

## 7. Modellüberprüfung

*Modell* als abstrakte Beschreibung eines Systems oder Systemteils  
in der Regel dargestellt als *Zustandsübergangssystem* (abstrakter Automat)

*Modellüberprüfung* (engl. *model checking*): Überprüfung (Verifikation), ob ein solches System eine geforderte, formal spezifizierte Eigenschaft hat

*Modell-basiertes Vorgehen* – im Gegensatz zu einem *axiomatischen, deduktiven* Ansatz, d.h. der Ableitung (durch formalen Beweis) einer Formel aus anderen Formeln (z.B. den Axiomen einer Theorie)

*Eigenschaften-orientiertes Vorgehen*: es werden nur gewisse Eigenschaften geprüft – im Gegensatz zu einer Verifikation des gesamten Verhaltens eines Systems

# Modellüberprüfung: Anforderungen

- Geeignete Sprache zur Beschreibung von Modellen (Zustandsübergangssystemen)  
Häufig wird die Eingabesprache der Systeme benutzt, mit deren Hilfe anschließend Modellüberprüfung maschinell durchgeführt wird.
- Sprache zur Beschreibung der geforderten und zu überprüfenden Eigenschaft(en)  
In der Regel werden Eigenschaften formalisiert in der Sprache einer *Temporallogik*

*Erfüllung* einer Eigenschaft durch ein Modell: das Modell “macht die Formel wahr”, die die Eigenschaft beschreibt

vgl. Modellbegriff der bisher behandelten Logiken

# Temporallogik: CTL

Der Wahrheitsbegriff der Temporallogik ist *dynamisch*

– im Gegensatz zu dem statischen Wahrheitsbegriff der normalen Aussagenlogik und Prädikatenlogik: in den verschiedenen Zuständen eines Modells können Aussagen unterschiedliche Wahrheitswerte annehmen

Zeit wird i.a. als *diskret* angenommen – im Gegensatz zur üblicheren Vorstellung von Zeit als einer kontinuierlichen Größe.

Dies entspricht den diskret definierten Zustandsübergängen in Systemen.

Unterschiedliche Zeitbegriffe:

- *Lineare* Zeit: eine lineare (total geordnete) Folge von Zeitpunkten (Zuständen)
- *verzweigte* Zeit (*branching time*): betrachte verschiedene mögliche zukünftige Folgezustände (“Welten”)

zukünftige Zeit hat “Baum-Struktur”

↪ geeignet zur Modellierung von Nicht-Determinismus

Hier (zunächst) benutzte Temporallogik: *CTL – Computation Tree Logic*

# Syntax von CTL

induktive Definition der CTL-Formeln (abstrakte BNF):

$$\begin{aligned} \phi ::= & \perp \mid \top \mid p \mid \\ & (\neg\phi) \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \mid (\phi \Rightarrow \phi) \mid \\ & \mathbf{AX} \phi \mid \mathbf{EX} \phi \mid \\ & \mathbf{AG} \phi \mid \mathbf{EG} \phi \mid \\ & \mathbf{AF} \phi \mid \mathbf{EF} \phi \mid \\ & \mathbf{A} [\phi \mathbf{U} \phi] \mid \mathbf{E} [\phi \mathbf{U} \phi] \end{aligned}$$

$\top$  und  $\perp$  stehen für die Konstanten *wahr* und *falsch*;  
 $p$  steht für eine atomare Formel;  
die aussagenlogischen Verknüpfungen sind die üblichen.

Präzedenz:

$\neg$  und die unären temporalen Verknüpfungen **AX** usw. binden am stärksten,  
gefolgt von binären logischen Operatoren  $\wedge$ ,  $\vee$ ,  $\Rightarrow$ , **AU** und **EU**  
in der angegebenen Reihenfolge

## Syntax von CTL (2)

*Unterformel*: definiert in der üblichen Weise

**AX** usw. sind *temporale Verknüpfungen* – jeweils aus zwei Komponenten zusammengesetzt:

**A**: “entlang aller Pfade” – notwendigerweise, *Always*

**E**: “entlang mindestens eines Pfade” – möglicherweise, *Exists*

**X**: “nächster Zustand” – *neXt*

**F**: “in einem zukünftigen Zustand” – *Future*

**G**: “in allen zukünftigen Zuständen” – *Globally*

**U**: “bis” – *Until*

# Semantik von CTL

Abstrakte Charakterisierung eines Zustandsübergangssystems (*state transition system*) als CTL-Modell:

