Multiscale System Modeling of Biochemical Pathways

Krishnendu Ghosh
Department of Computer & Information Tech.
Miami University
ghoshk@miamioh.edu

ABSTRACT

Querying by temporal logic as a reasoning mechanism on a system representing multiscale processes is important in understanding the details of multiscale processes, in particular in models of biochemical pathways. A novel formalism representing a system of multiscale biochemical pathways is described. The definitions of multiscale model in discrete domains are represented in the form of a labeled transition system. A polynomial time algorithm is constructed for identification of systems representing multiscale pathways.

General Terms

Theory

Keywords

Formal modeling, systems biology, model checking, algorithms

1. INTRODUCTION

Model checking in systems biology [2] has received significant interest in recent years. The emphasis is to represent of biological processes by finite state machines(FSM) and apply temporal logics to verify interesting biological relatoinships. A key computational problem in model checking is state explosion [5]. Several approaches have been created to address it. The challenge in modeling biological systems is imprecise and incomplete information. Additionally, the abstractions of biological models become large in terms of state space to account for interconnected biological processes executing at different orders of time scale. Biological processes executing at different orders of time scales in a system are *multiscale* processes. The system representing multiscale processes is a *multiscale system*. Communication between molecular processes occur at different time scale than in a cellular processes [6]. The need to create an integrated system for studying biological entities executing at multiple time scales, namely molecular, cellular and organic levels is essential for a detailed understanding of biology of the processes. In this paper, we describe modeling of biological multiscale processes in a system. We construct an algorithm to identify partial ordering of identical processes in two multiscale systems. The processes are represented as *labels* on the transitions of a labeled transition system.

We motivate the construction of our formalism for a biological system. In the example, the processes are biochemical pathways. Consider four biochemical pathways, $\mathcal{A}, \mathcal{B}, \mathcal{C}$ and $\mathcal{D}$ modifying chemicals V, W, X, Y and Z:

$$\begin{aligned} \mathcal{A} &: X \xrightarrow{\epsilon} Y. \\ \mathcal{B} &: Y \xrightarrow{M} X + Z. \\ \mathcal{C} &: Y \xrightarrow{N} X + V. \\ \mathcal{D} &: Y \xrightarrow{P} X + W. \end{aligned}$$

The notation, $C_s \xrightarrow{\alpha} C_p$ denotes a set of substrates, C_s in the presense of a catalyst (chemical), α produces a set of products, C_p. Also, $\alpha \in \{M, N, P, \epsilon\}$ where M, N, P denote catalysts and ϵ represents absence of a catalyst. Figure 1(a) shows a finite state machine (FSM) representing pathways $\mathcal{A}, \mathcal{B}, \mathcal{C}$ and $\mathcal{D}$. The initial state, S contains one mole of the chemicals X, M, N and P. The transition, with label a represents pathway $\mathcal{A}$ being executed consuming 0.25 moles of chemical, X. Similarly, transitions with labels b, c and d represent execution of pathways, $\mathcal{B}, \mathcal{C}$ and $\mathcal{D}$, respectively. Each transition represent consumption of 0.25 mole of the substrate of each of the reaction. Pathways, $\mathcal{B}, \mathcal{C}$ and $\mathcal{D}$ execute nondeterministically after completion of pathway $\mathcal{A}$ executing four times successively. Each state stores the concentration of the chemicals formed or consumed in the form of labels on the states of FSM. For simplicity, the state labels are not shown in Figure 1. The quantities of the chemicals present in the system are stored in the states and represented by labels on the states. In Figure 1(b), a FSM represents the system of four pathways similar to Figure 1(a). The transitions in Figure 1(b) represent consumption of 0.5 moles of substrates and hence, the number of transitions and states in the FSM are less. For example, the two successive transitions labeled, a in Figure 1(a) can be collapsed to one transition in Figure 1(b). Hence, the state between successive transitions labeled with a in Figure 1(a) is not in Figure 1(b). The FSMs of Figure 1(a) and Figure 1(b) represent identical partial ordering of the pathways but the state space is different. A path in a FSM is of the form $s_1, e_1, s_2, e_2 \ldots$ where $s_1, s_2, \ldots$ and $e_1, e_2, \ldots$ represent states and edges, respectively. Figure 1(c) represents a FSM without successive pathways in any path. Therefore, for every path in the FSMs of Figure 1(a) and Figure 1(b) with representation successive pathways as a single pathway there exists a

