Proceedings of the 2nd International Workshop on Biological Processes & Petri Nets (BioPPN2011)
online: http://ceur-ws.org/Vol-724 pp.87-101

Two modeling methods for signaling pathways with multiple signals using UPPAAL

Shota Nakano and Shingo Yamaguchi

Graduate School of Science and Engineering, Yamaguchi University
2-16-1 Tokiwadai, Ube, 755-8611, Japan
E-mail : {r033vk, shingo}@yamaguchi-u.ac.jp

Abstract. Model checking has been attracting attention to analyze signaling pathways. There are two or more, i.e. multiple, signals flowing in a signaling pathway. Most of previous work, however, have treated only one, i.e. single, signal. To analyze a signaling pathway more precisely, it is necessary to treat multiple signals. There is few previous work treating multiple signals. It is known that a primary issue in model checking is the state-space explosion. Multiple signals make it difficult to analyze by model checking. In this paper, we propose two modeling methods for signaling pathways with multiple signals. These methods transform a Petri net model of a signaling pathway to an automaton model of UPPAAL. The first method uses multiple automata as a model of UPPAAL. The second method uses a single automaton as a model of UPPAAL. We apply these methods to an example. And we find that the single automaton modeling method is more effective than the multiple automata modeling method from the viewpoint of the number of signals, the number of states explored, and checking time. These results show that the model size to be analyzed is improved by devising of modeling method.

1 Introduction

Signaling pathway is a signaling mechanism to unify the behavior of cells. Since a pathway is large and complex, there are unknown mechanisms and components. Some researchers have applied model checking to analysis of signaling pathways. Model checking is an automatic and usually quite fast verification technique for finite state concurrent systems. In 2003, Chabrier et al.[1] applied NuSMV to analysis of biological pathways, and in 2009, Bos et al.[2] applied UPPAAL to analysis of a signaling pathway. They also described that model checking is powerful, but the state-space explosion may happen. Note that both [1] and [2] treated only one signal.

There are two or more, i.e. multiple, signals in a signaling pathway because a ligand joins a receptor repeatedly. To analyze the signaling pathway more precisely, it is necessary to treat multiple signals. There is few previous work treating multiple signals. Kwiatkowska et al.[3] described that model checking of multiple signals cause the state-space explosion further than that of a single signal in performing model checking by PRISM.

In this paper, we propose two modeling methods for signaling pathways with multiple signals. These methods transform a Petri net model of a signaling pathway to an automaton model of UPPAAL. Cassez et al.[4] proposed a method for transforming a time Petri net to an timed automaton. But this method tends to increase the size of an automaton. Our methods can give a smaller automaton model. The first method uses multiple automata as a model of UPPAAL. The second method uses a single automaton as a model of UPPAAL. We apply these methods to an example. Then we evaluate the methods from the viewpoint of the number of signals, the number of states explored, and checking time. Finally, we discuss whether we can increase the model size to be analyzed by devising of modeling method.

In Sect.2, we present a Petri net model of signaling pathways and an automaton model of UPPAAL. In Sect.3, we propose two modeling method for a signaling pathway with multiple signals. In Sect.4, we apply the two proposed methods to an example and evaluate the two methods. Finally, Sect.5 gives a conclusion and some future work.

2 Preliminary

2.1 Petri net model

Matsuno et al.[5, 6] have proposed a Petri net model of signaling pathways. The Petri net model can represent multiple signals as multiple tokens. Places denote static elements including chemical compounds, conditions, states, substances and cellular organelles. Tokens indicate the presence of these elements. The number of tokens is given to represent the amount of chemical substances. Transitions denote active elements including chemical reactions, events, actions, conversions and catalyzed reactions. Directed arcs connecting the places and the transitions represent the relations between corresponding static elements and active elements.

The formal definition of a Petri net model of a signaling pathway is as follows.

Definition 1. *A Petri net model of a signaling pathway is a 5-tuple* $TPNR = (T, P, \mathcal{E}, D, R)$.

T : *A set of transitions* $\{t_1, t_2, \cdots, t_{|T|}\}$

P : *A set of places* $\{p_1, p_2, \cdots, p_{|P|}\}$

$\mathcal{E}$: *A set of directed arcs between the places and the transitions*

$D : T \to \mathbb{N}$: *A function to assign a firing delay time to a transition*

$R : T \to [0, 1]$: *A function to assign a firing rate to a transition. Note that* $\sum_{t \in p^\bullet} R(t) = 1$. □

There are one or more source transitions and one or more sink transitions. Every node is on a path from a source transition to a sink transition. If a firing of a transition t_i is decided, tokens required for the firing are reserved. We call these tokens as reserved tokens. When $D(t_i)$ passes, t_i fires to remove the reserved tokens from each input place of t_i and put non-reserved tokens into each output place of t_i. If a place p_i has two or more output transitions, each output transition can't fire over $R(t_i)$. Let $X(t)$ be the firing count of t. Then, $\forall t \in p^{\bullet}$: $\frac{X(t)}{\sum_{t' \in p^{\bullet}} X(t')} \leq R(t)$. Figure 1 is a Petri net model of a part of IL-1 signaling pathway[6].

2.2 Automaton model of UPPAAL

UPPAAL[7] is a model checking tool for verification of real-time systems. This tool can use timed automata as state transition models. We use the following notations: C is a set of clocks and $B(C)$ is the set of conjunctions over simple conditions of the form $x \bowtie c$ or $x - y \bowtie c$, where $x, y \in C$, $c \in \mathbb{N}$ and $\bowtie \in \{<, \leq, =, \geq, >\}$. A timed automaton is a finite directed graph annotated with conditions over and resets of non-negative real valued clocks.

