SCIENCE
PASSION
TECHNOLOGY

Model Checking for CTL

Bettina Könighofer
bettina.koenighofer@tugraz.at

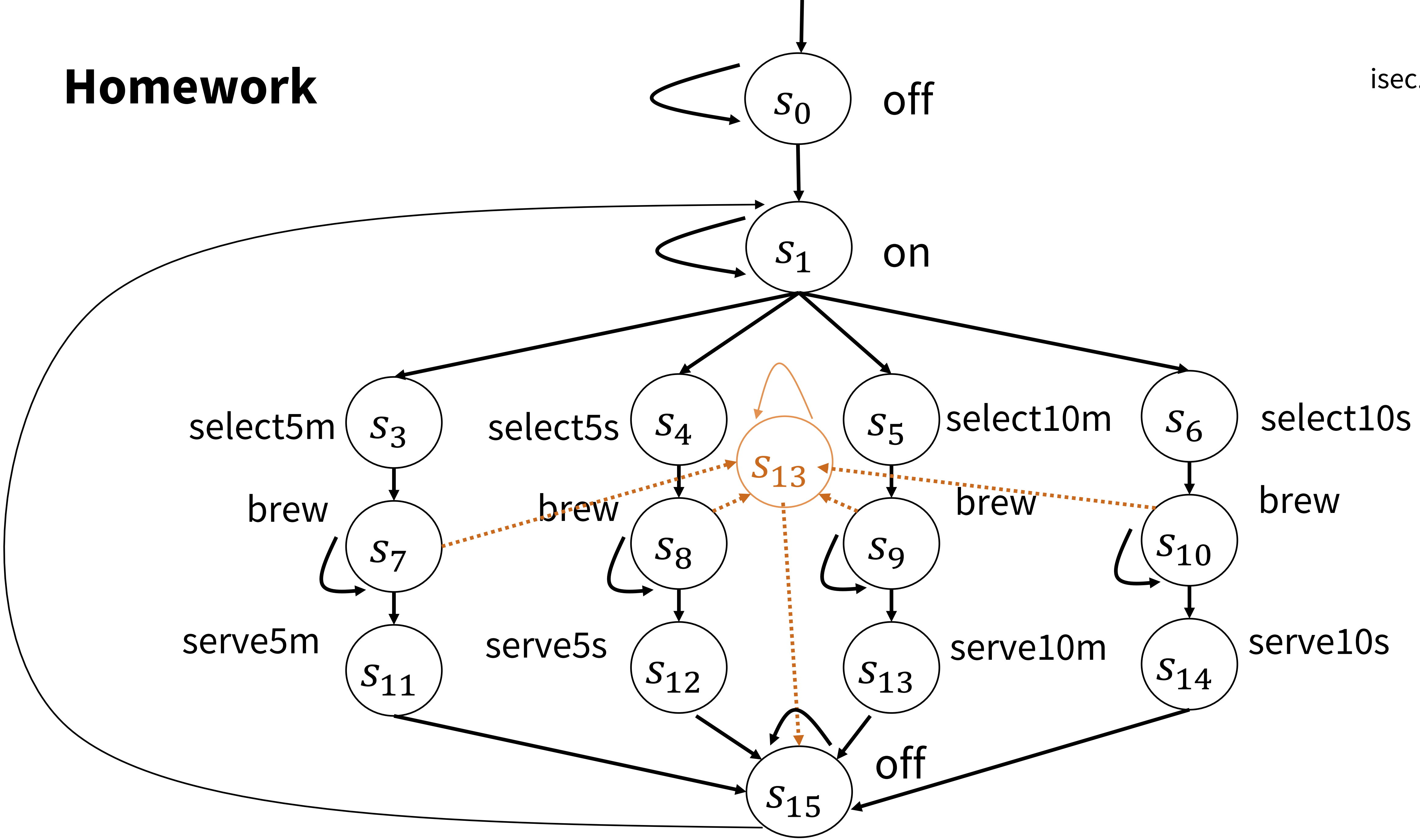
Plan for Today

- Presentation of Homework
- Properties of CTL and LTL
- CTL Model Checking

Homework

- Task 5a: Draw a Kripke Structure to model the coffee machine:
  1. Initially, the brewer is in the off state until it is switched on.
  2. Once the brewer is switched on, the user can select the number of cups and the strength of the coffee. The user can choose either five or ten cups, with a strength of either medium or strong.
  3. After the selections are made, the coffee machine starts brewing.
  4. During brewing, if an error is detected, the brewer enters an error state.
  5. Alternatively, the brewer may complete the brewing process and serve the coffee.
  6. After serving or entering the error state, the coffee machine can be turned off, ready to be turned on again later.

Homework



s_0
off
s_1
on
select5m
s_3
select5s
s_4
s_{13}
s_5
select10m
s_6
select10s
brew
s_7
brew
s_8
s_9
brew
s_{10}
brew
serve5m
s_{11}
serve5s
s_{12}
s_{13}
serve10m
s_{14}
serve10s
off
s_{15}

Homework: Translate sentences in temporal logic

1. The error state is always eventually reachable.

$$AGEF\ (err)$$

2. Ten cups of coffee are always eventually served.

$$AGF\ (serve10m \vee serve10s)$$

3. It is always possible to select ten cups of coffee, and once selected, ten cups will always eventually be served, unless an error occurs.

$$serve10 \coloneqq serve10m \vee serve10s$$
$$select10 \coloneqq select10m \vee select10s$$

$$AG\ (select10) \wedge AG(select10 \rightarrow AF(serve\ 10 \vee error))$$

Homework: Translate sentences in temporal logic

4. The error state may never be reached.

$$EG(\neg err)$$

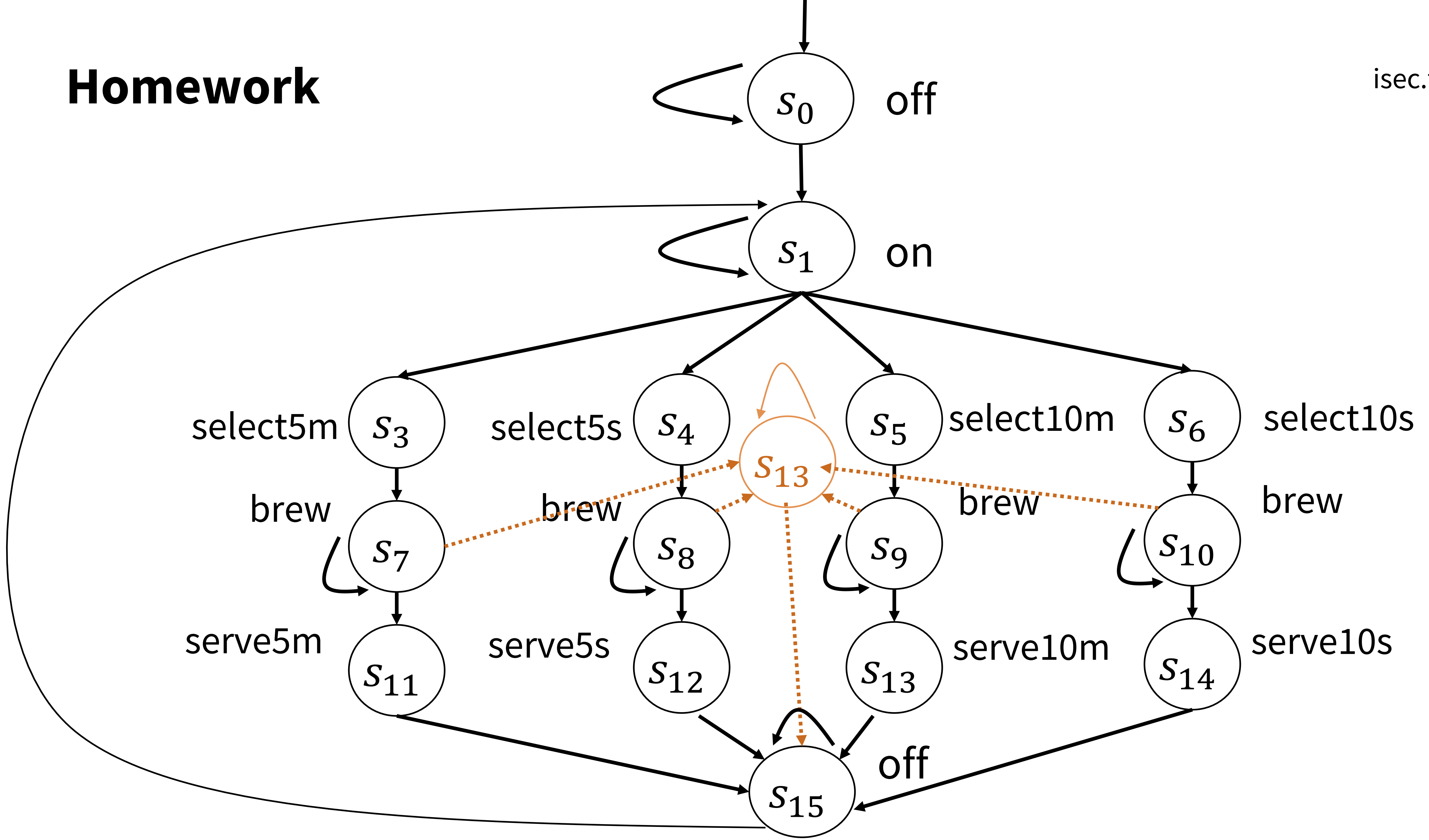
5. It is not possible for the machine to serve ten cups of coffee in the current time step and then serve five more cups in the next time step

$$\neg EF\ (serve10 \rightarrow Xserve5)$$

6. The selected amount of coffee will be served in the next time step.

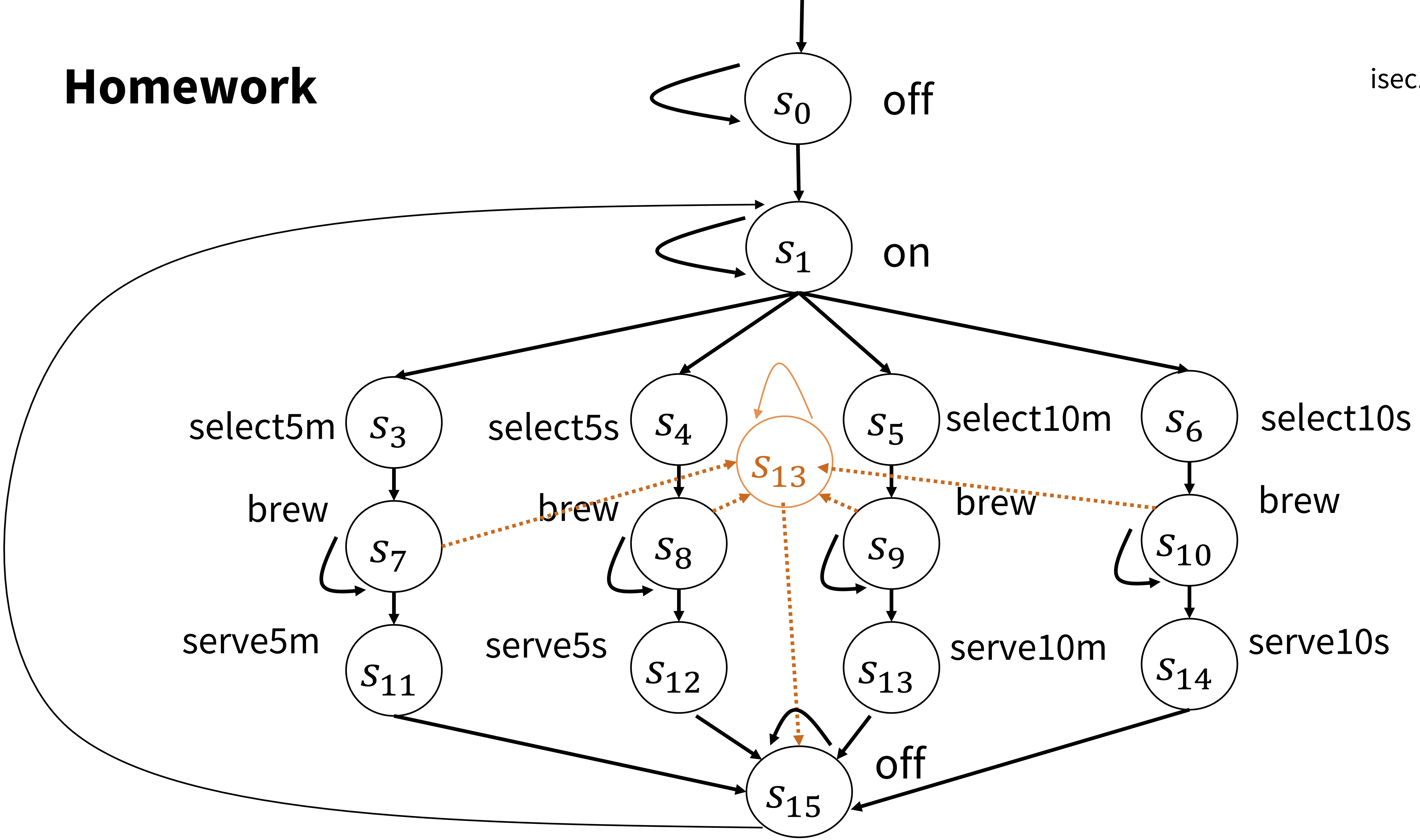
$$AG((select5m \rightarrow Xsever5m) \wedge (select5s \rightarrow serve5s) \ldots)$$

Homework



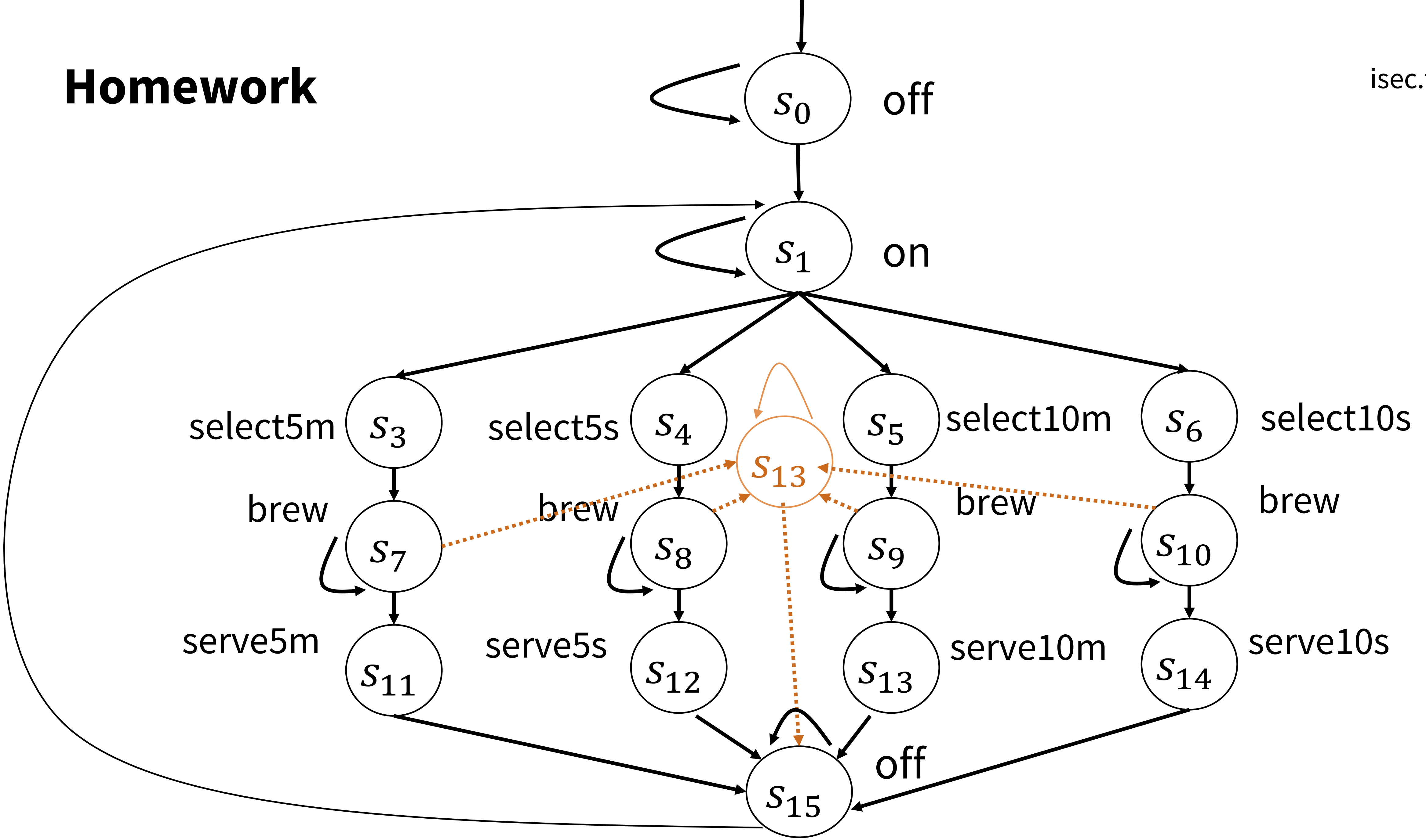
Does $M \vDash \boldsymbol{AGEF}\ (\boldsymbol{err})$? ✓ (The error state is always eventually reachable)