BICT 2014, December 01-03, Boston, United States

DOI 10.4108/icst.bict.2014.258008

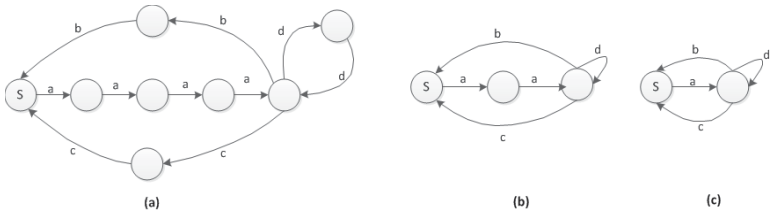

Figure 1: Finite state machine representing identical partial ordering of pathways $\mathcal{A}, \mathcal{B}, \mathcal{C}$ and $\mathcal{D}$ represented by edge labels a,b,c and d, respectively. (a) System of pathways with transitions representing .25 mole of the substrate consumption. (b) System of pathways with transitions representing 0.5 mole of the substrate consumption. (c) System of pathways without identical successive pathways.

path in Figure 1(c) and vice versa. Formally, we construct a preorder relation to capture the partial ordering of the pathways represented by the edges on the FSMs in Figure 1(a)-(c). We consider only the edge labels on the FSM to evaluate the identifiability of partial ordering of pathways. The example shows the FSM in Figure 1(a) is detailed and has a larger state space than the FSM in Figure 1(b). Temporal logics queries posed on FSM in Figure 1(b) will be efficient because of smaller state space. The answers to the query in temporal logics formula, φ such as LTL and CTL, if φ does not invlove the **X** (next state) operator, would not be changed. The example is a drastic simplification of reality because there could be more than four pathways and the exact number of moles of substrates before one or multiple different pathways intiate is not known. The consumption of moles represented by the transitions in the FSMs in Figure 1(a) and Figure 1(b) are implicitly modeling time taken by a pathway to consume the substrate of a pathway before the execution of the next pathway that uses the products of the first pathway as its substrate. We design a novel formalism that is natural and succinct for modeling multiscale processes in a system. The incomplete and imprecise information of initial concentrations of the species could lead to different multiscale processes with different state labels (though the partial ordering of the edge labels maybe the same). We formulate the *problem* to identify the partial ordering on edge labels of two FSMs representing the multiscale scale systems. Identification of the processes is based on the edge labels of the FSMs only. Incomplete information of biological data is addressed by nondeterminism of the transitions from a state in the FSM. The contributions of this work: (i) Design of a nondeterministic system model representing multiscale pathways and (ii), a polynomial time algorithm that is able to identify the partial ordering (equivalences) of pathways of two multiscale systems.

2. BACKGROUND AND PRIOR WORK

We review the literature on stuttering on systems, algorithms for computing equivalences on Kripke structures, asynchronous modeling and temporal logics in biological systems. These are different theories but form the basis of our work. Stuttering on systems had been mentioned by Lamport [10] for modeling concurrent programs and reasoning by temporal logics.

Definition 1. (Kripke structure) Given a set of propositions, AP, a *Kripke structure*, $\mathcal{K} = \langle S, S_0, E, L \rangle$ consists of

1. S is the set of states.
2. $S_0 \subseteq S$ is the initial set of states.
3. $E \subseteq S \times S$ is the transition relation.
4. $L : S \rightarrow 2^{AP}$ where L is the labeling function that labels each state with a subset from the set, AP.

Definition 2. (Stuttering Equivalence [3] on Paths) Two infinite paths in Kripke structure $\mathcal{K}$, $\mu = s_o \overset{\alpha_0}{\rightarrowtail} s_1 \overset{\alpha_1}{\rightarrowtail} s_2 \ldots$ and $\nu = r_0 \overset{\beta_0}{\rightarrowtail} r_1 \overset{\beta_1}{\rightarrowtail} \ldots$ are stuttering equivalent ($\equiv_s$)ent if there are two infinite ordered sequences of positive integers, $i = 0 < i_0 < i_1 < \ldots$ and $j = 0 < j_0 < j_1 < \ldots$ such that $\forall k \geq 0\ L(s_{i_k}) = L(s_{i_k+1}) = \ldots = L(s_{i_{k+1}-1}) = L(r_{j_k}) = L(r_{j_k+1}) = \ldots = L(r_{j_{k+1}-1})$. The indices i_k and j_k are the starting points of μ and ν, respectively.

Definition 3. (Stuttering Equivalence [3]) Two Kripke structures $\mathcal{K}$ and $\mathcal{K}'$ are stuttering equivalent iff

1. The initial states of $\mathcal{K}$ and $\mathcal{K}'$ are the same.
2. For all paths, μ from an initial state, $s_0 \in S_0$ of $\mathcal{K}$, there exists a path ν of $\mathcal{K}'$ from the same initial state of s_0 such that $\mu \equiv_s \nu$.
3. For all paths, ν from an initial state of $s_0 \in S_0$ of $\mathcal{K}'$, there exists a path μ of $\mathcal{K}$ from the same initial state of s such that $\nu \equiv_s \mu$.

