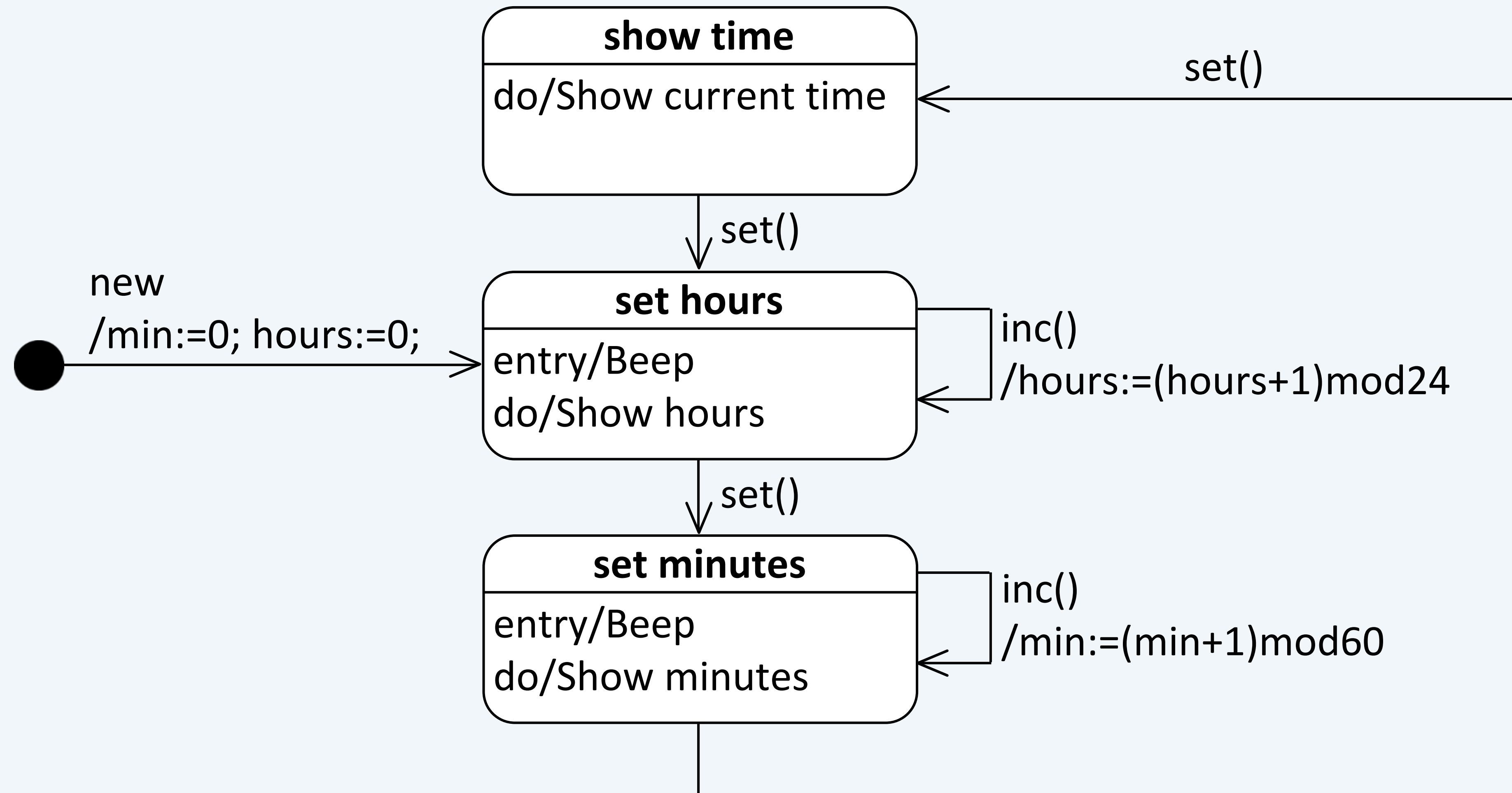
Selbsttransition



- Wenn **event1** eintritt
 - Objekt verlässt **state1** und **Activity2** wird ausgeführt
 - **Activity3** wird ausgeführt
 - Objekt betritt **state1** und **Activity1** wird ausgeführt

Beispiel Digitaluhr

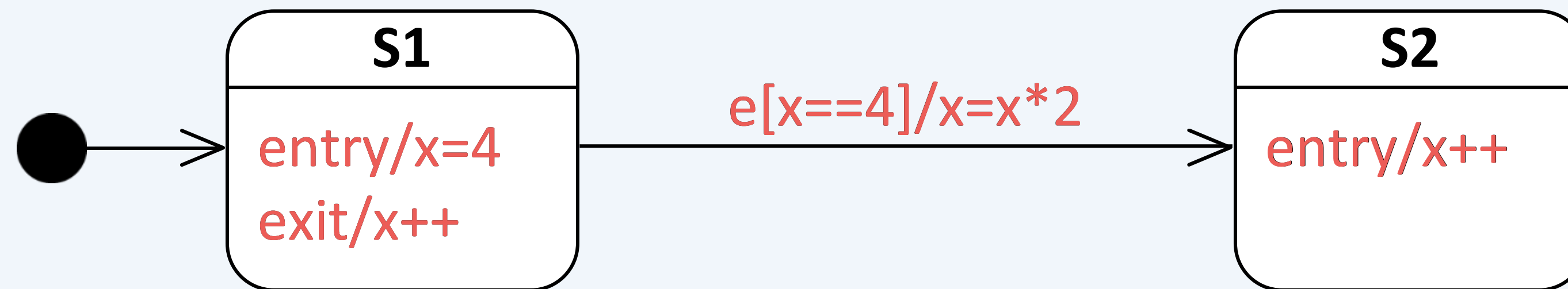
- Modelliert werden sollen die Zustände, die eine Digitaluhr beim Stellen der Uhr einnehmen kann.



DigitalClock
– min: int – hours: int
+ set(): void + inc(): void

Ausführungsreihenfolge von Aktivitäten – Beispiel

- Angenommen, **S1** ist aktiv
... welchen Wert hat **x** nach dem Eintritt von **e** ?



S1 wird aktiv, **x** bekommt den Wert **4**

e tritt ein, die Bedingung wird überprüft und ist wahr

S1 wird verlassen, **x** erhält den Wert **5**

Der Zustandsübergang findet statt, **x** erhält den Wert **10**

S2 wird betreten, **x** wird um **1** erhöht und hat nun den Wert **11**

Zustandsdiagramm Die Ereignisarten



Christian Huemer und Marion Scholz

Zustandsübergang: Ereignistypen (1/2)

- CallEvent
Empfang einer Nachricht (Operationsaufruf)
 - Bsp.: `occupy()`,
`register()`
- SignalEvent
Empfang eines Signals
 - Bsp.: `rightmousedown`, `receiveSMS`
- TimeEvent
 - Relativ: bezogen auf den Zeitpunkt des Eintritts
in den gerade aktiven Zustand
 - Bsp.: `after(5 seconds)`
 - Absolut
 - Bsp.: `when(time==16:00)`,
`when(date==20250101)`

Zustandsübergang: Ereignistypen (2/2)

■ ChangeEvent

Das Erfüllen einer Bedingung wird dauerhaft überwacht

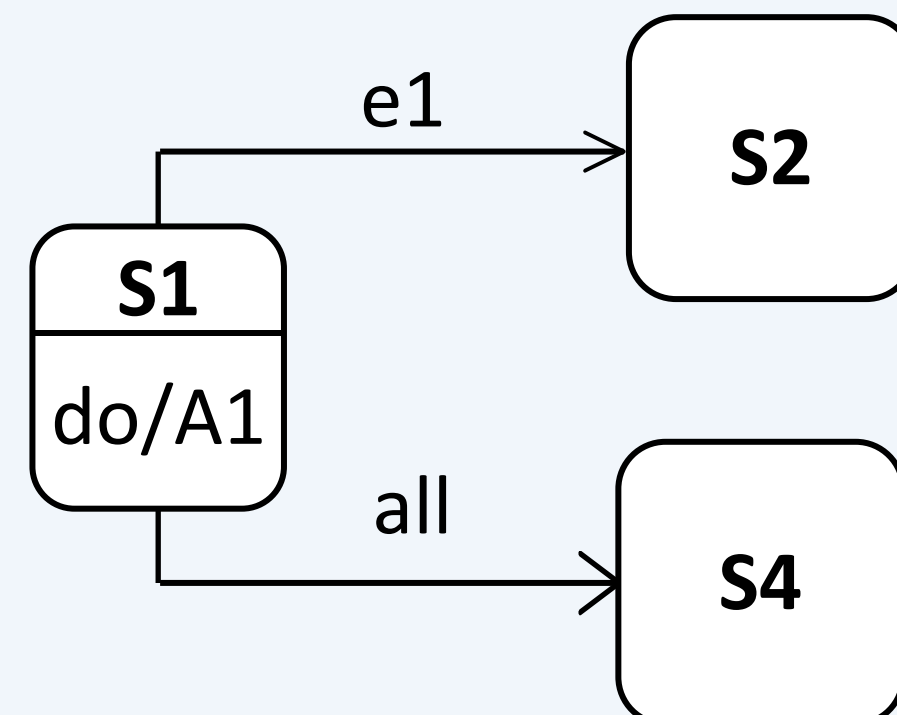
- Bsp.: `when (x > y)`

■ Completion event (Beendigungseignis)

Wird erzeugt, wenn alle do-Aktivitäten des Vorzustands abgeschlossen sind

■ Any receive event

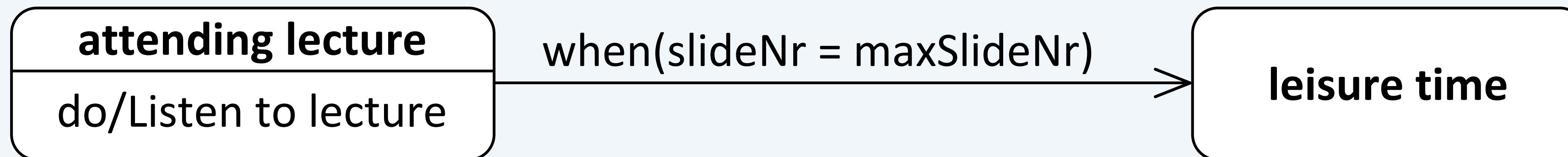
- Tritt ein, wenn irgendein Ereignis eintritt, das beim aktiven Zustand keine andere Transition auslöst
- Eine Art “else-Transition”
- Schlüsselwort **a11**



