Introduction to Software Verification

Orna Grumberg

Lectures Material
winter 2018-19
Lecture 5
Model of a system
Kripke structure / transition system

Labeled by atomic propositions AP
(critical section, variable value...)

Reactive Systems
Kripke Structure \( M=(S,R,L,S_0) \)

Given \( AP \) - finite set of atomic propositions

- \( S \) - (finite) set of states
- \( R \subseteq S \times S \) - total transition relation
  - For every \( s \in S \) there exists \( s' \in S \) such that \((s,s') \in R\).
  - Totality means that every path is infinite
- \( L:S \to 2^{AP} \) - labeling function that associates every state with the atomic propositions true in that state
- \( S_0 \subseteq S \) - set of initial states (optional)
**CTL***

**State formulas:**
- $p \in \text{AP}$
- $\neg g_1, g_1 \lor g_2, g_1 \land g_2$ where $g_1, g_2$ are state formulas
- $Ef, Af$ where $f$ is a path formula

**Path formulas:**
- Every state formula $g$ is a path formula
- $\neg f_1, f_1 \lor f_2, f_1 \land f_2, Xf_1, Gf_1, Ff_1, f_1Uf_2$ where $f_1, f_2$ are path formulas

**CTL*** - set of all state formulas
Semantics of CTL*

\( \pi = s_0, s_1, \ldots \) is a path in \( M \) if \( R(s_i, s_{i+1}) \) for every \( i \).
\( \pi^i \) - the suffix of \( \pi \) starting at \( s_i \).

State formulas:

- \( M, s \models p \iff p \in L(s) \)
- \( M, s \models Ef \iff \) there is a path \( \pi \) from \( s \) s.t. \( M, \pi \models f \)
- \( M, s \models Af \iff \) for every path \( \pi \) from \( s \), \( M, \pi \models f \)
Semantics of CTL*

π = s₀,s₁,... is a path in M if R(sᵢ,sᵢ₊₁) for every i.
πᵢ - the suffix of π starting at sᵢ.

Path formulas:
• M, π ⊨ g, where g is a state formula ⇔ M, s₀ ⊨ g
Semantics of CTL*

\( \pi = s_0, s_1, \ldots \) is a path in \( M \) if \( R(s_i, s_{i+1}) \) for every \( i \).

\( \pi^i \) - the suffix of \( \pi \) starting at \( s_i \).

Path formulas:

- \( M, \pi \models g \), where \( g \) is a state formula \( \iff M, s_0 \models g \)
- \( M, \pi \models Xf \iff M, \pi^1 \models f \)

\[ \circ \rightarrow \circ \rightarrow \circ \rightarrow \circ \quad \circ \circ \circ \models Xf \]

\[ \circ \rightarrow \circ \rightarrow \circ \rightarrow \circ \quad \circ \circ \circ \models f \]
Semantics of CTL*

\( \pi = s_0, s_1, \ldots \) is a path in \( M \) if \( R(s_i, s_{i+1}) \) for every \( i \).

\( \pi^i \) - the suffix of \( \pi \) starting at \( s_i \).

\[ M, \pi \models g \Longleftrightarrow M, s_0 \models g \]

\[ M, \pi \models Xf \Longleftrightarrow M, \pi^1 \models f \]

\[ M, \pi \models Gf \Longleftrightarrow \text{for every } k \geq 0, M, \pi^k \models f \]

\[ M, \pi \models Ff \Longleftrightarrow \text{there exists } k \geq 0, \text{s.t. } M, \pi^k \models f \]

\[ M, \pi \models f_1 U f_2 \Longleftrightarrow \text{there exists } k \geq 0, \text{s.t. } M, \pi^k \models f_2 \]

\[ \text{and for every } 0 \leq j < k, M, \pi^j \models f_1 \]
Semantics of CTL*

π = s₀,s₁,... is a path in M if R(sᵢ,sᵢ₊₁) for every i.
πᵢ - the suffix of π starting at sᵢ.

