Signaling P Systems and Verification Problems ⋆

Cheng Li[1], Zhe Dang[1]⋆⋆, Oscar H. Ibarra[2], and Hsu-Chun Yen[3]

[1]School of Electrical Engineering and Computer Science
Washington State University
Pullman, WA 99164, USA

[2]Department of Computer Science
University of California
Santa Barbara, CA 93106, USA

[3]Department of Electrical Engineering
National Taiwan University, Taipei, Taiwan 106, R.O.C.

Abstract. We introduce a new model of membrane computing system (or P system), called signaling P system. It turns out that signaling systems are a form of P systems with promoters that have been studied earlier in the literature. However, unlike non-cooperative P systems with promoters, which are known to be universal, non-cooperative signaling systems have decidable reachability properties. Our focus in this paper is on verification problems of signaling systems; i.e., algorithmic solutions to a verification query on whether a given signaling system satisfies some desired behavioral property. Such solutions not only help us understand the power of "maximal parallelism" in P systems but also would provide a way to validate a (signaling) P system in vitro through digital computers when the P system is intended to simulate living cells. We present decidable and undecidable properties of the model of non-cooperative signaling systems using proof techniques that we believe are new in the P system area. For the positive results, we use a form of "upper-closed sets" to serve as a symbolic representation for configuration sets of the system, and prove decidable symbolic model-checking properties about them using backward reachability analysis. For the negative results, we use a reduction via the undecidability of Hilbert's Tenth Problem. This is in contrast to previous proofs of universality in P systems where almost always the reduction is via matrix grammar with appearance checking or through Minsky's two-counter machines. Here, we employ a new tool using Diophantine equations, which facilitates elegant proofs of the undecidable results. With multiplication being easily implemented under maximal parallelism, we feel that our new technique is of interest in its own right and might find additional applications in P systems.

1 Introduction

P systems [19, 20] are abstracted from the way the living cells process chemical compounds in their compartmental structure. A P system consists of a finite number of membranes, each of which contains a multiset of objects (symbols). The membranes are organized as a Venn diagram or a tree structure where a membrane may contain other membranes. The dynamics of the P system is governed by a set of rules associated with each membrane. Each rule

⋆ The work by Cheng Li and Zhe Dang was supported in part by NSF Grant CCF-0430531. The work by Oscar H. Ibarra was supported in part by NSF Grants CCR-0208595 and CCF-0430945.

⋆⋆ Corresponding author (`zdang@eecs.wsu.edu`).

specifies how objects evolve and move into neighboring membranes. In particular, a key feature of the model of P systems is that rules are applied in a nondeterministic and maximally parallel manner. Despite the short (only five years) history of membrane computing, there has already been a notably large collection of papers in the area (see the P systems website: psystems.disco.unimb/it) and membrane computing has been selected as a fast "Emerging Research Front" in Computer Science by the Institute for Scientific Information (ISI) (esi-topics.com/erf/october2003.html). Due to the key feature inherent in the model, P systems have a great potential for implementing massively concurrent systems in an efficient way that would allow us to solve currently intractable problems (in much the same way as the promise of quantum and DNA computing). It turns out that P systems are a powerful model: even with only one membrane (i.e., 1-region P systems) and without priority rules, P systems are already universal [19, 23]. In such a one-membrane P system, rules are in the form of $u \to v$, which, in a maximally parallel manner, replaces multiset u (in current configuration which is a multiset of symbol objects) with multiset v.

Signals are a key to initiate biochemical reactions between and inside living cells. Many examples can be found in a standard cell biology textbook [3]. For instance, in signal transduction, it is known that guanine-nucleotide binding proteins (G proteins) play a key role. A large heterotrimeric G protein, one of the two classes of G proteins, is a complex consisting of three subunits: G_α, G_β, and G_γ. When a ligand binds to a G protein-linked receptor, it serves as a signal to activate the G protein. More precisely, the GDP, a guanine nucleotide, bound to the G_α subunit in the unactivated G protein is now displaced with GTP. In particular, the G protein becomes activated by being dissociated into a G_β-G_γ complex and a G_α-GTP complex. Again, the latter complex also serves as a signal by binding itself to the enzyme adenylyl cyclase. With this signal, the enzyme becomes active and converts ATP to cyclic AMP. As another example, apoptosis (i.e., suicide committed by cells, which is different from necrosis, which is the result from injury) is also controlled by death signals such as a CD95/Fas ligand. The signal activates caspase-8 that initiates the apoptosis. Within the scope of Natural Computing (which explores new models, ideas, paradigms from the way nature computes), motivated by these biological facts, it is a natural idea to study P systems, a molecular computing model, augmented with a signaling mechanism.

