Approximate Analysis of Homeostasis of Gene Networks by Linear Temporal Logic using Network Motifs

Sohei Ito[1], Shigeki Hagihara[2] and Naoki Yonezaki[2]
[1]*Department of Fisheries Distribution and Management, National Fisheries University, 2-7-1 Nagata-Honmachi, Shimonoseki, Yamaguchi, Japan*
[2]*Department of Computer Science, Tokyo Institute of Technology, 2-12-1 Ookayama, Meguro-ku, Tokyo, Japan*

Keywords: Gene Regulatory Network, Homeostasis, Temporal Logic, Strong Satisfiability, Network Motif, Systems Biology.

Abstract: We proposed a novel framework to analyse homeostasis of gene networks using linear temporal logic. We formulate a kind of homeostasis as *strong satisfiability* of reactive system specifications. Both behaviours and properties of gene networks are specified in linear temporal logic and homeostasis of the network is checked by strong satisfiability checkers. Though this framework is simple and applicable for many networks, the computational complexity is heavy and large networks cannot be directly analysed. In this paper we present an approximate analysis method to mitigate this computational difficulty. We approximately specify a network specification using fewer propositions such that approximated specifications guarantee homeostasis of the network. However it is difficult to find such safely approximated specifications for any gene network. Thus we present approximate specifications for *network motifs*, which are common patterns appearing in many gene networks. We demonstrate our approximate method and see that our approximate method is quite efficient in analysing large networks.

1 INTRODUCTION

Although homeostasis in biological systems is a remarkable feature of life, it has been considered to be elusive and difficult to be analysed. Ito et al. (Ito et al., 2014) proposed a mathematical and precise definition of homeostasis in gene networks and provided a method for analysing it. Their approach is based on Ito et al.'s constraint-based modelling of gene networks (Ito et al., 2010; Ito et al., 2013b; Ito et al., 2013a) using linear temporal logic (LTL) (Emerson, 1990). In their method, possible behaviours of gene networks are characterised as LTL formulae, which means that possible behaviours are behaviours that satisfy the constraints (called network specifications) given as LTL-formulae. With network specifications and given biological property, the *homeostasis* of network is analysed by checking whether the formulae is *realisable* or not. The specification which satisfies realisability (homeostasis) can respond to any input sequence (any stimulus) without violating the specification (breaking its internal functions). This framework for analysing gene networks belongs to the same lineage as the verification of reactive system specifications (Pnueli and Rosner, 1989; Abadi et al., 1989). The problem, however, is the computational complexity of realisability problem of LTL which is 2EXPTIME-complete in the size of a formula (Pnueli and Rosner, 1989). Since the size of a formula is proportional to the size of a network, direct analysis of a large network is intractable in general.

In this paper, we propose the notion of *weak homeostasis* which is close to Ito et al.'s definition but a bit weaker. We formulate this notion by *strong satisfiability* (Mori and Yonezaki, 1993) which is weaker than realisability. Strong satisfiability is proposed to approximate realisability and has a more efficient checking algorithm than realisability has. However, the complexity of checking strong satisfiability is still high (EXPSPACE-complete (Shimakawa et al., 2013)) and we need to devise some efficient method to mitigate this difficulty. Fortunately, we found that we can import the approximate analysis method for checking satisfiability (Ito et al., 2013b) to strong satisfiability checking, which is the main contribution of this paper. The key idea of approximate analysis is to simplify a network specification using fewer propositions and approximate the possible behaviours of a

Ito S., Hagihara S. and Yonezaki N..
Approximate Analysis of Homeostasis of Gene Networks by Linear Temporal Logic using Network Motifs.
DOI: 10.5220/0005177000930101
In *Proceedings of the International Conference on Bioinformatics Models, Methods and Algorithms* (BIOINFORMATICS-2015), pages 93-101
ISBN: 978-989-758-070-3

network. We prove that the certain class of approximate specifications can be used instead of the original specifications to check weak homeostasis of gene networks. The problem is that it is difficult to find such *safe* approximate specifications for arbitrary gene networks. Thus we use Ito et al.'s approximate specifications (Ito et al., 2013b) for *network motifs*, which are common patterns in gene networks (Alon, 2007). We demonstrate our approximate method for several networks from real biological systems. This experiment shows that the cost of analysis is drastically reduced by our approximate analysis.

This paper is organised as follows. In section 2 we introduce LTL and show how we model possible behaviours of gene networks. In section 3 we define the notion of weak homeostasis using strong satisfiability of LTL. In section 4 we introduce the approximate method for analysing weak homeostasis. We also present approximate specifications for network motifs. In section 5 we show experimental results of our approximate method and see how we benefit from it. The final section offers conclusion and future directions.

2 PRELIMINARY

In this section we introduce linear temporal logic (LTL) upon which our constraint-based modelling method is based. Then we review how we characterise possible behaviours of a given network using LTL (Ito et al., 2010; Ito et al., 2013b; Ito et al., 2013a).

2.1 Linear Temporal Logic

Let A be a finite set. We write A^ω for the set of all infinite sequences on A. We write $\sigma[i]$ for the i-th element of $\sigma \in A^\omega$. Let AP be a set of propositions. A *time structure* is a sequence $\sigma \in (2^{AP})^\omega$ where 2^{AP} is the powerset of AP. The formulae in LTL are defined as follows.

