PAPER Special Section on Formal Approach

Probabilistic Model Checking of the One-Dimensional Ising Model

Toshifusa SEKIZAWA[†,††a)], ***Nonmember***, **Tatsuhiro TSUCHIYA**[†], ***Member***, **Koichi TAKAHASHI**[††], ***Nonmember***, ***and*** **Tohru KIKUNO**[†], ***Fellow***

SUMMARY Probabilistic model checking is an emerging verification technology for probabilistic analysis. Its use has been started not only in computer science but also in interdisciplinary fields. In this paper, we show that probabilistic model checking allows one to analyze the magnetic behaviors of the one-dimensional Ising model, which describes physical phenomena of magnets. The Ising model consists of elementary objects called spins and its dynamics is often represented as the Metropolis method. To analyze the Ising model with probabilistic model checking, we build Discrete Time Markov Chain (DTMC) models that represent the behavior of the Ising model. Two representative physical quantities, i.e., energy and magnetization, are focused on. To assess these quantities using model checking, we devise formulas in Probabilistic real time Computation Tree Logic (PCTL) that represent the quantities. To demonstrate the feasibility of the proposed approach, we show the results of an experiment using the PRISM model checker.

key words: *verification, probabilistic model checking, the Ising model, Discrete Time Markov Chain*

1. Introduction

Model Checking [1] is a powerful technique for the automatic verification of hardware and software systems. The technique has successfully been applied to verify many systems. *Probabilistic model checking* is an extension of conventional model checking to probabilistic systems and has been studied for more than two decades [2]–[4]. Examples of the applications of this technique include: randomized distributed algorithms, distributed agreement protocol [5], wireless LAN [6], etc. Early studies of probabilistic model checking were mainly aimed at reasoning about asymptotic behaviors when time goes to infinity, while recent work was aimed at verifying qualitative properties. Many of these studies are based on Markov chains [7], which are well-known random processes extensively used in modeling and analyzing probabilistic systems. A recent survey of probabilistic model checking can be found in [8]. Several probabilistic model checkers have been implemented and are now publicly available. Examples include PRISM [9] and MRMC [10].

Manuscript received July 24, 2008.
Manuscript revised November 20, 2008.

[†]The authors are with the Graduate School of Information Science and Technology, Osaka University, Suita-shi, 565–0871 Japan.
[††]The authors are with Research Center for Verification and Semantics (CVS), National Institute of Advanced Industrial Science and Technology (AIST), Toyonaka-shi, 560–0083 Japan.
a) E-mail: sekizawa-t@aist.go.jp

DOI: 10.1587/transinf.E92.D.1003

As many phenomena seen in fields other than computer science also exhibit probabilistic behavior, the use of probabilistic model checking has been started in an interdisciplinary manner. Analyses of molecular systems [11] and systems biology [12] are such examples.

In this study, we show that probabilistic model checking allows one to analyze magnetic dynamics. We deal with the Ising model which is a simple model that describes physical magnetic dynamics. Normally, the Ising model is analyzed using computer simulations. The difference between probabilistic model checking and computer simulation is that model checking explores all possibilities, whereas computer simulation is based on random sampling. One advantage of probabilistic model checking is that various physical properties are expressible in temporal logics, and these properties can be verified automatically. We demonstrate that our approach can be used to verify physical properties in various ways, including reachability checking, computing probability, and computing transition time.

The main contribution of this study is the application of probabilistic model checking to the analysis of magnetic fields. To the best of our knowledge, our work is the first attempt to investigate such an application*. In recent years, as stated above, probabilistic model checking has been applied to fields other than computer science. However, such applications are still few and have not been sufficiently studied. Our work is aimed at broadening the use of model checking in this direction.

The roadmap of this paper is as follows. In Sect. 2, we introduce the Ising model. Next in Sect. 3, we build Discrete Time Markov Chain (DTMC) models of the one-dimensional Ising model. In Sect. 4, we verify some physical properties using PRISM, a probabilistic model checker, to show that probabilistic model checking has potential for analyzing such probabilistic behaviors.

2. The Ising Model

Physics deals with nature to understand fundamental principles underlying phenomena. Magnet behavior is a widely known physical phenomenon. However its mechanism is quite difficult to understand because its macroscopic behavior is caused by a complex system. In fact, "real" magnet behaviors are observed as a result of cooperative phenom-

*The preliminary results of our work were presented in [13].

Copyright © 2009 The Institute of Electronics, Information and Communication Engineers

ena of microscopic factors. For example, electrons' spin act as a representative factor of magnetization of iron. Although much research has been devoted to understanding the mechanism through direct observation, there is also a different approach that uses simplified models.

The *Ising model* [14], [15] is such a simplified model for magnets named after a physicist, Ernst Ising, who proposed the model in 1924. The Ising model is defined on a collection of elementary objects called *spins*. Each spin is located at a site of a lattice and can have only one of two values; +1 (named *up*) or −1 (named *down*). A collection of states of all spins is said to be a *configuration*. The energy of the Ising model is defined as a function of the state of spins. Interactions among spins are a fundamental element of energy.

In the standard form of the Ising model, interactions among spins are restricted to nearby spins. The energy E is defined as the sum of two terms: 1) the sum of interactions of spins, and 2) the interaction of spins with an external magnetic field, that is,

$$E = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j - H \sum_k \sigma_k,$$

where σ_i is the value of the spin at the i-th site in the lattice, J is the interaction coefficient, H is the external magnetic field, and $\langle i, j \rangle$ denotes the interaction of two spins, σ_i and σ_j, located at nearby lattice sites.