In this paper, we investigate one-membrane signaling P systems (signaling systems in short) where the rules are further equipped with signals. More precisely, in a signaling system M, we have two types of symbols: object symbols and signals. Each configuration is a pair consisting of a set S of signals and a multiset α of objects. Each rule in M is in the form of $s, u \to v, s'$ or $s, u \to \Lambda$, where s, s' are signals and u, v are multisets of objects. The rule is enabled in the current configuration (S, α) if s is present in the signal set S and u is a sub-multiset of the multiset α. All the rules are fired in maximally parallel manner. In particular, in the configuration as a result of the maximally parallel move, the new signal set is formed by collecting the set of signals s' that are emitted from all the rules actually fired during the move (and every signal in the old signal set disappears). Hence, a signal may trigger an unbounded number of rule instances in a maximally parallel move.

We focus on verification problems of signaling systems; i.e., algorithmic solutions to a verification query on whether a given signaling system does satisfy some desired behavioral property. Such solutions not only help us understand the power of the maximally parallelism that is pervasive in P systems but also would provide a way to validate a (signaling) P system in vitro through digital computers when the P system is intended to simulate living cells. However, since one-membrane P systems are Turing-complete, so are signaling systems. Therefore, to study the verification problems, we have to look at restricted signaling systems. A signaling system is non-cooperative if each rule is in the form of $s, a \to \Lambda$ or in the form of $s, a \to bc, s'$, where a, b, c are object symbols. All the results can be generalized to non-cooperative signaling systems augmented with rules $s, a \to v, s'$. We study various reachability queries for non-cooperative signaling systems M; i.e., given two formulas $Init$ and $Goal$ that define two sets of configurations, are there configurations C_{init} in $Init$ and C_{goal} in $Goal$ such that C_{init} can reach C_{goal} in zero or more maximally parallel moves in M? We show that, when $Init$ is a Presburger formula (roughly, in which one can compare integer linear constraints over multiplicities of symbols against constants) and $Goal$ is a region formula (roughly, in which one can compare multiplicities of symbols against constants), the reachability query is decidable. Notice that, in this case, common reachability queries like halting and configuration reachability are expressible. We also show that introducing signals into P systems indeed increases its computing power; e.g., non-cooperative signaling systems are strictly stronger than non-cooperative P systems (without signals). On the other hand, when $Goal$ is a Presburger formula, the query becomes undecidable. Our results generalize to queries expressible in a subclass of a CTL temporal logic and to non-cooperative signaling systems with rules $S, a \to v, S'$ (i.e., the rule is triggered with a set of signals in S). We also study the case when a signal has bounded strength and, in this case, non-cooperative signaling systems become universal.

Non-cooperative signaling systems are also interesting for theoretical investigation, since the signaling rules are context-sensitive and the systems are still nonuniversal as we show. In contrast to this, rules $a \to v$ in a non-cooperative P system are essentially context-free. It is difficult to identify a form of restricted context-sensitive rules that are still nonuniversal. For instance, a communicating P system (CPS) with only one membrane [22] is already universal, where rules are in the form of $ab \to a_x b_y$ or $ab \to a_x b_y c_{come}$ in which a, b, c are objects, x, y (which indicate the directions of movements of a and b) can only be $here$ or out. Also one membrane catalytic systems with rules like $Ca \to Cv$ (where C is a catalytic) are also universal. More examples including non-cooperative signaling systems with promoters, which will be discussed further in this section, are also universal. Our non-cooperative signaling systems use rules in the form of $s, a \to v, s'$, which are in a form of context-sensitive rules, since the signals constitute part of the triggering condition as well as the outcome of the rules.