- $p \in AP$ is a formula.
- If ϕ and ψ are formulae, then $\neg\phi, \phi \wedge \psi, \phi \vee \psi$ and $\phi U \psi$ are also formulae.

Let σ be a time structure and ϕ be a formula. We write $\sigma \models \phi$ to mean that ϕ is true in σ, and we say σ *satisfies* ϕ. The satisfaction relation $\models$ is defined as follows.

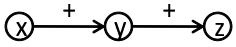

Figure 1: A gene network in which x, y and z are genes. Plus-edges represent activation relationship.

$$
\begin{array}{lll}
\sigma \models p & \text{iff} & p \in \sigma[0] \text{ for } p \in AP \\
\sigma \models \neg\phi & \text{iff} & \sigma \not\models \phi \\
\sigma \models \phi \wedge \psi & \text{iff} & \sigma \models \phi \text{ and } \sigma \models \psi \\
\sigma \models \phi \vee \psi & \text{iff} & \sigma \models \phi \text{ or } \sigma \models \psi \\
\sigma \models \phi U \psi & \text{iff} & (\exists i \geq 0)(\sigma^i \models \psi \\
 & & \text{and } \forall j (0 \leq j < i) \sigma^j \models \phi)
\end{array}
$$

where $\sigma^i = \sigma[i]\sigma[i+1]\ldots$, i.e. the i-th suffix of σ. We say σ a *model* of ϕ when $\sigma \models \phi$.

In the rest of the paper we use the following abbreviations: $\bot \equiv p \wedge \neg p$ for some $p \in AP$, $\top \equiv \neg\bot$, $\phi \rightarrow \psi \equiv \neg\phi \vee \psi$, $\phi \leftrightarrow \psi \equiv (\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$, $F\phi \equiv \top U \phi$, $G\phi \equiv \neg F \neg\phi$, and $\phi W \psi \equiv (\phi U \psi) \vee G\phi$. We assume that $\wedge, \vee$ and U bind more strongly than $\rightarrow$ and unary connectives bind more strongly than binary ones.

2.2 Conceptualising Behaviours of a Gene Network as Time Structure

The basic idea of modelling possible behaviours of a gene network is that we abstract time series of dynamic behaviours of gene networks as time structures. For example, given a network depicted in Fig. 1 in which gene x activates gene y and gene y activates gene z, we consider an example dynamic behaviour of this network depicted in Fig. 2. The expression levels x_y and y_z in Fig. 2 are the threshold of gene x to activate gene y and that of gene y to activate gene z, respectively. If a gene is expressed beyond a threshold to activate (or inhibit) a gene, its regulation effects start to work. For example, when gene x is expressed beyond the threshold x_y (e.g. duration between time t_1 and t_3), gene y is ON and begins to be expressed.

If we verbally describe the network behaviour, we only need to mention that whether a gene is ON or OFF, whether a gene is expressed beyond its thresholds[1] and how such situation changes over time. Such atomic facts to describe a situation of a network can be represented by *propositions*. In the case of network depicted in Fig. 1, we introduce the following propositions to describe the behaviour:

- on_x, on_y, on_z: whether gene x, y and z are ON, respectively.
- x_y, y_z: whether gene x is expressed beyond the

[1] In general, there should be multiple thresholds for each gene.

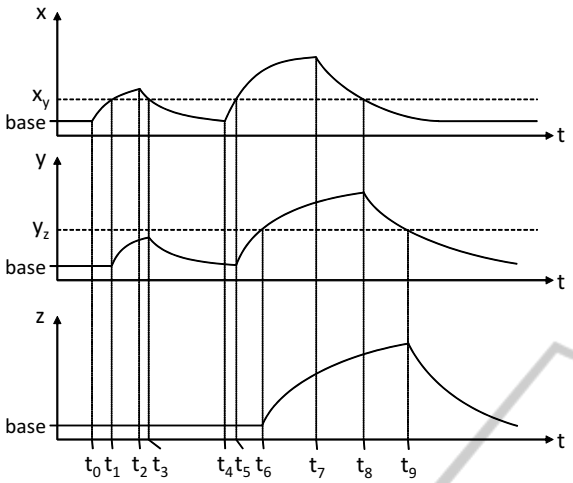

Figure 2: Example behaviour of the network depicted in Fig. 1. The level x_y is the threshold of gene x for activating gene y and y_z is the threshold of gene y for activating gene z.

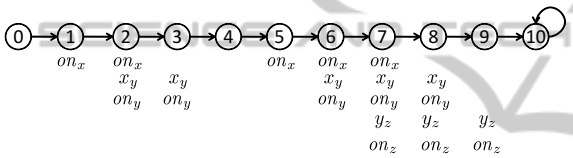

Figure 3: Representation of behaviour depicted in Fig. 2 as a time structure.

threshold x_y, and whether gene y is expressed beyond the threshold y_z, respectively[2].

Using these propositions as the set AP of atomic propositions, we have a time structure $\sigma \in (2^{AP})^\omega$ depicted in Fig. 3. Note that state 0 corresponds to the interval $[0,t_0)$, state 1 to $[t_0,t_1)$, and so on.

