Formal Analysis of Memory Contention in a Multiprocessor System

Liya Liu, Osman Hasan, and Sofiène Tahar

Dept. of Electrical & Computer Engineering, Concordia University
1455 de Maisonneuve W., Montreal, Quebec, H3G 1M8, Canada
{liy_liu,o_hasan,tahar}@ece.concordia.ca

Abstract. Multi-core processors along with multi-module memories are extensively being used in high performance computers these days. One of the main performance evaluation metrics in such configurations is the memory contention problem and its effect on the overall memory access time. Usually, this problem is analyzed using simulation or numerical methods. However, these methods either cannot guarantee accurate analysis or are not scalable due to the unacceptable computation times. As an alternative approach, this paper uses theorem proving to analyze the memory contention problem of a multiprocessor system. For this purpose, the paper presents the higher-order-logic formalization of the expectation of a discrete random variable and Discrete-time Markov Reward Models. These foundations are then utilized to analyze the memory contention problem of a multi-processor system configuration with two processors and two memory modules using the HOL theorem prover.

1 Introduction

The extensive computation requirements in complex engineering systems and the trend to move towards smart consumer electronic devices has brought a paradigm shift towards using multi-core processors in all sorts of embedded systems. These processors usually share information with one another by accessing shared variables in a common memory space. In order to avoid concurrent updates to these shared variables, which may lead to erroneous results, only one processor at a time is allowed to access the memory. However, this configuration leads to the well-known memory contention problem, which results in an overall performance degradation as the processors may have to wait for accessing the memory. This problem is usually alleviated by using a multi-module memory, as depicted in Figure 1. The main idea is to divide the cache memory into submodules so that the processors can simultaneously access different sub-modules in parallel. This configuration tends to minimize the memory contention problem but cannot rectify it completely since two or more processors may want to access the same memory sub-module as well. Thus, rigorous performance analysis is conducted to determine the optimized size of sub-modules of memory for a given memory access rate.

Due to the random nature of time dependent memory access requests, the above mentioned configurations are modeled as classified Discrete-time Markov

J. Iyoda and L. de Moura (Eds.): SBMF 2013, LNCS 8195, pp. 195–210, 2013.

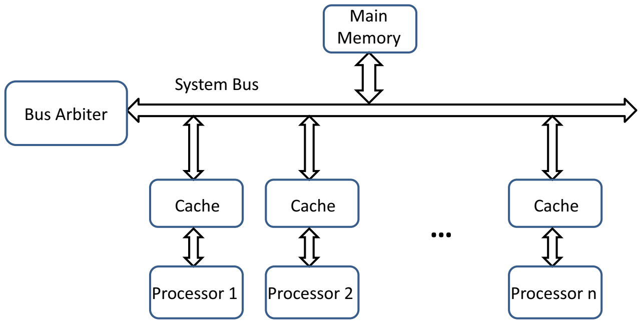

Fig. 1. A Multiprocessor System with Multimodule Memory

Chains (DTMCs) [2]. Then performance characteristics, such as the average number of memory accesses and the steady state probabilities of processors waiting, can be deduced based on the properties of classified Markov chains and Discrete-time Markov Reward Models (DMRMs) [18]. These properties are expressed in terms of the transition probabilities of the given Markov chain and thus provide useful insights for system optimization.

Traditionally, the above mentioned performance analysis is conducted analytically, using paper-and-pencil proof methods [19], computer simulations [5] or numerical methods [17]. The paper-and-pencil proof methods do not scale well to the complex multi-processor systems. Moreover, they are prone to human errors. Computer based simulations or numerical methods are scalable but due to the usage of computer arithmetic and pseudo random numbers and their inherent incompleteness cannot guarantee accurate results.

The accuracy of the above mentioned performance analysis is becoming quite important these days due to the increasing usage of multi-processor systems in safety-critical domains like medicine and transportation. Recently, probabilistic model checking has been used to analyze DMRMs (e.g., [4] and [8]). The typical model checking tools are *PRISM* [16] and *MRMC* [13], which provide precise system analysis by modeling the stochastic behaviors using probabilistic state machines and exhaustively verifying their probabilistic properties. These tools can be used for performance analysis of multi-processor systems as well. However, some algorithms implemented in these model checking tools are also based on numerical methods. For example, the Power method [15], which is a well-known iterative method, is applied to compute the steady-state probabilities (or limiting probabilities) of Markov chains in PRISM. Thus, most of the stationary properties analyzed in model checkers are time bounded. Moreover, probabilistic model checking often utilizes unverified algorithms and optimization techniques. Finally, model checking cannot be used to verify generic mathematical expressions for statistical properties, like expectation.

In order to provide an accurate and complete approach for analyzing the memory contention problem of multi-processor systems, we propose to use higher-order-logic theorem proving. The high expressiveness of higher-order logic allows us to formally express the systems that can be modeled using classified Markov chains and DMRMs. Whereas, the soundness of theorem proving guarantees the correctness and completeness of the analysis. In this paper, we develop the formalization of Discrete-time Markov Reward Models based on the formalization of expectation and conditional expectation functions for discrete random variables along with the available formalization of Discrete-Time Markov Chains (DTMCs) [10]. Compared to the work in [10], which is based on the formalized probability theory of Hurd [7], the formalization of DTMCs in the current paper is developed by building upon a more general probability theory developed by Mhamdi [12]. This update provides us with the flexibility to model time-inhomogeneous DTMCs/DMRMs or several random processes (involving DTMCs) containing distinct types of state spaces. This paper also presents the formal verification of some classical properties of expectation and DMRMs. The above mentioned formalizations allow us to analyze the memory contention problem of any multi-processor system. For illustration purposes, we formally analyze a typical multi-processor system [19] using the formalization of DMRMs and the irreducible and aperiodic Markov Chains [11].