When the interaction coefficient is negative, the system is called *anti-ferromagnetic*. Anti-ferromagnetic interaction tends to anti-align spins, that is, nearby spins become different in value. The energy of an anti-ferromagnet decreases if nearby spins have different values. In contrast, when the interaction coefficient is positive, the systems is called *ferromagnetic*. The energy of a ferromagnet decreases if spins are aligned and vice versa.

In the one-dimensional (1D) Ising model, spins are located at sites of a line. We assume that a model consists of n spins $\sigma_1, \sigma_2, \ldots, \sigma_n$ which are located in order at sites of a line, and that there is no external magnetic field. We also assume the periodic boundary condition such that $\sigma_{n+1} = \sigma_1$. Figure 1 shows a 1D Ising model with a periodic boundary condition for n spins. We express the energy E as

$$E(\sigma_1, \sigma_2, \ldots, \sigma_n) = -J \sum_{i=1}^{n} \sigma_i \sigma_{i+1}.$$

The intended purpose of the Ising model is to elucidate the phenomenon of magnetism. Magnetization is a physical quantity that represents the material's own magnetic field.

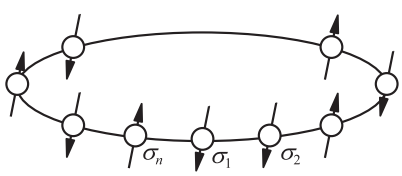

Fig. 1 One-dimensional Ising model with periodic boundary condition.

The magnetization M of the 1D Ising model is defined as

$$M(\sigma_1, \sigma_2, \ldots, \sigma_n) = \sum_{i=1}^{n} \sigma_i.$$

The Ising model itself does not determine its dynamics. The Metropolis method [16] serves as a standard dynamics model. The validity of the Metropolis method is elaborated in [15]. The Metropolis method is an example of the Monte Carlo method, a generic technique for simulating the stochastic behavior of various systems. The Metropolis method randomly selects an initial state and repeats probabilistic transitions depending on temperature. A sufficiently large number of transitions result in a thermal equilibrium. When applied to the Ising model, this method is called *random spin flipping*. In random spin flipping, probabilistic transitions are performed as described below:

1. Choose a spin σ_i at random for each flipping.
2. Evaluate the change in energy $\Delta E = E_{flipped} - E_{current}$ caused by spin flipping from σ_i to $\sigma_i' = -\sigma_i$, where $E_{current} = E(\sigma_1, \ldots, \sigma_i, \ldots, \sigma_n)$, and $E_{flipped} = E(\sigma_1, \ldots, \sigma_i', \ldots, \sigma_n)$.
3. If $\Delta E \leq 0$, spin flipping is accepted. Otherwise, the spin flipping is accepted with a probability $e^{-\Delta E/T}$, where T is the temperature.
4. Repeat steps 1 to 3 a sufficient number of times.

As can be seen, temperature T plays an important role in spin flipping. That is, when the temperature is higher, a spin has more chances of flipping. Note that the next configuration obtained by spin flipping only depends on current configuration, i.e., it is independent of past configurations. This independence is called the *memoryless property*. We call a series of n times of spin flipping an *update*.

One might think that a continuous model is better suited for examining magnetic fields. Indeed, one of the continuous models for magnetism is the Heisenberg model [17]–[19], where each spin has a continuous value. Such a model could be mechanically analyzed using a hybrid system consisting of finite discrete states and whose dynamics are specified by differential equations over continuous variables. We however do not consider hybrid systems, because i) the Ising model is widely accepted as an adequate model of magnetic fields and ii) in general, hybrid systems are more difficult to mechanically analyze than discrete systems.

3. Probabilistic Model Checking

Model checking is an automatic formal verification technique. Probabilistic model checking extends model checking to probabilistic systems. Given a model that represents a system under consideration, model checking automatically determines whether or not the model satisfies a given specification by exhaustively searching for the state space of the model.

3.1 Discrete Time Markov Chain

We consider systems represented as Discrete Time Markov Chains (DTMCs). The definition of a DTMC is as follows. Let AP be a set of atomic propositions. A DTMC is a quadruple $\mathcal{M} = (S, s^i, \mathcal{T}, \mathcal{L})$, where S is a finite set of states, $s^i \in S$ is the initial state, $\mathcal{T} : S \times S \to [0, 1]$ is a transition probability function such that $\forall s \in S,\ \sum_{s' \in S} \mathcal{T}(s, s') = 1$, and $\mathcal{L} : S \to 2^{\mathtt{AP}}$ is a labeling function. The current state $s \in S$ at computational time t has a transition to state $s' \in S$ at $t+1$ with probability $\mathcal{T}(s, s')$. A *path* is a sequence of states. The probability of a path $s_0, s_1, \cdots$ is $\prod_{i \geq 0} \mathcal{T}(s_i, s_{i+1})$. DTMC has the *Markov property*; that is, if we choose a state s at a computational time t then the next state at $t+1$ depends only on the current state and independent of the past states.

In a DTMC, time is modeled as discrete steps. Therefore, a 1D Ising model with spin flipping is easily transformed into a DTMC by representing configurations as states and individual spin flippings as probabilistic transitions [15]. Note that the Ising model satisfies the Markov property, because the spin flipping algorithm is memoryless.