2.3 Modelling Possible Behaviours of a Network in LTL

Based on the abstraction of behaviours of a network as a time structure, we characterise possible behaviours of a gene network as the set of the models of a suitable LTL formula, which is obtained by a given network. Formally, for a given network G, we specify an LTL formula φ_G which is intended to characterise the set of possible behaviours of G. Then the set of possible behaviours is the set $\{\sigma \in (2^{AP})^\omega \mid \sigma \models \varphi_G\}$ (i.e. σ is a model of φ_G).

The problem is how we obtain a such formula. This is solved by the following principles about behaviours of gene networks.

- A gene is ON when its activators are expressed beyond some thresholds.
- A gene is OFF when its inhibitors are expressed beyond some thresholds.
- If a gene is ON, its expression level increases.
- If a gene is OFF, its expression level decreases.

Fortunately, these principles can be naturally described in LTL. In the following we show how we describe the above principles.

Conditions for Activation and Inhibition of Genes. In simple situation such that gene x alone activates gene y, gene y is ON if gene x is expressed beyond the threshold x_y. This is described as

$$G(x_y \rightarrow on_y).$$

Another choice is $G(x_y \leftrightarrow on_y)$ which says that gene y is ON if, *and only if* gene x is expressed beyond the threshold x_y. If we consider no other (implicit) regulator for gene y, the latter specification may be reasonable. Similarly, if gene x alone inhibits gene y, gene y is OFF if gene x is expressed beyond the threshold x_y. This is described as

$$G(x_y \rightarrow \neg on_y).$$

As in the case of activation , we may write $G(x_y \leftrightarrow \neg on_y)$.

For more complicated situation, a gene has multiple regulators and the effect may be different from one another. For example, consider that gene u is activated by both gene x and y, and inhibited by gene z. Generally we do not know the regulation function of u which has three inputs. In such situation, we only describe sufficient conditions for u's activation and inhibition: gene u is ON if gene x is expressed beyond x_u, gene y beyond y_u and gene z below z_u. This is described as

$$G(x_u \wedge y_u \wedge \neg z_u \rightarrow on_u).$$

Moreover, gene u is OFF if gene x is expressed below x_u *and* gene y below y_u *and* gene z beyond z_u. This is described as

$$G(\neg x_u \wedge \neg y_u \wedge z_u \rightarrow \neg on_u).$$

If we (may partially) know about the regulation function, we can reflect such knowledge in the specification. For example the positive effect of gene x and y are merged by 'OR', we can describe as

$$G((x_u \vee y_u) \wedge \neg z_u \rightarrow on_u),$$
$$G((\neg x_u \wedge \neg y_u) \wedge z_u \rightarrow \neg on_u).$$

[2] Although the same symbols (i.e. x_y and y_x) are used to represent both thresholds and propositions, we can clearly distinguish them from the context.

Total Order of Thresholds. Since a gene may have multiple thresholds, we need to specify a total order of them. Assume that gene x has thresholds $x_1, x_2, \ldots, x_m$ in this order. This order relation can be described in LTL as follows:

$$\bigwedge_{1 \le i < m-1} G(x_{i+1} \rightarrow x_i).$$

For example, $G(x_2 \rightarrow x_1)$ means that if the current expression level is beyond the threshold x_2, it is also beyond x_1 since x_1 is lower than x_2. Note that the proposition x_i is interpreted as gene x is expressed beyond the threshold x_i.

Change of Expression Levels When Genes Are ON. Assume that gene x has its thresholds $x_1, x_2, \ldots, x_m$ in this order. If gene x is ON, the expression level of x increases over time. For example, if the current level of gene x is between x_i and x_{i+1} and x is ON, x will cross the threshold x_{i+1} in future (if gene x does not become OFF prematurely). This fact is simply described as follows:

$$G(on_x \wedge x_i \rightarrow (x_i U (x_{i+1} \vee \neg on_x)))$$

where $i \in \{1, \ldots, m-1\}$. This formula says that gene x *must* cross the threshold x_{i+1} unless gene x becomes OFF. That is, we do not allow that the expression level of gene x can be equilibrated between the level x_i and x_{i+1} if gene x is indefinitely ON. This specification is called *strong specification*. To allow such equilibrated behaviour, we specify as:

$$G(on_x \wedge x_i \rightarrow (x_i W \neg on_x))$$

where $i \in \{1, \ldots, m-1\}$. This kind of specification is called *weak specification*. The choice of strong specification and weak specification depends on a situation or an assumption of the analysis which we are to perform.

