

Research Article

Approximate Analyzing of Labeled Transition Systems

Qiong Yu,¹ Shihan Yang,² and Jinzhao Wu³

¹*School of Science, Guangxi University for Nationalities, No. 188, East Daxue Road, Nanning, Guangxi 530006, China*

²*School of Information Science and Engineering, Guangxi University for Nationalities, No. 188, East Daxue Road, Nanning, Guangxi 530006, China*

³*Guangxi Key Laboratory of Hybrid Computation and IC Design Analysis, No. 188, East Daxue Road, Nanning, Guangxi 530006, China*

Correspondence should be addressed to Shihan Yang; dr.yangsh@gmail.com

Received 11 June 2015; Accepted 6 July 2015

Academic Editor: Xiaoyu Song

Copyright © 2015 Qiong Yu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As the most important formal semantic model, labeled transition systems are widely used, which can describe the general concurrent systems or control systems without disturbance. However, under normal circumstance, transition systems are complex and difficult to use due to large amount of calculation and the state space explosion problems. In order to overcome these problems, approximate equivalent labeled transition systems are proposed by means of incomplete low-up matrix decomposition factorization. This technique can reduce the complexity of computation and calculate under the allowing errors. As for continuous-time linear systems, we develop a modeling method of approximated transition system based on the approximate solution of matrix, which provides a facility for approximately formal semantic modeling for linear systems and to effectively analyze errors. An example of application in the context of linear systems without disturbances is studied.

1. Introduction

Transition system is an important formal model of concurrent systems, which is widely used in concurrency theory, such as petri nets and process algebras. Petri nets can be used as a graphical tool to simulate the dynamic behavior and concurrent activities of the systems and also can be used as a mathematical tool to establish mathematical model describing the behaviors of the systems. It is a widely transition system and gives a method to describe Petri net using classical transition system [1]. Apart from Petri net theory, there is a process algebra, which is used to describe and study concurrent or distributed system behavior theory by algebraic methods. Thus, many process algebraic systems are proposed. Among them, the first is to propose the following three process algebra systems: CCS (Calculus of Communicating Systems) proposed by Milner [2, 3], CSP (Communicating Sequential Processes) proposed by Hoare [4], and ACP (Algebra of Communicating Processes) proposed by Baeten and Weijland [5]. In recent years, some new process calculus systems have been proposed. For example,

probabilistic process algebra [6–12] and stochastic process algebra [13] are proposed to describe possible or random information; real-time process algebra [14–17] is proposed to describe the action time. In the field of process algebra, as the most important semantic model, the transition system is widely used, especially labeled transition system [18], which can describe the general behavior of concurrent systems.

Approximation of transition systems has recently been introduced as a powerful tool for the approximation of continuous systems. The notion of simulation is one such formal notion of abstraction that has been used for reducing the complexity of finite state systems such as labeled transition systems. Approximation of purely discrete systems has traditionally been based on language inclusion and equivalence with notions such as simulation or bisimulation relations [19]. These concepts have been useful for simplifying problems such as safety verification or controller synthesis. However, in the cases that the quality requirements of simulation or bisimulation are so exacting sometimes it is so difficult for us to construct finite abstracts with the original system simulation or bisimulation. Although finite abstract with

the original system approximate simulation or approximate bisimulation is easy to construct, these methods can only ensure that the system approximately satisfies the given range and is not able to calculate the allowable error of transition between states. To solve this problem, this paper puts forward a design method for transition system—approximate simulation method based on ILU.

As a particular class of preconditioners, incomplete factorizations can be thought of as approximating the exact LU factorization of a given matrix (e.g., computed via Gaussian elimination) by disallowing certain filling of this. As opposed to other PDE-based preconditioners such as multigrid and domain decomposition, this class of preconditioners is primarily algebraic in nature and can be applied to any matrices in principle. However, one of the basic methods of preconditioned method is the use of incomplete LU decomposition. The most well known is the incomplete triangular decomposition ILU0 [20] that has nothing more than zero-element filling. Since ILU0 unchanged the structure of sparse matrix, the application of ILU0 is very convenient. However, because of this, it is so rough approximation of the coefficient matrix that the effectiveness of ILU0 is limited. To improve the approximation, filling with nonzero elements is allowed [21, 22]. While the ILU preconditioner works quite well for many problems, it does not perform well for some PDE problems [23]. Then, the generalization of this modified ILU (MILU) preconditioner is proposed by Gustafsson [24]. MILU is to add the fill-ins abandoned by ILU back to the diagonal type, which makes up for inaccuracies caused by discarded. And its accuracy can be calculated. However, mathematically, the transition between states can be described by matrix, and then the allowable error of the approximate transition is needed to be characterized. Furthermore, the allowable error can be defined by the MILU accuracy.

In the rest of this paper, Section 2 reviews the approximate simulation relations of transition systems. In Section 3, the approximate and incomplete factorizations are recalled. In Section 4, the approximate equivalent of labeled transition systems is proposed in detail. Section 5 proposes the theory of the approximate labeled transition systems. In Section 6, a case is studied. Then Section 7 draws the conclusion.

2. Approximation of Transition Systems

2.1. Labeled Transition Systems. We know that labeled transition systems allow modeling, in a unified framework, discrete, continuous, and hybrid systems. The main results [25] are reviewed here.

Definition 1. A labeled transition system with observation is a tuple $T = (Q, \Sigma, \rightarrow, Q^0, \Pi, \langle\langle \cdot \rangle\rangle)$ that consists of

- (i) a set Q of states,
- (ii) a set Σ of labels,
- (iii) a transition relation $\rightarrow \subseteq Q \times \Sigma \times Q$,
- (iv) a set $Q^0 \subseteq Q$ of initial states,
- (v) a set Π of observations,
- (vi) an observation map $\langle\langle \cdot \rangle\rangle : Q \rightarrow \Pi$.