Unterschied ChangeEvent und Bedingung

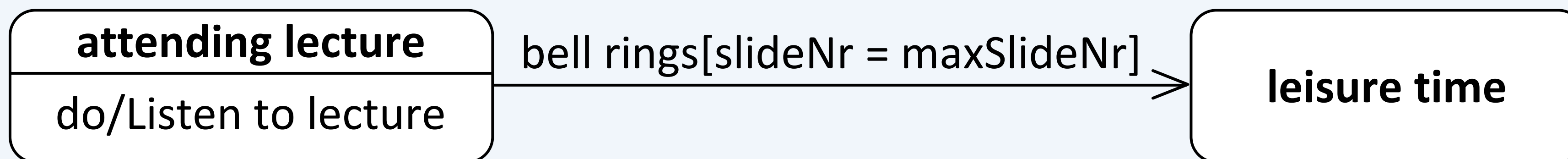
■ ChangeEvent:

- Bedingung wird permanent geprüft
- Wenn Bedingung wahr ist, kann zugehöriger Zustandsübergang ausgelöst werden (falls nicht durch zugehörige Überwachungsbedingung blockiert)



■ Bedingung:

- Wird nur geprüft, wenn zugeordnetes Ereignis eintritt
- Kann selbst keinen Zustandsübergang auslösen



Was passiert, wenn die letzte Folie erreicht ist, bevor die Glocke läutet?

Was passiert, wenn die Glocke läutet, bevor die letzte Folie erreicht ist?

Zustandsdiagramm Der Lebenszyklus eines Objekts



Christian Huemer und Marion Scholz

Start- und Endzustand, Terminierungsknoten

■ Startzustand



- "Beginn" des Zustandsdiagramms
- Genau eine ausgehende Transition
 - Wird sofort ausgelöst, wenn sich das System im Startzustand befindet
 - Keine Bedingungen und Ereignisse (Ausnahme: Ereignis zur Erzeugung des betrachteten Objekts)
 - Angabe von Aktivitäten ist erlaubt

■ Endzustand



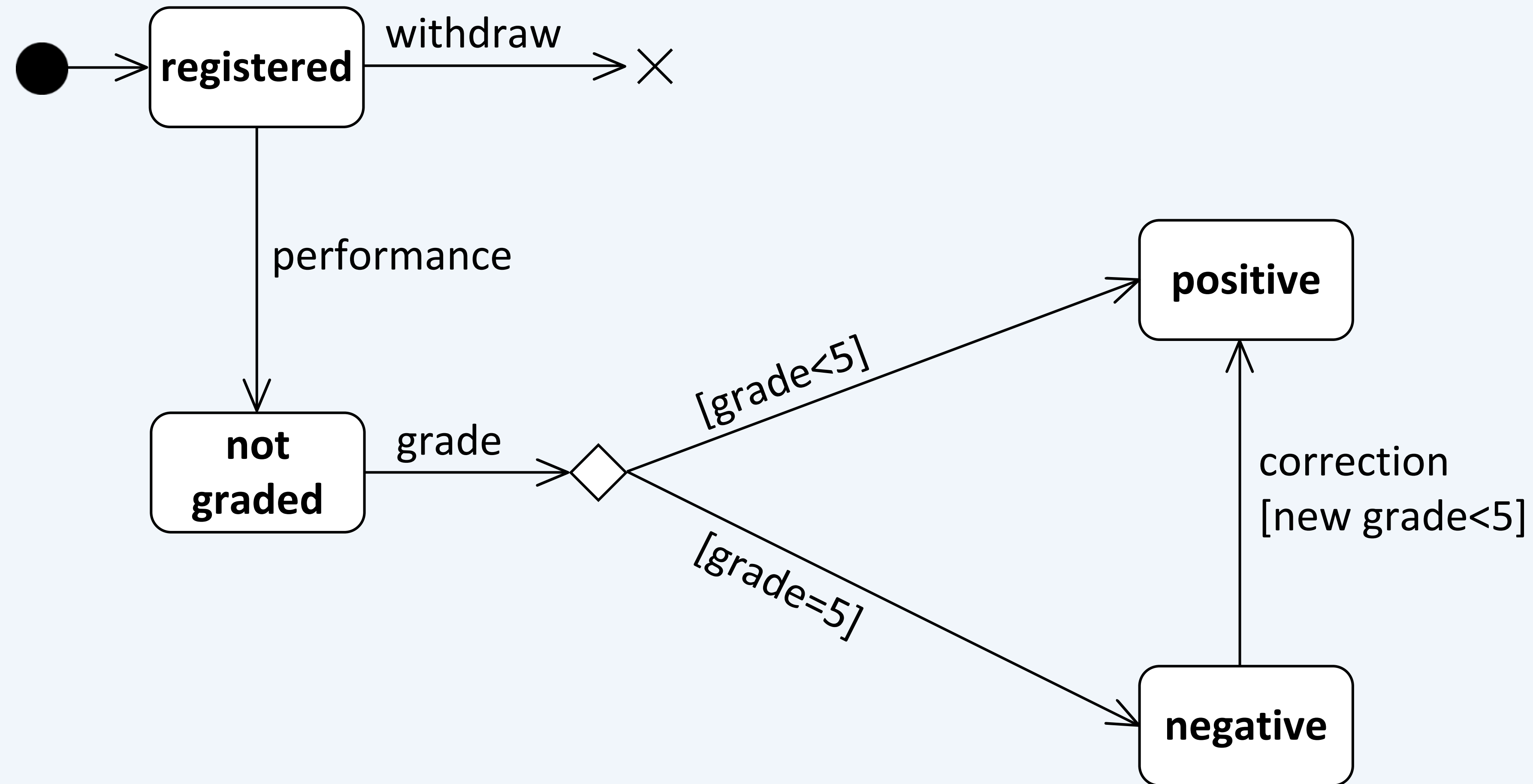
- Keine ausgehenden Transitionen
- Kein Pseudozustand!