Homework



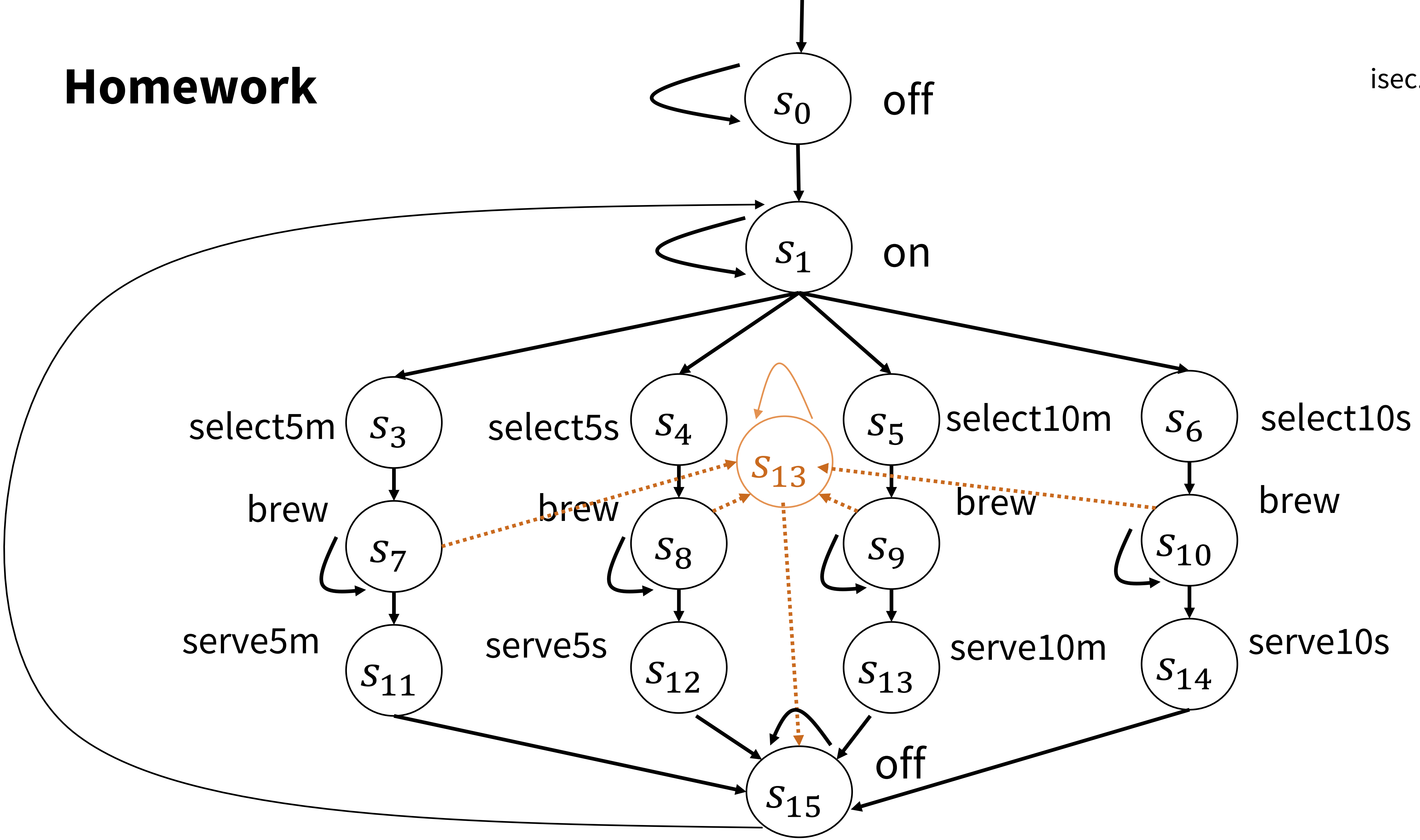
Does $M \models AGF\ (serve10m \lor serve10s)$? ✗

Homework



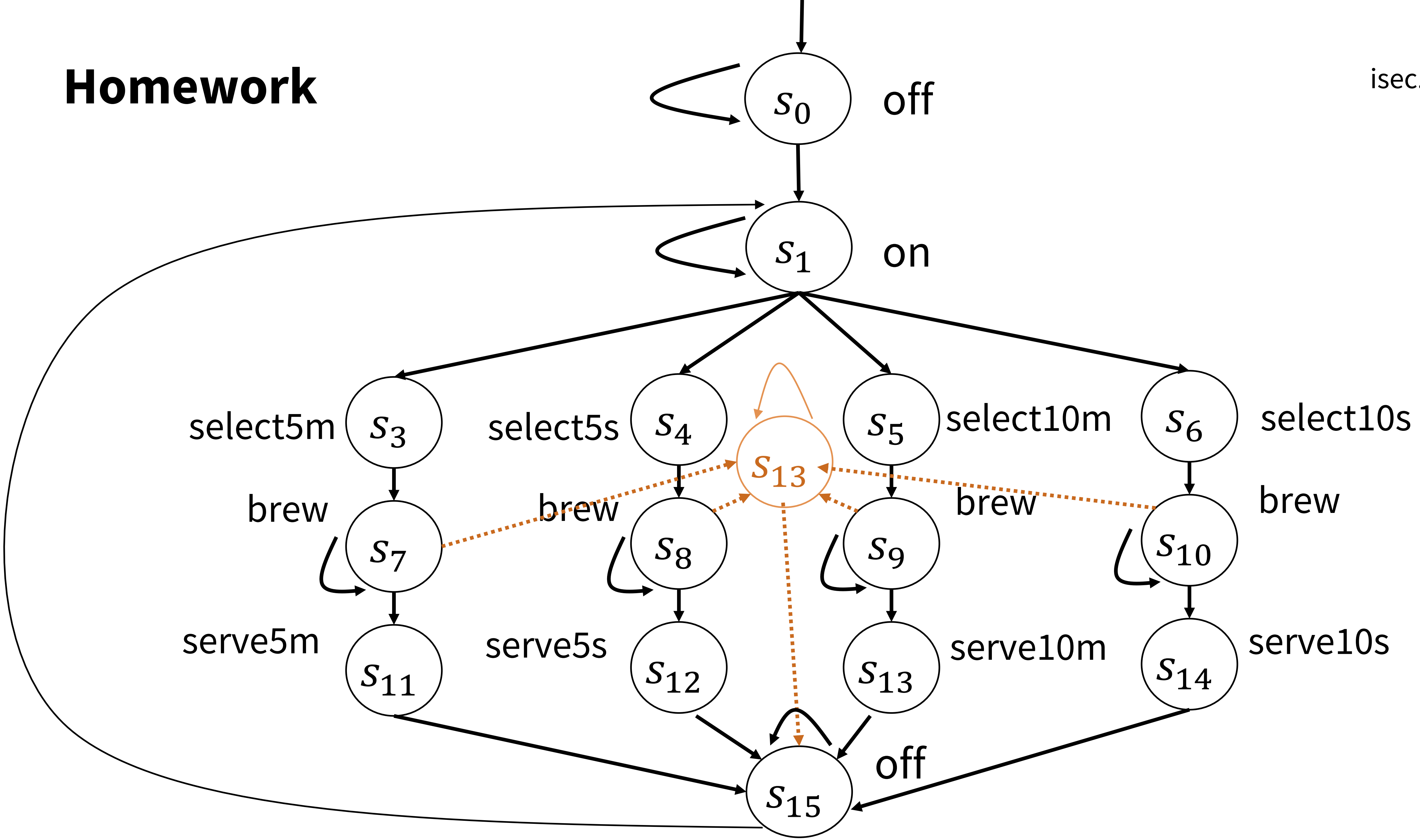
Does $M \vDash \boldsymbol{AG}\,(\boldsymbol{select10}) \wedge \boldsymbol{AG}(\boldsymbol{select10} \rightarrow \boldsymbol{AF}(\boldsymbol{serve\ 10} \vee \boldsymbol{error}))$? ✗

Homework



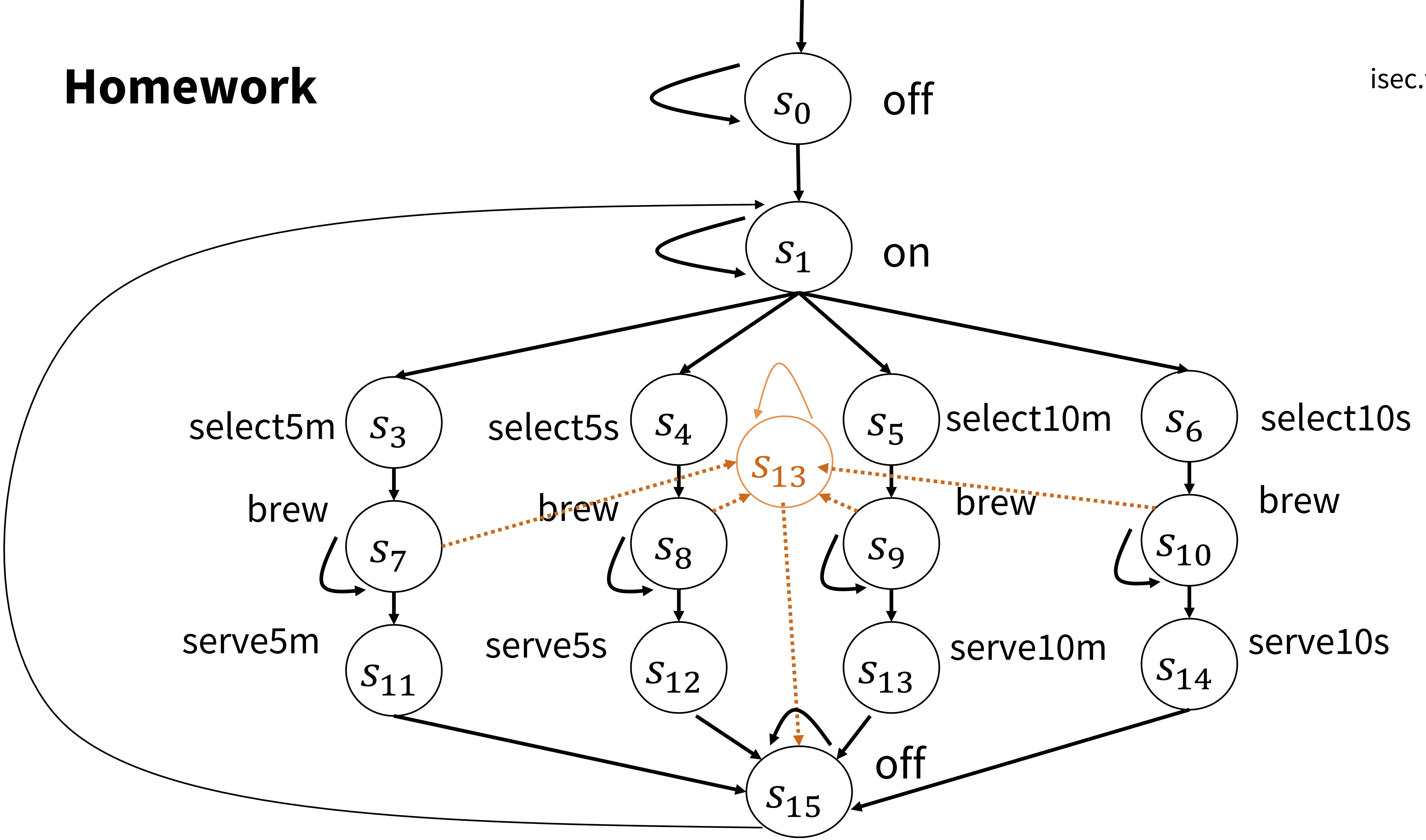
Does $M \vDash \boldsymbol{EG}(\neg \boldsymbol{err})$? ✓

Homework



Does $M \vDash \neg \boldsymbol{EF}\ (\boldsymbol{serve10} \rightarrow \boldsymbol{X serve5})$?

Homework



Does $M \vDash \boldsymbol{AG}((\boldsymbol{select5m} \rightarrow \boldsymbol{Xsever5m}) \wedge (\boldsymbol{select5s} \rightarrow \boldsymbol{serve5s}) \ldots)$?

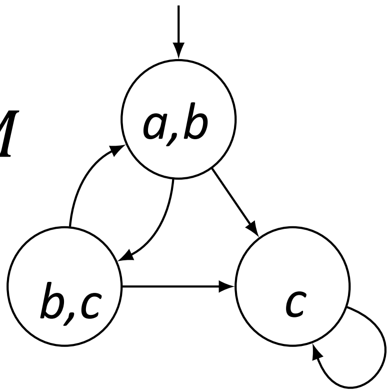
Plan for Today

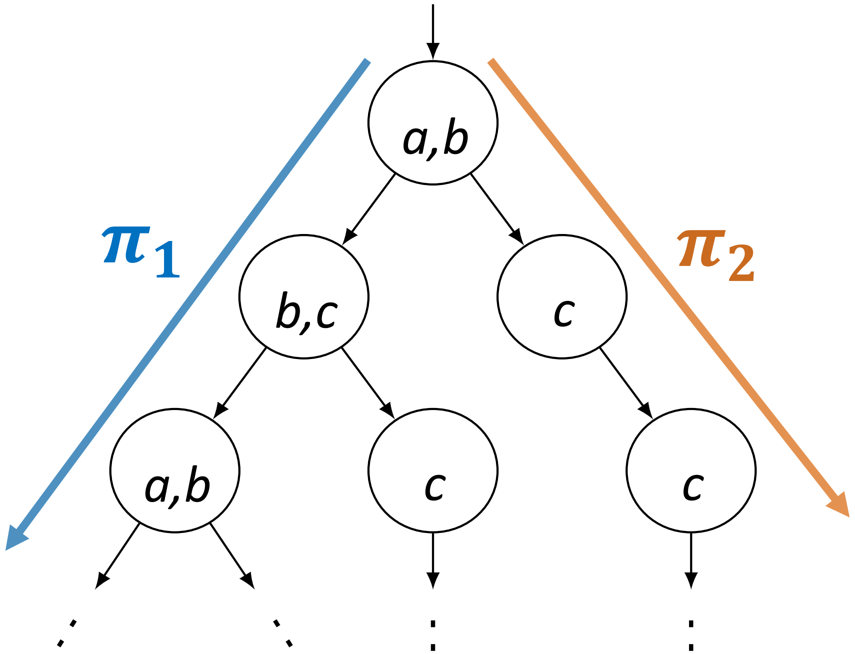
- Presentation of Homework
- Properties of CTL and LTL
  - LTL vs CTL
  - Counterexamples
  - Safety and Liveness Properties
- CTL Model Checking

Recap - CTL* - Path Quantifiers

- *Infinite* **path** $\pi = s_0, s_1, \ldots$
- Path quantifiers: **Aφ, Eφ**
  - They specify that **all paths** or **some paths** starting from a state $\boldsymbol{s}$ have property $\boldsymbol{\varphi}$.

Kripke structure M

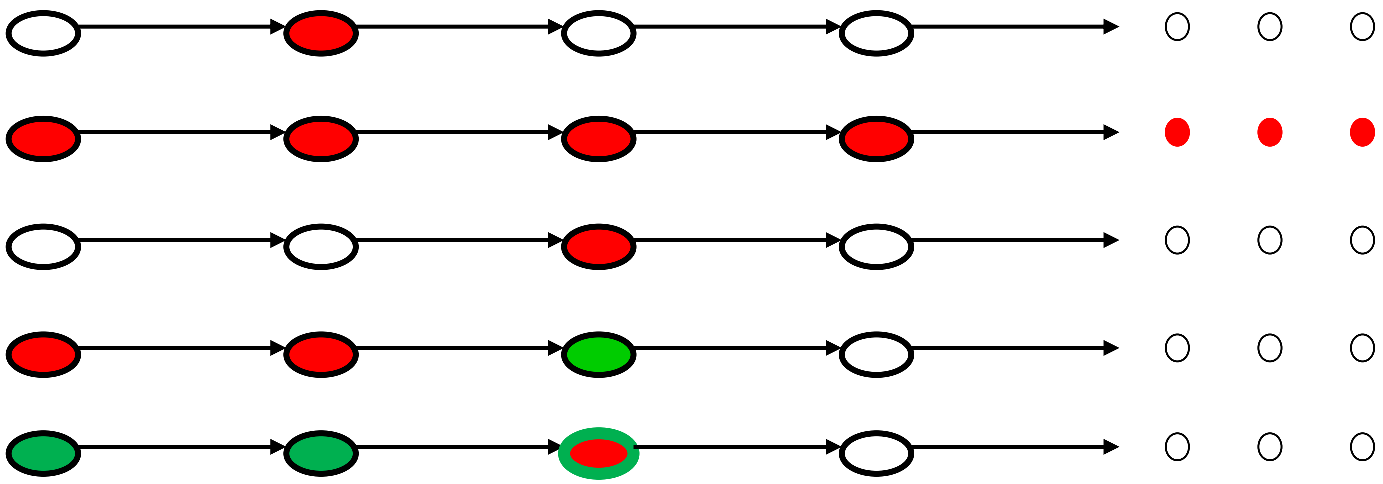




Recap - CTL* - Temporal Operators

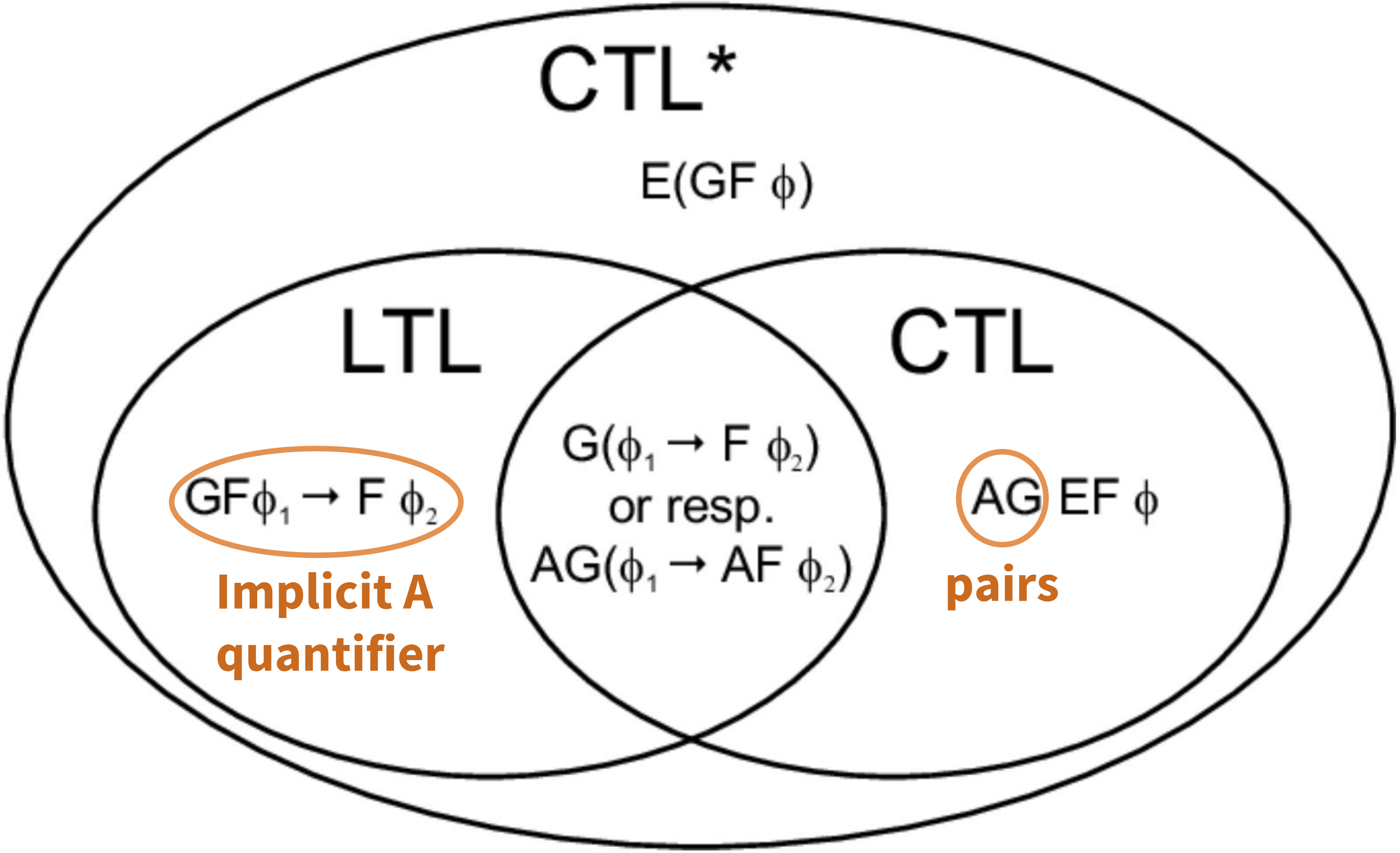
- Temporal operators
  - Describe **properties** along a given **path/execution**
- AP: a set of atomic propositions, $\boldsymbol{p}, \boldsymbol{q} \in \boldsymbol{AP}$

- **Next:** $\boldsymbol{Xp}$
- **Globally:** $\boldsymbol{Gp}$
- **Eventually:** $\boldsymbol{Fp}$
- **Until:** $\boldsymbol{pUq}$
- **Release:** $\boldsymbol{pRq}$



$\boldsymbol{pRq}$ … "$\boldsymbol{p}$ releases $\boldsymbol{q}$": $\boldsymbol{q}$ has to hold until $\boldsymbol{p}$ holds. However, $\boldsymbol{p}$ is not required to hold eventually.