2 Preliminaries

In this section, we present the foundations that we build upon to formalize expectation for discrete random variables and DMRM later.

2.1 Probability Theory

A *probability space* is a measure space $(\Omega, \Sigma, \mathcal{P}r)$ such that $\mathcal{P}r(\Omega) = 1$ [2]. Σ is a collection of subsets of Ω called *measurable sets*. In [12], a higher-order logic theory is developed where given a probability space `p`, the functions `space` and `subsets` return the corresponding Ω and Σ, respectively. Mathematically, a *random variable* is a measurable function between a probability space and a measurable space. This is formalized in HOL by a predicate `random_variable X p s` that returns true if `X` is a random variable on a probability space `p` and an outcome space `s`.

The *expectation* [20] of a random variable plays an important role in describing the characteristics of probability distributions. A *conditional expectation* represents the expected value of a *real* random variable considering a conditional probability distribution. Mhamdi [12] formalized general definitions of expectation and conditional expectation using the Lebesgue integral. These definitions can be used to find the expectations involving both discrete and continuous random variables. However, it is not a straightforward task to use these definitions to reason about the expectation of discrete random variables as the proofs of even the basic theorems require the Radon Nikodym derivative [6] and a series of intermediate theorems. In this paper, we formalize the expectation and conditional expectation for the discrete case to avoid these complex reasoning problems. These definitions are then used to formalize DMRM in HOL.

2.2 Discrete-Time Markov Chains

Given a probability space, a stochastic process $\{X_t : \Omega \rightarrow S\}$ represents a sequence of random variables X, where t represents the time that can be discrete (represented by non-negative integers) or continuous (represented by real numbers) [2]. The set of values taken by each X_t, commonly called *states*, is referred to as the *state space*. The *sample space* Ω of the process consists of all the possible state sequences based on a given state space S. Now, based on these definitions, a *Markov chain* is a Markov process [3], with finite or countably infinite state space Ω, that satisfies the following :

$$\mathcal{P}r\{X_{t_{n+1}} = f_{n+1}|X_{t_n} = f_n, \ldots, X_{t_0} = f_0\} = \mathcal{P}r\{X_{t_{n+1}} = f_{n+1}|X_{t_n} = f_n\}$$

for $0 \leq t_0 \leq \cdots \leq t_n$ and $f_0, \cdots, f_{n+1}$ in the state space. This means that the future state is only dependent on the current state and is independent of all the other past states. The Markov property can be formalized as follows:

Definition 1 (Markov Property).

```
⊢ ∀ X p s. mc_property X p s =
  (∀ t. random_variable (X t) p s) ∧
  ∀ f t n.
   increasing_seq t ∧ ℙ(⋂k∈ [0,n−1]{x | X tk x = f k}) ≠ 0 ⇒
  (ℙ({x | X tn+1 x = f (n + 1)}|
      {x | X tn x = f n} ∩ ⋂k∈ [0,n−1]{x | X tk x = f k}) =
    ℙ({x | X tn+1 x = f (n + 1)}|{x | X tn x = f n}))
```

where `increasing_seq t` is defined as `∀ i j. i < j ⇒ t i < t j`. The first conjunct indicates that the Markov property is based on a random process $\{X_t : \Omega \rightarrow S\}$. The quantified variable `X` represents a function of the random variables associated with time `t` which has the type `num`. This ensures the process is a *discrete time* random process. The random variables in this process are the functions built on the probability space `p` and a measurable space `s`. The conjunct $\mathbb{P}(\bigcap_{k\in [0,n-1]}\{x \mid X\ t_k\ x = f\ k\}) \neq 0$ ensures that the corresponding conditional probabilities are well-defined, where `f k` returns the $\mathtt{k}^{th}$ element of the state sequence.

A DTMC is usually expressed by specifying: an initial distribution p_0 which gives the probability of initial occurrence $\mathcal{P}r(X_0 = s) = p_0(s)$ for every state `s`; and transition probabilities $p_{ij}(t)$ which give the probability of going from i to j for every pair of states (i, j) in the state space [14]. For states i, j and a time t, the *transition probability* $p_{ij}(t)$ is defined as $\mathcal{P}r\{X_{t+1} = j|X_t = i\}$, which can be easily generalized to *n-step transition probability* as shown in Equation (1), and it can be formalized in Definition 2.

$$p_{ij}^{(n)}(t) = \begin{cases} \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases} & n = 0 \\ \mathcal{P}r\{X_{t+n} = j|X_t = i\} & n > 0 \end{cases} \tag{1}$$

Definition 2 (Transition Probability).

```
⊢ ∀ X p s t n i j. Trans X p s t n i j =
  if i ∈ space s ∧ j ∈ space s then
    if n = 0 then
      if (i = j) then 1 else 0
    else ℙ({x | X (t + n) x = j}|{x | X t x = i})
  else 0
```

Now, the Discrete Time Markov Chain (DTMC) can be formalized as follows:

Definition 3 (DTMC).

```
⊢ ∀ X p s p0 pij dtmc X p s p0 pij =
  mc_property X p s ∧ (∀ i. i ∈ space s ⇒ {i} ∈ subsets s) ∧
  ∀ i. i ∈ space s ⇒ (p0 i = ℙ{x | X t x = i}) ∧
  ∀ t i j. ℙ{x | X t x = i} ≠ 0 ⇒
           (pij t i j = Trans X p s t 1 i j)
```

Most of the applications actually make use of *time-homogenous DTMCs*, i.e., DTMCs with finite state-space and time-independent transition probabilities [1]. The time-homogenous property refers to the time invariant feature of a random process: $\forall\ t\ t'.\ p_{ij}(t) = p_{ij}(t')$ (in the sequel, $p_{ij}(t)$ is simply written as p_{ij}).