A state trajectory of T is a sequence of transitions,

$$q^0 \xrightarrow{\sigma^0} q^1 \xrightarrow{\sigma^1} q^2 \xrightarrow{\sigma^2} \dots, \quad (1)$$

where $q^0 \in Q^0$, $q^i \in Q$, $\sigma^{i-1} \in \Sigma$, $i = 1, 2, \dots$

For a given initial state and sequence of labels, there may exist several state trajectories of T . Thus, the systems we consider are possibly nondeterministic. The associated external trajectory

$$\pi^0 \xrightarrow{\sigma^0} \pi^1 \xrightarrow{\sigma^1} \pi^2 \xrightarrow{\sigma^2} \dots, \quad (2)$$

where $\pi^i = \langle\langle q^i \rangle\rangle$, $i = 0, 1, 2, \dots$,

describes the evolution of the observations under the dynamics of the labeled transition system.

The set of external trajectories of the labeled system T is called the language of T and is denoted by $L(T)$. The subset of Π reachable by the external trajectories of T is denoted by $\text{Reach}(T)$:

$$\text{Reach}(T) = \left\{ \pi \in \Pi \mid \exists \pi^0 \xrightarrow{\sigma^0} \pi^1 \xrightarrow{\sigma^1} \pi^2 \xrightarrow{\sigma^2} \dots \right. \\ \left. \in L(T), \exists j \in \mathbb{N}, \pi^j = \pi \right\}, \quad (3)$$

where \mathbb{N} is the set of positive integers.

An important problem for transition systems is the safety verification problem which consists in checking whether the reachable set $\text{Reach}(T)$ intersects a set of observations Π_U associated with unsafe states.

2.2. Approximate Simulation Relations. Exact simulation relations between two labeled transition systems require that their observations are (and remain) identical. Approximate simulation relations are less rigid since they only require that the observations of both systems are (and remain) arbitrarily close.

Let $T_1 = (Q_1, \Sigma_1, \rightarrow_1, Q_1^0, \Pi_1, \langle\langle \cdot \rangle\rangle_1)$ and $T_2 = (Q_2, \Sigma_2, \rightarrow_2, Q_2^0, \Pi_2, \langle\langle \cdot \rangle\rangle_2)$ be two labeled transition systems with the same set of labels ($\Sigma_1 = \Sigma_2 = \Sigma$) and the same set of observations ($\Pi_1 = \Pi_2 = \Pi$) (i.e., T_1 and T_2 are elements of $T(\Sigma, \Pi)$). Let us assume that the set of observation Π is a metric space; d_Π denotes the metric of Π .

Definition 2. A metric of a set A is a positive function $d : A \times A \rightarrow \mathbb{R} \cup +\infty$, where \mathbb{R} is the set of real numbers, such that the three following properties hold: for all $x, y, z \in A$

- (1) $d(x, y) = 0$ if and only if $x = y$,
- (2) $d(x, y) = d(y, x)$,
- (3) $d(x, z) \leq d(x, y) + d(y, z)$.

If the second property is replaced by $x = y \Rightarrow d(x, y) = 0$ then d is called a pseudometric. If the third property is dropped, then d is called a directed metric.

Definition 3. A relation $S_\delta \subseteq Q_1 \times Q_2$ is a δ -approximate simulation relation of T_1 by T_2 if for all $(q_1, q_2) \in S_\delta$

- (1) $d_\Pi(\langle\langle q_1 \rangle\rangle_1, \langle\langle q_2 \rangle\rangle_2) \leq \delta$,
- (2) $\forall q_1 \xrightarrow{\sigma} q'_1, \exists q_2 \xrightarrow{\sigma} q'_2$ such that $(q'_1, q'_2) \in S_\delta$.

Note that, in there, \rightarrow_1 describes a transition relation of T_1 , \rightarrow_2 describes a transition relation of T_2 , q'_1 represents the next state of q_1 , q'_2 represents the next state of q_2 , and $\sigma \in \Sigma$. And that, for $\delta = 0$, we have the usual notion of exact simulation relation.

Definition 4. T_2 approximately simulates T_1 with the precision δ (noted $T_1 \preceq_\delta T_2$), if there exists S_δ , a δ -approximate simulation relation of T_1 by T_2 such that, for all $q_1 \in Q_1^0$, there exists $q_2 \in Q_2^0$ such that $(q_1, q_2) \in S_\delta$.

If T_2 approximately simulates T_1 with the precision δ then the language of T_1 is approximated with precision δ by the language of T_2 .

Theorem 5. If $T_1 \preceq_\delta T_2$, then for all external trajectories of T_1 ,

$$\pi_1^0 \xrightarrow{\sigma^0} \pi_1^1 \xrightarrow{\sigma^1} \pi_1^2 \xrightarrow{\sigma^2} \dots, \quad (4)$$

there exists an external trajectory of T_2 with the same sequence of labels

$$\pi_2^0 \xrightarrow{\sigma^0} \pi_2^1 \xrightarrow{\sigma^1} \pi_2^2 \xrightarrow{\sigma^2} \dots \quad (5)$$

such that, for all $i \in \mathbb{N}$, $d_\Pi(\pi_1^i, \pi_2^i) \leq \delta$.

In Theorem 5, if T_2 approximately simulates T_1 with the precision δ , there must exist a state of T_2 approximate simulation the state of T_1 . But it is very difficult for us to calculate the degree of approximation.