At the heart of our decidability proof, we use a form of upper-closed sets to serve as a symbolic representation for configuration sets and prove that the symbolic representation is invariant under the backward reachability relation of a non-cooperative signaling system. From the studies in symbolic model-checking [7] for classic transition systems, our symbolic representation also demonstrates a symbolic model-checking procedure at least for reachability. In our undecidability proofs, we use the well-known result on the Hilbert's Tenth Problem: any r.e. set (of integer tuples) is also Diophantine. We note that, for P systems that deal with symbol objects, proofs for universality almost always use the theoretical tool through matrix grammar with appearance checking [17] or through Minsky's two-counter machines. Here, we employ a new tool using Diophantine equations, which facilitates elegant proofs of the undecidable results. With multiplication being easily implemented under maximal parallelism, we feel that our new technique is of interest in its own right and might find additional applications in P systems.

Signaling mechanisms have also been noticed earlier in P system studies. For instance, in a one-membrane P system with promoters [4], a rule is in the form of $u \to v|p$ where p is a multiset called a promoter. The rule fires as usual in a maximally parallel manner but only when objects in the promoter all appear in the current configuration. Notice that, since p may not be even contained in u, a promoter, just as a signal, may trigger an unbounded number of rule instances. Indeed, one can show that a signaling system can be directly simulated by a one-membrane P system with promoters. However, since one-membrane non-cooperative P systems with promoters are known to be universal [4], our decidability results on non-cooperative signaling systems have a nice implication: our signals are strictly weaker than promoters (and hence have more decidable properties). The decidability results also imply that, as shown in the paper, non-cooperative signaling systems and vector addition systems (i.e., Petri nets) have incomparable computing power, though both models have a decidable configuration-to-configuration reachability. This latter implication indicates that the maximal parallelism in P systems and the "true concurrency" in Petri nets are different parallel mechanisms. Other signaling mechanisms such as in [2] are also promoter-based.

All the proofs can be found in the Appendix, which may be read by the PC members at their discretion.

2 Preliminaries

We use $\mathbf{N}$ to denote the set of natural numbers (including 0) and use $\mathbf{Z}$ to denote the set of integers. Let $\Sigma = \{a_1, \ldots, a_k\}$ be an alphabet, for some k, and α be a (finite) multiset over the alphabet. In this paper, we do not distinguish between different representations of the multiset. That is, α can be treated as a vector in $\mathbf{N}^k$ (the components are the multiplicities of the symbols in Σ); α can be treated as a word on Σ where we only care about the counts of symbols (i.e., its Parikh map). For a $\sigma \subseteq \Sigma$, we use σ^* to denote the set of all multisets on σ.

A set $S \subseteq \mathbf{N}^k$ is a *linear set* if there exist vectors $v_0, v_1, \ldots, v_t$ in $\mathbf{N}^k$ such that $S = \{v \mid v = v_0 + a_1 v_1 + \ldots + a_t v_t,\ a_i \in \mathbf{N}\}$. A set $S \subseteq \mathbf{N}^k$ is *semilinear* if it is a finite union of linear sets. Let $x_1, \ldots, x_k$ be variables on $\mathbf{N}$. A *Presburger formula* is a Boolean combination of linear constraints in the following form: $\sum_{1 \leq i \leq k} t_i \cdot x_i \sim n$, where the t_i's and n are integers in $\mathbf{Z}$, and $\sim \in \{>, <, =, \geq, \leq, \equiv_m\}$ with $0 \neq m \in \mathbf{N}$. It is known that a set of multisets (treated as vectors) is semilinear iff the set is definable by a Presburger formula. Also, Presburger formulas are closed under quantification.

An n-dimensional *vector addition system* (VAS) is a pair $G = \langle x, W \rangle$, where $x \in \mathbf{N}^n$ is called the *start point* (or *start vector*) and W is a finite set of *addition vectors* in $\mathbf{Z}^n$. The *reachability set* of the VAS $\langle x, W \rangle$ is the set $R(G) = \{z \mid$ for some j, $z = x + v_1 + \ldots + v_j$, where, for all $1 \leq i \leq j$, each $v_i \in W$ and $x + v_1 + \ldots + v_i \geq 0\}$. The *halting reachability set* $R_h(G) = \{z \mid z \in R(G), z + v \not\geq 0$ for every v in $W\}$. An n-dimensional *vector addition system with states* (VASS) is a 5-tuple $\langle x, W, p_0, S, \delta \rangle$ where x and W are the same as that in a VAS, S is a finite set of *states*, $\delta \subseteq S \times S \times W$ is the *transition relation*, and $p_0 \in S$ is the *initial state*. Elements (p, q, v) of δ are called *transitions* and are usually written as $p \to (q, v)$. A *configuration* of a VASS is a pair (p, u) where $p \in S$ and $u \in \mathbf{N}^m$. (p_0, x) is the *initial configuration*. The transition $p \to (q, v)$ can be applied to the configuration (p, u) and yields the configuration $(q, u + v)$, provided that $u + v \geq 0$ (in this case, we write $(p, u) \to (q, u + v)$). It is well-known that Petri nets, VAS, and VASS are all equivalent.

