

# Software Engineering in der Praxis

## Praktische Übungen

# Model Checking II

Matthias Meitner    Marc Spisländer

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

- 1 Inhalt
  
- 2 Nachlese
  - Lernziele der letzten Übung
  
- 3 Einordnung der CTL
  - $CTL^*$  als Obermenge von  $CTL$  und  $LTL$
  
- 4 Die Produktionszelle
  - Strukturierte Vorgehensweise

# 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

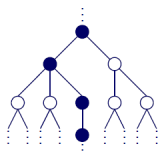
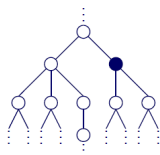
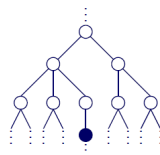
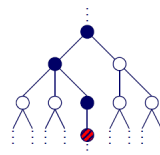
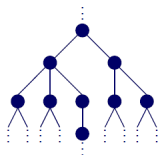
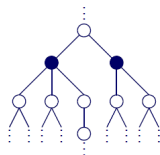
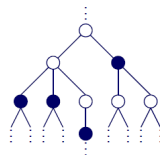
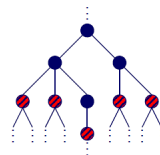


# Modelchecking I

- 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


 $EG\psi$ 

 $EX\psi$ 

 $EF\psi$ 

 $E[\phi U \psi]$ 

 $AG\psi$ 

 $AX\psi$ 

 $AF\psi$ 

 $A[\phi U \psi]$

# Definition der $CTL^*$

## Zustandsformeln, $CTL^*$ Formeln

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

- $\phi$  ist eine Zustandsformel.
- $\phi$  und  $\psi$  Zustandsformeln sind, dann sind auch  $\neg\phi$  und  $\phi \vee \psi$  Zustandsformeln.
- Wenn  $\phi$  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  $\phi$  und  $\psi$  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  $\phi$  und  $\psi$  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)

# *LT*L 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 *LT*L

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

# *LT*L 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 *LT*L

- *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

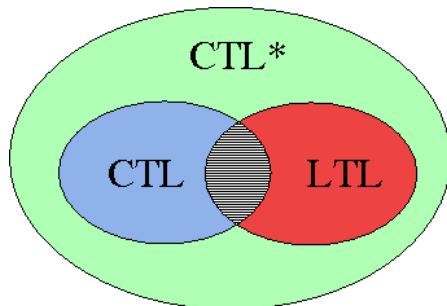


# Übliche Schreibweisen

## Achtung

Manchmal wird *LTL*-Formeln der Allquantor  $A$  vorangestellt.

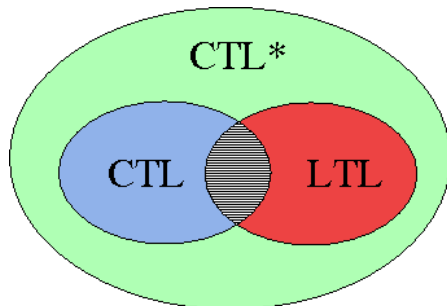
# CTL\*, CTL und LTL



## Beispiele

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

# CTL\*, CTL und LTL



## Beispiele

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

# Moore-Maschinen

## Informelle Beschreibung

Moore-Maschine definiert durch:

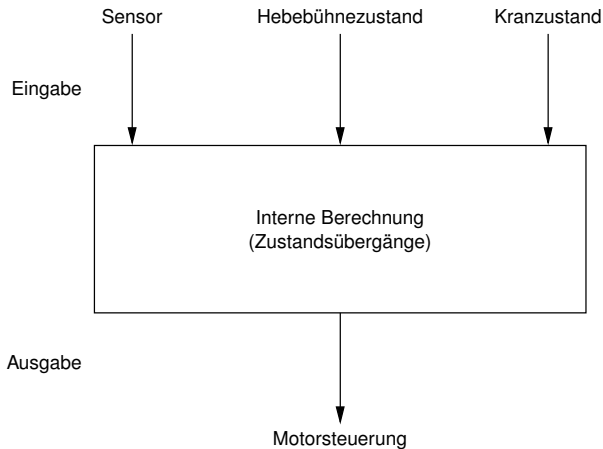
- Endliche Zustandsmenge  $Q$
- Übergangsfunktion  $Q \times \Sigma_i \rightarrow Q$
- Ausgabefunktion  $Q \rightarrow \Sigma_o$

# Die Produktionszelle

## Systematisches Vorgehen

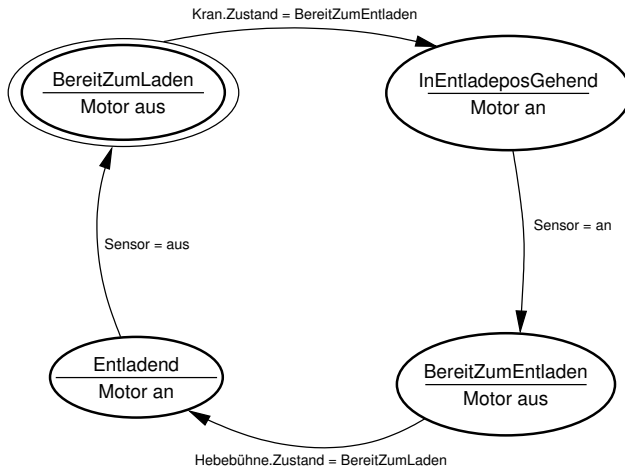
- Formuliere die einzelnen Komponenten (Zufuhrband, Hebebühne, Roboter, . . . ) als Moore-Automat
- Übertrage den Moore-Automaten in NuSMV

# Beispiel: Zufuhrband



# Zufuhrband

## Schritt 1: Moore-Automaten formulieren



# Zufuhrband

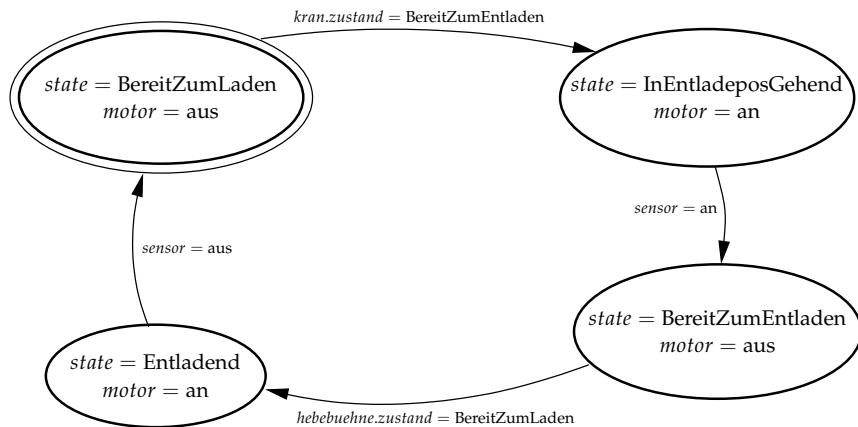
## Schritt 2: Moore-Automaten in NuSMV übertragen

- Definiere Modul *Zufuhrband*
- Repräsentiere die Zustände des Moore-Automaten durch eine Variable  
 $zustand : \{BereitZumLaden, InEntladeposGehend, BereitZumEntladen, Entladend\}$
- Repräsentiere die Ausgabe des Automaten durch Variable  
 $motor : \{an, aus\}$
- 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);
```