

CTL MODEL CHECKING

Slides by Alessandro Artale
<http://www.inf.unibz.it/~artale/>

*Some material (text, figures) displayed in these slides is courtesy of:
M. Benerecetti, A. Cimatti, M. Fisher, F. Giunchiglia, M. Pistore, M. Roveri, R. Sebastiani.*

– p. 1/32

Summary

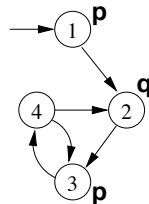
- CTL Model Checking: General Ideas.
- CTL Model Checking: The Labeling Algorithm.
- Labeling Algorithm in Details.
- CTL Model Checking: Theoretical Issues.

– p. 2/32

CTL Model Checking

CTL Model Checking is a formal verification technique s.t.

- The system is represented as a Kripke Model $\mathcal{KM}$:



- The property is expressed as a CTL formula φ , e.g.:

$$\mathbf{AG}(p \Rightarrow \mathbf{AF}q)$$

- The algorithm checks whether **all** the initial states, s_0 , of the Kripke model satisfy the formula ($\mathcal{KM}, s_0 \models \varphi$).

– p. 3/32

CTL M.C. Algorithm: General Ideas

The algorithm proceeds along two macro-steps:

1. Construct the set of states where the formula holds:
$$[[\varphi]] := \{s \in S : \mathcal{KM}, s \models \varphi\}$$

($[[\varphi]]$ is called the **denotation** of φ);
2. Then compare the denotation with the set of initial states:
$$I \subseteq [[\varphi]] ?$$

– p. 4/32

CTL M.C. Algorithm: General Ideas

To compute $\llbracket \varphi \rrbracket$ proceed “bottom-up” on the structure of the formula, computing $\llbracket \varphi_i \rrbracket$ for each subformula φ_i of φ .

For example, to compute $\llbracket \mathbf{AG}(p \Rightarrow \mathbf{AF}q) \rrbracket$ we need to compute:

- $\llbracket q \rrbracket$,
- $\llbracket \mathbf{AF}q \rrbracket$,
- $\llbracket p \rrbracket$,
- $\llbracket p \Rightarrow \mathbf{AF}q \rrbracket$,
- $\llbracket \mathbf{AG}(p \Rightarrow \mathbf{AF}q) \rrbracket$

– p. 5/32

CTL M.C. Algorithm: General Ideas

To compute each $\llbracket \varphi_i \rrbracket$ for generic subformulas:

- Handle boolean operators by standard set operations;
- Handle temporal operators $\mathbf{AX}$, $\mathbf{EX}$ by computing **pre-images**;
- Handle temporal operators $\mathbf{AG}$, $\mathbf{EG}$, $\mathbf{AF}$, $\mathbf{EF}$, $\mathbf{AU}$, $\mathbf{EU}$, by applying **fixpoint** operators.

– p. 6/32

Summary

- CTL Model Checking: General Ideas.
- **CTL Model Checking: The Labeling Algorithm.**
- Labeling Algorithm in Details.
- CTL Model Checking: Theoretical Issues.

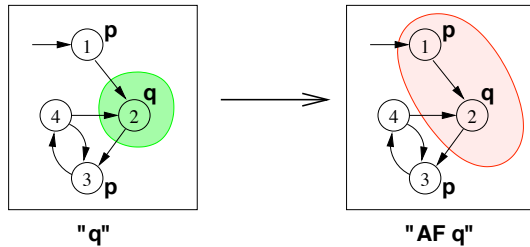
– p. 7/32

The Labeling Algorithm: General Idea

- The **Labeling Algorithm** given a Kripke Model and a CTL formula outputs the set of states satisfying the formula.
- **Main Idea:** Label the states of the Kripke Model with the subformulas of φ satisfied there.

– p. 8/32

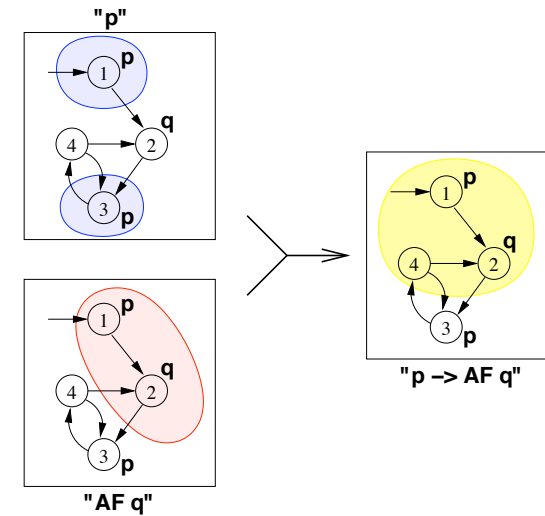
The Labeling Algorithm: An Example



- ▷ $\mathbf{AF}q \equiv (q \vee \mathbf{AX}(\mathbf{AF}q))$
- ▷ $\llbracket \mathbf{AF}q \rrbracket$ can be computed as the union of:
 - $\llbracket q \rrbracket = \{2\}$
 - $\llbracket q \vee \mathbf{AX}q \rrbracket = \{2\} \cup \{1\} = \{1, 2\}$
 - $\llbracket q \vee \mathbf{AX}(q \vee \mathbf{AX}q) \rrbracket = \{2\} \cup \{1\} = \{1, 2\}$ (fixpoint).

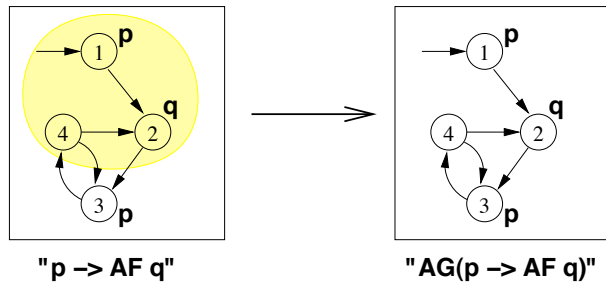
– p. 9/32

The Labeling Algorithm: An Example



– p. 10/32

The Labeling Algorithm: An Example



- ▷ $\mathbf{AG}\varphi \equiv (\varphi \wedge \mathbf{AX}(\mathbf{AG}\varphi))$
- ▷ $\llbracket \mathbf{AG}\varphi \rrbracket$ can be computed as the intersection of:
 - $\llbracket \varphi \rrbracket = \{1, 2, 4\}$
 - $\llbracket \varphi \wedge \mathbf{AX}\varphi \rrbracket = \{1, 2, 4\} \cap \{1, 3\} = \{1\}$
 - $\llbracket \varphi \wedge \mathbf{AX}(\varphi \wedge \mathbf{AX}\varphi) \rrbracket = \{1, 2, 4\} \cap \{\} = \{\}$ (fixpoint)

– p. 11/32

The Labeling Algorithm: An Example

- ▷ The set of states where the formula holds is empty, thus:
 - The initial state does not satisfy the property;
 - $\mathcal{K} \mathcal{M} \not\models \mathbf{AG}(p \Rightarrow \mathbf{AF}q)$.
- ▷ Counterexample: A lazo-shaped path: $1, 2, \{3, 4\}^\omega$ (satisfying $\mathbf{EF}(p \wedge \mathbf{EG}\neg q)$)

– p. 12/32

Summary

- CTL Model Checking: General Ideas.
- CTL Model Checking: The Labeling Algorithm.
- [Labeling Algorithm in Details.](#)
- CTL Model Checking: Theoretical Issues.

– p. 13/32

The Labeling Algorithm: General Schema

- ▷ Assume φ written in terms of $\neg, \wedge, \mathbf{EX}, \mathbf{EU}, \mathbf{EG}$ – minimal set of CTL operators
- ▷ The Labeling algorithm takes a CTL formula and a Kripke Model as input and returns the set of states satisfying the formula (i.e., the *denotation* of φ):
 1. For every $\varphi_i \in \text{Sub}(\varphi)$, find $\llbracket \varphi_i \rrbracket$;
 2. Compute $\llbracket \varphi \rrbracket$ starting from $\llbracket \varphi_i \rrbracket$;
 3. Check if $I \subseteq \llbracket \varphi \rrbracket$.
- ▷ Subformulas $\text{Sub}(\varphi)$ of φ are checked bottom-up
- ▷ To compute each $\llbracket \varphi_i \rrbracket$: if the main operator of φ_i is a
 - *Boolean Operator*: apply standard set operations;
 - *Temporal Operator*: apply recursive rules until a **fixpoint** is reached.