3. Approximate and Incomplete Factorizations

The general problem of finding a preconditioner for a large sparse linear system $Ax = b$ is to find a matrix M (the preconditioner). However, the matrix M should be with the following properties:

- (i) M is a nonsingular matrix,
- (ii) M is a good approximation to A in some sense,
- (iii) the system $Mx = b$ is much easier to solve than the original system $Ax = b$,
- (iv) the construction of M has some memory and CPU needs protection,
- (v) the condition number of the $M^{-1}A$ is much less than that of A , and its minimum singular value becomes larger.

That is, we are looking for a nearby problem which is easier to solve than the given one. The idea is to look for a matrix M such that the original linear system $Ax = b$ is transformed into an equivalent linear system

$M^{-1}Ax = M^{-1}b$. Mathematically, we will solve the preconditioned system $M^{-1}Ax = M^{-1}b$ (or the symmetric version $(M^{-1/2}AM^{-1/2})(M^{1/2}x) = M^{-1/2}b$ when both A and M are symmetric positive definite) by standard iterative methods (such as the conjugate gradient method), in which only the actions of A and M^{-1} are needed.

There are several preconditions: matrix splitting preconditioner, polynomial preconditioners, incomplete factorization preconditioners, approximate inverse preconditioners, and multilevel preconditioners. Here, we mainly focus on incomplete factorization preconditioners, which can be applied to a general purpose. They can be thought of as modifications of Gaussian Elimination, in which some sparse mode is required as the decomposition of matrices, such as making some of the special position of the elements be zero and also abandoning some absolute value in the process of decomposition of minor elements in order to ensure that sparse decomposition.

3.1. Incomplete Low-Up Matrix Decomposition (ILU). The most common type of incomplete factorization is based on taking a set S of matrix positions and keeping all positions outside this set equal to zero during the factorization. The resulting factorization is incomplete in the sense that fill is suppressed.

Definition 6. A matrix A , if $A = LU$, where L (U) is the lower (upper) triangular sparse matrix, is called a complete decomposition of the matrix A , that is, LU [26].

Definition 7. A matrix A , if $A = LU + R$, where L (U) is the lower (upper) triangular sparse matrix, is called an incomplete decomposition of the matrix A , that is, ILU . Here R is called error matrix [27].

Let the allowable fill-in positions be given by the index set S ; that is,

- (1) $l_{i,j} = 0$ if $j > i$ or $(i, j) \notin S$; $u_{i,j} = 0$ if $i > j$ or $(i, j) \notin S$.

A commonly used strategy is to define S by

- (2) $S = \{(i, j) \mid a_{i,j} \neq 0\}$.

That is, the only nonzeros allowed in the LU factors are those for which the corresponding entries in A are nonzero. Let the preconditioner M be defined by the product of the resulting LU factors, that is, $M = LU$. If M becomes a good preconditioner, it must be a good approximation to A in some measure. A typical strategy is to require the entries of M to match those of A on the set S :

- (3) $m_{i,j} = a_{i,j}$ if $(i, j) \in S$.

Even though conditions (1) and (3) together are sufficient (for certain classes of matrices) to determine the nonzero entries of L and U directly, it is more natural and simpler to compute these entries based on a simple modification of the Gaussian elimination algorithm. Matrix A is decomposed into L and U by the incomplete LU . And the corresponding entries of L and U are the same as those obtained by setting those entries of the full LU factors which belong to S .

```

(1) for  $r := 1$  Step  $i$  until  $n - 1$  do
(2)    $d := 1/a_{r,r}$ ;
(3)   for  $i := (r + 1)$  Step (1) until  $n$  do
(4)     if  $(i, r) \in S$  then;
(5)      $e := da_{i,r}$ ;  $a_{i,r} := e$ ;
(6)     for  $j := (r + 1)$  Step (1) until  $n$  do
(7)       if  $(i, j) \in S$  and  $(r, j) \in S$  then;
(8)        $a_{i,j} := a_{i,j} - ea_{r,j}$ ;
(9)       end if;
(10)    end for ( $j$  - loop);
(11)    end if;
(12)  end for ( $i$  - loop);
(13) end for ( $r$  - loop);

```

ALGORITHM 1: Algorithm ILU.

However, the incomplete LU factors of A , respectively, with the lower and upper triangular part of the matrix A , have the same nonzero structure.

The constructing process of ILU precondition is in the process of Gaussian elimination by discarding partially filled elements to get sparse triangular matrixes L and U such that $M = LU$. From the decomposition of ILU we can see that the preconditioners M and A have the same nonzero structure. Therefore, the error matrix $R = A - LU = A - M$ and the main difference from the usual Gaussian elimination lie in how to discard fill-ins in the Gaussian elimination.

3.2. Modified Incomplete Low-Up Matrix Decomposition (MILU). While the ILU preconditioner works quite well for many problems, it does not perform well for some problems. We will next describe the generalization of this modified ILU (MILU) preconditioner to a general matrix A [28].

The basic idea is that in condition (3), for ILU, the condition $m_{i,i} = a_{i,i}$ is removed and a new row sum condition is added. That is, (3) is replaced by

$$(4) \sum_{j=1}^n m_{i,j} = \sum_{j=1}^n a_{i,j} \quad \forall i \text{ and } m_{i,j} = a_{i,j} \text{ if } i \neq j \text{ and } (i, j) \in S.$$

Again, for certain classes of matrices, conditions (4) and (1) are sufficient to determine the LU factors in MILU directly. However, in practice it is easier to compute these LU factors by a modification of the ILU algorithm: instead of dropping the disallowed fill-ins in the ILU algorithm, these terms are added to the diagonal of the same row. Furthermore, the idea of MILU is to make up for inaccuracies caused by discarded by using fill-ins plus backup to cover the diagonal type.

