

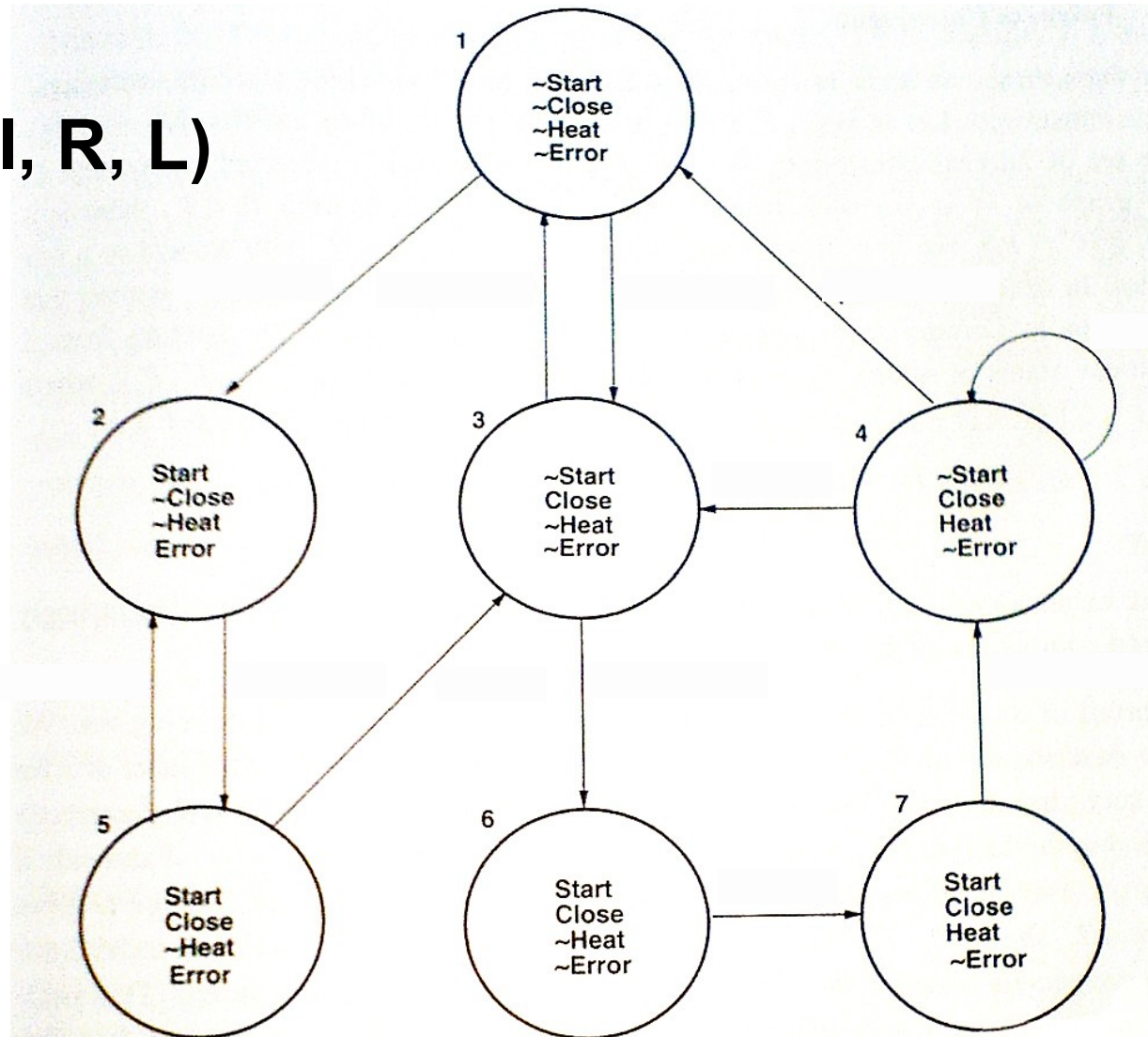
# CTL Model Checking

# Einführung/Historie

- ♦ Model Checking ist ...
- ♦ nur reaktive Systeme werden betrachtet
- ♦ vor CTL Model Checking gab es ...
- ♦ Queille, Sifakis, Clarke, Emerson und Sistla (1982)