■ Terminierungsknoten



- Objekt, dessen Verhalten modelliert wird, hört auf zu existieren

Beispiel Prüfungsantritt



Zustandsdiagramm

Die geschachtelten Zustände

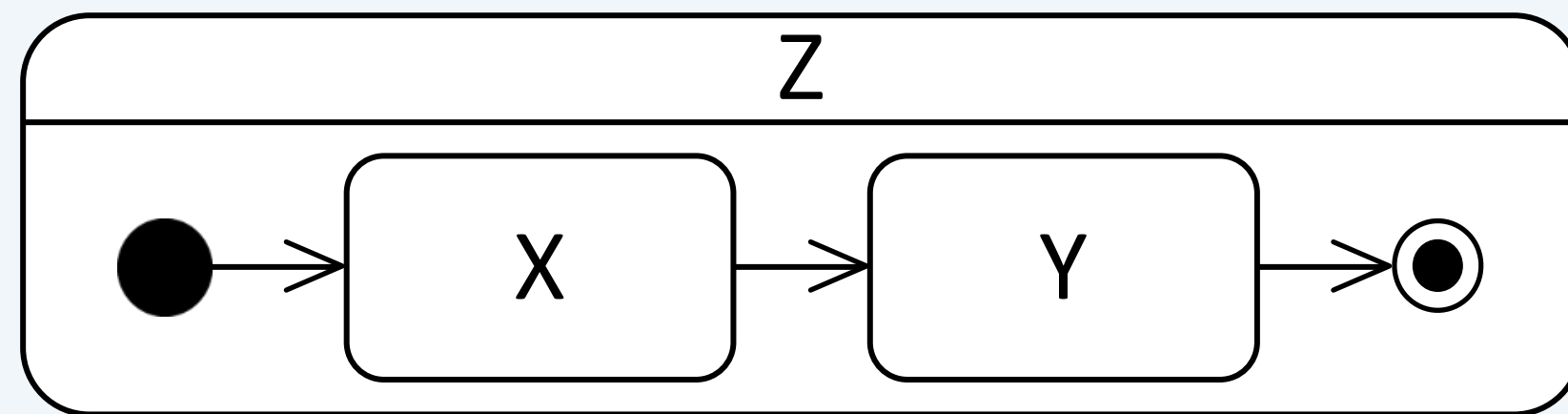


Christian Huemer und Marion Scholz

Komplexe Zustände

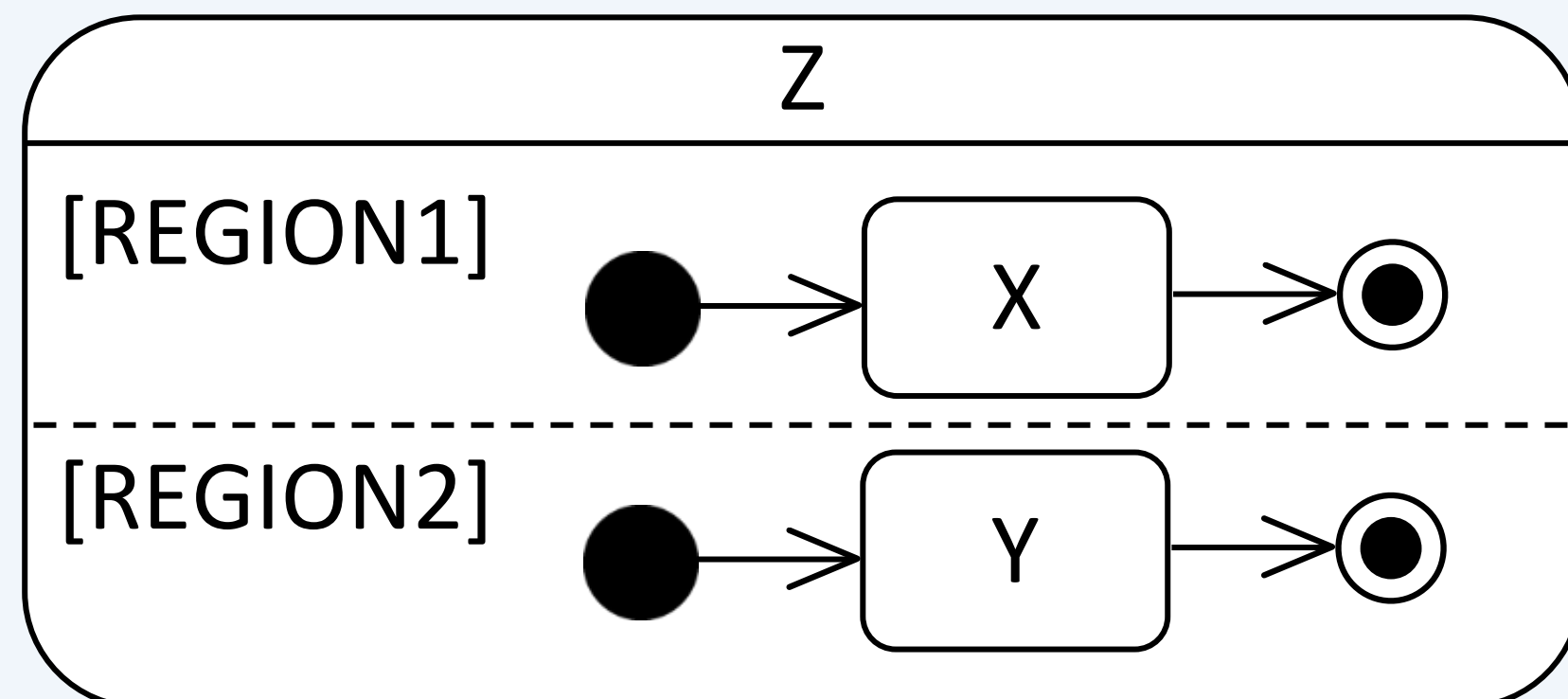
- Sind aus mehreren Subzuständen zusammengesetzt
→ geschachteltes Zustandsdiagramm

- Die Subzustände sind disjunkt



Zu einem Zeitpunkt kann nur **x** ODER **y** aktiv sein!

- Teilung des Superzustandes in mehrere Regionen
 - → die Subzustände sind nebenläufig, gleichzeitig aktiv
 - **z** = „orthogonaler Zustand“



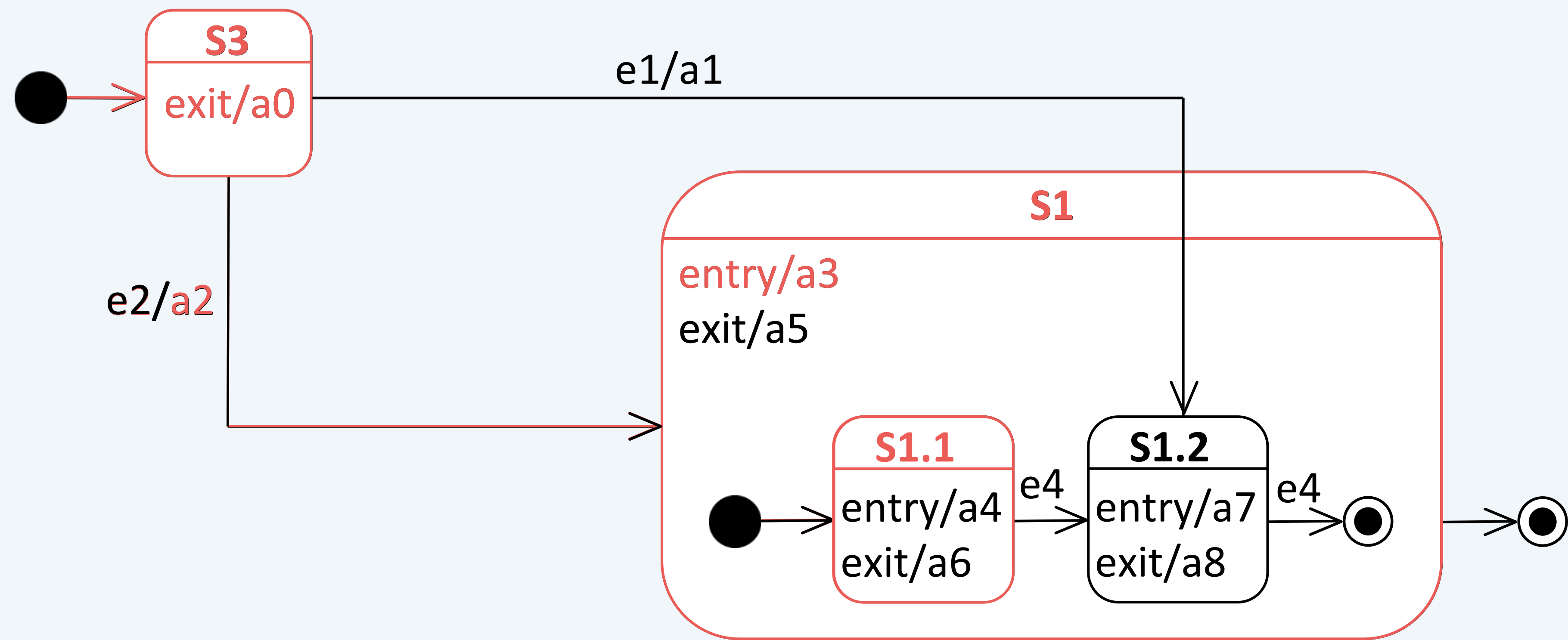
Zu einem Zeitpunkt sind **x** UND **y** aktiv!

Betreten eines komplexen Zustands (1/2)



- Transition an den Rand eines komplexen Zustands:
Startzustand wird aktiviert

Ereignis	Zustand	Ausgeführte Aktivitäten
„Beginn“	S3	
e2	S1/S1.1	a0-a2-a3-a4

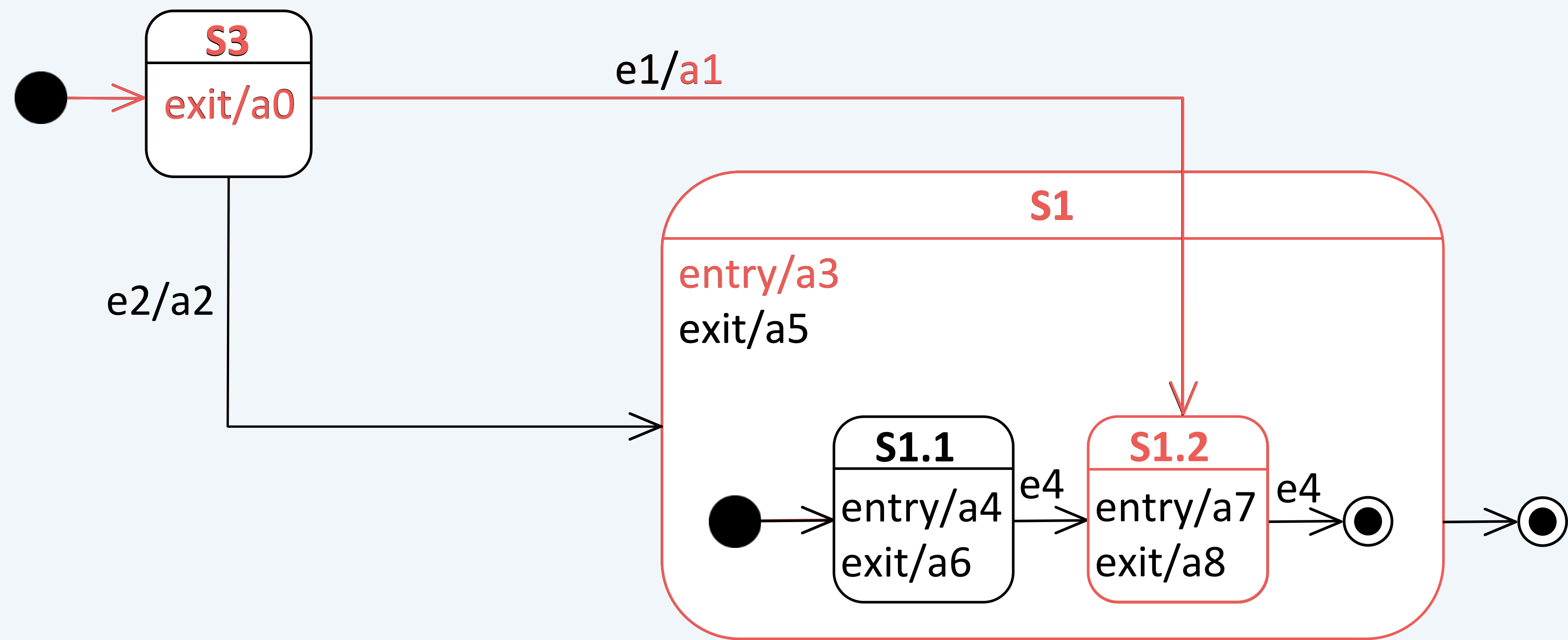


Betreten eines komplexen Zustands (2/2)