Incomplete factorization preconditioners can be portrayed accuracy and stability. Here, accuracy refers to the degree of preconditioner M and matrix A can be measured by the size of $\|M - A\|_F$. And stability refers to the degree of preconditioner M and the unit matrix E can be measured by the size of $\|E - M^{-1}A\|_F$ (for the left preconditioner). Below we give algorithm ILU and algorithm MILU [29].

From the analysis of Algorithms 1 and 2, the difference between algorithm ILU and MILU is that certain fill-ins given up in algorithm ILU are back to the main diagonal.

```

(1) for  $r := 1$  Step (1) until  $n - 1$  do
(2)    $d := 1/a_{r,r}$ ;
(3)   for  $i := (r + 1)$  Step (1) until  $n$  do
(4)     if  $(i, r) \in S$  then;
(5)      $e := da_{i,r}$ ;  $a_{i,r} := e$ ;
(6)     for  $j := (r + 1)$  Step (1) until  $n$  do
(7)       if  $(r, j) \in S$  then;
(8)       if  $(i, j) \in S$  then;
(9)        $a_{i,j} := a_{i,j} - ea_{r,j}$ ;
(10)      else;
(11)        $a_{i,i} := a_{i,i} - ea_{r,j}$ ;
(12)      end if;
(13)      end if;
(14)    end for ( $j$  - loop);
(15)    end if;
(16)  end for ( $i$  - loop);
(17) end for ( $r$  - loop);

```

ALGORITHM 2: Algorithm MILU.

4. Equivalent Labeled Transition Systems

In labeled transition systems, we can assume that Σ is a finite set of labels; a label $\sigma \in \Sigma$ describes an action. $\rightarrow \subseteq Q \times \Sigma \times Q$ is a transition relation, where Q is a set of states. Each transition \rightarrow is a tuple $\langle q, \sigma, q' \rangle$, where q and q' represent the pre- and poststates of the transition, respectively. A state trajectory of labeled transition systems is a sequence of transitions:

$$q^0 \xrightarrow{\sigma^0} q^1 \xrightarrow{\sigma^1} q^2 \xrightarrow{\sigma^2} \dots, \quad (6)$$

where $q^0 \in Q^0$ is the initial state and $Q^0 \subseteq Q$ is the initial set of states.

For a given initial state and sequence of labels, there may be several state trajectories of labeled transition systems. Thus, the systems we consider are possibly nondeterministic.

Here, we mainly consider the set Σ of labels. Let label $\sigma \in \Sigma$ has the following form:

$$\begin{aligned}
\dot{x}_1 &= a_{11}(t)x_1 + a_{12}(t)x_2 + \dots + a_{1n}(t)x_n \\
\dot{x}_2 &= a_{21}(t)x_1 + a_{22}(t)x_2 + \dots + a_{2n}(t)x_n \\
&\vdots \\
\dot{x}_n &= a_{n1}(t)x_1 + a_{n2}(t)x_2 + \dots + a_{nn}(t)x_n.
\end{aligned} \quad (7)$$

It can be abbreviated as $\dot{x} = A(t)x$, where $\dot{x} = (\dot{x}_1 \ \dot{x}_2 \ \dots \ \dot{x}_n)^T$, $x = (x_1 \ x_2 \ \dots \ x_n)^T$, and

$$A(t) = \begin{pmatrix} a_{11}(t) & a_{12}(t) & \dots & a_{1n}(t) \\ a_{21}(t) & a_{22}(t) & \dots & a_{2n}(t) \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}(t) & a_{n2}(t) & \dots & a_{nn}(t) \end{pmatrix}. \quad (8)$$

We can obtain the preconditioner $M(t)$ of matrix $A(t)$ by means of ILU. We assume the preconditioner $M(t)$ is

$$M(t) = \begin{pmatrix} b_{11}(t) & b_{12}(t) & \cdots & b_{1n}(t) \\ b_{21}(t) & b_{22}(t) & \cdots & b_{2n}(t) \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1}(t) & b_{n2}(t) & \cdots & b_{nn}(t) \end{pmatrix}. \quad (9)$$

According to Section 3, we can know that the equivalent equation of $\dot{x} = A(t)x$ is $M^{-1}(t)A(t)x = M^{-1}(t)\dot{x}$. Let σ' be symbol of $M^{-1}(t)A(t)x = M^{-1}(t)\dot{x}$. That is, $\sigma' : M^{-1}(t)A(t)x = M^{-1}(t)\dot{x}$. So label σ and label σ' are equivalent.

Definition 8. Label $\sigma' : M^{-1}(t)A(t)x = M^{-1}(t)\dot{x}$ is the equivalent label of label $\sigma : \dot{x} = A(t)x$, where x and \dot{x} are the pre- and poststate values of the transition, if matrix $M(t)$ is the preconditioner matrix of matrix $A(t)$.

The labeled transition system can only be used for symbolic reasoning. Two labels are equivalent if they have the same form, syntax, and semantics. Here, Definition 8 gives the equivalent of two labels. It has the characteristic that labels' equations are equivalent instead of having the same form, syntax, and semantics.

For all labels $\sigma_i \in \Sigma$, $i \in \mathbb{N}$, where \mathbb{N} is the set of positive integers, $\sigma_i : \dot{x} = A_i(t)x$; its equivalent label is $\sigma'_i : M_i^{-1}(t)A_i(t)x = M_i^{-1}(t)\dot{x}$, where matrix $M_i(t)$ is the preconditioner matrix of matrix $A_i(t)$. Let $\Sigma' = \{\sigma'_i \mid i \in \mathbb{N}\}$.

Definition 9. Set $\Sigma = \{\sigma_i \mid i \in \mathbb{N}\}$ of labels and set $\Sigma' = \{\sigma'_i \mid i \in \mathbb{N}\}$ are equivalent if $\forall \sigma_i \in \Sigma$; there exists $j \in \mathbb{N}$ making σ_i and σ_j equivalent.