We need special treatment for the level below x_1 (the lowest threshold) and the level above x_m (the highest threshold). In the case that the expression level of x is below x_1, none of the propositions among $x_1, \ldots, x_m$ are true. If gene x is ON, it will cross x_1 in future (unless gene x becomes OFF prematurely). This can be described as:

$$G(on_x \rightarrow F(x_1 \vee \neg on_x)).$$

If gene x is expressed above x_m, since we do not have the threshold over it, its expression level does not increase further. Instead, gene x will keep its level (unless gene x becomes OFF). This can be described as:

$$G(on_x \wedge x_m \rightarrow (x_m W \neg on_x)).$$

Change of Expression Level When Genes Are OFF. We also assume that gene x has its thresholds $x_1, x_2, \ldots, x_m$ in this order. The specification for the case where genes are OFF is symmetric to the case where genes are ON. Thus we only show formulae.

$$G(\neg on_x \wedge \neg x_i \rightarrow (\neg x_i U (\neg x_{i-1} \vee on_x))), \quad \text{(strong)}$$
$$G(\neg on_x \wedge \neg x_i \rightarrow (\neg x_i W on_x)), \quad \text{(weak)}$$
$$G(\neg on_x \rightarrow F(\neg x_m \vee on_x)),$$
$$G(\neg on_x \wedge \neg x_1 \rightarrow (\neg x_1 W \neg on_x)).$$

3 WEAK HOMEOSTASIS AS STRONG SATISFIABILITY

In this section we show how we formulate weak homeostasis of a gene network by the notion of strong satisfiability of reactive system specifications (Mori and Yonezaki, 1993).

A reactive system is a system which reacts to external events from an environment and produces output events or controls its internal states in appropriate timing. How it reacts is dictated by specifications. LTL is known to be suitable to write reactive system specifications formally (Pnueli and Rosner, 1989; Abadi et al., 1989). Formally, a reactive system specification is represented as the triple $\langle E, I, \varphi \rangle$ where E is a set of external propositions (corresponding to external events), I is a set of internal propositions (corresponding to internal or output events) and φ is an LTL formula consists of atomic propositions from $E \cup I$. Then the notion of strong satisfiability of a reactive system specification is defined as follows.

Definition 1 (Strong Satisfiability)**.** *LTL specification* $\langle E, I, \varphi \rangle$ *is* strongly satisfiable *if*

$$\forall \tilde{x} \in (2^E)^\omega \exists \tilde{y} \in (2^I)^\omega . \langle \tilde{x}, \tilde{y} \rangle \models \varphi.$$

Here $\tilde{x} = x_0 x_1 \ldots$ (each $x_i \subseteq E$), $\tilde{y} = y_0 y_1 \ldots$ (each $y_i \subseteq I$) and $\langle \tilde{x}, \tilde{y} \rangle = (x_0 \cup y_0)(x_1 \cup y_1) \ldots$.

Intuitively a specification $\langle E, I, \varphi \rangle$ is strongly satisfiable if for any infinite sequence of external propositions there exists an infinite sequence of internal propositions such that its behaviour satisfies the specification φ.

Now we consider the relationship of this notion to homeostasis of gene networks. Homeostasis is informally stated as the tendency of a system to maintain its internal condition desirable against any situation or stimulus. In other words, the problem of analysing homeostasis of a gene network is to check whether a network satisfies a given property against *any external input sequence*. The purpose of this section is to present a formal definition for this problem.

A gene network can be regarded as reactive systems, since it reacts to external inputs (e.g. from environment or other cells) and determines its internal states. In section 2.3, we show how we specify possible behaviours of a given network in LTL. Then we can regard a behaviour specification of a network (say φ) as a reactive system specification, if we determine which propositions correspond to inputs or outputs. For example, let us consider the example network depicted in Fig. 1 again. In that network we do not have external inputs. Thus we assume gene x accepts positive inputs from environment. Then we introduce two propositions in_x and e_x. The proposition in_x represents whether input is coming and e_x represents whether the level of the input is beyond the threshold above which gene x is activated. Then we have the following propositions: $\{in_x, on_x, on_y, on_z, e_x, x_y, y_z\}$. The division of external propositions E and internal propositions I is as follows: $E = \{in_x\}$, $I = \{on_x, on_y, on_z, e_x, x_y, y_z\}$. Note that the environment only controls in_x which means the environment is only able to determine whether it gives the input to gene x or not. Whether the level of input exceeds the level e_x is determined by the behaviour specification. Thus e_x is internal propositions. The specification for change of levels of inputs is the same as the case of gene expressions:

$$G(in_x \rightarrow F(e_x \vee \neg in_x))$$
$$G(in_x \wedge e_x \rightarrow (e_x W \neg in_x))$$
$$G(\neg in_x \rightarrow F(\neg e_x \vee in_x))$$
$$G(\neg in_x \wedge \neg e_x \rightarrow (\neg e_x W in_x))$$

Now we introduce a network property ψ (specified in LTL) of a given network which represents a desirable function of the network. We are to check whether the property holds against *any* input sequences. We can give any property like stability (e.g. a certain gene is always ON) or oscillation (e.g. when a gene is ON, it will later be OFF) in LTL.

The problem of checking whether a network whose behaviours are specified by φ satisfies ψ for *any* input sequence is formally stated as follows.