- Transition zu einem Subzustand:
Subzustand wird aktiviert

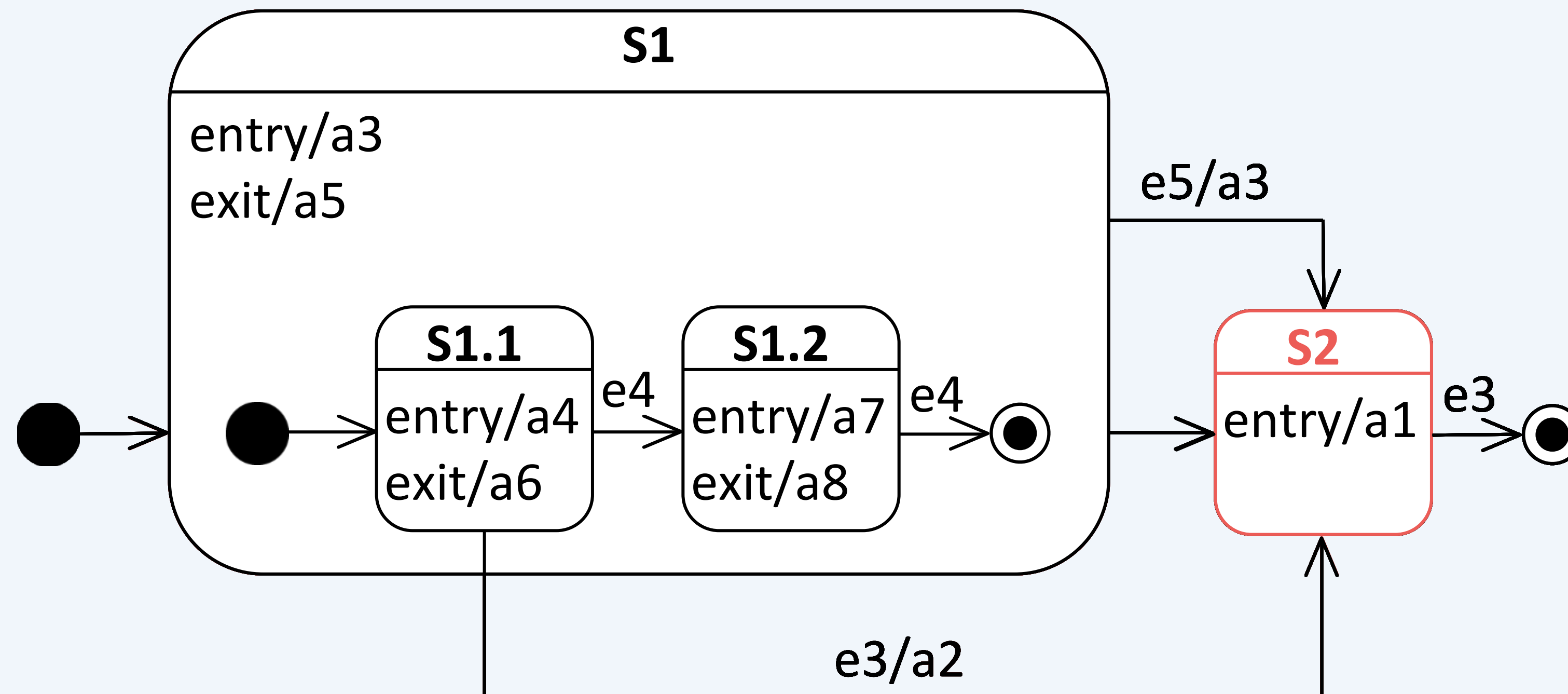
Ereignis	Zustand	Ausgeführte Aktivitäten
„Beginn“	S3	
e1	S1/S1.2	a0-a1-a3-a7



Verlassen eines komplexen Zustands (1/3)

■ Transition aus einem Subzustand

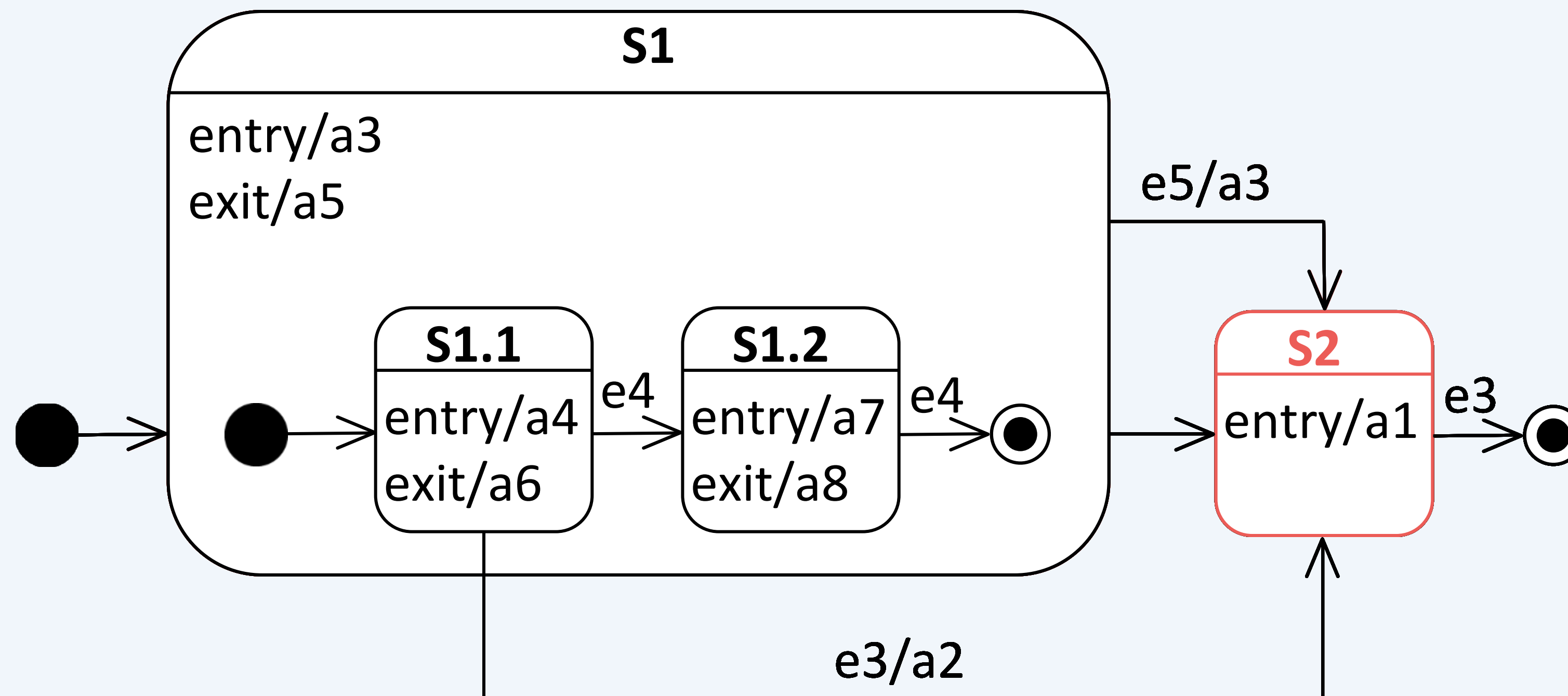
Ereignis	Zustand	Ausgeführte Aktivitäten
„Beginn“	S1/S1.1	a3-a4
e3	S2	a6-a5-a2-a1



Verlassen eines komplexen Zustands (2/3)

- Transition vom Rand eines komplexen Zustands

Ereignis	Zustand	Ausgeführte Aktivitäten
„Beginn“	S1/S1.1	a3-a4
e5	S2	a6-a5-a3-a1

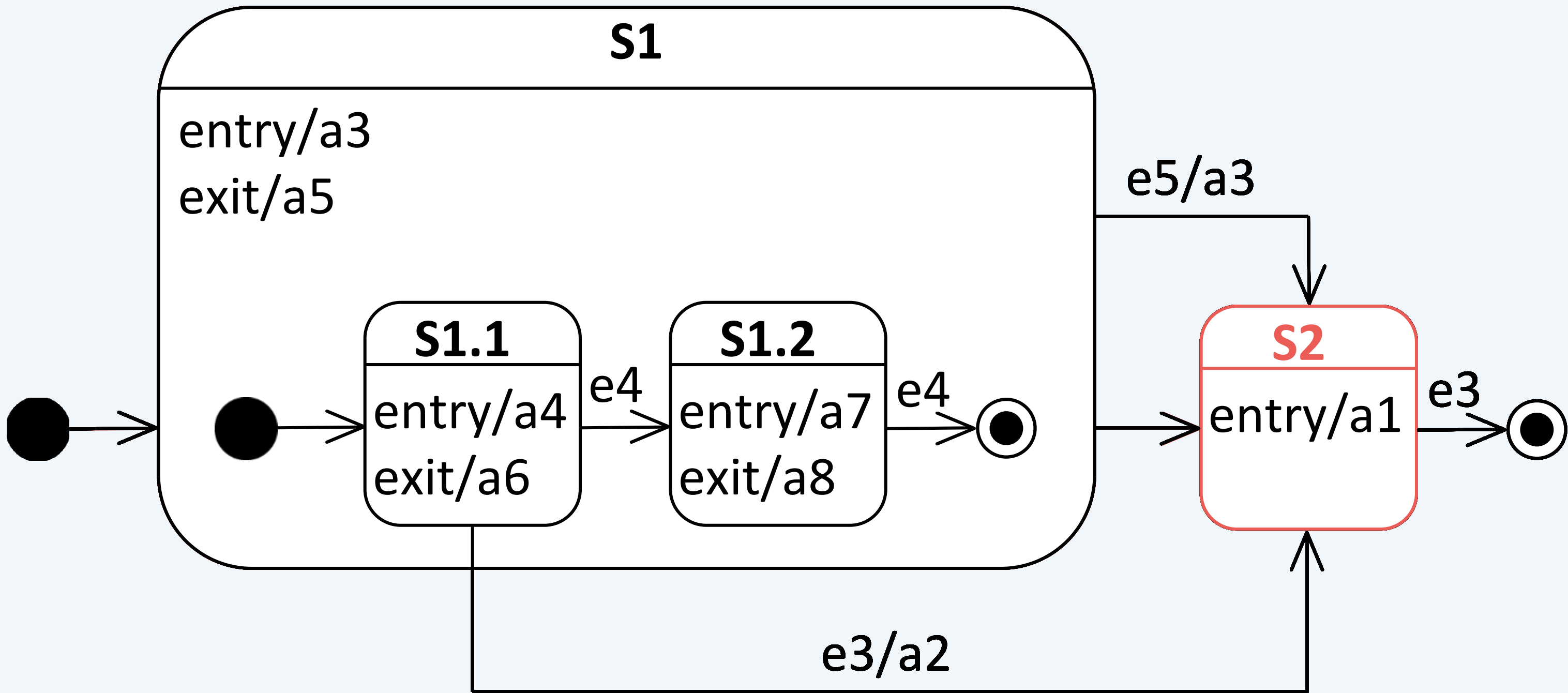


Verlassen eines komplexen Zustands (3/3)