A labeled transition system with observation is a tuple $T = (Q, \Sigma, \rightarrow, Q^0, \Pi, \langle\langle \cdot \rangle\rangle)$. Let $T' = (Q, \Sigma', \rightarrow, Q^0, \Pi, \langle\langle \cdot \rangle\rangle)$.

Definition 10. Labeled transition system with observation T and labeled transition system with observation T' are equivalent if set Σ of labels and set Σ' of labels are equivalent. One labeled transition system is called equivalent system of another labeled transition system.

Here, we only study labels of labeled transition systems.

5. Approximate Labeled Transition System

In general, we know that a labeled transition system with observations is used to prescribe control systems. In this section, our goal is to study $\dot{x} = Ax$ (the transition relationship between states).

The purpose of this paper is to characterize approximate similar transition systems that are generated by continuous-time linear systems, which are conceptually similar to the transition systems generated by discrete-time systems. We begin with studying continuous-time linear systems whose dynamics are closer to transition systems due to the existence

of an atomic time step [30]. Consider a continuous-time linear system without disturbances

$$C : \dot{x} = Ax, \quad (10)$$

where with time $t \in \mathbb{R}_+$, state $x(t) \in \mathbb{R}^n$ and matrix A of approximate dimension, given an initial condition x_0 .

Definition 11. Consider continuous-time system C given by $\dot{x} = Ax$ and observation set $\Pi = B^p$. The transition system $T_C^{r+} = (Q, \Sigma, \rightarrow, \Pi, \langle\langle \cdot \rangle\rangle)$ generated by C and Π consists of

- (i) state space $Q = \mathbb{R}^n$,
- (ii) label $\Sigma = \mathbb{R}_+$,
- (iii) transition relation $\rightarrow \subseteq Q \times \mathbb{R}_+ \times Q$ defined as $x \xrightarrow{t} x' \Leftrightarrow \exists u_{[0,t]}$ with $x' = \Phi_C(t, x, u_{[0,t]})$,
- (iv) observation $\Pi = B^p$,
- (v) observation map $\langle\langle \cdot \rangle\rangle : Q \rightarrow \Pi$.

Usually, the linear systems are presented by ordinary differential equations, partial differential equations, and difference equations.

For $t \in \mathbb{R}^+$, the transition relation is given by $q \xrightarrow{t} q'$ if and only if there exists a function $x(\cdot)$ such that $x(0) = q$, $x(t) = q'$, and for almost all $s \in [0, t]$, $\dot{x}(s) \in F(x(s))$, where F is a valued map.

5.1. Approximate Similar States. Consider the following continuous-time linear system of state q translated to another state q' without disturbances, where the continuous-time linear system is a nondeterministic continuous system. That is, $q \xrightarrow{\sigma} q'$:

$$\begin{aligned} \dot{x}_1 &= a_{11}(t)x_1 + a_{12}(t)x_2 + \cdots + a_{1n}(t)x_n \\ \dot{x}_2 &= a_{21}(t)x_1 + a_{22}(t)x_2 + \cdots + a_{2n}(t)x_n \\ &\vdots \\ \dot{x}_n &= a_{n1}(t)x_1 + a_{n2}(t)x_2 + \cdots + a_{nn}(t)x_n \end{aligned} \quad (11)$$

and state value of the initial state q is $x(0) = (b_1 \ b_2 \ \cdots \ b_n)$, $t \in [0, t_0]$.

The state is observed through the variable $\pi_1(t) = x_1(t)$ and the transition relation is expressed as \rightarrow_1 .

We know that the equation is the linear differential equations. We introduce the following mark:

$$A(t) = \begin{pmatrix} a_{11}(t) & a_{12}(t) & \cdots & a_{1n}(t) \\ a_{21}(t) & a_{22}(t) & \cdots & a_{2n}(t) \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}(t) & a_{n2}(t) & \cdots & a_{nn}(t) \end{pmatrix}, \quad (12)$$

where $A(t)$ is an n by n matrix. The above equations can be expressed as

$$\dot{x} = A(t)x, \quad (13)$$

where $x = (x_1 \ x_2 \ \cdots \ x_n)^T$, $\dot{x} = (\dot{x}_1 \ \dot{x}_2 \ \cdots \ \dot{x}_n)^T$.

However, $A(t)$ is a large sparse matrix for most mathematical models of practical problems in the real world, so it is very difficult for us to obtain the solution of the equation $\dot{x} = A(t)x$. Then we use ILU to calculate the approximate solution. The preconditioner $M = LU$, where L is a triangular matrix and U is an upper triangular matrix. Hence, the original linear system $\dot{x} = A(t)x$ is transformed into an equivalent linear system $M^{-1}Ax = M^{-1}\dot{x}$ (or the symmetric version $(M^{-1/2}AM^{-1/2})(M^{1/2}x) = M^{-1/2}\dot{x}$ when both A and M are symmetric positive definite). Because system $\dot{x} = A(t)x$ is a portrait of the transition process from the state q to another state q' , the elements of $A(t)$ present the dynamic changes from state q to another state q' . The preconditioner M is the approximate dynamic changed from the state q to another state q' . Hence, $A - M$, an error matrix, has the meaning that state q can be translated into another state q' in the error, and its degree of approximation is $\|M - A\|_F$. Then we can define the approximate distance between the two states.

Proposition 12. $d(x, y) = \|x - y\|_F$ is the distance between x and y , where $x, y \in X$ and X is a linear space.

Proof. In fact, for the norm axioms, $\forall x, y, z \in X$, $d(x, y) = \|x - y\|_F \geq 0$, and $d(x, y) = 0$ if and only if $\|x - y\|_F = 0$; that is, $x = y$.