Definition 2. *Let E be the set of external propositions, I be the set of internal propositions and E and I are disjoint. Let $AP = E \cup I$ be the set of atomic propositions. A property ψ is* weakly homeostatic *with respect to a behaviour specification of a network $\langle E, I, \varphi \rangle$ if $\langle E, I, \varphi \wedge \psi \rangle$ is strongly satisfiable. Here φ and ψ are written in LTL with AP.*

Note that Ito et al.'s definition of homeostasis is that $\langle E, I, \varphi \wedge \psi \rangle$ is *realisable* (Ito et al., 2014). The reason why this definition is *weak* homeostasis is that the network is not required to determine its internal states at some time-point only from the input sequences which is available *at that time* (i.e. finite input sequences), which is the requirement for realisable specifications. In other words, the network can use *infinite* input sequences to determine its internal states at any time. Thus the homeostasis we capture in this definition is *weak* compared to that of Ito et al.'s definition based on realisability. Although the homeostasis we capture in this work is weak, we still have an biological insight for a homeostasis of gene networks. Since strong satisfiability is necessary condition of realisability, if a specification is proved to be not strongly satisfiable (i.e. weakly homeostatic), we see that it is not realisable (i.e. homeostatic).

This definition reduces the problem of checking weak homeostasis to the problem of checking strong satisfiability of reactive system specifications. Unfortunately, the complexity of strong satisfiability checking of LTL formula is EXPSPACE-complete in the size of formulae (Shimakawa et al., 2013), which is still high. In our framework, the size of a formula obtained from a gene network is proportional to the size of the network. Due to the high-complexity of strong satisfiability checking, direct analyses of large networks are generally intractable. In the next section we introduce an approximate method to ease the analysis of large networks.

4 APPROXIMATE ANALYSIS

The factor which is critical to the performance of strong satisfiability checking is the size of a formula. Thus reducing the size of a formula is a natural solution to overcome this computational difficulty. However, it is unclear that we can safely reduce the size of a formula.

Ito et al. (Ito et al., 2013b) proved that such safe reduction of the size of a formula is feasible. They approximate the set of possible behaviours of a given network using fewer propositions. Their approximate method guarantees that if an approximate specification is *satisfiable*, the original specification is also *satisfiable*. Here we say a formula φ is satisfiable if there exists a behaviour σ such that $\sigma \models \varphi$. Since satisfiability is a weaker property than strong satisfiability, it is unclear that their method is also feasible in analysing weak homeostasis of a gene network.

This section extends their result to analyse weak homeostasis of a gene network. Intuitively, the idea of the approximate method is to shrink the set of possible behaviours by approximate specifications. This means that the network has fewer choices to react to the environmental inputs. If we can prove that the

network can still respond to any environmental inputs in such restricted choices compared to original behaviour sets, it guarantees that the network surely reacts to any inputs. For formal development of our approximate analysis, we first introduce some notions and related theorems.

Definition 3. *Let φ be an LTL formula. $Prop(\varphi)$ denotes the set of propositions occurring in φ. Moreover, if $Prop(\varphi)$ is partitioned into external propositions and internal propositions, $EP(\varphi)$ denotes the set of external propositions occurring in φ and $IP(\varphi)$ denotes the set of internal propositions occurring in φ. Clearly $EP(\varphi) \cap IP(\varphi) = \emptyset$ and $EP(\varphi) \cup IP(\varphi) = Prop(\varphi)$.*

The next definition is of the Büchi automaton, which is a kind of ω-automata accepting infinite words.

Definition 4. *A* Büchi automaton *is a quintuple $\langle Q, \Sigma, \delta, q_I, F\rangle$, where Q is a finite set of states, Σ is a finite alphabet, $\delta : Q \times \Sigma \to \mathfrak{P}(Q)$ is the state transition function, $q_I \in Q$ is the initial state, and $F \subseteq Q$ is the set of accepting states. A* run *of a Büchi automaton on an infinite word $\alpha = \alpha[0]\alpha[1]\cdots \in \Sigma^\omega$ is an infinite sequence $\rho = \rho[0]\rho[1]\cdots \in Q^\omega$, such that $\rho[0] = q_I$ and $\rho[i+1] \in \delta(\rho[i], \alpha[i])$ for all $i \geq 0$. An infinite word α is* accepted *by the automaton if the run over α visits at least one state in F infinitely often. We denote the set of infinite words accepted by an automaton $\mathcal{A}$ by $L(\mathcal{A})$.*

The next theorem (Vardi and Wolper, 1994) states that we can construct a Büchi automaton that exactly accepts the models of LTL formula φ.

Theorem 1. *Given an LTL formula φ, one can construct a Büchi automaton $\mathcal{A}_\varphi = \langle Q, \Sigma, \delta, q_I, F\rangle$ such that $|Q|$ is in $2^{O(|\varphi|)}$, $\Sigma = 2^{Prop(\varphi)}$ and $L(\mathcal{A}_\varphi) = \{\sigma \in (2^{Prop(\varphi)})^\omega \mid \sigma \models \varphi\}$.*

The above theorem says that the set of time structures which satisfies formula φ is obtained by $L(\mathcal{A}_\varphi)$.

Definition 5. *Let $A \subseteq B$ and $\sigma \in (2^B)^\omega$. We denote $\sigma|_A$ for the pointwise restriction of σ on A, i.e. $\sigma|_A = (\sigma[0]|_A)(\sigma[1]|_A)\ldots$. Assume $L \subseteq (2^B)^\omega$. We denote $L|_A$ for the element-wise restriction of set L on A, i.e. $L|_A = \{\sigma|_A \mid \sigma \in L\}$.*

Then we introduce an approximate relation between LTL formulae.