$\mathcal{M} = (S, \rightarrow, L)$  mit

$S$  Menge der Zustände

$\rightarrow$  Zustandsübergangsrelation (Transitionsrelation), d.h.  $\rightarrow \subseteq S \times S$

Für jedes  $s \in S$  gibt es ein  $s' \in S$  mit  $s \rightarrow s'$  (\*)

$L$  Markierung von Zuständen:

$L : S \rightarrow \mathcal{P}(A)$ , mit  $A$  eine Menge von Atomen

Intuitive Bedeutung der Markierung: jedem Zustand wird eine Menge von Aussagen (eine Konjunktion von Atomen) zugeordnet, die in dem Zustand wahr sind.

## Semantik von CTL (2)

Die Bedingung  $(*)$  stellt sicher, daß es zu jedem Zustand (mindestens) einen Nachfolger-Zustand gibt, d.h. es gibt keinen “deadlock”

Dies kann immer formal erreicht werden durch Einfügen eines neuen Zustands  $s_d$ , einer Transition  $s_d \rightarrow s_d$  (Schleife) und hinreichend vielen Transitionen  $s_i \rightarrow s_d$

Das Verhalten eines Systems wird beschrieben durch die möglichen *Pfade*, d.h. Folgen von Transitionen:

Ein Pfad von  $\mathcal{M}$  ist eine Folge von Zuständen aus  $S$ ,

$$s_1, s_2, \dots, s_i, s_{i+1}, \dots$$

in der für alle  $i \geq 1$  gilt:  $s_{i+1}$  ist ein Nachfolger-Zustand von  $s_i$ , d.h. es gilt jeweils  $s_i \rightarrow s_{i+1}$

Eine Formel  $\phi$  wird von einem Modell  $\mathcal{M}$  in Zustand  $s$  erfüllt:

$$\mathcal{M}, s \models \phi$$

## Semantik von CTL (3)

Definition der Relation  $\models$  induktiv über den Aufbau der CTL-Formeln:

- $\mathcal{M}, s \models \top$  und  $\mathcal{M}, s \not\models \perp$  für alle  $s \in S$
- $\mathcal{M}, s \models p$  genau dann, wenn  $p \in L(s)$
- $\mathcal{M}, s \models \neg\phi$  genau dann, wenn  $\mathcal{M}, s \not\models \phi$
- $\mathcal{M}, s \models \phi_1 \wedge \phi_2$  genau dann, wenn  $\mathcal{M}, s \models \phi_1$  und  $\mathcal{M}, s \models \phi_2$   
Analog für  $\phi_1 \vee \phi_2$ ,  $\phi_1 \Rightarrow \phi_2$  usw.
- $\mathcal{M}, s \models \mathbf{AX} \phi$  genau dann, wenn für jedes  $s'$  mit  $s \rightarrow s'$  gilt  $\mathcal{M}, s' \models \phi$   
**AX**: “in *jedem* Nachfolger-Zustand”
- $\mathcal{M}, s \models \mathbf{EX} \phi$  genau dann, wenn es ein  $s'$  mit  $s \rightarrow s'$  gibt, so daß gilt  $\mathcal{M}, s' \models \phi$   
**EX**: “in *einem* Nachfolger-Zustand”

## Semantik von CTL (4)

- $\mathcal{M}, s \models \mathbf{AG} \phi$  genau dann, wenn für jeden Pfad  $s = s_1 \rightarrow s_2 \rightarrow \dots$  und für jedes  $s_i$  entlang des Pfads gilt  $\mathcal{M}, s_i \models \phi$

**AG:** “für alle Pfade und in *jedem* Zustand entlang des Pfads”

- $\mathcal{M}, s \models \mathbf{EG} \phi$  genau dann, wenn für einen Pfad  $s \rightarrow s_2 \rightarrow \dots$  und jedes  $s_i$  entlang des Pfads gilt  $\mathcal{M}, s_i \models \phi$

**EG:** “es gibt einen Pfad beginnend mit  $s$ , so daß  $\phi$  in *jedem* Zustand entlang des Pfads gilt”

- $\mathcal{M}, s \models \mathbf{AF} \phi$  genau dann, wenn es für jeden Pfad  $s \rightarrow s_2 \rightarrow \dots$  ein  $s_i$  entlang des Pfads gibt, so daß gilt  $\mathcal{M}, s_i \models \phi$

