Parameterized Cache Coherence Protocol Verification using Invariant

Zhongqin BI¹†, Meijing SHAN²

¹ School of Computer and Information Engineering, Shanghai University of Electric Power, Shanghai 200090, China
² Department of Information Science and Technology, East China University of Political and Law, Shanghai 201620, China

Abstract

Verification of parameterized cache coherence protocol is very important in the share-memory multiprocessor system. In this paper, a new method was proposed to verify the correctness of parameterized cache coherence protocol based on the invariant. Firstly, we present the parameterized cache coherence protocol as semi-algebraic transition system, and then solve the invariant of transition system based on inductive assertion map. Finally, verify the correctness of parameterized cache coherence protocol by checking whether the invariant satisfies the specification of parameterized cache coherence protocol or not.

Keywords: Parameterized Cache Coherence Protocol; Invariant; Semi-algebraic Transition System; Inductive Assertion Map; Correctness

1. Introduction

To reduce memory access latency and network traffic, local caches are commonly used in a shared-memory multiprocessor system. This method enables processors to work on local copies of main memory blocks, reducing the number of memory accesses during the execution of program. Although this configuration improves the performance of the system, it introduces a new problem: cache coherence problem. This problem is always supported by a cache coherence protocol, which ensures the data consistency of the system. Due to the complexity of hardware architectures, how to verify the correctness of the cache coherence protocol is becoming a major task.

In the last twenty years, some methods were proposed to verify the correctness of the parameterized cache coherence protocol. In [1], Pong and Dubois use the symbolic state model (SSM) to represent the state space and the verification of protocol. This verification method is based on a forward exploration with ad hoc expansion and aggregation rules. Norris Ip and Dill [2] have incorporated these techniques in Murϕ. In [3], G. Delzanno uses EFSM to modal protocol, and reduces the original parameterized verification problem to a control state reachability problem for a system with integer data variables. Finally, they check parameterized safety properties of abstract protocols using symbolic backward reachability. Ahí, Ki Yung [4] verify some correctness properties of the DASH cache coherence protocol using Ωmega.

In this paper, we present a new method to verify the correctness of parameterized cache coherence
protocol based on invariant. Firstly, we use semi-algebraic transition system to modal the protocol. Then, generate the invariant for the semi-algebraic transition system based on inductive assertion map. Finally, verify the correctness of protocol through the invariant satisfies the specification of the protocol.

The rest of the paper is organized as follows. Section 2 introduces some preliminaries, including semi-algebraic transition system, invariant, inductive assertion map and so on. Section 3 presents the algorithm to generate non-linear invariants for semi-algebraic transition system. Section 4 illustrates the method to verify the correctness of the parameterized cache coherence protocol. Section 5 draws a conclusion and discusses the future work.

2. Semi-algebraic Transition System

In this section, we introduce a computational model to represent cache coherence protocols for arbitrary numbers of processors. Here we recall some related concepts, such as invariant and inductive assertion map.

Definition 1 (Transition System\cite{5}). A transition system is a 5-tuple \(<V, L, T, l_0, \Theta>\), where:
- \(V\) is a set of variables.
- \(L\) is a set of locations.
- \(T\) is a set of transitions. A state \(s\) is an interpretation of the variables in \(V\). Each transition \(\tau \in T\) is a tuple \(<l_1, l_2, \rho_\tau, \theta_\tau>\), where \(l_1\) and \(l_2\) are the pre- and post locations of the transition. The transition relation \(\rho_\tau\) is a first-order assertion over \(V \cup V'\), where \(V\) denotes the current-state variable and \(V'\) denotes the next-state variable.
- \(l_0 \in L\) is the initial location.
- \(\Theta\) is a first-order assertion over \(V\) denoting the initial condition.

Transition system is the standard computational model used to represent many types of programs. More details about transition system can be found in \cite{5}.

