An Improved POCTL Model Checking Algorithm Based on Preprocess Mechanism

Jian-xi PENG¹,a and Jian-ping LIU¹,b
¹foshan Polytechnic, Foshan, Guangdong 528137, China
²nhpjx@163.com, bjp68@tom.com

Keywords: PoCTL, Checking model, Preprocess mechanism, Marker algorithm, PM_LA

Abstract. In order to improve the defects of high time complexity and low efficiency in PoCTL model detecting, according to traditional model checking marker algorithm, an improved PoCTL model checking algorithm based on preprocess mechanism which could be applied in large scale and high complexity is proposed in this paper, e.i PM_LA algorithm. Uniqueness of public sub expression is preprocess marked through relative PoCTL formula firstly. Public sub express and PoCTL model status are assigned in balance status of keeping model detecting space secondly. And finally, validation is proceeded to guarantee that PoCTL formula validation being first-passed with very high probability. The simulation shows that PM_LA could reduce time complexity and improve validation performance.

Introduction

Model checking is a new automatic and intelligent computer software and hardware device validating method with many advantages, e.g., high efficiency, precision and applicability. It has been widely used in software and hardware integrated development, secure communication protocol design and other fields.

A lots model checking research have been done, as reference [1], detecting marker algorithm based on probability checking computing tree logic (PoCTL)is proposed. The algorithm could effectively detect whether it’s finite-state concurrent system could meets PoCTL formula specification and have formal description for PoCTL formula match degree. It makes a preparation for intelligent checking mechanism of concurrent system. Reference [2] has a deep analysis for the scale and complexity of PoCTL formula, and prove that the algorithm has low efficiency. An improved PoCTL formula marker algorithm is proposed in reference [3] based the research of reference [2]. The new algorithm has a pre-model evaluation mechanism to public sub expression which could reduce the redundancy verification. So validation rate stay in a high level. But the cost of pre-model evaluation mechanism is much higher which make whole system performance stay in low efficiency that could not be used widely. Beside these research, some analysis of symbol marker, abstract formalizing, combinatorial math have be done in reference [4]-[7], i.e., state space explosion [8]-[10].

Lots research show that most work is focused in unique characteristic of hardware and software, i.e., a model checking could be validated only one time [11]. With rapidly development of software and hardware, the scale of application is becoming larger and larger, so the traditional uniqueness checking could not be met with this situation. Now, some model checking mechanisms use divide-and-process strategy for large scale [12]. But these mechanisms is low efficient and could not meet the requirement of validation performance [13]. For the defects of high time complexity and low efficiency, according to traditional PoCTL model checking mark algorithm
basic, an improved PoCTL model checking algorithm based on preprocess mechanism which could be applied in large scale and high complexity is proposed in this paper, i.e, PM_LA algorithm.

Poctl Algorithm

With the premise of characteristics of unique system structure model [14] in model checking mechanism, software and hardware system structure model is described, in which, M present migration system, S present state set of checking, C present a formula set of PoCTL, and
\[ C = \{ \varphi_1, \varphi_2, \ldots, \varphi_m \} (m \in \mathbb{Z}) \].

So, model checking mechanism is to validate whether M could match one PoCTL formula of C, i.e,
\[ M, s \models \varphi | (\varphi \in C, s \in S) \].

Normally, PoCTL model checking mechanism would not validate every public sub expression. So, the algorithm is improved in this paper.

Mark Algorithm

As traditional process mechanism of validation computing tree logic formula, principle of mark algorithm is to choose a matched correlated combination \[ \{ \neg, \land, \bot, A, F, EU, EX \} \] to have formal description and replace one PoCTL formula \[ \varphi \].

And \[ \varphi \] serialization of sub expression is match detected gradually. The matched sub expression is used to mark information of migration system M until \[ \varphi \] serialization of sub expression match checking is finished. The marked M state sets are processed to validate whether the initial state of M could match one of relevant condition of PoCTL formula finally. The descriptions are as follow.

Description 1: Sub expression of a PoCTL formula \[ \varphi \] is replaced by \[ \psi \], in which, \[ \psi \subseteq \varphi \].

Description 2: A specific parse trees logic is generated by grammar structure of PoCTL formula \[ \varphi \]. Leaf note is sub expression of \[ \varphi \]. Besides that, it is one of correlated combination \[ \{ \neg, \land, \bot, A, F, EU, EX \} \].

