Software Engineering in der Praxis

Praktische Übungen

Model Checking II

Florin Pinte Marc Spisländer

Lehrstuhl für Software Engineering
Friedrich-Alexander-Universität Erlangen-Nürnberg

Modelchecking I

- Vertraut werden mit `NuSMV`
- Spezifikation von Zustandsautomaten
  - `MODULE`: asynchrone oder synchrone Automaten
  - `VAR`: spannen Zustandsraum auf
  - `ASSIGN`: Initialzustände und Zustandsübergänge
  - Constraints schränken Zustands- und Kantenmengen ein.
- Vertraut werden mit `CTL`

Modelchecking I

- Vertraut werden mit `NuSMV`
- Spezifikation von Zustandsautomaten
  - `MODULE`: asynchrone oder synchrone Automaten
  - `VAR`: spannen Zustandsraum auf
  - `ASSIGN`: Initialzustände und Zustandsübergänge
  - Constraints schränken Zustands- und Kantenmengen ein.
- Vertraut werden mit `CTL`

Modelchecking I

- Vertraut werden mit `NuSMV`
- Spezifikation von Zustandsautomaten
  - `MODULE`: asynchrone oder synchrone Automaten
  - `VAR`: spannen Zustandsraum auf
  - `ASSIGN`: Initialzustände und Zustandsübergänge
  - Constraints schränken Zustands- und Kantenmengen ein.
- Vertraut werden mit `CTL`

Modelchecking I

- Vertraut werden mit `NuSMV`
- Spezifikation von Zustandsautomaten
  - `MODULE`: asynchrone oder synchrone Automaten
  - `VAR`: spannen Zustandsraum auf
  - `ASSIGN`: Initialzustände und Zustandsübergänge
  - Constraints schränken Zustands- und Kantenmengen ein.
- Vertraut werden mit `CTL`

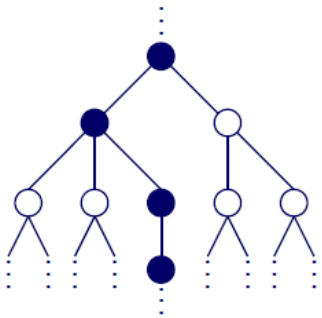
Modelchecking I

- Vertraut werden mit `NuSMV`
- Spezifikation von Zustandsautomaten
  - `MODULE`: asynchrone oder synchrone Automaten
  - `VAR`: spannen Zustandsraum auf
  - `ASSIGN`: Initialzustände und Zustandsübergänge
  - Constraints schränken Zustands- und Kantenmengen ein.
- Vertraut werden mit `CTL`

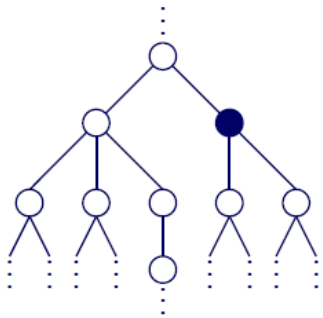
Modelchecking I

- Vertraut werden mit `NuSMV`
- Spezifikation von Zustandsautomaten
  - `MODULE`: asynchrone oder synchrone Automaten
  - `VAR`: spannen Zustandsraum auf
  - `ASSIGN`: Initialzustände und Zustandsübergänge
  - Constraints schränken Zustands- und Kantenmengen ein.
- Vertraut werden mit `CTL`

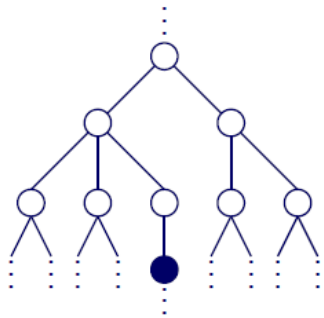
Grafische Veranschaulichung der CTL-Formeln