Path formulas:
• M, π ⊨ g, where g is a state formula ⇔ M, s₀ ⊨ g
• M, π ⊨ Xf ⇔ M, π¹ ⊨ f
• M, π ⊨ Gf ⇔ for every k ≥ 0, M, πᵏ ⊨ f
• M, π ⊨ Ff ⇔ there exists k ≥ 0, s.t. M, πᵏ ⊨ f
• M, π ⊨ f₁ U f₂ ⇔ there exists k ≥ 0, s.t. M, πᵏ ⊨ f₂
  and for every 0 ≤ j<k, M, πᵢ ⊨ f₁
Semantics of CTL*

$M \models g \iff \text{for every initial state } s: M,s \models g$
Examples – from last week

For \( p \in AP \), when does \( \pi \) (or \( s \)) satisfy the formula:

- \( \pi \models (Fb) U a \)

- \( \pi \models (Ga) U (Gb) \)
  equivalent to \( Gb \lor (Ga \land FGb) \)
Examples

For $p \in AP$, when does $\pi$ (or $s$) satisfy the formula:

- $\pi \models GF\ p$
- $\pi \models FG\ p$
- $s \models EGF\ p$
- $s \models EG\ EF\ p$
LTL

State formulas:
• $A f$ where $f$ is a path formula

Path formulas:
• $p \in AP$
• $\neg f_1, f_1 \lor f_2, f_1 \land f_2, X f_1, G f_1, F f_1, f_1 U f_2$ where $f_1, f_2$ are path formulas

LTL - set of all state formulas
CTL

CTL - set of all state formulas

- $p \in AP$
- $\neg g_1, \ g_1 \lor g_2, \ g_1 \land g_2$
- $AX \ g_1, \ AG \ g_1, \ AF \ g_1, \ A \ g_1 \ U \ g_2,$
- $EX \ g_1, \ EG \ g_1, \ EF \ g_1, \ E \ g_1 \ U \ g_2,$

where $g_1, g_2$ are state formulas
Semantics of CTL

Recall: path $\pi = s_0, s_1, ...$

- $M,s \models p \iff p \in L(s)$ for $p \in AP$

- $M,s \models \varphi_1 \lor \varphi_2 \iff M,s \models \varphi_1$ or $M,s \models \varphi_2$

- $M,s \models \text{EX} \varphi \iff$ there is $s'$ s.t. $R(s,s')$ and $M,s' \models \varphi$

- $M,s \models \text{EG} \varphi \iff$ there is a path $\pi$ from $s$, s.t. for every $i \geq 0$, $M,s_i \models \varphi$
Semantics of CTL

• $M, s \models E[\varphi_1 U \varphi_2] \iff$ there is a path $\pi$ from $s$ and there is $k \geq 0$ s.t. $M, s_k \models \varphi_2$ and for every $k > i \geq 0$, $M, s_i \models \varphi_1$

• $M, s \models AG \varphi \iff$ for every path $\pi$ from $s$ and for every $i \geq 0$, $M, s_i \models \varphi$

• $M, s \models AF \varphi \iff$ for every path $\pi$ from $s$ there exists $i \geq 0$ s.t. $M, s_i \models \varphi$
Examples (LTL)

1. $AG \neg (\text{start } \land \neg \text{ready})$
2. $AG (\text{req } \rightarrow \text{F ack})$
3. $A \text{GF-enabled}$
4. $A \text{FG deadlock}$
5. $A (\text{GF-enabled } \rightarrow \text{GF running})$

Cannot express existential properties: “from any state the system can...”
Examples (CTL)

1. EF (start $\land \neg \text{ready}$)
2. AG (req $\rightarrow$ AF ack)
3. AG (AF enabled)
4. AF (AG deadlock)
5. AG (EF restart)
6. AG (non_critical $\rightarrow$ EX tryring)
7. AG (try $\rightarrow$ A[try U succeed])
Equivalence

• **Path formulas** $\psi_1, \psi_2$ are equivalent if:
  For every $M$ and path $\pi$
  \[
  M, \pi \models \psi_1 \text{ iff } M, \pi \models \psi_2
  \]

• **State formulas** $\varphi_1, \varphi_2$ are equivalent if:
  For every $M$ and state $s$
  \[
  M, s \models \varphi_1 \text{ iff } M, s \models \varphi_2
  \]
Expressiveness