Definition 6. *Let φ and φ' be LTL formulae such that $EP(\varphi) = EP(\varphi')$ and $IP(\varphi') \subseteq IP(\varphi)$. We define the relation $\sqsubseteq$ as follows:*

$$\varphi' \sqsubseteq \varphi \overset{\text{def}}{\Longleftrightarrow} L(\mathcal{A}_{\varphi'}) \subseteq L(\mathcal{A}_\varphi)|_{Prop(\varphi')}$$

Note that $Prop(\varphi') \subseteq Prop(\varphi)$.

We say that φ' is a *lower approximation* of φ if $\varphi' \sqsubseteq \varphi$. The formula φ' has fewer propositions than φ. Our approximate method is to check strong satisfiability of the specification $\langle EP(\varphi'), IP(\varphi'), \varphi'\rangle$ instead of $\langle EP(\varphi), IP(\varphi), \varphi\rangle$ in checking strong satisfiability. To guarantee the correctness of this approximate method, we need to prove that if the approximate specification $\langle EP(\varphi'), IP(\varphi'), \varphi'\rangle$ is strongly satisfiable, so is $\langle EP(\varphi), IP(\varphi), \varphi\rangle$. The rest of this section is devoted to prove this correctness.

First we prove the following lemma.

Lemma 1. *Assume $\varphi' \sqsubseteq \varphi$. For any $\sigma' \in (2^{Prop(\varphi')})^\omega$, if $\sigma' \models \varphi'$ then there exists $\sigma \in (2^{Prop(\varphi)})^\omega$ such that $\sigma \models \varphi$ and $\sigma|_{Prop(\varphi')} = \sigma'$.*

Proof. By definition of $\varphi' \sqsubseteq \varphi$, we have $Prop(\varphi') \subseteq Prop(\varphi)$ and $L(\mathcal{A}_{\varphi'}) \subseteq L(\mathcal{A}_\varphi)|_{Prop(\varphi')}$. Suppose $\sigma' \models \varphi'$ for $\sigma' \in (2^{Prop(\varphi')})^\omega$, we have $\sigma' \in L(\mathcal{A}_{\varphi'})$ by theorem 1. By assumption we have $\sigma' \in L(\mathcal{A}_\varphi)|_{Prop(\varphi')}$. By definition 5, there exists $\sigma \in (2^{Prop(\varphi)})^\omega$ such that $\sigma \in L(\mathcal{A}_\varphi)$ and $\sigma|_{Prop(\varphi)} = \sigma'$. □

From this theorem we immediately have the following:

Corollary 1. *If $\varphi' \sqsubseteq \varphi$, there is a mapping $\ell_{\varphi',\varphi} : (2^{Prop(\varphi')})^\omega \to (2^{Prop(\varphi)})^\omega$ such that if $\sigma' \models \varphi'$ then $\ell_{\varphi',\varphi}(\sigma') \models \varphi$.*

Now we prove our main theorem.

Theorem 2. *Suppose $\varphi' \sqsubseteq \varphi$. If $\langle EP(\varphi'), IP(\varphi'), \varphi'\rangle$ is strongly satisfiable then $\langle EP(\varphi), IP(\varphi), \varphi\rangle$ is strongly satisfiable.*

Proof. Since φ' is strongly satisfiable, for any $\tilde{x} \in (2^{EP(\varphi')})^\omega$ there exists $\tilde{y} \in (2^{IP(\varphi')})^\omega$ such that $\langle \tilde{x}, \tilde{y}\rangle \models \varphi'$. By definition of $\varphi' \sqsubseteq \varphi$, we have $\langle \tilde{x}, \tilde{y}\rangle \in L(A_\varphi)|_{Prop(\varphi')}$. From corollary 1, there exists a function $\ell_{\varphi',\varphi}$ such that $\ell_{\varphi',\varphi}(\langle \tilde{x}, \tilde{y}\rangle) \models \varphi$. Since $EP(\varphi') = EP(\varphi)$, we have $\langle \tilde{x}, \tilde{z}\rangle = \ell_{\varphi',\varphi}(\langle \tilde{x}, \tilde{y}\rangle)$ for some $\tilde{z} \in (2^{IP(\varphi)})^\omega$. □

To prove the correctness of approximate analysis of weak homeostasis, we need to prove the following corollary (proof is omitted).

Corollary 2. *Suppose $\varphi' \sqsubseteq \varphi$ and $Prop(\psi) \subseteq Prop(\varphi')$. If $\langle EP(\varphi' \wedge \psi), IP(\varphi' \wedge \psi), \varphi' \wedge \psi\rangle$ is strongly satisfiable then $\langle EP(\varphi \wedge \psi), IP(\varphi \wedge \psi), \varphi \wedge \psi\rangle$ is strongly satisfiable.*

Thanks to corollary 2, in analysing weak homeostasis of a network whose behaviour specification is φ, we can simplify the specification φ to φ' such that $\varphi' \sqsubseteq \varphi$. The problem is that it is unclear whether we can systematically obtain such approximate specification φ' for any LTL formula φ. Ito et al., however, showed that for a specific class of networks,