3.2 Probabilistic Real Time Computation Tree Logic

Probabilistic real time Computation Tree Logic (PCTL) [20] is a probabilistic extension of the temporal logic Computation Tree Logic (CTL). It is interpreted on a DTMC. Using PCTL formulas, we can describe physical specifications. The syntax of PCTL is defined as follows:

$$\begin{aligned} \varphi ::= {} & \top \mid \bot \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \to \varphi \\ & \mid \mathbb{P}_{\sim\lambda}(\psi) \\ \psi ::= {} & \varphi \ \mathtt{U}\ \varphi \mid \varphi \ \mathtt{U}^{\leq t}\ \varphi \end{aligned}$$

where p is an atomic proposition in AP, $\sim \in \{<, \leq, \geq, >\}$ is a relational operator, $\lambda \in [0, 1]$ is a probability, and t is a nonnegative integer or infinity. Intuitively, *state formula* φ represents the conditions of states, and *path formula* ψ represents the conditions of paths, which are sequences of states. The symbols $\top, \bot, \neg, \vee, \wedge$ and $\to$ have their usual meanings. The symbol U is the "until" operator, and the symbol $\mathtt{U}^{\leq t}$ is the "bounded until" operator. Intuitively, the formula $\varphi_1 \ \mathtt{U}^{\leq t} \varphi_2$ expresses that φ_2 holds at some point within t computational steps, and φ_1 holds until the point. The formula $\mathbb{P}_{\sim\lambda}(\psi)$ is evaluated to be true with respect to a given state v if and only if ψ holds for a path starting from v with a probability of at least or at most λ. For example, the specification that "the probability that a stable state is reached within 10 steps is greater than 0.3" is written as the PCTL formula $\mathbb{P}_{>0.3}\left(\top \ \mathtt{U}^{\leq 10} \mathit{stable}\right)$.

3.3 PRISM

In this work, we use a probabilistic model checker PRISM [9], [21] which takes a DTMC as a model and PCTL formulas as specifications. PRISM checks the truth value of the PCTL formula with respect to the initial state. When no initial state is specified, PRISM checks whether the PCTL formula holds for all states in the given model. In such a case, the output of PRISM will be "true" if and only if the property holds for all states.

In addition to model checking against PCTL specifications, PRISM supports additional extended features. One extension is actual probability measurement which is supported by the PRISM language expression P=?[ψ]. For the formula P=?[ψ], PRISM calculates the actual probability that ψ holds in the model. Another useful feature is *transition rewards*, which are nonnegative values associated with certain transitions. When taking a transition from state v to state v' with a transition reward r, the reward r is earned. Through a sequence of transitions, earned rewards are accumulated. Such accumulated transition rewards allow us quantitative evaluations relating to model behaviors. The PRISM language expression of *reachability reward* R=?[F φ {v}] returns the expected value of rewards accumulated during a sequence of transitions that occur from a state v to a state where φ holds for the first time.

In this work, we use PRISM 3.2 beta 1. As the version number suggests, this version of PRISM is still beta released. However, it provides some useful enhancements. We decided to use this version to achieve better performance.

4. Model Checking the Ising Model

In this section, we describe how we represent and analyze the Ising model using PRISM.

4.1 Building the 1D Ising Model on PRISM

Here, we show how to build a DTMC model for the 1D Ising model using PRISM. In the following, we assume that the 1D Ising model consists of n spins $\sigma_1, \sigma_2, \ldots, \sigma_n$ which are placed on a line in this order, that the periodic boundary condition $\sigma_{n+1} = \sigma_1$ holds, that the interaction coefficient $J = -1$, i.e., the Ising model is anti-ferromagnetic, and that there is no external magnet field, i.e., $H = 0$.

The temperature T is fixed in a DTMC model of PRISM (PRISM model, for short). Hence, the temperature dependence can be analyzed by constructing a set of PRISM models that have different temperatures.

One PRISM model consists of n modules, each of which represents a spin. Then, the state of DTMC is $(\sigma_1, \sigma_2, \ldots, \sigma_n)$. Because PRISM allows no integer variables to be negative, we represent the value of spin by 0 and 1, instead of -1 and $+1$. Figure 2 shows the relationship between a configuration and a DTMC state represented by PRISM modules. Using the value of a spin, we define E^P and M^P as the energy and magnetization of the PRISM model as follows: $E^P = (E + n)/2$ and $M^P = (M + n)/2$. These linear transformations change the range of the quantities from $[-n, n]$ to $[0, n]$. It is clear that these transformations preserve a one-to-one correspondence. We assume

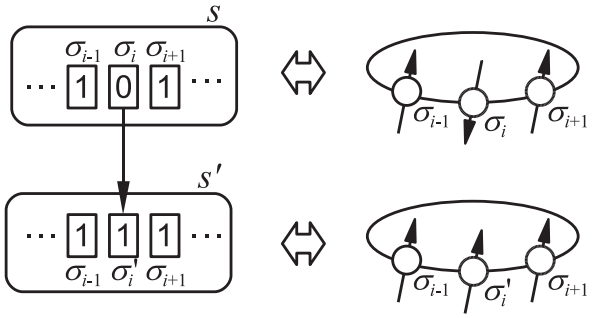

Fig. 2 DTMC states and corresponding configurations. The value of a spin is represented by a PRISM module.

that all spins are down in the initial state. It follows that the energy $E^P = n$ and the magnetization $M^P = 0$ in the initial state.

The behavior of the random spin flipping algorithm is specified as follows. In PRISM semantics, one of the modules is nondeterministically selected and is executed in each step. This allows the non-deterministic choice of a spin in Step 1 of the algorithm. The decision process of Steps 2 and 3, which determines whether or not to flip the spin, is specified using case-splitting with respect to the selected spin and its two nearby spins. The calculation of the energy difference ΔE is implicitly performed by this case-splitting. For example, if all the three spins have a value of 1, then $\Delta E < 0$ always holds, in which case the spin flipping is accepted with a probability of 1.