Definition 4 (Time homogeneous DTMC).

```
⊢ ∀ X p s p0 pij. th_dtmc X p s p0 pij =
   dtmc X p s p0 pij ∧ FINITE (space s) ∧
   ∀ t i j.
     ℙ{x | X t x = i} ≠ 0 ∧ ℙ{x | X (t + 1) x = i} ≠ 0 ⇒
     (Trans X p s (t + 1) 1 i j = Trans X p s t 1 i j)
```

Using these fundamental definitions, we formally verified most of the classical properties of DTMCs with finite state-space in HOL [11]. Some of the relevant ones to the context of this paper are presented here.

The *joint probability distribution* of a DTMC is the probability of a chain of states to occur:

$$\mathcal{P}r(X_t = S_0, \cdots, X_{t+n} = S_n) = \prod_{k=0}^{n-1} \mathcal{P}r(X_{t+k+1} = S_{k+1} | X_{t+k} = S_k)\mathcal{P}r(X_t = S_0)$$

Theorem 1 (Joint Probability Distribution).

```
⊢ ∀ X p s t n S p0 pij n.
  dtmc X p s p0 pij ⇒
  ℙ(⋂_{k=0}^{n} {x | X (t + k) x = EL k S}) =
  ∏_{k=0}^{n-1} ℙ({x | X (t + k + 1) x = EL (k + 1) S}|
         {x | X (t + k) x = EL k S})ℙ{x | X t x = EL 0 S}
```

The Chapman-Kolmogorov Equation [2] $p_{ij}^{(m+n)} = \sum_{k \in \Omega} p_{ik}^{(m)} p_{kj}^{(n)}$ is a widely used property of time homogeneous DTMCs. It basically gives the probability of going from state i to j in $m+n$ steps. Assuming the first m steps take the system from state i to some intermediate state k and the remaining n steps then take the system from state k to j, we can obtain the desired probability by adding the probabilities associated with all the intermediate steps.

Theorem 2 (Chapman-Kolmogorov Equation).

$$\begin{array}{l} \vdash \forall\ \mathtt{X\ p\ s\ i\ j\ t\ m\ n}\ \mathtt{p_0}\ \mathtt{p_{ij}}. \\ \quad \mathtt{th\_dtmc\ X\ p\ s\ p_0\ p_{ij}} \Rightarrow \\ \quad \mathtt{Trans\ X\ p\ s\ t\ (m\ +\ n)\ i\ j\ =} \\ \quad\quad \sum_{\mathtt{k \in space\ s}} \mathtt{(Trans\ X\ p\ s\ (t\ +\ m)\ n\ i\ k\ *\ Trans\ X\ p\ s\ t\ m\ k\ j)} \end{array}$$

The unconditional probabilities associated with a Markov chain are called *absolute probabilities*, which can be computed by applying the initial distributions and n-step transition probabilities, as $p_j^{(n)} = \mathcal{P}r(X_n = j) = \sum_{k \in \Omega} \mathcal{P}r(X_0 = k)\mathcal{P}r(X_n = j | X_0 = k)$. Using $p_i^{(n)}$ for the probability $\mathcal{P}r(X_n = j)$, we verified the following result:

Theorem 3 (Absolute Probability).

$$\begin{array}{l} \vdash \forall\ \mathtt{X\ p\ s\ j\ n}\ \mathtt{p_0}\ \mathtt{p_{ij}}. \\ \quad \mathtt{th\_dtmc\ X\ p\ s\ p_0\ p_{ij}} \Rightarrow \\ \quad \mathbb{P}\{\mathtt{x\ |\ X\ n\ x\ =\ j}\}\ = \\ \quad\quad \sum_{\mathtt{k \in space\ s}} \mathbb{P}\{\mathtt{x\ |\ X\ 0\ x\ =\ k}\}\mathbb{P}(\{\mathtt{x\ |\ X\ n\ x\ =\ j}\}|\{\mathtt{x\ |\ X\ 0\ s\ =\ k}\}) \end{array}$$

2.3 Aperiodic and Irreducible Markov Chain

Aperiodic and irreducible DTMCs are considered to be the most widely used classified Markov chains in analyzing Markovian systems due to their attractive stationary properties, i.e., their limit probability distributions are independent of the initial distributions.

The foremost concept of classified DTMCs is the *first passage time* τ_j, or the *first hitting time*, which is defined as the minimum time required to reach a state j from the initial state i, $\tau_j = min\{t > 0 : X_t = j\}$.

Definition 5 (First Passage Time).

$$\vdash \forall\ \mathtt{X\ x\ j.\ FPT\ X\ x\ j\ =\ MINSET}\ \{\mathtt{t}\ |\ \mathtt{0} < \mathtt{t} \wedge \mathtt{(X\ t\ x\ =\ j)}\}$$

where `X` is a random process and `x` is a sample in the probability space associated with the random variable X_t.

The conditional distribution of τ_j, defined as the probability of the events starting from state i and visiting state j at time n, is expressed as $f_{ij}^{(n)} = \mathcal{P}r\{\tau_j = n | X_0 = i\}$.

Definition 6 (Probability of First Passage Events).

```
⊢ ∀ X p i j n.
  f X p i j n = ℙ({x | FPT X x j = n}|{x | X 0 x = i})
```

Another important notion is the probability of the events starting from state i and visiting state j at all times n, which is expressed as $f_{ij} = \sum_{n=1}^{\infty} f_{ij}^{(n)}$. It can be expressed in HOL as `(λ n. f X p i j n) sums` f_{ij}. Now f_{jj} provides the probability of events starting from state j and eventually returning back to j. A state j in a DTMC is called *persistent* if $f_{jj} = 1$.