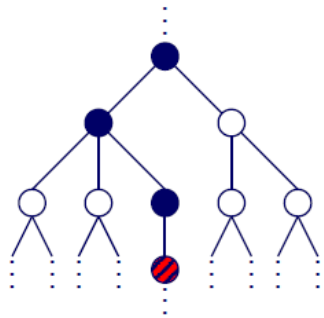
$\mathsf{EG}\psi$



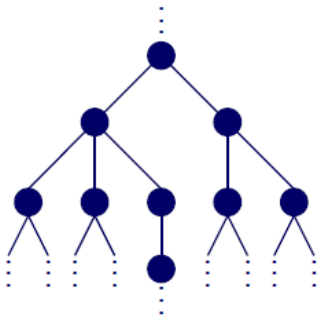
$\mathsf{EX}\psi$



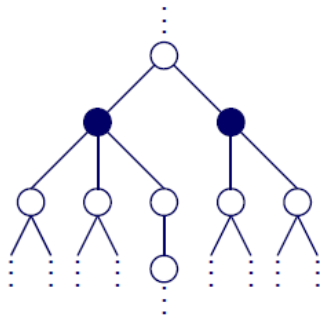
$\mathsf{EF}\psi$



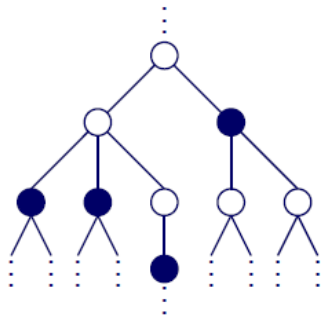
$\mathsf{E}[\phi\,\mathsf{U}\,\psi]$



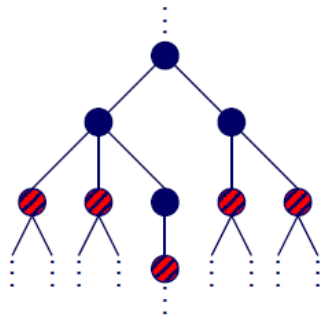
$\mathsf{AG}\psi$



$\mathsf{AX}\psi$



$\mathsf{AF}\psi$



$\mathsf{A}[\phi\,\mathsf{U}\,\psi]$

Definition der CTL^*

Zustandsformeln, CTL^* Formeln

Sei $\phi \in AP$ eine Aussage. Die Menge der Zustandsformeln ist durch folgende induktive Definition gegeben:

- ϕ ist eine Zustandsformel.
- ϕ und ψ Zustandsformeln sind, dann sind auch $\neg\phi$ und $\phi \vee \psi$ Zustandsformeln.
- Wenn ϕ eine Pfadformel ist, dann ist $E\phi$ eine Zustandsformel.

Definition der CTL^*

Pfadformeln

Die Menge der Pfadformeln ist gegeben durch:

- Alle Zustandsformeln sind Pfadformeln.
- Wenn ϕ und ψ Pfadformeln sind, dann sind auch $X\phi$ und $\phi \,U\, \psi$ Pfadformeln.

Weitere Quantoren

Die Quantoren A,F und G werden auf die übrigen Operatoren zurückgeführt.
Z.B. $A\phi := \neg E \neg \phi$ (dual zu E)

Definition der CTL^*

Pfadformeln

Die Menge der Pfadformeln ist gegeben durch:

- Alle Zustandsformeln sind Pfadformeln.
- Wenn ϕ und ψ Pfadformeln sind, dann sind auch Xϕ und ϕ U ψ Pfadformeln.

Weitere Quantoren

Die Quantoren A,F und G werden auf die übrigen Operatoren zurückgeführt.
Z.B. Aϕ := $\neg$E$\neg\phi$ (dual zu E)

LTL und CTL als Untermengen von CTL^*

Einschränkungen bei CTL

- Verwendung von X, F, G und U stets nach A oder E
- Aussagen immer nur über Pfade
- Verzweigte Logik

Einschränkungen bei LTL

- *Beliebige* Verwendung von X, F, G, U
- jedoch keine A oder E
- Lineare Logik

LTL und CTL als Untermengen von CTL^*

Einschränkungen bei CTL

- Verwendung von X, F, G und U stets nach A oder E
- Aussagen immer nur über Pfade
- Verzweigte Logik