Definition 2. *A timed automaton is a 6-tuple* (L, l_0, C, Act, E, I).
L : *A set of locations*
$l_0 \in L$: *The initial location*
C : *A set of clocks*
Act : *A set of actions*
$E \subseteq L \times Act \times B(C) \times 2^C \times L$: *A set of edges between locations with an action, a guard and a set of clocks to be reset*
$I : L \to B(C)$: *A function to assign to an invariant to a location* □

Timed automata often form a network over a common set of clocks and actions, consisting of n timed automata $A_i = (L_i, l_i^0, C, Act, E_i, I_i)$, $1 \leq i \leq n$. For the semantics of the automaton model, refer to [8].

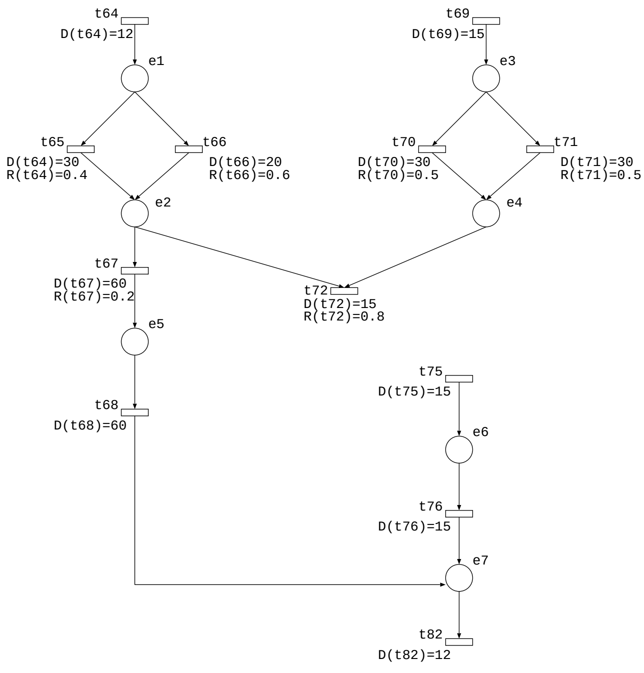


Fig. 1. A part of Petri net model of IL-1 signaling pathway

In model checking in UPPAAL, we describe a property to be analyzed in Timed Computation Tree Logic (TCTL) and we can verify whether the model with clock variables satisfies the property by running model checking. TCTL on UPPAAL includes `E<> p` (There exists a path where `p` eventually holds) and `A[] p` (For all paths `p` always holds).

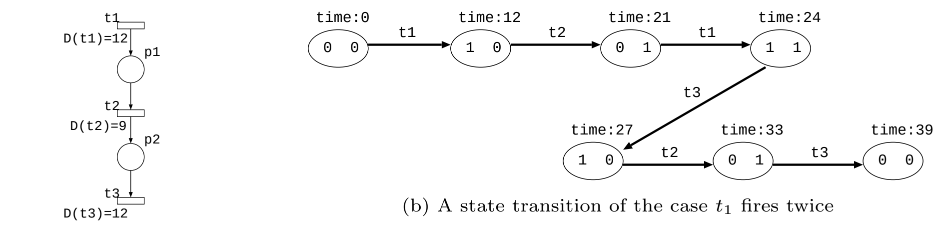
3 Two modeling methods for signaling pathways with multiple signals

A Petri net model representing a signaling pathway must be correct. We use UPPAAL to verify whether a Petri net model is correct, because the Petri net model includes time concept. The model used in this paper is timed Petri net, but it is easy to extend our method to time Petri nets.

It is known that model checking is powerful but may cause the state-space explosion. The more signals are, the more the problem becomes severe. Kwiatkowska et al.[3] mentioned that model checking with multiple signals cause the state-space explosion further than that with a single signal. It is known that there is a tradeoff between expressive power and analytical power. To balance the powers, we need to devise modeling methods. Concretely, we propose two modeling methods. The first method uses multiple automata as a model of UPPAAL. It is named "multiple automata modeling method." The second method uses a single automaton as a model of UPPAAL. It is named "single automaton modeling method."

3.1 Multiple automata model and single automaton model

We give the representation model used in each modeling method. Both models have the same expressive power as a Petri net model; provided that the Petri net model has no transition branches. Note that those models can treat transition joins. We give a simple example to show the difference between the multiple automata model and the single automaton model.



(a) A Petri net model

(b) A state transition of the case t_1 fires twice

Fig. 2. A Petri net model $TPNR_1$ of a signaling pathway

(1) Multiple automata model
The multiple automata model preserves the structure of a given Petri net model as much as possible. A Petri net model with n signals is represented as the following:
- The multiple automata model of the Petri net model consists of n automata.
- Each automaton represents the state transition of a single signal.
- A place or a transition is represented as a location.
- An arc is represented as an edge.
- Firing delay time is implemented by using a location invariant and a guard.