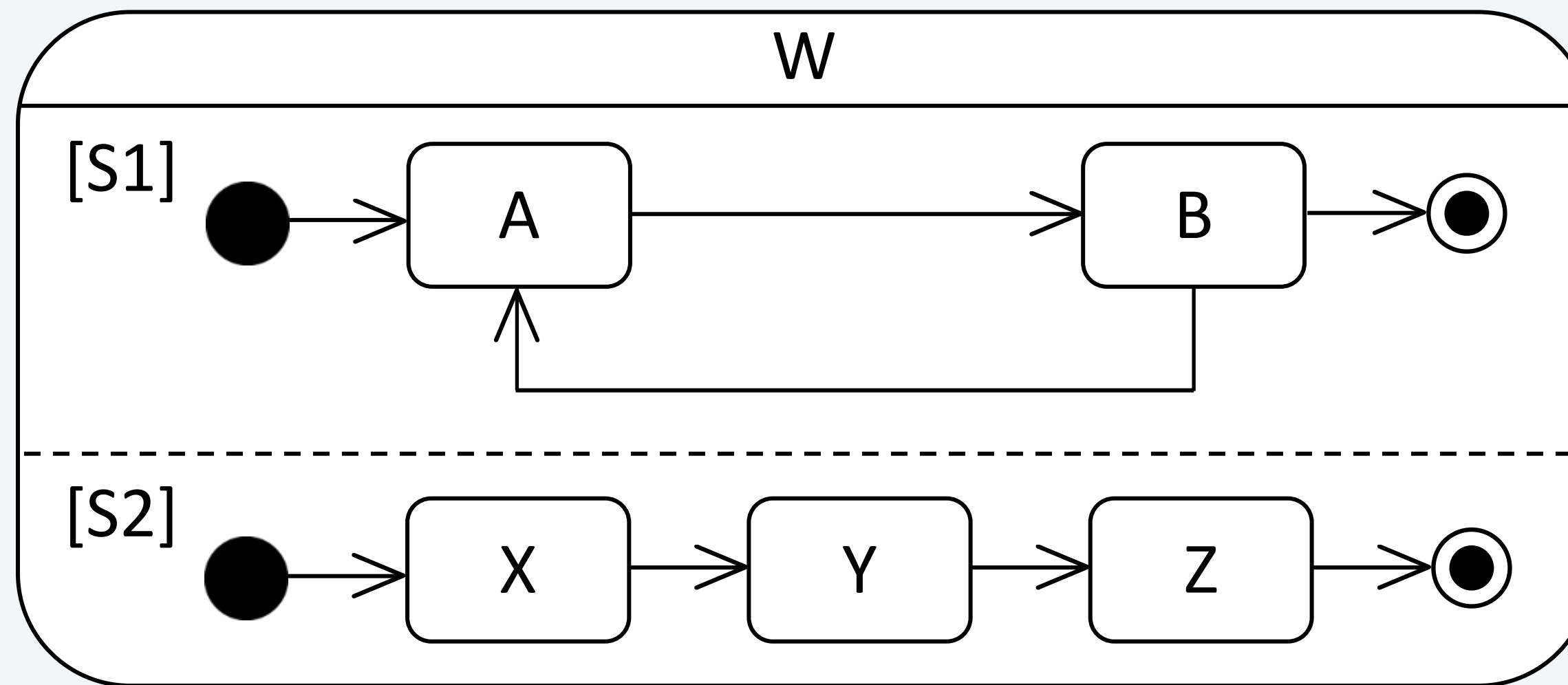
- Transition durch Beendigung der Subzustandsfolge

Ereignis	Zustand	Ausgeführte Aktivitäten
„Beginn“	S1/S1.1	a3-a4
e4	S1/S1.2	a6-a7
e4	S2	a8-a5-a1



Orthogonale (Nebenläufige) Zustände

■ Bsp.:



Zu einem Zeitpunkt ist jeweils ein Subzustand jeder der beiden orthogonalen (=parallelen) Regionen von W aktiv!

■ Mögliche Kombinationen von gleichzeitig aktiven Zuständen:

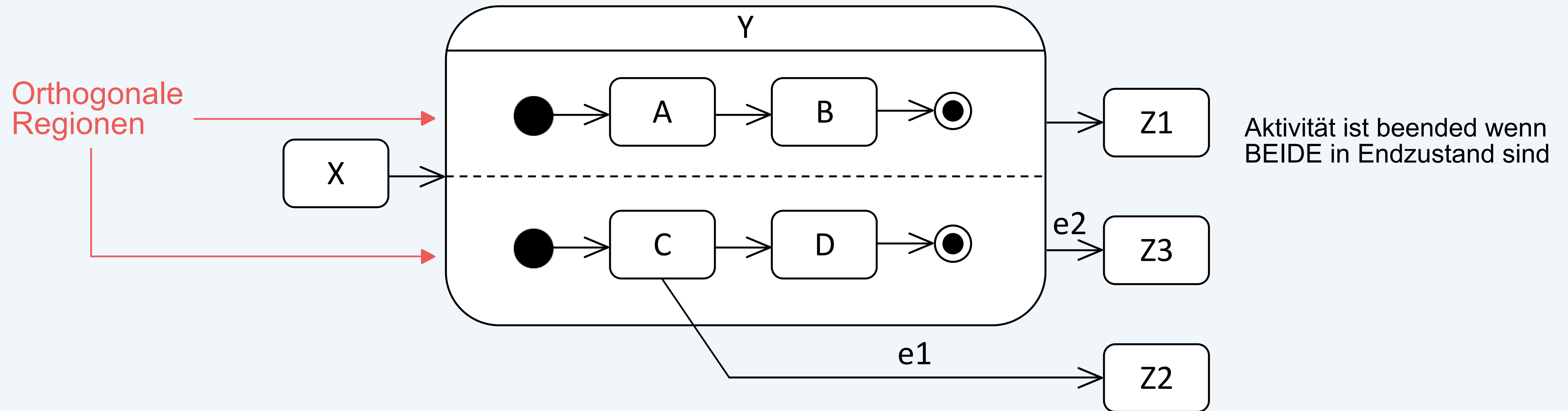
A & X oder A & Y oder A & Z oder A & Endzustand von $[S2]$

B & X oder B & Y oder B & Z oder B & Endzustand von $[S2]$

oder Endzustand von $[S1]$ & X oder Endzustand von $[S1]$ & Y oder Endzustand von $[S1]$