Description 3: Assuming there is a same sub expression \[ \psi \] in many PoCTL formula \[ \varphi \], so \[ \psi \] is the public sub expression of these PoCTL formula \[ \varphi \].

Description 4: Setting \[ \{ \neg, \land, \bot, A, F, EU, EX \} \] is n dimension correlated word of correlated combination \[ \{ \neg, \land, \bot, A, F, EU, EX \} \].

Description 5: Setting \[ \psi_1, \psi_2 \] and \[ \varphi \] are known PoCTL formulas, if there is a condition as equation 1, \[ \psi_1 \] and \[ \psi_2 \] could be considered as the direct sub expression of \[ \varphi \].

Redundancy Validation of Public Sub Expression

**Corollary 1.** Given PoCTL formula \[ \psi \] and \[ \varphi \], in which, \[ \psi \] is the sub expression of \[ \varphi \]. So \[ \psi \subseteq \varphi \], the validation result \[ M, s \models \varphi | (\varphi \in C, s \in S) \] of model checking marking algorithm must meet \[ M, s \models \varphi | (\psi \subseteq \varphi \in C, s \in S) \]. The proving process is that, if \[ \psi \subseteq \varphi \], \[ \varphi = \theta_1 op_1 \theta_2 op_2 \ldots \theta_n op_n (\psi) \ldots \] and
\( \theta_k, \omega_k (0 < k \leq i) \) are PoCTL formula and correlated word in turn. Assuming there is a formula meet the condition \( \delta = \theta_1 \omega_1 \), so \( \delta \subset \varphi \), and match the correlated condition of description 5.

As the proving process turns out, pre-validated \( M, s \models \delta(\delta \in C, s \in S) \) must meet \( M, s \models \varphi(\varphi \in C, s \in S) \). Assuming \( \text{SAT}(\varphi) \) is the computing of the process of \( M, s \models \varphi(\varphi \in C, s \in S) \), the obtained output value is the set which match \( \varphi \). So PoCTL formula mark algorithm need to launch validation from the direct public sub expression of description 5. \( \delta \) has certain association with \( \varphi \) like descriptions as below.

\[
\begin{align*}
(1) \quad & \delta = \neg \psi \quad \text{SAT}(\delta) = S - \text{SAT}(\psi) \\
(2) \quad & \delta = \psi \land \psi_1 \quad \text{SAT}(\delta) = \text{SAT}(\psi) \land \text{SAT}(\psi_1) \\
(3) \quad & \delta = \psi \lor \psi_1 \quad \text{SAT}(\delta) = \text{SAT}(\psi) \lor \text{SAT}(\psi_1) \\
(4) \quad & \delta = \psi \rightarrow \psi_1 \\
\text{SAT}(\delta) &= \text{SAT}(\neg \psi \lor \psi_1) = \text{SAT}(\neg \psi) \lor \text{SAT}(\psi_1) = (S - \text{SAT}(\psi)) \lor \text{SAT}(\psi_1) \\
(5) \quad & \delta = \text{AX} \psi \quad \text{SAT}(\delta) = \text{SAT}(\neg \text{EX} - \psi) = \text{SAT}(\text{EX} \neg \psi) = S - \text{SAT}(\text{EX} (S - \text{SAT}(\psi))) \\
(6) \quad & \delta = \text{EX} \psi \quad \text{SAT}(\delta) = \text{SAT}(\text{EX}(\psi)) \\
(7) \quad & \delta = \mathcal{A}[\psi \cup \psi_1] \\
\text{SAT}(\delta) &= \text{SAT}(-E[\psi \cup \psi_1] (\neg \psi \land \neg \psi_1)) \lor \text{EG} \neg \psi_1) \\
&= (S - \text{SAT}(\neg \psi, (\neg \psi \land \neg \psi_1))) \lor (S - \text{SAT}(\neg \psi_1)) \\
(8) \quad & \delta = E[\psi U \psi_1] \quad \text{SAT}(\delta) = \text{SAT}(\psi, \psi_1) \\
(9) \quad & \delta = \text{EF} \psi \quad \text{SAT}(\delta) = \text{SAT}(T, \psi) \\
(10) \quad & \delta = \text{EG} \psi \quad \text{SAT}(\delta) = \text{SAT}(\text{AF} \neg \psi) \\
(11) \quad & \delta = \text{AF} \psi \quad \text{SAT}(\delta) = \text{SAT}(\psi) \\
(12) \quad & \delta = \text{AG} \psi \quad \text{SAT}(\delta) = \text{SAT}(\text{EF} \neg \psi) = S - \text{SAT}(\text{EF} \neg \psi) = S - \text{SAT}(T, \neg \psi)
\end{align*}
\]