A deterministic asynchronous model [4] was defined on a FSM to model interleaving processes. The processes on asynchronous system were the state labels and the paths of the state transition systems were stuttering equivalent. The state space was reduced by partial ordered methods that characterized transitions to be *invisible* if the transitions connected states with identical labels. Process algebra modeling of multiscale biological systems has been addressed with construction of bisimulation relations [6], rate of reaction based on stochastic semantics [1], semantic equivalences [8] and Stochastic Pi Machine [12]. Model reduction techniques such as zooming of states that allow reduction in model state space size and retrieval of the original model with the aim of studying the model at different levels of granularity has been stated [14]. Fisher and colleagues [7] describe a *bounded* asychronous model that had a scheduling mechanism. The scheduling mechanism controlled the number of executions of the processes with the objective of a single time scale among the execution of the processes.

The scheduler introduced a form of nondeterminism in the system by selecting the next process for execution. Processes represented by genes modeled with timed automata was described [13] and the model addressed "process A and process B synthesizes same amount of product in different time scales".
Computing bisimulation on structures [11] is an established research area. Computation of stuttering bisimulation are described [9] on a related problem, relational coarsest partition with stuttering(RCPS) problem. The algorithms for computing bisimulation equivalences are have been used to identify deterministic structures [3].

3. MODEL OF MULTISCALE PROCESSES

We formalize the model to represent multiscale processes.

Definition 4. (Labeled transition system (LTS)) Given a set of propositions, AP being the set of labels for states and EL, a set of labels for edges a *labeled state transition system* is defined as $\mathcal{M} = \langle S_0, S, E, L_e, L\rangle$ where,

1. $\langle S, S_0, E, L\rangle$ forms a Kripke structure.
2. $L_e : E \mapsto EL$ is an edge-labeling function.

We are given (1) a set $\mathcal{C}$ of chemicals, (2) for each chemical $C \in \mathcal{C}$ a finite set of numbers, $0, 1, \ldots, k$ where $k \in \mathbb{N}$ represent the number of moles for each chemical C and (3) a set of reaction tuples, $\mathcal{R}tuple$. The LTS, $\mathcal{M}_e = \langle S, S_0, E, L_e, L\rangle$ representation of a system of chemical reactions is given by:

- AP is the set of all the atomic formulas $c_0 = 0$, $c_1 = 1$, or $c_i = k$ for all $C \in \mathcal{C}$, $i \in \mathbb{N}$. The atomic formula represent the molar concentration of a chemical.
- S is the set of all subsets s of AP where, for each $C \in \mathcal{C}$, exactly one of the formulas $c_0 = 0$, or $c_1 = 1, \ldots,$ or $c_i = k$, is in s. The states contain concentrations of all the chemicals in the system and represented by the atomic formulas of each chemical that are true in the state.
- S_0 is the set of initial states of the LTS. An initial state contains concentration of all the chemicals before any reaction. Hence, $\mid S_0 \mid = 1$.