Einschränkungen bei LTL

- *Beliebige* Verwendung von X, F, G, U
- jedoch keine A oder E
- Lineare Logik

LTL und CTL als Untermengen von CTL^*

Einschränkungen bei CTL

- Verwendung von X, F, G und U stets nach A oder E
- Aussagen immer nur über Pfade
- Verzweigte Logik

Einschränkungen bei LTL

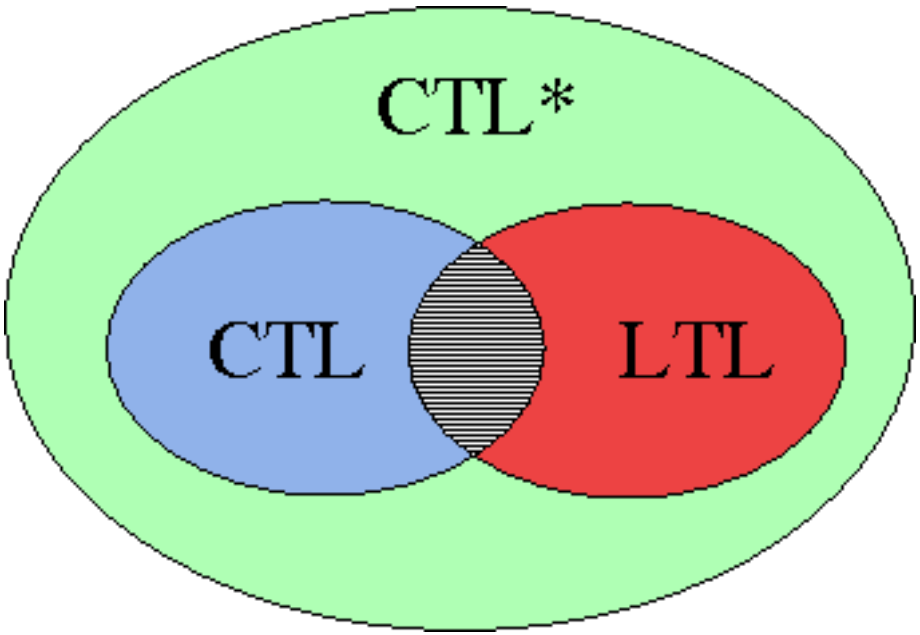
- *Beliebige* Verwendung von X, F, G, U
- jedoch keine A oder E
- Lineare Logik

Übliche Schreibweisen

Achtung

Manchmal wird LTL-Formeln der Allquantor A vorangestellt.