In this example, we use a Petri net model, $TPNR_1$, which is shown in Fig.2. Figure 2(a) is a Petri net model and Fig.2(b) is a state transition of the case t_1 fires twice. Since $TPNR_1$ is state machine, the reachability graph has a similar structure as the net. Figure 3 shows the multiple automata model of $TPNR_1$. In this example, we assume that source transition t_1 fires twice. There are two signals in this model. The two signals are represented as two automata. Two places and three transitions are represented as five locations. Four arcs are represented as four edges. Firing delay time $D(t_1) = 12$ is implemented by using location invariant (`gc1<=12`) of `t1` and guard (`gc1>=12`) of edge `(t1,p1)`, where `gc1` denotes a global clock. Similarly, firing delay time $D(t_2) = 9$ is implemented by using location invariant (`c<=9`) of

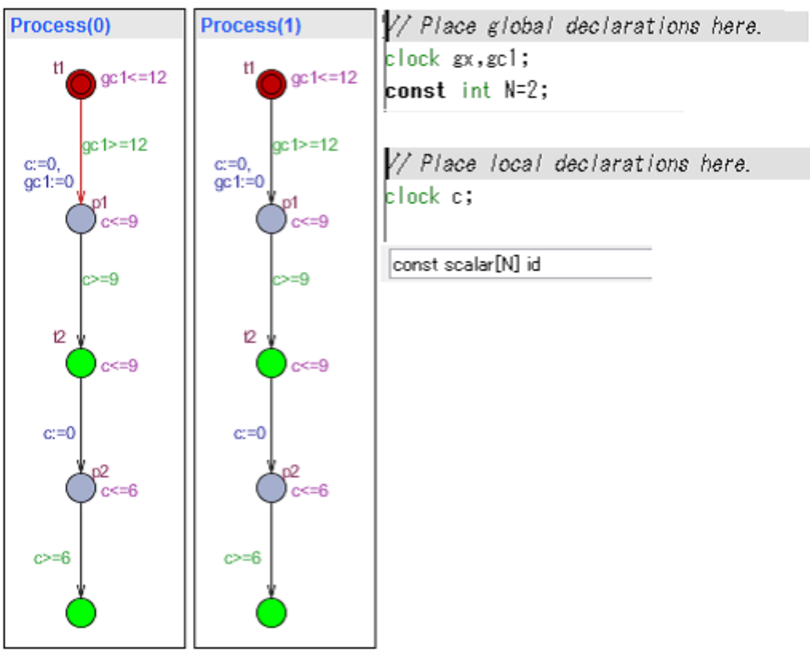


Fig. 3. The multiple automata model of $TPNR_1$

Table 1. Variables of single automaton model

Variable	Meaning
clock `gc1`, `gc2`, $\cdots$, `gc`m	Clock for source transitions
clock `c[`k`]`	Clock of the i-th signal
int `xf[`k`]`	Firing delay time of the i-th signal
int `mp1`, `mp2`, $\cdots$, `mp`$\|P\|$	The number of signals in p_i
int `tf1`, `tf2`, $\cdots$, `tf`$\|T\|$	Signal ID of the first reserved signal in transition t_i
int `tx1[`k`]`, `tx2[`k`]`, $\cdots$, `tx`$\|T\|$`[`k`]`	Signal IDs reserved in transition t_i
int `l1`, `l2`, $\cdots$, `r1`, `r2`, $\cdots$	The firing count of a transition in a place branch
void `t1res()`, `t2res()`,$\cdots$	Reset `c[]` and `xf[]`
void `freetx1()`, `freetx2()`,$\cdots$	Organize reserved signal IDs of transition t_i

`p1` and `t2` and guard (`c>=9`) of edge `(p1,t2)`, where `c` denotes a local clock. $D(t_3)$ is implemented by a similar way. A current state of the automata represents where the signals are now.

(2) Single automaton model
The single automaton model represents a given Petri net model as a single automaton with only one location even if there are two or more signals. A Petri net model with n signals is represented as the following:

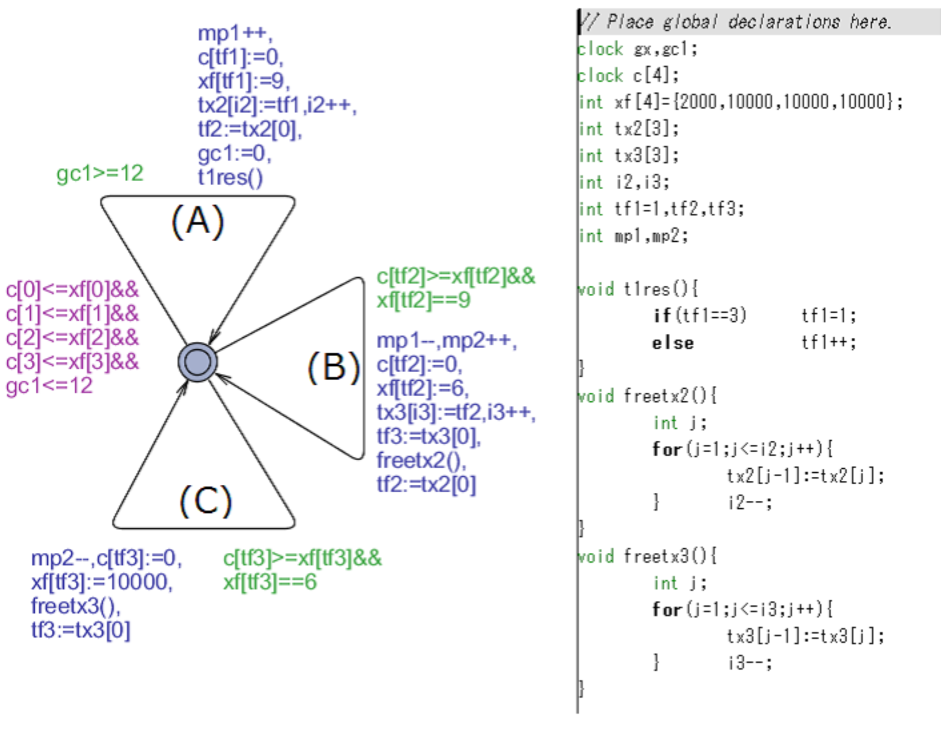


Fig. 4. The single automaton model of $TPNR_1$

- The single automaton model consists of a single automaton with only one location.
- The automaton represents multiple signals by using clocks `c[]` and variables `xf[]`.
- The markings of the places are represented as variables `mp1`, `mp2`, $\cdots$, `mp`$|P|$.
- A firing of each transition is represented as a self loop of the location.
- Firing delay time is implemented by using a location invariant and a guard.