\neg, \lor, X, U, E suffice to express all CTL*:
• \( Ff \equiv \text{true U } f \)
• \( Gf \equiv \neg F (\neg f) \)
• \( Af \equiv \neg E (\neg f) \)

In CTL: \( EX, EG, EU \) are sufficient

• \( A \ [pUq] \equiv (\neg EG \neg q) \land \neg E[\neg q U (\neg p \land \neg q)] \)
LTL vs. CTL

- **A (FG p)** has no equivalent in CTL
  “in all paths, p globally holds from some point on”

- **Failed attempts:**
  
  **AFAGP**: “in every path there is a point from which all reachable states satisfy p.”

![Diagram](image)

All paths satisfy FGp
- s₀,s₀,s₀,...
- s₀,...s₀,s₁,s₂,s₂,s₂...

But first one does not sat FAGp
LTL and CTL vs. CTL*

• $E (GFp)$ has no equivalent in LTL or CTL
Theorem:

• The expressive powers of LTL and CTL are incomparable. That is,
  - There is an LTL formula that has no equivalent CTL formula
  - There is a CTL formula that has no equivalent LTL formula

• CTL* is more expressive than either of them
Explicit Model Checking for CTL
Model Checking \textit{CE81, QS82}

An efficient procedure that receives:
- A finite-state model describing a system
- A temporal logic formula describing a property

It returns
yes, if the system has the property
no + Counterexample, otherwise
CTL Model Checking $M \models f$

- **Goal:** For each $s$, computes $\text{label}(s)$, which is the set of subformulas of $f$, true in $s$

- The Model Checking algorithm works *iteratively* on subformulas of $f$, from *simpler* subformulas to more *complex* ones

- For checking $AG(\text{request } \Rightarrow \text{AF grant})$
  - Check $\text{grant}$, $\text{request}$
  - Then check $\text{AF grant}$
  - Next check $\text{request } \Rightarrow \text{AF grant}$
  - Finally check $AG(\text{request } \Rightarrow \text{AF grant})$
Model Checking $M \models f$ (cont.)

- We check subformula $g$ of $f$ only after all subformulas of $g$ have already been checked

- For subformula $g$, the algorithm adds $g$ to label(s) for every state $s$ that satisfies $g$

- When we finish checking $g$, the following holds:
  - $g \in \text{label}(s) \iff M,s \models g$
Model Checking $\mathcal{M} \models f$ (cont.)

Alternative description
Denote $S_g = \{ s \mid \mathcal{M}, s \models g \}$

- The goal of model checking is to compute $S_g$ for each subformula $g$ of $f$
  - In particular, $S_f$
Model Checking $M \models f$ (cont.)

- $M \models f$ if and only if $f \in \text{labels}(s)$ for all initial states $s$ of $M$

- $M \models f$ if and only if $S_0 \subseteq S_f$

- The algorithm has time complexity: $O(|M| \times |f|)$
Model Checking Atomic Propositions

- For atomic proposition $p \in \text{AP}$:
  $p \in \text{label}(s) \iff p \in L(s)$

  Held by alg  Defined by $M$

How do we handle more complex formulas?

Observation:
- Sufficient to handle $\neg$, $\lor$, EX, EU, EG
Model Checking \( \neg, \lor \) formulas

\( \neg f_1 \): add to label(s) if and only if \( f_1 \notin \text{labels}(s) \)

\( f_1 \lor f_2 \): add to label(s) if and only if 
\( f_1 \in \text{labels}(s) \) or \( f_2 \in \text{labels}(s) \)
Model Checking $g = \text{EX } f_1$

add $g$ to label(s) if and only if $s$ has a successor $t$ such that $f_1 \in \text{labels}(t)$

procedure $\text{CheckEX } (f_1)$

$T := \{ t \mid f_1 \in \text{label}(t) \}$

while $T \neq \emptyset$ do

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

for all $s$ s.t. $R(s,t)$ do

if $\text{EX } f_1 \notin \text{label}(s)$ then

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

end for all

end while