Definition 2(Semi-algebraic Transition System\cite{6}). A semi-algebraic transition system is a transition system \(<V, L, T, l_0, \Theta>\). For each transition \(\tau \in T\) is a quadruple \(<l_1, l_2, \rho_\tau, \theta_\tau>\), where \(l_1\) and \(l_2\) and \(\rho_\tau\) is same as in definition 1, \(\theta_\tau\) is the guard of the transition. Only if \(\theta_\tau\) holds, the transition can take place. The transition relation \(\rho_\tau\) is an algebraic assertion over \(V \cup V'\), the initial condition \(\Theta\) and the guard of the transition \(\theta_\tau\) is an algebraic assertion over \(V\) which contains polynomial equations and inequalities.

It is easy to see that semi-algebraic transition system is the extension of algebraic transition system in \cite{7}.

From the definition of semi-algebraic transition system, we found cache coherence protocol can be denoted as semi-algebraic transition system.

Example 1: [Synapse N+1 protocol] Synapse N+1 protocol was developed by Synapse for the N+1 computer. This protocol is a write-allocation protocol with only three states, namely, invalid, valid, and dirty. Dirty is an exclusive state, only one cache can have a dirty line. The protocol is specified as Fig. 1.

The description of this protocol is described as following.

model SYNAPSE {
    var invalid, valid, dirty;
    states normal;
transition \( r_1 := \{
\text{from := normal; to := normal;}
\text{guard := invalid} \geq 1;
\text{action := \{invalid'=invalid+dirty-1, dirty'=0, valid'=valid+1\};}
\}; \)

transition \( r_2 := \{
\text{from := normal; to := normal;}
\text{guard := valid} \geq 1;
\text{action := \{invalid'=invalid+dirty+valid-1, dirty'=1, valid'=0\};}
\}; \)

transition \( r_3 := \{
\text{from := normal; to := normal;}
\text{guard := invalid} \geq 1;
\text{action := \{invalid'=invalid+dirty+valid-1, dirty'=1, valid'=0\};}
\}; \)

strategy \( s_1 \{ \)
\begin{align*}
& \text{setMaxState(0); setMaxAcc(100);}
& \text{Region init := \{state=normal && valid=0 && dirty=0 && invalid=k(>0)\}}
& \text{Transitions t :=\{r_1, r_2, r_3\};}
& \text{Region reach := post*(init, t);} \\
\end{align*}
\}

Synapse N+1 protocol can be presented as the following semi-algebraic transition system \( \Psi=<V, L, T, l_0, \Theta> \), where

- \( V=\{x_i, x_v, x_d\} \), which \( x_i, x_v, x_d \) represent the number of caches with state invalid, valid, dirty respectively.
- \( L \) is a set of all reachable states which Synapse N+1 protocol can reach.
- \( T=\{r_1, r_2, r_3\} \), where
  - \( r_1=\langle \text{normal, normal, } \rho_{11}, \theta_{11} \rangle, \rho_{11} = \{x_i' = x_i + x_d - 1, x_d' = 0, x_v' = x_v + 1\}, \theta_{11} = \{x_i \geq 1\}; \)
  - \( r_2=\langle \text{normal, normal, } \rho_{21}, \theta_{21} \rangle, \rho_{21} = \{x_i' = x_i + x_d + x_v - 1, x_d' = 1, x_v' = 0\}, \theta_{21} = \{x_i \geq 1\}; \)
  - \( r_3=\langle \text{normal, normal, } \rho_{31}, \theta_{31} \rangle, \rho_{31} = \{x_i' = x_i + x_d + x_v - 1, x_d' = 1, x_v' = 0\}, \theta_{31} = \{x_i \geq 1\}. \)
- \( l_0=\text{normal}. \)
Definition 3 (Invariant). Let $\Psi = <V, L, T, l_0, \Theta>$ be a semi-algebraic transition system. An invariant at a location $l \in L$ is defined as an assertion $\Phi$ over $V$, such that $\Phi$ holds on all states that can be reached at location $l$. An invariant of the semi-algebraic transition system is an assertion $\Phi$ which holds at all locations of the semi-algebraic transition system.

Definition 4 (Inductive Assertion Map). An assertion map $\eta$ is inductive if and only if the followings hold:

- **Initiation.** The assertion at $l_0$ subsumes the initial condition $\Theta \models \eta(l_0)$

- **Consecution.** For each transition $\tau$ from $l_i$ to $l_j$,
  $$\theta_\tau \land \eta(l_i) \land \rho_\tau \models \eta(l_j)'.
  $$

From the previous work of Floyd [8] and Hoare [9], if $\eta$ is an inductive assertion map then $\eta(l)$ is an invariant at $l$, but not vice versa. Actually, all known invariant generation methods are inductive assertion generation methods. The following algorithm is also based on inductive assertion generation methods.