The label(edge label) on a transition is a reaction tuple, $rtup \in \mathcal{R}tup$. A labeled transition is represented, $s \xrightarrow{e} s'$ where $e = rtup$. A r-transition implies that a reaction is taking place and the consumption(production) of substrates(products) is based on the x, number of moles to be consumed in a unit of time. A transition represents a time step. The number of moles consumed or produced is computed using conservation of mass action. An ϵ-edge represents no reaction from a state. It maintains the LTS to be total. The LTS representation of a system of chemical reactions creates a nondeterministic structure. The reactions are modeled in successive transitions if there are minimal substrates to start the reaction in successive states of a path in the LTS. The label on an edge, $e \in E$ is given by $L_e(e) = \alpha$ and is written as: $s \xrightarrow{\alpha} s'$. A path in the labeled state transition system is finite or infinite sequence $\sigma = s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} \ldots$. We can think of a path as a sequence of edges. For an edge, e in a LTS $\mathcal{M}$, let $\Pi(e)$ be the set of paths starting with e, and let $\Pi(\mathcal{M})$ be the set of all paths in $\mathcal{M}$. We use variable π_e for an element of $\Pi(e)$. A prefix of length m of a path π_{e_1}, beginning from edge, e_1 is a finite sequence, $\pi^m_{e_1} = e_{0,1}, e_{1,1}, \ldots, e_{m-1,1}$ where $m \in \mathbb{N}$. We compare two LTSs, $\mathcal{M}_1$ and $\mathcal{M}_2$. S_i, E_i and $\Pi(\mathcal{M}_i)$ denote the set of states, edges and paths in $\mathcal{M}_i$. When we refer to edge e_i, unless we state otherwise, we mean that $e_i \in E_i$ and $i = 1, 2$ for different LTSs.
The edge labels on the LTS represent the individual processes. The labels on the states are the quantities of the chemicals. An example with reference to biological system model: a path fragment in $\mathcal{M} = \langle S, S_0, E, L_e, L\rangle$ is given by e, s, e' where $e, e' \in E, s \in S, L_e(e) = \mathcal{A}, L_e(e') = \mathcal{B}$ and $L(s) = \{a, b, c\}$ implies the quantities of biochemicals formed are a, b and c after the execution of process, $\mathcal{A}$ and before execution of process y. In the formalism, we consider only the edge labels (processes) in the LTS. A path segment of the form $e_1 \rightarrowtail e_2 \rightarrowtail e_3 \rightarrowtail e_4$ where $L_e(e_1) = L_e(e_2) = L_e(e_3) = \mathcal{A}$ and $L_e(e_4) = \mathcal{B}$ represents execution of process $\mathcal{A}$ 3 times before process $\mathcal{B}$ can start execution. For a different set of initial quantities of the species, the order of pathways could be different such as execution of process $\mathcal{A}$ only once before execution of process $\mathcal{B}$. Hence, computationally, a construct that identifies the *stutter* e_1, e_2, e_3 is useful in capturing the notion that y executes after x. The ordering of the processes in two different systems is important to identify multiscale systems modeling identical pathways. We define the following constructs to identify the order of processes in the paths in a LTS, motivated from the definitions in section 2.

Definition 5. (Path signature) For an infinite path, $\pi = e_0, e_1, e_2, e_3, \ldots$ in a labeled state transition system $\mathcal{M}, (\alpha_0, \alpha_1, \alpha_2, \ldots)$ is the sequence of edge labels in π. The path signature is the subsequence of labels $\tilde{\pi} = \alpha_0, \alpha_{i_1}, \alpha_{i_2}$ where $0 \leq i_1 \leq i_2 \leq \ldots$, α_{i_j} is in $\tilde{\pi}$ iff $\alpha_{i_j} \neq \alpha_{i_{j-1}}$ and $\alpha_0 \neq \alpha_{i_1}$.

Definition 6. (Path signature equivalence on paths) Paths $\pi \in \Pi(\mathcal{M}), \pi' \in \Pi(\mathcal{M}')$ are path signature equivalent iff their path signatures are identical and denoted by $\pi \equiv_{psig} \pi'$.

Definition 7. (Path signature on edges) Given two LTSs, $\mathcal{M}_1$ and $\mathcal{M}_2$, the relation path signature on edges ($\equiv_{psig}$) is defined on edges $e_1 \in E_1$ and $e_2 \in E_2$. $e_1 \equiv_{psig} e_2$ if and only if the following conditions hold:

1. $L_e(e_1) = L_e(e_2)$.
2. For all paths, $\pi_{e_1} \in \Pi(e_1)$ $\exists$ a path $\pi_{e_2} \in \Pi(e_2)$ such that $\pi_{e_1} \equiv_{psig} \pi_{e_2}$.
3. For all paths, $\pi_{e_2} \in \Pi(e_2)$ $\exists$ a path $\pi_{e_1} \in \Pi(e_1)$ such that $\pi_{e_1} \equiv_{psig} \pi_{e_2}$.

Definition 8. A relation, R_e defined on the edges of $\mathcal{M}_1$ and $\mathcal{M}_2$ is given by $(e_1, e_2) \in R_e, e_1 \in E_1$ and $e_2 \in E_2$ where, $L_e(e_1) = L_e(e_2)$.

4. COMPUTING EQUIVALENCES ON LTS

In a LTS, $\mathcal{M}$, a path is *stuttering* if it has a block of successive edges with same edge labels. A finite path segment $\sigma = e_0 \rightarrowtail e_1 \rightarrowtail e_2 \rightarrowtail e_3 \ldots \rightarrow e_m \rightarrowtail \ldots$, is identically labeled (il) if the labels of all the edges are identical. We explicitly allow $m = 0$; in that case we write $e_0 \rightsquigarrow e_0$. Notation $e_0 \overset{+}{\rightarrowtail} e'$ means that for some $m \geq 0, e_0 \rightsquigarrow e_m \rightarrowtail e'$, and $L_e(e_0) \neq L_e(e')$. Notation for $e_1 = e_{0,1}$ and $e_2 = e_{0,2}$ for the paths beginning from e_1 and e_2.