**AF:** “für alle Pfade beginnend mit  $s$  gibt es einen zukünftigen Zustand entlang des Pfads, so daß gilt . . . .”

- $\mathcal{M}, s \models \mathbf{EF} \phi$  genau dann, wenn für einen Pfad  $s \rightarrow s_2 \rightarrow \dots$  und ein  $s_i$  entlang des Pfads gilt  $\mathcal{M}, s_i \models \phi$

**EF:** “es gibt einen Pfad beginnend mit  $s$  und einen zukünftigen Zustand entlang des Pfads, so daß gilt . . . .”

## Semantik von CTL (5)

- $\mathcal{M}, s \models \mathbf{A} [\phi_1 \mathbf{U} \phi_2]$  genau dann, wenn gilt:  
jeder Pfad  $s \rightarrow s_2 \rightarrow \dots$  erfüllt  $\phi_1 \mathbf{U} \phi_2$ , d.h.  
es gibt ein  $s_i$  entlang des Pfads, so daß  $\mathcal{M}, s_i \models \phi_2$ ,  
und  $\mathcal{M}, s_j \models \phi$  für jedes  $s_j$  mit  $j < i$

**AU:** “für alle Pfade beginnend mit  $s$  gilt  $\phi_1$  solange, bis in einem Zustand  $\phi_2$  gilt.”

- $\mathcal{M}, s \models \mathbf{E} [\phi_1 \mathbf{U} \phi_2]$  genau dann, wenn gilt:  
es gibt einen Pfad  $s \rightarrow s_2 \rightarrow \dots$ , der  $\phi_1 \mathbf{U} \phi_2$  erfüllt  
(wie vorher).

**EU:** “es gibt einen Pfad beginnend mit  $s$ , entlang dem  $\phi_1$  solange gilt, bis in einem Zustand  $\phi_2$  gilt.”

# Äquivalenzen von CTL-Formeln

Zwei CTL-Formeln  $\phi$  und  $\psi$  heißen (semantisch) *äquivalent*, falls gilt: wenn in einem Zustand in einem Modell  $\phi$  erfüllt ist, dann ist in dem Zustand auch  $\psi$  erfüllt:

$$\mathcal{M}, s \models \phi \text{ genau dann, wenn } \mathcal{M}, s \models \psi$$

Notation:  $\phi \Leftrightarrow \psi$

Bemerkung: Dies ist völlig analog zum entsprechenden Begriff in Aussagen- bzw. Prädikatenlogik.

1. “deMorgan-Regeln”: **A** und **E** bzw. **G** und **F** können jeweils aufgefaßt werden als universelle bzw. ezistentielle Quantoren über Pfaden bzw. den Zuständen entlang eines Pfads.

Es gelten entsprechende Beziehungen zwischen den Operatoren:

$$\neg \mathbf{AF} \phi \Leftrightarrow \mathbf{EG} \neg \phi$$

$$\neg \mathbf{EF} \phi \Leftrightarrow \mathbf{AG} \neg \phi$$

$$\neg \mathbf{AX} \phi \Leftrightarrow \mathbf{EX} \neg \phi$$

# Weitere CTL-Äquivalenzen

2.

$$\mathbf{AF} \phi \Leftrightarrow \mathbf{A} [\top \mathbf{U} \phi]$$

$$\mathbf{EF} \phi \Leftrightarrow \mathbf{E} [\top \mathbf{U} \phi]$$

3. Die üblichen aussagenlogischen Äquivalenzen gelten auch, wenn die Teilformeln CTL-Formeln sind.

4. “Fixpunkt-Charakterisierung” der temporalen CTL-Operatoren:

$$\mathbf{AG} \phi \Leftrightarrow \phi \wedge \mathbf{AX} \mathbf{AG} \phi$$

$$\mathbf{EG} \phi \Leftrightarrow \phi \wedge \mathbf{EX} \mathbf{EG} \phi$$

$$\mathbf{AF} \phi \Leftrightarrow \phi \vee \mathbf{AX} \mathbf{AF} \phi$$

$$\mathbf{EF} \phi \Leftrightarrow \phi \vee \mathbf{EX} \mathbf{EF} \phi$$

$$\mathbf{A} [\phi \mathbf{U} \psi] \Leftrightarrow \psi \vee (\phi \wedge \mathbf{AX} \mathbf{A} [\phi \mathbf{U} \psi])$$

$$\mathbf{E} [\phi \mathbf{U} \psi] \Leftrightarrow \psi \vee (\phi \wedge \mathbf{EX} \mathbf{E} [\phi \mathbf{U} \psi])$$

# Reduzierte CTL-Operator-Mengen

Aufgrund der angegebenen Beziehungen zwischen den CTL-Operatoren kann deren Menge reduziert werden.