According to formula (1) to (4), \( \text{SAT}(\delta) \) output is obtained after \( \text{SAT}(\varphi) \) have been processed. Based on formal equivalence mechanism of PoCTL formula mark algorithm, the last 8 formulas are formal expressed by correlated combination \( \{\neg, \land, \lor, \text{AF}, \text{EU}, \text{EX}\} \). As the same mechanism, \( \text{SAT}(\delta) \) output is obtained after \( \text{SAT}(\varphi) \) have been processed. The last result is: if \( \varphi \subset \delta \), \( M, s \models \varphi(\varphi \in C, s \in S) \) is validated before the validation of \( M, s \models \delta(\delta \in C, s \in S) \).

Reasoning 1: Presetting PoCTL formula \( \varphi_1 \) and \( \varphi_2 \), if \( \varphi_1 \cap \varphi_2 = \{\psi_1\} \), \( \psi_1 \) is public sub expression of \( \varphi_1 \) and \( \varphi_2 \). So model mark checking algorithm validate \( M, s \models \varphi_1(\varphi_1 \in C, s \in S) \) to \( M, s \models \varphi_1(\varphi_1 \in C, s \in S) \) and
Reasoning 2: Presetting PoCTL formula \( \varphi_1 \) = \( \psi \land \neg \phi \), \( \psi \in \{ \neg, \land, AF, EU, EX \} \), if \( \delta \subseteq \psi_1 \) or \( \delta \subset \psi_2 \), model mark checking algorithm definitely redundant validate \( M, s \models \delta \delta \in C, s \in S \) during \( M, s \models \varphi (\varphi \in C, s \in S) \) validation. Through above analysis, existing PoCTL formula has defects in model checking mark algorithm, e.i, redundant validation of public sub expression.

**PM_LA Algorithm**

**Pre-process Algorithm**

For the defects of public sub redundancy validation during existing model checking validating, the uniqueness of public sub expression is to be pre-process marked through correlated PoCTL formula. So to ensure only one model checking validating process of public sub expression of PoCTL formula, extension mechanism with syntax tree and logical structure is used to present PoCTL formula in this paper. The extension syntax analysis tree is generated as follow.

Supposed syntax tree structure of PoCTL formula is \( T<rootT> \), the extension syntax tree structure \( S<rootS> \) could be obtained by the steps.

1. \( S<rootS> \) is initial operated, i.e, rootS=null.
2. \( T<rootT> \) is postorder traversaled (LRD). With the output node N, when node N is null, the process jump to step (5), if not, jump to setp (3).
3. If the output node N of step (2) has uniqueness and \( N \subset S \), a new node M is generated. If N has uniqueness and \( N \subset S \), process jump to step (2). Otherwise, if N has non-uniqueness, N is correlated word, process jump to step (4).
4. If the output node N of step (2) belong to middle nodes and \( N \subset S \), and if root node is N which match sub tree of S, N is public sub expression of this syntax analysis tree. Otherwise, a new node M is generated, T is traced back to the sub node of S and mark is proceed by correlated word. Process jump to step (2).
5. If rootS=M, the output is root information of S, i.e, rootS.

An example PoCTL formula is given as formula (1) and formula (2).

\[
\begin{align*}
&\neg AX \neg p \land EX(p \land q) \lor EX(\neg p) \lor EX(p \land q) \\
&EF((AX \neg p) \land EX(p \land q))
\end{align*}
\]

The node set of public sub expression of two formulas is \( \{ p, q, \neg p, p \land q, EX(p \land q), AX \neg p \} \), in which, p is the direct sub expression of \( \neg p \) and \( p \land q \) is the sub expression of \( EX(p \land q) \). The public sub expression of PoCTL formula exist in every sub formula. How to having only one validation for many existing public sub expressions and avoiding redundancy validation become the key part of new syntax tree structure. Pre-process mechanism is applied in uniqueness mark of public sub expression of PoCTL formula in this paper. Many existing redundant syntax tree structure of PoCTL formula are combined to ensure unified validation of redundant public sub expression.