Recap - LTL/CTL/CTL*



Recap - LTL - Syntax

- **State formulas**
  - $\boldsymbol{A}\boldsymbol{g}$ where $\boldsymbol{g}$ is a path formula
- **Path formulas**
  - $\boldsymbol{p} \in \boldsymbol{AP}$
  - $\neg g_1, \quad g_1 \vee g_2, \quad g_1 \wedge g_2, \quad \boldsymbol{X} g_1, \quad \boldsymbol{G} g_1, \quad g_1 \boldsymbol{U} g_2, \quad g_1 \boldsymbol{R} g_2$
    where g_1 and g_2 are path formulas

Recap - CTL - Syntax

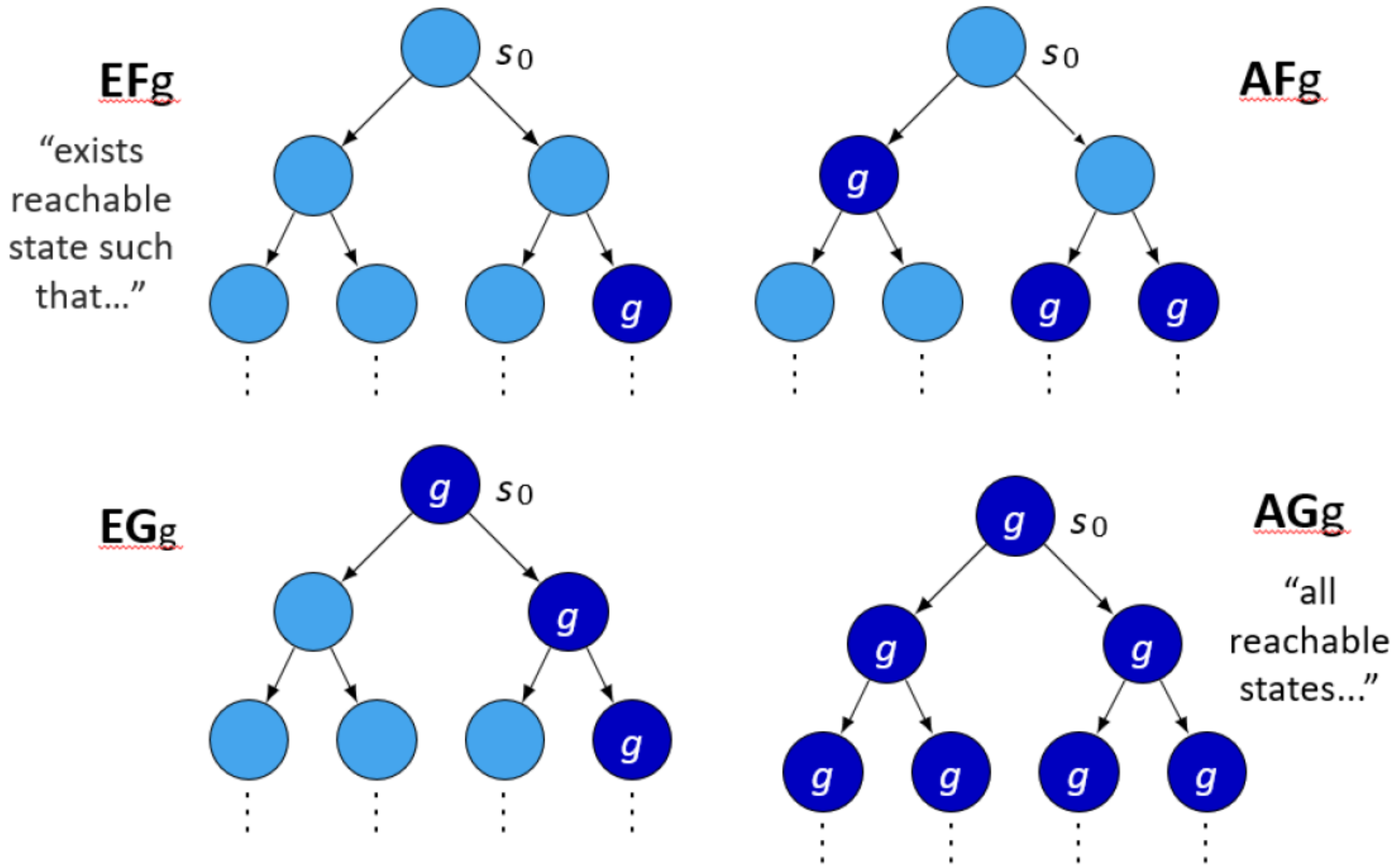
- **State formulas**
  - $\boldsymbol{p} \in \boldsymbol{AP}$
  - $\neg f_1,\quad f_1 \vee f_2,\quad f_1 \wedge f_2$
  - $\boldsymbol{AX} f_1,\quad \boldsymbol{AG} f_1,\quad \boldsymbol{A}(f_1 \boldsymbol{U} f_2),\quad \boldsymbol{A}(f_1 \boldsymbol{R} f_2)$
  - $\mathbf{E}\boldsymbol{X} f_1,\quad \boldsymbol{EG} f_1,\quad \boldsymbol{E}(f_1 \boldsymbol{U} f_2),\quad \boldsymbol{E}(f_1 \boldsymbol{R} f_2)$

where f_1 and f_2 are path formulas

Recap - CTL - Syntax

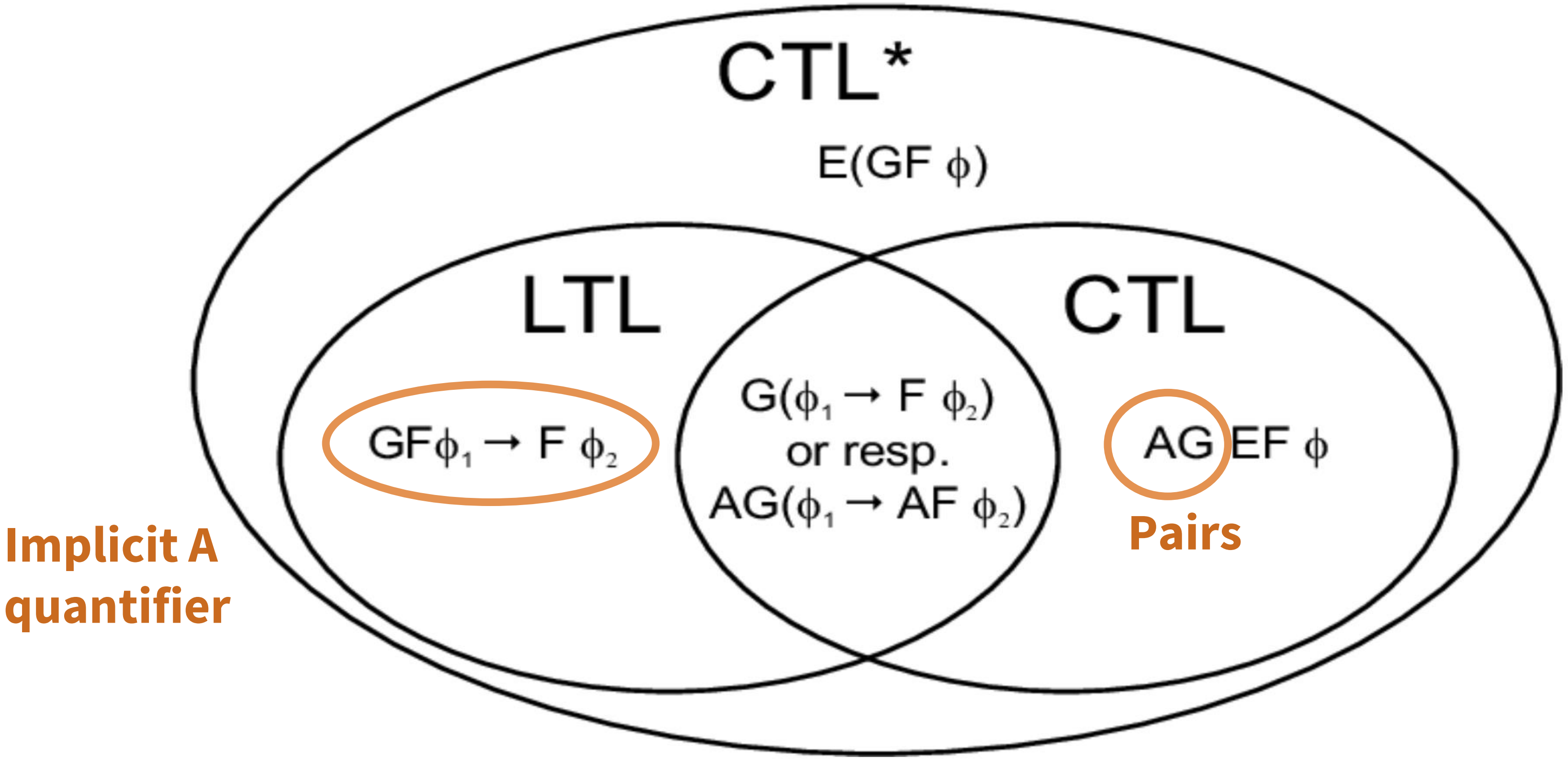
- **State formulas**
  - $\boldsymbol{p} \in \boldsymbol{AP}$
  - $\neg f_1,\ \ f_1 \vee f_2,\ \ f_1 \wedge f_2$
  - $\boldsymbol{AX} f_1,\ \ \boldsymbol{AG} f_1,\ \ \boldsymbol{A}(f_1 \boldsymbol{U} f_2),\ \ \boldsymbol{A}(f_1 \boldsymbol{R} f_2)$
  - $\mathbf{E}\boldsymbol{X} f_1,\ \ \boldsymbol{EG} f_1,\ \ \boldsymbol{E}(f_1 \boldsymbol{U} f_2),\ \ \boldsymbol{E}(f_1 \boldsymbol{R} f_2)$

where f_1 and f_2 are path formulas



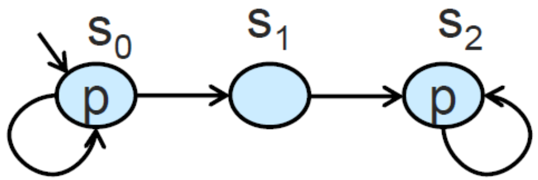
LTL/CTL/CTL*

- The expressive powers of LTL and CTL are incomparable
  - There are LTL formulas that have no equivalent CTL formula
  - There are CTL Formulas that have no equivalent LTL formula



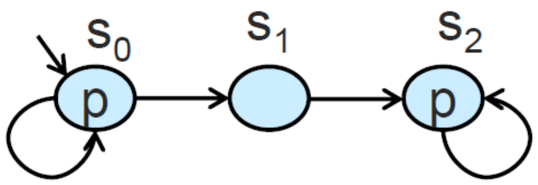
LTL vs CTL 1/2

- **Exercise**: Does the LTL formula $\boldsymbol{AFG\,p}$ have an equivalent in CTL?
  - $\boldsymbol{AFG\,p}$ = "for all paths, eventually p always holds"
- Hint:
  - Consider M
  - ***Does*** M $\vDash$ $\boldsymbol{AFGp}$?
  - ***Does*** M $\vDash$ $\boldsymbol{AFAGp}$?



LTL vs CTL 1/2

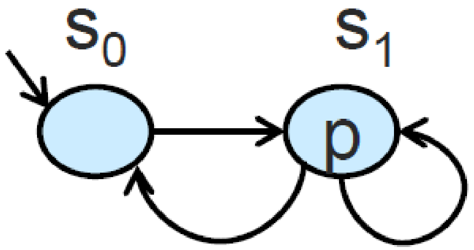
- **Exercise**: Does the LTL formula $\boldsymbol{AFG\ p}$ have an equivalent in CTL?
  - $\boldsymbol{AFG\ p}$ = “for all paths, eventually p always holds”
  - $\boldsymbol{AFAGp}$ = “for all paths, there is a point from which all reachable states satisfy p"



- M $\models$ $\boldsymbol{AFGp}$
  - All paths satisfy $\boldsymbol{FGp}$
  - $s_0, s_0, s_0, \ldots$
  - $s_0, s_0, \ldots s_0, s_1, s_2, s_2, s_2, \ldots .$
- M $\not\models$ $\boldsymbol{AFAGp}$
  - $s_0, s_0, s_0, \ldots$ does not satisfy $\boldsymbol{FAGp}$

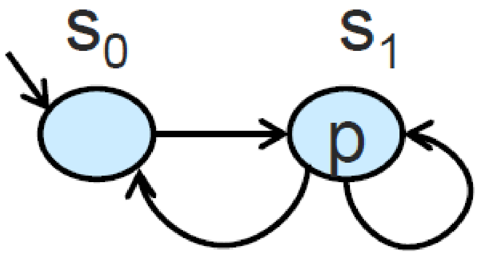
LTL vs CTL 1/2

- **Exercise**: Does the LTL formula $AFG\ p$ have an equivalent in CTL?
  - $AFG\ p$ = “for all paths, eventually p always holds”
- Hint:
  - Consider M
  - ***Does*** M ⊨ $AFGp$?
  - ***Does*** M ⊨ $AFEGp$?



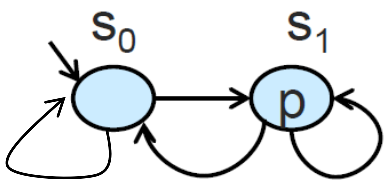
LTL vs CTL 1/2

- **Exercise**: Does the LTL formula $\boldsymbol{AFG\ p}$ have an equivalent in CTL?
  - $\boldsymbol{AFG\ p}$ = “for all paths, eventually p always holds”
  - $\boldsymbol{AFEGp}$ = “for all paths, there is a point from which there is a path where p globally holds "
- M $\nvDash$ $\boldsymbol{AFGp}$
  - $s_0, s_1, s_0, s_1, s_0, s_1$ ... does not satisfy $\boldsymbol{FGp}$
- M $\vDash$ $\boldsymbol{AFEGp}$
  - All paths satisfy $\boldsymbol{FEGp}$



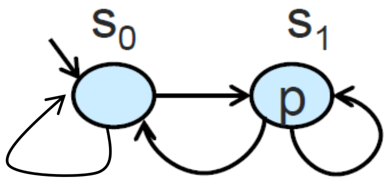
LTL vs CTL 2/2

- **Exercise**: Does $AG(EF\ p)$ have an equivalent in LTL?
  - $AG(EF\ p)$ = "from all reachable states, it is possible to reach a state that satisfies p"
- Hint:
  - Consider M
  - ***Does*** M $\vDash AG(EF\ p)$?
  - ***Does*** M $\vDash AGFp$?