Vgl. in Aussagenlogik: die Menge  $\{\perp, \wedge, \neg\}$  reicht aus, um alle anderen Verknüpfungen (und damit alle aussagenlogischen Formeln) hinzuschreiben.

Umsetzungen in CTL:

$$\mathbf{AX} \quad \rightsquigarrow \quad \neg \mathbf{EX} \neg$$

$$\mathbf{AG} \phi \quad \rightsquigarrow \quad \neg \mathbf{EF} \neg \phi \quad \rightsquigarrow \quad \neg(\mathbf{E} [\top \mathbf{U} \neg \phi])$$

$$\mathbf{EG} \phi \quad \rightsquigarrow \quad \neg \mathbf{AF} \neg \phi \quad \rightsquigarrow \quad \neg(\mathbf{A} [\top \mathbf{U} \neg \phi])$$

$\rightsquigarrow$  die Operatoren **AU**, **EU** und **EX** sind ausreichend.

Andere ausreichende Operator-Mengen:

**EG, EU, EX**

**AG, AU, AX**

**AF, EU, EX**

und andere

# Muster für CTL-Spezifikationen

In CTL-Spezifikationen kommen sich wiederholende Muster für praktisch relevante Fragestellungen vor.

“In jedem Zustand gilt: wenn ein Gerät *angefordert* ist, wird es schließlich auch *bereitgestellt*”:

**AG** (*angefordert*  $\Rightarrow$  **AF** *bereit*)

“*bereit* (z.B. für einen Prozeß) ist auf jedem Pfad unendlich oft wahr”:

**AG** (**AF** *bereit*)

“*beendet* (z.B. für einen Prozeß) wird auf jeden Fall erreicht”:

**AF** (**AG** *beendet*)

## Muster für CTL-Spezifikationen (2)

“Von jedem Zustand aus ist es möglich, den *Start*-Zustand zu erreichen”:

**AG** (**EF** *start*)

“ *q* tritt erst ein (d.h. ein Zustand wird erreicht, in dem *q* wahr ist), wenn auch *p* eintritt”, oder

“Solange nicht *p* eingetreten ist, kann auch nicht *q* eintreten”:

**AG** (**A** [ $\neg q$  **U** *p*])

# Algorithmus für Modell-Überprüfung

Ausgangsfrage: wie kann

$$\mathcal{M}, s \models \psi \quad (*)$$

algorithmisch überprüft werden?

Allgemeinere Fragestellung: finde ein  $s$  oder *alle*  $s$ , für die  $(*)$  gilt.

Hier wird ein Ansatz zur Behandlung des allgemeinen Problems behandelt.

Idee:

- Schrittweise Entwicklung einer Markierung der Zustände in  $\mathcal{M}$  mit denjenigen Teilformeln von  $\psi$ , die in den jeweiligen Zuständen erfüllt sind,
- ausgehend von den atomaren Formeln,
- bis zu einer Markierung mit  $\psi$  selbst.

# Algorithmus für Modell-Überprüfung (2)

Vorgehen:

- Ausgangssituation: gegeben eine Transitionsstruktur  $\mathcal{M} = (S, \rightarrow, L)$  und eine CTL-Formel  $\phi$
- Transformation von  $\phi$  in eine Form, in der nur die reduzierte Menge von Verknüpfungen  $\perp, \wedge, \neg, \mathbf{AF}, \mathbf{EU}, \mathbf{EX}$  benutzt wird
- Schrittweise Markierung der Zustände mit Teilformeln von  $\phi$  entsprechend der folgenden Fallunterscheidung.

$\psi$  eine Teilformel von  $\phi$ ;

Zustände sind mit unmittelbaren Unterformeln von  $\psi$  markiert.

Fallunterscheidung nach der Struktur von  $\psi$ :

$\perp$ : kein Zustand wird mit  $\perp$  markiert.