### 3. Automated Invariant Generation Algorithm

Recently, several methods [6, 7, 10-17] were proposed to handle the real algebraic invariant generation. These methods can generate invariants for transition system, but they have one of the following difficulties.

1. Some methods may generate weak invariant due to use widening/narrowing operator, such as the algorithms in [10-12];
2. Due to the high complexity of real algebraic algorithms, some methods can not deal with transition system with many variables or many parameters, such as the algorithms in [6, 7, 13, 14];
3. Due to the ignorance of loop conditions and the branch conditions, some methods are not incomplete for transition system, such as the algorithms in [15-17].

Because of the above difficulties, these methods cannot generate the invariant of semi-algebraic transition systems which represent the cache coherence protocol. In this section, we present a new algorithm to generate non-linear invariant for semi-algebraic transition system. This algorithm can overcome the difficulties we mentioned above. Given a semi-algebraic transition system, an invariant is hypothesized to be a parameterized formula which named template. By the definition of invariant, all reachable states should satisfy the invariant. We choose $n$ reachable states to solve the parameters in template, and then verify the candidate invariants using inductive assertion map. The algorithm is described as follows. More details about this algorithm can be found in [18].

**Algorithm 1 (Non-linear Invariant Generation)**

**Input:** A semi-algebraic transition system $\Psi = <V, L, T, l_0, \Theta>$;  
**Output:** An invariant as the form of template, or report the invariant as the form of template is not exists.

1. Assume this semi-algebraic transition system has an invariant in the form of $I(u, x)$, where the vector $u$ includes $n$ variables;
2. Execute the transition system some times, and obtained $n$ reachable states $x_i$;
3. Substitute the reachable state $x_i$ into $I(u, x)$, and obtain $n$ equations about parameters $u$, is denoted as...
linear equation $F$;
(4). Obtain the fundamental set of solution by solving the equation $F$;
(5). Substitute the above solutions into $I(u, x)$, and obtain candidate invariants;
(6). Simplify the candidate invariants using Göbner Bases;
(7). Verify the candidate invariants by quantifier elimination tools QEPCAD based on the consecution condition of inductive assertion map. If the result is “true”, return the candidate invariants, else return not exists.

For Example 1, we obtain the following invariant if we assume Synapse N+1 protocol has an invariant with the quadric form.

$$x_i + x_d + x_v = k \land x_d x_v = 0 \land x_d^2 - x_d = 0$$

(1)

4. Verification of Parameterized Cache Coherence Protocol

In this section, we will present a gallery of verification of parameterized cache coherence protocol, such as Synapse N+1, Illinois, and MESI protocol.

4.1. Synapse N+1 Protocol

From the specification of Synapse N+1 protocol, this protocol is safe if and only if the following state sets are unreachable.

$$UN_1: x_d \geq 1 \land x_v \geq 1$$
$$UN_2: x_d \geq 2$$

From the formula (1), we notice, $x_d x_v = 0$ implies that unsafe state set $UN_1$ is unreachable, $x_d^2 - x_d = 0$ implies that $UN_2$ is unreachable. Therefore, the generated invariants suffice to guarantee the safety requirement.

4.2. Illinois Protocol

The Illinois protocol is a snoopy cache, write-invalidate, write-in coherence policy, originally proposed by Papamarcos and Patel. This protocol adds a shared status to reduce unnecessary invalidation traffic. So this protocol includes four states: invalid, valid-exclusive, dirty, and shared. The protocol is formally specified by Fig. 2, where

$$P \equiv \{x_i = 0 \land x_v = 0 \land x_d = 0\}$$

This protocol can be represented as the following semi-algebraic transition system $\Psi = \langle V, L, T, l_0, \Theta \rangle$, where