The greatest common divisor (*gcd*) of a set is a frequently used mathematical concept in defining classified states. For a state j, a *period* of j is any n such that $p_{jj}^{(n)}$ is greater than 0. We write $d_j = gcd\ \{n : p_{jj}^{(n)} > 0\}$ as the gcd of the set of all periods.

A state i is said to be *accessible* from a state j (written $j \to i$), if the n-step transition probability of the events from state i to j is nonzero. Two states i, j are called *communicating states* (written $i \leftrightarrow j$) if they are mutually accessible. The formalization of these foundational notions is given in Table 1.

Table 1. Formalization of Classified States

Definition	Condition	HOL Formalization
Persistent State	$f_{jj} = 1$	`⊢ Persistent_state X p j =` `∀ x. {t ∣ 0 < t ∧ (X t x = j)} ≠ ∅) ∧` `(λ n. f X p j j n) sums 1`
Periods of a State	$0 < \mathrm{n}$ $0 < p_{jj}^{n}$	`⊢ Period_set X p s j =` `{n ∣ Persistent_state X p j ∧ 0 < n ∧` `∀ t. 0 < Trans X p s t n j j}`
gcd of a Set	gcd A	`⊢ GCD_SET A =` `MAXSET {r ∣ ∀ x. x ∈ A ⇒ divides r x}`
gcd of a Period Set	d_j	`⊢ Period X p s j = GCD_SET (Period_set X p s j)`
Periodic State	$d_j > 1$	`⊢ Periodic_state X p s j =` `1 < Period X p s j ∧ Period_set X p s j ≠ ∅`
Aperiodic State	$d_j = 1$	`⊢ Aperiodic_state X p s j =` `(Period X p s j = 1) ∧ Period_set X p s j ≠ ∅`
Accessibility	$\mathrm{i} \to \mathrm{j}$	`⊢ Accessibility X p s i j =` `∀ t. ∃ n. 0 < n ∧ 0 < Trans X p s t n i j`
Communicating State	$\mathrm{i} \leftrightarrow \mathrm{j}$	`⊢ Communicating_states X p s i j =` `Accessibility X p s i j ∧ Accessibility X p s j i`

Now, a DTMC is considered as *aperiodic* if every state in its state space is an aperiodic state; and a DTMC is said to be *irreducible* if every state in its state space can be reached from any other state including itself in finite steps.

Definition 7 (Aperiodic DTMC).

```
⊢ ∀ X p s p0 pij. Aperiodic_mc X p s p0 pij =
  th_dtmc X p s p0 pij ∧
  ∀ i. i ∈ space s ⇒ Aperiodic_state X p s i
```

Definition 8 (Irreducible DTMC).

```
⊢ ∀ X p s p0 pij. Irreducible_mc X p s p0 pij =
  th_dtmc X p s p0 pij ∧
  (∀ i j. i ∈ space s ∧ j ∈ space s ⇒
        Communicating_states X p s i j)
```

The above mentioned definitions are mainly used to formally specify and analyze the dynamic features of Markovian systems within the sound environment of a theorem prover. In this paper, we will be using them to formalize a behavior of the multi-processor system in Section 4.

2.4 Long-Term Properties

The long-run probability distributions (limit probability distributions) are often considered in the convergence analysis of random variables in stochastic systems. It is not very easy to verify that the stationary behaviors of a certain state exists in a generic non-trivial DTMC, because the computations required in such an analysis are often tremendous. However, in aperiodic and irreducible DTMCs, we can prove that any state in the state space possesses a convergent probability distribution, by the following theorems.

For any state i in the finite state space S of an aperiodic DTMC, there exists an $N < \infty$ such that $0 < p_{ii}^{(n)}$, for all $n \geq N$.

Theorem 4 (Positive Return Probability).

```
⊢ ∀ X p s p0 pij i t.
  Aperiodic_DTMC X p s p0 pii ∧ i ∈ space s ⇒
  ∃ N. ∀ n. N ≤ n ⇒ 0 < Trans X p s t n i i
```

Applying Theorem 4, we can prove that, for any aperiodic and irreducible DTMC with finite state space S, there exists an N, for all $n \geq N$, such that the n-step transition probability $\mathrm{p}_{ij}^{(n)}$ is non-zero, for all states i and $j \in S$.

Theorem 5 (Existence of Positive Transition Probabilities).

```
⊢ ∀ X p s p0 pij i j t.
  Aperiodic_DTMC X p s p0 pij ∧ Irreducible_DTMC X p s p0 pij ∧
  i ∈ space s ∧ j ∈ space s ⇒
  ∃ N. ∀ n. N ≤ n ⇒ 0 < Trans X p s t n i j
```

Utilizing Theorems 4 and 5, the convergence of the probability distributions in an aperiodic and irreducible DTMC can be verified as the following theorem:

Theorem 6 (Convergent Probability Distributions).

```
⊢ ∀ X p s p0 pij i j.
  Aperiodic_DTMC X p s p0 pij ∧ Irreducible_DTMC X p s p0 pij ⇒
  convergent (λ t. ℙ{x | X t x = i})
```

As multiprocessor systems are usually modeled as aperiodic and irreducible DTMCs, the theorems presented above are very useful in analyzing their long-term behaviors. However, to the best of our knowledge, the second requirement for analyzing multiprocessor systems, i.e., the reward or cost factors for DTMCs have not been formalized so far. Therefore, we build upon the foundations, presented in this section, to formalize the Discrete-time Markov Reward Models in order to facilitate the performance analysis of multi-processor systems in HOL.

3 Formalization of Discrete-time Markov Reward Models

In this section, we formally define expectation and conditional expectation of a discrete random variable and then use these results along with the formal DTMC definition to formalize a Discrete-time Markov Reward Model (DMRM).