Definition 9. The subset of ordered pairs, $Pre^{st}(Y)$ defined from the set of ordered pairs, $(e_1, e_2) \in R_e$ represented by the Y is: $Pre^{st}(Y) = \{(e_1, e_2) \in Y \mid \forall e'_1, e_1 \rightarrowtail e'_1$implies $\exists$ an il-path segment $e_2 \rightarrowtail \ldots \rightarrowtail e_{m,2} \rightarrowtail e'_2, \forall i \leq m, (e_1, e_{i,2}) \in Y \wedge (e'_1, e'_2) \in Y,$ and $\forall e'_2, e_2 \rightarrowtail e'_2$ implies $\exists$ an il-path segment $e_1 \rightarrowtail \ldots \rightarrowtail e_{m,1} \rightarrowtail e'_1, \forall i \leq m, (e_{i,1}, e_2) \in Y \wedge (e'_1, e'_2) \in Y\}$.

The algorithm to compute fixed point for two labeled transition structures allowing stuttering on the edge labels is based on Fixed Point Computation algorithm. The input of the algorithm is R_e as stated earlier.

Algorithm 1 Fixed Point Computation of Path Signature

Input: Set of Ordered Pairs,R_e
Output: Set of ordered pairs in the greatest fixed point,Y_∞.

```
1: Y := R_e;
2: Y' = 0;
3: while (Y ≠ Y')
4:     {
5:         Y' := Y;
6:         Y := Y ∩ Pre^st(Y);
7:     }
8: Y_∞ = Y'
```

Lemma 1. *The algorithm terminates after finite number of steps and computes fixed point, given by* $Y = Pre^{st}(Y)$.

Proof. The loop that begins in line (3) takes finite number of steps,$i \in \mathbb{N}$ for the algorithm to terminate because there are finite number of ordered pairs of edges in R_e.
Claim: The algorithm computes the fixed point, i.e $Y = Pre^{st}(Y)$. Let Y_∞ be the set of ordered pairs at the end of the loop and $Y_\infty = Y' = Y$. By definition of the set, $Y' = \{(e_1, e_2) \mid e_1 \in E_1, e_2 \in E_2, L_e(e_1) = L_e(e_2)\}$. For every $(e_1, e_2) \in Y'$ implies $(e_1, e_2) \in Y$ because at the end of the loop, $Y_\infty = Y' = Y$. The statement in line(6) in the algorithm, every $(e_1, e_2) \in Y$ implies $(e_1, e_2) \in Pre^{st}(Y)$. Each $(e_1, e_2) \in Pre^{st}(Y)$ implies through definition 9 that $(e_1, e_2) \in Y$. Therefore, $Y = Pre^{st}(Y)$.

The time complexity of the algorithm is $O(m^2)$ where $m = \mid R_e \mid$. In the worst case, the set of ordered pairs in $Pre^{st}(Y)$ is constructed by removing a pair (e_1, e_2) at a time . The while loop iterates m times over m computations in $Pre^{st}(Y)$. The above algorithm computing the greatest fixed point is based on the following recursive relation, Y_i defined on the ordered pairs (e_1, e_2) :
$Y_{i+1} = Y_i \cap Pre^{st}(Y_i)$, where $Y_0 = \{(e_1, e_2) \mid L_e(e_1) = L_e(e_2)\}$. The greatest fixed point is the first $i \in \mathbb{N}$ such that $Y_\infty = Y_{i+1} = Y_i$. The following is the correctness proof of the Fixed Point Computation algorithm.

Definition 10. (Path Signature i-Length Stutter Equivalence) The relation path signature i- length equivalence ($\equiv^i_{stpsig}$) is defined on for all $e_1 \in E_1$ and $e_2 \in E_2$ where $i \in \mathbb{N}$. $e_1 \equiv^i_{stpsig} e_2$ iff the following conditions hold:

1. $L_e(e_1) = L_e(e_2)$.
2. For every e'_1 there exists e'_2 such that $e_1 \rightarrowtail e'_1$, $e_2 \overset{+}{\rightarrowtail} e'_2$ and $(e'_1, e'_2) \in Y_{i-1}$.
3. For every e'_2 there exists e'_1 such that $e_1 \overset{+}{\rightarrowtail} e'_1$, $e_2 \rightarrowtail e'_2$ and $(e'_1, e'_2) \in Y_{i-1}$.

Lemma 2. *If* $e_1 \equiv^{i+1}_{stpsig} e_2$ *then* $e_1 \equiv^i_{stpsig} e_2$.

