SLR 204: Basics of verification of distributed systems

Vadim Malvone
vadim.malvone@telecom-paris.fr
Formal verification of CTL properties
Introduction

- Algorithm by Clarke, Emerson, and Sistla.

- Principle:
  - It is based on a marking algorithm which takes a Kripke structure $M$, a formula $\phi$ of CTL, and consists in associating the states $s$ in $M$ that satisfies $\phi'$, for each sub-formula $\phi'$ of $\phi$.
  - Formally, we use the set $[\phi'] = \{s \in S \mid s \models \phi'\}$.
  - Finally, we can decide if $s \models \phi$ by consulting the different markings, i.e. $s \models \phi$ iff $s \in [\phi]$. 
A more convenient CTL: the Existential Normal Form (ENF)

- CTL is formed according to the grammar:
  \[
  \varphi := p \mid \neg \varphi \mid \varphi \lor \varphi \mid \text{EX} \varphi \mid E \varphi U \varphi \mid A \varphi U \varphi
  \]

  where \( p \in \text{AP} \)

- CTL in ENF is formed according to the grammar:
  \[
  \varphi := p \mid \neg \varphi \mid \varphi \lor \varphi \mid \text{EX} \varphi \mid E \varphi U \varphi \mid \text{EG} \varphi
  \]

  where \( p \in \text{AP} \)
Equivalence between the two languages

\[ \psi_1 \cup \psi_2 = \phi_1 \land \phi_2 \]

Where:
- \( \phi_1 = \neg (E (\neg \psi_2) \cup (\neg \psi_1 \land \neg \psi_2)) \);
- \( \phi_2 = \neg (EG (\neg \psi_2)) \).

Along any path: \( \psi_2 \) must hold eventually and \( (\neg \psi_1 \land \neg \psi_2) \) can only happen after \( \psi_2 \).

- \( \phi_1 \) means that \( \psi_1 \) cannot become false, while \( \psi_2 \) stays false!
- \( \phi_2 \) means that \( \psi_2 \) cannot remain false forever! (i.e. \( \psi_2 \) will eventually become true along any path).
The set of subformulas of $\varphi$

Function sub($\varphi$)

- if ($\varphi = p$) then return $p$;
- if ($\varphi = \neg \psi$) then return sub($\psi$) $\cup$ $\varphi$;
- if ($\varphi = \psi_1 \lor \psi_2$) then return sub($\psi_1$) $\cup$ sub($\psi_2$) $\cup$ $\varphi$;
- if ($\varphi = \mathbf{E} \psi$) then return sub($\psi$) $\cup$ $\varphi$;
- if ($\varphi = \mathbf{E} \psi_1 \mathbf{U} \psi_2$) then return sub($\psi_1$) $\cup$ sub($\psi_2$) $\cup$ $\varphi$;
- if ($\varphi = \mathbf{EG} \psi$) then return sub($\psi$) $\cup$ $\varphi$;

Complexity: $O(|\varphi|)$
The main procedure

Procedure Labelling $(M, \phi)$

for all $\phi' \in \text{sub}(\phi)$

switch($\phi'$)

  case $p$: \([\phi'] = \{s \in S \mid p \in \text{Lab}(s)\};
  case $\neg \psi$: \([\phi'] = S \setminus [\psi];
  case $\psi_1 \lor \psi_2$: \([\phi'] = [\psi_1] \cup [\psi_2];
  case EX $\psi$: EX$(M, \phi')$;
  case E $\psi_1 U \psi_2$: EU$(M, \phi')$;
  case EG $\psi$: EG$(M, \phi')$;
Case EX $\psi$ (I)
Case EX $\psi$ (II)

Procedure EX(M, $\varphi$)

for all $s \in S$

if $((s, s') \in R)$ and $(s' \in [\psi])$ then

$[\varphi] = [\varphi] \cup \{s\}$;

Complexity: $O(|M|)$
Case E $\psi_1 U \psi_2$ (I)

$\Psi_1$ $\subseteq$ $Q'$ $\subseteq$ $\mathcal{E}$

$\psi_1$ $\psi_2$
Case E $\psi_1 U \psi_2$ (II)

- Collect in a set $Q$ all the states satisfying $\psi_2$.
  - All these states also satisfy $E \psi_1 U \psi_2$.

- Traverse backward: from states in $Q$, we add in $[E \psi_1 U \psi_2]$ all the states $t$ satisfying $\psi_1$ and reaching at least a state $s$ in $[E \psi_1 U \psi_2]$.

- Formally, if $s \in Q$, $(t, s) \in R$ and $t \in [\psi_1]$ then $t \in [E \psi_1 U \psi_2]$.

- Recall that: $E \psi_1 U \psi_2 = (\psi_2 \lor (\psi_1 \land EX E \psi_1 U \psi_2 ))$. 
Case E $\psi_1 U \psi_2$ (III)
**Case E $\psi_1 U \psi_2$ (IV)**

Procedure EU($M, \varphi$)

1. $Q = \text{empty}$;
2. for all $s \in S$
   - if ($s \in [\psi_2]$) then
     - $[\varphi] = [\varphi] U \{s\}$;
     - $Q = Q U \{s\}$;
3. while $Q \neq \emptyset$
   - chose $s \in Q$;
   - $Q = Q \setminus \{s\}$;
   - for all $s' \in S$
     - if (($s', s) \in R$) and ($s' \notin [\varphi]$) and ($s' \in [\psi_1]$) then
       - $[\varphi] = [\varphi] U \{s'\}$;
       - $Q = Q U \{s'\}$;

Complexity: $O(|M|)$
Case EG $\psi$ (I)

- We start by using a new Kripke structure $M'$ with:
  - $S' = [\psi]$;
  - $R_{|S'xS'}$ (restriction of $R$ to $S'$);
  - $L_{|S'xS'}$ (restriction of $L$ to $S'$).

- To solve the operator EG we need to find the strongly connected components in $M'$.

- A directed graph is called strongly connected if there is a path in each direction between each pair of vertices of the graph.

- There are several algorithms based on depth first search (DFS) that compute strongly connected components in linear time.
Case $\text{EG } \psi$ (II)
Case EG $\psi$ (III)

Procedure EG(M, $\varphi$)

\[
G = \text{SCC}[\psi]; \\
[\varphi] = \bigcup_{C \in G} \{s \in S \mid s \in C \}; \\
Q = \bigcup_{C \in G} \{s \in S \mid s \in C \};
\]

while $Q \neq \emptyset$

chose $s \in Q$;

$Q = Q \setminus \{s\}$;

for all $s' \in S'$

if $((s', s) \in R)$ and $(s' \notin [\varphi])$ then

$[\varphi] = [\varphi] \cup \{s'\}$;

$Q = Q \cup \{s'\}$;

Complexity: $O(|M|)$
The algorithm presented shows that the model checking problem for CTL can be solved in linear time in the size of the model $M$ and the size of the property $\varphi$, that is:

$$\text{in time } O(|M| \cdot |\varphi|)$$

where $|M|$ is the size of the model $M$ and $|\varphi|$ is the number of sub-formulas of $\varphi$. 