3.1 Expectation

The *expectation* (also called *expected value*) of a discrete random variable X is $E[X] = \sum_{i \in space\ s_x} i\mathcal{P}r\{X = i\}$. Whereas, the *conditional expectation* of a discrete random variable X given a condition Y is $E[Y|X] = \sum_{i \in space\ s_x} i\mathcal{P}r\{Y|X = i\}$. These definitions can be formalized as:

Definition 9 (Expectation).

$\vdash \forall$ X p s_x. expec X p s_x = $\sum_{\texttt{i} \in \texttt{space}\ s_x} \texttt{i}\mathbb{P}\{\texttt{x} \mid \texttt{X x = i}\}$

Definition 10 (Conditional Expectation).

$\vdash \forall$ X Y y p s_x.
  cond_expec Y X y p s_x = $\sum_{\texttt{i} \in \texttt{space}\ s_x} \texttt{i}\mathbb{P}(\{\texttt{x} \mid \texttt{Y x = y}\} | \{\texttt{x} \mid \texttt{X x = i}\})$

where X is a discrete random variable, which has type 'a $\rightarrow$ real, s_x is a finite state space, and $\{\texttt{x} \mid \texttt{Y x = y}\}$ is a discrete event given in the conditional probability to calculate the expectation.

Utilizing these two formal definitions, we can verify some interesting properties of expectation that play a vital role in the performance analysis of multi-processor systems. We can prove that the total expectation of a random variable X is $\mathrm{E[Y]} = \sum_{\texttt{j} \in \texttt{space}\ s_x} \mathrm{E[Y|}X_j] \mathcal{P}r\{X_j\}$. Here, X_j represents a discrete event involved in the event space (subsets s_x) and j is any state in the state space (space s_x) of random variable X.

Theorem 7 (Total Expectation).

$\vdash \forall$ X Y p s_x s_y.
  random_variable X p s_x $\wedge$ random_variable Y p s_y $\wedge$
  ($\forall$ x. x $\in$ space s_x $\Rightarrow$ {x} $\in$ subsets s_x) $\wedge$
  ($\forall$ x. x $\in$ space s_y $\Rightarrow$ {x} $\in$ subsets s_y) $\wedge$
  FINITE (space s_x) $\wedge$ FINITE (space s_y) $\Rightarrow$
   (expec Y p s_x =
   $\sum_{\texttt{j} \in \texttt{space}\ s_x}$ (λj. cond_expec Y X j p s_x * $\mathbb{P}\{\texttt{x} \mid \texttt{X x = j}\}$))

For a random process $\{X_t\}_{t \geq 0}$, with sample space $\mathtt{s_x}$, and discrete event $\{Y_t = y\}_{t \geq 0}$, in the event space (subsets $\mathtt{s_y}$) (for all y in the finite state space $\mathtt{s_y}$), the total expectation of the steady-state of the random variable Y is: $\lim_{t \to \infty} \mathrm{E}[Y_t] = \sum_{\mathtt{j} \in \mathtt{space\ s_x}} \lim_{t \to \infty} \mathrm{E}[Y_t | X_t = j] \lim_{t \to \infty} \mathcal{P}r\{X_t = j\}$

Theorem 8 (Total Expectation of Steady-state Probabilities).

```
⊢ ∀ X Y p sx sy.
   (∀ t. random_variable (X t) p sx) ∧
   (∀ t. random_variable (Y t) p sy) ∧
   (∀ x. x ∈ space sx ⇒ {x} ∈ subsets sx) ∧
   (∀ x. x ∈ space sy ⇒ {x} ∈ subsets sy) ∧
   (∀ j. convergent (λt. ℙ{x | X t x = j}) ∧
   (∀ i. convergent (λt. cond_expec (Y t) (X t) i p sx)) ∧
   FINITE (space sx) ∧ FINITE (space sy) ⇒
    (lim (λ t. expec (Y t) p sy) =
    ∑j∈space sx (lim (λ t. cond_expec (Y t) (X t) j p sx) *
                 lim (λ t. ℙ{x | X t x = j})))
```

3.2 Discrete-time Markov Reward Models

Discrete-time Markov Reward Models (DMRMs) are extended DTMCs that consider the costs, or dually bonuses (rewards). In the performance analysis of some real-world systems, DMRMs allow numerous quantitative measures of the system, such as the elapsed time, power consumption, size of message queue, net profit, etc.

Mathematically, a DMRM is defined on a DTMC $\{\mathrm{X}_t\}_{t \geq 0}$ with a real valued reward function r_{xy}, which associates a real reward (or cost) to a state x in the state space of X for all t, $t \geq 0$ by the conditional expectation of the reward (or cost) given the state x.

Definition 11 (Discrete-time Markov Reward Model).

```
⊢ ∀ X Y p sx sy p0 pij rxy. dmrm X Y p sx sy p0 pij =
   dtmc X p sx p0 pij ∧ (∀ t. random_variable (Y t) p sy) ∧
   (∀ y. y ∈ space sy ⇒ {y} ∈ subsets sy) ∧
   (∀ x t. ℙ{x | Y t x = y} ≠ 0 ⇒
           (rxy t x = cond_expec (Y t) (X t) x p sy))
```

where the quantified variable `X` refers to the random variables involved in the underlying DTMC, `Y` indicates the random reward, `p` is the probability space, $\mathtt{s_x}$ refers to the state space of the DTMC, $\mathtt{s_y}$ represents the measurable state space of random variable `Y`, $\mathtt{p_0}$ and $\mathtt{p_{ij}}$ are the initial distribution and transition probability of the DTMC, and $\mathrm{r}_{\mathrm{x}y}$ denotes the reward function. The first conjunct in this definition ensures that the underlying stochastic process is a DTMC, the second and third conjuncts constrain the expected values are discrete random variables `(Y t)` and the last condition gives the conditional expectation distributions by the reward function.