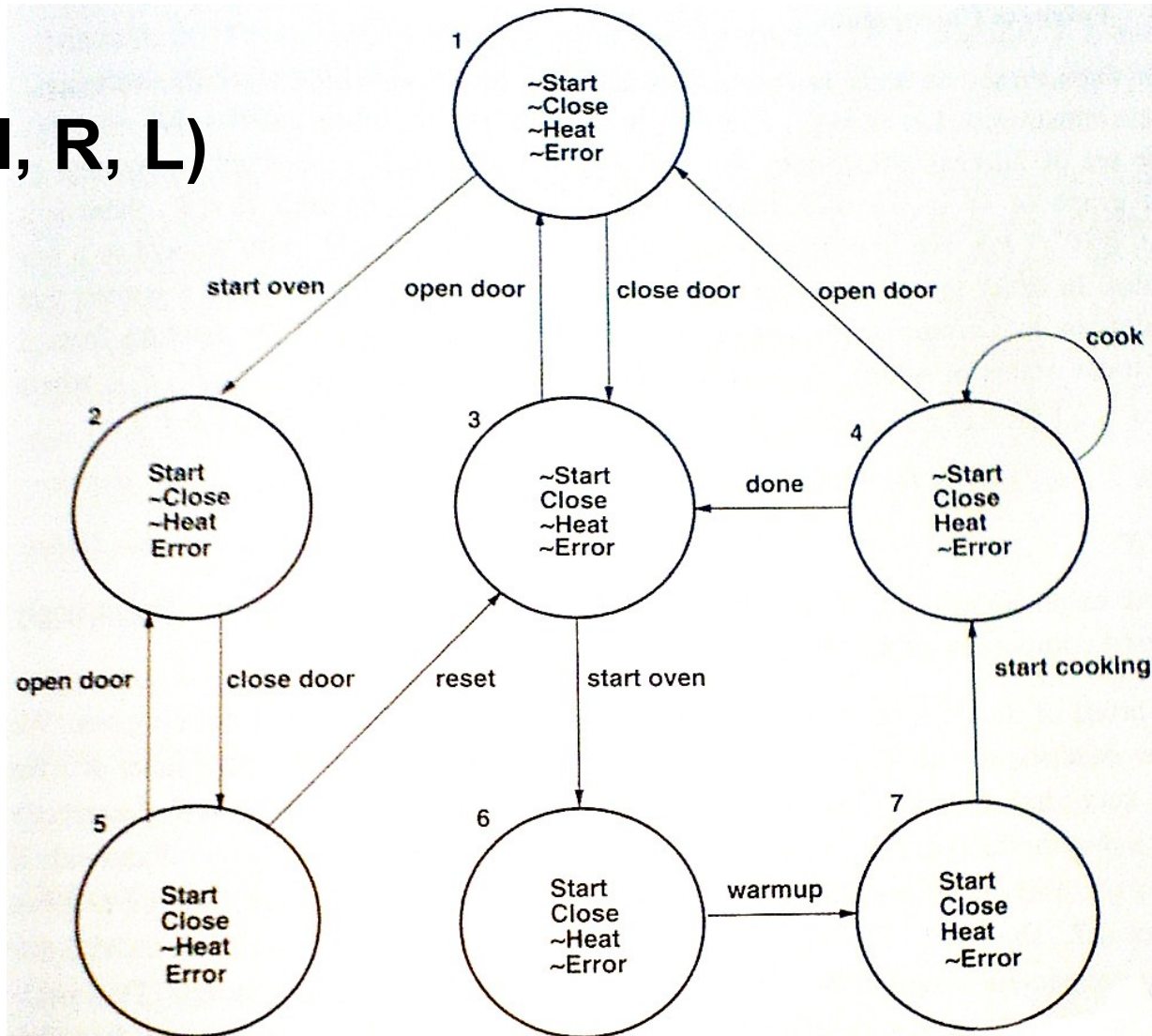
# Crash Course: Kripke-Struktur

$M = (S, I, R, L)$



# Crash Course: Kripke-Struktur

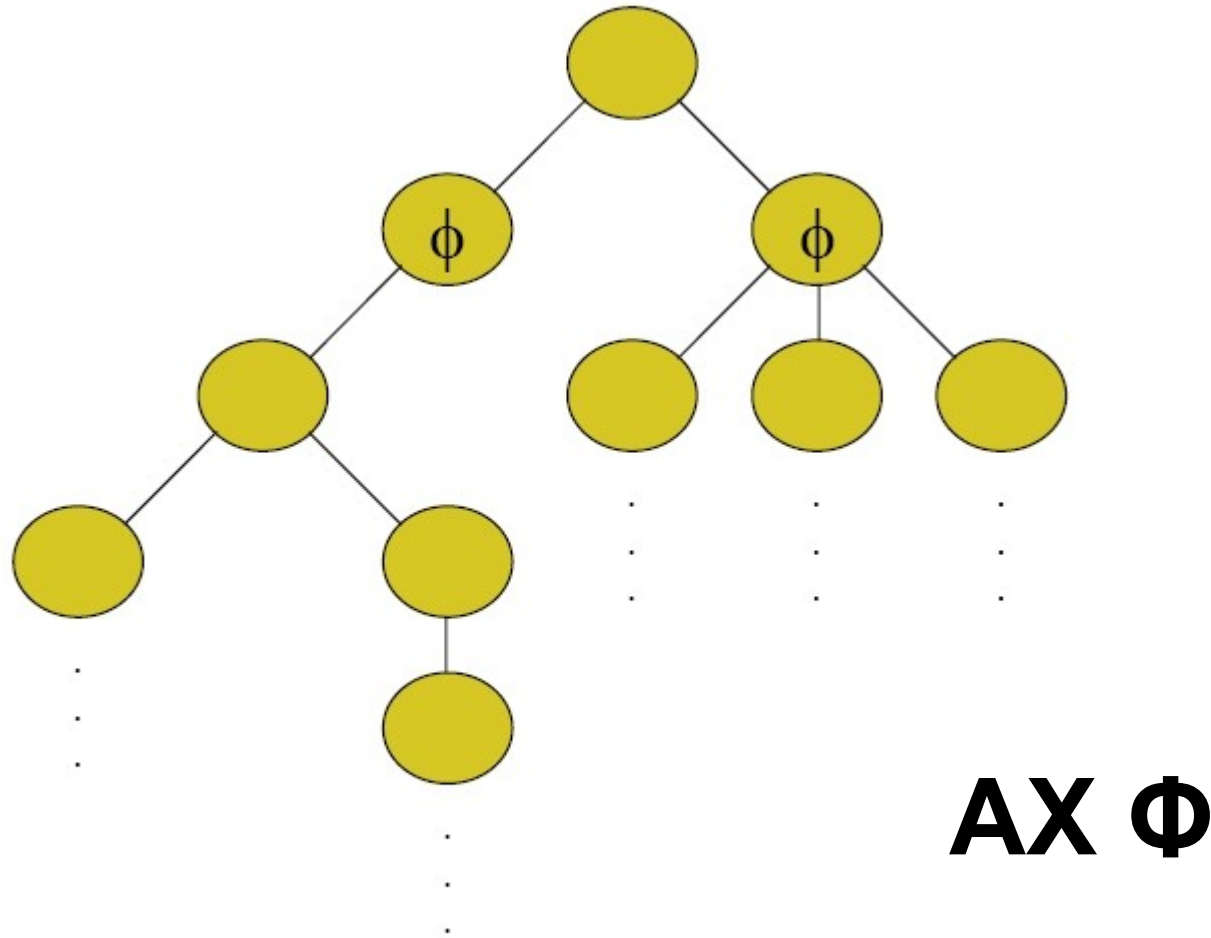