Proof. We prove by cases:
Case: $i = 0$. $e_1 \equiv^1_{stpsig} e_2$ implies $L_e(e_1) = L_e(e_2)$. Therefore, $e_1 \equiv^0_{stpsig} e_2$.
Case: $i > 0$. By condition (2) of definition 10, $e_1 \equiv^{i+1}_{stpsig} e_2$ implies $(e_{i,1}, e_{i,2}) \in Y_i$ where $e_1 \rightarrowtail e_{i,1}, e_2 \overset{+}{\rightarrowtail} e_{i,2}$. By the relation, $Y_{i+1} = Y_i \cap Pre^{st}(Y_i)$, implies $(e_{i,1}, e_{i,2}) \in Pre^{st}(Y_i)$. By definition 9, it implies $(e_{i,1}, e_{i,2}) \in Y_{i-1}$. Hence, $e_1 \equiv^i_{stpsig} e_2$. By identical reasoning, condition(3) of definition 10, $(e'_{i,1}, e'_{i,2}) \in Y_{i-1}$.

Lemma 3. *If* $(e_1, e_2) \in Y_{i+1}$ *then* $e_1 \equiv^{i+1}_{stpsig} e_2$.

Proof. By definition of $Y_{i+1} = Y_i \cap Pre^{st}(Y_i)$, $(e_1, e_2) \in Y_{i+1}$ imply $(e_1, e_2) \in Y_i$ and $(e_1, e_2) \in Pre^{st}(Y_i)$. By definition 9, for all e'_1 there exists e'_2 such that $e_1 \rightarrowtail e'_1, e_2 \overset{+}{\rightarrowtail} e'_2, (e'_1, e'_2) \in Y_i$. Conditions (1) and (2) of definition 10, $e_1 \equiv^{i+1}_{stpsig} e_2$.
By identical reasoning, for all e'_2 there exists e'_1 such that $(e'_1, e'_2) \in Y_i$. Conditions (1) and (3) of definition 10 implies $e_1 \equiv^{i+1}_{stpsig} e_2$.

Lemma 4. *If* $e_1 \equiv^{i+1}_{stpsig} e_2$ *then* $(e_1, e_2) \in Y_{i+1}$.

Proof. By lemma 2, $e_1 \equiv^{i+1}_{stpsig} e_2$ then $e_1 \equiv^i_{stpsig} e_2$. We want to show, if $e_1 \equiv^i_{stpsig} e_2$ then $(e_1, e_2) \in Y_{i+1}$.. Assume for all $k \leq i$, if $e_1 \equiv^k_{stpsig} e_2$ then $(e_1, e_2) \in Y_k$. By condition (2) of definition 10, $e_1 \equiv^k_{stpsig} e_2$ implies every e'_1 there exists e'_2 such that $e_1 \rightarrowtail e'_1, e_2 \overset{+}{\rightarrowtail} e'_2, (e_1, e_2) \in Y_{k-1}$ and $(e'_1, e'_2) \in Y_{k-1}$. By definition 9, $(e_1, e_2) \in Pre^{st}(Y_k)$. By induction hypothesis, $(e_1, e_2) \in Y_k$. $(e_1, e_2) \in Y_{k+1}$ by $Y_{k+1} = Y_k \cap Pre^{st}(Y_k)$. Therefore, $(e_1, e_2) \in Y_{i+1}$.

Theorem 1. *(Invariant for Algorithm) For all ordered pairs,* $(e_1, e_2) \in Y_i$ *iff* $e_1 \equiv^i_{stpsig} e_2$.

Proof. If $(e_1, e_2) \in Y_i$ then $e_1 \equiv^i_{stpsig} e_2$ is true by lemma 3. Conversely, by lemma 4, $e_1 \equiv^i_{stpsig} e_2$ implies for all ordered pairs, $(e_1, e_2) \in Y_i$.

Theorem 2. $e_1 \equiv_{psig} e_2$ *iff* $\forall i \in \mathbb{N}, e_1 \equiv^i_{stpsig} e_2$.