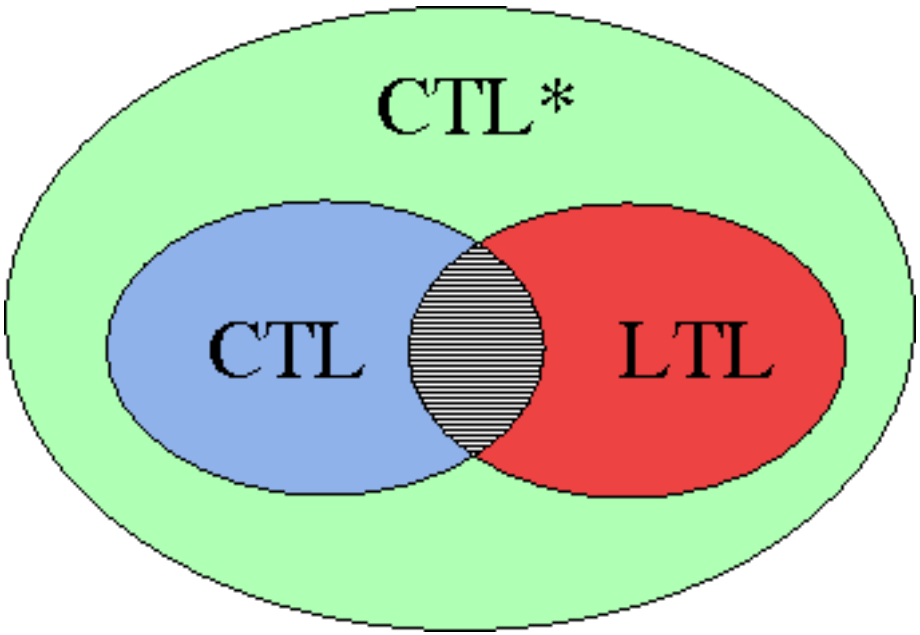
CTL^*, CTL und LTL



Beispiele

- $\mathsf{GF}\phi \Rightarrow \mathsf{F}\psi$
- $A(FGp)$
- $AG(EFp)$
- $\mathsf{E}(\mathsf{GF}\phi)$

CTL^*, CTL und LTL



Beispiele

- $\mathsf{GF}\phi \Rightarrow \mathsf{F}\psi$
- $A(FGp)$
- $AG(EFp)$