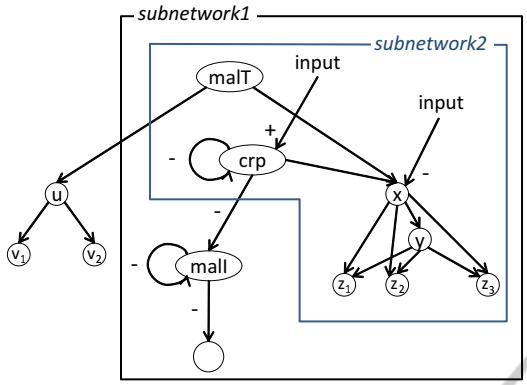

Figure 4: A network in *E. coli*.

called *network motifs*, we have approximate specifications (Ito et al., 2013b). Network motifs are common network patterns recurring in many gene networks (Alon, 2007). Thus approximate specifications for network motifs are useful when we analyse real gene networks. We use Ito et al.'s approximate specifications for five network motifs (Ito et al., 2013b), negative auto-regulation, coherent type 1 feed-forward loop, incoherent type 1 feed-forward loop, single-input module and multi-output feed-forward loop. We cannot find space for showing them. Interested reader may wish to consult it.

5 EXPERIMENTAL RESULTS

In this section we show experimental results of our approximate method in analysing weak homeostasis of gene networks. For the experiment we use a network in *Escherichia coli* (Alon, 2007) depicted in Fig. 4 and a network in *Arabidopsis thaliana* which is obtained from ReIN[3] and is depicted in Fig. 5. In the network of Fig. 4, we have one single-input module (consisting of gene u, v_1 and v_2), two negative auto-regulations (gene *crp* and *malI*), and one multi-output feed-forward loop (consisting of gene x, y, z_1, z_2 and z_3). In Fig. 5, we have one negative auto-regulation (gene *AP2*) and one single-input module whose master gene is *GL1/GL3* and target genes are those regulated by the master gene. Some target genes have another regulator but such case can be easily taken into consideration in the approximate specification.

We show the part of the behaviour specification of the network of Fig. 4.

[3]http://arabidopsis.med.ohio-state.edu/REIN/

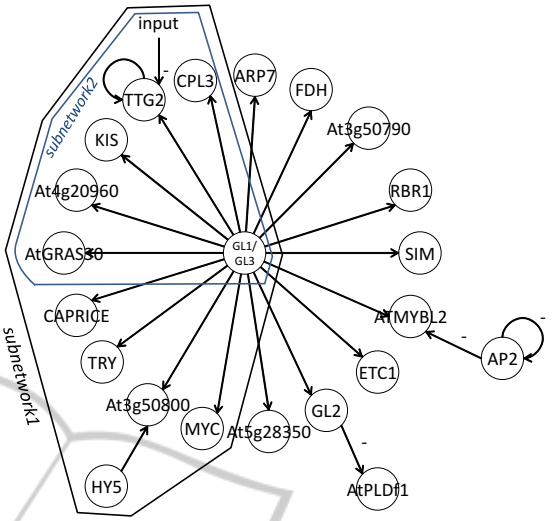

Figure 5: A network in *Arabidopsis*.

$$G(u_{v_1} \leftrightarrow on_{v_1}) \wedge$$
$$G(u_{v_2} \leftrightarrow on_{v_2}) \wedge$$
$$G(on_u \rightarrow F(u_{v_1} \vee \neg on_u)) \wedge$$
$$G((on_u \wedge u_{v_1}) \rightarrow (u_{v_1} U(\neg on_u \vee u_{v_2}))) \wedge$$
$$G((on_u \wedge u_{v_2}) \rightarrow (u_{v_2} W \neg on_u)) \wedge \ldots$$

This specification can be approximated as:

$$G(on_u \rightarrow F(on_{v_1} \vee \neg on_u)) \wedge$$
$$G((on_u \wedge on_{v_1}) \rightarrow (on_{v_1} W \neg on_u)) \wedge$$
$$G((on_u \wedge on_{v_2}) \rightarrow (on_{v_2} W \neg on_u)) \wedge \ldots$$

As we can see, we no longer use propositions u_{v_1} and u_{v_2}.

In this experiment we use three variations of specifications for each network – specification for the entire network and its two subnetworks. Subnetworks are obtained by eliminating some genes and edges from the entire network as shown in Fig. 4 and 5. We assume that the network depicted in Fig. 4 receives two inputs and the network depicted in Fig. 5 receives one input, as depicted. The property we consider in this experiment is that if a certain gene is activated, it will be suppressed afterward. We consider the same type of property for both of the networks. This property is described as:

$$G(on \rightarrow F \neg on)$$

where *on* proposition is for gene z_1 in the network Fig. 4 and for gene *TTG2* in the network Fig. 5. This amounts to check whether the networks of Fig. 4 (Fig. 5) can suppress gene z_1 (*TTG2*) against any environmental input sequence. In the network of Fig. 4, gene z_1 is activated by gene x and gene x receives the negative input. Thus when the negative input never comes,

Table 1: Experimental results for checking weak homeostasis of networks (i.e. we have no assumptions of environmental inputs). Columns 'E' and 'I' respectively show the numbers of external propositions and internal propositions. Column 'S' shows the size of a formula. Column 'T' shows the time of analysis (in seconds). The lower half of the table shows the result of approximate analysis.

