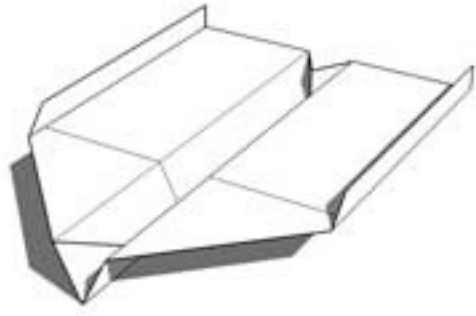


EDV & Multimedia Interaktionsdesign

11 – Modellierung

Prof. Dr. Jochen Koubek





Modell



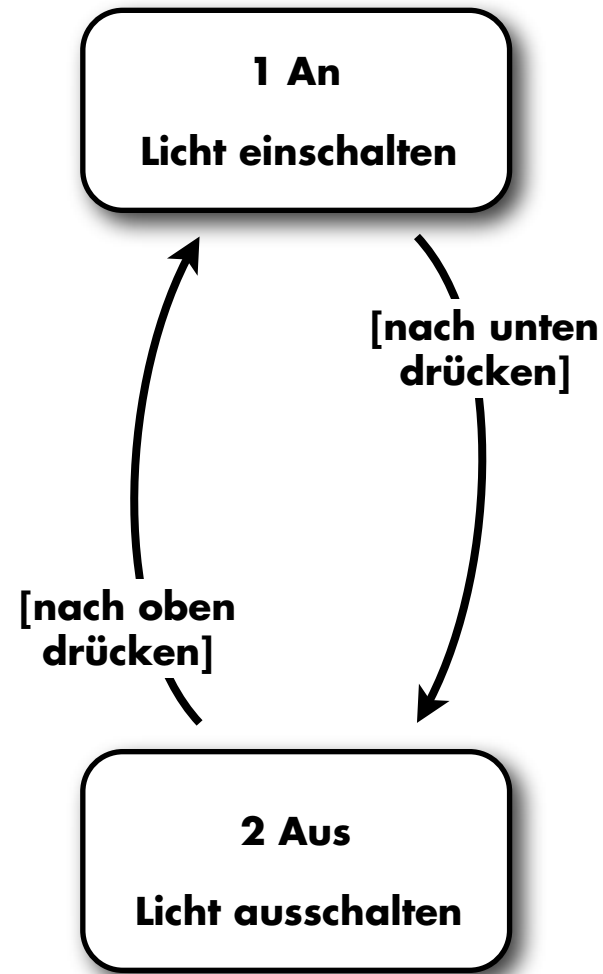
Ein Modell ist die Abbildung eines Weltausschnittes in einem anderen Medium (z.B. Papier, Begriffe, Mathematik, Datenstrukturen und Algorithmen)

Ein Modell konzentriert sich auf bestimmte Merkmale der Vorlage (z.B. Aussehen, Funktion, Verhalten) und lässt andere weg.

Jedes Modell wird zu einem bestimmten Zeitraum (wann?) von jemandem (von wem?) für jemanden (für wen?) aus einem Grund (warum?) für Anwendungsfälle (wozu?) eines bestimmten Zeitraum (für wann?) konstruiert.



Endliche Automaten



Lichtschalter

Ein endlicher Automat ist ein Modell für das Verhalten eines interaktiven Systems.

Zu jedem Zeitpunkt befindet er sich in genau einem **Zustand** von einer begrenzten (endlichen) Menge an möglichen Zuständen.

Je nach **Ereignis** (z.B. Eingabe, Messwert, Zeit) wechselt er in einen anderen Zustand (**Zustandsübergang**).

In jedem Zustand werden verschiedene **Aktionen** ausgeführt, die das **Verhalten** des Systems bestimmen.