It is important to note that this definition provides a general DMRM, in which the state space can be finite or infinite, the underlying DTMC can be time-homogeneous or time-inhomogeneous, and the reward is a function of time (this feature facilitates the modeling of the impulse reward in some systems [4]).

Very often, the underlying DTMC in a DMRM is considered as a time-homogeneous DTMC with a finite state space and the rewards or costs are considered as constants for the corresponding states. We formalize this frequently used DMRM as follows:

Definition 12 (DMRM with Time-homogeneous Property).

$\vdash \forall$ X Y p s_x s_y p_0 p_{ij} r_{xy}. th_dmrm X Y p s_x s_y p_0 p_{ij} r_{xy} =
dmrm X Y p s_x s_y p_0 p_{ij} r_{xy} $\wedge$ FINITE (space s_y) $\wedge$
($\forall$ x t t'. r_{xy} t x = r_{xy} t' x) $\wedge$
($\forall$ t i j.
$\mathbb{P}\{$x | X t x = i$\}$ $\neq$ 0 $\wedge$ $\mathbb{P}\{$x | X (t + 1) x = i$\}$ $\neq$ 0 $\Rightarrow$
p_{ij} X p s (t + 1) 1 i j = p_{ij} X p s t 1 i j)

where the first conjunct states that this model is a DMRM, the second condition constrains that the reward space is a finite space, the third one ensures the rewards are constant for every state x in the state space of the random variable (X t) and the last conjunct refers to the time-homogeneity of the transition probabilities of the underlying DTMC.

If the underlying DTMC of a DMRM is an aperiodic DTMC, then the conditional expectations are convergent. This property can be verified as follows:

Theorem 9 (Convergent Property).

$\vdash \forall$ X Y p s_x s_y p_0 p_{ij} r_{xy} i.
th_dmrm X Y p s_x s_y p_0 p_{ij} r_{xy} $\wedge$ APERIODIC_MC X p s_x p_0 p_{ij} $\Rightarrow$
convergent (λt. cond_expec (Y t) (X t) i p s_y)

The expected cumulated reward over a long period is always of interest as the cumulative property verified in the following theorem, which can be used to obtain the expected steady-state reward.

Theorem 10 (Cumulative Property).

$\vdash \forall$ X Y p s_x s_y p_0 p_{ij} r_{xy} i.
th_dmrm X Y p s_x s_y p_0 p_{ij} r_{xy} $\wedge$ APERIODIC_MC X p s_x p_0 p_{ij} $\wedge$
i $\in$ space s_x $\Rightarrow$
(lim (λt. cond_expec (Y t) (X t) i p s_y) = lim (λt. r_{xy} t i))

The expected steady-state reward can be achieved by applying the following theorem:

Theorem 11 (Expected Steady-state Reward).

$\vdash \forall$ X Y p s_x s_y p_0 p_{ij} r_{xy} i.
th_dmrm X Y p s_x s_y p_0 p_{ij} r_{xy} $\wedge$ APERIODIC_MC X p s_x p_0 p_{ij} $\wedge$
i $\in$ space s_x $\Rightarrow$
(lim (λ t. expec (X t) p s_x) =
$\sum_{y \in \text{space } s_y}$ lim (λ t. r_{xy} t i) lim (λ t. $\mathbb{P}\{$x | (Y t) x = y$\}$))

The HOL script of these formalizations is available in [9] and the verified theorems are used in the next section to analyze the memory contention problem of a particular multi-processor system.

4 Application

In this section, we present a formal performance analysis of a multiprocessor system by reasoning about the expectation of memory access requests.

4.1 Memory Contention Problem

Consider a multi-processor system with two memory modules and two processors. This system can be modeled as a Discrete-time Markov Reward Model (DMRM) [19], depicted in Figure 2, by assuming that access time of any memory module is a constant and all the memory modules are synchronized. The states of the system are denoted by the pair (i, j), where i represents the number of the processors waiting for the memory module 1 and j refers to the amount of the processors waiting for the memory module 2. Due to the fact that memory access time is always longer than any other data transaction of the processor, it is reasonable to assume that $0 \leq i$, $0 \leq j$, and $i + j = 2$ in every memory cycle. Thus, the states set {(1, 1), (0, 2), (2, 0)} provides all the possible states of the given system. Also, q_k $(k = 1, 2)$ represents the probabilities that a processor requests a direct memory access. If both processors are accessing two different memory modules (in this case, the system stays in state (1, 1)) and will complete the task by the end of this memory cycle, then the expectation of the number of memory requests completed in this memory cycle is 2. If there are two requests to access memory module 1 in a memory cycle, then only one request can be completed in this memory cycle. We can obtain the same expectation when memory module 2 is requested to be accessed. We denote the random variable Y as the number of requests completed in every memory cycle in the steady state and the request state space is the set {0, 1, 2}. The conditional expectations of Y can be mathematically described as:

$$\begin{aligned} &\text{E[Y|system in state (1,1)]} = 2;\\ &\text{E[Y|system in state (2,0)]} = 1; \quad (2)\\ &\text{E[Y|system in state (0,2)]} = 1. \end{aligned}$$

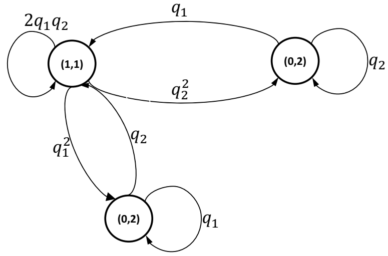

Fig. 2. The State Diagram for the Memory Interference Problem

In order to analyze the performance of such a system, we are interested in learning the steady probabilities of the states, in which the memory modules are efficiently used, and the expected number of memory requests satisfied in each memory cycle in the steady state.