- $V = \{x_i, x_v, x_d, x_s\}$, which $x_i, x_v, x_d, x_s$ represent the number of caches with state invalid, valid-exclusive, dirty, and shared respectively.
- $L$ is a set of all reachable states which Illinois protocol can reach.
- $T = \{t_0, t_1, ..., t_{10}\}$, where
  - $t_0$ is normal, normal, $\rho_{t_0}, \theta_{t_0} = \{x_i = x_i - 1, x_v = x_v + 1\}$, $\theta_{t_0} = \{x_i \geq 1 \land x_v = 0\}$;
  - $t_2$ is normal, normal, $\rho_{t_2}, \theta_{t_2} = \{x_i = x_i - 1, x_v = x_v + 1, x_d = x_d + 1, x_e = 0\}$, $\theta_{t_2} = \{x_i \geq 1 \land x_v \geq 1\}$;
  - $t_3$ is normal, normal, $\rho_{t_3}, \theta_{t_3} = \{x_i = x_i - 1, x_v = x_v + 1, x_d = x_d + 1, x_e = 0\}$, $\theta_{t_3} = \{x_i \geq 1 \land x_v \geq 1\}$;
  - $t_4$ is normal, normal, $\rho_{t_4}, \theta_{t_4} = \{x_i = x_i - 1, x_v = x_v + 1, x_d = x_d + 1, x_e = 0\}$, $\theta_{t_4} = \{x_i \geq 1 \land x_v \geq 1\}$;
\( \tau_5 = \langle \text{normal, normal, } \rho_{\tau_5}, \theta_{\tau_5} \rangle, \rho_{\tau_5} = \{ x_e' = x_e-1, x_d = x_d+1 \}, \theta_{\tau_5} = \{ x_e \geq 1 \} \); 
\( \tau_6 = \langle \text{normal, normal, } \rho_{\tau_6}, \theta_{\tau_6} \rangle, \rho_{\tau_6} = \{ x_i = x_i + 1, x_d = x_d+1, x_s = 0 \}, \theta_{\tau_6} = \{ x_s \geq 1 \} \); 
\( \tau_7 = \langle \text{normal, normal, } \rho_{\tau_7}, \theta_{\tau_7} \rangle, \rho_{\tau_7} = \{ x_i = x_i + x_e + x_d + x_s - 1, x_e' = 0, x_s' = 0, x_d' = 1 \}, \theta_{\tau_7} = \{ x_d \geq 1 \} \); 
\( \tau_8 = \langle \text{normal, normal, } \rho_{\tau_8}, \theta_{\tau_8} \rangle, \rho_{\tau_8} = \{ x_d = x_d-1, x_i = x_i+1 \}, \theta_{\tau_8} = \{ x_d \geq 1 \} \); 
\( \tau_9 = \langle \text{normal, normal, } \rho_{\tau_9}, \theta_{\tau_9} \rangle, \rho_{\tau_9} = \{ x_s = x_s-1, x_i = x_i+1, x_e' = 0 \}, \theta_{\tau_9} = \{ x_s \geq 1 \} \); 
\( \tau_{10} = \langle \text{normal, normal, } \rho_{\tau_{10}}, \theta_{\tau_{10}} \rangle, \rho_{\tau_{10}} = \{ x_e = x_e-1, x_i = x_i+1 \}, \theta_{\tau_{10}} = \{ x_e \geq 1 \} \); 

This protocol is safe if and only if the following states sets are unreachable:

**UNS1:** \( x_d \geq 1 \land x_e + x_s \geq 1 \);

**UNS2:** \( x_d \geq 2 \);

**UNS3:** \( x_e \geq 2 \);

**UNS4:** \( x_e \geq 1 \land x_s \geq 1 \).

If we assume this protocol has an invariant with the quadric form, we will obtain the invariant.

\[ x_e + x_d + x_e \cdot k = 0 \land x_d x_e = 0 / x_e^2 = x_d / x_e x_s = 0 / x_e x_d = 0 \]

Notice that \( x_d x_e = 0 / x_e x_d = 0 \) implies that UNS1 is unreachable, \( x_e^2 = x_d \) implies that UNS2 is unreachable, \( x_e^2 = x_e \) implies that UNS3 is unreachable, \( x_e x_s = 0 \) implies that UNS4 is unreachable too. Therefore, this protocol is safe.