Zustandsdiagramm (UML-Notation)

[http://de.wikipedia.org/wiki/Zustandsdiagramm_\(UML\)](http://de.wikipedia.org/wiki/Zustandsdiagramm_(UML))

http://en.wikipedia.org/wiki/UML_state_machine

Startzustand

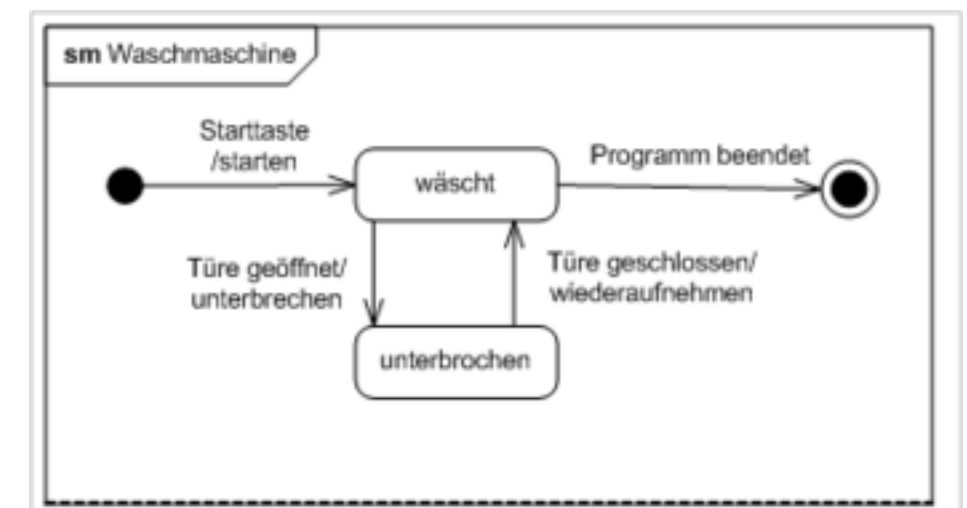
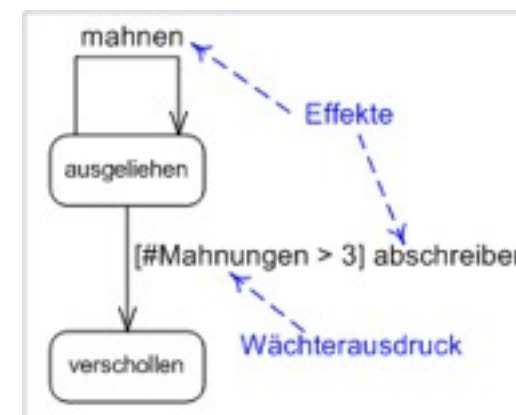
Endzustand (optional)