– p. 14/32

Denotation of Formulas: The Boolean Case

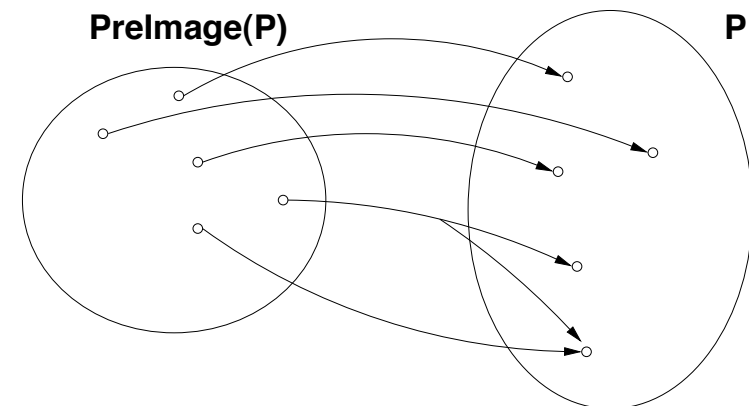
Let $\mathcal{KM} = \langle S, I, R, L, \Sigma \rangle$ be a Kripke Model.

$$\begin{aligned}
 \llbracket \text{false} \rrbracket &= \{\} \\
 \llbracket \text{true} \rrbracket &= S \\
 \llbracket p \rrbracket &= \{s \mid p \in L(s)\} \\
 \llbracket \neg \varphi_1 \rrbracket &= S \setminus \llbracket \varphi_1 \rrbracket \\
 \llbracket \varphi_1 \wedge \varphi_2 \rrbracket &= \llbracket \varphi_1 \rrbracket \cap \llbracket \varphi_2 \rrbracket
 \end{aligned}$$

– p. 15/32

Denotation of Formulas: The EX Case

- ▷ $\llbracket \mathbf{EX} \varphi \rrbracket = \{s \in S \mid \exists s'. \langle s, s' \rangle \in R \text{ and } s' \in \llbracket \varphi \rrbracket\}$
- ▷ $\llbracket \mathbf{EX} \varphi \rrbracket$ is said to be the **Pre-image of $\llbracket \varphi \rrbracket$** ($\text{PRE}(\llbracket \varphi \rrbracket)$).
- ▷ Key step of every CTL M.C. operation.



– p. 16/32

Denotation of Formulas: The EG Case

- From the semantics of the $\Box$ temporal operator:
$$\Box\varphi \equiv \varphi \wedge \bigcirc(\Box\varphi)$$
- Then, the following equivalence holds:
$$\mathbf{EG}\varphi \equiv \varphi \wedge \mathbf{EX}(\mathbf{EG}\varphi)$$
- To compute $\llbracket \mathbf{EG}\varphi \rrbracket$ we can apply the following recursive definition:
$$\llbracket \mathbf{EG}\varphi \rrbracket = \llbracket \varphi \rrbracket \cap \mathbf{PRE}(\llbracket \mathbf{EG}\varphi \rrbracket)$$

– p. 17/32

Denotation of Formulas: The EG Case

- We can compute $X := \llbracket \mathbf{EG}\varphi \rrbracket$ inductively as follows:
$$\begin{aligned} X_1 &:= \llbracket \varphi \rrbracket \\ X_2 &:= X_1 \cap \mathbf{PRE}(X_1) \\ &\dots \\ X_{j+1} &:= X_j \cap \mathbf{PRE}(X_j) \end{aligned}$$
- When $X_n = X_{n+1}$ we reach a **fixpoint** and we stop.
- Termination.** Since $X_{j+1} \subseteq X_j$ for every $j \geq 0$, thus a **fixed point always exists** (Knaster-Tarski's theorem).

– p. 18/32

Denotation of Formulas: The EU Case

- From the semantics of the $\mathcal{U}$ temporal operator:
$$\varphi \mathcal{U} \psi \equiv \psi \vee (\varphi \wedge \bigcirc(\varphi \mathcal{U} \psi))$$
- Then, the following equivalence holds:
$$(\varphi \mathbf{EU} \psi) \equiv \psi \vee (\varphi \wedge \mathbf{EX}(\varphi \mathbf{EU} \psi))$$
- To compute $\llbracket (\varphi \mathbf{EU} \psi) \rrbracket$ we can apply the following recursive definition:
$$\llbracket (\varphi \mathbf{EU} \psi) \rrbracket = \llbracket \psi \rrbracket \cup (\llbracket \varphi \rrbracket \cap \mathbf{PRE}(\llbracket (\varphi \mathbf{EU} \psi) \rrbracket))$$

– p. 19/32

Denotation of Formulas: The EU Case

- We can compute $X := \llbracket (\varphi \mathbf{EU} \psi) \rrbracket$ inductively as follows:
$$\begin{aligned} X_1 &:= \llbracket \psi \rrbracket \\ X_2 &:= X_1 \cup (\llbracket \varphi \rrbracket \cap \mathbf{PRE}(X_1)) \\ &\dots \\ X_{j+1} &:= X_j \cup (\llbracket \varphi \rrbracket \cap \mathbf{PRE}(X_j)) \end{aligned}$$
- When $X_n = X_{n+1}$ we reach a **fixpoint** and we stop.
- Termination.** Since $X_{j+1} \supseteq X_j$ for every $j \geq 0$, thus a **fixed point always exists** (Knaster-Tarski's theorem).

– p. 20/32

The Pseudo-Code

We assume the Kripke Model to be a global variable:

```
FUNCTION Label( $\varphi$ ) {  
  case  $\varphi$  of  
    true:      return  $S$ ;  
    false:     return  $\{\}$ ;  
    an atom  $p$ : return  $\{s \in S \mid p \in L(s)\}$ ;  
     $\neg\varphi_1$ :     return  $S \setminus \text{Label}(\varphi_1)$ ;  
     $\varphi_1 \wedge \varphi_2$ : return  $\text{Label}(\varphi_1) \cap \text{Label}(\varphi_2)$ ;  
     $\text{EX}\varphi_1$ :     return  $\text{PRE}(\text{Label}(\varphi_1))$ ;  
     $(\varphi_1 \text{EU}\varphi_2)$ : return  $\text{Label\_EU}(\text{Label}(\varphi_1), \text{Label}(\varphi_2))$ ;  
     $\text{EG}\varphi_1$ :     return  $\text{Label\_EG}(\text{Label}(\varphi_1))$ ;  
  end case  
}
```

– p. 21/32

PreImage

$$\llbracket \text{EX}\varphi \rrbracket = \text{PRE}(\llbracket \varphi \rrbracket) = \{s \in S \mid \exists s'. \langle s, s' \rangle \in R \text{ and } s' \in \llbracket \varphi \rrbracket\}$$

```
FUNCTION PRE( $\llbracket \varphi \rrbracket$ ) {  
  var  $X$ ;  
   $X := \{\}$ ;  
  for each  $s' \in \llbracket \varphi \rrbracket$  do  
    for each  $s \in S$  such that  $\langle s, s' \rangle \in R$  do  
       $X := X \cup \{s\}$ ;  
  return  $X$   
}
```

– p. 22/32

Label_EG

$$\llbracket \text{EG}\varphi \rrbracket = \llbracket \varphi \rrbracket \cap \text{PRE}(\llbracket \text{EG}\varphi \rrbracket)$$

```
FUNCTION LABEL_EG( $\llbracket \varphi \rrbracket$ ) {  
  var  $X, \text{OLD-}X$ ;  
   $X := \llbracket \varphi \rrbracket$ ;  
   $\text{OLD-}X := \emptyset$ ;  
  while  $X \neq \text{OLD-}X$   
  begin  
     $\text{OLD-}X := X$ ;  
     $X := X \cap \text{PRE}(X)$   
  end  
  return  $X$   
}
```

– p. 23/32

Label_EU

$$\llbracket (\varphi \text{EU}\psi) \rrbracket = \llbracket \psi \rrbracket \cup (\llbracket \varphi \rrbracket \cap \text{PRE}(\llbracket (\varphi \text{EU}\psi) \rrbracket))$$

```
FUNCTION LABEL_EU( $\llbracket \varphi \rrbracket, \llbracket \psi \rrbracket$ ) {  
  var  $X, \text{OLD-}X$ ;  
   $X := \llbracket \psi \rrbracket$ ;  
   $\text{OLD-}X := S$ ;  
  while  $X \neq \text{OLD-}X$   
  begin  
     $\text{OLD-}X := X$ ;  
     $X := X \cup (\llbracket \varphi \rrbracket \cap \text{PRE}(X))$   
  end  
  return  $X$   
}
```

– p. 24/32

Summary

- CTL Model Checking: General Ideas.
- CTL Model Checking: The Labeling Algorithm.
- Labeling Algorithm in Details.
- CTL Model Checking: Theoretical Issues.