LTL vs CTL 2/2

- **Exercise**: Does $AG(EF\ p)$ have an equivalent in LTL?
  - $AG(EF\ p)$ = "from all reachable states, it is possible to reach a state that satisfies p"
  - $AGF\ p$ = "In all paths, p holds infinitely often"



- M $\models AG(EF\ p)$
  - All reachable states satisfy EFp
- M $\not\models AGFp$
  - s_0, s_0, s_0 ... does not satisfy GFp

Plan for Today

- Presentation of Homework
- Properties of CTL and LTL
  - LTL vs CTL
  - Counterexamples
  - Safety and Liveness Properties
- CTL Model Checking

Counterexamples

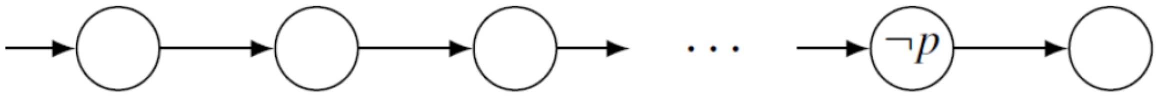
- **Given M and φ s.t. $M \not\models \varphi$.**
  A counterexample is trace π of M violating φ

- Counterexamples are a central feature of MC
- Used for debugging
  - Should be easy-to-understand by human
  - Should have finite representation

Counterexamples

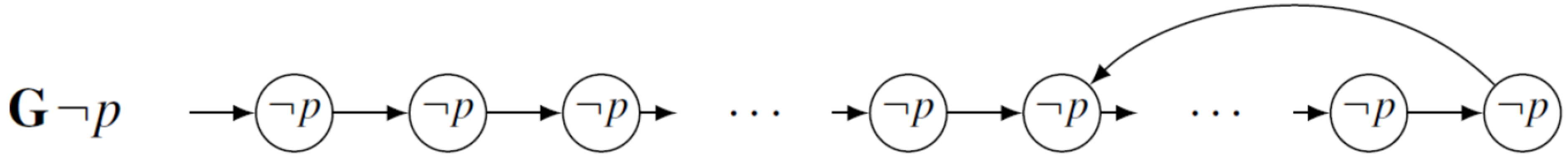
- $AX\ p$
  - A counterexample for $AX\ p$ is a **transition** from an initial state to a state **violating** p.
  - A counterexample for $AX\ p$ is a **witness** for $EX\ \neg p$
- $AG\ p$
  - A counterexample for $AG\ p$ is a **finite path** from an initial state to a state **violating** p.
  - A counterexample for $AG\ p$ is a **witness** for $EF\ \neg p$

$\mathbf{F}\,\neg p$



Counterexamples

- ***AF p***
  - A counterexample for $\boldsymbol{AF\ p}$ is an **infinite path** with all of its states **violating** $\boldsymbol{p}$.
  - A counterexample for $\boldsymbol{AF\ p}$ is a **witness** for $\boldsymbol{EG\ \neg p}$
  - **Finite representation** of counterexamples for $\boldsymbol{AF\ p}$:
    - **Lasso:** $\boldsymbol{\pi = \pi_0(\pi_1)^\omega}$
    - π_0 and π_1 are **finite** paths
    - ω indicates **infinitely many repetitions** of π_1



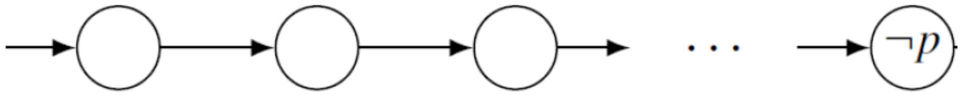
Plan for Today

- Presentation of Homework
- Properties of CTL and LTL
  - LTL vs CTL
  - Counterexamples
  - Safety and Liveness Properties
- CTL Model Checking

Safety and Liveness

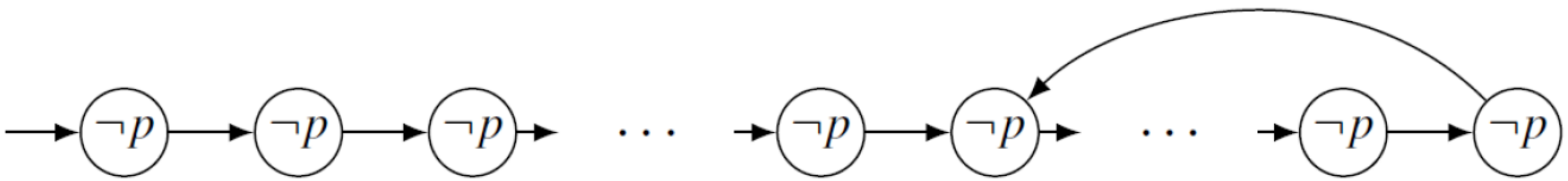
- Safety properties state that “something **bad** will **never** happen”
  - E.g.: $AG\neg p$
  - A counterexample is a **finite (loop-free) path**

$bad = \neg p$



- Liveness properties state that “something **good** will happen **eventually**”
  - E.g.: $AF\ p, A(pUq)$
  - A counterexample is an **infinite path** showing that the good property **NEVER** holds.

$good = p$



Model Checking Problem

- Given a Kripke structure M and a CTL formula f
- Model Checking Problem
  - Does $M \models f$?

- Algorithm:
  - Compute all states satisfying f:
    $$[\![f]\!]_M = \{s \in S \mid M, s \models f\}$$
  - If $S_0 \subseteq [\![f]\!]_M$ then it holds that $M \models f$

Illustrative Example – Mutual Exclusion

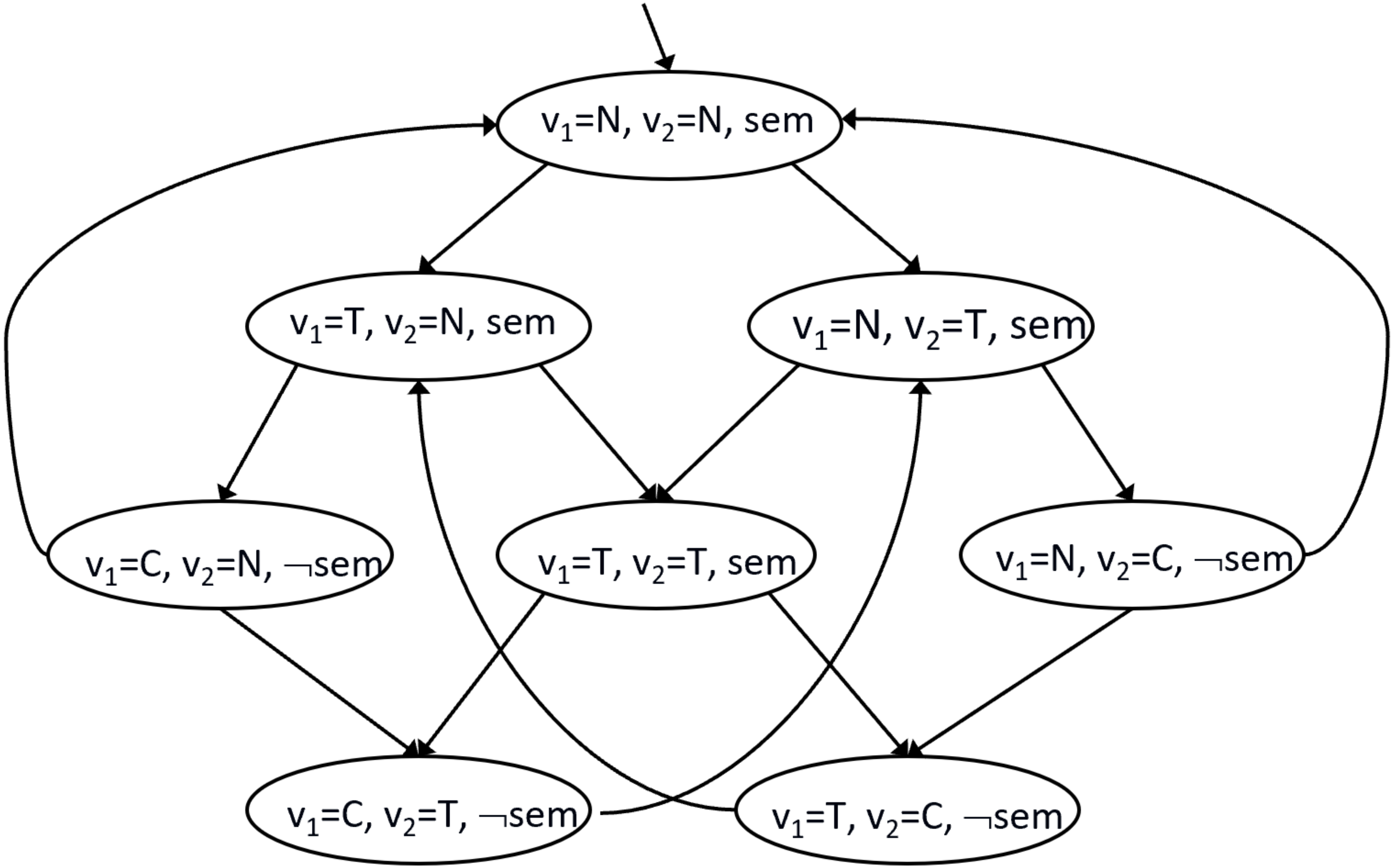
- Given a Kripke structure M and a CTL formula f
- Two processes P_1 and P_2 with a joint semaphor signal sem
- Each process P_i has a variable v_i describing its state:
  - $v_i = \mathrm{N}$ Non-critical
  - $v_i = \mathrm{T}$ Trying
  - $v_i = \mathrm{C}$ Critical

- Each process runs the following program

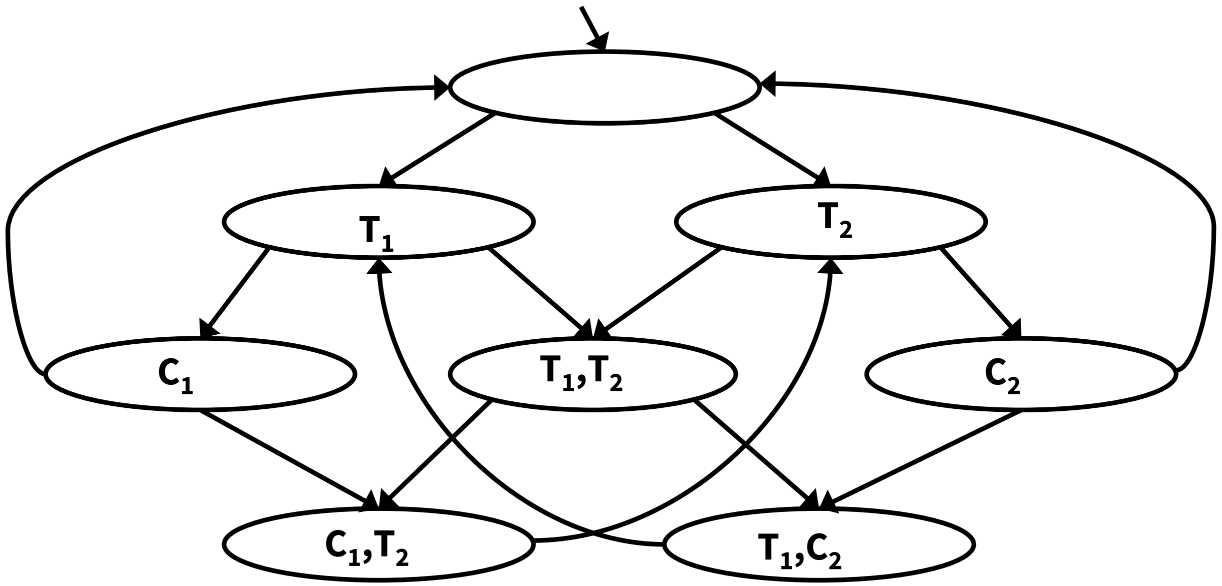
```
while (true) {
    if (v_i == N)  v_i = T;
    else if (v_i == T && sem)  { v_i = C; sem = 0; }
    else if (v_i == C)  {v_i = N; sem = 1; }
}
```

Atomic action (→ each of the three branches)

Mutual Exclusion

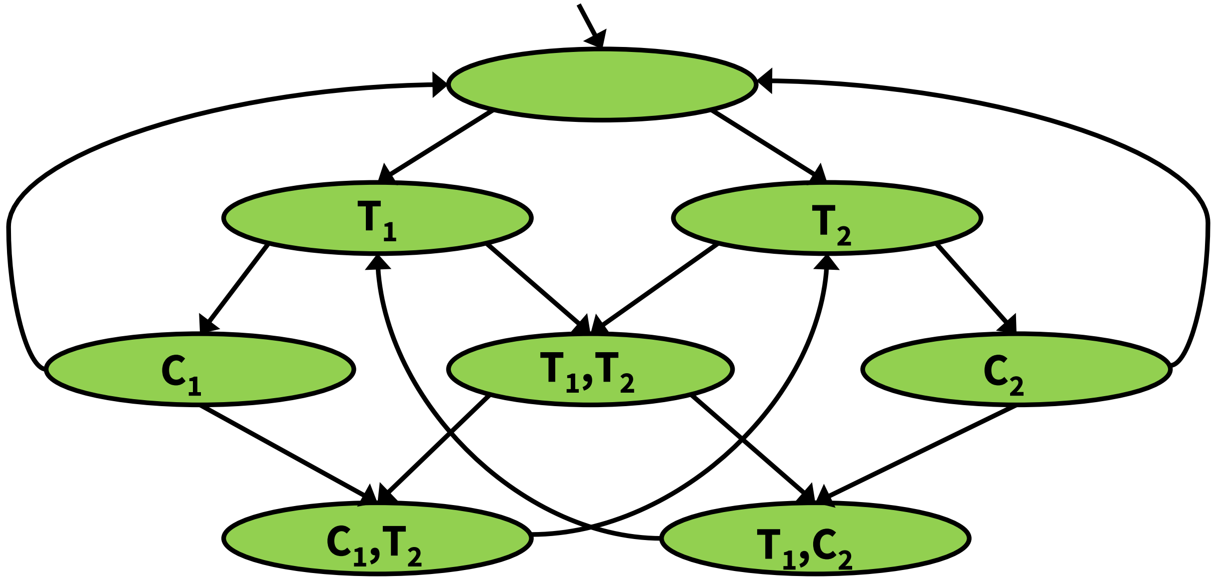


Mutual Exclusion



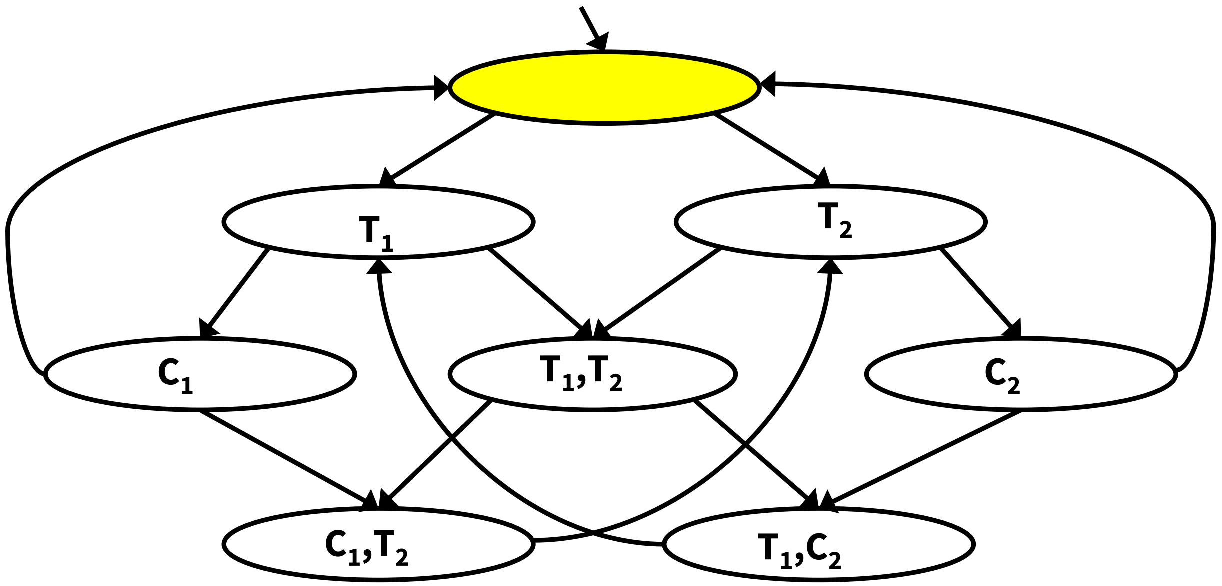
- We define atomic propositions: $AP = \{C_1, C_2, T_1, T_2)$
- A state is labeled with T_i if $v_i = T$
- A state is labeled with C_i if $v_i = C$

Mutual Exclusion – 1/4



- Does it hold that $M \models \varphi$?
  - $\varphi = \boldsymbol{AG}\neg(\boldsymbol{C_1} \wedge \boldsymbol{C_2})$ ✓

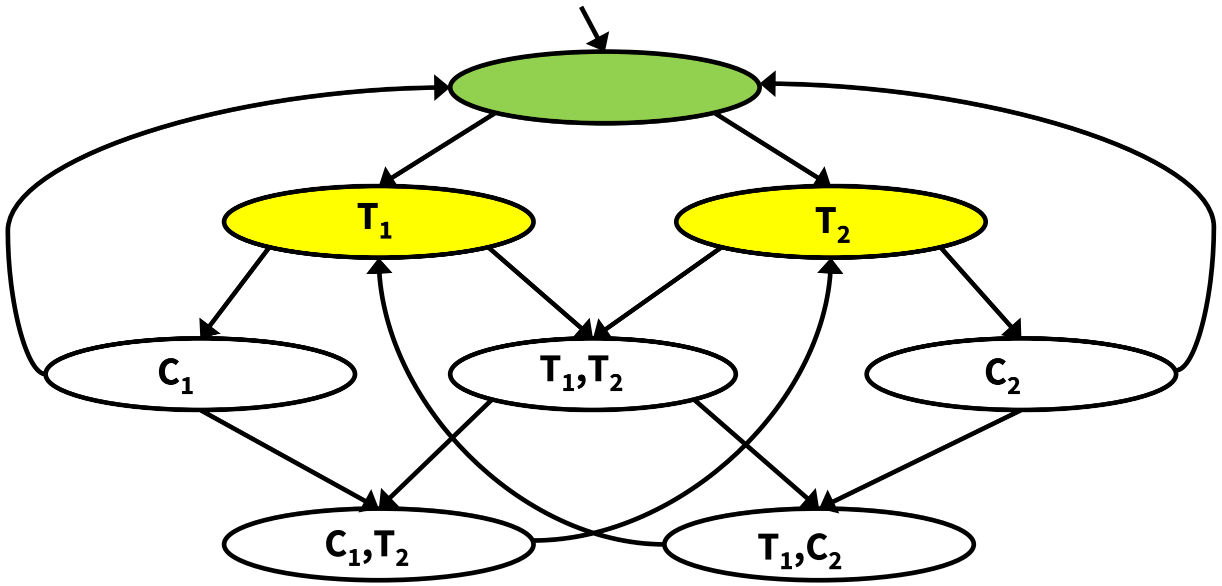
Mutual Exclusion – 1/4



S_0

- Does it hold that $\mathrm{M} \vDash \varphi$?
  - $\varphi = AG\neg(C_1 \land C_2)$
- S_i ...reachable states from an initial state after i steps

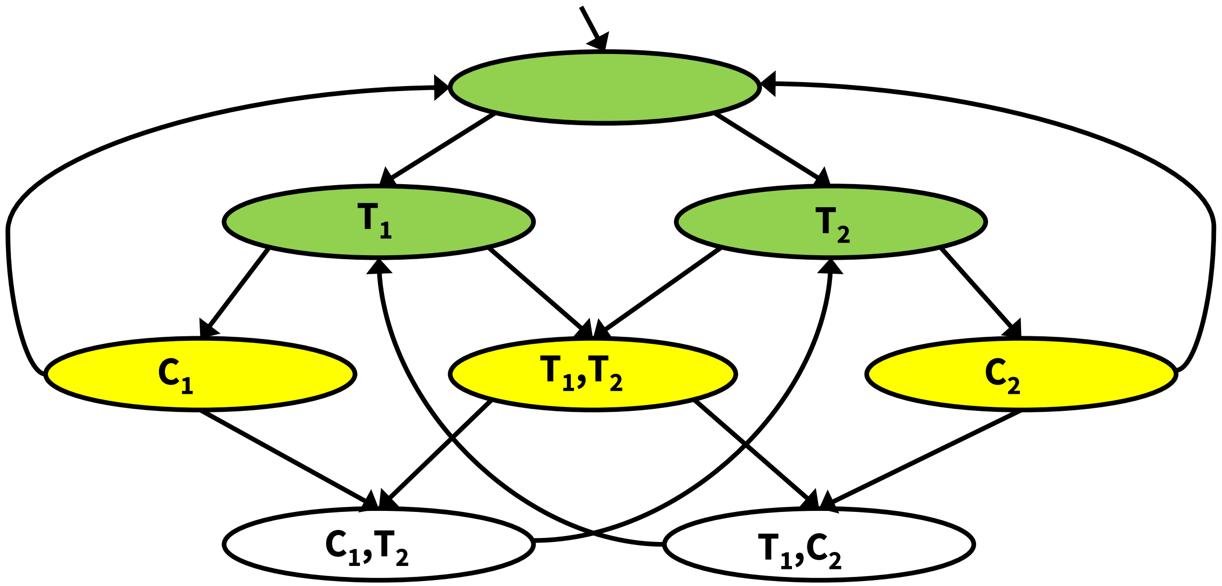
Mutual Exclusion – 1/4



S_1

- Does it hold that $M \models \varphi$?
  - $\varphi = AG\neg(C_1 \wedge C_2)$
- S_i ...reachable states from an initial state after i steps