Extensional syntax tree structure of formula (1) and formula (2) combination is shown as figure 1.
PM_LA model Checking Mechanism

There are two parts in input parameter of PM_LA algorithm, i.e., extensional syntax tree structure of generated PoCTL formula set S and system model M. For arbitrary PoCTL formula $\phi$, after processed by the algorithm, output is the validation result of $M, s \models \phi (\exists_c, s \in S)$. There is difference from traditional model mark checking algorithm, PM_LA has pre-process before model checking validation to identify whether the sub public expression of PoCTL formula is public or direct. To mark uniqueness for every sub expression of PoCTL formula, if public sub expression is validated, it would not be validated in next validation. To improve model checking efficiency, after PoCTL formula checking, uniqueness mark of the non-public sub expression is deleted.

Assuming $\phi$ is sub expression of PoCTL formula, there is $\phi \subseteq \phi$ and $\psi$ is the direct sub expression of $\phi$. If all $\psi$ are pre-processed, the direct sub expression is uniqueness marked. Compliance operation of PM_LA algorithm detecting $\phi$ is as below.

(1) Whether $\phi$ has uniqueness mark is judged. If yes, the process jump to step (2), if no, jump to step (3).

(2) Whether $\phi$ has $M, s \models \phi (\exists_c, s \in S)$ validation is judged. If yes, the process jump to step (5), if no, jump to step (3).

(3) According to the uniqueness mark information of correlated word and pre-process mechanism, PM_LA algorithm is used to search $\phi$ status which meet requirement. The process jump to step (4).

(4) If $\phi$ is public sub expression of PoCTL formula, $\phi$ relative status information is reflected with system model M in real time. The uniqueness mark of $\phi$ of extensional syntax tree structure is detected. The process jump to step (5).

(5) If $\phi$ is equivalent to $\phi$, the uniqueness mark of non-public sub expression is deleted. Otherwise, the uniqueness mark status information of $\phi$ is outputed.

The pseudo code checking mechanism of PM_LA algorithm is as follow.

Input: Kripke* module,Tress* ctl_tree,Formular* f;

//system model M, PoCTL extensional syntax tree structure and sub expression which need to be validated
Output: Kripke* module;
Simulation
Preparation

To highlight the advantages of PM_LA proposed in this paper, VC or openGL platform is used to simulate PM_LA model checking algorithm. By static process and analyzing IDA Pro through $M = (S, \rightarrow, L)$ system model, operation of the system model being processed by PM_LA is validated which generate correlated status set $S = \{blk1, blk2, ..., blkn\}$. In status S, the uniqueness mark of pre-process mechanism is composed of M model call's argument presented by $L = \{\text{emt}, \text{syscall1}, \text{syscall2}, ..., \text{syscallm}\}$ (emt presents no capability of calling M model argument). PoCTL formula validation characteristic information is obtained by sequential regularities invoked by M model. There are different serial numbers correspond to different PoCTL formula sets.

1. $\text{EF}((\text{callset} = \text{GetModuleFileName}) \& \text{EF}(\text{callset} = \text{WriteFile}))$
2. $\text{EF}((\text{callset} = \text{FindFileFirst})\&(\text{callset} = \text{FindFileFirstEx})) \& \text{EF}(\text{callset} = \text{FindNextFile})$
3. $\text{EF}(\text{callset} = \text{GrtSystemDirectoty} \rightarrow (\text{EF}(\text{callset} = \text{CopyFile})\|\text{EF}(\text{callset} = \text{MoveFile})))$
4. $\text{EF}(\text{callset} = \text{GetTempDirection} \rightarrow \text{EF}(\text{callset} = \text{MoveFile}))$
5. $\text{EF}(\text{callset} = \text{RegQueryValueEX} \rightarrow \text{EF}(\text{callset} = \text{RegSetValueEX}))$
6. $\text{EF}(\text{callset} = \text{FindNextFile}) \& \text{EF}((\text{callset} = \text{CreatFileMapping})\&\text{EF}(\text{WriteFile})))$
7. $\text{EF}((\text{callset} = \text{GetSystemDirectory}\&(\text{callset} = \text{GetTempDirectory})) \& \text{EF}(\text{callset} = \text{WriteFile}))$
(8) $\text{EF}((\text{callset}=\text{fopen}) \rightarrow \text{EF}((\text{callset}=\text{send}))$
(9) $\text{EF}((\text{callset}=\text{bind}) \rightarrow (\text{EF}((\text{callset}=\text{listen}) \rightarrow \text{EF}((\text{callset}=\text{send})))$
(10) $\text{EF}((\text{callset}=\text{EnumProcesses}) \rightarrow \text{EF}((\text{callset}=\text{OpenProcess}))$