$M = (S, I, R, L)$



# Crash Course: CTL

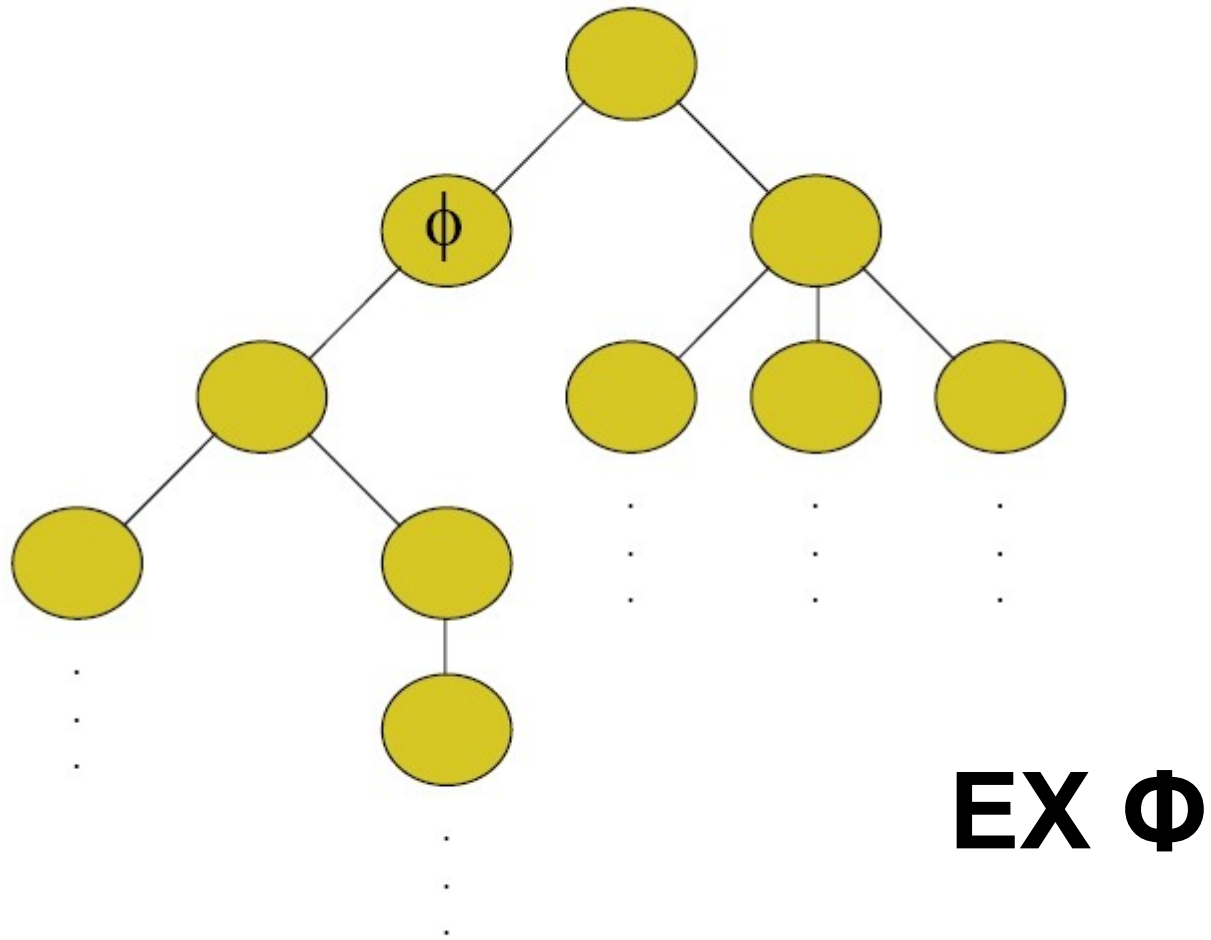
- Aussagenlogik + Zeitoperatoren = CTL
- Pfadoperatoren: **A, E**
- Zustandsoperatoren: **X, F, G, U**