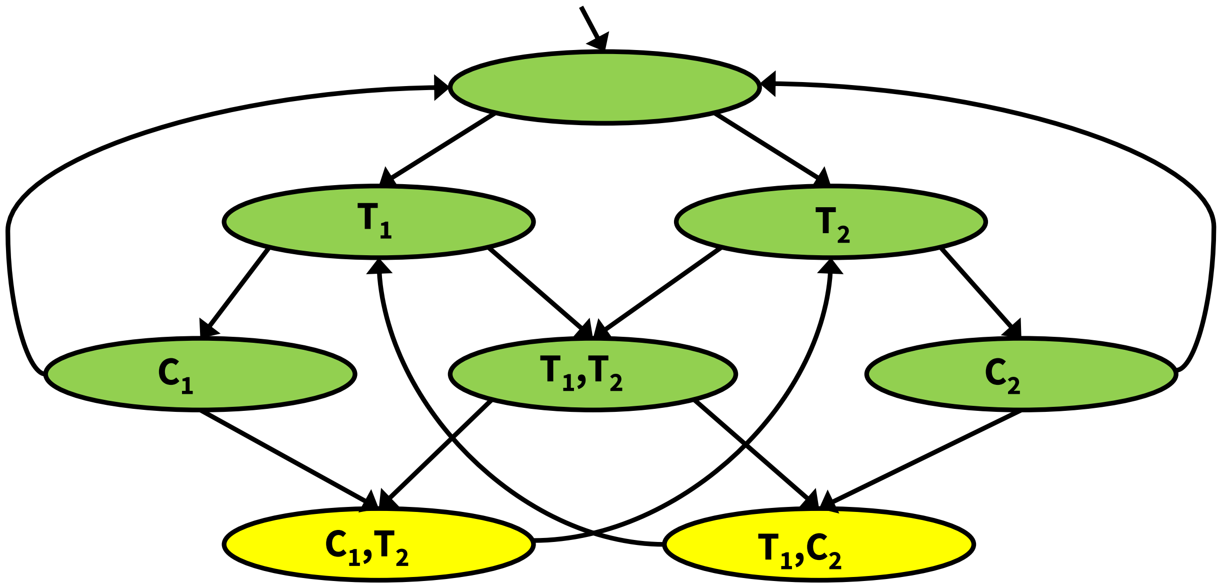
Mutual Exclusion – 1/4



S_2

- Does it hold that $M \vDash \varphi$?
  - $\varphi = AG\neg(C_1 \wedge C_2)$
- S_i ...reachable states from an initial state after i steps

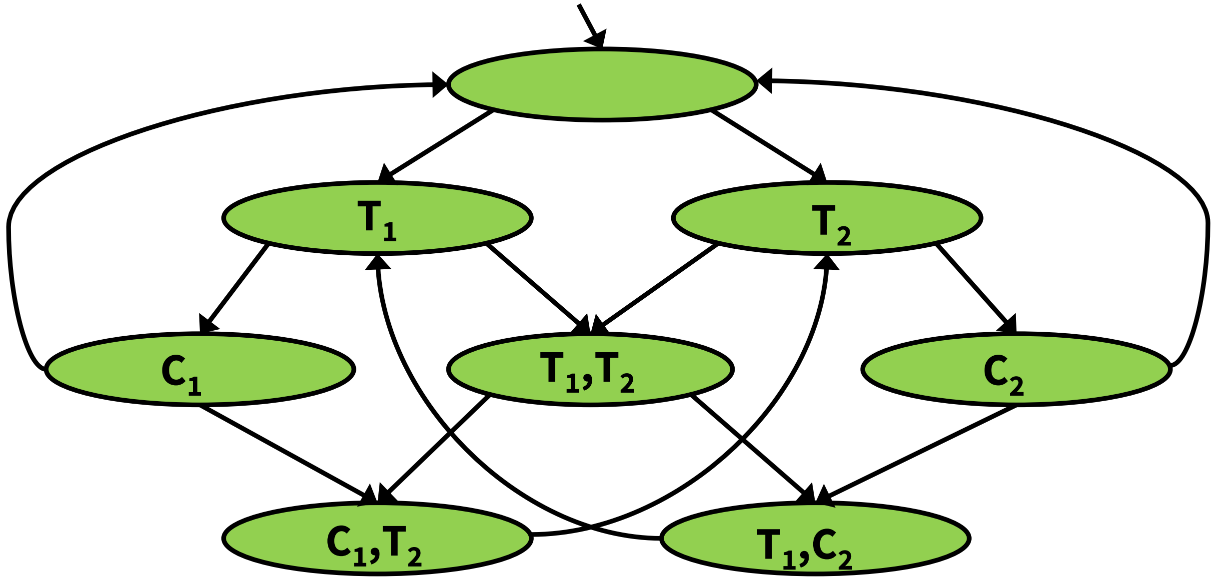
Mutual Exclusion – 1/4



S_3

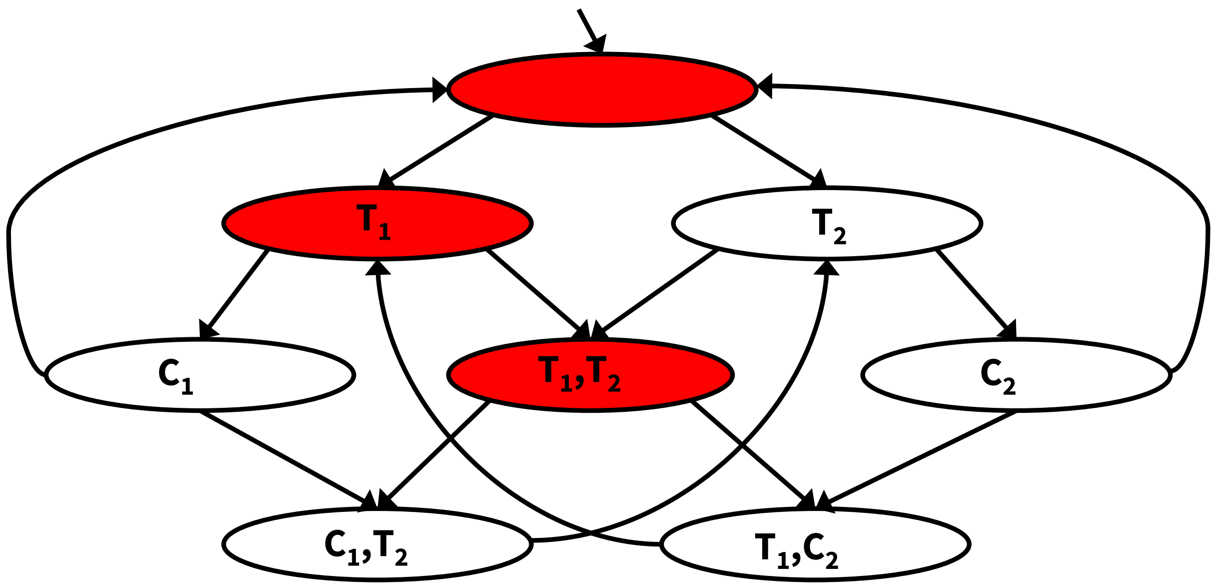
- Does it hold that $M \vDash \varphi$?
  - $\varphi = AG\neg(C_1 \wedge C_2)$
- S_i ...reachable states from an initial state after i steps

Mutual Exclusion – 1/4



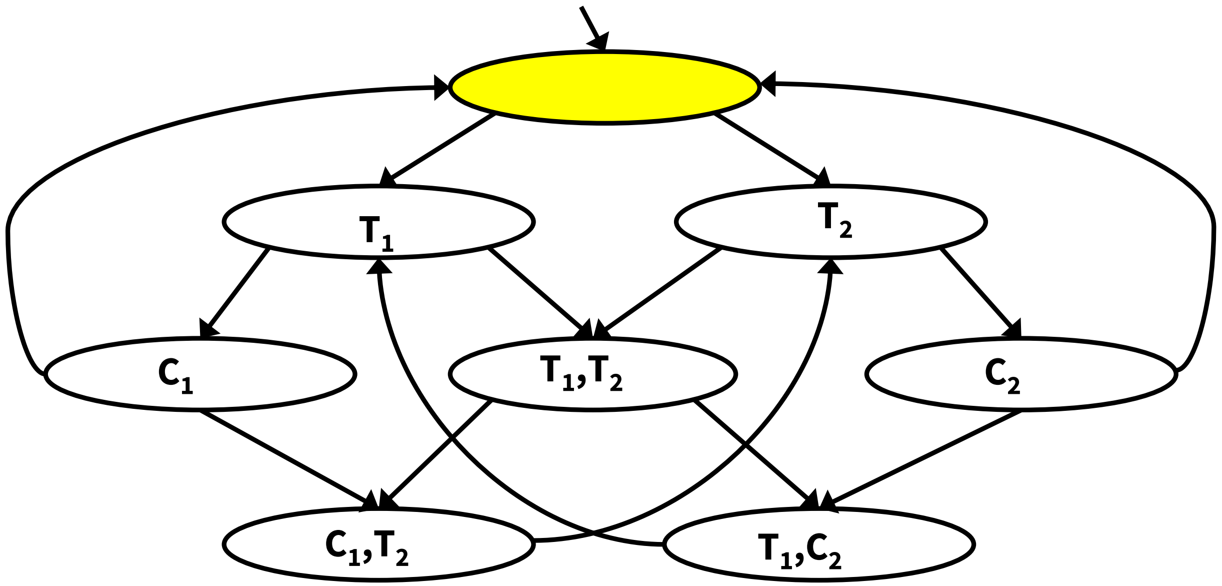
- Does it hold that $M \vDash \varphi$?
  - $\varphi = \boldsymbol{AG}\neg(\boldsymbol{C_1} \wedge \boldsymbol{C_2})$ ✓

Mutual Exclusion – 2/4



- Does it hold that $\mathrm{M} \vDash \varphi$?
  - $\varphi = \boldsymbol{AG}\neg(\boldsymbol{T_1} \wedge \boldsymbol{T_2})$ ✗

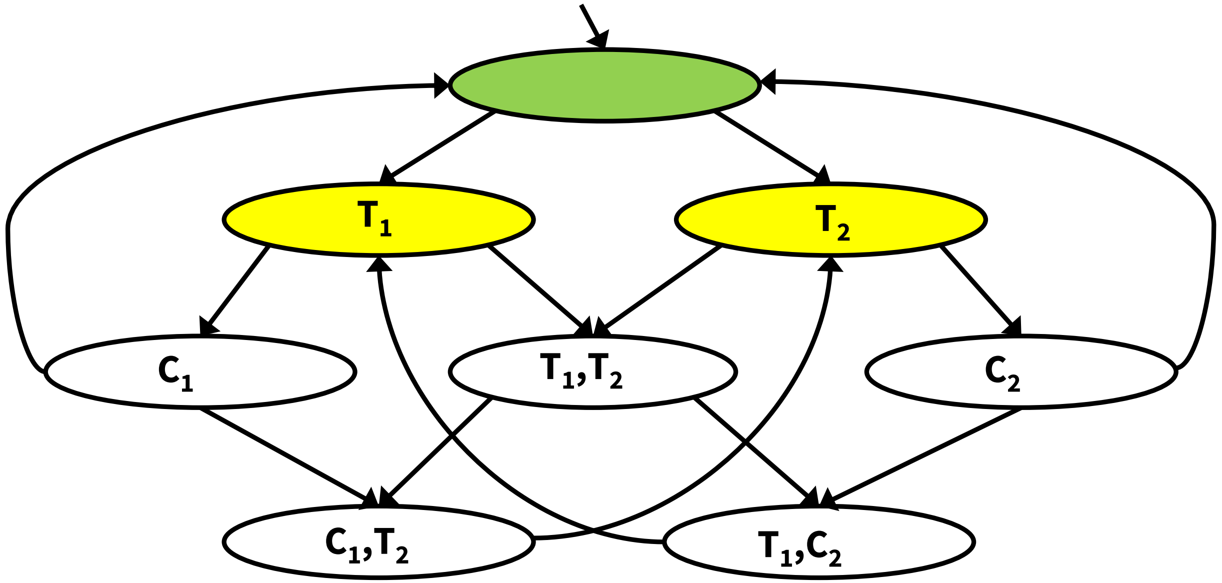
Mutual Exclusion – 2/4



S_1

- Does it hold that $M \models \varphi$?
  - $\varphi = AG\neg(T_1 \wedge T_2)$ ✗
- S_i ...reachable states from an initial state after i steps

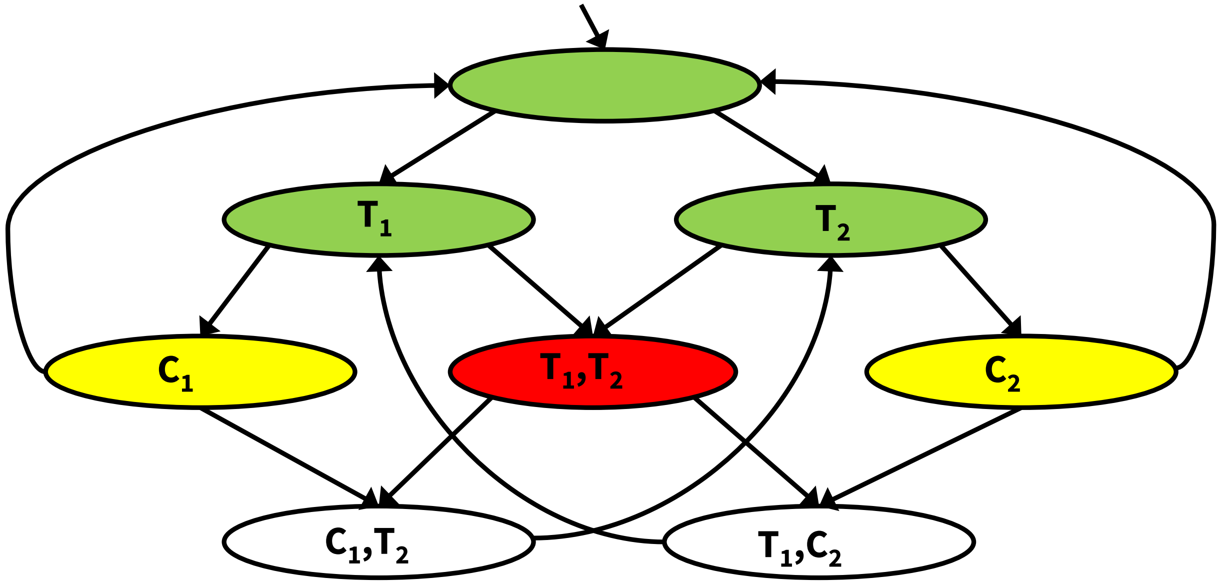
Mutual Exclusion – 2/4



S_2

- Does it hold that $M \vDash \varphi$?
  - $\varphi = AG\neg(T_1 \wedge T_2)$ ✗
- S_i ...reachable states from an initial state after i steps

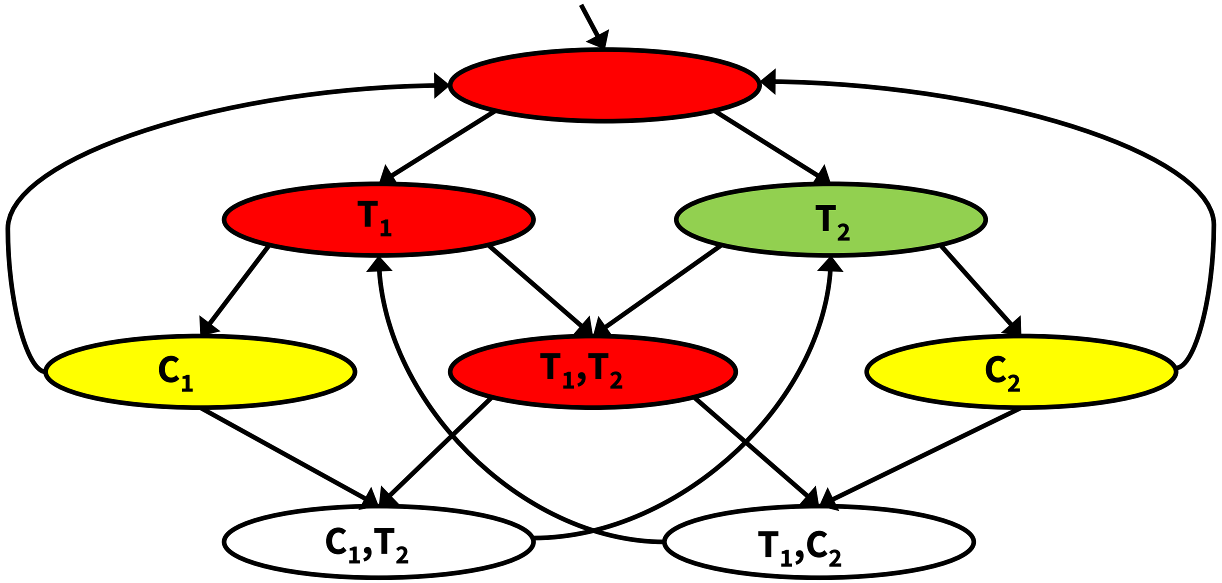
Mutual Exclusion – 2/4



S_3

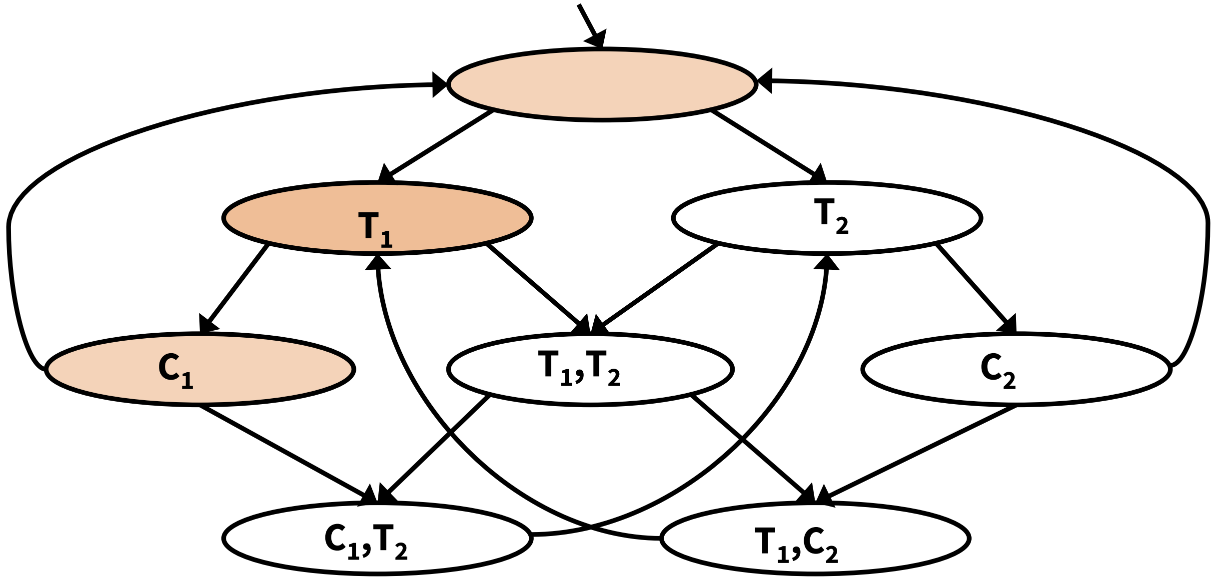
- Does it hold that $\mathrm{M} \vDash \varphi$?
  - $\varphi = AG\neg(T_1 \wedge T_2)$ ✗
- S_i ...reachable states from an initial state after i steps