In our model, both the energy E^P and the magnetization M^P are declared using a PRISM language "formula", which is a shorthand expression to prevent code duplication. We can use these "formula" names in any places of a model or properties. We also assign a transition reward of 1 to every transition, so that the accumulated reward coincides with the total number of random spin flipping judgments.

We show a fragment of the PRISM code for the 1D Ising model in Appendix as an example. Readers are referred to Appendix for details.

4.2 Verifying Basic Specifications

To show the usefulness of our model checking approach, here we demonstrate how one can verify important physical properties by reachability checking, computing probabilities, and computing transition time. In this subsection, unless otherwise noted, we fix the number of spins n to 12 and fix temperature T to 1.0.

Here, we focus our discussion on *equilibria*. Equilibrium is a balanced state in which macroscopic physical quantities do not change. Equilibrium is an important condition when studying physical phenomena, because it represents the stable condition of the observed system. At equilibrium, the energy is minimized because of the principle of minimum energy, and it is well-known that spontaneous magnetization becomes zero. Therefore, $E^P = 0$ and $M^P = n/2$ hold if and only if the state is at equilibrium.

We define the formula of *equilibrium* as follows:

$$equilibrium := E^P = 0 \wedge M^P = n/2$$

In PCTL formulas, we use *equilibrium* to represent the set of states in which *equilibrium* holds. Defining *equilibrium* using the keyword `label` allows us to do this in the PRISM language for PCTL formulas.

Reachability Checking

We consider a specification, namely, the reachability of an equilibrium from an arbitrary state. The following PCTL formula expresses that equilibrium is reachable through paths from a given state:

$$\mathbb{P}_{\geq 1}\left[\top \,\mathtt{U}\, equilibrium\right]$$

As stated in Sect. 3, PRISM simultaneously evaluate the formula with respect to all states. The verification result obtained was true. This result ensures that equilibrium is always reachable from every state in the model.

The above formula asserts reachability to equilibrium. We next verify whether the equilibrium lasts with high probability even if the system fluctuates. To do this we introduce the notion of the *neighborhood of equilibrium*. We define the neighborhood of equilibrium as states in which the differences in energy and magnetization from equilibrium are $\Delta E^P \leq 2$ and $\Delta M^P \leq \pm 1$ respectively, i.e., the states where $0 \leq E^P \leq 2$ and $5 \leq M^P \leq 7$ hold. The following PCTL formula shows that "equilibrium is reachable from an arbitrary state, and after reaching equilibrium, the system stays in the neighborhood of equilibrium within 100 times of spin flipping with a probability of more than 70%."

$$\mathbb{P}_{\geq 1}\left[\top \,\mathtt{U}\, (equilibrium \wedge \psi_{\mathrm{in}})\right]$$

where $\psi_{\mathrm{in}} = \mathbb{P}_{\leq 0.3}\left[\chi_{\mathrm{lhs}} \,\mathtt{U}^{\leq 100}\, \chi_{\mathrm{rhs}}\right]$, $\chi_{\mathrm{lhs}} = (E^P \leq 2) \wedge (5 \leq M^P \wedge M^P \leq 7)$, and $\chi_{\mathrm{rhs}} = (2 < E^P) \vee (M^P < 5 \vee 7 < M^P)$. The result of verification indicates that this specification holds for all states. Note that the probability 0.3 is required to be constant from the definition of PCTL.

Computing Probability

Observable physical phenomena often show probabilistic behaviors. To understand such systems, actual values of probabilities give us much information. Therefore it is important to establish a method of obtaining such probabilities.

As an illustrative example, here, we consider the calculation of the probability $p(T, n)$ of transitions from equilibrium to a state in which the largest magnetization $M^P_{\mathtt{max}} = n$ holds. For each transition, M^P can be changed by $+1$, 0 or -1. Therefore, it takes $n/2$ transitions if we look at a direct transition to reach $M^P_{\mathtt{max}}$ from $M^P = n/2$. In this case, it is already known that the probability $p(T, n)$ is calculated analytically as follows:

$$p(T, n) = \frac{(n/2)!}{n^{n/2}} \left(e^{-4/T}\right)^{n/2}.$$

Table 1 Energy difference ΔE caused by a spin flip.

σ_{i-1}	σ_i	σ_{i+1}	σ_{i-1}	σ'_i	σ_{i+1}	ΔE
↓	↓	↓	↓	↑	↓	+4
↓	↓	↑	↓	↑	↑	0
↓	↑	↓	↓	↓	↓	−4
↓	↑	↑	↓	↓	↑	0
↑	↓	↓	↑	↑	↓	0
↑	↓	↑	↑	↑	↑	−4
↑	↑	↓	↑	↓	↓	0
↑	↑	↑	↑	↓	↑	+4

In this formula, a constant value 4 appears as ΔE in the exponent. This is formed as a result of the normalization of the interaction coefficient J to -1. When this normalization is applied, the definition of the energy E, given in Sect. 2, yields Table 1 showing the energy difference caused by a spin flip. In this table σ_i is the spin flipped and σ'_i represents the new value caused by the flip. In random spin flipping, the probability of accepting a flip becomes less than one only when the energy difference ΔE is greater than zero. In such a case, ΔE is always 4.

To compute the actual probability of the specific transition, we use the following formula written in the PRISM language:

```
P=? [ true U<=6 (MP=12) {"equilibrium"} ]
```

In this formula, {"equilibrium"} means the initial state where $E^P = 0 \wedge M^P = n/2$ holds. If two or more states satisfy the initial condition, PRISM automatically chooses a state as a starting point for verification.