Network	E	I	S	T
Fig. 4	2	30	679	>3600
Fig. 4 (sub1)	2	24	572	>3600
Fig. 4 (sub2)	2	19	459	408.87
Fig. 5	1	46	869	>3600
Fig. 5 (sub1)	1	23	449	>3600
Fig. 5 (sub2)	1	13	269	0.10
Fig. 4 (appr.)	2	18	448	176.80
Fig. 4 (appr.) (sub1)	2	14	354	8.35
Fig. 4 (appr.) (sub2)	2	10	278	0.752
Fig. 5 (appr.)	1	30	692	11.07
Fig. 5 (appr.) (sub1)	1	16	379	0.17
Fig. 5 (appr.) (sub2)	1	9	231	0.04

gene x is easy to be ON and afterward gene z_1 will be ON. However, *malT* is the gene necessary to activate gene x, and its expression can be controlled by the network. Thus the network can control *malT* to be OFF so that gene x cannot be ON. Therefore the property is homeostatic. Similar informal reasoning shows that the property for the network of Fig. 5 is also homeostatic.

We show the results of each analysis in table 1. These experiments are performed on a computer with Intel Core i7-3820 3.60GHz CPU and 32GB memory. We used Shimakawa et al.'s strong satisfiability checker (Shimakawa et al., 2014) for this experiment.

In example analyses reported in table 1, non-approximated analyses were successful only for the smallest network specifications (subnetwork2 for both networks). In approximated analyses, however, all specifications were successful. By comparing the results of the network specification Fig.4 (sub2) and its approximated version, we see that approximated method improves the analysis speed by 540 times. These results show that our approximate method is effective.

6 CONCLUSIONS

In this paper we presented an approximate method for analysing homeostasis of gene networks using network motifs. We are investigating that whether our approximate analysis method can be used to check not only strong satisfiability but also realisability of LTL to enable approximate analysis of homeostasis based on the formulation by realisability (Ito et al., 2014). For further improvement, we are interested in whether Ito et al.'s modular method (Ito et al., 2013a) is available in analysing (weak) homeostasis of gene networks. Since modular analysis can be used in combination with the approximate analysis, we further extend the limits of tractable networks. Using these results, we now should try to solve real problems in biology.

ACKNOWLEDGEMENTS

This work was supported by JSPS KAKENHI Grant Number 26730153. We also appreciate the help of Dr. Masaya Shimakawa and Dr. Takashi Tomita for checking proofs and giving invaluable comments.

REFERENCES

Abadi, M., Lamport, L., and Wolper, P. (1989). Realizable and unrealizable specifications of reactive systems. In *ICALP '89: Proceedings of the 16th International Colloquium on Automata, Languages and Programming*, volume 372 of *LNCS*, pages 1–17, London, UK. Springer-Verlag.

Alon, U. (2007). Network motifs: theory and experimental approaches. *Nat. Rev. Genet.*, 8(6):450–461.

Emerson, E. A. (1990). Temporal and modal logic. In van Leeuwen, J., editor, *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B)*, pages 995–1072. MIT Press.

Ito, S., Hagihara, S., and Yonezaki, N. (2014). A qualitative framework for analysing homeostasis in gene networks. In *Proceedings of the 5th International Conference on Bioinformatics Models, Methods and Algorithms (BIOINFORMATICS2014)*, pages 5–16.

Ito, S., Ichinose, T., Shimakawa, M., Izumi, N., Hagihara, S., and Yonezaki, N. (2013a). Modular analysis of gene networks by linear temporal logic. *J. Integrative Bioinformatics*, 10(2).

Ito, S., Ichinose, T., Shimakawa, M., Izumi, N., Hagihara, S., and Yonezaki, N. (2013b). Qualitative analysis of gene regulatory networks using network motifs. In *Proceedings of the 4th International Conference on Bioinformatics Models, Methods and Algorithms (BIOINFORMATICS2013)*, pages 15–24.

Ito, S., Izumi, N., Hagihara, S., and Yonezaki, N. (2010). Qualitative analysis of gene regulatory networks by satisfiability checking of linear temporal logic. In *Proceedings of the 10th IEEE International Conference on Bioinformatics & Bioengineering*, pages 232–237.

Mori, R. and Yonezaki, N. (1993). Several realizability concepts in reactive objects. In *Information Modeling and Knowledge Bases IV*, pages 407–424.

Pnueli, A. and Rosner, R. (1989). On the synthesis of a reactive module. In *POPL '89: Proceedings of the 16th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 179–190, New York, NY, USA. ACM.

Shimakawa, M., Hagihara, S., and Yonezaki, N. (2013). Complexity of strong satisfiability problems for reactive system specifications. *IEICE Transactions*, 96-D(10):2187–2193.

Shimakawa, M., Hagihara, S., and Yonezaki, N. (2014). Bounded strong satisfiability checking of reactive system specifications. *IEICE Transactions*, 97-D(7):1746–1755.

Vardi, M. Y. and Wolper, P. (1994). Reasoning about infinite computations. *Inf. Comput.*, 115:1–37.