Figure 4 shows the single automaton model of $TPNR_1$. Figure 5 is the state transition of the single automaton model. Places p_1, p_2 are represented as a single location with variables `mp1`, `mp2`, where `mp`i denotes the marking of p_i. The state transition of marking is same as Fig.2(b). Table 1 is the variables used in single automaton model. A firing of t_1, t_2, t_3 is represented as edge (A), (B), (C). A global clock `gc1` is assigned for the source transition t_1. Clocks `c[]` and variables `xf[]` are assigned for signals. k is the maximum number of signals in the pathway. `xf[`i`]` denotes the firing delay time of the i-th signal. Multiple signals are represented as a single location with `c[]` and `xf[]`. Firing delay time of the i-th signal is implemented by using location invariant `(c[`i`]<=xf[`i`])` and guard `(c[`i`]>=xf[`i`])`. In a firing of transition t_1 (Edge (A)), action `(xf[tf1]:=9)` means setting the firing delay time of next transition t_2. `tf`i denotes the signal ID of the first reserved signal in transition t_i. Action `(tx2[i2]:=tf1)` means reserving the signal that fires on t_1 to next transition t_2. `tx`i`[`k`]` stores signal IDs reserved in transition t_i. `(tf2:=tx2[0])` removes the first reserved signal ID of transition t_2. In a firing of the transition t_2 (Edge (B)), action `(mp1--, mp2++, c[tf2]:=0, xf[tf2]:=0, tx3[i3]:=tf2, tf3:=tx3[0])`

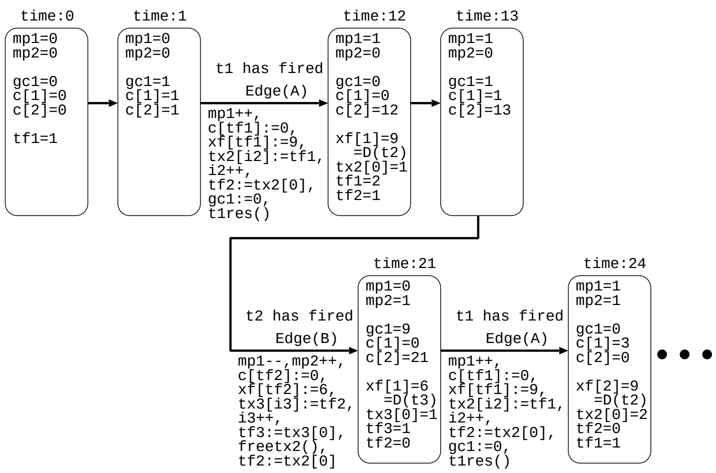


Fig. 5. The state transition of single automaton model of $TPNR_1$

is the same role as above. `freetx2()` organizes reserved signal IDs of t_2. `xf[0]` denotes the end time. In this model, we can analyze the model until `xf[0]`.

3.2 Transformation algorithms

In this subsection, we give, for each modeling method, an algorithm for transforming a Petri net to an automaton model. Due to limitations of space, we restrict the algorithm to the modeling of a single path Petri net model.

(1) Multiple automata modeling method
<<Transformation to Multiple Automata Model>>
Input: Petri net model $TPNR = (P, T, \mathcal{E}, D, R)$, number n of signals
Output: Multiple automata model $A_1, A_2, \cdots, A_n$

Foreach $i = 1$ to n, make A_i according to the following:

1. $L \leftarrow P \cup T$
2. $C \leftarrow \{\texttt{gc1},\ \texttt{c}i\}$
3. $E \leftarrow \{(p, \emptyset, (\texttt{c}i\texttt{>=}\ D(t)), \emptyset, t) |\ (p,t) \in A\}$
$\cup \{(t, (\texttt{gc1:=0},\ \texttt{c}i\texttt{:=0}), (\texttt{gc1>=}D(t)), \{\texttt{gc1},\ \texttt{c}i\}, p) |\ |{}^\bullet t| = 0, (t,p) \in A\}$
$\cup \{(t, (\texttt{c}i := 0), \emptyset, \{\texttt{c}i\}, p) |\ |t^\bullet| > 0, (t,p) \in A\}$
4. $I \leftarrow \{(t, (\texttt{gc1<=}D(t_1))) |\ t \in T, |{}^\bullet t| = 0\}$
$\cup \{(t, (\texttt{c}i\texttt{<=}D(t))) |\ t \in T, |t^\bullet| > 0\}$
$\cup \{(p, (\texttt{c}i\texttt{<=}D(t))) |\ p \in P$, t is the output transition of $p\}$