The analytic solution of $p(1.0, 12)$ is 9.10300×10^{-15}. The probability computed exactly matched the analytic solution in the calculation error range. This agreement of the two probabilities suggests us that the PRISM model correctly represents the Ising model. The correctness of a model can be confirmed by similar computations. Once the model is confirmed to be correct, we can reliably compute various probabilities even if they are hard to solve analytically.

Computing Transition Time

Using transition rewards, one can measure the expected time required for the system to move from state to state. Recall that we assigned a transition reward of 1 to every transition. The expected value of rewards can be considered as the expected transition time, since one judgment of spin flipping corresponds to a transition and a passage of discrete computational time. In the 1D Ising model, the spontaneous magnetization is zero in the absence of an external magnetic field. However, temperature contributes to magnetization fluctuation. To understand the behavior, we measure the expected time of changing magnetization starting from the stable magnetization $M^P = 6\ (= n/2)$ to $M^P \neq 6$. One can instruct PRISM to compute the expected time by specifying in the PRISM language as follows:

```
R=? [ F (MP=VAL) {"equilibrium"} ]
```

where `VAL` is an integer target magnetization which varies from 0 to n. For each integer value in `VAL`, PRISM returns the evaluated expected transition time. We computed the expected transition times at fixed temperatures: 1.0, 1.5, 2.0 and 3.0. Figure 3 shows the results. The X-axis is the normalized magnetization $M = (2M^P - n)/n$. The Y-axis is the normalized transition time R/n. This is because n times of spin flipping correspond to one update. Because of the obvious symmetry of the Ising model, we only show the result of $T = 3.0$ for $M < 0$.

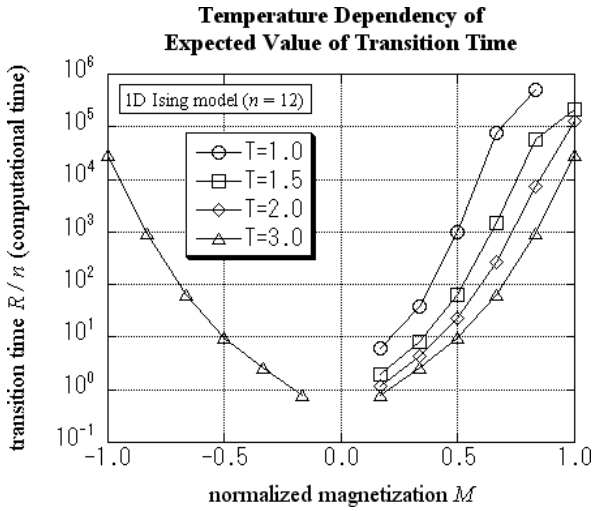

Fig. 3 Temperature dependency of expected value of transition time against magnetization ($n = 12$).

Some characteristics are observable in Fig. 3. One characteristic is the temperature dependence. As we can see, the expected transition time increases as temperature decreases. This is natural, because the probability of spin flipping is lower when the temperature is lower, and it takes much time to transit to excited states where the system has a higher magnetization.

Computing the expected transition time is a common practice for observing how a system behaves from a fixed initial state. In physics, it is common to analyze physical properties based on an equilibrium or a ground state where the system has the lowest energy. Therefore, this approach can be applied to many analyses that focus on the differences in quantities between states. We can also see a symmetric behavior that is induced by the fact that the Ising model has a symmetric structure. Unnecessary verification can be avoided using such a symmetry, as we show in Fig. 3.

Table 2 shows the time required for constructing the model (4096 states and 52924 transitions) and the time required for the three verifications discussed in this subsection. The measurement was performed using Windows XP Professional on a computer equipped with an Intel Centrino U1300 1.06 GHz processor and a 1.5 GB RAM.

In Table 2, the elapsed time for computing transition time increases as the temperature T decreases for a fixed M^P. As a termination criterion, PRISM checks that the result has converged sufficiently. This explains why the expected transition time increases when temperature de-

Table 2 Elapsed time for constructing a model, a verification, and computations.

	time (seconds)			
model construction	0.047			
reachability	0.016			
probability	0.354			
transition time	$T = 1.0$	$T = 1.5$	$T = 2.0$	$T = 3.0$
$M^P = 7$	0.125	0.156	0.110	0.093
$M^P = 8$	0.625	0.282	0.219	0.218
$M^P = 9$	12.14	1.969	0.984	0.580
$M^P = 10$	388.0	28.17	8.531	3.093
$M^P = 11$	1041	405.5	116.6	28.81
$M^P = 12$	N/A	824.8	697.2	340.4

creases, because a low temperature causes a long transition time, resulting in a long convergence time.

5. Evaluation

Our aim is to explore the potential applicability of probabilistic model checking to the Ising model, in the field of physics. In this paper, we limit our discussion to magnetics. Our proposed approach is, however, readily extendible to a wide range of applications, not limited in physics. Because of its simplicity and powerful expressiveness, the Ising model has been used to explain many kinds of phenomenon including sociology [22] and neural networks [23]. To deal with such an application in the proposed approach, it suffices to specify its own dynamics and properties of interest in a well-defined modeling language, provided by a model checker such as PRISM. This would be much easier than writing a new code for computer simulation from scratch.