What is more,

$$\begin{aligned} d(x, y) &= \|x - y\|_F = \|y - x\|_F = d(y, x), \\ d(x, y) &= \|x - y\|_F = \|x - z + z - y\|_F \\ &\leq \|x - z\|_F + \|z - y\|_F = d(x, z) + d(z, y). \end{aligned} \quad (14)$$

□

Definition 13. $\forall q, q' \in Q$, the approximate distance between q and q' is

$$d_Q(q, q') = \|M - A\|_F. \quad (15)$$

Let us assume that q_1, q_2, q_3 are elements of Q and the state trajectory is

$$q_1 \xrightarrow{\sigma_1} q_2 \xrightarrow{\sigma_2} q_3. \quad (16)$$

The mathematical relationship of $q_1 \xrightarrow{\sigma_1} q_2, q_2 \xrightarrow{\sigma_2} q_3$ is respective $\dot{x} = A_1(t)x, \dot{z} = A_2(t)z$. This system is a continuous transition, so a transition from state q_1 to state q_3 can be expressed as $q_1 \xrightarrow{\sigma_2\sigma_1} q_3$. This can be represented by using Figure 1. A line with an arrow indicates state transition, where the end of a line without arrow points to the prestate and the end of a line with arrow points to the poststate, a label marks on the line.

Call $d_Q(q, q') = \|M - A\|_F$ as the degree of approximation of state q translated into state q' . Furthermore, because $x(0) = q, x(t_0) = q'$, the solution y of the equivalent linear system $M^{-1}Ax = M^{-1}\dot{x}$ is the approximate solution of the original

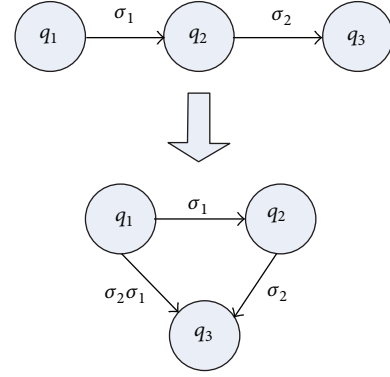


FIGURE 1: Continuous transition of states.

linear system $\dot{x} = A(t)x$, where $t \in [0, t_0]$, $y(0) = q, y(t_0) = p'$. The mathematical relationship of $q \xrightarrow{\sigma} p'$ is

$$\dot{y} = A(t)y, \quad (17)$$

where $y = (y_1 \ y_2 \ \dots \ y_n)^T, \dot{y} = (\dot{y}_1 \ \dot{y}_2 \ \dots \ \dot{y}_n)^T$, and state value of initial state $y(0) = (b_1 \ b_2 \ \dots \ b_n)$.

The state is observed through the variable $\pi_2(t) = y_1(t)$ and the transition relation is expressed as \rightarrow_2 . Therefore, the accuracy between the exact solution x and approximate solution y of the linear systems is $\delta = \|M - A\|_F$. That is, $\delta = \lim_{t \rightarrow +\infty} \|x - y\|_F = \|M - A\|_F$. And $d_\Pi(\pi_1(t), \pi_2(t)) = \|x_1(t) - y_1(t)\|_F \leq \|x - y\|_F = \delta$.

Theorem 14. Let y be the approximate solution of $\dot{x} = A(t)x$; then $\dot{y} = A(t)y$ simulates approximately $\dot{x} = A(t)x$ with precision $\lim_{t \rightarrow +\infty} \|A(t)\|_F \|M - A\|_F$.

Proof.

$$\begin{aligned} \lim_{t \rightarrow +\infty} \|\dot{x} - \dot{y}\|_F &= \lim_{t \rightarrow +\infty} \|A(t)x - A(t)y\|_F \\ &= \lim_{t \rightarrow +\infty} \|A(t)(x - y)\|_F \\ &\leq \lim_{t \rightarrow +\infty} \|A(t)\|_F \|x - y\|_F \\ &\leq \lim_{t \rightarrow +\infty} \|A(t)\|_F \lim_{t \rightarrow +\infty} \|x - y\|_F \\ &= \lim_{t \rightarrow +\infty} \|A(t)\|_F \|M - A\|_F. \end{aligned} \quad (18)$$

So $\dot{y} = A(t)y$ simulates approximately $\dot{x} = A(t)x$. And therefore, we can say the state p' simulates approximately the state q' . □

Here, approximate simulation of states is controlled by labels, where labels are special. Furthermore, the degree of approximate similar states can be obtained by the approximate solution of labels' mathematical models.

5.2. Approximate Similar Linear Systems. We have given an approximate simulation method for a state. Now we consider approximate simulation of linear systems.

Let the transition system generated by C and Π be expressed as $T_C^{r+} = (Q, \Sigma, \rightarrow, \Pi, \langle\langle \cdot \rangle\rangle)$. Furthermore, for any state in Q we have given its approximate state; that is, $\forall q_i \in Q$, $i = 1, 2, \dots$, q_i is approximately simulated by state q'_i by means of ILU, and the equivalent label of label $\sigma_i : \dot{x} = A_i(t)x$ is $\sigma'_i : M_i^{-1}(t)A_i(t)x = M_i^{-1}(t)\dot{x}$, where matrix $M_i(t)$ is the preconditioner of matrix $A_i(t)$. Let $Q' = \{q'_i \mid i = 1, 2, \dots\}$, $\Sigma' = \{\sigma'_i \mid i = 1, 2, \dots\}$, and then $T' = (Q', \Sigma', \rightarrow, \Pi, \langle\langle \cdot \rangle\rangle)$.

Definition 15. T' approximately simulates T , if there exists q'_i and σ'_i , $i = 1, 2, \dots$, where state q'_i is an approximate simulation state of state q_i ; label σ'_i is the equivalent label of label σ_i . One labeled transition system is called approximate system of another labeled transition system.

