Explicit-State
CTL Model Checking
CTL MC without Fairness
The Basic Idea

❖ The **CTL model checking question** to be answered: Given a Kripke structure $M = (S, S_0, R, L)$ over a set of atomic propositions $AP$ and a CTL formula $\varphi$ over $AP$, does $M \models \varphi$ hold?

❖ The **approach** to answer the CTL model checking question:
  - For every subformula $\psi$ of $\varphi$, compute the set $S_\psi = \{s \in S \mid M, s \models \psi\}$.
  - Compute $S_\psi$ starting from the inner-most subformulae (i.e. the most nested ones) going to the outer ones exploiting the already computed sets.
  - Check that $S_0 \subseteq S_\varphi$. 
**The Basic Idea**

- The **CTL model checking question** to be answered: Given a Kripke structure $M = (S, S_0, R, L)$ over a set of atomic propositions $AP$ and a CTL formula $\varphi$ over $AP$, does $M \models \varphi$ hold?

- The **approach** to answer the CTL model checking question:
  - For every subformula $\psi$ of $\varphi$, compute the set $S_\psi = \{s \in S \mid M, s \models \psi\}$.
  - Compute $S_\psi$ starting from the inner-most subformulae (i.e. the most nested ones) going to the outer ones exploiting the already computed sets.
  - Check that $S_0 \subseteq S_\varphi$.

- It is enough to consider the **basic operators** of CTL, i.e. consider the syntax

  $$\varphi ::= p \mid \neg \varphi \mid \varphi \lor \varphi \mid EX \varphi \mid E[\varphi U \varphi] \mid EG \varphi$$

  for $p \in AP$. 

The Depth of Formulae

Let $\varphi$ be a CTL formula in the restricted syntax. The depth of $\varphi$ denoted as $d(\varphi)$ is defined inductively as follows:

- If $\varphi = p$ for some $p \in AP$, $d(\varphi) = 0$.
- If $\varphi = \neg \psi$, $d(\varphi) = 1 + d(\psi)$.
- If $\varphi = \psi_1 \lor \psi_2$, $d(\varphi) = 1 + \max(d(\psi_1), d(\psi_2))$.
- If $\varphi = EX\psi$, $d(\varphi) = 1 + d(\psi)$.
- If $\varphi = EG\psi$, $d(\varphi) = 1 + d(\psi)$.
- If $\varphi = E[\psi_1 U \psi_2]$, $d(\varphi) = 1 + \max(d(\psi_1), d(\psi_2))$. 
The Main Algorithm

❖ Let $SF(\varphi)$ denote all subformulae of $\varphi$.

❖ The main algorithm for model checking CTL formulae may be written as follows:

**Input:** A Kripke structure $M = (S, S_0, R, L)$, a CTL formula $\varphi$ in the restricted syntax

**Output:** YES if $M \models \varphi$, NO otherwise.

**Data:** $Label : S \rightarrow 2^{SF(\varphi)}$, $i \in \{1, \ldots, d(\varphi)\}$

**Method:**

\[
\begin{align*}
&\text{for all } s \in S \text{ do } Label(s) := L(s) \cap SF(\varphi) \\
&\text{for } i := 1 \text{ to } d(\varphi) \text{ do} \\
&\quad \text{for all } \psi \in SF(\varphi) \text{ such that } d(\psi) = i \text{ do} \\
&\qquad \text{Check}(\psi) \\
&\quad \text{// Label the states } s \in S \text{ that satisfy } \psi, \text{i.e. add } \psi \text{ to } Label(s) \text{ iff } M, s \models \psi. \\
&\text{return YES if } \forall s_0 \in S_0. \varphi \in Label(s_0); \text{return NO otherwise.}
\end{align*}
\]

❖ As we will see, the overall time complexity of the algorithm is in $O(|SF(\varphi)| \cdot (|S| + |R|))$. 

Introduction – p.6/19
Check(¬\(\varphi\)), Check(\(\varphi_1 \lor \varphi_2\))

Check(¬\(\varphi\))
for all \(s \in S\) such that \(\varphi \notin Label(s)\) do
\(Label(s) := Label(s) \cup \{\neg\varphi\}\)

Diagram: 
```
  p ---\(\neg\)--- p
     |     |
     |     v
     p ---\(\neg\)--- p
```
Check(¬ϕ), Check(ϕ₁ ∨ ϕ₂)

Check(¬ϕ)
  for all s ∈ S such that ϕ ∉ Label(s) do
  Label(s) := Label(s) ∪ {¬ϕ}

Check(ϕ₁ ∨ ϕ₂)
  for all s ∈ S such that ϕ₁ ∈ Label(s) or ϕ₂ ∈ Label(s) do
  Label(s) := Label(s) ∪ {ϕ₁ ∨ ϕ₂}
Check($EX\varphi$)

for all $s_2 \in S$ such that $\varphi \in Label(s_2)$ do
for all $s_1 \in S$ such that $R(s_1, s_2)$ do

$Label(s_1) := Label(s_1) \cup \{EX\varphi\}$
The idea:

- Label first states already labelled by $\varphi_2$.
- Look at *predecessors* of states labelled with $\varphi_1 \mathrel{U} \varphi_2$, and if they are labelled with $\varphi_1$, label them with $\varphi_1 \mathrel{U} \varphi_2$ as well.
Check($E[\varphi_1 U \varphi_2]$)

Check($E[\varphi_1 U \varphi_2]$)

\[ T := \{ s \in S \mid \varphi_2 \in \text{Label}(s) \} \]

for all $s \in T$ do

\[ \text{Label}(s) := \text{Label}(s) \cup \{ E[\varphi_1 U \varphi_2] \} \]

while $T \neq \emptyset$ do

Choose $s_2 \in T$

\[ T = T \setminus \{ s_2 \} \]

for all $s_1 \in S$ such that $R(s_1, s_2)$ do

if $\varphi_1 \in \text{Label}(s_1)$ and $E[\varphi_1 U \varphi_2] \not\in \text{Label}(s_1)$ then

\[ \text{Label}(s_1) := \text{Label}(s_1) \cup \{ E[\varphi_1 U \varphi_2] \} \]

\[ T = T \cup \{ s_1 \} \]
A strongly connected component (SCC) $C$ of a directed graph $G$ is a maximal subgraph of $G$ such that every node of $C$ is reachable from every other node of $C$ along a directed path fully contained in $C$.

An SCC $C$ is nontrivial iff either it has more than one node or it contains one node with a self-loop.

SCCs of a finite oriented graph $(V, E)$ can be computed using the Tarjan’s algorithm in time $O(|E| + |V|)$. 
Lemma 1. Let $M = (S, S_0, R, L)$ be a Kripke structure, $S' = \{s \in S \mid M, s \models \varphi\}$, and $R' = R \cap (S' \times S')$. For any $s \in S$, $M, s \models EG\varphi$ iff

1. $s \in S'$ and
2. there exists a path in the oriented graph $G' = (S', R')$ that leads from $s$ to some node $t$ in a nontrivial SCC $C$ of $G'$.

Proof. See Clarke et al. Model Checking.
Check($EG\varphi$)

$S' := \{ s \in S \mid M, s \models \varphi \}$
$R' := R \cap (S' \times S')$

Use Tarjan’s algorithm to compute $SCC = \{ C \mid C$ is a nontrivial SCC of $(S', R') \}$

$T := \bigcup_{C \in SCC} \{ s \mid C = (V, E) \land s \in V \}$

for all $s \in T$ do $Label(s) := Label(s) \cup \{ EG\varphi \}$

while $T \neq \emptyset$ do
    Choose $s_2 \in T$
    $T = T \setminus \{ s_2 \}$
    for all $s_1 \in S'$ such that $R'(s_1, s_2)$ do
        if $EG\varphi \not\in Label(s_1)$ then
            $Label(s_1) := Label(s_1) \cup \{ EG\varphi \}$
            $T = T \cup \{ s_1 \}$
CTL MC: An Example

Let us check whether $\varphi = AG(Start \Rightarrow AF \text{Heat})$ holds in the following system.

$\varphi = \neg \psi$ where $\psi = EF(Start \land EG \neg \text{Heat}) = E[true \ U (Start \land EG \neg \text{Heat})]$. 

![Diagram of a system with states and transitions](image-url)
CTL MC: An Example

Let us check whether $\varphi = AG(Start \Rightarrow AF Heat)$ holds in the following system.

$\varphi = \neg \psi$ where $\psi = EF(Start \land EG \neg Heat) = E[true U (Start \land EG \neg Heat)]$
CTL MC: An Example

- Let us check whether $\varphi = AG(\text{Start} \Rightarrow AF \text{Heat})$ holds in the following system.
- $\varphi = \neg \psi$ where $\psi = EF(\text{Start} \land EG \neg \text{Heat}) = E[true \ U (\text{Start} \land EG \neg \text{Heat})]$
CTL MC: An Example

- Let us check whether $\varphi = AG(\text{Start} \Rightarrow AF \text{Heat})$ holds in the following system.
- $\varphi = \neg \psi$ where $\psi = EF(\text{Start} \land EG \neg \text{Heat}) = E[true U (\text{Start} \land EG \neg \text{Heat})]$. 

![Diagram of the system with states and transitions]

---

Introduction – p.14/19
CTL MC: An Example

- Let us check whether $\varphi = AG(Start \Rightarrow AF Heat)$ holds in the following system.
- $\varphi = \neg \psi$ where $\psi = EF(Start \land EG \neg Heat) = E[true \ U (Start \land EG \neg Heat)]$
CTL MC: An Example

- Let us check whether $\varphi = AG(\text{Start} \Rightarrow AF \text{Heat})$ holds in the following system.
- $\varphi = \neg \psi$ where $\psi = EF(\text{Start} \land EG \neg \text{Heat}) = E[true U (\text{Start} \land EG \neg \text{Heat})]$

- So, $\varphi$ does not hold in the given system.
CTL MC with Fairness
Let $M = (S, S_0, R, L)$ be a Kripke structure. Let us assume that fairness assumptions are encoded as a set $F \subseteq 2^S$. $F$ can be made a part of the definition of Kripke structures, leading to the so-called fair Kripke structures.