Mutual Exclusion – 2/4



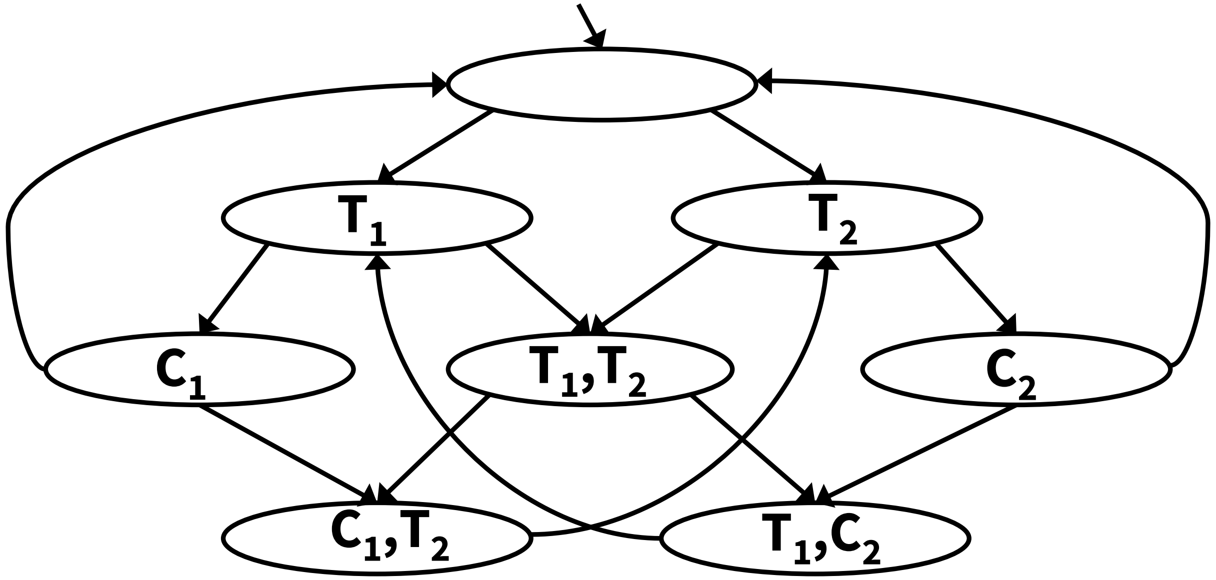
- Does it hold that $\mathrm{M} \vDash \varphi$?
  - $\varphi = \boldsymbol{AG}\neg(\boldsymbol{T}_1 \wedge \boldsymbol{T}_2)$ ✗
- Model checker returns a **counterexample**

Mutual Exclusion – 3/4



- Does it hold that $\mathrm{M} \vDash \varphi$?
  - $\varphi = \boldsymbol{EG}\neg(\boldsymbol{T}_1 \wedge \boldsymbol{T}_2)$ ✓

Mutual Exclusion – 4/4



- Does it hold that $M \vDash \varphi$?

  - $\varphi = \mathbf{AG}\ \mathbf{EF}\ (T_1)$ ✓
  - Form any state it is always possible to reach the state labeled with T_1.

CTL MC Algorithm

- Does $M \models f$?

- MC algorithm works iteratively on **sub-formulas** of f

- For checking **AG(** request $\rightarrow$ **AF** grant**)**
  - Check grant, request
  - Then check **AF** grant
  - Next check request $\rightarrow$ **AF** grant
  - Finally check **AG(** request $\rightarrow$ **AF** grant**)**

CTL MC Algorithm

- Does $M \vDash f$?

- MC algorithm works iteratively on **sub-formulas** of f

- For every sub-formula g of f:
  - Add g to label(s) for every state s that satisfies g
  - $g \in \text{label}(s) \Leftrightarrow M, s \vDash g$

- label(s) = set of sub-formulas of f that are true in s
- **$M \vDash f$ if and only if $f \in$ label(s) for all initial states $s \in S_0$ of M**

- MC algorithm needs to handle **AP** and **$\neg$, $\vee$, EX, EU, EG**

CTL MC Algorithm: Checking $\mathbf{AP}$, $\neg$,$\vee$ - Formulas

- label(s) = set of sub-formulas of f that are true in s

- Procedure for labeling the states:
  - For $p \in AP$: $p \in$ label(s) if and only if $p \in$ L(s)
  - For subformulas f_1 and f_2 that have already been checked
    - $\neg f_1$ add to label(s) if and only if $f_1 \notin label(s)$
    - $f_1 \vee f_2$ add to label(s) if and only if $f_1 \in labels(s)$ **or** $f_2 \in label(s)$

CTL MC Algorithm: Checking $g = EX\, f_1$

- Procedure for labeling the states satisfying $g = EX f_1$:
  - Add g to label(s) if and only if s has a successor t such that $f_1 \in$ label(t)

```
procedure CheckEX (f1)
    T := { t | f1 ∈ label(t) }
    while T ≠ ∅  do
      choose t ∈T;  T := T \ {t};
      for all s  such that  R(s,t) do
            if EX f1 ∉ label(s) then
                        label(s) : = label(s) ∪ { EX f1};
```

CTL MC Algorithm: Checking $g = E(f_1 U f_2)$

- **Exercise**: Procedure for labeling the states satisfying $g = E(f_1 U\ f_2)$
  - Hint: Rewrite the procedure CheckEX

```
procedure CheckEX (f₁)
  T := { t | f₁ ∈ label(t) }



while T ≠ ∅  do
    choose t ∈T;  T := T \ {t};
    for all s  such that  R(s,t) do
        if EX f₁ ∉ label(s) then
            label(s) : = label(s) ∪ { EX f₁};
```

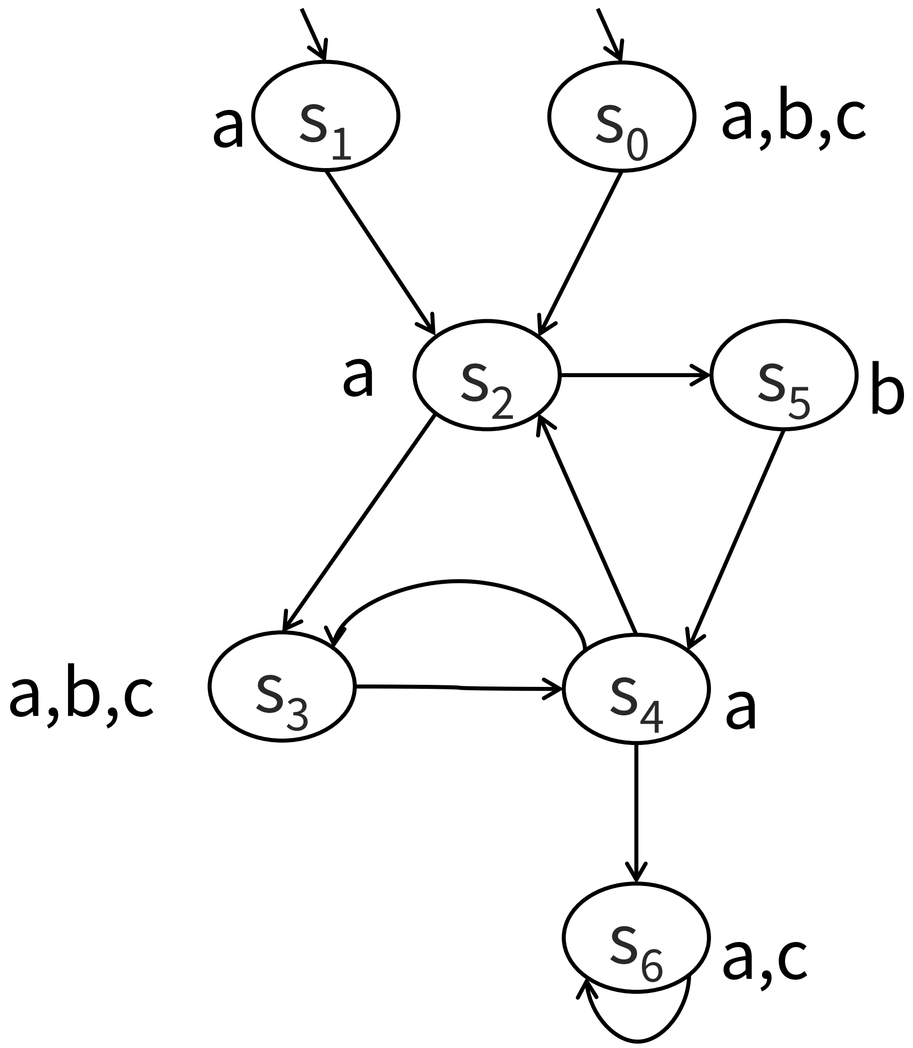
```
procedure CheckEU (f₁,f₂)
  T := { t | f₂ ∈ label(t) }

  for all t∈T do
     label(t) := label(t) ∪ { E(f₁ U f₂) }

  while T ≠ ∅  do
     choose t ∈T;  T := T \ {t};
     for all s  such that  R(s,t) do
         if E(f₁ U f₂) ∉ label(s) and f₁ ∈ label(s) then
            label(s) : = label(s) ∪ {E(f₁ U f₂) };
            T : = T ∪ {s}
```

CTL MC: Checking $g = E(f_1 U f_2)$

- Does it hold that $M \models E(aUb)$?



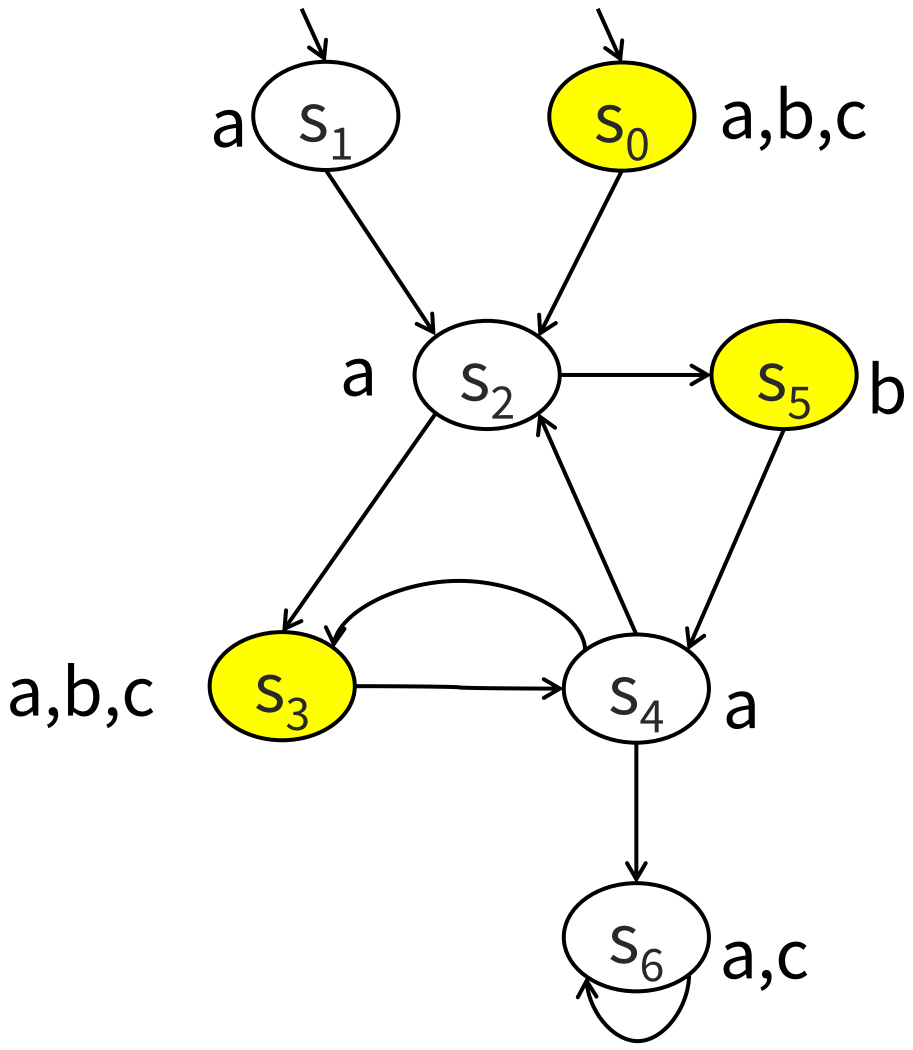

```
procedure CheckEU (f1,f2)
 T := { t | f2 ∈ label(t) }

 for all t∈T do
   label(t) := label(t) ∪ { E(f1 U f2) }

 while T ≠ ∅  do
   choose t ∈T;  T := T \ {t};
   for all s  such that  R(s,t) do
        if E(f1 U f2) ∉ label(s) and f1 ∈ label(s) then
           label(s) : = label(s) ∪ {E(f1 U f2) };
           T : = T ∪ {s}
```

CTL MC: Checking $g = E(f_1 U f_2)$

- Does it hold that M $\models E(aUb)$?



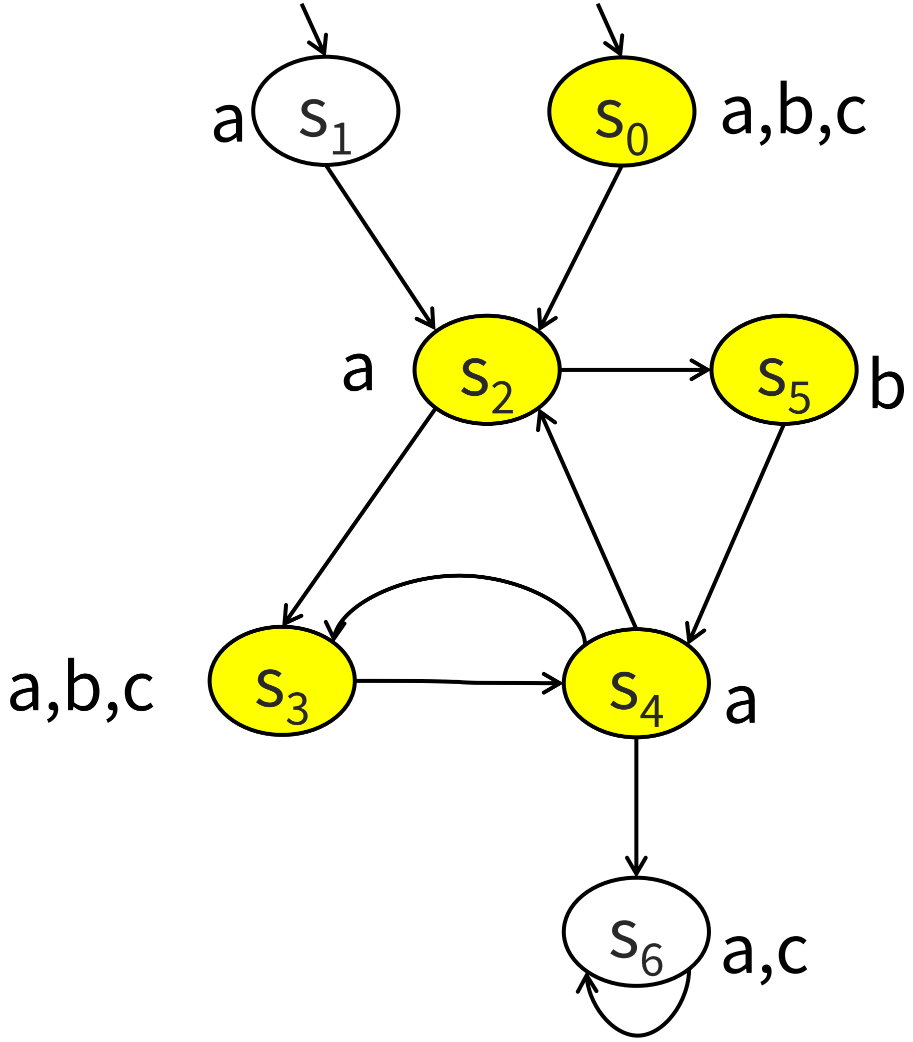

```
procedure CheckEU (f1,f2)
  T := { t | f2 ∈ label(t) }

  for all t∈T do
    label(t) := label(t) ∪ { E(f1 U f2) }

  while T ≠ ∅  do
    choose t ∈T;  T := T \ {t};
    for all s  such that  R(s,t) do
        if E(f1 U f2) ∉ label(s) and f1 ∈ label(s) then
          label(s) : = label(s) ∪ {E(f1 U f2) };
          T : = T ∪ {s}
```

CTL MC: Checking $g = E(f_1 U f_2)$

- Does it hold that M ⊨ $E(aUb)$?