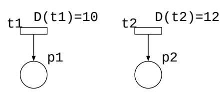
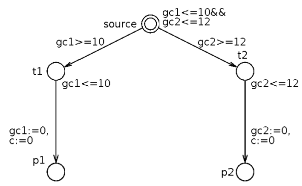
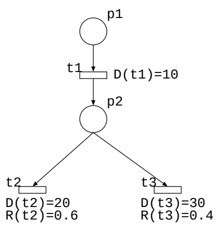
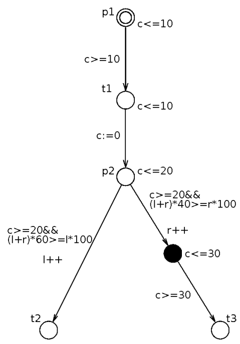
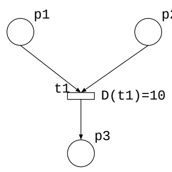
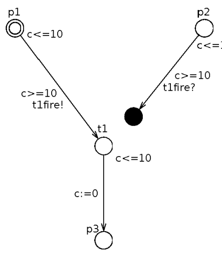
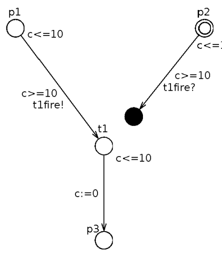
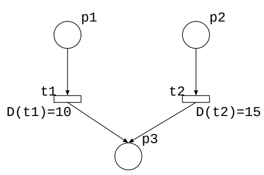
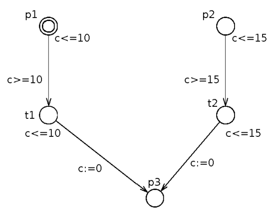
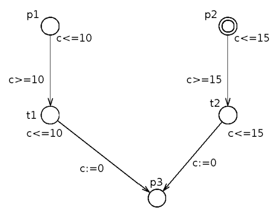
(2) Single automaton modeling method
<<Transformation to Single Automaton Model>>
Input: Petri net model $TPNR = (P, T, \mathcal{E}, D, R)$, number n of signals
Output: Single automaton model A, variables `mp1`, `mp2`, $\cdots$, `mp`$|P|$, `tf1`, `tf2`, $\cdots$, `tf|T|`, `xf[]`, `tx1[]`, `tx2[]`, $\cdots$, `tx`$|T|$`[]`.

1. $L \leftarrow \{l_0\}$
2. $C \leftarrow \{\texttt{gc1},\ \texttt{c[1]},\ \texttt{c[2]}, \cdots,\ \texttt{c[k]}\}$
3. $E \leftarrow \{(l_0, (\texttt{mp1++},\ \texttt{c[tf1]:=0},\ \texttt{xf[tf1]:=}D(t_2),\ \texttt{tx2[i2]:=tf1},\ \texttt{i2++},\ \texttt{tf2:=tx2[0]},\ \texttt{gc1:=0},\ \texttt{t1res()}), (\texttt{gc1>=}D(t_1)), \{\texttt{gc1},\ \texttt{c[tf1]}\}, l_0)\}$
$\cup \{(l_0, (\texttt{mp}|P|\texttt{--},\ \texttt{c[tf|T|]:=0},\ \texttt{xf[tf}|T|\texttt{]:=10000},\ \texttt{freetx|T|()},\ \texttt{tf}|T|\ \texttt{:=tx}|T|\texttt{[0]}), (\texttt{c[tf}|T|\texttt{]>=xf[tf}|T|\texttt{]\ \&\&\ xf[tf}|T|\texttt{]==}D(t_{|T|}))), \{\texttt{c[tf}|T|\ \texttt{]}\}, l_0)\}$
$\cup \bigcup_{i=2to|T|-1} \{(l_0, (\texttt{mp}(i-1)\texttt{--},\ \texttt{mp}i\texttt{++},\ \texttt{c[tf}i\texttt{]:=0},\ \texttt{xf[tf}i\texttt{]:=}D(t_{(i+1)}), \texttt{tx}\ (i+1)\texttt{[i}(i+1)\texttt{]:=tf}i, \texttt{i}(i+1)\texttt{++},\ \texttt{tf}(i+1)\texttt{:=tx}(i+1)\texttt{[0]},\ \texttt{freetx}i\texttt{()},\ \texttt{tf}i\texttt{:=tx}i\texttt{[0]}),\ (\texttt{c[tf}i\texttt{]>=xf[tf}i\texttt{]\&\&xf[tf}i\texttt{]==}D(t_i)), \{\texttt{c[tf}i\texttt{]}\}, l_0)\}$
`t1res()` resets `c[]` and `xf[]`.
`freetx`i`()` organizes reserved signal IDs of transition t_i.
4. $I \leftarrow \{(l_0, (\texttt{c[0]<=xf[0]\&\&c[1]<=xf[1]\&\&} \cdots\ \texttt{\&\&c[k]<=xf[k]\&\&}\ \texttt{gc1<=}D(t_1)))\}$

3.3 Pattern lists

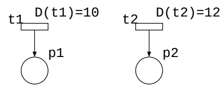
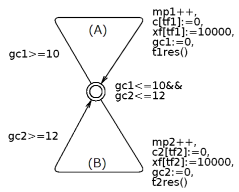
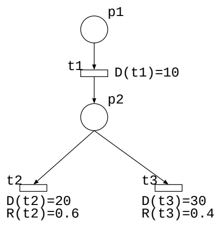
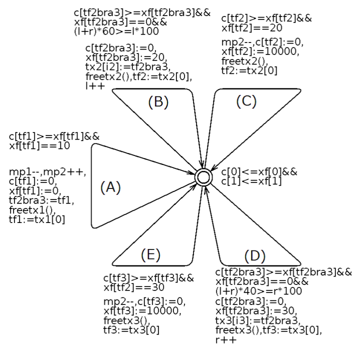
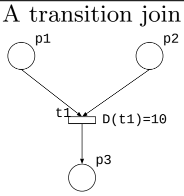
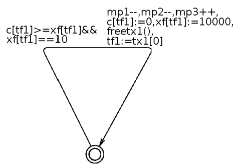
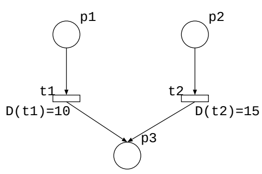
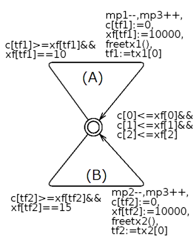
We give the pattern lists of transforming TPNR to multiple automata model and single automaton model to ease the transformation. Multiple automata model