# Crash Course: CTL



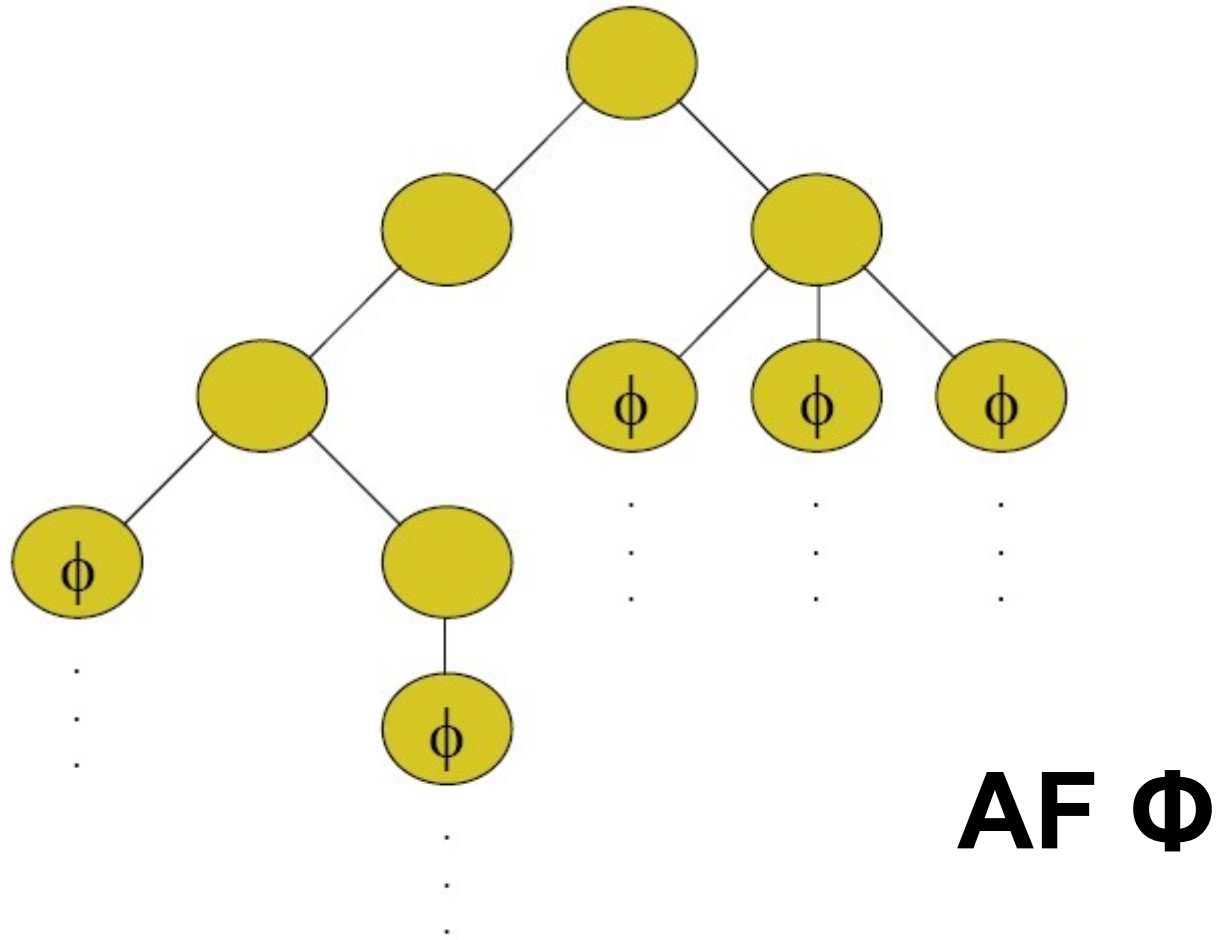
CTL / LTL - Lars Biermann

# Crash Course: CTL



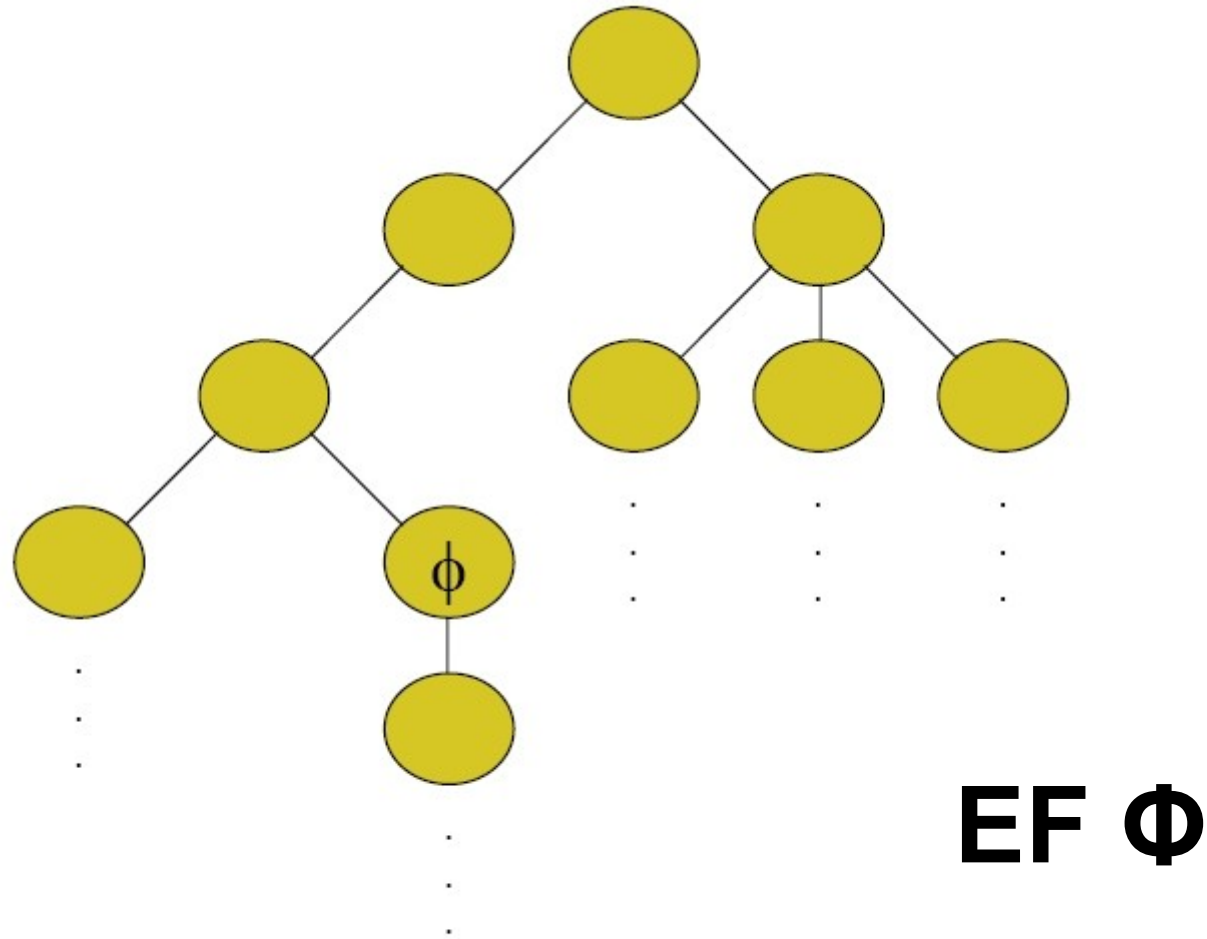
CTL / LTL - Lars Biermann

# Crash Course: CTL



CTL / LTL - Lars Biermann

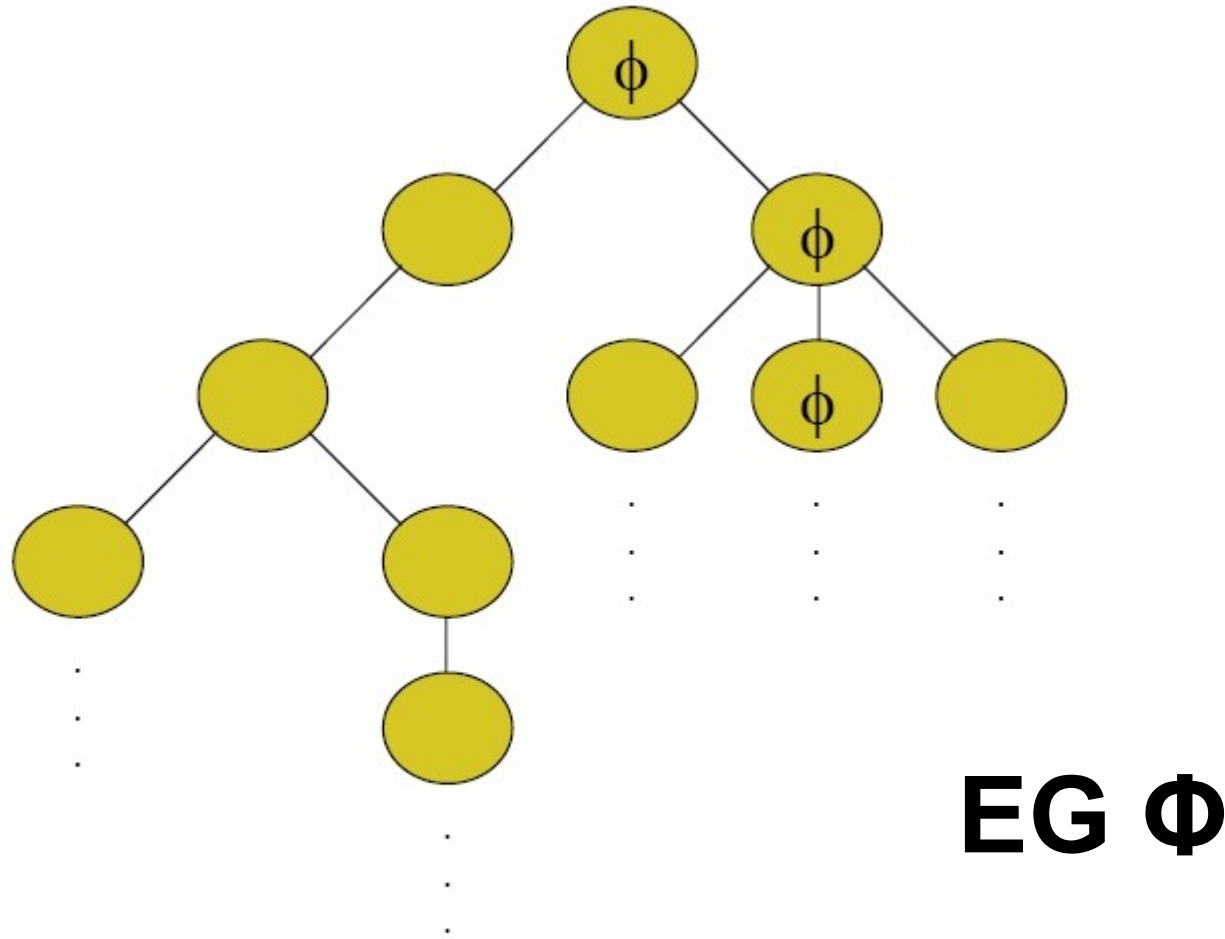
# Crash Course: CTL



CTL / LTL - Lars Biermann

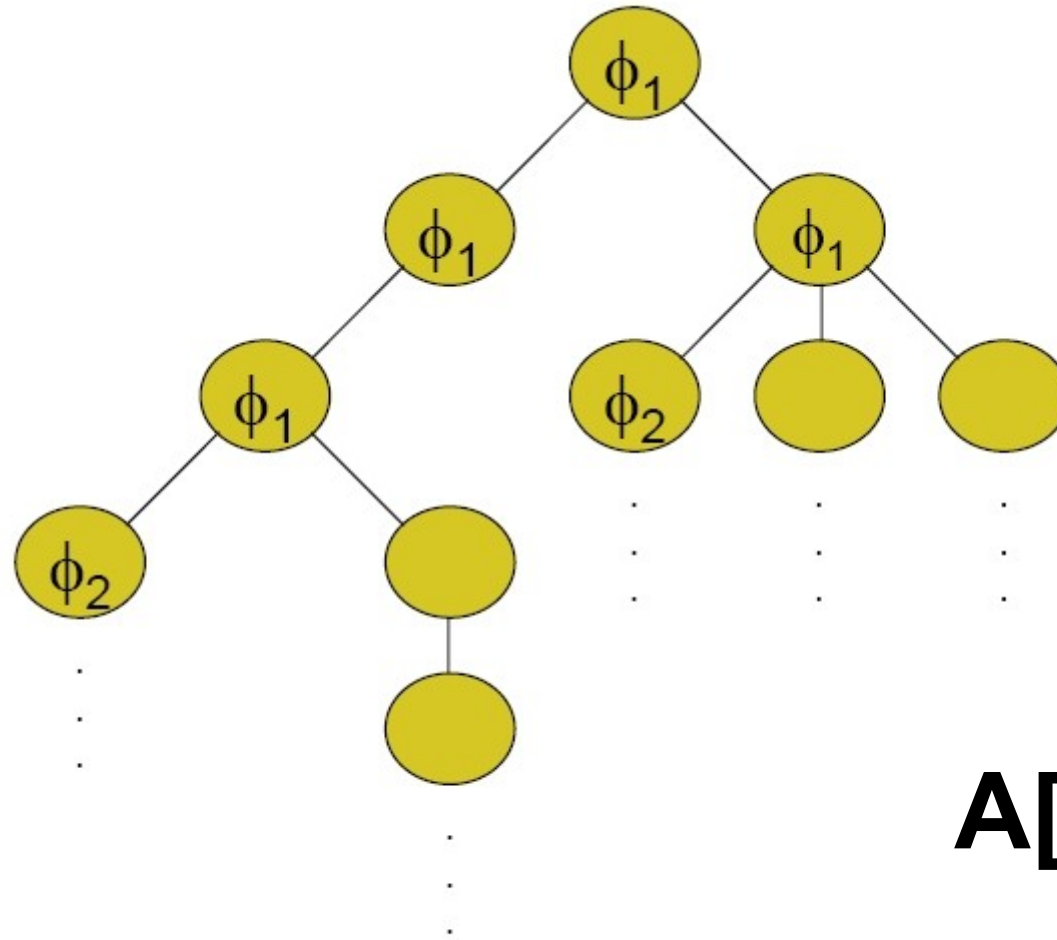


# Crash Course: CTL



CTL / LTL - Lars Biermann

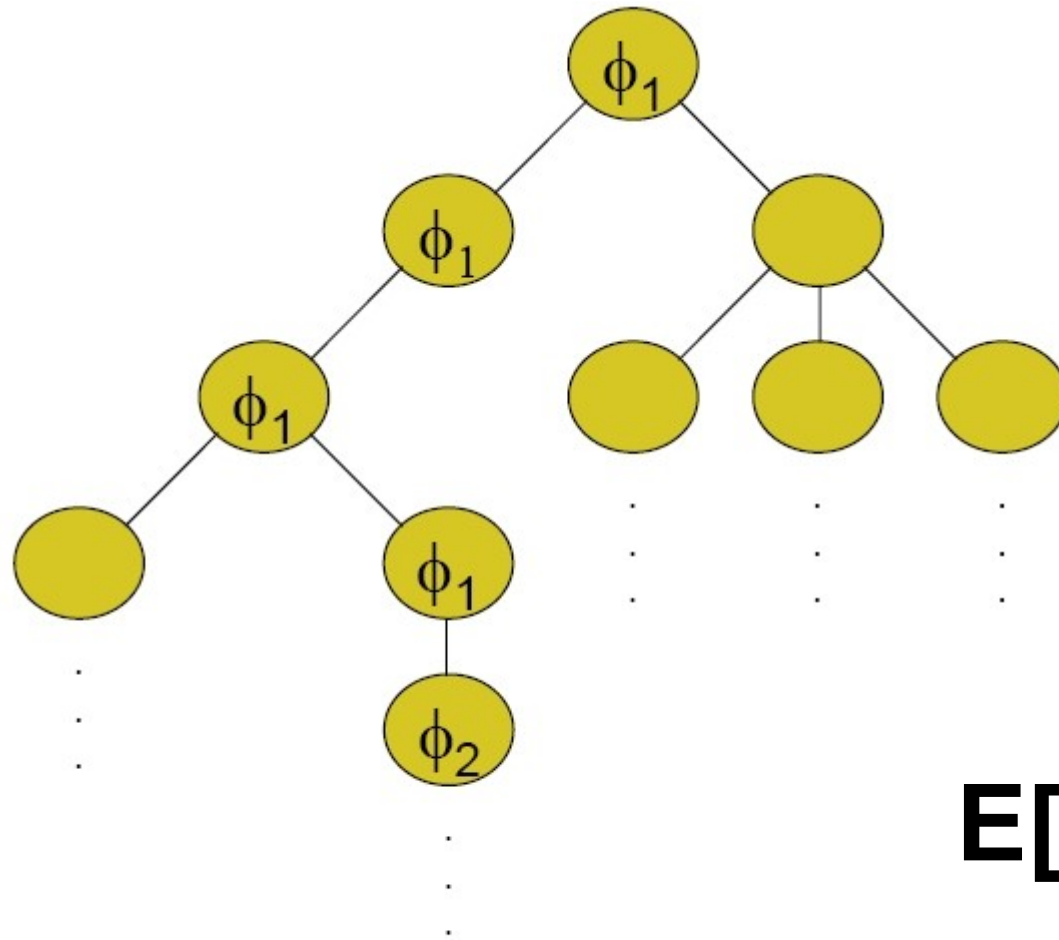
# Crash Course: CTL



**$A[\Phi_1 U \Phi_2]$**

CTL / LTL - Lars Biermann

# Crash Course: CTL



**$E[\Phi_1 U \Phi_2]$**

CTL / LTL - Lars Biermann

# Crash Course: CTL

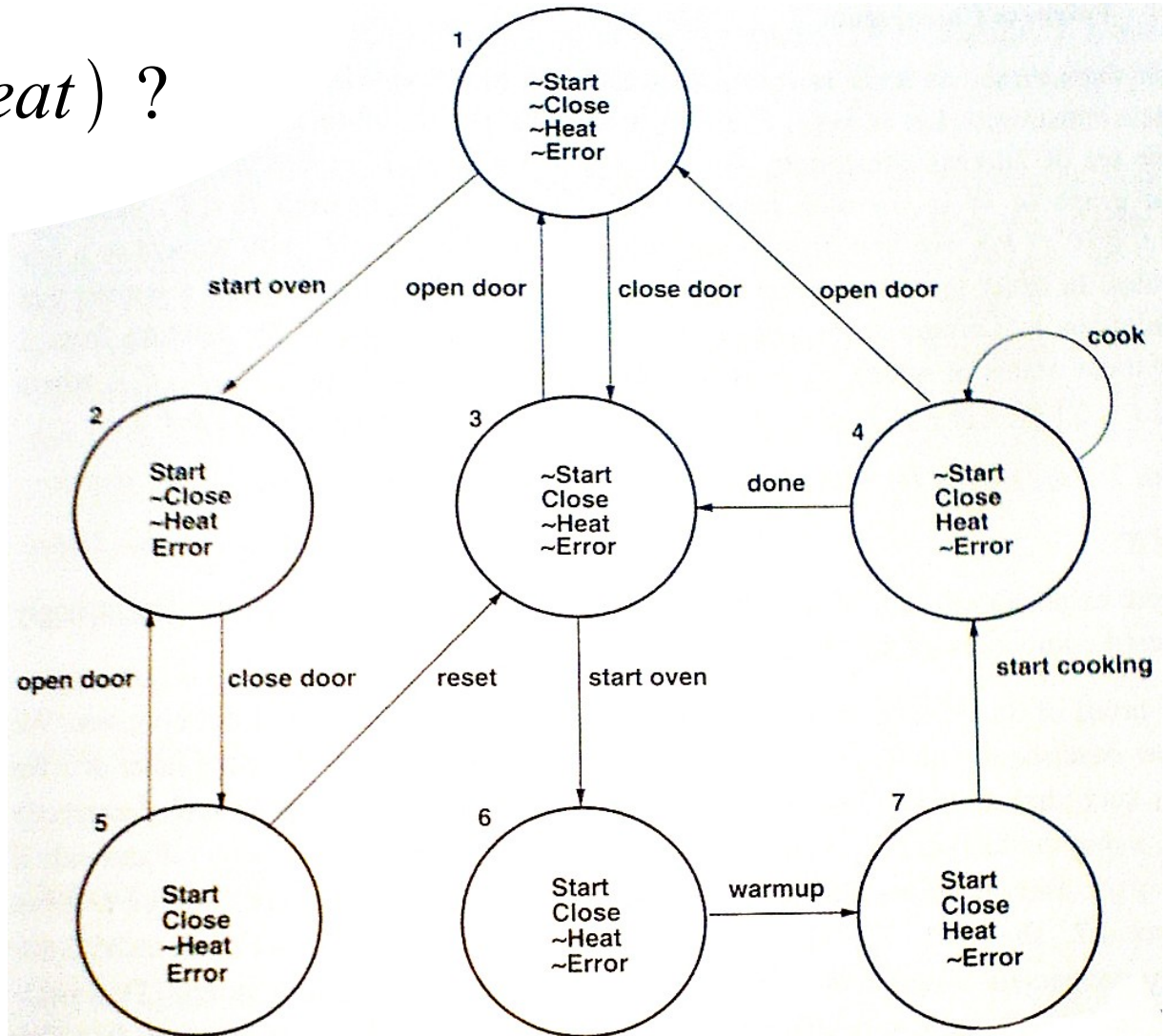
- ♦ Aussagenlogik + Zeitoperatoren = CTL
- ♦ Pfadoperatoren: **A**, **E**
- ♦ Zustandsoperatoren: **X**, **F**, **G**, **U**