- $\mathsf{E}(\mathsf{GF}\phi)$

Moore-Maschinen

Informelle Beschreibung

Moore-Maschine definiert durch:

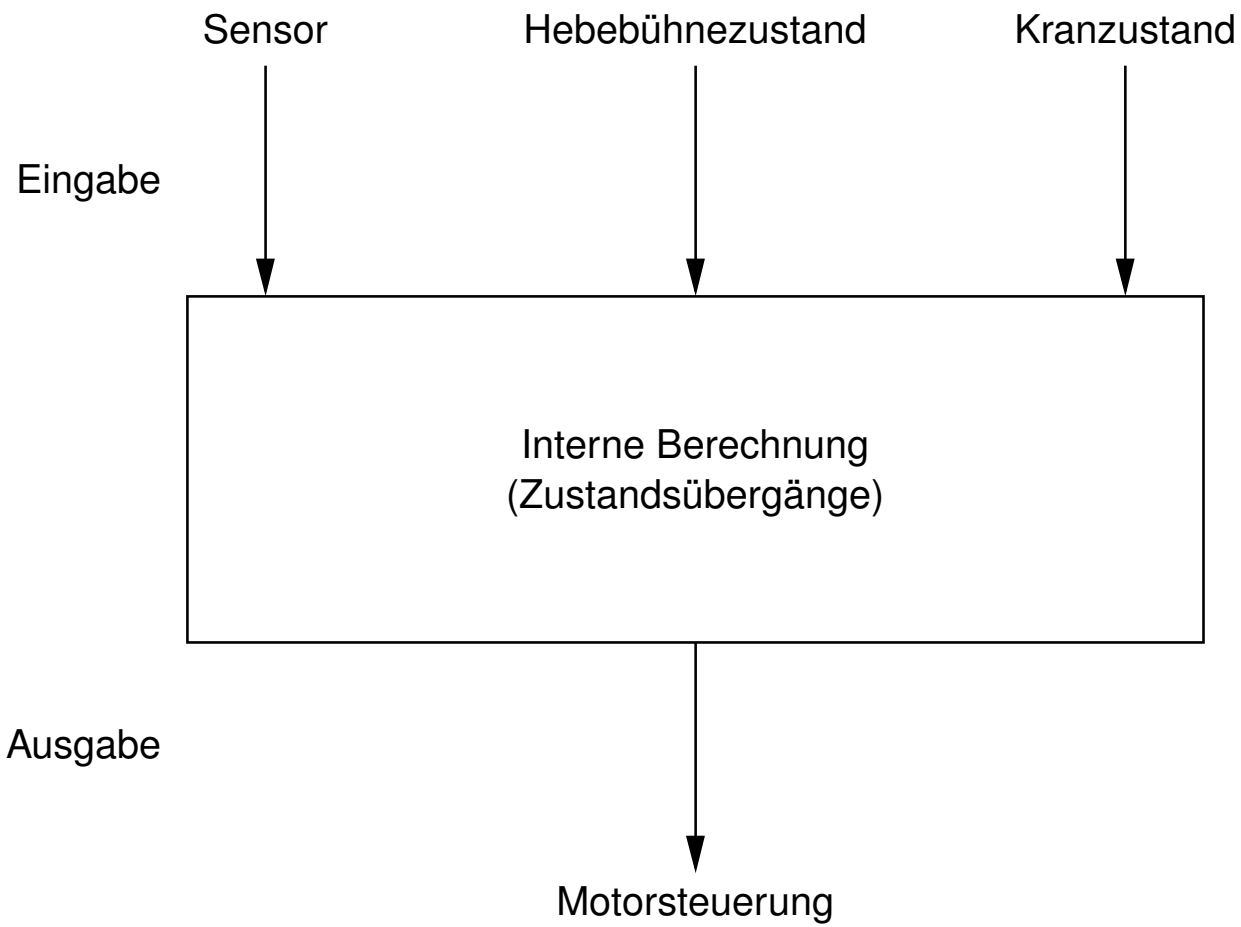
- Endliche Zustandsmenge Q
- Übergangsfunktion $Q \times \Sigma_{\mathrm{i}} \longrightarrow Q$
- Ausgabefunktion $Q \longrightarrow \Sigma_{\mathrm{o}}$