Table 2. The pattern list of transforming TPNR to multiple automata model

<table>
<tr><th>TPNR model</th><th colspan="2">multiple automata model</th></tr>
<tr><td>Multiple source transitions

</td><td colspan="2">
</td></tr>
<tr><td>A place branch

</td><td colspan="2">
</td></tr>
<tr><td>A transition join

</td><td>
</td><td>
</td></tr>
<tr><td>A place join

</td><td>
</td><td>
</td></tr>
</table>

and single automaton model can be created by combination of these patterns. Table 2 is the pattern list of transforming TPNR to multiple automata model. `c` is a local clock and `gc` is a global clock.

Table 3. The pattern list of transforming TPNR to single automaton model

TPNR model	single automaton model
Multiple source transitions 	
A place branch 	
A transition join 	
A place join 	

- The first column is for multiple source transitions. Location `source` is a global source. This location implements all of the firing delay times of source transition with global clocks, `gc1`, `gc2`.
- The second column is for a place branch. A place branch is implemented by using `l` and `r`. `l` denotes the firing count of `t2`, and `r` denotes the firing count of `t3`. And guard (`(l+r)*40>=l*100`) limits that firing count of `t2`. `t3` is implemented by a similar way of `t2` but firing delay time of `t3` is longer than that of `t2`. So, the blacken location is added to implement the firing delay time.
- The third column is for a transition join. A transition join is implemented by channels `t1fire!` and `t1fire?`. A signal on `p1` and a signal on `p2` are synchronized, then two signals fires on `t1`. The blacken location is added to abandon the signal on `p2`.
- The fourth column is for a place join. A place join is implemented by a similar way of <<Transformation to Multiple Automata Model>>.

Table 3 is the pattern list of transforming TPNR to single automaton model. `c` is a local clock and `gc` is a global clock.

- The first column is for multiple source transitions. Edge (A) implements a firing of t_1 and edge (B) implements a firing of t_2. Firing delay times of t_1 and t_2 are implemented by global clocks `gc1` and `gc2`.
- The second column is for a place branch. Edge (A) implements the firing of transition t_1 with substituting 0 in `xf[]` and `tf2bra3` stores the signal ID that isn't reserved. Edges (B) and (D) are limiting the firing count by similar way of multiple automata model and reserve a signal to transition t_2 or t_3. Edges (C) and (E) implement the firing of transitions t_2 and t_3.
- The third column is for a transition join. A transition join is implemented by only updating variables `mp1` and `mp2`.
- The fourth column is for a place join. A place join is implemented by a similar way of <<Transformation to Single Automaton Model>>. Edge (A) implements the firing of transition t_1 and edge (B) implements the firing of transition t_2.

4 Application example: IL-1 signaling pathway

In this section, we apply the proposed modeling methods to IL-1 signaling pathway. And we compare the obtained models and discuss the merits and demerits of the modeling methods.

IL-1 is a proinflammatory cytokine, and plays an important role in regulating the mechanism of proinflammatory. There cause multiple signals flowing in the signaling pathway because a ligand joins a receptor repeatedly. The whole Petri net model of IL-1 signaling pathway is shown in Fig.3 of [9].

The correctness of the model must be examined. We can check the correctness of the model by model checking. For example,

– We can check the reachability with time concept. Checking the reachability helps us to understand signaling pathways. Let c be a clock, we write a TCTL expression of this property as `E<>` $M(p_1)$`>0 &&` c`==30`. This expression means there exists a path where a signal is on p_1 when c is 30.
– We can also check that there is no retention in the model. Retention means that signals are accumulated at any place. We write a TCTL expression of this property as `A[]` $M(p_1)$`<=5`. This expression means that the number $M(p_1)$ of signals for place p_1 is always 5 or less.

In this paper, we focus on checking the retention. If there is retention in the Petri net model, we consider the error in the Petri net model or the possible of unknown paths in the signaling pathway.

We apply two modeling methods to a part of IL-1, which is shown in Fig.1. The size of the Petri net model is $|P| = 7$, $|T| = 12$. Figure 6 shows the multiple automata model of Fig.1. Figure 7 shows the single automaton model of Fig.1.