The relationship between the Ising model and Markov chains is described in [15], and the relationship between Markov chains and probabilistic checking is evident. One advantage of using model checking is that it is possible to verify properties without obtaining an exact solution. All properties expressible by PCTL can be verified. One typical approach to analyzing physical phenomena is to use a hybrid approach in which properties are represented by differential equations. Compared with model checking, the use of such differential equations often becomes complex, and solutions are not always obtained. As an example of the difficulty in obtaining an exact solution, it is reported that the three-dimensional Ising model is intractable [24].

Our verification results suggest that probabilistic model checking seems reasonable for analyzing physical phenomena observed in the 1D Ising model. In our DTMC model, the number of spins is set to twelve, which is less than that of computer simulation, typically more than a hundred. However, the computed results perfectly match theoretical calculation results. This is unlikely the case when using computer simulation.

Historically, computer simulation has long been used to analyze the Ising model. The computation cost between probabilistic model checking and computer simulation cannot be simply compared because they have different advantages and disadvantages. We think that probabilistic model checking and computer simulation can be used in a complementary manner by taking both advantages. Exhaustive search is definitely an advantage of model checking. This advantage allows us to validate the correctness of models by checking various specifications. For example, probabilistic model checking can verify phenomena that are comparatively difficult to observe by computer simulation. Another advantage of model checking is the reusability of models. Once models are built, various specifications can be verified by only writing formulas. The expressive power of PCTL gives us concise expressions of complicated conditions. On the other hand, computer simulation is based on evaluation along a time series. Therefore, it is well-suited for statistical analysis, such as for computing the median of a physical quantity in a long period. Some cooperation between model checking and computer simulation is conceivable. For example, numerical computation results by probabilistic model checking can be used to facilitate the parameter settings of computer simulation. In the case of the 1D Ising model, the expected transition time in Sect. 4.2 can help estimate the sampling time of a simulation.

Both model checking and computer simulation model a system based on theoretical analyses and mathematical models. However, there seems no study that addresses the problem of constructing common models amenable to the two approaches. Such a common model could be the first step in cooperation. An immediate benefit of using a common model is that it becomes easy to confirm that the model correctly represents the target system. We can check the correctness of the model by probabilistic model checking, and then simulate by computer simulation. Conversely, the evaluation results of computer simulation could be confirmed on the ground of verification by model checking.

Although model checking has advantages, it also has drawbacks. One serious problem is the *state space explosion problem* where the size of the state space exponentially grows. This problem makes the complexity of verification high and limits scalability. For the 1D Ising model with n spins, the number of states is 2^n. Therefore, the state space easily grows as the number of spins increases, and the number of spins that can be dealt with is restricted. For much large systems, such as the two-dimensional Ising model, a straightforward approach will not work. *Abstraction* is known as an effective technique for reducing the size of the state space. For non-probabilistic systems, various kinds of abstraction techniques have been proposed, including abstraction in the presence of symmetry [25] and predicate abstraction [26]. In recent years, these techniques have been extended to probabilistic model checking, including symmetry reduction for probabilistic model checking [27], and probabilistic predicate abstraction [28].

6. Conclusion

We verified the one-dimensional Ising model using probabilistic model checking with DTMCs and specifications de-

scribed in PCTL. Some macroscopic physical properties relating to energy and magnetization were verified. Consequently, we demonstrated that our models are reasonable in representing the 1D Ising model and probabilistic model checking has power to analyze it.

In all DTMC models, the interaction coefficient was fixed to negative, i.e., the system was anti-ferromagnet. However, the analysis of ferromagnetic behavior is essentially the same as that of anti-ferromagnetic behavior. Behaviors relating to other physical quantities such as entropy and specific heat can be analyzed in the same way. In this paper, we assumed that there is no external magnetic field for simplicity. For a system with an external magnetic field, our approach can also work by incorporating the external factor into models.

We hope to extend this work to analyze more practical problems such as phase transition appearing in the two-dimensional Ising model. To achieve this, it is necessary to solve the state space explosion problem. As future work, we plan to study the application of abstraction techniques to the Ising model. Some of the authors proposed finite approximation analysis using predicate abstraction [29] for a deterministic one-dimensional cellular automaton. It seems that this approach can be extended for a DTMC.