Die Produktionszelle

Systematisches Vorgehen

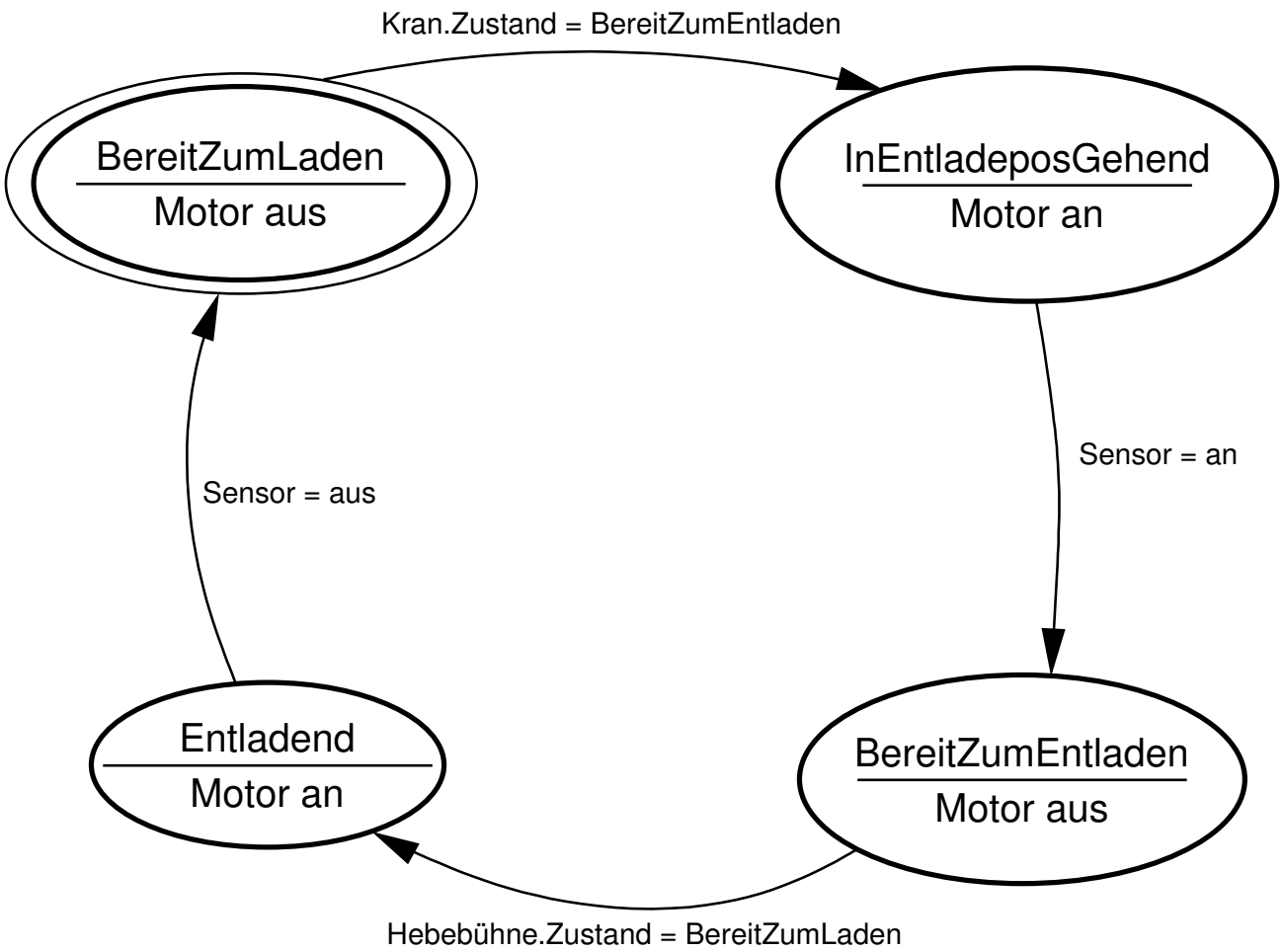
- Formuliere die einzelnen Komponenten (Zufuhrband, Hebebühne, Roboter, . . .) als Moore-Automat
- Übertrage den Moore-Automaten in NuSMV