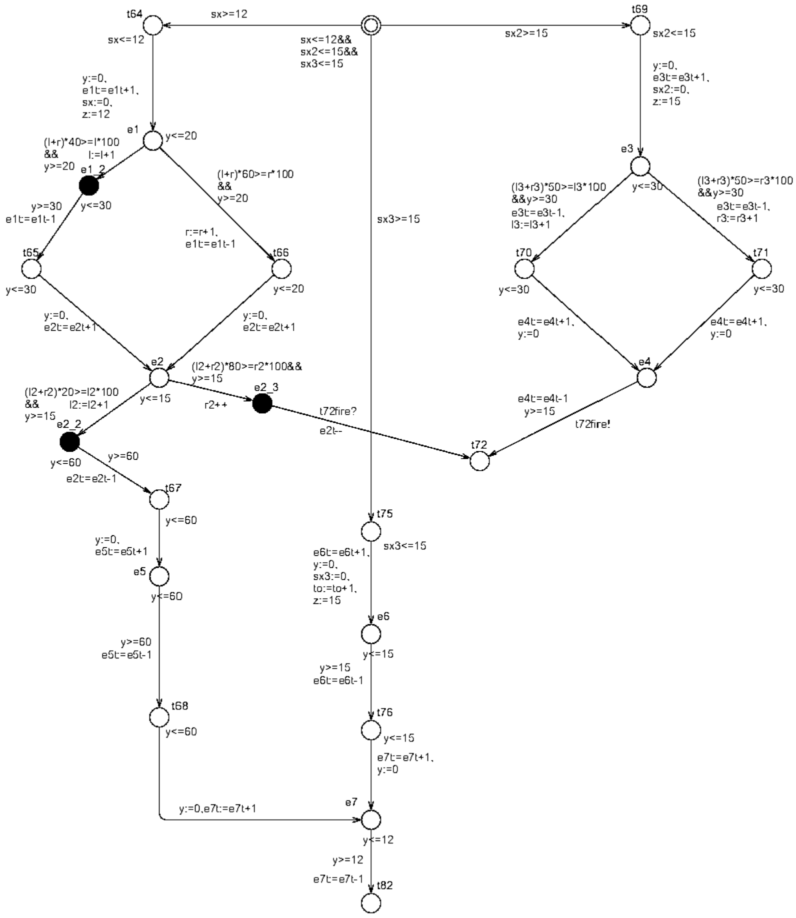


Fig. 6. The multiple automata model of the part of IL-1 signaling pathway. Each automaton represents the behavior of a single signal.

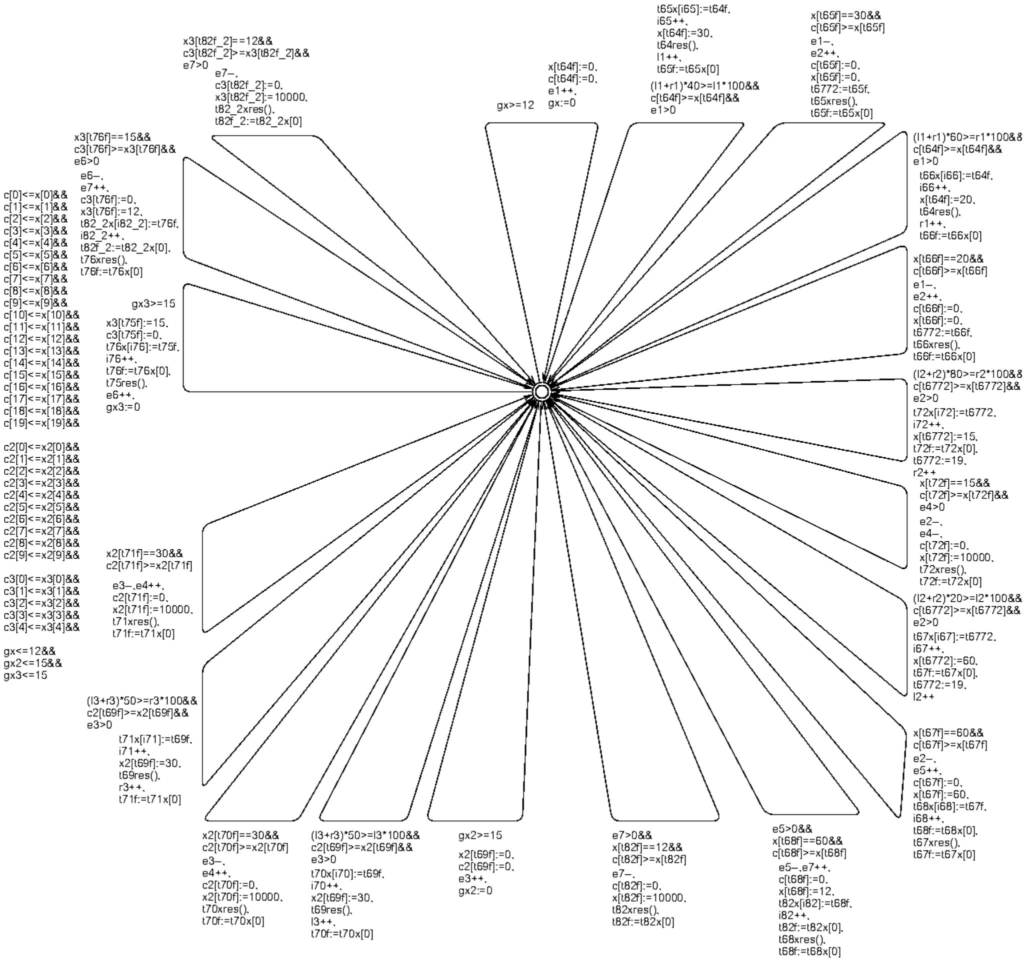


Fig. 7. The single automaton model of the part of IL-1 signaling pathway. This automaton represents the behavior of multiple signals.

This Petri net model includes three source transitions, therefore we applied the first column of each pattern list at the case of three source transitions. Places e_1, e_2, and e_3 have two output transitions, therefore we applied the second column of each pattern list. We applied the third column of each pattern list for transition t_{72} because t_{72} has two input places. And we applied the fourth column of each pattern list for place e_7 because e_7 has two input places.