– p. 25/32

Correctness and Termination

- The Labeling algorithm works recursively on the structure φ .
- For most of the logical constructors the algorithm does the correct things according to the semantics of CTL.
- To prove that the algorithm is *Correct* and *Terminating* we need to prove the correctness and termination of both **EG** and **EU** operators.

– p. 26/32

Monotone Functions and Fixpoints

Definition. Let S be a set and F a function, $F : 2^S \rightarrow 2^S$, then:

1. F is **monotone** iff $X \subseteq Y$ then $F(X) \subseteq F(Y)$;
2. A subset X of S is called a **fixpoint** of F iff $F(X) = X$;
3. X is a **least fixpoint** (LFP) of F , written $\mu X.F(X)$, iff, for every other fixpoint Y of F , $X \subseteq Y$
4. X is a **greatest fixpoint** (GFP) of F , written $\nu X.F(X)$, iff, for every other fixpoint Y of F , $Y \subseteq X$

Example. Let $S = \{s_0, s_1\}$ and $F(X) = X \cup \{s_0\}$.

– p. 27/32

Knaster-Tarski Theorem

Notation: $F^i(X)$ means applying F i -times, i.e., $F(F(\dots F(X)\dots))$.

Theorem[Knaster-Tarski]. Let S be a finite set with $n + 1$ elements. If $F : 2^S \rightarrow 2^S$ is a monotone function then:

1. $\mu X.F(X) \equiv F^{n+1}(\emptyset)$;
2. $\nu X.F(X) \equiv F^{n+1}(S)$.

– p. 28/32

Correctness and Termination: EG Case

The function LABEL_EG computes:

$$[[\mathbf{EG}\varphi]] = [[\varphi]] \cap \mathbf{PRE}([[\mathbf{EG}\varphi]])$$

applying the semantic equivalence:

$$\mathbf{EG}\varphi \equiv \varphi \wedge \mathbf{EX}(\mathbf{EG}\varphi)$$

Thus, $[[\mathbf{EG}\varphi]]$ is the **fixpoint** of the function:

$$F(X) = [[\varphi]] \cap \mathbf{PRE}(X)$$

– p. 29/32

Correctness and Termination: EG Case

Theorem. Let $F(X) = [[\varphi]] \cap \mathbf{PRE}(X)$, and let S have $n + 1$ elements. Then:

1. F is monotone;
2. $[[\mathbf{EG}\varphi]]$ is the **greatest fixpoint** of F .

– p. 30/32

Correctness and Terminationpr: EU Case

The function LABEL_EU computes:

$$[[(\varphi\mathbf{EU}\psi)]] = [[\psi]] \cup ([[\varphi]] \cap \mathbf{PRE}([[(\varphi\mathbf{EU}\psi)]]))$$

applying the semantic equivalence:

$$(\varphi\mathbf{EU}\psi) \equiv \psi \vee (\varphi \wedge \mathbf{EX}(\varphi\mathbf{EU}\psi))$$

Thus, $[(\varphi\mathbf{EU}\psi)]$ is the **fixpoint** of the function:

$$F(X) = [[\psi]] \cup ([[\varphi]] \cap \mathbf{PRE}(X))$$

– p. 31/32

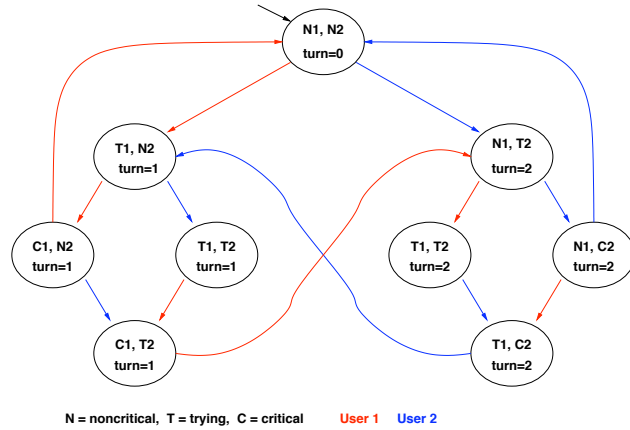
Correctness and Termination: EU Case

Theorem. Let $F(X) = [[\psi]] \cup ([[\varphi]] \cap \mathbf{PRE}(X))$, and let S have $n + 1$ elements. Then:

1. F is monotone;
2. $[(\varphi\mathbf{EU}\psi)]$ is the **least fixpoint** of F .

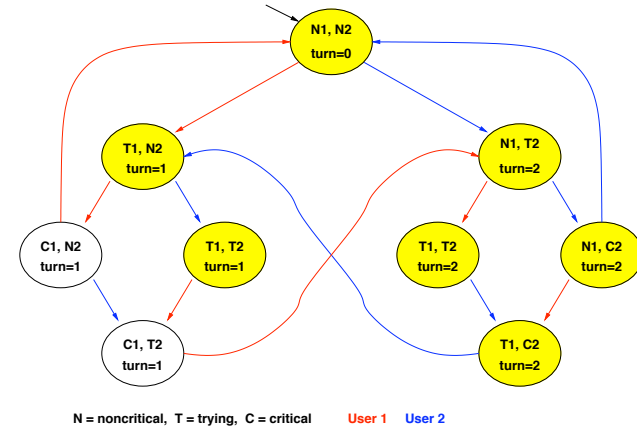
– p. 32/32

Example 1: fairness



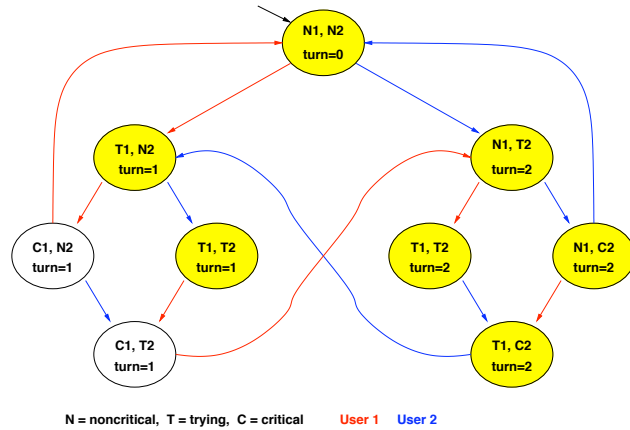
$$M \models \text{AGAF}C_1 ? \implies M \models \neg \text{EFEG} \neg C_1 ?$$

Example 1: fairness

$$[\neg C_1]$$


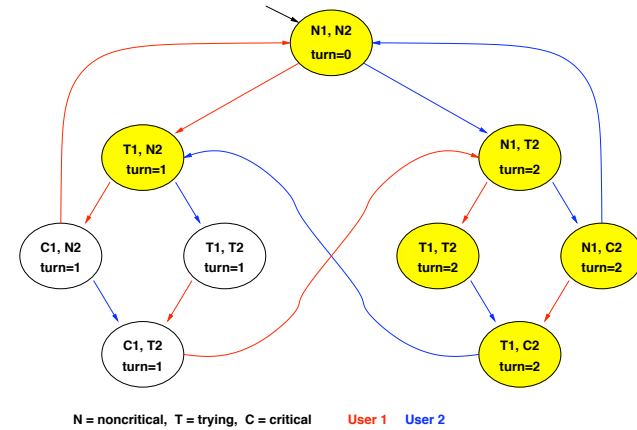
$$M \models \text{AGAF}C_1 ? \implies M \models \neg \text{EFEG} \neg C_1 ?$$

Example 1: fairness

$$[\text{EG} \neg C_1], \text{ step 0:}$$


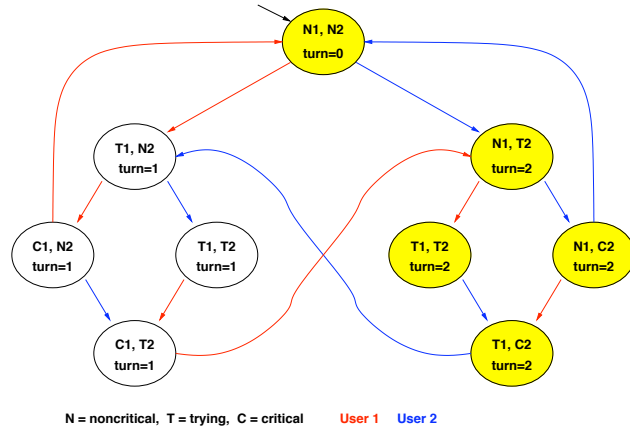
$$M \models \text{AGAF}C_1 ? \implies M \models \neg \text{EFEG} \neg C_1 ?$$