Proof. Assume $e_1 \equiv_{psig} e_2$. Condition (1) of the definition 7 implies condition (1) of the definition 10. Given all paths, $\pi_{e_1} \in \Pi(e_1)$ $\exists$ a path $\pi_{e_2} \in \Pi(e_2)$ such that $\pi_{e_1} \equiv_{psig} \pi_{e_2}$. Let $\pi_{e_1} \in \Pi(e_1)$. The number of edges in $\mathcal{M}_1$ and $\mathcal{M}_2$ are finite. Hence, the paths π_{e_1} and π_{e_2} have finite number of distinct edges . The edges, $e_{x,1}$ and $e_{y,2}$ where $x, y \in \mathbb{N}$ are the last edges before it forms cycle in the paths, π_{e_1} and π_{e_2} (Notation of finite path, $\approxeq$). Therefore, $\pi_{e_1} \approxeq e_1 \rightarrowtail$

$e_{1,1} \ldots \rightarrowtail e_{x,1}$. Similarly, $\pi_{e_2} \cong e_2 \overset{+}{\rightarrowtail} e_{1,2} \ldots \overset{+}{\rightarrowtail} e_{y,2}$. The prefixes of the paths, π_{e_1} and π_{e_2} are given by $\pi^j_{e_a}$ where $e_a \in \{e_1, e_2\}$ and $j \in \mathbb{N}$. Condition (2) of the definition 7 states $\pi_{e_1} \equiv_{psig} \pi_{e_2}$ implies the prefixes of the paths, $\pi^i_{e_1} \equiv_{psig} \pi^i_{e_2}$. By induction on the lengths of prefixes of the paths, π_{e_1} and π_{e_2} for $i = x, x-1, \ldots, 1$ and $e_1 \equiv_{psig} e_2$:
For every $e_{i-1,1}$ there exists $e_{i-1,2}$ such that $e_{i,1} \rightarrowtail e_{i-1,1}$, $e_{i,2} \overset{+}{\rightarrowtail} e_{i-1,2}$ and $(e_{i-1,1}, e_{i-1,2}) \in Y_{i-1}$. By induction and condition (2) for the definition 10 holds true implying, $e_1 \equiv^i_{stpsig} e_2$.
Conversely, assume $e_1 \equiv^i_{stpsig} e_2$ for all $i \in \mathbb{N}$. Condition (1) in the definition 10 implies condition (1) in the definition 7. We show condition(2) of the definition 10 implies condition (2) of the definition 7. Proof by cases:

Case:(Finite Paths:) For $i \in \mathbb{N}$, construct path,π_{e_2} iteratively from relation $e_1 \equiv^i_{stpsig} e_2$ implies $e_1 \equiv_{psig} e_2$ for paths π_{e_1} and π_{e_2} for finite length.

Case:(Infinite Paths) We prove the following:
Claim: If $(e_1, e_2) \in Y_\infty$ then for every infinite $\pi_{e_1} \in \Pi(e_1)$, there exists $\pi_{e_2} \in \Pi(e_2)$ such that $e_1 \equiv_{psig} e_2$. Here, $Y_\infty = Y_{i+1} = Y_i$. Let $Y_\infty = Y_k, k \in \mathbb{N}$.
We prove the claim by constructing a path $\pi_{e_2} \in \Pi(e_2)$ starting from e_2 such that $(e_1, e_2) \in Y_\infty$. By condition (2) of definition 10, for every e'_1 there exists e'_2 such that $e_1 \rightarrowtail e'_1$, $e_2 \overset{+}{\rightarrowtail} e'_2$ and $(e'_1, e'_2) \in Y_{i-1}$. For $i = 0, 1, \ldots \leq k$, the path signature of π_{e_2} is given by $e_{0,2} \overset{+}{\rightarrowtail} e_{1,2} \ldots \overset{+}{\rightarrowtail} e_{p,2}$ whenever $\pi_{e_1} = e_{0,1} \rightarrowtail e_{1,1} \ldots \rightarrowtail e_{k,1}$ and $(e_{i,1}, e_{i,2}) \in Y_i$. The path signature of π_{e_2} is constructed iteratively : $e_{0,2} \overset{+}{\rightarrowtail} e_{1,2} \overset{+}{\rightarrowtail} \ldots$ where $(e_1, e_2) \in Y_i$,$(e_{1,1}, e_{1,2}) \in Y_{i-1}, \ldots$. For each path segment of the form, $e_{j,2} \overset{+}{\rightarrowtail} e_{j+1,2}, j \in \mathbb{N}$ in the path signature of π_{e_2} and by the definition of *il*-path segment, there exists a finite path segment such that $e_{j,2} \rightarrowtail d_1 \rightarrowtail \ldots \rightarrowtail d_m \rightarrowtail e_{j+1,2}$. , $d_m \in E_2$ and by definition of 9, $(e_{j,1}, d_m) \in Y_i$. Iteratively, *il*- path segments are constructed for each path segment, $e_{j,2} \overset{+}{\rightarrowtail} e_{j+1,2}$ in the path signature of π_{e_2}. Hence, $\pi_{e_2} = e_{0,2} \rightarrowtail d_1 \ldots \rightarrowtail d_m \rightarrowtail e_{1,2}, \ldots$. Therefore, $e_1 \equiv_{psig} e_2$. We show the path π_{e_2} constructed from the infinite path, π_{e_1} and Y_i is infinite. Let $\pi^m_{e_1}$ and $\pi^m_{e_2}$ represent the prefix of length $m \in \mathbb{N}$ of the path π_{e_1} and π_{e_2}, respectively. By above reasoning, $\pi^m_{e_1} \equiv_{psig} \pi^m_{e_2}$. Since π_{e_1} is infinite and $\forall e_{m,1} \in \{e_{m,1}, e_{m+1,1}, \ldots\}$, there exists $e_{m,2}, e_{m,2} \rightarrowtail e_{m+1,2}$. such $(e_{m,1}, e_{m,2}) \in Y_i$. This can only be true if π_{e_2} is infinite. Hence, for every edge $e_1 \in \pi_{e_1}$ there exists an $e_2 \in \pi_{e_2}$ such that $(e_1, e_2) \in Y_i$.
The reasoning for the cases of finite and infinite path and condition(2) of the definition 10 imply condition (2) of the definition 7. By similar reasoning, if $(e_1, e_2) \in Y_\infty$ then for every infinite $\pi_{e_2} \in \Pi(e_2)$, there exists $\pi_{e_1} \in \Pi(e_1)$ such that $e_1 \equiv_{psig} e_2$.
We showed condition(2) of the definitions are equivalent. Condition(3) of the definitions are equivalent by similar reasoning. □