- ♦ Die Menge  $\neg$ ,  $\wedge$ , **EX**, **EU**, **EG** reicht aus

# Einfaches Beispiel

Gilt  $\mathbf{AG}(Start \rightarrow \mathbf{AF} Heat)$  ?

$\downarrow$   
 $\neg \mathbf{E}[true \mathbf{U}(Start \wedge \mathbf{EG} \neg Heat)]$

$\downarrow$   
 $\neg \mathbf{EF}(Start \wedge \mathbf{EG} \neg Heat)$



# Der Algorithmus

- **Input** = Modell + Spezifikation
- $f := \Phi_1 \wedge \Phi_2 \wedge \Phi_3 \wedge \Phi_4 \wedge \Phi_5$
- welche Zustände erfüllen die Formel
- ist Startzustand dabei: erfüllt
- keine Ausführungsverfolgung - nur Zustände

# Der Algorithmus

1. AG ( Start  $\rightarrow$  AF Heat )
2. AG ( Start  $\rightarrow$  AF Heat )
3. AG ( **Start**  $\rightarrow$  AF Heat )
4. AG ( **Start**  $\rightarrow$  AF Heat )
5. AG ( **Start**  $\rightarrow$  AF Heat )

# Der Algorithmus

- ♦ minimale Menge:  $\neg$ ,  $\wedge$ , **EX**, **EU**, **EG**
  1. atomare Formel:  $f = \Phi$ , TRUE, FALSE
  2. Negation:  $f = \neg\Phi$
  3. Konjunktion:  $f = \Phi_1 \wedge \Phi_2$
  4.  $f = \mathbf{EX} \Phi$
  5.  $f = \mathbf{E}[\Phi_1 \mathbf{U} \Phi_2]$
  6.  $f = \mathbf{EG} \Phi$

# Der Algorithmus

1. atomare Formel: Markierung schon in  $L(s)$
2. **procedure** *CheckNegation*( $f$ )  
    **for all**  $s \in \{s \mid f \notin \text{label}(s)\}$   
        **do**  $\text{label}(s) := \text{label}(s) \cup \{\neg f\}$   
    **end procedure**
3. **procedure** *CheckConjunction*( $f_1, f_2$ )  
    **for all**  $s \in \{s \mid f_1 \in \text{label}(s) \wedge f_2 \in \text{label}(s)\}$   
        **do**  $\text{label}(s) := \text{label}(s) \cup \{f_1 \wedge f_2\}$   
    **end procedure**
4. **EX** (für Vortrag nicht benötigt)

# Der Algorithmus

5. **procedure** *CheckEU* ( $f_1, f_2$ )  
     $T := \{s \mid f_2 \in \text{label}(s)\}$   
    **for all**  $s \in T$  **do**  $\text{label}(s) := \text{label}(s) \cup \{\mathbf{E}[f_1 \mathbf{U} f_2]\}$   
    **while**  $T \neq \emptyset$  **do**  
        **choose**  $s \in T$   
         $T := T \setminus \{s\}$   
        **for all**  $t$  **such that**  $R(t, s)$  **do**  
            **if**  $\mathbf{E}[f_1 \mathbf{U} f_2] \notin \text{label}(t)$  **and**  $f_1 \in \text{label}(t)$  **then**  
                 $\text{label}(t) := \text{label}(t) \cup \{\mathbf{E}[f_1 \mathbf{U} f_2]\}$   
                 $T := T \cup \{t\}$   
            **end if**  
        **end for all**  
    **end while**  
**end procedure**