Beispiel: Zufuhrband



Zufuhrband

Schritt 1: Moore-Automaten formulieren



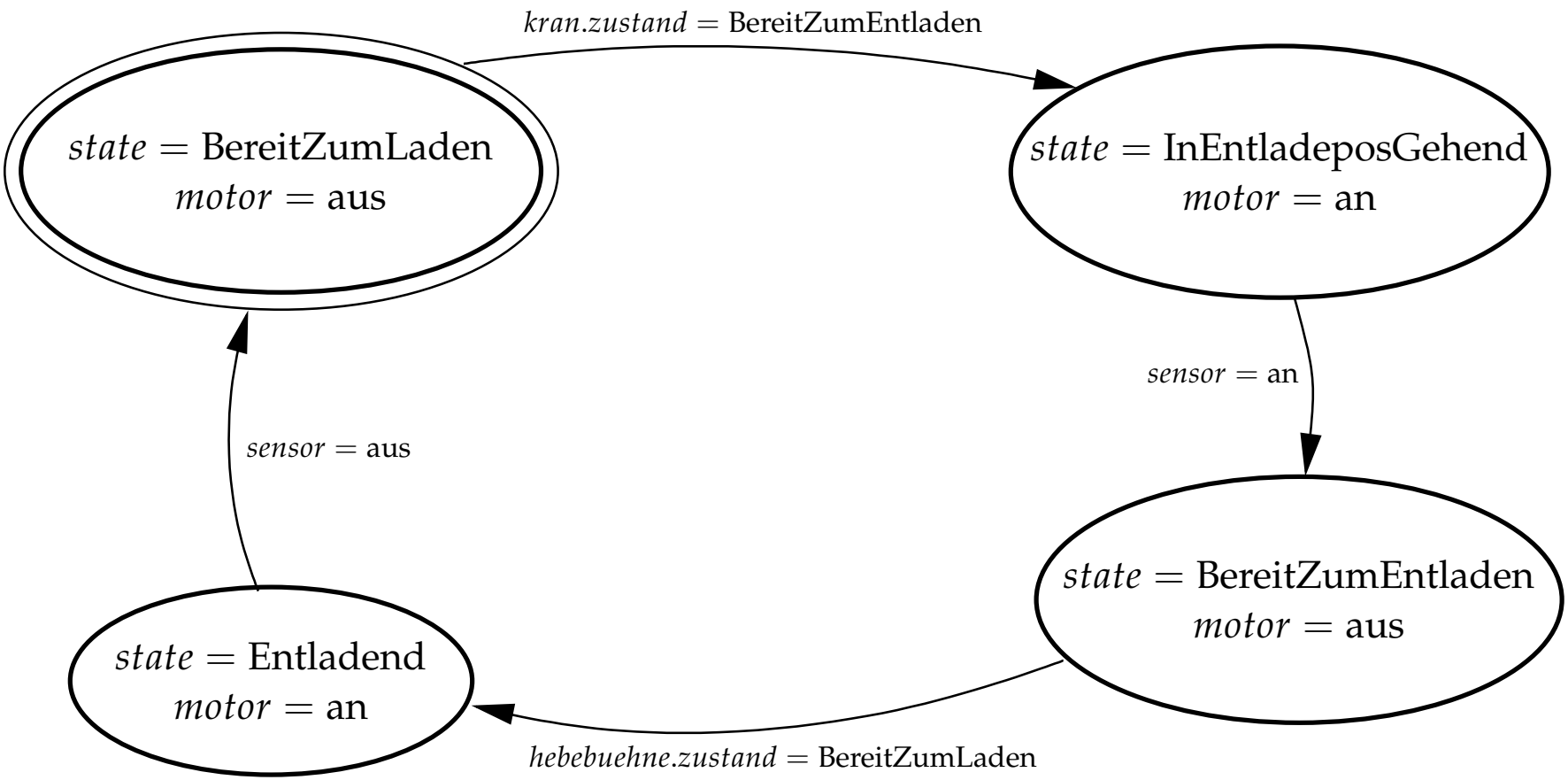
Zufuhrband

Schritt 2: Moore-Automaten in NuSMV übertragen

- Definiere Modul *Zufuhrband*
- Repräsentiere die Zustände des Moore-Automaten durch eine Variable
  $zustand : \{\text{BereitZumLaden}, \text{InEntladeposGehend}, \text{BereitZumEntladen}, \text{Entladend}\}$
- Repräsentiere die Ausgabe des Automaten durch Variable
  $motor : \{\text{an}, \text{aus}\}$
- Repräsentiere die Eingabe des Automaten durch Modulparameter $(sensor, kran, hebebuehne)$

Zufuhrband

Schritt 2: Moore-Automaten in NuSMV übertragen



Zufuhrband

Schritt 2: Moore-Automaten in NuSMV formulieren

```
MODULE Zufuhrband(sensor, kran, hebebuehne)
VAR
    zustand: {BereitZumLaden, InEntladeposGehend,
        BereitZumEntladen, Entladend};
    motor  : {an, aus};
ASSIGN
    next(zustand) := case
        zustand = BereitZumLaden
            kran.zustand = BereitZumEntladen     : InEntladeposGehend;
        zustand = InEntladeposGehend &
            sensor.zustand = an                  : BereitZumEntladen;
        zustand = BereitZumEntladen &
            hebebuehne.zustand = BereitZumEntladen : Entladend;
        zustand = Entladend &
            sensor.zustand = aus                   : BereitZumLaden;
    esac;
```

Zufuhrband

Schritt 2: Moore-Automaten in NuSMV formulieren

```
next(motor) := case
    zustand = BereitZumLaden & motor = aus &
        kran.zustand = BereitZumEntladen     : an;
    zustand = InEntladeposGehend & motor = aus &
        sensor.zustand = an                  : aus;
    zustand = BereitZumEntladen & motor = aus &
        hebebuehne.zustand = BereitZumEntladen : an;
    zustand = Entladend & motor = an &
        sensor.zustand = aus                 : aus;
esac;
```

Zufuhrband

Schritt 2: Moore-Automaten in NuSMV formulieren

```
MODULE Sensor
VAR
    zustand: {an, aus};
ASSIGN
    next(zustand) := {an, aus};

MODULE main
VAR
    zufuhrbandSensor: Sensor;
    kran            : Kran;
    hebebuehne      : Hebebuehne;
    zufuhrband      : Zufuhrband(zufuhrbandSensor, kran, hebebuehne);
```