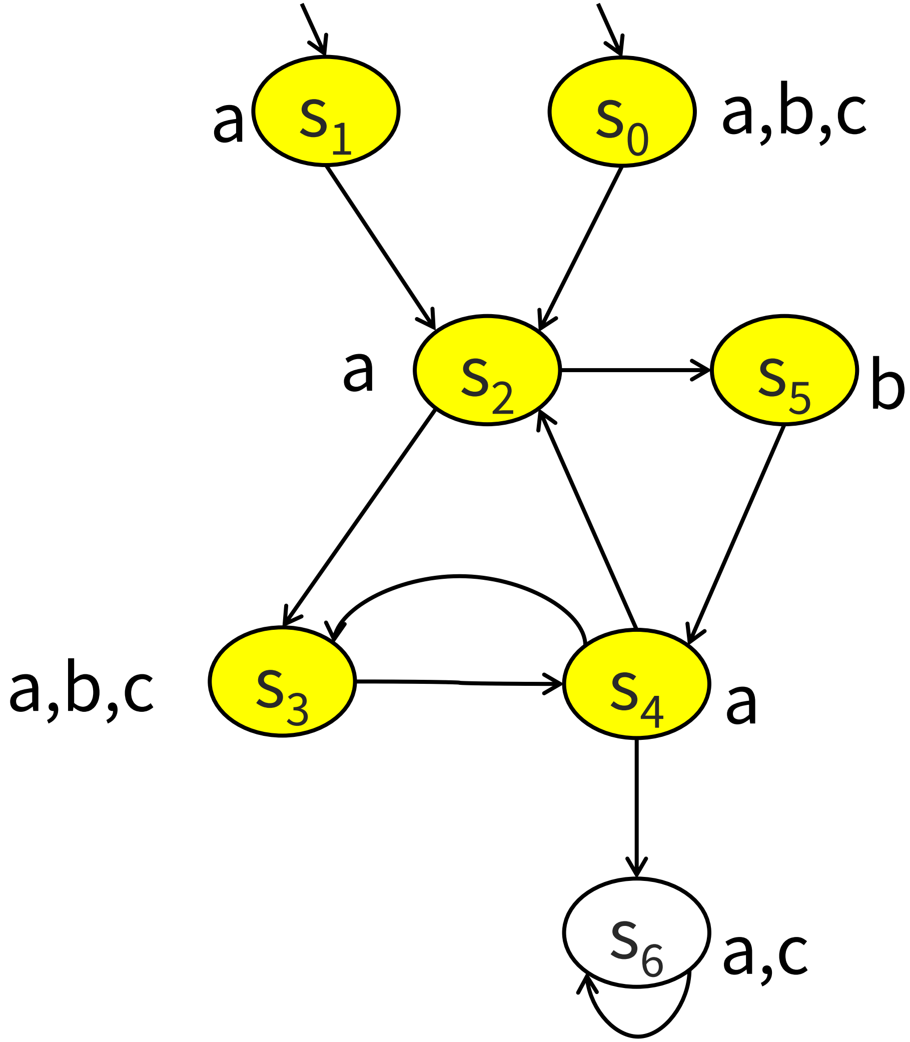
procedure CheckEU (f_1,f_2)
 T := { t | f_2 ∈ label(t) }

 for all t∈T do
  label(t) := label(t) ∪ { E(f_1 U f_2) }

 while T ≠ ∅ do
  choose t ∈T; T := T \ {t};
  for all s such that R(s,t) do
   if E(f_1 U f_2) ∉ label(s) and f_1 ∈ label(s) then
    label(s) : = label(s) ∪ {E(f_1 U f_2) };
    T : = T ∪ {s}

CTL MC: Checking $g = E(f_1 U f_2)$

- Does it hold that M $\models E(aUb)$? ✓



[[E(aUb)]] = {0,1,2,3,4,5}

procedure CheckEU (f_1,f_2)
  T := { t | $f_2 \in$ label(t) }

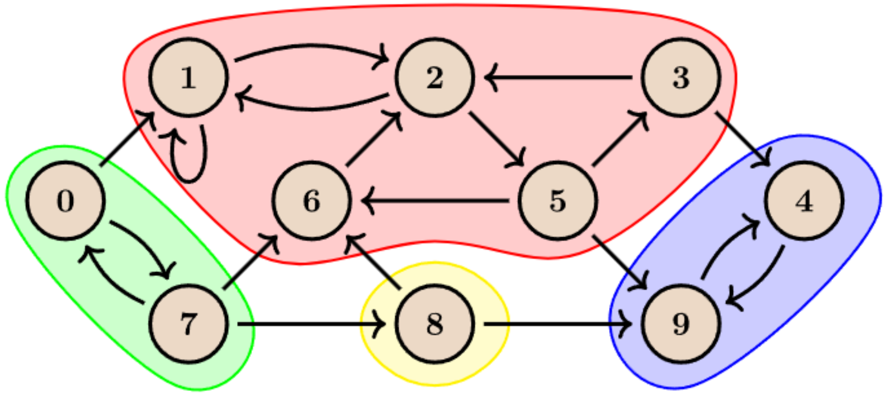
  for all t$\in$T do
    label(t) := label(t) $\cup$ { E(f_1 U f_2) }

  while T $\neq \emptyset$ do
    choose t $\in$T; T := T \ {t};
    for all s such that R(s,t) do
      if E(f_1 U f_2) $\notin$ label(s) and $f_1 \in$ label(s) then
        label(s) : = label(s) $\cup$ {E(f_1 U f_2) };
        T : = T $\cup$ {s}

CTL MC Algorithm: Checking $g = EGf_1$

- s ⊨ **EG** f_1 iff there is a path π starting at s, such that π ⊨ **G** f_1

  iff there is a path from s to a **strongly connected component (SCC)**, where all states satisfy f_1
- An SCC is a subgraph C s.t. every node in C is reachable from any other node in C
  - C is nontrivial if it contains at least one edge. Otherwise, it is trivial.
- An SCC C is maximal (MSCC) if it is not contained in any other SCC
  - Possible to find all MSCC in **linear time** O(|S|+|R|) (Tarjan)



CTL MC Algorithm: Checking $g = EGf_1$

1. Remove from M all states such that $f_1 \notin labels(s)$

2. Resulting model: $M' = (S', R', L')$
   - $S' = \{ s \mid M, s \vDash f_1\}$
   - $R' = (S' \times S') \cap R$
   - $L'(s') = L(s')$ for every $s' \in S'$

3. Theorem: $M, s \vDash EG\ f_1$ if and only if
   - $s \in S'$ and
   - there is **a path** in M' from s to some state t in a **nontrivial MSCC of M′**.

CTL MC Algorithm: Checking $g = EGf_1$

procedure CheckEG (f_1)

$S' := \{s \mid f_1 \in label(s)\}$

$MSCC := \{ C \mid C$ is a nontrivial MSCC of $M' \}$

$T := \bigcup_{C \in MSCC} \{ s \mid s \in C\}$

for all $t \in T$ do

    label(t) := label(t) $\cup$ { EG f_1}

while $T \neq \emptyset$ do

    choose $t \in T$; $T := T \setminus \{t\}$;

    for all $s \in S'$ such that $R'(s,t)$ do

        if EG $f_1 \notin$ label(s) then

            label(s) : = label(s) $\cup$ {EG f_1};

            $T := T \cup \{s\}$

CTL MC Algorithm: Complexity

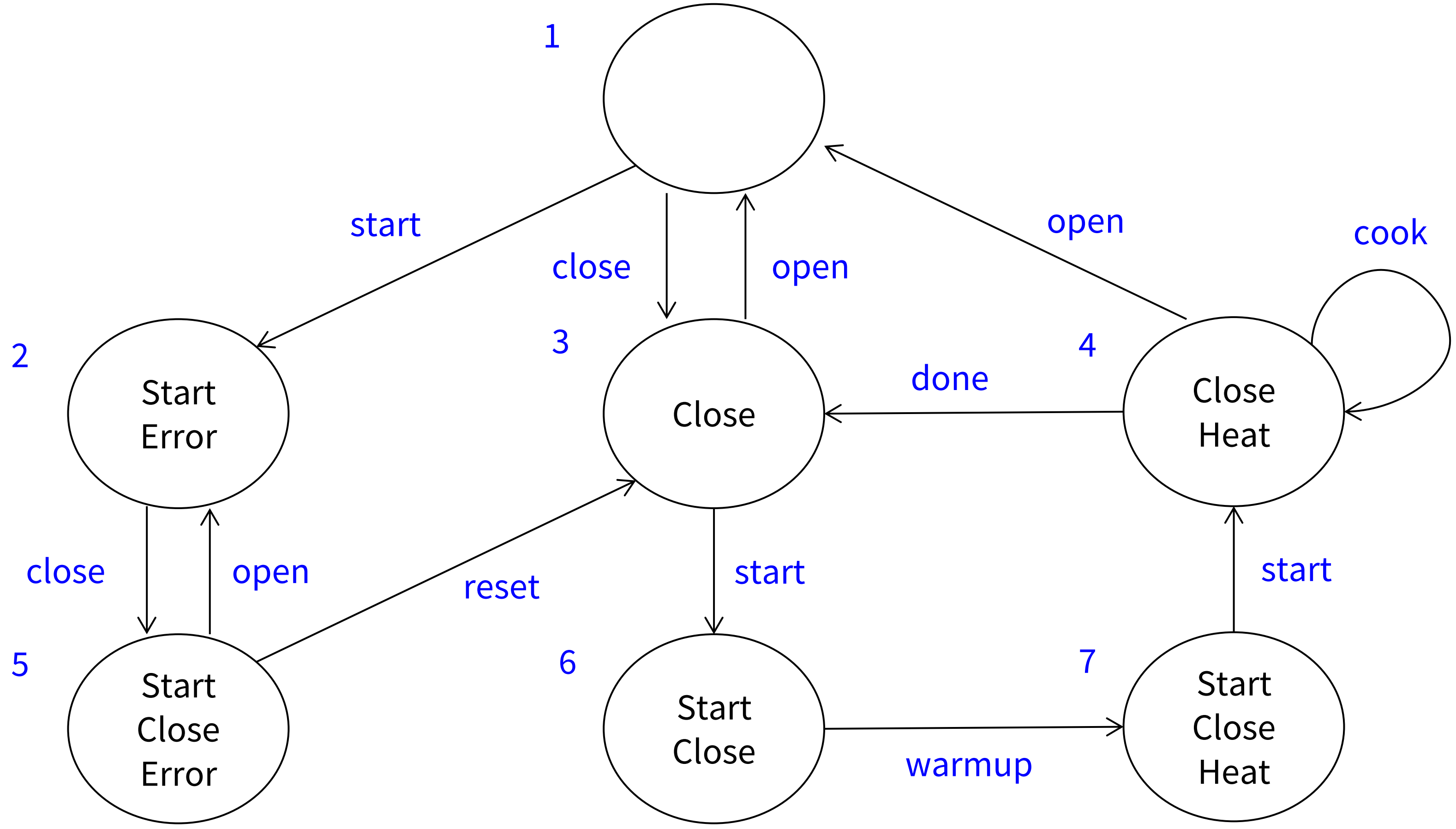
- Steps per sub-formula:
  - MC atomic propositions: O(|S|) steps
  - MC $\neg, \vee$ formulas O(|S|) steps
  - MC $g = EX\ f_1$ O(|S| + |R|) steps
  - MC $g = E(f_1 U\ f_2)$ O(|S| + |R|)
  - MC $g = EG f_1$
    - Computing M′ : O (|S| + |R|)
    - Computing MSCCs using Tarjan's algorithm: O (|S′| + |R′|)
    - Labeling all states in MSCCs: O (|S′|)
    - Backward traversal: O (|S′| + |R′|)
  - => Overall steps per subformula: O (|S| + |R|)

CTL MC Algorithm: Complexity

- Complexity of CTL MC:
  - Steps per sub-formula: $O(|S| + |R|)$
  - Number of sub-formulas in f: $O(|f|)$
  - **Total:** $\mathbf{O(|M| \times |f|)}$
- For comparison
  - Complexity of LTL MC is $O(|M| \times 2^{|f|})$

Model Checking Example

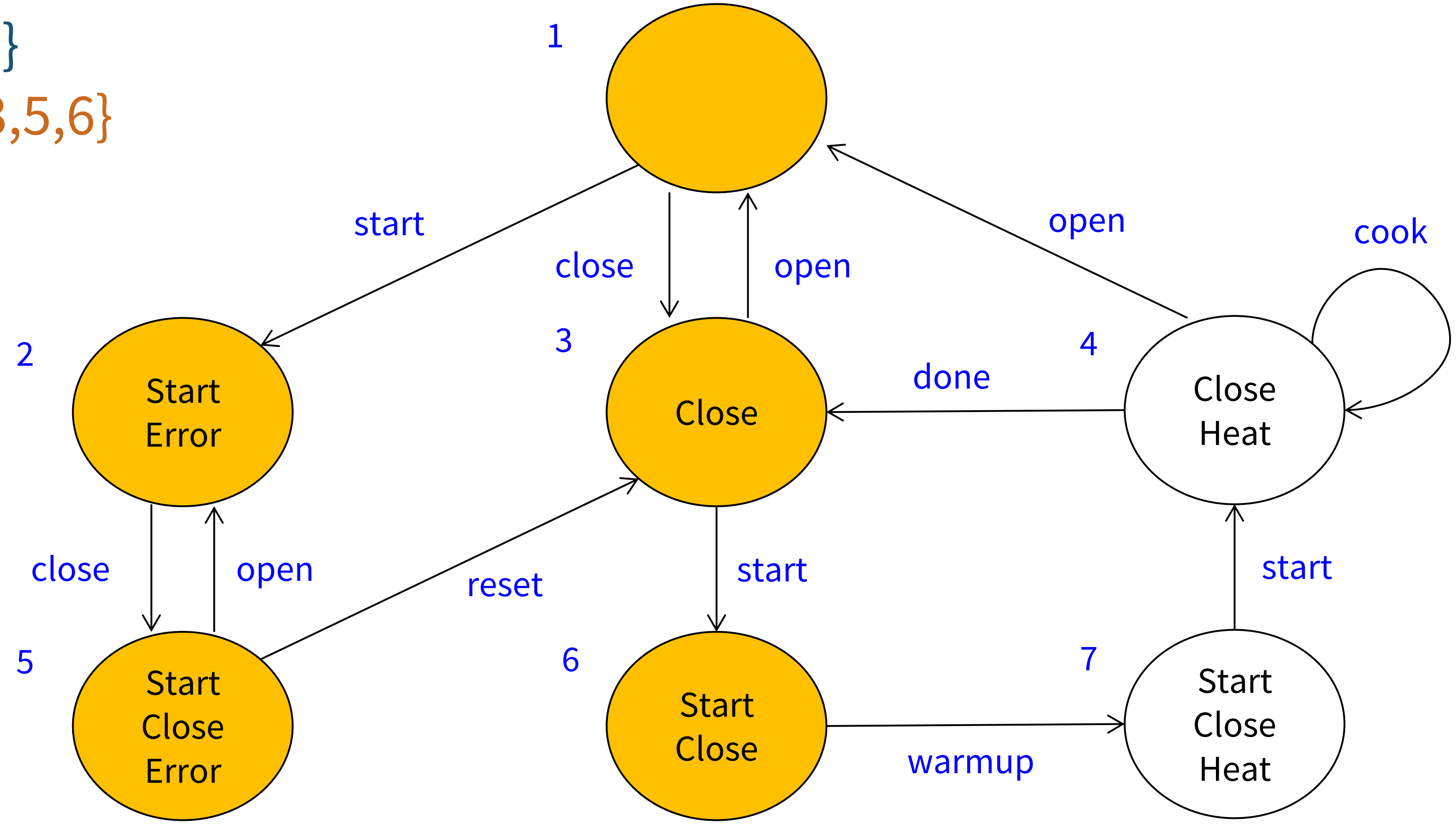
- Does $M \vDash f$ with $f = \neg E(true\ U\ (start \land EG\ \neg Heat))$



Model Checking Example

- Does $M \vDash f$ with $f = \neg E(true\ U\ (\boldsymbol{start} \land EG\ \neg \boldsymbol{heat}))$

⟦start⟧ = {2,5,6,7}
⟦¬heat ⟧ = {1,2,3,5,6}



Model Checking Example

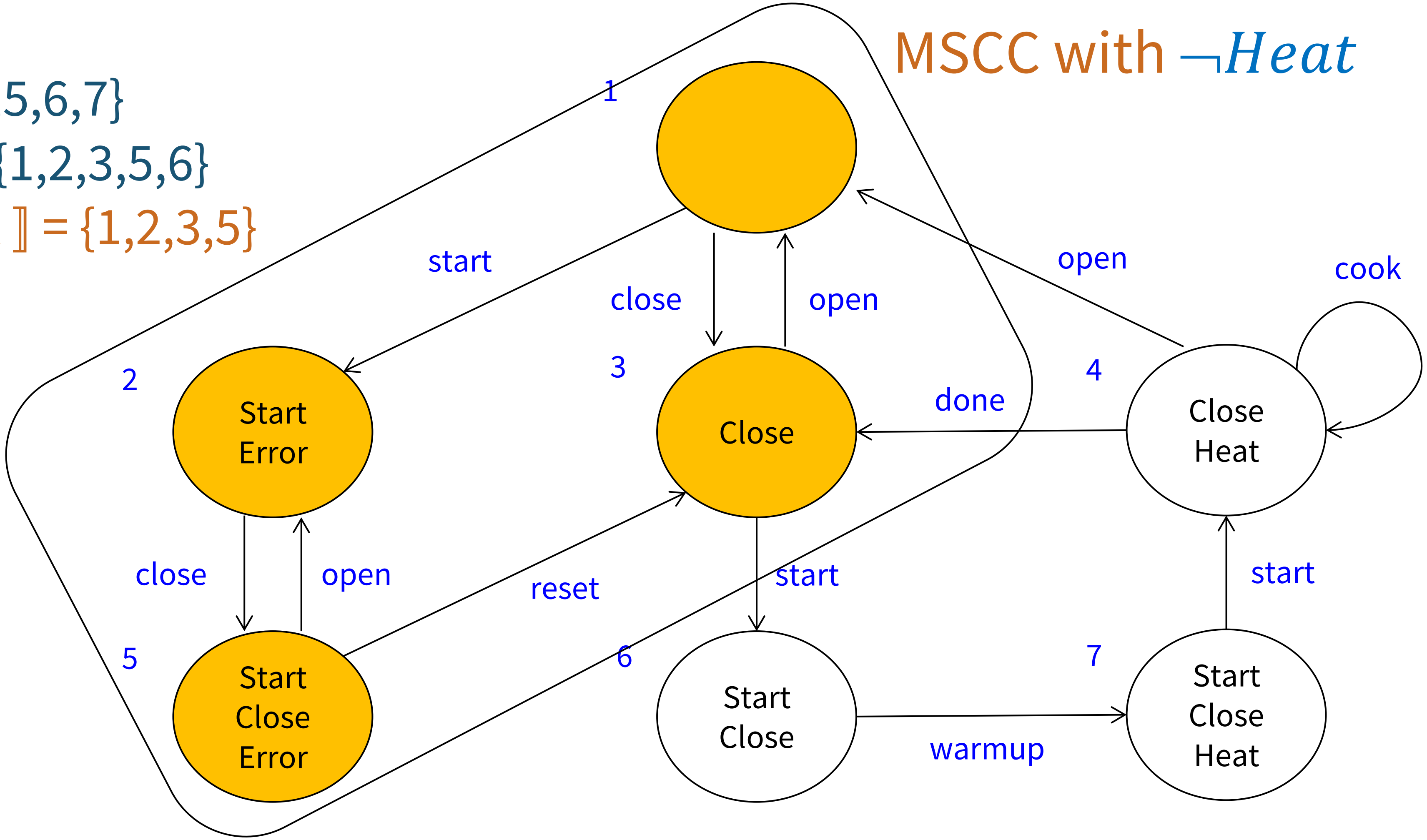
- Does $M \vDash f$ with $f = \neg E(true\ U\ (start \wedge \boldsymbol{EG} \neg \boldsymbol{heat}))$