Zustand

Verhalten

entry | do | exit

Übergang (Transition)
[*Wächterausdruck*] Effekt



Übung

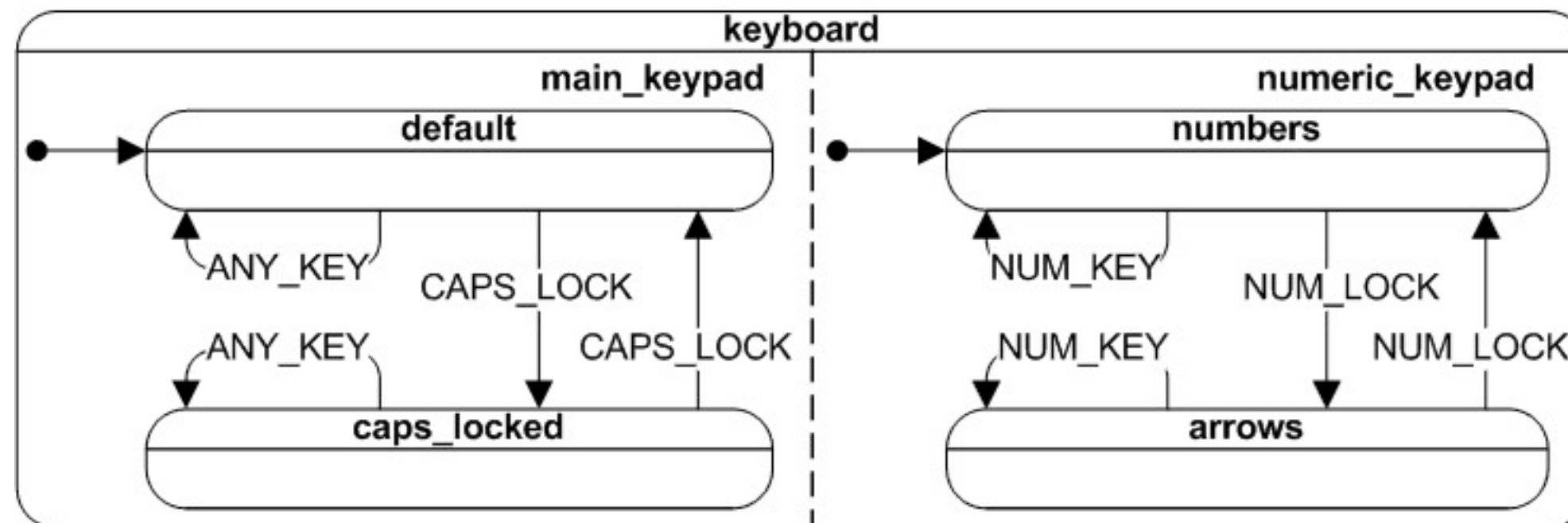


Modellieren Sie diesen
Auto-Lichtschalter mit einem
Zustandsdiagramm



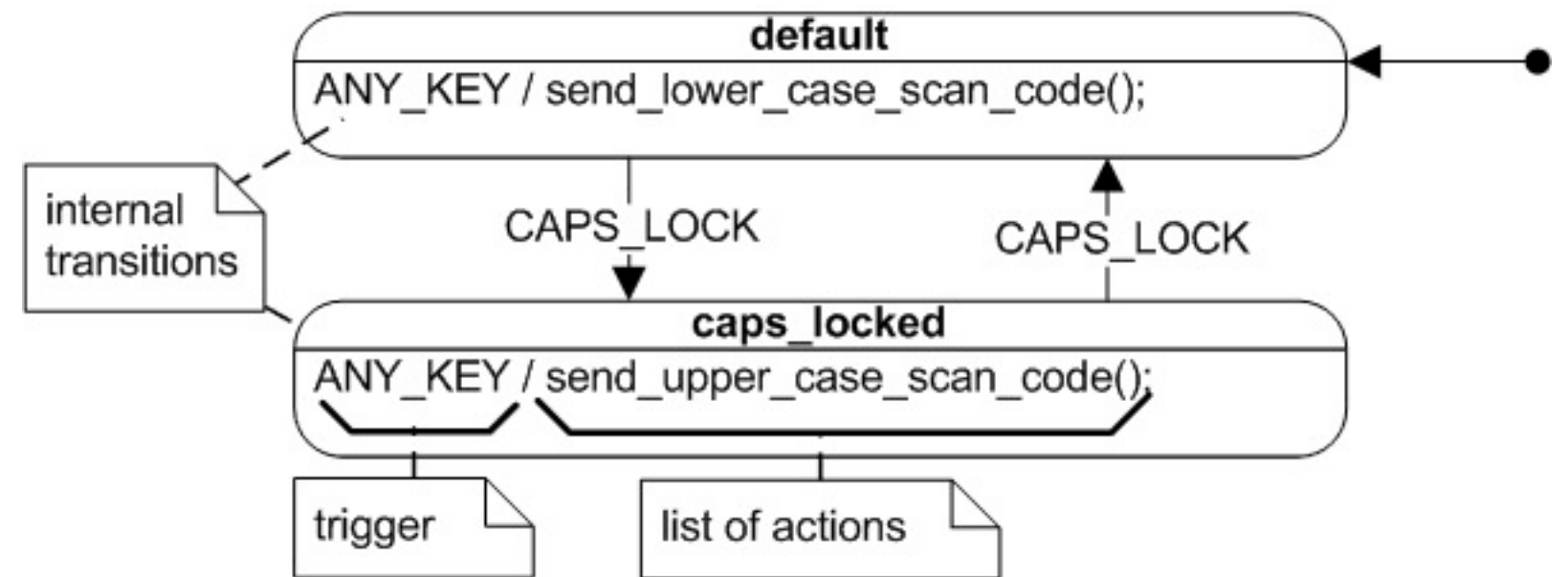
Wie ist es mit diesem?