& Z oder Endzustand von $[S1]$ & Endzustand von $[S2]$

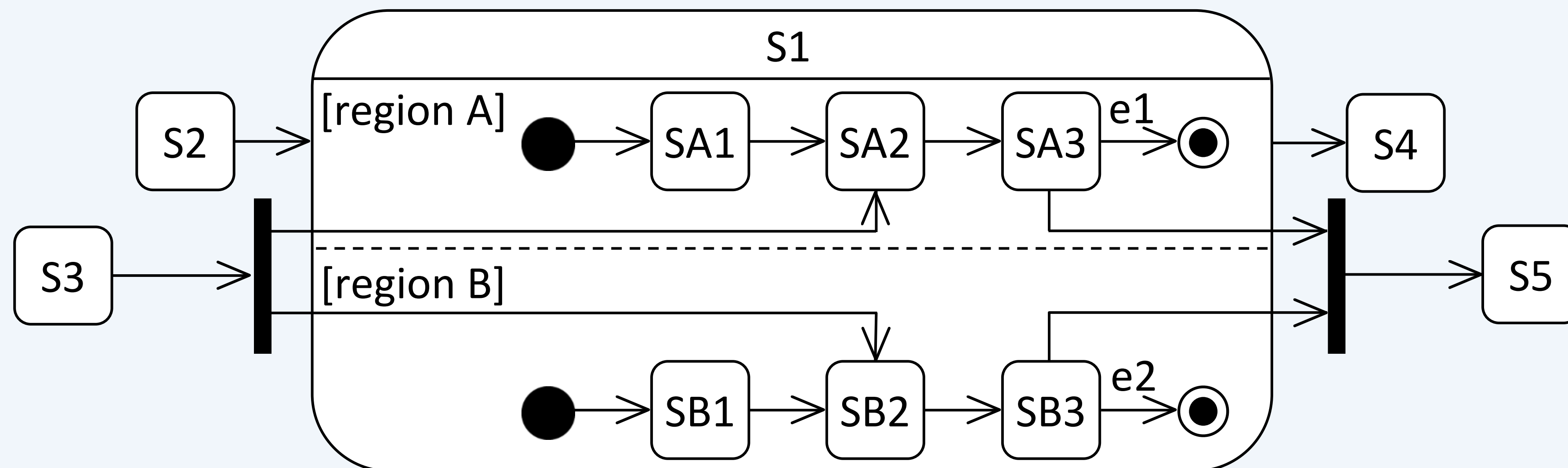
Verlassen von orthogonalen Zuständen



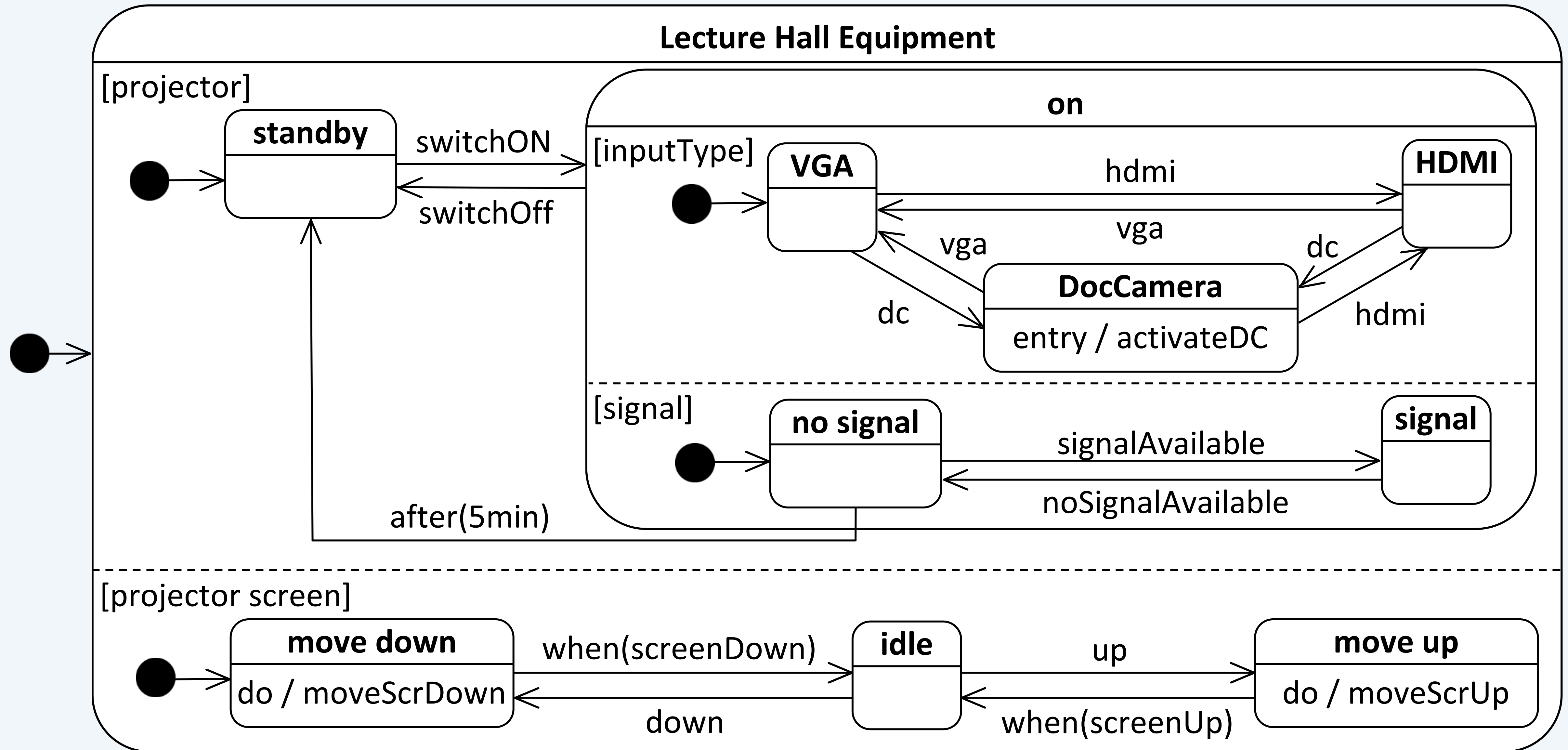
- Der komplexe Zustand **Y** wird verlassen, wenn
 - **B** und **D** verlassen worden sind (Folgezustand **Z1**)
[= die Subzustandsfolgen beendet sind]
 - in irgendeinem Subzustand Ereignis **e2** eintritt (Folgezustand **Z3**)
 - im Zustand **C** Ereignis **e1** eintritt (Folgezustand **Z2**)

Komplexe Transition für orthogonale Zustände

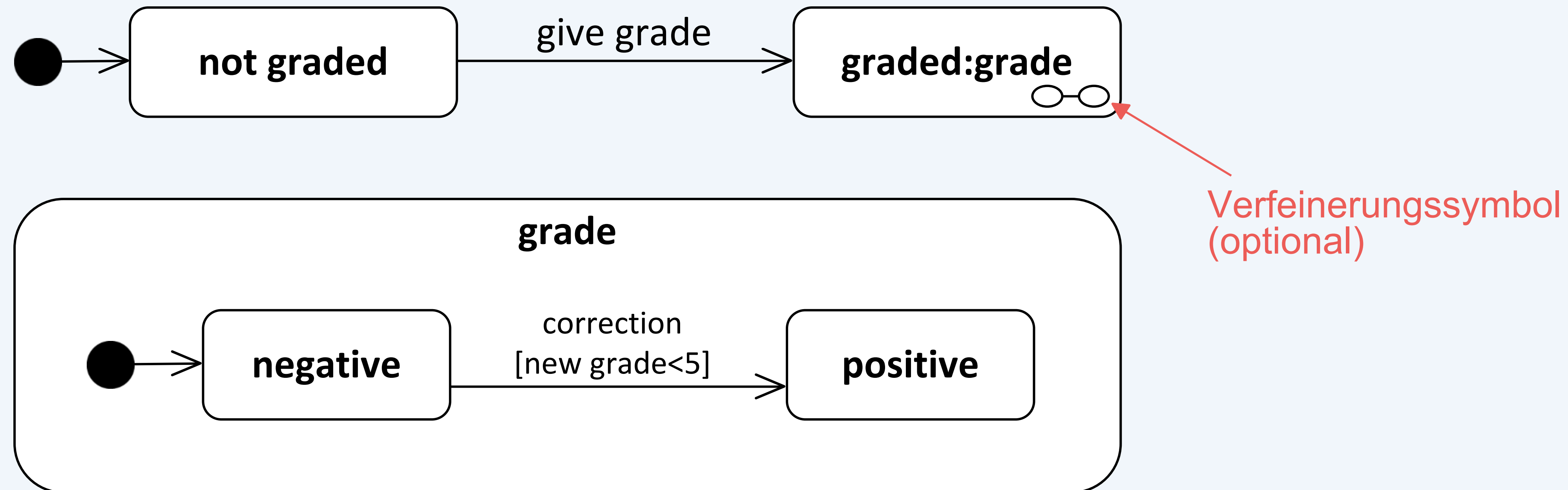
- Wird ein orthogonaler Zustand aktiviert, so werden **alle** Initialknoten seiner **nebenläufigen Regionen aktiviert**
- Möchte man den Kontrollfluss jedoch an anderen Positionen starten, verwendet man Parallelisierungs- bzw. Synchronisierungsknoten
- Parallelisierungsknoten
 - Zielzustände müssen unterschiedlichen Regionen angehören
 - Quellzustand muss außerhalb des orthogonalen Zustands liegen
(Regeln für Synchronisierungsknoten analog umgekehrt)



Beispiel Hörsaaltechnik



- Um Teile eines Zustandsdiagramms in anderen Zustandsdiagrammen wiederzuverwenden
- Notation: **Zustand:Unterautomat**
- Sobald der Unterautomatenzustand aktiviert wird, wird der Startzustand des referenzierten Unterautomaten aktiviert
 - Vergleichbar mit Unterprogrammaufruf in Programmiersprachen



Zustandsdiagramm Der historische Zustand

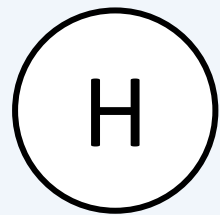


Christian Huemer und Marion Scholz

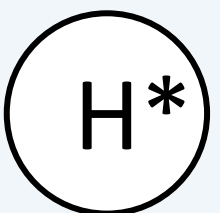
Historischer Zustand

- Historische Zustände merken sich den letzten internen **Zustand** in einem komplexen Zustand, bei Verlassen dieses komplexen Zustands.
- Zu einem späteren Zeitpunkt kann zu diesem Zustand **zurückgekehrt** werden
 - Alle **entry**-Aktivitäten werden wiederum ausgeführt

- **Flacher History-Zustand** merkt sich **eine Ebene**

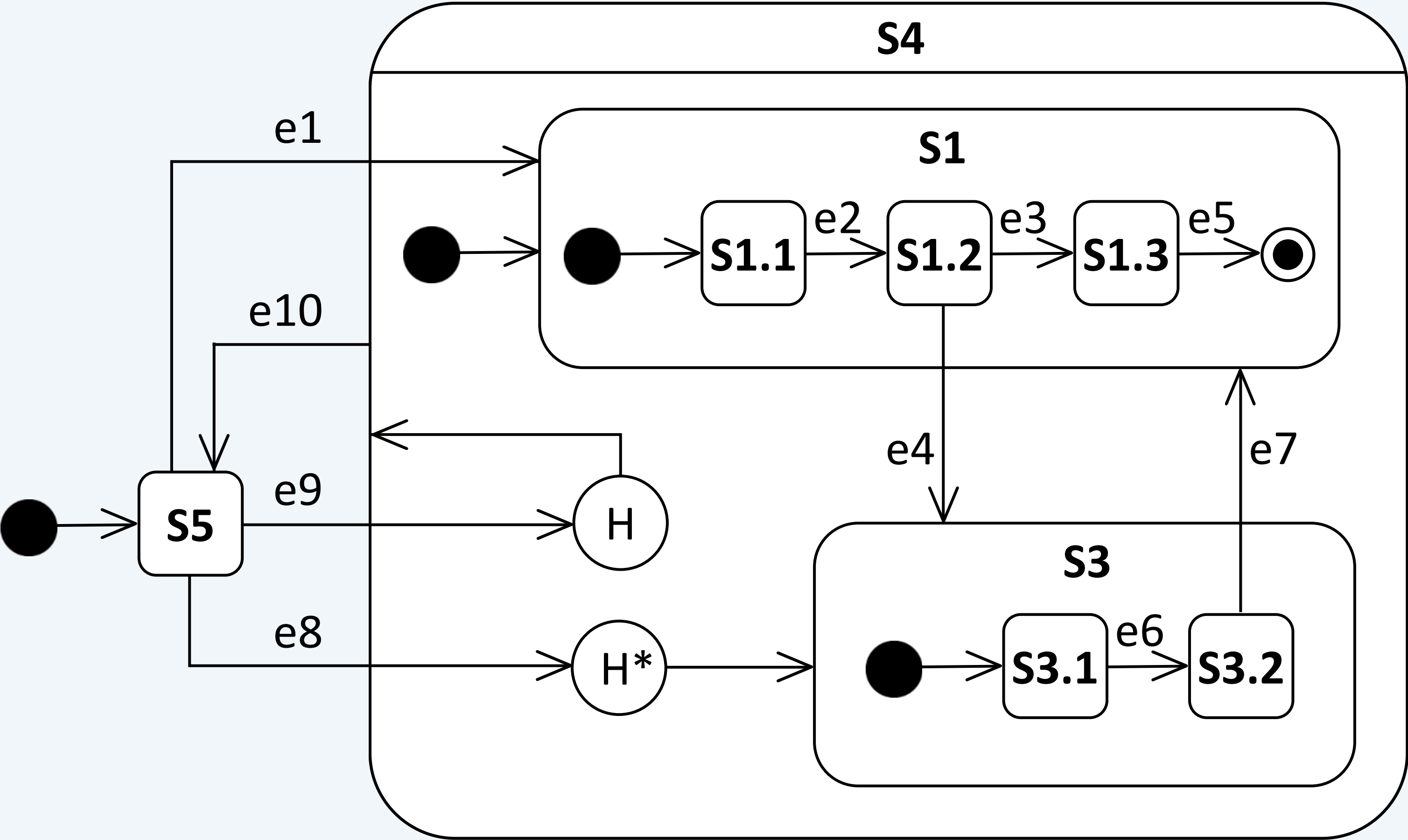


- Über einen **tiefen History-Zustand** »H*« werden alle Subzustände auf **Schachtelungsebene** festgehalten