$p$  (atomar):  $s$  wird mit  $p$  markiert, falls  $p \in L(s)$

## Algorithmus für Modell-Überprüfung (3)

$\psi_1 \wedge \psi_2$ :  $s$  wird mit  $\psi$  markiert, falls  $s$  bereits sowohl mit  $\psi_1$  wie auch mit  $\psi_2$  markiert ist

$\neg\psi_1$ :  $s$  wird mit  $\psi$  markiert, falls  $s$  *nicht* bereits mit  $\psi_1$  markiert ist

**EX**  $\psi_1$ :  $s$  wird mit  $\psi$  markiert, falls einer seiner Nachfolger mit  $\psi_1$  markiert ist.

**AF**  $\psi_1$ :

- Jedes  $s$ , das bereits mit  $\psi_1$  markiert ist, wird mit  $\psi$  markiert.
- Wenn alle Nachfolger-Zustände von  $s$  mit **AF**  $\psi_1$  sind, wird  $s$  selbst auch mit **AF**  $\psi_1$  markiert.
- Dieser Vorgang wird solange wiederholt, bis keine Änderung mehr eintritt.

## Algorithmus für Modell-Überprüfung (4)

**E** [ $\psi_1$  **U**  $\psi_2$ ]:

- Jedes  $s$ , das bereits mit  $\psi_2$  markiert ist, wird mit  $\psi$  markiert.
- $s$  wird mit  $\psi$  markiert, falls  $s$  bereits mit  $\psi_1$  markiert ist und mindestens einer der Nachfolger mit **E** [ $\psi_1$  **U**  $\psi_2$ ] markiert ist.
- Dieser Vorgang wird solange wiederholt, bis keine Änderung mehr eintritt.

Wenn dieser Prozeß für  $\phi$  abgeschlossen ist, können alle Zustände, die  $\phi$  erfüllen, abgelesen werden.

*Modifikation/Variante:*

Direkte Behandlung von **EG**  $\psi_1$ :

- Alle Zustände  $s$  werden mit **EG**  $\psi_1$  markiert.
- Markierung mit **EG**  $\psi_1$  wird für einen Zustand  $s$  *gestrichen*, falls keiner der Nachfolger von  $s$  mit  $\psi_1$  markiert ist.

Solange wiederholen, bis keine Änderung mehr eintritt.

# Etwas Fixpunkt-Theorie

Das Folgende gilt auch in allgemeinerem Kontext; wir beschränken uns hier auf mengenwertige Funktionen.

$S$  Menge (z.B. von Zuständen)

$F : \mathcal{P}(S) \rightarrow \mathcal{P}(S)$  Funktion auf der Potenzmenge von  $S$

$F$  heißt *monoton*, wenn für beliebige Teilmengen  $X, Y$  von  $S$  gilt

$$X \subseteq Y \Rightarrow F(X) \subseteq F(Y)$$

$X$  ist *Fixpunkt* von  $F$ , falls gilt  $F(X) = X$

Ein Fixpunkt ist *größter* bzw. *kleinster* Fixpunkt von  $S$ , wenn er größer bzw. kleiner als jeder andere Fixpunkt von  $S$  ist.

Notation:  $F^0(X) := X$        $F^{n+1}(X) := F(F^n(X))$

## Etwas Fixpunkt-Theorie (2)

“Knaster-Tarski-Theorem” :

Ist  $F : \mathcal{P}(S) \rightarrow \mathcal{P}(S)$  eine monotone Funktion, dann ist

$F^\infty(\emptyset)$  der kleinste Fixpunkt

$F^\infty(S)$  der größte Fixpunkt

von  $F$ .

Für eine endliche Menge  $S := \{s_0, \dots, s_n\}$  reicht es,  $F^{n+1}$  für die Fixpunkte zu nehmen.

Der Satz gibt eine Berechnungsvorschrift für kleinste bzw. größte Fixpunkte über endlichen Mengen und garantiert deren Terminierung.