Unabhängige Teilsysteme (Orthogonale Regionen)

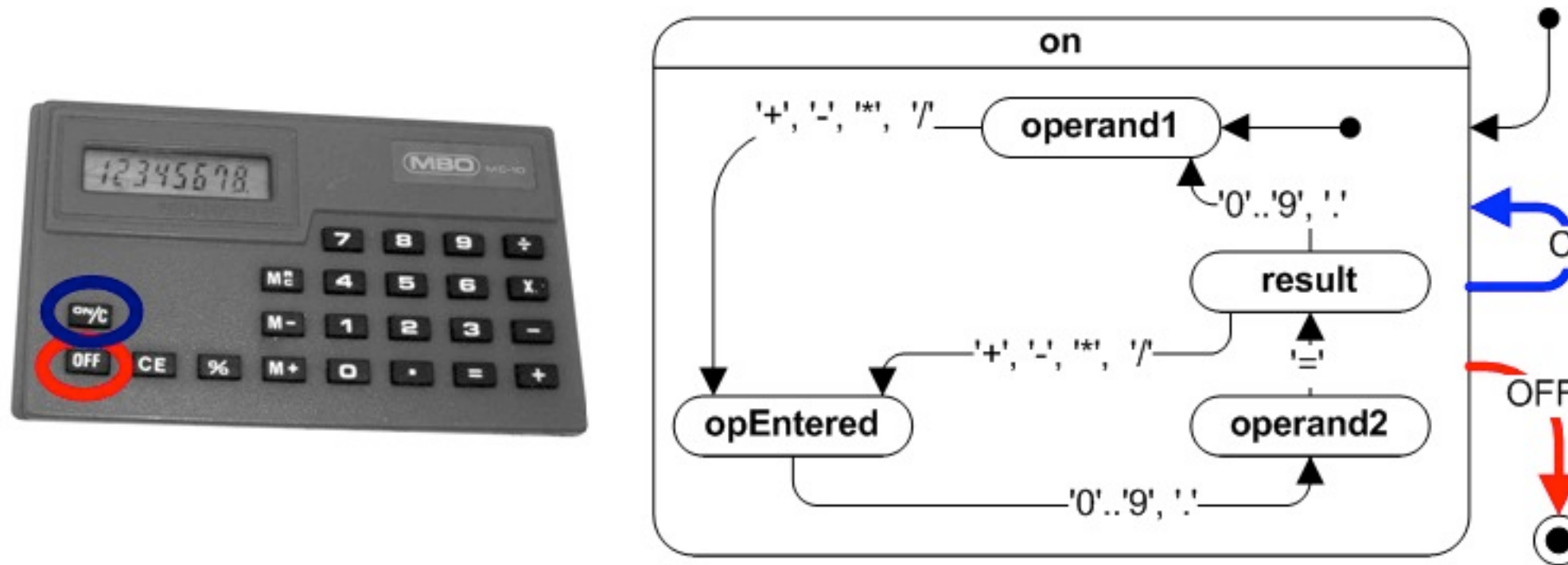


Interne Übergänge

Was ist mit der Helligkeitsregelung?