References

[1] E.M. Clarke, O. Grumberg, and D.A. Peled, Model Checking, The MIT Press, Cambridge, Massachusetts, 1999.
[2] S. Hart, M. Sharir, and A. Pnueli, "Termination of probabilistic concurrent programs," POPL, pp.1–6, 1982.
[3] M.Y. Vardi, "Automatic verification of probabilistic concurrent finite-state programs," FOCS, pp.327–338, IEEE, 1985.
[4] C. Baier and M.Z. Kwiatkowska, "Model checking for a probabilistic branching time logic with fairness," Distributed Computing, vol.11, no.3, pp.125–155, 1998.
[5] M. Kwiatkowska and G. Norman, "Verifying randomized byzantine agreement," FORTE, LNCS, vol.2529, pp.194–209, Springer, Texas, USA, 2002.
[6] M. Kwiatkowska, G. Norman, and J. Sproston, "Probabilistic model checking of the IEEE 802.11 wireless local area network protocol," PAPM-PROBMIV, LNCS, vol.2399, pp.169–187, Springer, Copenhagen, Denmark, 2002.
[7] J.G. Kemeny and J.L. Snell, Finite Markov Chains, Van Nostrand, Princeton, New Jersey, 1960.
[8] C. Baier and J.P. Katoen, Principles of Model Checking, The MIT Press, Cambridge, Massachusetts, 2008.
[9] A. Hinton, M. Kwiatkowska, G. Norman, and D. Parker, "PRISM: A tool for automatic verification of probabilistic systems," TACAS, LNCS, vol.3920, pp.441–444, Springer, Vienna, Austria, 2006.
[10] J.P. Katoen, M. Khattri, and I.S. Zapreev, "A Markov reward model checker," QEST, pp.243–244, IEEE Computer Society, 2005.
[11] R. Barbuti, S. Cataudella, A. Maggiolo-Schettini, P. Milazzo, and A. Troina, "A probabilistic model for molecular systems," Fundam. Inform, vol.67, no.1-3, pp.13–27, 2005.
[12] M. Kwiatkowska, G. Norman, and D. Parker, "Using probabilistic model checking in systems biology," ACM SIGMETRICS Performance Evaluation Review, vol.35, no.4, pp.14–21, 2008.
[13] T. Sekizawa, T. Tsuchiya, T. Kikuno, and K. Takahashi, "Analysing the one dimensional Ising model by probabilistic model checking," Proc. IASTED Asian Conference on Modelling and Simulation, pp.199–204, ACTA Press, Beijing, China, 2007.
[14] E. Ising, "Beitrag zur theorie des ferromagnetismus," Zeitschrift für Physik, vol.31, pp.254–258, 1925.
[15] R. Kindermann and L.J. Snell, Markov Random Fields and Their Applications, Amer Mathematical Society, 1980.
[16] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, and A.H. Teller, "Equation of state calculations by fast computing machines," Journal of Chemical Physics, vol.21, no.6, pp.1087–1092, June 1953.
[17] W. Heisenberg, "Zur theorie des ferromagnetismus," Zeitschrift für Physik, vol.49, pp.619–636, 1928.
[18] C. Kittel, Introduction to Solid State Physics, Sixth ed., John Wiley & Sons, New York, 1995.
[19] N. Mott and R. Peierls, "Werner Heisenberg. 5 December 1901 – 1 February 1976," Biographical Memoirs of Fellows of the Royal Society, vol.23, pp.212–252, 1977.
[20] H. Hansson and B. Jonsson, "A logic for reasoning about time and reliability," Formal Asp. Comput, vol.6, no.5, pp.512–535, 1994.
[21] PRISM, "Probabilistic symbolic model checker." http://www.prismmodelchecker.org/
[22] W. Weidlich, "The statistical description of polarization phenomena in society," British Journal of Mathematics and Statistical Psychology, vol.24, pp.251–256, 1971.
[23] H. Köhler, S. Diederich, W. Kinzel, and M. Opper, "Learning algorithm for a neural network with binary synapses," Zeitschrift für Physik B Condensed Matter, vol.78, no.2, pp.333–342, 1990.
[24] S. Istrail, "Statistical mechanics, three-dimensionality and NP-completeness: I. universality of intractability for the partition function of the ising model across non-planar surfaces (extended abstract)," STOC, pp.87–96, 2000.
[25] E.M. Clarke, S. Jha, R. Enders, and T. Filkorn, "Exploiting symmetry in temporal logic model checking," Formal Methods in System Design, vol.9, no.1/2, pp.77–104, 1996.
[26] S. Graf and H. Saïdi, "Construction of abstract state graphs with PVS," 9th International Conference on Computer Aided Verification (CAV), LNCS, vol.1254, pp.72–83, Springer, Haifa, Israel, 1997.
[27] M. Kwiatkowska, G. Norman, and D. Parker, "Symmetry reduction for probabilistic model checking," Proc. 18th International Conference on Computer Aided Verification (CAV'06), LNCS, vol.4114, pp.234–248, Springer-Verlag, Seattle, WA, USA, 2006.
[28] M. Kattenbelt, M. Kwiatkowska, G. Norman, and D. Parker, "Game-based probabilistic predicate abstraction in PRISM," Proc. 6th Workshop on Quantitative Aspects of Programming Languages (QAPL'08), 2008.
[29] K. Takahashi, Y. Tanabe, and T. Sekizawa, "Finite approximation analysis of one dimensional cellular automata," Japan Society for Software Science and Technology, Computer Software, 2006.

Appendix: An Example of PRISM Code

Figure A· 1 specifies a PRISM code in which the number of spins n is 12. The model consists of the model type, declarations of constant variables, modules, formulas and reward definition.

The model type `probabilistic` specifies that the code is a model of DTMC. Following the type specification, some global constants are declared.