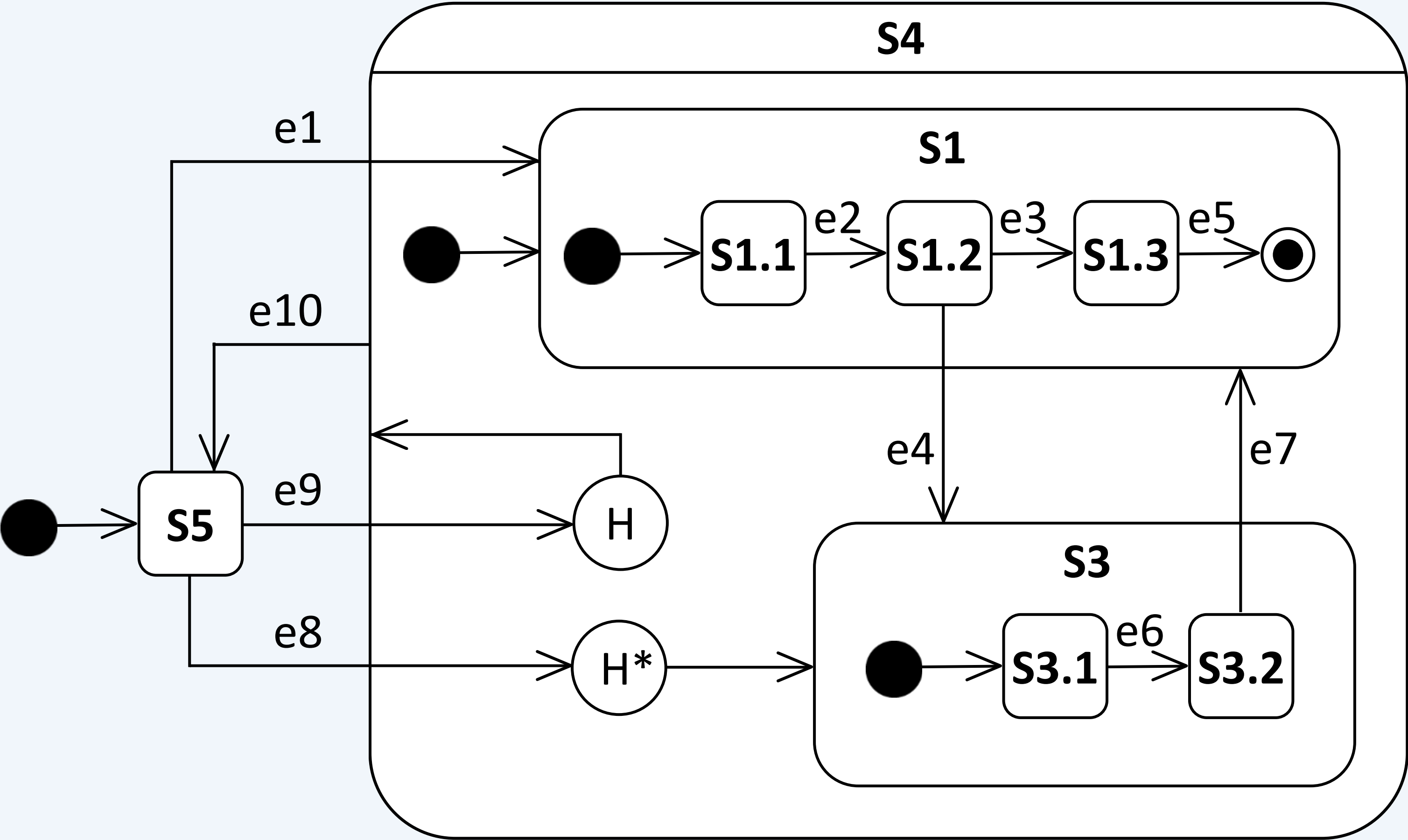
Statt Stern kann ich auch Zahl spezifizieren über wie viele Ebenen ich mir Zustand merken möchte

Beispiel historischer Zustand (1/4)



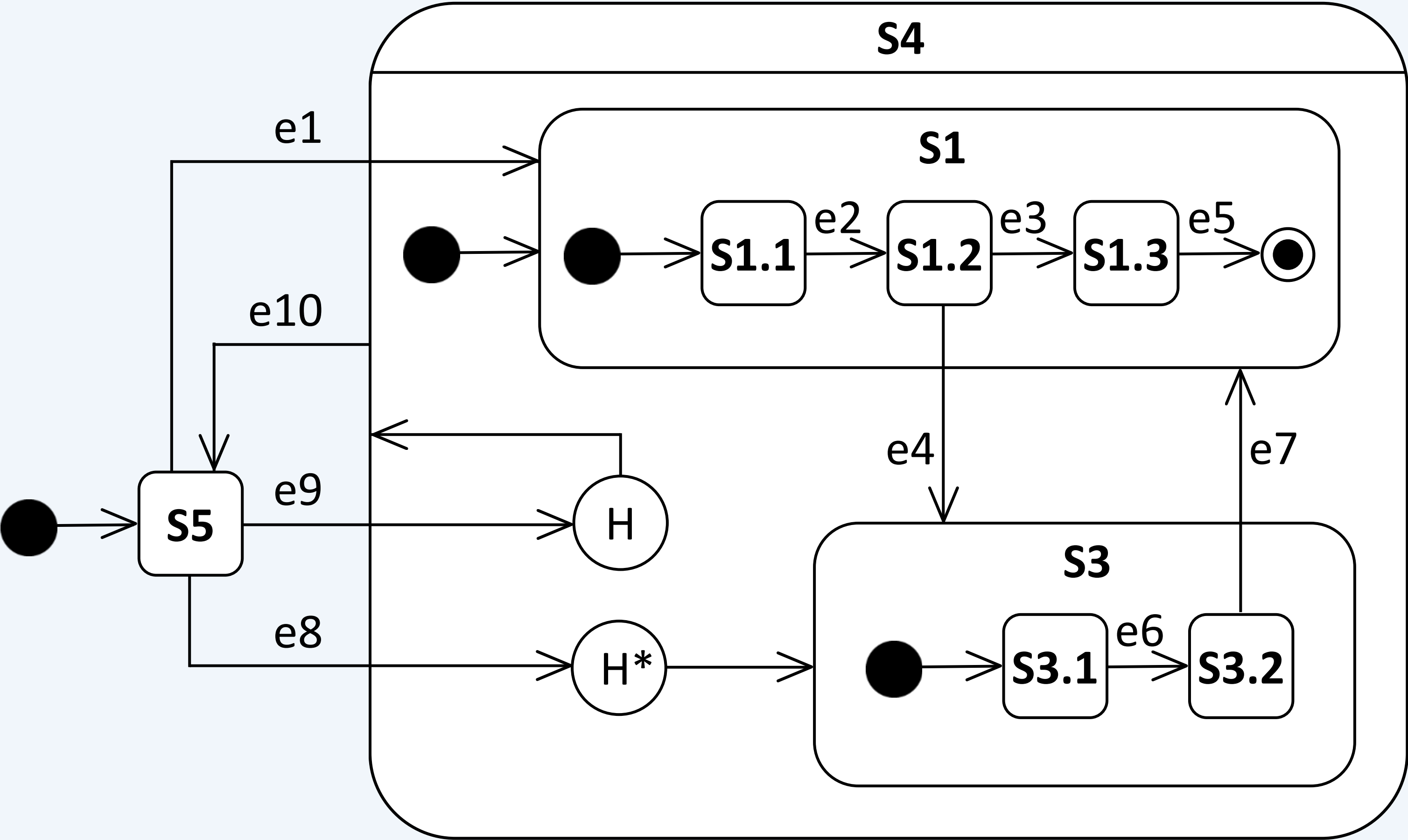
Ereignis	Zustand
„Beginn“	S5
e1	S4/S1/S1.1
e2	S4/S1/S1.2
e10	S5
e9	(H→) S4/S1/S1.1

Beispiel historischer Zustand (2/4)



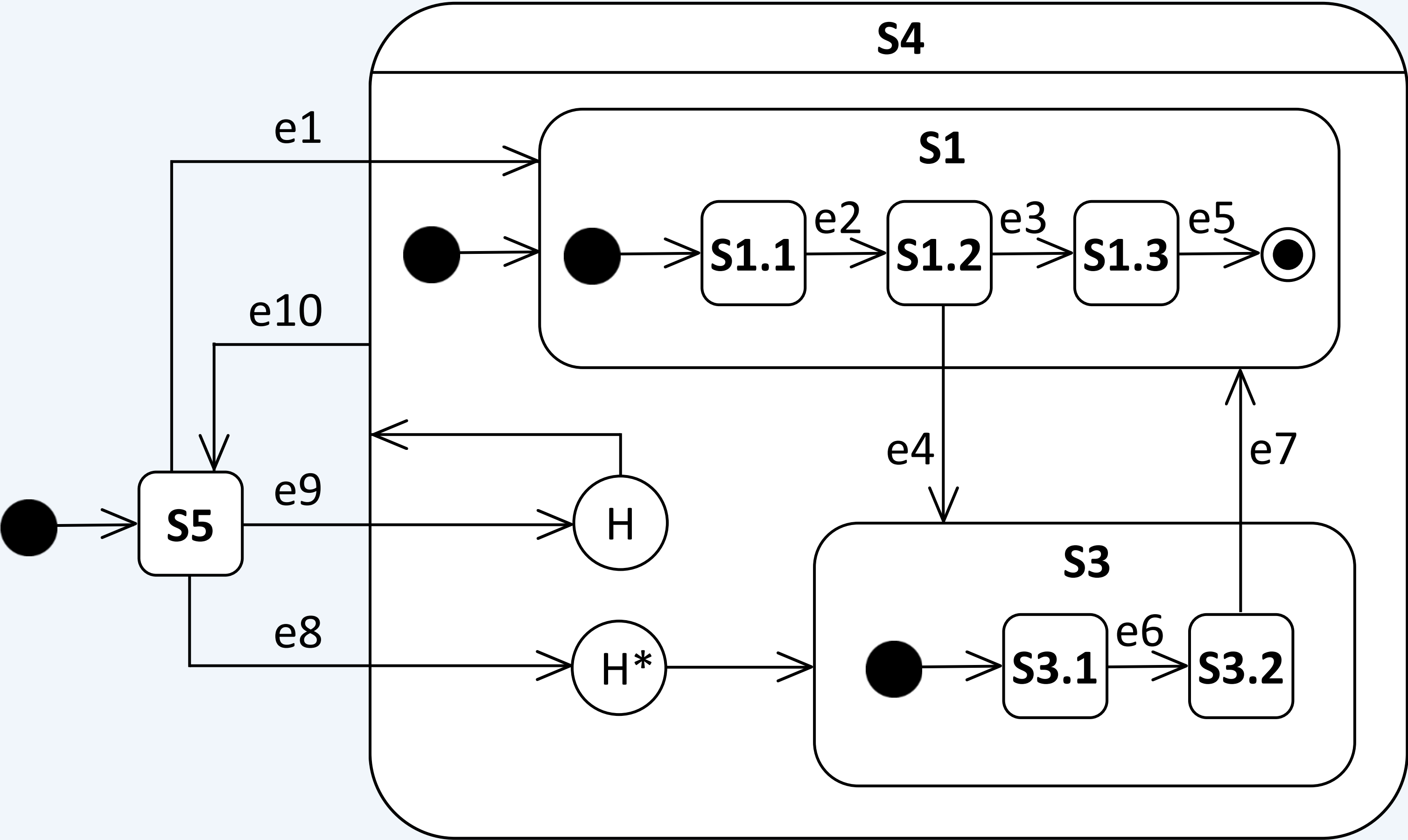
Ereignis	Zustand
„Beginn“	S5
e1	S4/S1/S1.1
e2	S4/S1/S1.2
e10	S5
e8	(H*→) S4/S1/S1.2