# Der Algorithmus

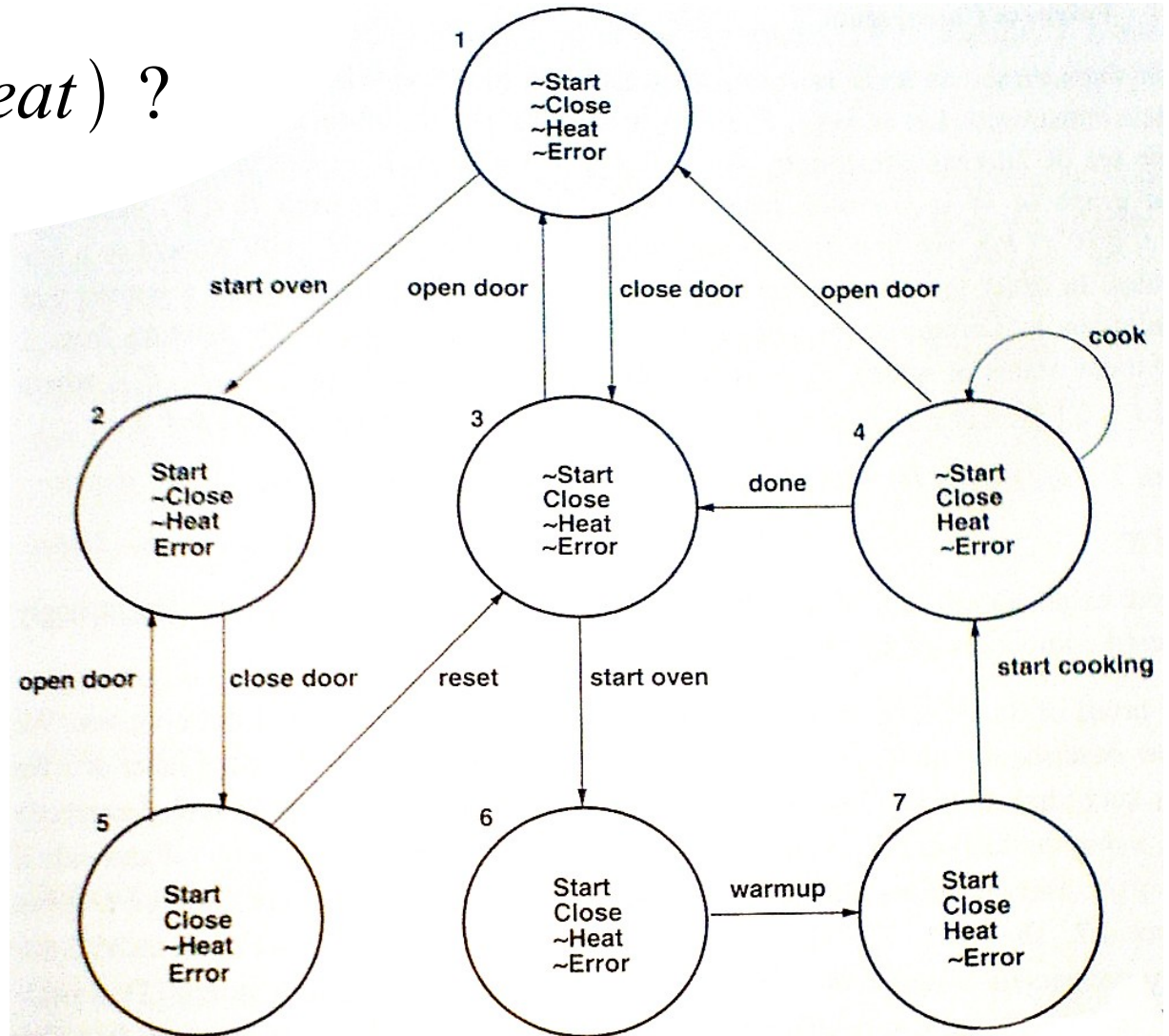
6. **procedure** *CheckEG*( $f_1$ )  
     $S' := \{s \mid f_1 \in \text{label}(s)\}$   
     $SCC := \{C \mid C \text{ is a nontrivial SCC of } S'\}$   
     $T := \cup_{C \in SCC} \{s \mid s \in C\}$   
    **for all**  $s \in T$  **do**  $\text{label}(s) := \text{label}(s) \cup \{\mathbf{EG} f_1\}$   
    **while**  $T \neq \emptyset$  **do**  
        **choose**  $s \in T$   
         $T := T \setminus \{s\}$   
        **for all**  $t$  **such that**  $t \in S'$  **and**  $R(t, s)$  **do**  
            **if**  $\mathbf{EG} f_1 \notin \text{label}(t)$  **then**  
                 $\text{label}(t) := \text{label}(t) \cup \{\mathbf{EG} f_1\}$   
                 $T := T \cup \{t\}$   
            **end if**  
        **end for all**  
    **end while**  
**end procedure**

# Der Algorithmus am Beispiel

Gilt  $AG(Start \rightarrow AF Heat)$  ?

$\downarrow$   
 $\neg E[true U (Start \wedge EG \neg Heat)]$

$\downarrow$   
 $\neg EF(Start \wedge EG \neg Heat)$



# Ausblick

- ♦ CTL Model Checking: 1982
- ♦ Kripke-Struktur zu groß
- ♦ 200 Variablen ->  $10^{28}$  Zustände
- ♦ Lösung: Zustände -> Formeln
- ♦ 1992: BDDs und SAT solver

# Fragen

- ♦ was ist ... ?
- ♦ woher kommt ... ?
- ♦ was, wenn man ... ?
- ♦ aber da war doch noch ... ?
- ♦ ist das jetzt richtig wenn man da ... ?
- ♦ gibt's da auch was von ratiopharm® ... ?

# Vielen Dank

# Quellen

- ♦ [en.wikipedia.org/wiki/Computational\\_tree\\_logic](https://en.wikipedia.org/wiki/Computational_tree_logic)
- ♦ [en.wikipedia.org/wiki/Temporal\\_logic](https://en.wikipedia.org/wiki/Temporal_logic)
- ♦ [en.wikipedia.org/wiki/Model\\_checking](https://en.wikipedia.org/wiki/Model_checking)
- ♦ **Clarke, Grumberg, Peled: Model Checking.**  
MIT Press, 3. Auflage, 2001
- ♦ CTL-Vortrag von **Lars Biermann**
- ♦ Automatic Verification of Finite State Concurrent Systems using temporal logic specifications