Table 4 shows the size and the number of clocks and variables of the obtained models. The size of each automaton of the multiple automata model is $|L| = 23$, and $|E| = 26$. In comparison, the size of the single automaton model is $|L| = 1$, and $|E| = 19$. The number of clocks of the multiple automata model is $3+n$, and the number of variables is 6, where n is the number of signals. The number of clocks of the single automaton model is 48, and the number of variables is

Table 4. Evaluation results of example 1: IL-1 signaling pathway $|P|$=7, $|T|$=12

Model	Number of signals n	Number of clocks	Number of states explored	Checking time(min)
Multiple	39	42	139858	2
automata	78	81	770854	110
$\lvert L\rvert$=23, $\lvert E\rvert$=26	128	131	2205987	180
variables:6	129	132	Out of memory	
Single	107	48	581918	5
automaton	325	48	2393918	20
$\lvert L\rvert$=1, $\lvert E\rvert$=19	702	48	5505758	35
variables:53	703	48	Out of memory	

40. In addition, the number of arrays is 13. The number of clocks continue to increase according to the number of signals n in the multi automata model, but the number of clocks is constant in the single automaton model. The size of the single automaton model is smaller than the size of the multiple automata model, but the number of variables of single automaton model is larger than that of the multiple automata model under the same expressive power.

We can analyze the retention property by using those automaton models. Table 4 shows the number of signals, the number of states explored, checking time. Model checking is performed on the PC with CPU Xeon 2.13GHz and memory 3.2Gbyte. We could check the retention property of the multiple automata model until the number of signals is 128. Meanwhile we could check that of the single automaton model until the number of signals is 702. The single automaton model enables us to analyze more signals than the multiple automata model. The number of states explored of the single automaton model is smaller than the multiple automata model, and checking time is also shorter.

5 Conclusion

In this paper, we proposed two modeling methods for signaling pathways with multiple signals. Those modeling methods use different representation model. Next we gave for each representation model, a transformation algorithm, and a pattern list. Then we applied these proposed methods to the IL-1 signaling pathway. The results show that signaling pathway with multiple signals should be represented as not automata but variables, and the model size to be analyzed can be increased by devising of modeling method. We have also applied these method to endocytosis signaling pathway, and a similar trend have been obtained.

As future work, we plan to develop a method treating transition branches. An approach is to prepare child process for implementing parallel part and devising the clock handling. And we think that it is easy to extend our method to time Petri net model. We will verify that our method can be applied to time Petri net model.

Acknowledgement
The authors would like to thank Prof. Hiroshi Matsuno, Prof. Qi-Wei Ge, and Mr. Yuki Murakami for their valuable advices.

References

1. N.Chabrier, F.Fages, "Symbolic model checking of biochemical networks," Lecture Notes in Computer Science 2602, pp.149–162, 2003.
2. W. J. Bos , "Interactive Signaling Network Analysis Tool," Master Thesis, University of Twente, 2009.
3. M. Kwiatkowska, G. Norman, D. Parker, "Probabilistic model checking for systems biology," Symbolic Systems Biology, Jones and Bartlett, 2010.
4. F. Cassez, O. H. Roux, "From Time Petri Nets to Timed Automata", Petri Net, Theory and Applications, InTech Education and Publishing, 2008.
5. C. Li, S. Suzuki, Q. W. Ge, M. Nakata, H. Matsuno, S. Miyano, "Structural modeling and analysis of signaling pathways based on Petri nets," Journal of Bioinformatics and Computational Biology, vol.4, no.5, pp.1119–1140, 2006.
6. H. Matsuno, C. Li, S. Miyano, "Petri net based descriptions for systematic understanding of biological pathways," IEICE Trans. Fundamentals, vol. E89-A, no. 11, pp. 3166–3174, 2006.
7. UPPAAL Group, UPPAAL, http://www. uppaal. com/, 1995.
8. UPPAAL Group, A Tutorial on Uppaal, http://www.cs.aau.dk/∼adavid/ RTSS05/ UPPAAL-tutorial.pdf, 2005.
9. Y. Miwa, Y. Murakami, Q. W. Ge, C. Li, H. Matsuno, S. Miyano, "Delay Time Determination for the Timed Petri Net Model of a Signaling Pathway Based on Its Structural Information," IEICE Trans. Fundamentals, vol.E93-A, no.12, pp.2717–2729, 2010.
10. T. Murata, "Petri nets: Properties, analysis and applications," Proc. of the IEEE, vol.77, no.4, pp.541–580, 1989.
11. V. N. Reddy, M. L. Mavrovouniotis, M. N. Liebman, "Petri net representations in metabolic pathways," Proc. Int. Conf. Intell. Syst. Mol. Biol., vol.1, pp. 328–336, 1993.
12. M. Calder, V. Vyshemirsky, D. Gilbert and R. Orton, "Analysis of signalling pathways using the PRISM model checker," Proc. Computational Methods in Systems Biology (CMSB'05), pp. 179–190, 2005.
13. M. Heiner, I. Koch and J. Will, "Model Validation of Biological Pathways Using Petri Nets–demonstrated for apoptosis," Biosystems, vol.75, no.1–3, pp.15–28, 2004.
14. D. Gilbert and M. Heiner, "From Petri nets to differential equations – anintegrative approach for biochemical network analysis," Proc. 27th International Conference on Application and Theory of Petri Nets, LNCS 4024, pp.181–200, 2006. Biosystems, vol.75, no.1–3, pp.15–28, 2004.