Effective Verification of Systems with a Dynamic Number of Components

P. Vařeková, P. Moravec, I. Černá, and B. Zimmerova

Faculty of Informatics
Masaryk University, Brno

SAVCBS’07
Contents

1 Dynamic systems

2 Properties
   - Dynamic system properties in general
   - Properties we are interested in

3 Verification
$S$ - Dynamic system

$S_n$ - Dynamic system with $n$ clients deployed
Dynamic system – Definition
Dynamic system — Definition
Dynamic system — Definition
Dynamic system — Definition
Dynamic system – Definition

- Client t1
- Client t1
- Client t1
- Client t1
- Provider

Effective Verification of Dynamic systems

P. Vařeková, P. Moravec, I. Černá, and B. Zimmerova
We use Component Interaction automata

A hierarchy of component names: (α)

Can be modelled by • Finite transitions systems or • Regular-like expressions
Contents

1 Systems with a Dynamic Number of Components

2 Properties
   • Dynamic system properties in general
   • Properties we are interested in

3 Verification
Properties — Example

"If a client of the system sends a request, then he will receive a response."
"If a client of the system sends a request, then he will receive a response."

∀Sn (n ∈ ℤ₀):
"If client i ∈ {1,...,n} sends a request, then he will receive a response."
"If a client of the system sends a request, then he will receive a response."

\[ \forall S_n \ (n \in \mathbb{N}_0) : \]
"If client \( i \in \{1, \ldots, n\} \) sends a request, then he will receive a response."

\[ \forall S_n \ (n \in \mathbb{N}_0) : \]
\[ \varphi_n = \bigwedge_{i \in \{1, \ldots, n\}} G(\mathcal{P}(i, \text{request}, \alpha) \Rightarrow F \mathcal{P}(\alpha, \text{response}, i)) \]
"If a client of the system sends a request, then he will receive a response."

\[ \forall S_n \ (n \in \mathbb{N}_0) : \]
\[ "If \ client \ i \in \{1, \ldots, n\} \ sends \ a \ request, \ then \ he \ will \ receive \ a \ response." \]

\[ \forall S_n \ (n \in \mathbb{N}_0) : \]
\[ \varphi_n = \bigwedge_{i \in \{1, \ldots, n\}} G(\mathcal{P}(i, \text{request}, \alpha) \implies F \mathcal{P}(\alpha, \text{response}, i)) \]

\[ \forall n \in \mathbb{N}_0 : S_n \models \varphi_n \]
Properties — Introduction

- Property: \( \{ \varphi_i \}_{i \in \mathbb{N}_0} \)

- Property is satisfied \( \iff \forall n \in \mathbb{N}_0 : S_n \models \varphi_n \)

- We use
  - \( \varphi_i \) - temporal logic CI–LTL
  - CI–LTL - an extension of action based LTL
    - \( \mathcal{P}(l) \) - \( l \) is performed as the first action of the path
    - \( \mathcal{E}(l) \) - \( l \) is enabled in the first state of the path
Properties — Main restriction

Restrictions

- no distinctions among clients,

- properties whose violation involves only a finite number of observed components

\[ \text{Property}(S, m) \]
- no distinction among clients
- violation involves \( m \) observed components
Example 1/3:

- "If a client of the system sends a request, then he will receive a response."

- path $\pi$ violates it $\Rightarrow$ a client "send a request and does not receive a response"

- we can observe only this client, to show that this property is violated in $\pi$

- we need to observe 1 client.

- $\in \text{Property}(S, 1)$
Example 2/3:

”Two clients can not be able to receive a response at the same time.”

path $\pi$ violates it $\Rightarrow$ clients $j_1$ and $j_2$ ”can receive a response at the same time”

we can observe only clients $j_1$ and $j_2$, to show that this property is violated in $\pi$

we need to observe 2 clients.

$\in \text{Property}(S, 2)$
Example 3/3:

- "The system does not contain a deadlock."

- path $\pi$ violates it $\Rightarrow$ all clients and provider reach the state from which they can not continue

- we must observe all clients, to show that this property is violated in $\pi$

- we need to observe $n$ clients in $S_n$.

- $\not\in \text{Property}(S, m)$ for any $m \in \mathbb{N}_0$
If a component **tries to emit** an event on its required interface, the counterpart is **able to absorb** it.

Interface automata, SOFA

System does not contain a **deadlock**.

FOCUS, JavaA, rCOS, SOFA

Situation when communication of components in the group never finished is unreachable.

SOFA

A state in which more than half of clients are in a critical section is unreachable.
Contents

1 Systems with a Dynamic Number of Components

2 Properties
   - Dynamic system properties in general
   - Properties we are interested in

3 Verification
Verification problem

Input: $S$, 
\[ \{ \varphi_i \}_{i \in \mathbb{N}_0} \in \text{Property}(S, m) \text{ for some } m \in \mathbb{N}_0 \]

Question: $\forall i \in \mathbb{N}_0 : S_i \models \varphi_i$?

Verification of infinitely many finite state transition systems.

Our solution

find $k \in \mathbb{N}_0$ such that if $S_0 \models \varphi_0$,  
$S_1 \models \varphi_1$,  
\vdots  
$S_k \models \varphi_k$,  
then $\forall n \in \mathbb{N}_0 : S_n \models \varphi_n$.

Verification of finitely many finite state transition systems.
Input \( S, \{\varphi_i\}_{i \in \mathbb{N}_0}, m: \{\varphi_i\}_{i \in \mathbb{N}_0} \in \text{Property}(S, m) \).

Intermediate data
- \( X \) - set containing all labels necessary for verification of \( \{\varphi_i\}_{i \in \mathbb{N}_0} \)
- \( |D|_X \in \mathbb{N}_0 \cup \{\infty\} \)

Output
\( k = |D|_X + m \in \mathbb{N}_0 \cup \{\infty\} \)
Verification — Problem

Input

\[ S, \]
\[ \{ \varphi_i \}_i \in \mathbb{N}_0, \]
\[ m: \{ \varphi_i \}_i \in \mathbb{N}_0 \in \text{Property}(S, m). \]

Intermediate data

- \( X \) - set containing all labels necessary for verification of \( \{ \varphi_i \}_i \in \mathbb{N}_0 \)
- \( |D|_X \in \mathbb{N}_0 \cup \{ \infty \} \)

Output

\[ k = |D|_X + m \in \mathbb{N}_0 \cup \{ \infty \} \]
Conclusions

- Dynamic systems
- Properties
- Properties whose violation involves finite number of clients
- Verification
Conclusions

Thank you for your attention.