Example 1: fairness

$$[\text{EG} \neg C_1], \text{ step 1:}$$


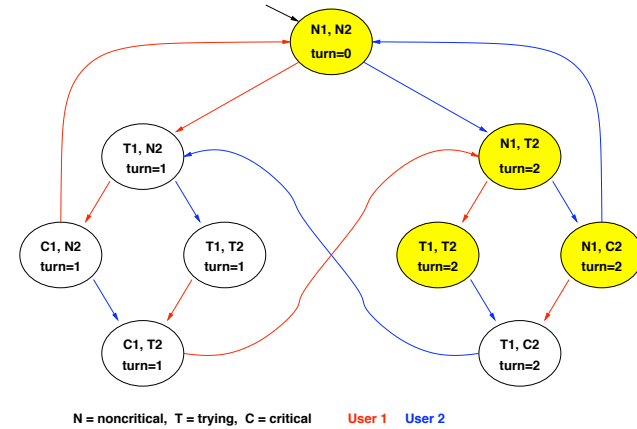
$$M \models \text{AGAF}C_1 ? \implies M \models \neg \text{EFEG} \neg C_1 ?$$

Example 1: fairness

[EG-C₁], step 2:

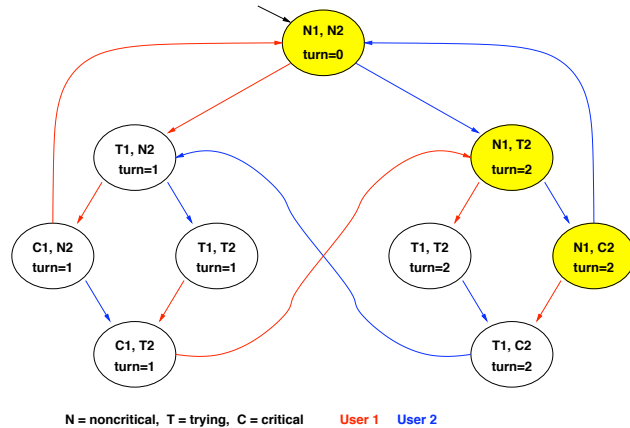
$$M \models \text{AGAF}C_1 ? \implies M \models \neg \text{EFEG}-C_1 ?$$

Example 1: fairness

[EG-C₁], step 3:

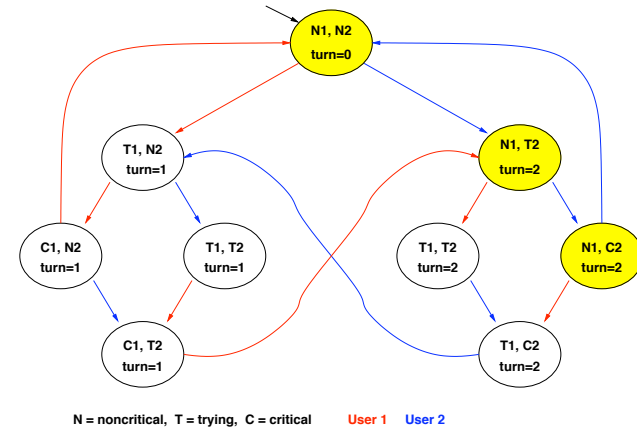
$$M \models \text{AGAF}C_1 ? \implies M \models \neg \text{EFEG}-C_1 ?$$

Example 1: fairness

[EG-C₁], step 4:

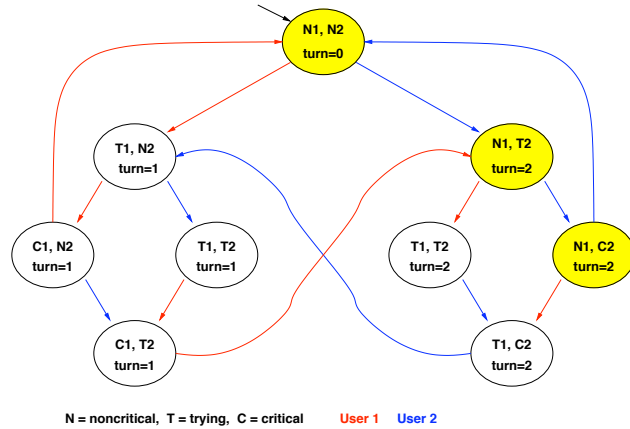
$$M \models \text{AGAF}C_1 ? \implies M \models \neg \text{EFEG}-C_1 ?$$

Example 1: fairness

[EG-C₁], FIXPOINT!

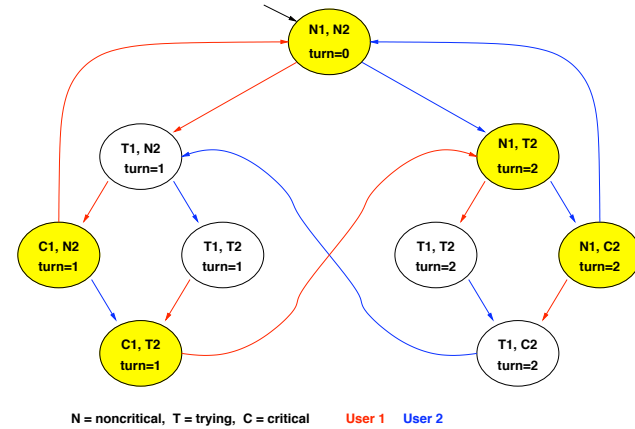
$$M \models \text{AGAF}C_1 ? \implies M \models \neg \text{EFEG}-C_1 ?$$

Example 1: fairness

[EFEG-C₁], STEP 0

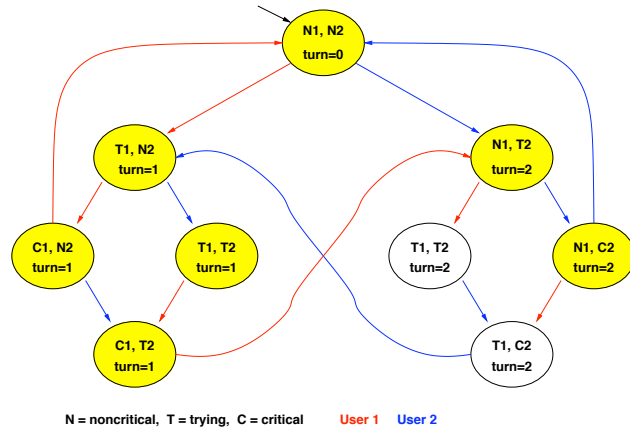
$$M \models \text{AGAF}C_1 ? \implies M \models \neg \text{EFEG}-C_1 ?$$

Example 1: fairness

[EFEG-C₁], STEP 1

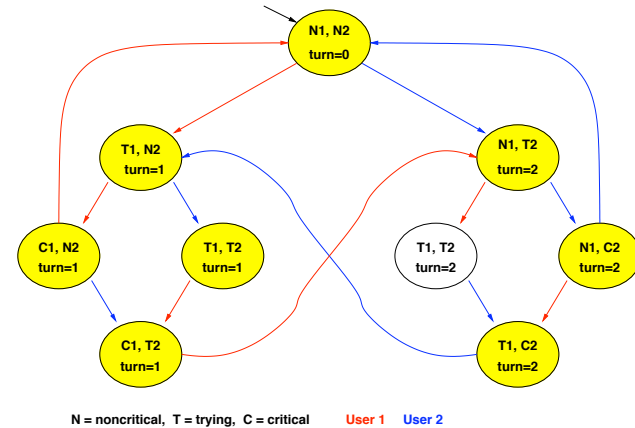
$$M \models \text{AGAF}C_1 ? \implies M \models \neg \text{EFEG}-C_1 ?$$

Example 1: fairness

[EFEG-C₁], STEP 2

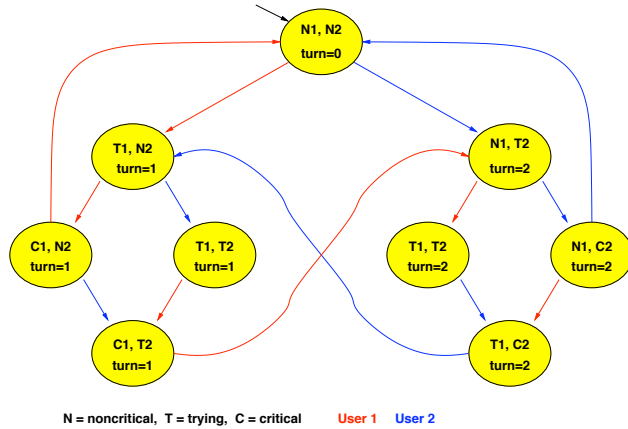
$$M \models \text{AGAF}C_1 ? \implies M \models \neg \text{EFEG}-C_1 ?$$