Let $\pi$ be a path in $M$. $\text{inf}(\pi) = \{s \in S \mid s \text{ appears infinitely many times in } \pi\}$.

A path $\pi$ is fair wrt. $F$ iff for every $P \in F$, $\text{inf}(\pi) \cap P \neq \emptyset$.

We write $M, s \models_F \varphi$ to indicate that the state formula $\varphi$ is true in $s \in S$ wrt. the fairness assumptions $F$.

Similarly, for a fair path $\pi$ and a path formula $\psi$, we introduce the notation $M, \pi \models_F \psi$ wrt. the fairness assumptions $F$.

The above allows one to handle weak fairness.

Indeed, the assumption that if an action is infinitely often enabled, it is infinitely often taken, is equal to the assumption that the action is either infinitely often disabled or taken.
The fair semantics of CTL* is almost the same as the basic one, up to the following changes:

- $M, s \models_F p$ for $p \in AP$ holds iff there exists a fair path starting from $s$ and $p \in L(s)$.
- $M, s \models_F E \psi$ for a path formula $\psi$ holds iff there exists a fair path $\pi$ starting from $s$ such that $M, \pi \models_F \psi$.
- $M, s \models_F A \psi$ for a path formula $\psi$ holds iff for all fair paths $\pi$ starting from $s$, $M, \pi \models_F \psi$.

An example of a fairness assumption could be a set of states that satisfy the formula $\neg Send \lor Receive$. 
CheckFair\((F, EG\varphi)\)

- We say that an SCC \(C = (V, E)\) is fair wrt. \(F\) iff for each \(P \in F\), \(V \cap P \neq \emptyset\).

- Lemma 2. Let \(M = (S, S_0, R, L)\) be a Kripke structure, \(F \subseteq 2^S\) a set of fairness assumptions, \(S' = \{s \in S \mid M, s \models F \varphi\}\), and \(R' = R \cap (S' \times S')\). For any \(s \in S\), \(M, s \models F EG\varphi\) iff
  1. \(s \in S'\) and
  2. there exists a path in the oriented graph \(G' = (S', R')\) that leads from \(s\) to some node \(t\) in a nontrivial fair SCC \(C\) of \(G'\).

Proof. Analogical to Lemma 1.\(\square\)
\textbf{CheckFair}(F,EG\varphi)

\begin{itemize}
  \item We say that an SCC \( C = (V, E) \) is fair wrt. \( F \) iff for each \( P \in F, V \cap P \neq \emptyset \).
  
  \item Lemma 2. Let \( M = (S, S_0, R, L) \) be a Kripke structure, \( F \subseteq 2^S \) a set of fairness assumptions, \( S' = \{ s \in S \mid M, s \models F \varphi \} \), and \( R' = R \cap (S' \times S') \). For any \( s \in S \), \( M, s \models_F EG\varphi \) iff
    \begin{enumerate}
      \item \( s \in S' \) and
      \item there exists a path in the oriented graph \( G' = (S', R') \) that leads from \( s \) to some node \( t \) in a nontrivial fair SCC \( C \) of \( G' \).
    \end{enumerate}
\end{itemize}

\textit{Proof}. Analogical to Lemma 1. \hfill \Box

\begin{itemize}
  \item The procedure \textbf{CheckFair}(F,EG\varphi) is identical to \textbf{Check}(EG\varphi) up to dealing with fair SCCs.

  \item The complexity of identifying fair SCCs in a graph \( G = (V, E) \) is a bit higher, namely \( O((|V| + |E|).|F|) \), as one has to check each SCC to see whether it has a vertex from each fairness assumption.
\end{itemize}
The Rest of Fair CTL MC

❖ Let $fair_F$ be an additional atomic proposition that is true in a state iff some fair computation path starts there. Such states can be labelled by $\text{CheckFair}(F, EG\text{ true})$.

❖ In order to label states $s$ s.t. $M, s \models_F p$ for $p \in AP$, $\text{Check}(p \land fair_F)$ can be used.

❖ In order to label states $s$ s.t. $M, s \models_F EX\varphi$, $\text{Check}(EX(\varphi \land fair_F))$ can be used assuming that states of $M$ have already been labelled wrt. $M, s \models_F \varphi$.

❖ In order to label states $s$ s.t. $M, s \models_F E[\varphi_1 U \varphi_2]$, $\text{Check}(E[\varphi_1 U (\varphi_2 \land fair_F)])$ can be used provided that states of $M$ have already been labelled wrt. $M, s \models_F \varphi_1, M, s \models_F \varphi_2$.

❖ The overall time complexity will be in $O(|SF(\varphi)|.(|S| + |R|).|F|)$.

❖ An example: Check that $AG(\text{Start} \Rightarrow AF Heat)$ holds in the microwave example under the assumption $F = \{P\}$ where $P = \{s \mid s \models \text{Start} \land \text{Closed} \land \neg \text{Error}\}$.
  • It holds as there is no nontrivial fair SCC satisfying $EG\neg\text{Heat}$.