4.2 Formalization of Memory Contention Problem

To formally analyze the properties of this system, we first describe this multiprocessor system in HOL. As shown in Figure 2, this kind of system can be described as a DMRM with an aperiodic and irreducible DTMC [19]. First of all, we define the state space for the requests as a general function in HOL:

Definition 13 (Request State Space).

```
⊢ ∀ n. request n = {(r:real) | r ∈ [0, n]}
⊢ ∀ n. request_space n = (request n, POW (request n))
```

where variable `n` refers to the number of memory modules in the system and `POW (request n)` is the sigma algebra of the request set. In the case of the two-processor system, at most two requests can be created in a memory cycle, thus, $n = 2$.

Now, the system state space and the transition probabilities can be formally expressed as the functions presented in Definition 14 and the conditional expected value is described as a function in Definition 15 using higher-order logic.

Definition 14 (State Space & Transition Probabilities).

```
⊢ sys_state = {(0, 2); (2, 0); (1, 1)}
⊢ sys_space = (sys_state, POW sys_state)
⊢ ∀ q1 q2 t i j. Lt q1 q2 t i j = case (i,j) of
     ((1,1),(1,1)) → 2q1q2 | ((0,2),(1,1)) → q1   |
     ((2,0),(1,1)) → q2    | ((1,1),(0,2)) → q2^2 |
     ((0,2),(0,2)) → q2    | ((2,0),(2,0)) → q1   |
     ((1,1),(2,0)) → q1^2  | (_,_) → 0
```

where `sys_space` is a pair, in which the first element is a set `sys_state` and the second element is the sigma algebra of `sys_state`, the function `Lt` returns the transition probabilities.

Definition 15 (Conditional Expected Requests).

```
⊢ ∀ t i j. rewards t (i, j) =
    if (i, j) = (1, 1) then 2 else
    if (i, j) = (2, 0) then 1 else
    if (i, j) = (0, 2) then 1 else 0
```

where the function `rewards` corresponds to Equation (2).

These functions can now be used to model the multiprocessor system of Figure 2 as follows:

Definition 16 (Multiprocessor Model).

$\vdash$ $\forall$ X Y p q_1 q_2 p_0. opera_sys_model X Y p q_1 q_2 p_0 =
th_dmrm X Y p sys_space (request_space 2) p_0 (Lt q_1 q_2) rewards $\wedge$
Aperiodic_DTMC X p sys_space p_0 (Lt q_1 q_2) $\wedge$
Irreducible_DTMC X p sys_space p_0 (Lt q_1 q_2) $\wedge$
$0 < q_1 \wedge 0 < q_2 \wedge q_1 < 1 \wedge q_2 < 1 \wedge (q_1 + q_2 = 1)$

where variable X indicates the system state (the pair containing the number of requests for each memory module) at discrete time points, variable Y refers to the requests, which is a random variable, p denotes the probability space, q_1 and q_2 are the parameters in the transition probabilities described previously, and function p_0 represents a general initial distribution, the request state space is request_space and the system state space is sys_space, which are defined in Definition 13 and 14, respectively.

Note that, the definitions presented above provide the flexibility on modifying the argument, i.e., n in Definition 13, or the functions in Definitions 14 and 15 in case of describing more complex systems.

4.3 Performance Analysis of Memory Contention

As the underlying DTMC in the model described in Definition 16 is an aperiodic and irreducible DTMC, we can directly apply Theorem 6 to prove that for all states in the system state space, the probability distributions are convergent in the long-term as the following theorem.

Theorem 12 (Convergence of the State Distribution).

$\vdash$ $\forall$ X Y p q_1 q_2 p_0 i.
opera_sys_model X Y p q_1 q_2 p_0 $\wedge$ i $\in$ space sys_space $\Rightarrow$
convergent (λ t. $\mathbb{P}\{$x | X t x = i$\}$)

Applying Theorems 2, 3, 5 and 6, we obtain the steady-state probabilities (the limit of the probability mass functions for all states in the state space):

Theorem 13 (Steady Probabilities).

$\vdash$ $\forall$ X Y p q_1 q_2 p_0.
opera_sys_model X Y p q_1 q_2 p_0 $\Rightarrow$
$\lim_{t\to\infty} \mathbb{P}\{x \mid X\ t\ x = (2, 0)\} = \frac{q_1^3}{1-2q_1q_2}$ $\wedge$
$\lim_{t\to\infty} \mathbb{P}\{x \mid X\ t\ x = (0, 2)\} = \frac{q_2^3}{1-2q_1q_2}$ $\wedge$
$\lim_{t\to\infty} \mathbb{P}\{x \mid X\ t\ x = (1, 1)\} = \frac{q_1q_2}{1-2q_1q_2}$

Utilizing the formalizations of expectation presented in Section 3.1, we can prove the expectation of the number of memory requests completed per memory cycle in the steady state in the following theorem:

Theorem 14 (Expected Steady-state Rewards).