Theorem 16. The precision of the T' approximate simulation T is

$$\delta = \max_i \left(\lim_{t \rightarrow +\infty} \|A_i(t)\|_F \|M_i - A_i\|_F \right), \quad (19)$$

$$i = 1, 2, \dots$$

According to Theorem 14, we know that a state q' can simulate approximately the state q with the precision $\lim_{t \rightarrow +\infty} \|A(t)\|_F \|M - A\|_F$. Thus, for each $q_i \in Q$, $\exists q'_i \in Q'$ simulates approximately state q_i with the precision $\lim_{t \rightarrow +\infty} \|A_i(t)\|_F \|M_i - A_i\|_F$. This means that each element of set Q has its approximate element with corresponding precision. But the set Q' generally approximates to the set Q with some other precision. Fortunately, we know that in this situation the precision between the set Q and the set Q' can be defined by the maximum precision of their elements. So the precision between set Q' of approximate states and set Q of states is

$$\delta = \max_i \left(\lim_{t \rightarrow +\infty} \|A_i(t)\|_F \|M_i - A_i\|_F \right), \quad (20)$$

$$i = 1, 2, \dots$$

But because the precision of the T' approximate simulation T is caused by set Q' of approximate state, Theorem 16 is reasonable.

6. A Case Study

Now let us show the approximate theorem by taking example of some of the computations by considering linear systems without disturbances. Consider the linear systems $\dot{x} = Ax$, where

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 1 \\ 1 & -1 & 1 \\ 2 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad (21)$$

and the initial state is $x(0) = (1 \ 0 \ 0)$.

The linear system is approximately simulated by means of ILU. Firstly, we let

$$A = \begin{pmatrix} 1 & 2 & 1 \\ 1 & -1 & 1 \\ 2 & 0 & 1 \end{pmatrix}, \quad (22)$$

where A is the matrix of the equations. The LU factors of A are

$$L' = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 2 & \frac{4}{3} & 1 \end{pmatrix}, \quad (23)$$

$$U' = \begin{pmatrix} 1 & 2 & 1 \\ 0 & -3 & 0 \\ 0 & 0 & -1 \end{pmatrix}.$$

In terms of Algorithm 1, the ILU algorithm gives

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 2 & 0 & 1 \end{pmatrix}, \quad (24)$$

$$U = \begin{pmatrix} 1 & 2 & 1 \\ 0 & -3 & 0 \\ 0 & 0 & -1 \end{pmatrix},$$

which is different from setting the (3, 2) element of the LU factors to zero.

Now, we use the method of ILU to solve the problem. The preconditioner

$$M = LU = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 2 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 1 \\ 0 & -3 & 0 \\ 0 & 0 & -1 \end{pmatrix} \quad (25)$$

$$= \begin{pmatrix} 1 & 2 & 1 \\ 1 & -1 & 1 \\ 2 & 4 & 1 \end{pmatrix}.$$

Then, mathematically, the equivalent system of the original linear system $\dot{x} = Ax$ is the preconditioned system $M^{-1}\dot{x} = M^{-1}Ax$. That is,

$$\begin{pmatrix} 1 & 2 & 1 \\ 1 & -1 & 1 \\ 2 & 4 & 1 \end{pmatrix}^{-1} \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} \quad (26)$$

$$= \begin{pmatrix} 1 & 2 & 1 \\ 1 & -1 & 1 \\ 2 & 4 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 & 2 & 1 \\ 1 & -1 & 1 \\ 2 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}.$$

In the example, we use ILU method to approximate simulation states, and the precise of approximate simulation states can be calculated. That is,

$$\begin{aligned}
 M - A &= \begin{pmatrix} 1 & 2 & 1 \\ 1 & -1 & 1 \\ 2 & 4 & 1 \end{pmatrix} - \begin{pmatrix} 1 & 2 & 1 \\ 1 & -1 & 1 \\ 2 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 4 & 0 \end{pmatrix}, \\
 \|M - A\|_F &= \sqrt{(0^2 + 0^2 + 0^2) + (0^2 + 0^2 + 0^2) + (0^2 + 4^2 + 0^2)} \\
 &= 4, \\
 \|A\|_F &= \sqrt{(1^2 + 2^2 + 1^2) + (1^2 + (-1)^2 + 1^2) + (2^2 + 0^2 + 1^2)} \\
 &= \sqrt{14}.
 \end{aligned} \tag{27}$$

So the precision of approximate simulation states is $\delta = \|A\|_F \|M - A\|_F = 4\sqrt{14} \approx 14.9666$, where the precision retains four significant numbers after the decimal point. If we want to use MILU to solve the problem of approximate simulation states, we just deal with A , that is, using Algorithm 2 to decomposition A .

7. Conclusions

In this paper, we extend the notion of approximate simulation relations. The semantic analysis of equivalent labeled transition systems is developed by equivalent labels. And an effective characterization of approximate simulation relations of states based on ILU is also developed, where error about the state set approximate is controlled. Furthermore, the technique can be used to approximately simulate labeled transition systems generated by continuous-time system. Finally, a case study of application in the context of linear systems without disturbances is shown.

The approximate simulation relation is a key performance consideration of systems. Our future work will focus on these two items in two steps: firstly, developing methods to approximate simulation linear systems with disturbances; secondly, solving the nonlinear systems.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grants nos. 11371003 and 11461006, the Natural Science Foundation of Guangxi under Grants nos. 2014GXNSFAA118359 and 2012GXNSFGA060003, the Science and Technology Foundation of Guangxi under Grant no. 10169-1, and the Scientific Research Project no. 201012MS274 from the Guangxi Education Department.