A signaling system is simply a P system [19] augmented with signals. Formally, a (1-membrane) *signaling system* M is specified by a tuple $\langle \Sigma, Sig, R \rangle$, where $\Sigma = \{a_1, \ldots, a_k\}$ is the alphabet, Sig is a nonempty finite set of *signals*, and R is a finite set of *rules*. Each rule is in the form of $s, u \to v, s'$, where $s, s' \in Sig$ and u and v are multisets over alphabet Σ. (Notice that a rule like $s, u \to v$ (without emitting signal) can be treated as a short hand of $s, u \to v, s_{\mathrm{garbage}}$ where s_{garbage} is a "grabage" signal that won't trigger any rules) A *configuration* C is a pair consisting of a set S of signals and a multiset α on Σ. As with the standard semantics of P systems [19–21], each evolution step, called a *maximally parallel move*, is a result of applying all the rules in M in a maximally parallel manner. More precisely, let $s_i, u_i \to v_i, s'_i$, $1 \leq i \leq m$, be all the rules in M. We use $\mathbf{R} = (r_1, \ldots, r_m) \in \mathbf{N}^m$ to denote a multiset of rules, where there are r_i instances of rule $s_i, u_i \to v_i, s'_i$, for each $1 \leq i \leq m$. Rule $s_i, u_i \to v_i, s'_i$ is *actually fired* in $\mathbf{R}$ if $r_i \geq 1$ (there is at least one instance of the rule in $\mathbf{R}$). Let $C = (S, \alpha)$ and $C' = (S', \alpha')$ be two configurations. The rule multiset $\mathbf{R}$ is *enabled* under configuration C if

- multiset α contains multiset $\cup_{1 \leq i \leq m} r_i \cdot u_i$ (i.e., the latter multiset is the multiset union of r_i copies of multiset u_i, for all $1 \leq i \leq m$), and
- set $S \supseteq \{s_i : r_i > 0, 1 \leq i \leq m\}$ (i.e., for every rule actually fired in $\mathbf{R}$, the signal s_i that triggers the rule must appear in the set S of the configuration C).

(We say that a rule is enabled under configuration C if the rule multiset that contains exactly one instance of the rule is enabled under the configuration.) The result $C' = (S', \alpha')$ of applying $\mathbf{R}$ over $C = (S, \alpha)$ is as follows: set S' is obtained by replacing the entire S by the new signal set formed by collecting all the signals s'_i emitted from the rules that are actually fired in $\mathbf{R}$, and, multiset α' is obtained by replacing, in parallel, each of the r_i copies of u_i in α with v_i. The rule multiset $\mathbf{R}$ is *maximally enabled* under configuration C if it is enabled under C and, for any other rule multiset $\mathbf{R}'$ that properly contains $\mathbf{R}$, $\mathbf{R}'$ is not enabled under the configuration. Notice that, for the same C, a maximally enabled rule multiset may not be unique (i.e., M is in general nondeterministic). C can reach C' through a maximally parallel move, written $C \to_M C'$, if there is a maximally enabled rule multiset $\mathbf{R}$ such that C' is the result of applying $\mathbf{R}$ over C. We use $C \leadsto_M C'$ to denote the fact that C' is reachable from C; i.e., for some n and $C_0, \ldots, C_n$, we have $C = C_0 \to_M \ldots \to_M C_n = C'$. We simply say that C is reachable if the initial configuration C is understood. We say that configuration C is *halting* if there is no rule enabled in C.