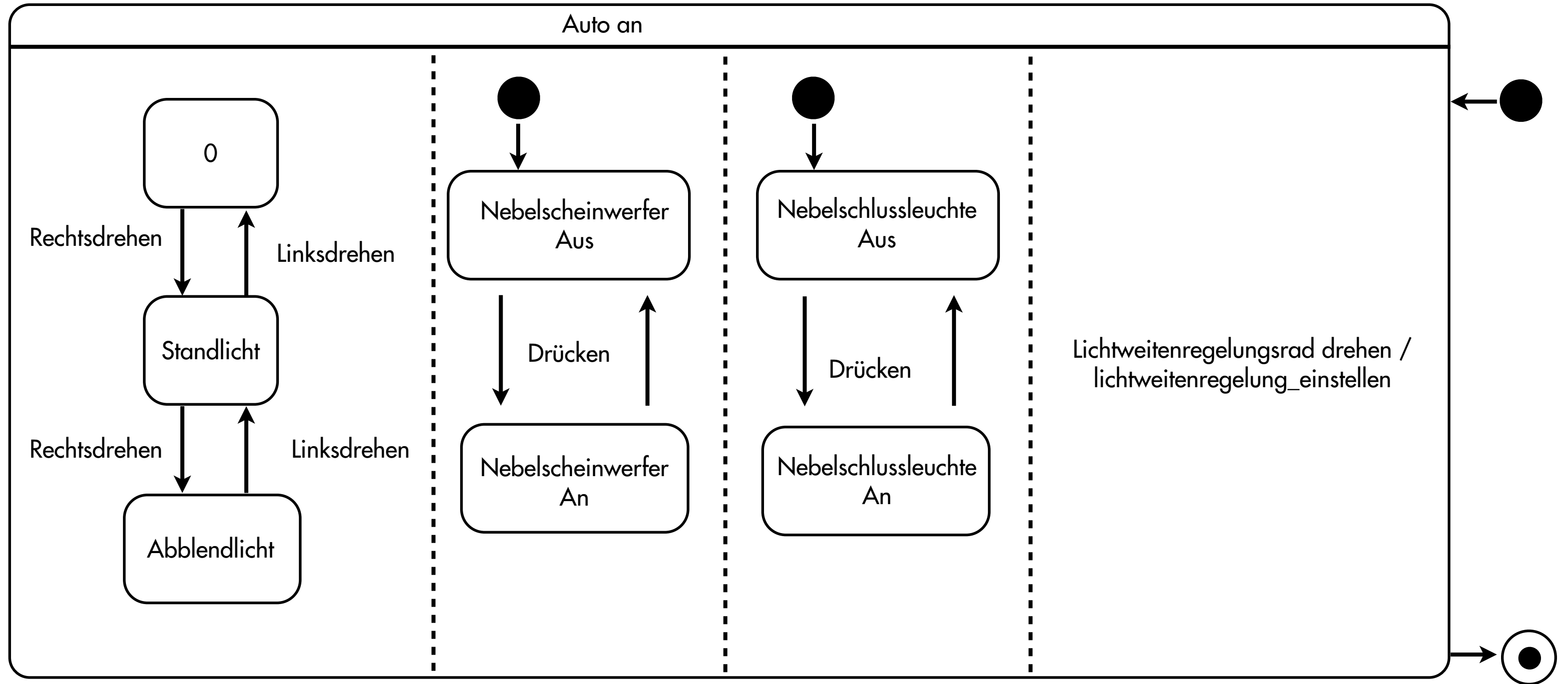


Zustandshierarchien

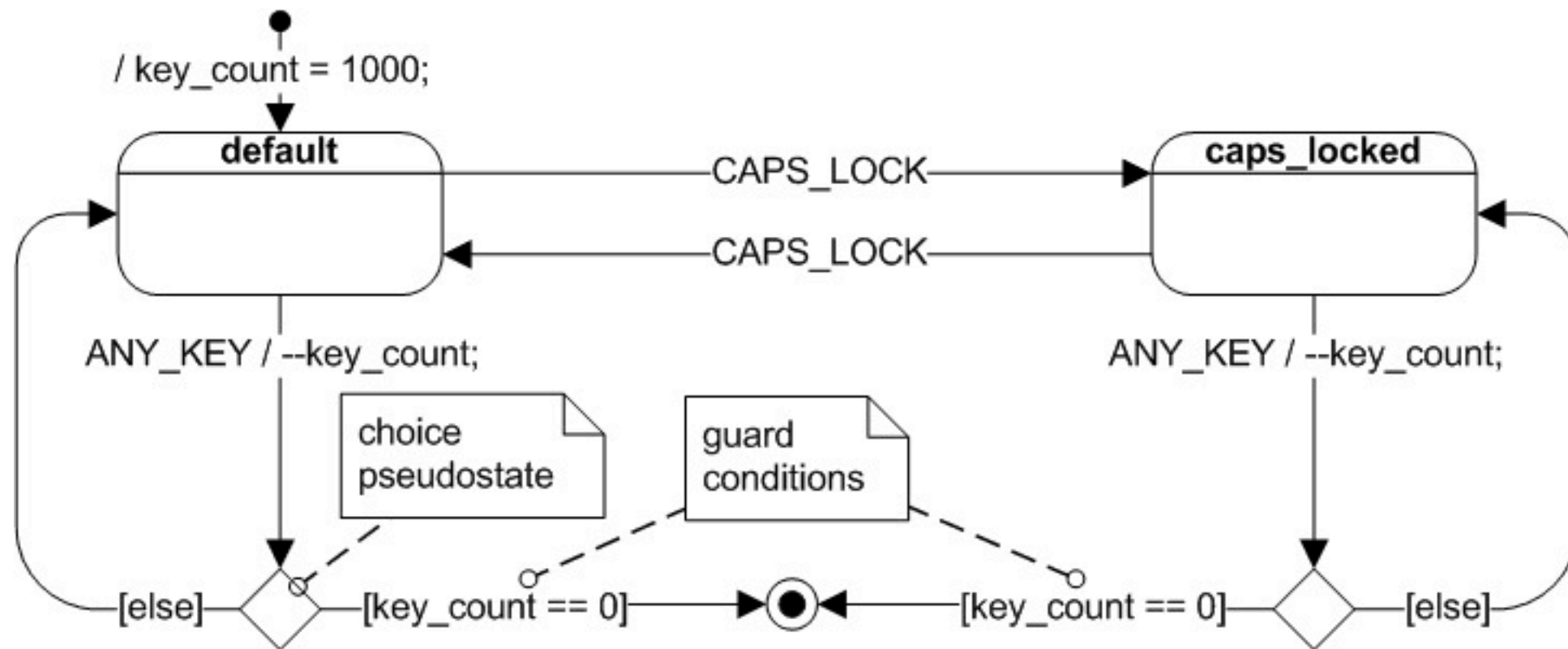


Ein **Zusammengesetzter Zustand** besteht aus eine Oberzustand und mehreren Unterzuständen. Die Rechentasten des Taschenrechners funktionieren nur im eingeschalteten Zustand.

Auto-Lichtschalter



Zustandsvariablen und Wächterausdrücke



Implementierung von Endlichen Automaten

Im Rahmen unserer Prototypen arbeiten wir mit

```
if (zustand == 1) {...} else if (zustand == n)
```

bzw.

```
switch (zustand) {case 1: ... case n: ... }
```

Wobei der aktuelle Zustand z.B. in einer Integer-Variable gespeichert wird.

Obere Zustandshierarchien

Processing bietet mit den Funktionen `setup()`, `draw()` und `stop()` Rahmen für entry-, do- und exit-Verhalten des Hauptprogramms.

In GameStudio müssen diese Rahmen in der Funktion `main()` implementiert werden, wobei die Schleife `while(1) { ... }` das do-Verhalten umschließt.

Übung

Modellieren Sie den Mensa-Kartenautomaten