Example 1: fairness

[EFEG-C₁], STEP 3

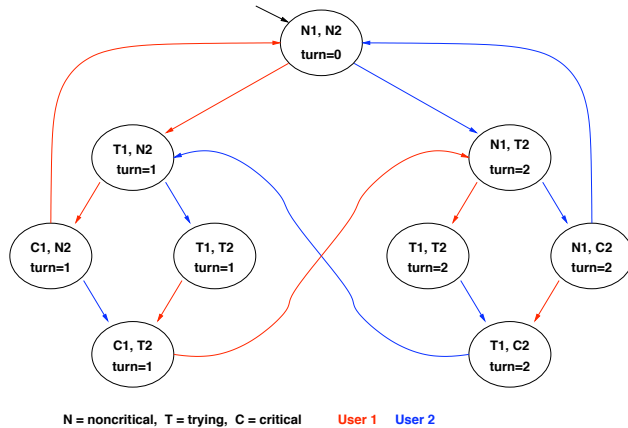
$$M \models \text{AGAF}C_1 ? \implies M \models \neg \text{EFEG}-C_1 ?$$

Example 1: fairness

[EFEG-C₁], STEP 4

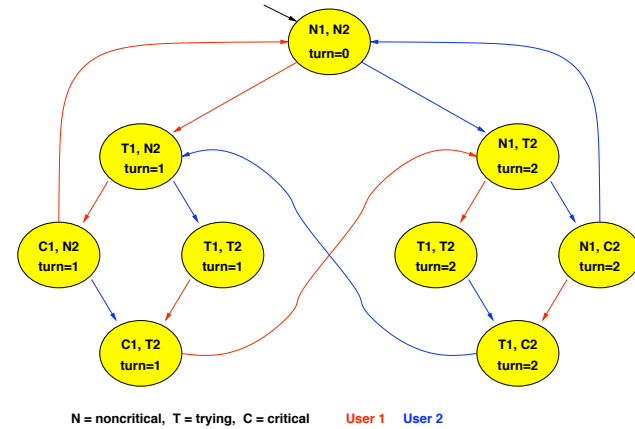
$$M \models \mathbf{AGAF}C_1 ? \implies M \models \neg \mathbf{EFEG}\neg C_1 ?$$

Example 1: fairness

[¬EFEG-C₁]

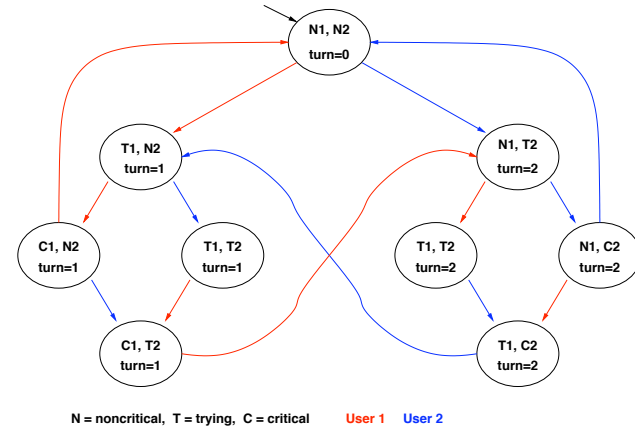
$$M \models \mathbf{AGAF}C_1 ? \implies M \models \neg \mathbf{EFEG}\neg C_1 ? \implies \mathbf{NO!}$$

Example 1: fairness

[EFEG-C₁], FIXPOINT!

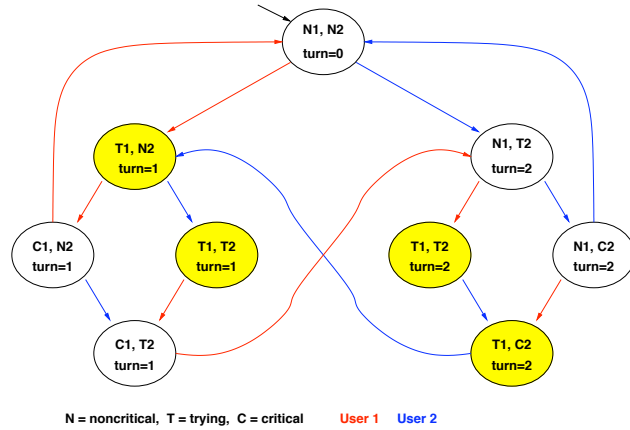
$$M \models \mathbf{AGAF}C_1 ? \implies M \models \neg \mathbf{EFEG}\neg C_1 ?$$

Example 2: liveness



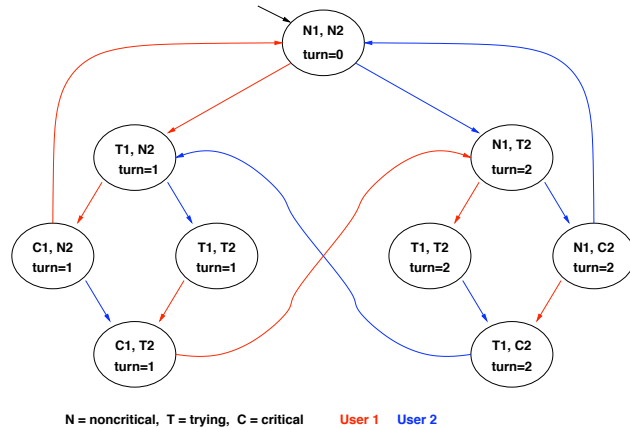
$$M \models \mathbf{AG}(T_1 \rightarrow \mathbf{AF}C_1) ? \implies M \models \neg \mathbf{EF}(T_1 \wedge \mathbf{EG}\neg C_1) ?$$

Example 2: liveness

 $[T_1]$:

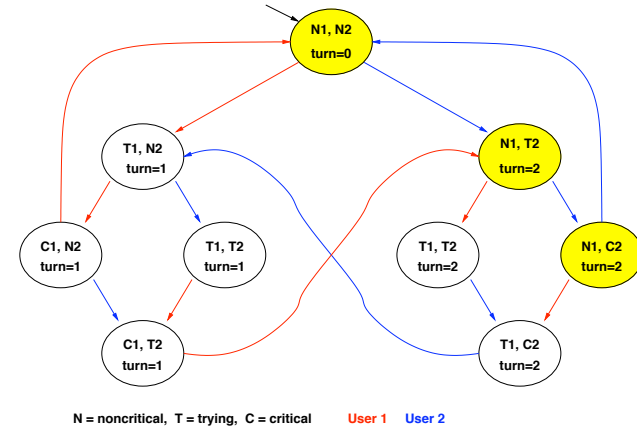
$$M \models \mathbf{AG}(T_1 \rightarrow \mathbf{AFC}_1) ? \implies M \models \neg \mathbf{EF}(T_1 \wedge \mathbf{EG}\neg C_1) ?$$

Example 2: liveness

 $[T_1 \wedge \mathbf{EG}\neg C_1]$:

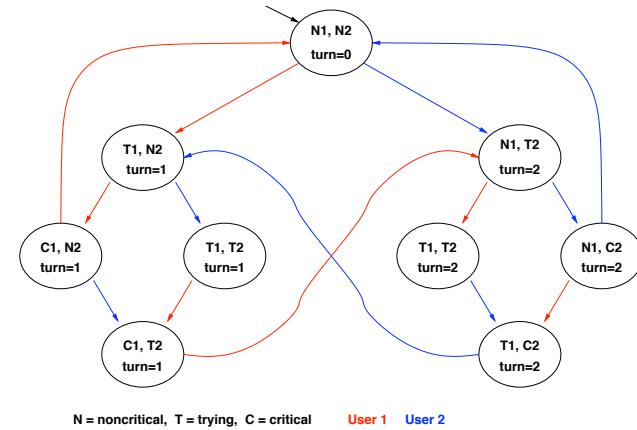
$$M \models \mathbf{AG}(T_1 \rightarrow \mathbf{AFC}_1) ? \implies M \models \neg \mathbf{EF}(T_1 \wedge \mathbf{EG}\neg C_1) ?$$