When the signals are ignored in a signaling system, we obtain a 1-membrane P system. Clearly, signaling systems are universal, since, as we have mentioned earlier, 1-membrane P systems are known to be universal. A non-cooperative signaling system is a signaling system where each rule is either a *split-rule* in the form of $s, a \to bc, s'$ or a *die-rule* in the form of $s, a \to \Lambda$, where $s, s' \in Sig$ and symbols $a, b, c \in \Sigma$. The two rules are called a-rules (since a appears at the LHS). Intuitively, the split-rule, when receiving signal s, makes an a-object split into a b-object and a c-object with signal s' emitted. On the other hand, the die-rule, when receiving signal s, makes an a-object die (i.e., becomes null). In particular, for a configuration C, an a-object is *enabled* in C if there is an enabled a-rule in C; in this case, we also call a to be *an enabled symbol in C*. In the rest of the paper, we will focus on various reachability queries for non-cooperative signaling systems.

3 Configuration Reachability of Non-cooperative Signaling Systems

We first investigate the *configuration-reachability* problem that decides whether one configuration can reach another.

Given: a non-cooperative signaling system M and two configurations C_{init} and C_{goal},
Question: Can C_{init} reach C_{goal} in M?

In this section, we are going to show that the problem is decidable. The proof performs backward reachability analysis. That is, we first effectively compute (a symbolic representation of) the set of all configurations C' such that $C' \leadsto_M C_{\text{goal}}$. Then, we decide whether the initial configuration C_{init} is in the set.

Before proceeding further, we first introduce the symbolic representation. Let $\mathcal{C}$ be a set of configurations. We say that $\mathcal{C}$ is *upper-closed* if $\mathcal{C} = \{(S, \alpha) : \alpha$ is the multiset union of β and some multiset in $\sigma^*\}$, for some $S \subseteq Sig$, multiset β and some symbol-set $\sigma \subseteq \Sigma$. In this case, we use $[S, \beta, \sigma^*]$ to denote the set $\mathcal{C}$. We say that $\mathcal{C}$ is *m-bounded* if $|\beta| \leq m$. Let $\mathcal{C}$ be a finite union of upper-closed sets of configurations. The *pre-image* of $\mathcal{C}$ is defined as $Pre_M(\mathcal{C}) = \{C' : C' \to_M C \in \mathcal{C}\}$. We use $Pre^*_M(\mathcal{C})$ to denote the set of all configurations C' such that $C' \leadsto C$ for some $C \in \mathcal{C}$. The main result of this section is as follows.

Theorem 1. *Let $\mathcal{C}$ be a finite union of upper-closed sets of configurations in M. Then, $Pre^*_M(\mathcal{C})$ can also be effectively represented as a finite union of upper-closed sets of configurations in M.*

The complex proof of Theorem 1 constructs an intermediate signaling system $\hat{M}$ whose $Pre^*_{\hat{M}}$ is easier to compute. The theorem can be established after we prove that Pre^*_M-computation can be realized by $Pre^*_{\hat{M}}$-computation and that $Pre^*_{\hat{M}}(\mathcal{C})$ can be effectively represented as a finite union of upper-closed sets.

Now, we can show that the configuration-reachability problem for non-cooperative signaling systems is decidable. This result implies that non-cooperative signaling systems are not universal (the set of reachable configurations is recursive). Notice that $\mathcal{C} = \{C_{\rm goal}\}$ is an upper-closed set. Since, from Theorem 1, $Pre^*_M(\mathcal{C})$ is effectively a finite union of upper-closed sets, one can also effectively answer the reachability at the beginning of this Section by checking whether $C_{\rm init}$ is an element in one of the upper-closed sets. Hence,

Theorem 2. *The configuration-reachability problem for non-cooperative signaling systems is decidable.*

Reachability considered so far is only one form of important verification queries. In the rest of this section, we will focus on more general queries that are specified in the computation tree logic (CTL) [6] interpreted on an infinite state transition system [5]. To proceed further, more definitions are needed.