References

- [1] J. Kleijn, M. Koutny, and M. Pietkiewicz-Koutny, "Regions of Petri nets with a/sync connections," *Theoretical Computer Science*, vol. 454, pp. 189–198, 2012.
- [2] R. Milner, *A Calculus of Communicating Systems*, Springer, New York, NY, USA, 1980.
- [3] R. Milner, *Communication and Concurrency*, Prentice Hall, New York, NY, USA, 1989.
- [4] C. A. R. Hoare, *Communicating Sequential Processes*, Prentice Hall, 1985.
- [5] J. C. Baeten and W. P. Weijland, *Process Algebra*, Cambridge University Press, Cambridge, UK, 1990.
- [6] S. Tini, "Non-expansive ϵ -bisimulations for probabilistic processes," *Theoretical Computer Science*, vol. 411, no. 22–24, pp. 2202–2222, 2010.
- [7] S. Kramer, C. Palamidessi, R. Segala, A. Turrini, and C. Braun, "A quantitative doxastic logic for probabilistic processes and applications to information-hiding," *Journal of Applied Non-Classical Logics*, vol. 19, no. 4, pp. 489–516, 2009.
- [8] S. Georgievska and S. Andova, "Testing reactive probabilistic processes," *Electronic Proceedings in Theoretical Computer Science*, vol. 28, pp. 99–113, 2010.
- [9] M. Schröder and A. Simpson, "Representing probability measures using probabilistic processes," *Journal of Complexity*, vol. 22, no. 6, pp. 768–782, 2006.
- [10] S. Oh, "Geostatistical integration of seismic velocity and resistivity data for probabilistic evaluation of rock quality," *Environmental Earth Sciences*, vol. 69, no. 3, pp. 939–945, 2013.
- [11] M. Hennessy, "Exploring probabilistic bisimulations, part I," *Formal Aspects of Computing*, vol. 24, no. 4–6, pp. 749–768, 2012.
- [12] S. Andova, S. Georgievska, and N. Trčka, "Branching bisimulation congruence for probabilistic systems," *Theoretical Computer Science*, vol. 413, no. 1, pp. 58–72, 2012.
- [13] W. Du and S. Tang, "On the evolution rule for regional industrial structure based on the stochastic process theory," *Management Science and Engineering*, vol. 5, no. 3, pp. 55–60, 2011.
- [14] A. Hermann Reynisson, M. Sirjan, L. Aceto et al., "Modelling and simulation of asynchronous real-time systems using timed Rebeca," *Science of Computer Programming*, vol. 89, pp. 41–68, 2014.
- [15] R. Cardell-Oliver, "Conformance tests for real-time systems with timed automata specifications," *Formal Aspects of Computing*, vol. 12, no. 5, pp. 350–366, 2000.
- [16] M. M. Jaghoori, F. S. de Boer, T. Chothia, and M. Sirjani, "Schedulability of asynchronous real-time concurrent objects," *Journal of Logic and Algebraic Programming*, vol. 78, no. 5, pp. 402–416, 2009.
- [17] K. Lampka, S. Perathoner, and L. Thiele, "Analytic real-time analysis and timed automata: a hybrid methodology for the performance analysis of embedded real-time systems," *Design Automation for Embedded Systems*, vol. 14, no. 3, pp. 193–227, 2010.
- [18] A. Girard and G. J. Pappas, "Approximation metrics for discrete and continuous systems," Tech. Rep., Department of Computer & Information Science, 2005.
- [19] E. M. Clarke, O. Grumberg, and D. A. Peled, *Model Checking*, MIT Press, 2000.
- [20] R. Kechroud, A. Soulaïmani, Y. Saad, and S. Gowda, "Preconditioning techniques for the solution of the Helmholtz equation by the finite element method," *Mathematics and Computers in Simulation*, vol. 65, no. 4–5, pp. 303–321, 2004.

- [21] E. F. D'Azevedo, P. A. Forsyth, and W.-P. Tang, "Towards a cost-effective ILU preconditioner with high level fill," *BIT*, vol. 32, no. 3, pp. 442–463, 1992.
- [22] D. P. Young, R. G. Melvin, F. T. Johnson, J. E. Bussioletti, L. B. Wigton, and S. S. Samant, "Application of sparse matrix solvers as effective preconditioners," *SIAM Journal on Scientific and Statistical Computing*, vol. 10, no. 6, pp. 1186–1199, 1989.
- [23] T. Dupont, R. P. Kendall, and J. Rachford, "An approximate factorization procedure for solving self-adjoint elliptic difference equations," *SIAM Journal on Numerical Analysis*, vol. 5, pp. 559–573, 1968.
- [24] T. Gustafsson, "A class of first-order factorization methods," *BIT Numerical Mathematics*, vol. 18, pp. 142–156, 1978.
- [25] A. Girard and G. J. Pappas, "Approximation metrics for discrete and continuous systems," Tech. Rep. MS-CIS-05-10, Department of Computer Information Systems, University of Pennsylvania, 2005.
- [26] M. A. F. Araghi and A. Fallahzadeh, "Inherited LU factorization for solving fuzzy system of linear equations," *Soft Computing*, vol. 17, no. 1, pp. 159–163, 2013.
- [27] A. El Kacimi and O. Laghrouche, "Wavelet based ILU preconditioners for the numerical solution by PUFEM of high frequency elastic wave scattering," *Journal of Computational Physics*, vol. 230, no. 8, pp. 3119–3134, 2011.
- [28] G. Ewald, *Combinatorial Convexity and Algebraic Geometry*, Springer, 1996.
- [29] O. Axelsson, *Iterative Solution Methods*, Cambridge University Press, 1994.
- [30] G. J. Pappas, "Bisimilar linear systems," *Automatica*, vol. 39, no. 12, pp. 2035–2047, 2003.