### 4.3. MESI Protocol

The MESI protocol is a write-once cache-based protocol. It is the most common protocol which supports write-back cache. Every cache line is marked with one of the four following states: modified, exclusive, shared, and invalid. It is formally specified by Fig. 3. This protocol can be represented as the following semi-algebraic transition system \( \Psi = \langle V, L, T, l_0, \Theta \rangle \), where

- \( V = \{ x_i, x_m, x_e, x_s \} \), which \( x_i, x_m, x_e, x_s \) represent the number of caches with state invalid, modified, exclusive, shared respectively.
- \( L \) is a set of all reachable states which MESI protocol can reach.
- \( T = \{ \tau_1, \tau_2, \tau_3, \tau_4 \} \), where
\begin{itemize}
  \item $r_1=$<normal, normal, $\rho_1$, $\theta_1$>, $\rho_1=$\{\text{$x'_s=x_s+x_m+1$, $x'_e=0$, $x'_m=0$, $x'_i=x_i-1$}\}, $\theta_1=$\{\text{$x_i \geq 1$}\};
  \item $r_2=$<normal, normal, $\rho_2$, $\theta_2$>, $\rho_2=$\{\text{$x'_m=x_m+1$, $x'_e=x_e-1$}\}, $\theta_2=$\{\text{$x_e \geq 1$}\};
  \item $r_3=$<normal, normal, $\rho_3$, $\theta_3$>, $\rho_3=$\{\text{$x'_i=x_i+x_m+x_e+x_s-1$, $x'_m=0$, $x'_e=1$, $x'_s=0$}\}, $\theta_3=$\{\text{$x_i \geq 1$}\};
  \item $r_4=$<normal, normal, $\rho_4$, $\theta_4$>, $\rho_4=$\{\text{$x'_i=x_i+x_m+x_e+x_s-1$, $x'_m=0$, $x'_e=1$, $x'_s=0$}\}, $\theta_4=$\{\text{$x_i \geq 1$}\}.
\end{itemize}

$\text{Fig. 3 MESI Protocol}$

- $I_0 =$ normal.
- $\Theta =$\{\text{$x'_m=0$} \& \text{$x'_e=0$} \& \text{$x'_s=0$} \& \text{$x_i=k(k>0)$}\}.

This protocol is safe if and only if the following states sets are unreachable.
- UNS1: $x'_e+x'_s \geq 1 \& x'_m \geq 1$;
- UNS2: $x'_m \geq 2$;
- UNS3: $x'_e \geq 1 \& x'_s \geq 1$;
- UNS4: $x'_e \geq 2$.

If we assume this protocol has an invariant with the quadric form, we will obtain the following invariant.

\[ x'_i+x'_m+x'_e+x'_s=k \& x'_i+x'_m+x'_e=0 \& x'_m+x'_e=0 \& x'_s=0 \& x'_e=0 \& x'_m=0 \& x'_e=0 \]

Noticed that $x'_m+x'_e=0$ implies that UNS1 is unreachable, $x'_m^2-x'_m=0$ implies that UNS2 is unreachable, $x'_e=0$ implies that UNS3 is unreachable, $x'_s^2-x'_s=0$ implies that UNS4 is unreachable too. Therefore, this protocol is safe.

Using the above method, we can also verify the correctness of MOESI, DEC Firefly, and Xerox PARC Dragon protocol.

5. Conclusion

In this paper we have proposed a new method to verify the correctness of parameterized cache coherence protocol. However, because the invariant generation of semi-algebraic transition system is based on the template, it is incomplete. How to estimate the bound of the invariants is an important task. In the verification of parameterized cache coherence protocol, we find the quadratic form of the invariant is sufficient. How to prove whether this result is right or not is our future work.
Acknowledgement

The author would like to thank Lu Yang and Zhenbing Zeng for their help and support. Thanks also to the colleagues in our seminar course on program verification. This work was supported by National Nature Science Foundation of China (No. 61073189), Research Fund for Excellent Youth Scholars of Shanghai Higher Education (No. Z2011-035), and Research Fund of Shanghai University of Electric Power (No. K2010-009).

References