Beispiel historischer Zustand (3/4)



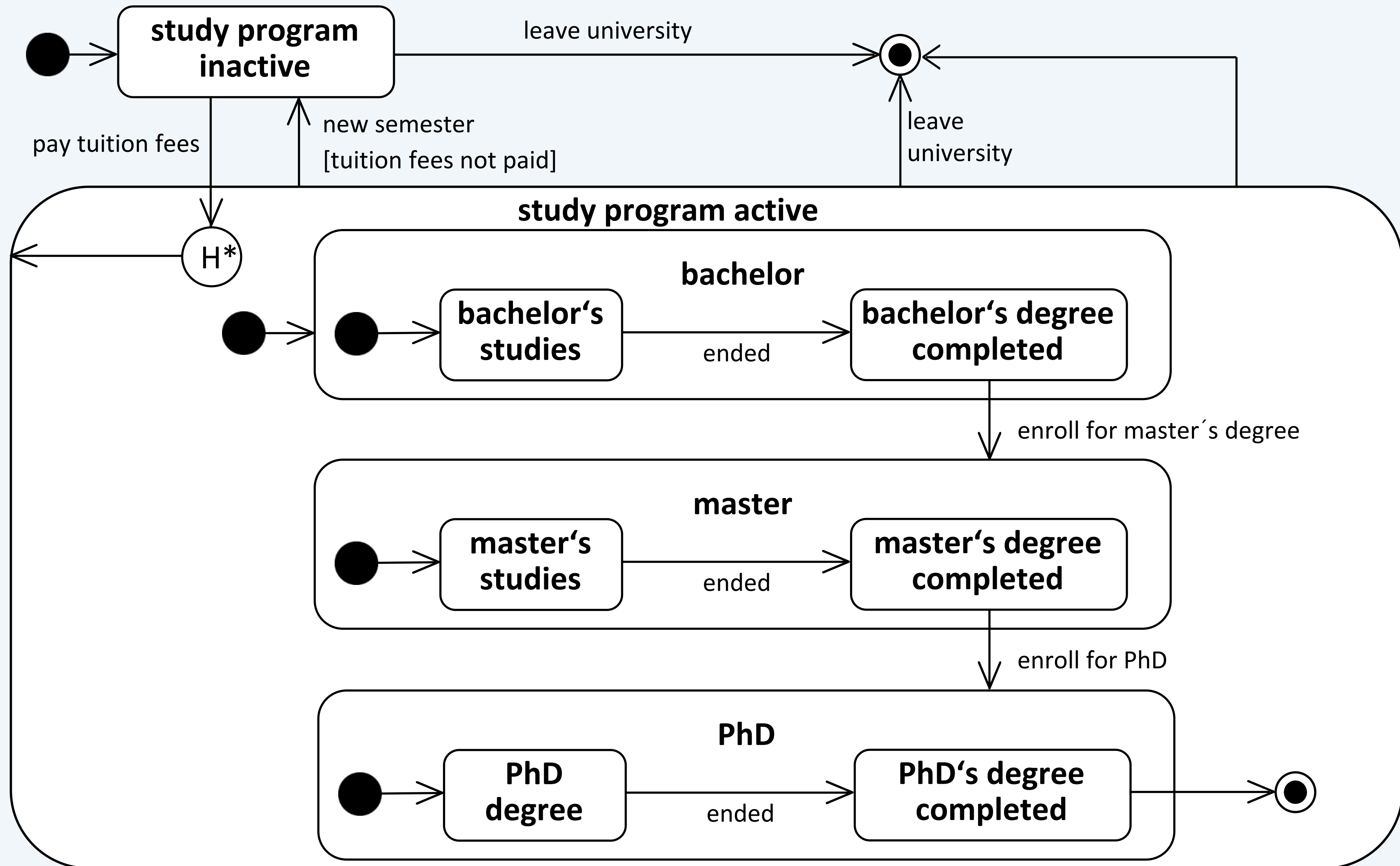
Ereignis	Zustand
„Beginn“	S5
e9	(H→) S4/S1/S1.1

Beispiel historischer Zustand (4/4)



Ereignis	Zustand
„Beginn“	S5
e8	(H*→) S4/S3/S3.1

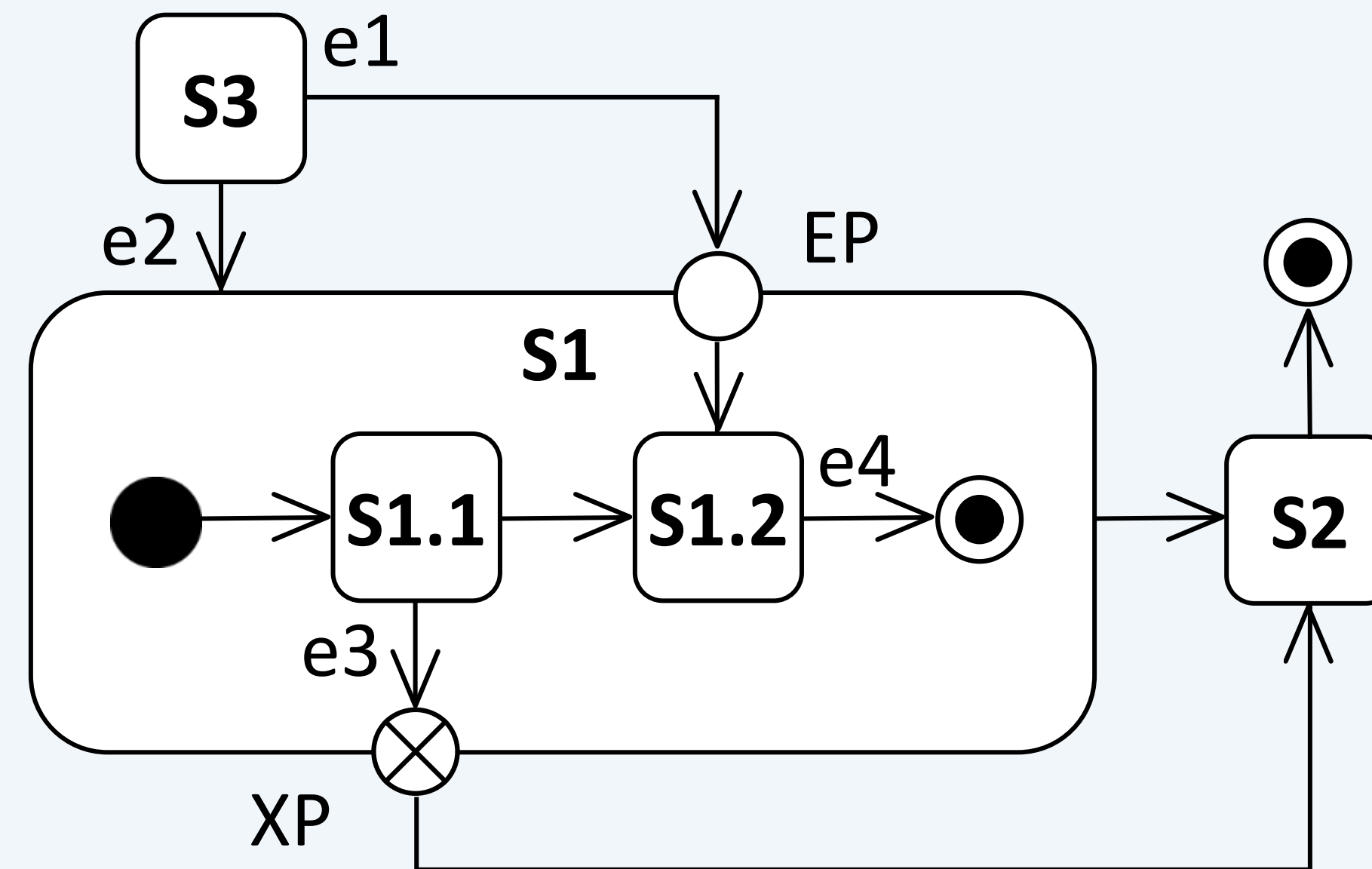
Beispiel Zustände einer akademischen Ausbildung



Ein- und Ausstiegspunkt

■ Kapselungsmechanismus

- Transition in einen bestimmten Subzustand eines komplexen Zustands, ohne dass die äußere Transition den Aufbau des komplexen Zustands kennen muss
- analog Transition aus einem komplexen Zustand



Beispiel Ein- und Ausstiegspunkt