⟦start⟧ = {2,5,6,7}

⟦¬heat⟧ = {1,2,3,5,6}

⟦(EG ¬heat⟧ = {1,2,3,5}

MSCC with $\neg Heat$

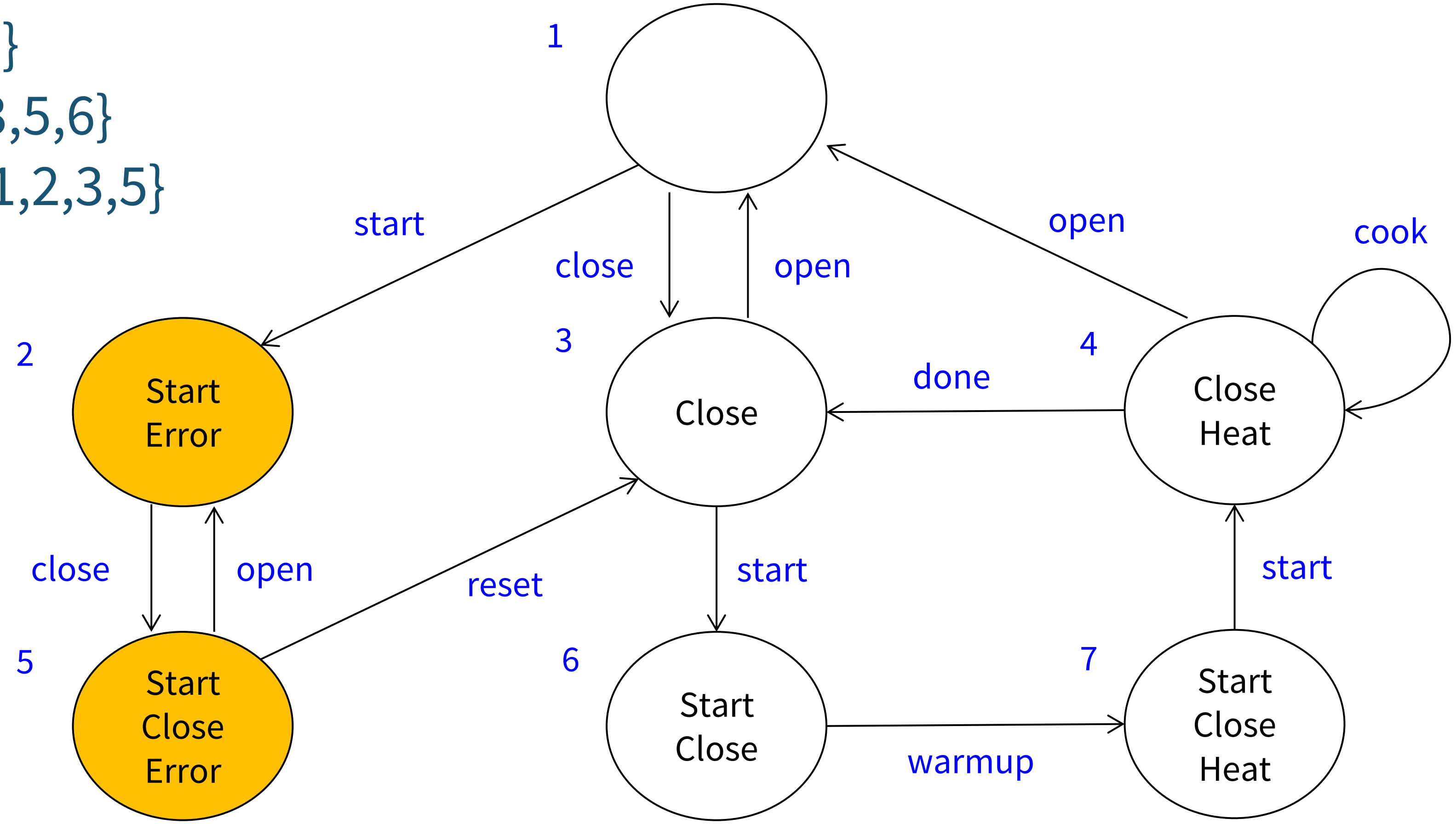


Model Checking Example

- Does $M \models f$ with $f = \neg E(true\ U\ (\boldsymbol{start} \wedge \boldsymbol{EG}\ \neg \boldsymbol{heat}))$

⟦ start ∧ EG ¬heat ⟧ = {2, 5}

⟦start⟧ = {2,5,6,7}
⟦¬heat ⟧ = {1,2,3,5,6}
⟦(EG ¬heat ⟧ = {1,2,3,5}

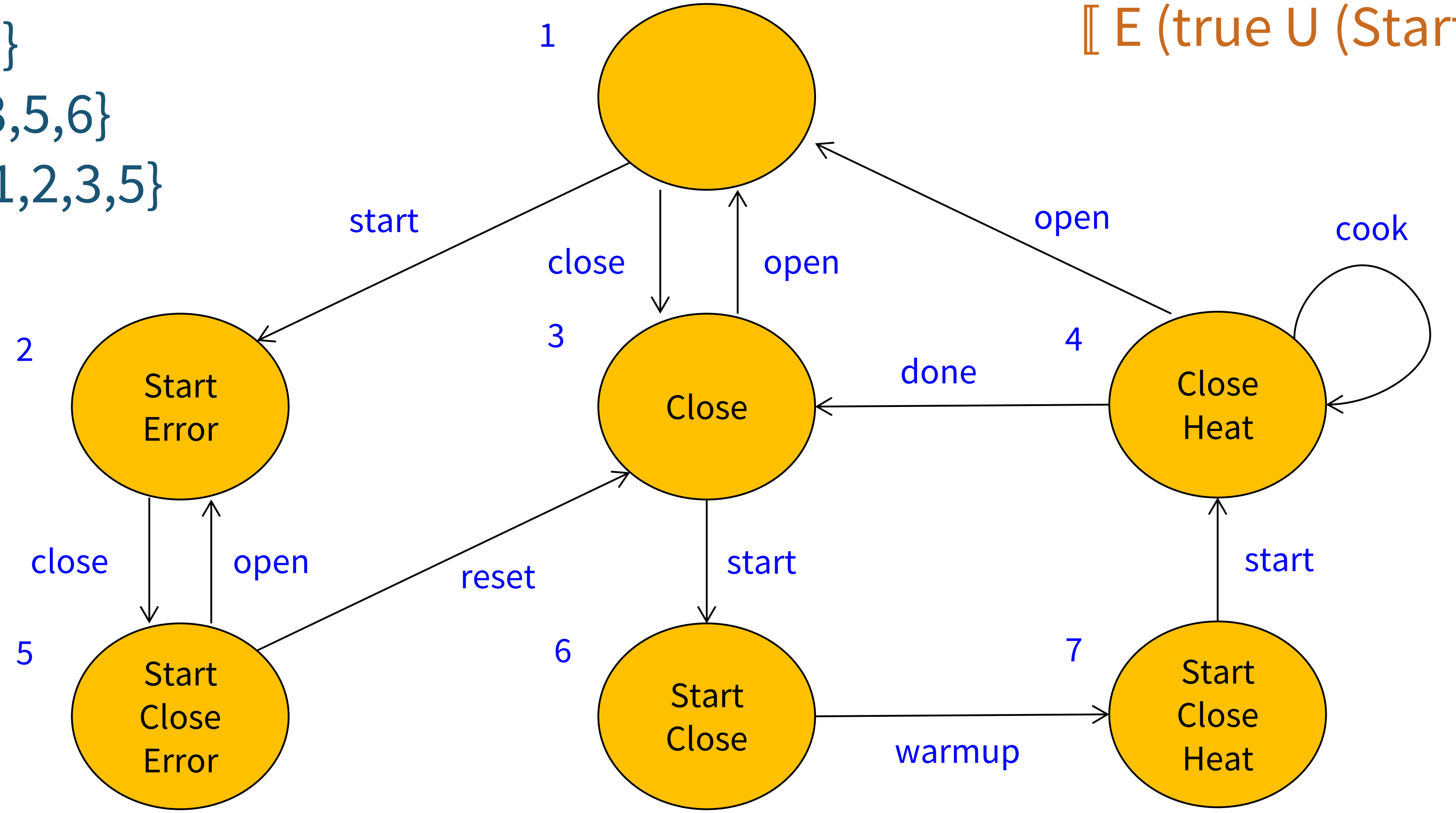


Model Checking Example

- Does $M \vDash f$ with $f = \neg E(\boldsymbol{true}\ \boldsymbol{U}\ (\boldsymbol{start} \wedge \boldsymbol{EG}\ \neg \boldsymbol{heat}))$

⟦start⟧ = {2,5,6,7}
⟦¬heat⟧ = {1,2,3,5,6}
⟦(EG ¬heat⟧ = {1,2,3,5}

⟦start ∧ EG ¬heat⟧ = {2, 5}
⟦E (true U (Start ∧ EG ¬Heat))⟧ = {1,2,3,4,5,6,7}



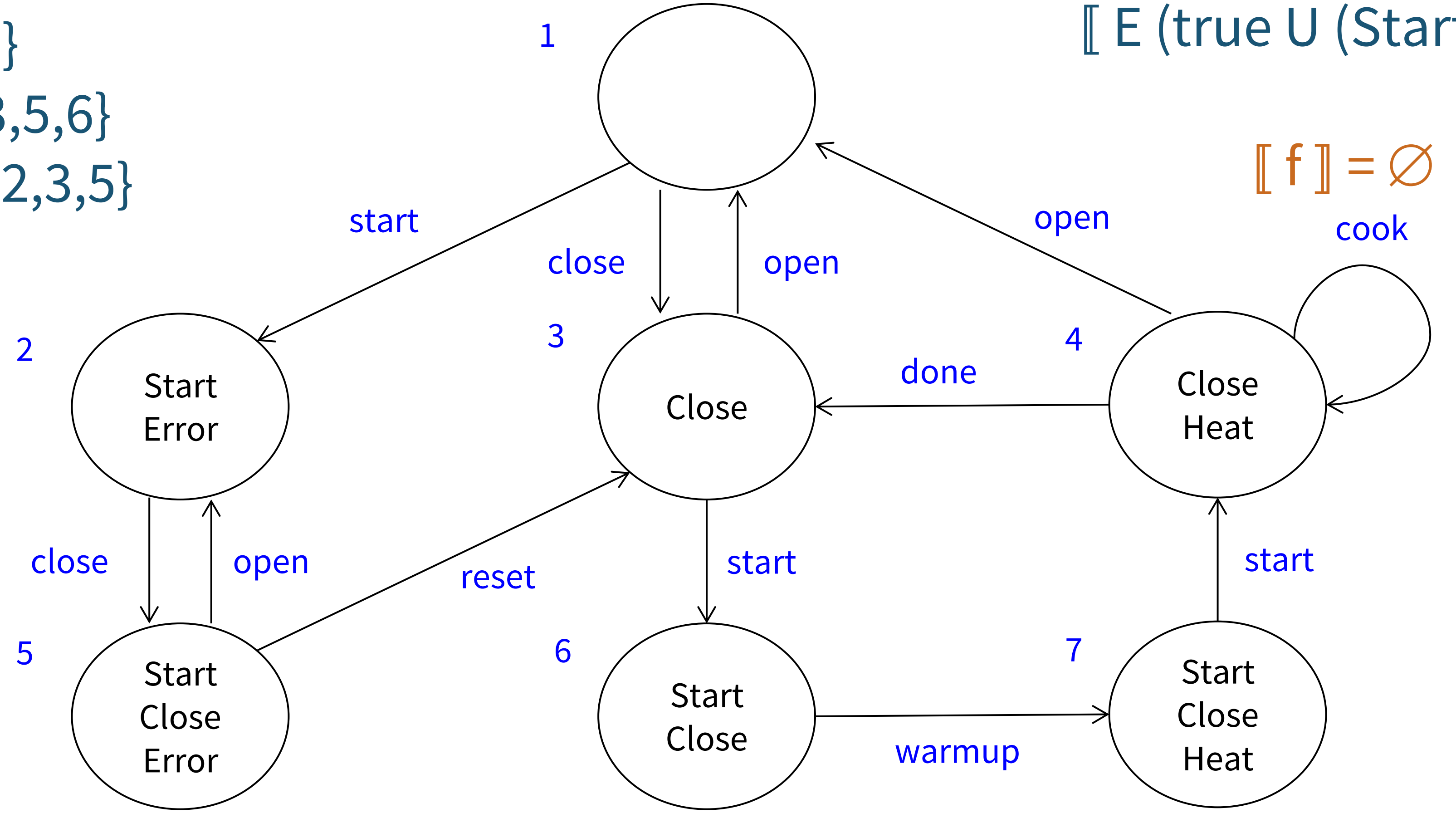
Model Checking Example

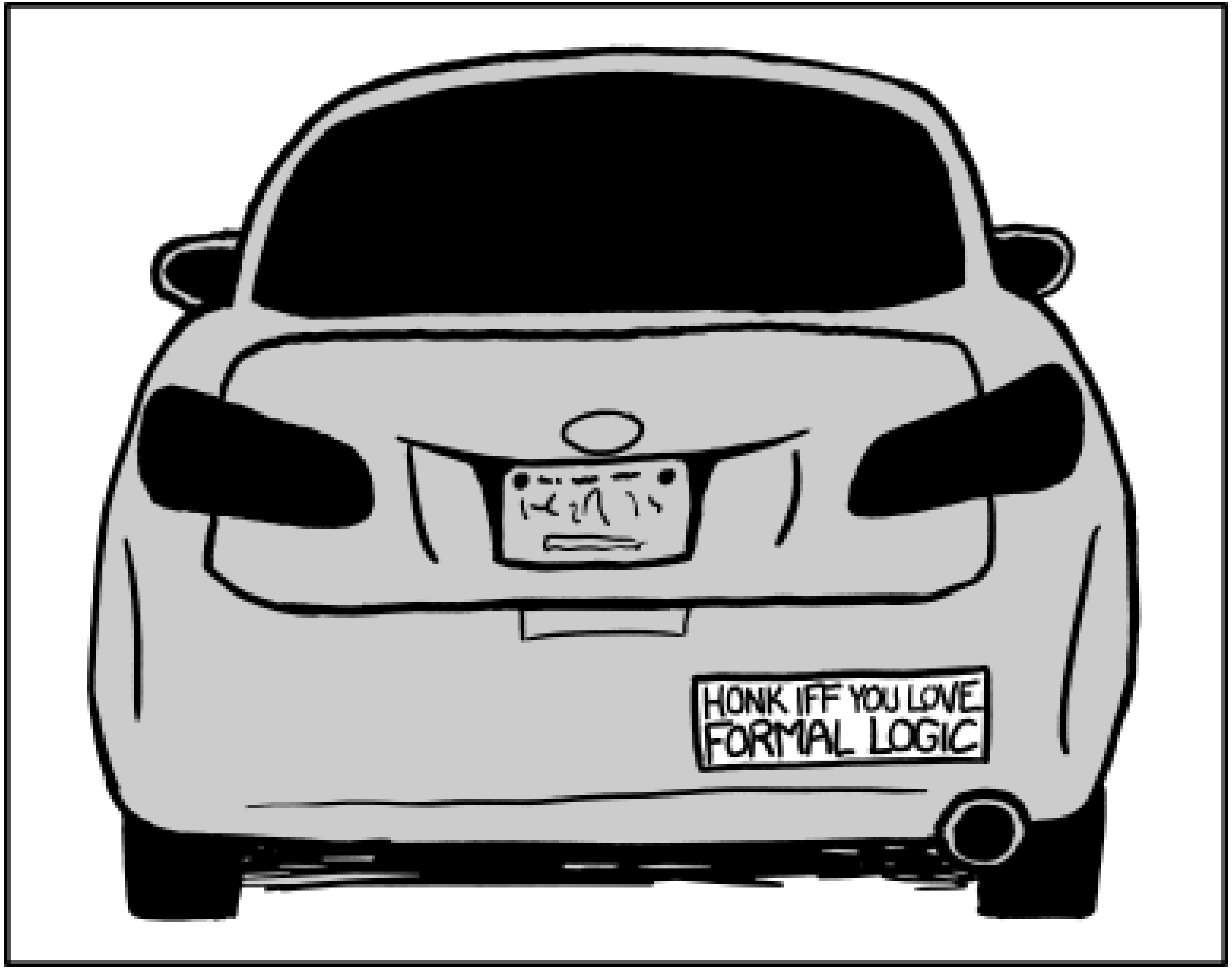
- Does $M \vDash f$ with $f = \neg E(true\ U\ (\boldsymbol{start} \wedge \boldsymbol{EG}\ \neg \boldsymbol{heat}))$

⟦start⟧ = {2,5,6,7}
⟦¬heat ⟧ = {1,2,3,5,6}
⟦(EG heat ⟧ = {1,2,3,5}

⟦ start ∧ EG ¬heat ⟧ = {2, 5}
⟦ E (true U (Start ∧ EG ¬Heat)) ⟧ = {1,2,3,4,5,6,7}

⟦ f ⟧ = ∅




HONK IFF YOU LOVE
FORMAL LOGIC


ANY
QUESTIONS?