**Simulation**

Different PoCTL formula sets are simulated by PM_LA algorithm as shown in table 1.

<table>
<thead>
<tr>
<th>Prograname (exe)</th>
<th>Quantity of state</th>
<th>Quantity of migration</th>
<th>Quantity of owned public sub expression</th>
<th>Process time (s)</th>
<th>Traditional mark algorithm LA</th>
<th>PM_LA algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exl 523 2204</td>
<td>1</td>
<td>0</td>
<td>0.128</td>
<td>0.778</td>
<td>0.379</td>
<td></td>
</tr>
<tr>
<td>Exl 523 2204</td>
<td>3</td>
<td>37.4%</td>
<td>0.344</td>
<td>2.103</td>
<td>1.072</td>
<td></td>
</tr>
<tr>
<td>Exl 523 2204</td>
<td>5</td>
<td>71.8%</td>
<td>0.472</td>
<td>3.983</td>
<td>1.891</td>
<td></td>
</tr>
<tr>
<td>Exl 523 2204</td>
<td>7</td>
<td>13.3%</td>
<td>0.619</td>
<td>5.549</td>
<td>3.735</td>
<td></td>
</tr>
<tr>
<td>Exl 523 2204</td>
<td>9</td>
<td>52.1%</td>
<td>0.823</td>
<td>7.349</td>
<td>5.922</td>
<td></td>
</tr>
<tr>
<td>Exl 523 2204</td>
<td>10</td>
<td>3.9%</td>
<td>0.986</td>
<td>8.836</td>
<td>7.138</td>
<td></td>
</tr>
</tbody>
</table>

Table 1 shows that there are different results when processing different quantities of PoCTL formula sets for same executed program Exl. When model M is same and low quantity of public sub expression of PoCTL formula, the process time and efficiency of traditional model checking model algorithm (including reference [1] and [2]) are poor. Otherwise, if quantity of public sub expression of PoCTL formula is high, performance of the checking mechanism will be improved, compliance of public sub expression will be increased gradually. On the other side, the process time of PM_LA algorithm is shorter than traditional mark algorithm. The efficiency of PM_LA is better. To compare traditional LA algorithm and PM_LA algorithm in same system model M and PoCTL formula set, groups program system models (being less than 10M Bytes and from computer system directory exe files) are selected to analyzed, as shown in figure 2 and figure 3.

![Figure 2. Process time comparison of program state and model checking.](image)
Figure 2 shows that when the quantity of executable program state of system model is more, the PM_LA performance and efficiency is higher. Comparing to the algorithm of reference [1] and [2], efficiency of PM_LA is much better. Figure 3 shows that with quantity of PoCTL formula increasing, PM_LA algorithm performance and efficiency could keep increasing as linearly too. The simulation proves that in large scale and high complex PoCTL formula checking, PM_LA algorithm could complete validation with high efficiency.

Conclusion
To improve some defects such as high time complexity and low efficiency in PoCTL model checking, an improved PM_LA model checking algorithm is proposed in this paper. The kernel of this algorithm is to pre-process mark the uniqueness of public sub expression through correlated PoCTL formula firstly. The public sub expression and PoCTL model state are assigned with keeping balance of model checking space secondly. Validation is proceeded lastly. So that first-pass is guaranteed with high probability. Simulation shows that PM_LA could reduce correlated time complexity and improve performance of the validation. It is proved that PM_LA could be applied in large scale and high complexity environment.

Acknowledgment
This research was supported by Supported by Science and Technology Planning Project of Guangdong Province, China (No.2013B040500001, No.2014B040404002).

Reference