# CTL-Operatoren und Fixpunkte

Am Beispiel **EG** – zu zeigen die Äquivalenz:  $\mathbf{EG} \phi \Leftrightarrow \phi \wedge \mathbf{EX} \mathbf{EG} \phi$

Notation:  $\langle \phi \rangle$  Menge der Zustände (bezgl. eines festen Modells), die  $\phi$  erfüllen.

$$\langle \mathbf{EX} \psi \rangle = \{s \mid \exists s'. s \rightarrow s' \wedge s' \in \langle \psi \rangle\}$$

damit:

$$\langle \mathbf{EG} \phi \rangle = \langle \phi \rangle \cap \{s \mid \exists s'. s \rightarrow s' \wedge s' \in \langle \mathbf{EG} \phi \rangle\}$$

d.h.  $\mathbf{EG} \phi$  ist Fixpunkt der Funktion

$$F(X) = \langle \phi \rangle \cap \{s \mid \exists s'. s \rightarrow s' \wedge s' \in X\}$$

Satz:  $F$  wie oben,  $S$  mit  $n$  Elementen

1.  $F$  ist monoton.
2.  $\langle \mathbf{EG} \phi \rangle$  ist größter Fixpunkt von  $F$ .
3.  $\langle \mathbf{EG} \phi \rangle = F^{n+1}(S)$

Aus dem Satz folgt die Korrektheit für die Funktion  $SAT_{EG}$ , die den größten Fixpunkt von  $F$  berechnet.

# Pseudo-Code für Modell-Überprüfung

**function** SAT ( $\phi$ )

**return** – Menge der Zustände, die  $\phi$  erfüllen

**case**  $\phi$  **of**

$\perp$  :  $\emptyset$

$\top$  :  $S$

atomic :  $\{s \in S \mid \phi \in L(s)\}$

$\neg\phi_1$  :  $S \setminus \text{SAT}(\phi_1)$

$\phi_1 \wedge \phi_2$  :  $\text{SAT}(\phi_1) \cap \text{SAT}(\phi_2)$

$\phi_1 \vee \phi_2$  :  $\text{SAT}(\phi_1) \cup \text{SAT}(\phi_2)$

$\phi_1 \Rightarrow \phi_2$  :  $\text{SAT}(\neg\phi_1 \vee \phi_2)$

**AX**  $\phi_1$  :  $\text{SAT}(\neg\mathbf{EX} \neg\phi_1)$

**EX**  $\phi_1$  :  $\text{SAT}_{EX}(\phi_1)$

**AG**  $\phi_1$  :  $\text{SAT}(\neg\mathbf{EF} \neg\phi_1)$

**EG**  $\phi_1$  :  $\text{SAT}(\neg\mathbf{AF} \neg\phi_1)$

## Pseudo-Code für Modell-Überprüfung (2)

**AF**  $\phi_1$  :  $SAT_{AF}(\phi_1)$

**EF**  $\phi_1$  :  $SAT(\mathbf{E} [\top \mathbf{U} \phi_1])$

**A** [ $\phi_1 \mathbf{U} \phi_2$ ]:  $SAT(\neg(\mathbf{E} [\neg\phi_2 \mathbf{U} (\neg\phi_1 \wedge \neg\phi_2)]) \vee \mathbf{EG} \neg\phi_2)$

**E** [ $\phi_1 \mathbf{U} \phi_2$ ]:  $SAT_{EU}(\phi_1, \phi_2)$

**end case**

**function**  $SAT_{EX}(\phi)$

– bestimmt die Menge der Zustände, die **EX**  $\phi$  erfüllen

**local var**  $X, Y$

**begin**

$X := SAT(\phi)$

$Y := \{s_0 \in S \mid s_0 \rightarrow s_1 \text{ für ein } s_1 \in X\}$

**return**  $Y$

**end**

## Pseudo-Code für Modell-Überprüfung (3)

**function**  $SAT_{AF}(\phi)$

– bestimmt die Menge der Zustände, die **AF**  $\phi$  erfüllen

**local var**  $X, Y$

**begin**

$X := S$

$Y := SAT(\phi)$

**repeat until**  $X = Y$

$X := Y$

$Y := Y \cup \{s \mid \text{für alle } s' \text{ mit } s \rightarrow s' \text{ gilt } s' \in Y\}$

**end repeat**

**return**  $Y$

**end**

## Pseudo-Code für Modell-Überprüfung (4)

**function**  $\text{SAT}_{EU}(\phi, \psi)$

– bestimmt die Menge der Zustände, die  $\mathbf{E}[\phi \mathbf{U} \psi]$  erfüllen

**local var**  $V, X, Y$

**begin**

$V := \text{SAT}(\phi)$

$X := S$

$Y := \text{SAT}(\psi)$

**repeat until**  $X = Y$

$X := Y$

$Y := Y \cup (V \cap \{s \mid \text{es gibt ein } s' \text{ mit } s \rightarrow s' \text{ und } s' \in Y\})$

**end repeat**

**return**  $Y$

**end**