Example 2: liveness

 $[\mathbf{EG}\neg C_1]$, STEPS 0-4: (see previous example)

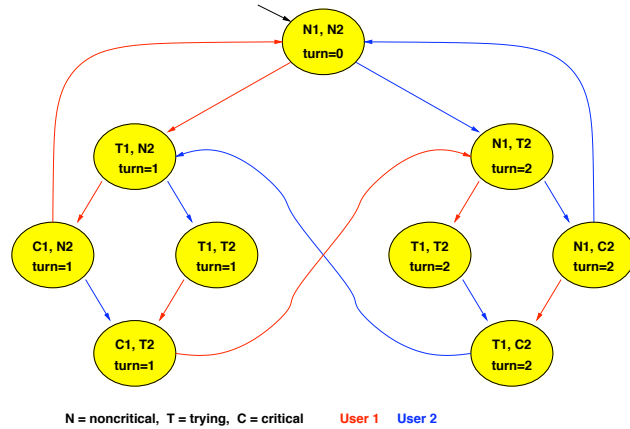
$$M \models \mathbf{AG}(T_1 \rightarrow \mathbf{AFC}_1) ? \implies M \models \neg \mathbf{EF}(T_1 \wedge \mathbf{EG}\neg C_1) ?$$

Example 2: liveness

 $[\mathbf{EF}(T_1 \wedge \mathbf{EG}\neg C_1)]$:

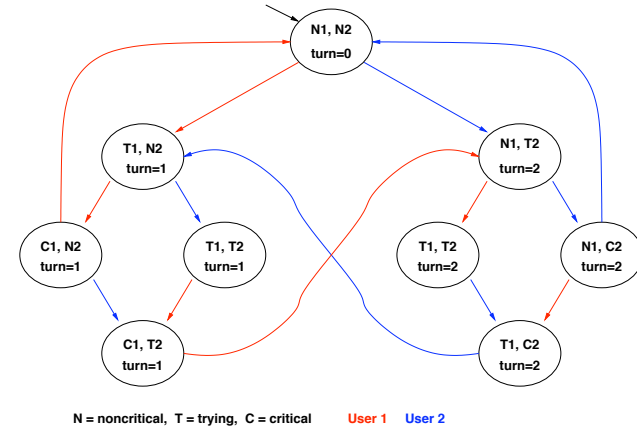
$$M \models \mathbf{AG}(T_1 \rightarrow \mathbf{AFC}_1) ? \implies M \models \neg \mathbf{EF}(T_1 \wedge \mathbf{EG}\neg C_1) ?$$

Example 2: liveness

$$[\neg \mathbf{EF}(T_1 \wedge \mathbf{EG}\neg C_1)] :$$


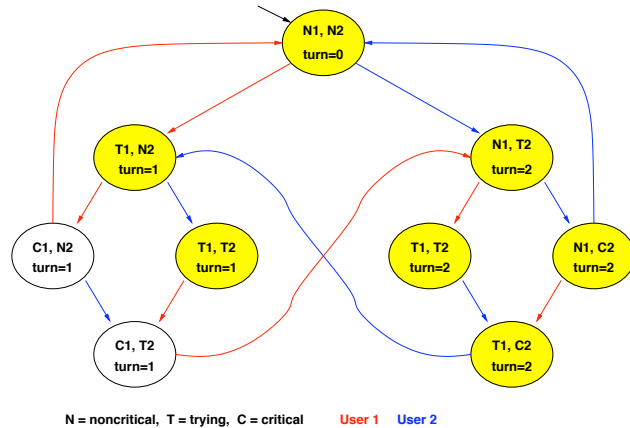
$$M \models \mathbf{AG}(T_1 \rightarrow \mathbf{AFC}_1) ? \implies M \models \neg \mathbf{EF}(T_1 \wedge \mathbf{EG}\neg C_1) ? \text{ YES!}$$

Example 1: fairness



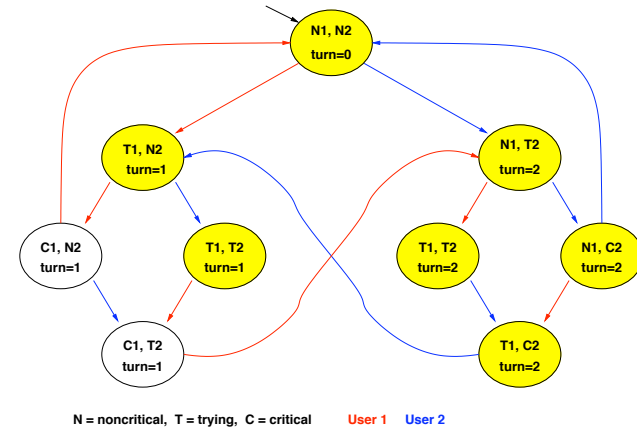
$$M \models \mathbf{AGAFC}_1 ? \implies M \models \neg \mathbf{EFEG}\neg C_1 ?$$

Example 1: fairness

$$[\neg C_1]$$


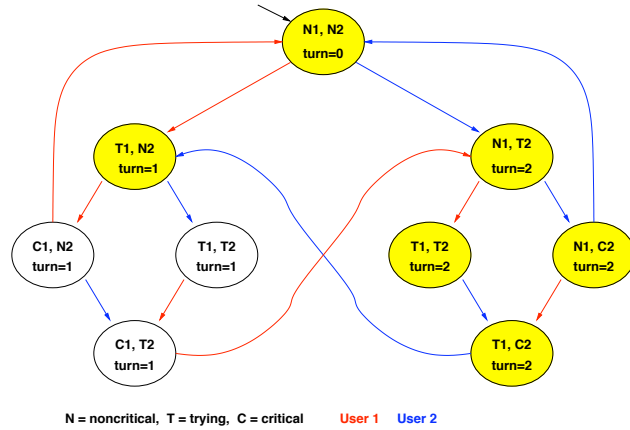
$$M \models \mathbf{AGAFC}_1 ? \implies M \models \neg \mathbf{EFEG}\neg C_1 ?$$

Example 1: fairness

$$[\mathbf{EG}\neg C_1], \text{ step 0:}$$


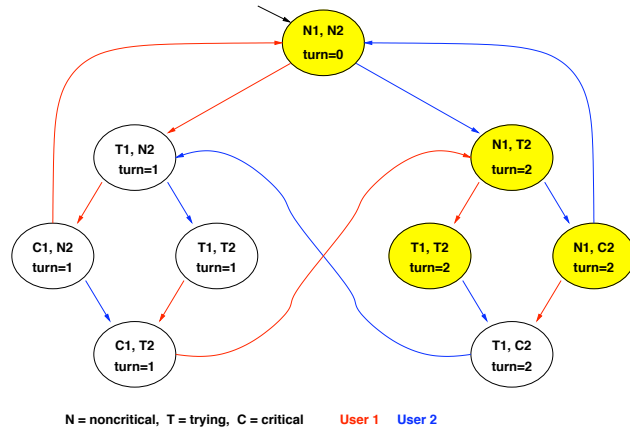
$$M \models \mathbf{AGAFC}_1 ? \implies M \models \neg \mathbf{EFEG}\neg C_1 ?$$

Example 1: fairness

[EG-C₁], step 1:

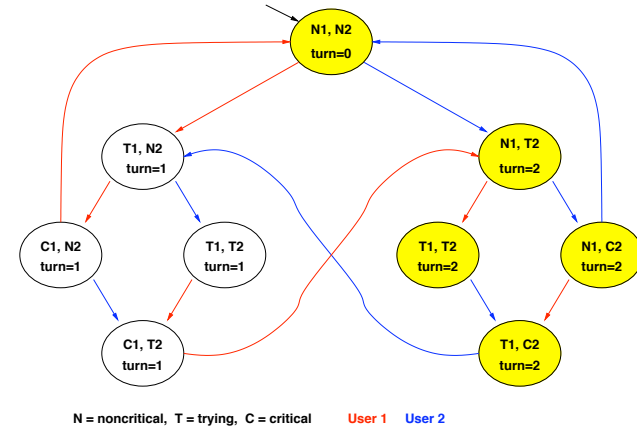
$$M \models \text{AGAF}C_1 ? \implies M \models \neg \text{EFEG}-C_1 ?$$

Example 1: fairness

[EG-C₁], step 3:

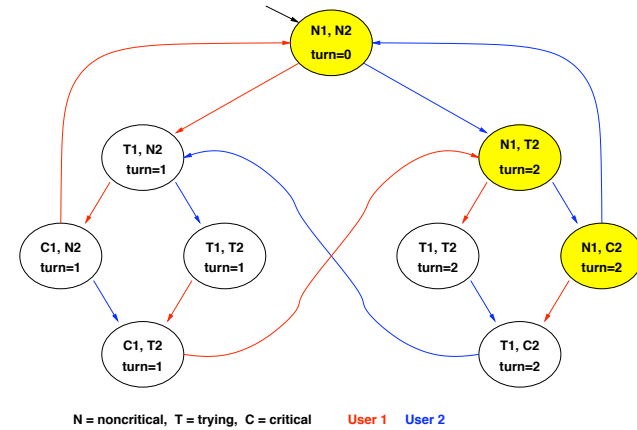
$$M \models \text{AGAF}C_1 ? \implies M \models \neg \text{EFEG}-C_1 ?$$

Example 1: fairness

[EG-C₁], step 2:

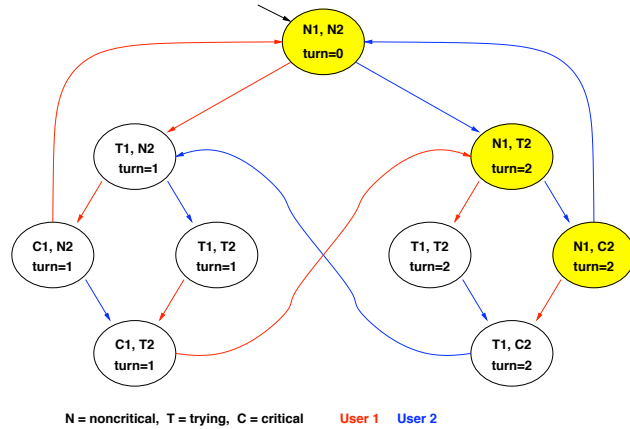
$$M \models \text{AGAF}C_1 ? \implies M \models \neg \text{EFEG}-C_1 ?$$

Example 1: fairness

[EG-C₁], step 4:

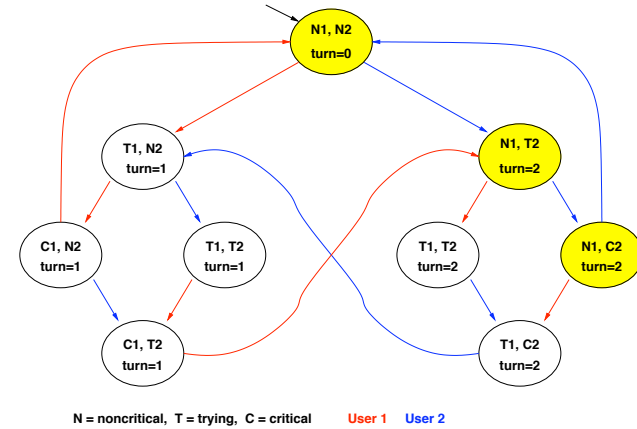
$$M \models \text{AGAF}C_1 ? \implies M \models \neg \text{EFEG}-C_1 ?$$

Example 1: fairness

 $[EG-C_1]$, FIXPOINT!

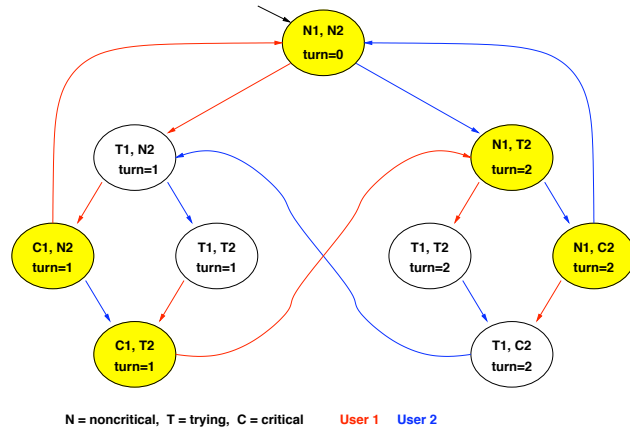
$$M \models \mathbf{AGAF}C_1 ? \implies M \models \neg \mathbf{EFEG} \neg C_1 ?$$

Example 1: fairness

 $[EFEG-C_1]$, STEP 0

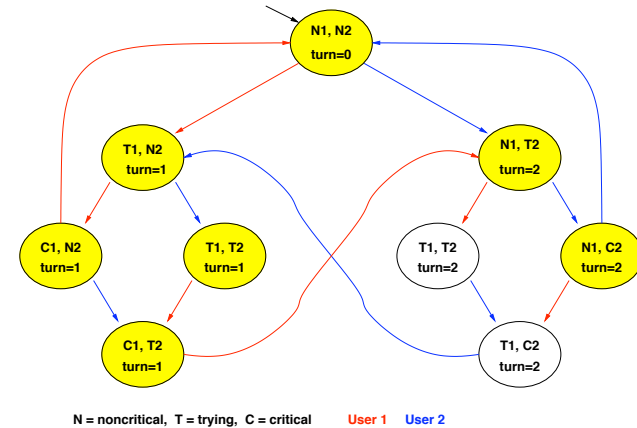
$$M \models \mathbf{AGAF}C_1 ? \implies M \models \neg \mathbf{EFEG} \neg C_1 ?$$

Example 1: fairness

 $[EFEG-C_1]$, STEP 1

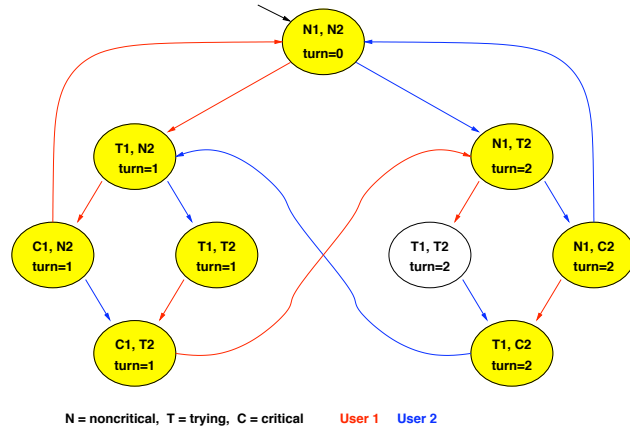
$$M \models \mathbf{AGAF}C_1 ? \implies M \models \neg \mathbf{EFEG} \neg C_1 ?$$

Example 1: fairness

 $[EFEG-C_1]$, STEP 2

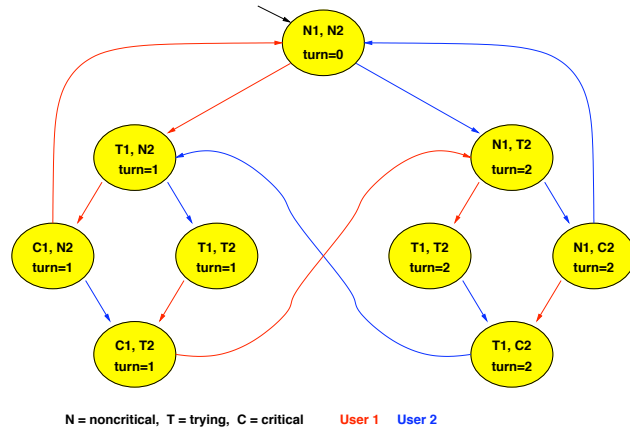
$$M \models \mathbf{AGAF}C_1 ? \implies M \models \neg \mathbf{EFEG} \neg C_1 ?$$

Example 1: fairness

[EFEG-C₁], STEP 3

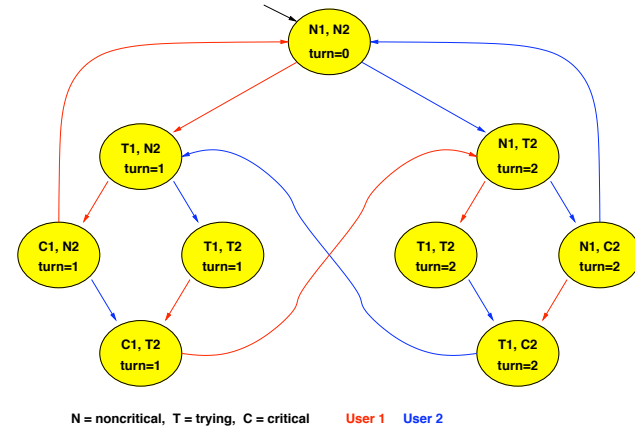
$$M \models \text{AGAF}C_1 ? \implies M \models \neg \text{EFEG}-C_1 ?$$

Example 1: fairness

[EFEG-C₁], FIXPOINT!