5. CONCLUSION

In this paper we created a representation of labeled transition system to model multiscale processes. We defined the notion of path signature equivalence that was able to relate behavior of two multiscale systems with different levels of granularity. A polynomial time algorithm is constructed to identify two systems representing identical partial ordering of processes. Future work includes a rigorous evaluation on biologial pathways using experimental data.

6. ACKNOWLEDGEMENT

The author is grateful to Professor John Schlipf for his suggestions related to this paper.

7. REFERENCES

[1] R. Barbuti, G. Caravagna, A. Maggiolo-Schettini, P. Milazzo, and S. Tini. Foundational aspects of multiscale modeling of biological systems with process algebras. *Theoretical Computer Science*, 431:96–116, 2012.

[2] N. Chabrier-Rivier, M. Chiaverini, V. Danos, F. Fages, and V. Schächter. Modeling and querying biomolecular interaction networks. *Theoretical Computer Science*, 325(1):25–44, 2004.

[3] E. Clarke, E. Emerson, and A. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 8(2):244–263, 1986.

[4] E. Clarke, O. Grumberg, M. Minea, and D. Peled. State space reduction using partial order techniques. *International Journal on software tools for technology transfer (STTT)*, 2(3):279–287, 1999.

[5] E. Clarke, O. Grumberg, and D. Peled. *Model checking*. MIT press, 2000.

[6] A. Degasperi and M. Calder. Multi-scale modelling of biological systems in process algebra with multi-way synchronisation. In *Proceedings of the 9th International Conference on Computational Methods in Systems Biology*, pages 195–208. ACM, 2011.

[7] J. Fisher, T. Henzinger, M. Mateescu, and N. Piterman. Bounded asynchrony: Concurrency for modeling cell-cell interactions. *Formal Methods in Systems Biology*, pages 17–32, 2008.

[8] V. Galpin and J. Hillston. A semantic equivalence for bio-pepa based on discretisation of continuous values. *Theoretical Computer Science*, 412(21):2142–2161, 2011.

[9] J. Groote and F. Vaandrager. An efficient algorithm for branching bisimulation and stuttering equivalence. *Automata, Languages and Programming*, pages 626–638, 1990.

[10] L. Lamport. What good is temporal logic. *Information processing*, 83:657–668, 1983.

[11] R. Paige and R. E. Tarjan. Three partition refinement algorithms. *SIAM Journal on Computing*, 16(6):973–989, 1987.

[12] V. Sharma and A. Compagnoni. Computational and mathematical models of the JAK-STAT signal transduction pathway. In *Proceedings of the 2013 Summer Computer Simulation Conference*, page 15. Society for Modeling & Simulation International, 2013.

[13] H. Siebert and A. Bockmayr. Temporal constraints in the logical analysis of regulatory networks. *Theoretical Computer Science*, 391(3):258–275, 2008.

[14] M. Sunnåker, H. Schmidt, M. Jirstrand, and G. Cedersund. Zooming of states and parameters using a lumping approach including back-translation. *BMC systems biology*, 4(1):28, 2010.