The model includes n modules, each of which represents a spin. A module is defined by two parts: a local variable and commands. For example, `module Spin1` represents spin σ_1 and its probabilistic transitions. In `module Spin1`, the local variable `s1` takes an integer value of 0 or 1 representing the value of spin σ_1, which is initialized by 0. After that, probabilistic transitions follow. Every probabilistic transition is expressed by a "command" consisting

```
probabilistic              // DTMC

const double e = 2.71828;  // the base of the natural logarithm
const double T = 1.0;      // temperature
const double p = 1 / func(pow, e, 4.0/T);
const int n = 12;          // the number of spins

module Spin1
  s1 : [0..1] init 0;
  [] s0=0 & s1=0 & s2=0 -> (s1'=1);
  [] s0=0 & s1=0 & s2=1 -> (s1'=1);
  [] s0=0 & s1=1 & s2=0 -> p : (s1'=0) + (1-p) : true;
  [] s0=0 & s1=1 & s2=1 -> (s1'=0);
  [] s0=1 & s1=0 & s2=0 -> (s1'=1);
  [] s0=1 & s1=0 & s2=1 -> p : (s1'=1) + (1-p) : true;
  [] s0=1 & s1=1 & s2=0 -> (s1'=0);
  [] s0=1 & s1=1 & s2=1 -> (s1'=0);
endmodule

formula E= func(mod, (1+s0+s1), 2) +func(mod, (1+s1+s2), 2)
          +func(mod, (1+s2+s3), 2) +func(mod, (1+s3+s4), 2)
          +func(mod, (1+s4+s5), 2) +func(mod, (1+s5+s6), 2)
          +func(mod, (1+s6+s7), 2) +func(mod, (1+s7+s8), 2)
          +func(mod, (1+s8+s9), 2) +func(mod, (1+s9+s10), 2)
          +func(mod, (1+s10+s11), 2) +func(mod, (1+s11+s0), 2);
formula M=s0+s1+s2+s3+s4+s5+s6+s7+s8+s9+s10+s11;

module Spin2 = Spin1[s0=s1, s1=s2, s2=s3, s3=s4, s4=s5,
                     s5=s6, s6=s7, s7=s8, s8=s9, s9=s10, s10=s11, s11=s0] endmodule
module Spin3 = Spin1[s0=s2, s1=s3, s2=s4, s3=s5, s4=s6,
                     s5=s7, s6=s8, s7=s9, s8=s10, s9=s11, s10=s0, s11=s1] endmodule
module Spin4 = Spin1[s0=s3, s1=s4, s2=s5, s3=s6, s4=s7,
                     s5=s8, s6=s9, s7=s10, s8=s11, s9=s0, s10=s1, s11=s2] endmodule
module Spin5 = Spin1[s0=s4, s1=s5, s2=s6, s3=s7, s4=s8,
                     s5=s9, s6=s10, s7=s11, s8=s0, s9=s1, s10=s2, s11=s3] endmodule
module Spin6 = Spin1[s0=s5, s1=s6, s2=s7, s3=s8, s4=s9,
                     s5=s10, s6=s11, s7=s0, s8=s1, s9=s2, s10=s3, s11=s4] endmodule
module Spin7 = Spin1[s0=s6, s1=s7, s2=s8, s3=s9, s4=s10,
                     s5=s11, s6=s0, s7=s1, s8=s2, s9=s3, s10=s4, s11=s5] endmodule
module Spin8 = Spin1[s0=s7, s1=s8, s2=s9, s3=s10, s4=s11,
                     s5=s0, s6=s1, s7=s2, s8=s3, s9=s4, s10=s5, s11=s6] endmodule
module Spin9 = Spin1[s0=s8, s1=s9, s2=s10, s3=s11, s4=s0,
                     s5=s1, s6=s2, s7=s3, s8=s4, s9=s5, s10=s6, s11=s7] endmodule
module Spin10 = Spin1[s0=s9, s1=s10, s2=s11, s3=s0, s4=s1,
                      s5=s2, s6=s3, s7=s4, s8=s5, s9=s6, s10=s7, s11=s8] endmodule
module Spin11 = Spin1[s0=s10, s1=s11, s2=s0, s3=s1, s4=s2,
                      s5=s3, s6=s4, s7=s5, s8=s6, s9=s7, s10=s8, s11=s9] endmodule
module Spin12 = Spin1[s0=s11, s1=s0, s2=s1, s3=s2, s4=s3,
                      s5=s4, s6=s5, s7=s6, s8=s7, s9=s8, s10=s9, s11=s10] endmodule

rewards [] true : 1; endrewards
```

Fig. A· 1 A PRISM code for the 1D Ising model ($n = 12$).

of a guard followed by probabilistic choices of updates to variables. Such guards represent conditions of spin σ_1 and its nearby spins. The following updates to variables describe the random spin flipping algorithm. For example, the third command means that if guard $\sigma_0 = 0 \wedge \sigma_1 = 1 \wedge \sigma_2 = 0$ holds, then either update σ_1 to $\sigma_1' = 0$ occurs with a prob-

ability $p = e^{-4/T}$, or no variables are updated. Modules `Spin2` to `Spin10` are renamed copies of module `Spin1`. Such duplications of modules are realized by "module renaming", which allows to change the name of the module and its definitions at a textual level.

As mentioned in Sect. 4.1, two physical quantities `E` and `M` are defined using the shorthand expression "formula". These expressions do not appear in the code, but are used to express specifications, such as the formula *equilibrium*.

At last, a transition reward of 1 is assigned to every probabilistic transition, which corresponds to one judgment of spin flipping.

Toshifusa Sekizawa received his M.S. degrees in physics from Gakushuin University in 1998. He previously worked at Nihon Unisys Ltd. and Japan Science and Technology Agency. He is currently working in the National Institute of Advanced Industrial Science and Technology. He is also a Ph.D. student of the Graduate School of Information Science and Technology, Osaka University.

Tatsuhiro Tsuchiya received the M.E. and Ph.D. degrees from Osaka University in 1995 and 1998, respectively. He is currently an associate professor of the Department of Information Systems Engineering at Osaka University. His research interests are in the areas of model checking and distributed fault-tolerant systems.

Koichi Takahashi received his B.S. and M.S. degrees in mathematics from Nagoya University in 1986 and 1988, and his Ph.D. degree in information science from the University of Tokyo in 2002. Since 1988, he has worked at the Electrotechnical Laboratory (currently the National Institute of Advanced Industrial Science and Technology). His research interests include theoretical verifications.

Tohru Kikuno received his M.S. and Ph.D. degrees in computer science from Osaka University in 1972 and 1975, respectively. He was with Hiroshima University from 1975 to 1987. Since 1990, he has been a Professor at Osaka University. His research interests include the quantitative evaluation of software development and the analysis and design of fault-tolerant systems.