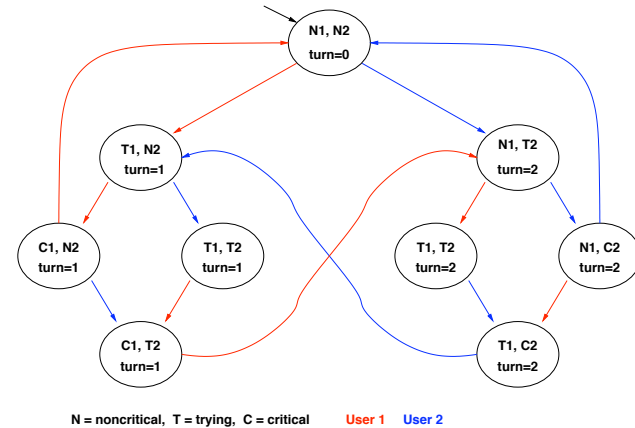
$$M \models \text{AGAF}C_1 ? \implies M \models \neg \text{EFEG}-C_1 ?$$

Example 1: fairness

[EFEG-C₁], STEP 4

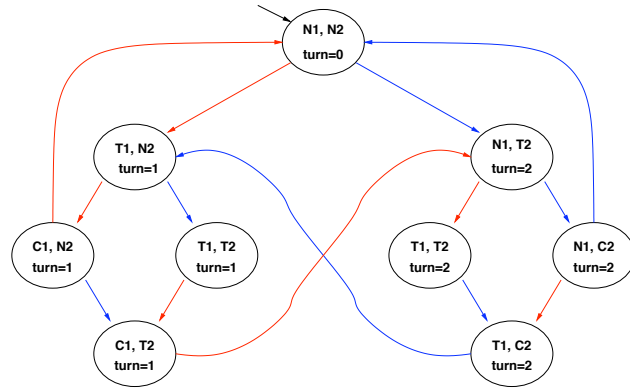
$$M \models \text{AGAF}C_1 ? \implies M \models \neg \text{EFEG}-C_1 ?$$

Example 1: fairness

[¬EFEG-C₁]

$$M \models \text{AGAF}C_1 ? \implies M \models \neg \text{EFEG}-C_1 ? \implies \text{NO!}$$

Example 2: liveness

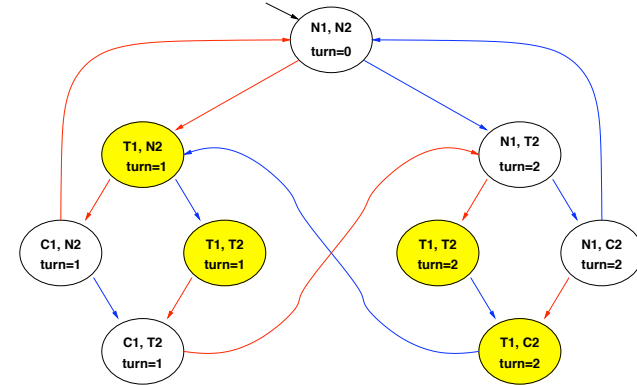


N = noncritical, T = trying, C = critical User 1 User 2

$$M \models \mathbf{AG}(T_1 \rightarrow \mathbf{AFC}_1) ? \implies M \models \neg \mathbf{EF}(T_1 \wedge \mathbf{EG} \neg C_1) ?$$

Example 2: liveness

$[T_1]$:

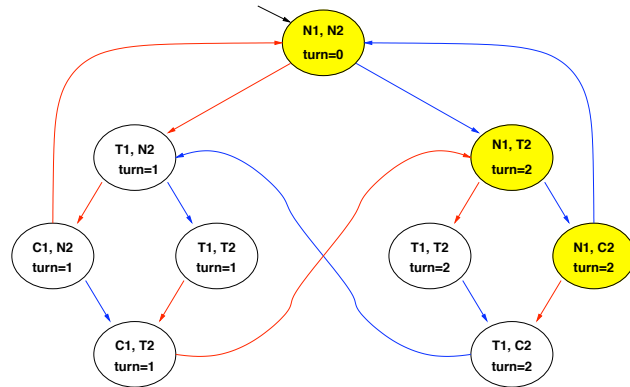


N = noncritical, T = trying, C = critical User 1 User 2

$$M \models \mathbf{AG}(T_1 \rightarrow \mathbf{AFC}_1) ? \implies M \models \neg \mathbf{EF}(T_1 \wedge \mathbf{EG} \neg C_1) ?$$

Example 2: liveness

$[\mathbf{EG} \neg C_1]$, STEPS 0-4: (see previous example)

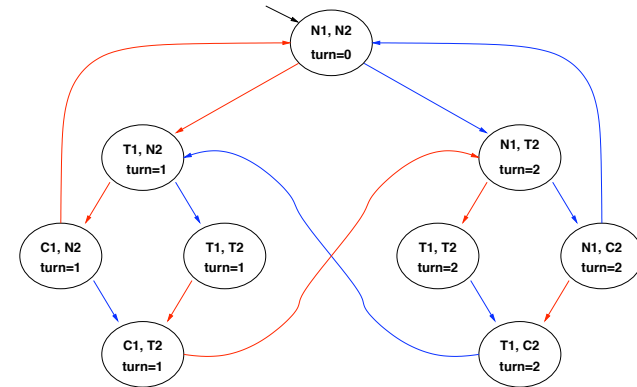


N = noncritical, T = trying, C = critical User 1 User 2

$$M \models \mathbf{AG}(T_1 \rightarrow \mathbf{AFC}_1) ? \implies M \models \neg \mathbf{EF}(T_1 \wedge \mathbf{EG} \neg C_1) ?$$

Example 2: liveness

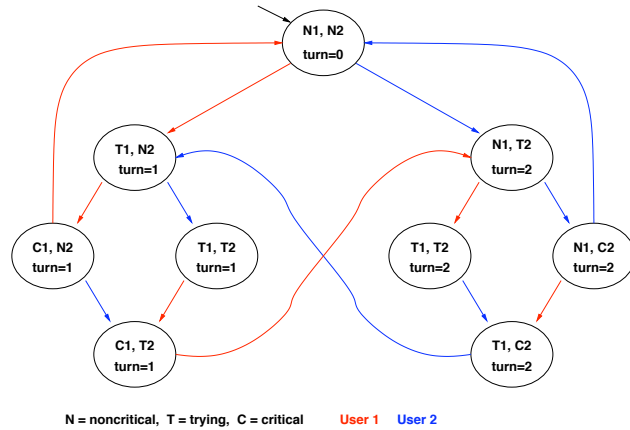
$[T_1 \wedge \mathbf{EG} \neg C_1]$:



N = noncritical, T = trying, C = critical User 1 User 2

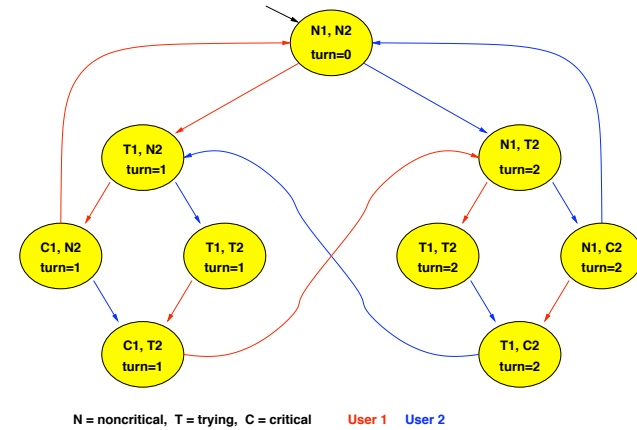
$$M \models \mathbf{AG}(T_1 \rightarrow \mathbf{AFC}_1) ? \implies M \models \neg \mathbf{EF}(T_1 \wedge \mathbf{EG} \neg C_1) ?$$

Example 2: liveness

$$[\mathbf{EF}(T_1 \wedge \mathbf{EG}\neg C_1)] :$$


$$M \models \mathbf{AG}(T_1 \rightarrow \mathbf{AFC}_1) ? \implies M \models \neg \mathbf{EF}(T_1 \wedge \mathbf{EG}\neg C_1) ?$$

Example 2: liveness

$$[\neg \mathbf{EF}(T_1 \wedge \mathbf{EG}\neg C_1)] :$$


$$M \models \mathbf{AG}(T_1 \rightarrow \mathbf{AFC}_1) ? \implies M \models \neg \mathbf{EF}(T_1 \wedge \mathbf{EG}\neg C_1) ? \text{ YES!}$$