$\vdash \forall$ `X Y p` q_1 q_2 p_0. `opera_sys_model X Y p` q_1 q_2 $p_0 \Rightarrow$
$\lim_{t\to\infty}$ `(`λ `t. expec (Y t) p request_space) =` $\frac{1-q_1 q_2}{1-2q_1 q_2}$

Theorems 13 and 14 can be used for optimizing the system design. For example, we can obtain the maximum value of the expectation of completed requests from Theorem 14 and find out the conditions to achieve the best efficiency ($q_1 = q_2 = 1 / 2$). Similarly, when $q_1 = 0.97$ and $q_2 = 0.03$, we can obtain the steady-state probability $\lim_{t\to\infty} \mathbb{P}\{x \mid X\ t\ x = (0, 2)\} = 2.8669e^{-5}$ by applying Theorem 13, however, classical simulators, such as Matlab, compute $\lim_{t\to\infty} \mathbb{P}\{x \mid X\ t\ x = (0, 2)\} = 0$ due to the underlying algorithms for accelerating the convergent speed and the round-off error in the intermediate steps. Moreover, the algorithms can never provide a positive transition probability matrix, which exists according to Theorem 5, because of the round-off errors or the slow convergent speed. Our approach can overcome all these problems and provide accurate results.

Our general definition of DMRMs offers the flexibility of describing the states as arbitrary types, such as the pairs in this application, instead of the abstract non-negative integers. On the other hand, this application illustrates an approach to formally analyze the distributed systems using theorem proving. It is important to note that the system can be more complex (i.e., the number of the processors and memory modules can be very large), and we can analyze it by defining new functions, such as `sys_space`, `request_space`, `Lt` and `rewards`.

The proof script for modeling and verifying the properties of the memory contention in a multiprocessor (two processors and two memory modules) is about 700 lines long and is available in [9]. The ability to formally verify theorems involving DMRMs and the short script clearly indicates the usefulness of the formalization, presented in the previous sections in this paper, as without them the reasoning could not have been done in such a straightforward way.

5 Conclusion

This paper presents a method to formally analyze the performance of multiprocessor systems based on the formalization of Discrete-time Markov Reward Models (DMRMs) using higher-order logic. Due to the inherent soundness of theorem proving, our work guarantees to provide accurate results, which is a very useful feature while analyzing stationary behaviors and long-term expectation on certain key measures for a system associated with safety or mission-critical systems. In order to illustrate the usefulness of the proposed approach, we formally analyzed the memory contention problem in a system with two processors and two memory modules, which is modeled as a DMRM with the underlying aperiodic and irreducible DTMC, using the formalizations of DTMCs. Our results exactly matched the results obtained using paper-and-pencil analysis in [19], which ascertains the precise nature of the proposed approach.

As DMRMs have been widely applied in performance and reliability analysis, especially in predicting the reliability for fault-tolerant systems and software, the presented work opens the door to a new and promising research direction on formally analyzing the Discrete-time Markov Reward Models. We plan to apply the formalization presented in this paper to formally analyze some real-world systems modeled as DMRMs. Also, we plan to extend our work to the Continuous-time Markov Reward Models (CMRMs) and Markov Decision Process (MDP), which will enable us to formally analyze software reliability and hardware performance of a wider range of systems.

References

1. Baier, C., Katoen, J.: Principles of Model Checking. MIT Press (2008)
2. Bhattacharya, R.N., Waymire, E.C.: Stochastic Processes with Applications. John Wiley & Sons (1990)
3. Chung, K.L.: Markov Chains. Springer (1960)
4. Cloth, L., Katoen, J., Khattri, M., Pulungan, R.: Model checking Markov Reward Models with Impulse Rewards. In: International Conference on Dependable Systems and Networks, pp. 722–731. IEEE Computer Society Press (2005)
5. Gamerman, D., Lopes, H.F.: Fundamentals of Applied Probability Theory. Chapman & Hall/CRC (2006)
6. Goldberg, R.R.: Methods of Real Analysis. Wiley (1976)
7. Hurd, J.: Formal Verification of Probabilistic Algorithms. PhD Thesis, University of Cambridge, UK (2002)
8. Kwon, Y., Agha, G.: A Markov Reward Model for Software Reliability. In: International Parallel and Distributed Processing Symposium, pp. 1–6. IEEE (2007)
9. Liu, L.: HOL Script: Formal Analysis of Memory Contention in a Multiprocessor System (2013), `http://hvg.ece.concordia.ca/projects/prob-it/dmrm.html`
10. Liu, L., Hasan, O., Tahar, S.: Formal Reasoning About Finite-State Discrete-Time Markov Chains in HOL. Journal of Computer Science and Technology 28(2), 217–231 (2013)
11. Liu, L., Hasan, O., Aravantinos, V., Tahar, S.: Formal Reasoning about Classified Markov Chains in HOL. In: Blazy, S., Paulin-Mohring, C., Pichardie, D. (eds.) ITP 2013. LNCS, vol. 7998, pp. 295–310. Springer, Heidelberg (2013)
12. Mhamdi, T., Hasan, O., Tahar, S.: Formalization of entropy measures in HOL. In: van Eekelen, M., Geuvers, H., Schmaltz, J., Wiedijk, F. (eds.) ITP 2011. LNCS, vol. 6898, pp. 233–248. Springer, Heidelberg (2011)
13. MRMC (2013), `http://www.mrmc-tool.org/trac/`
14. Norris, J.R.: Markov Chains. Cambridge University Press (1999)
15. Parker, D.A.: Implementation of Symbolic Model Checking for Probabilitics Systems. PhD Thesis, University of Birmingham, Birmingham, UK (2002)
16. PRISM (2013), `www.prismmodelchecker.org`
17. Sczittnick, M.: MACOM - A Tool for Evaluating Communication Systems. In: International Conference on Modelling Techniques and Tools for Computer Performance Evaluation, pp. 7–10 (1994)
18. Tijms, H.C.: A First Course in Stochastic Models. Wiley, New York (2003)
19. Trivedi, K.S.: Probability and Statistics with Reliability, Queuing, and Computer Science Applications. John Wiley & Sons (2002)
20. Yates, R.D., Goodman, D.J.: Probability and Stochastic Processes: A Friendly Introduction for Electrical and Computer Engineers. Wiley (2005)