Let M be a non-cooperative signaling system with symbols Σ and signals Sig. We use variables $\#(a), a \in \Sigma$, to indicate the number of a-objects in a configuration and use variable S over 2^{Sig} to indicate the signal set in the configuration. A *region formula F* (the word "region" is borrowed from [1]) is a Boolean combination of formulas in the following forms: $\#(a) > n,\ \#(a) = n,\ \#(a) < n,\ S = sig$, where $a \in \Sigma$, $n \in \mathbf{N}$, and $sig \subseteq Sig$. Region-CTL formulas f are defined using the following grammar: $f ::= \ F \mid f \wedge f \mid f \vee f \mid \neg f \mid \exists \circ f \mid \forall \circ f \mid f\ \exists\mathcal{U}\ f \mid f\ \forall\mathcal{U}\ f$, where F is a region formula. In particular, the eventuality operator $\exists \diamond\ f$ is the shorthand of $true\ \exists\mathcal{U}\ f$, and, its dual $\forall\Box\ f$ is simply $\neg \exists \diamond\ \neg f$. We use Region-CTL$^\diamond$ to denote a subset of the Region-CTL, where formulas are defined with: $f ::= \ F \mid f \wedge f \mid f \vee f \mid \neg f \mid \exists \circ f \mid \forall \circ f \mid \exists \diamond\ f \mid \forall\Box\ f$, where F is a region formula. Each f is interpreted as a set $[f]$ of configurations that satisfy f, as follows:

- $[F]$ is the set of configurations that satisfy the region formula F;
- $[f_1 \wedge f_2]$ is $[f_1] \cap [f_2]$; $[f_1 \vee f_2]$ is $[f_1] \cup [f_2]$; $[\neg f_1]$ is the complement of $[f_1]$;
- $[\exists \circ f_1]$ is the set of configurations C_1 such that, for some execution $C_1 \to_M C_2 \to_M \ldots$, we have $C_2 \in [f_1]$;
- $[\forall \circ f_1]$ is the set of configurations C_1 such that, for any execution $C_1 \to_M C_2 \to_M \ldots$, we have $C_2 \in [f_1]$;
- $[f_1\ \exists\mathcal{U}\ f_2]$ is the set of configurations C_1 such that, for some execution $C_1 \to_M C_2 \to_M \ldots$, we have $C_1, \ldots, C_n$ are all in $[f_1]$ and C_{n+1} is in $[f_2]$, for some n;

– $[f_1\ \forall\mathcal{U}\ f_2]$ is the set of configurations C_1 such that, for any execution $C_1 \to_M C_2 \to_M \ldots$, we have $C_1, \ldots, C_n$ are all in $[f_1]$ and C_{n+1} is in $[f_2]$, for some n.

Below, we use P to denote a Boolean combination of Presburger formulas over the $\#(a)$'s and formulas in the form of $S = sig$, where $sig \subseteq Sig$. The Region-CTL model-checking problem for non-cooperative signaling systems is to answer the following question:

Given: a non-cooperative signaling system M, a Presburger formula P, and a Region-CTL formula f,

Question: Does every configuration satisfying P also satisfy f?

It is known that the Region-CTL model-checking problem for non-cooperative P systems with rules $a \to b$ is undecidable [8]. From this result, one can show that the Region-CTL model-checking problem for non-cooperative signaling systems is undecidable as well.

Theorem 3. *The Region-CTL model-checking problem for non-cooperative signaling systems is undecidable.*

In contrast to Theorem 3, the subset, Region-CTL$^\diamond$, of Region-CTL is decidable for non-cooperative signaling systems:

Theorem 4. *The Region-CTL$^\diamond$ model-checking problem for non-cooperative signaling systems is decidable.*

Using Theorem 4, the following example property can be automatically verified for a non-cooperative signaling system M:

> "From every configuration satisfying $\#_a - \#_b < 6$, M has some execution that first reaches a configuration with $\#_a > 15$ and then reaches a halting configuration containing the signal s_1 and with $\#_b < 16$."

Notice that, above, "halting configurations" (i.e., none of the objects is enabled) form a finite union of upper-closed sets.

4 Presburger Reachability of Non-cooperative Signaling Systems

Let M be a non-cooperative signaling system and C_{init} be a given initial configuration. In this section, we are going to investigate a stronger form of reachability problems. As we have mentioned earlier, a multiset α (over alphabet Σ with k symbols) of objects can be represented as a vector in $\mathbf{N}^k$. Let $P(x_1, \ldots, x_k)$ be a Presburger formula over k nonnegative integer variables $x_1, \ldots, x_k$. The multiset α *satisfies* P if $P(\alpha)$ holds. A configuration (S, α) of the non-cooperative signaling system M *satisfies* P if α satisfies P. An *equality* is a Presburger formula in the form of $x_i = x_j$, for some $1 \le i, j \le k$. An *equality formula*, which is a special form of Presburger formulas, is a conjunction of a number of equalities. The *Presburger-reachability* problem is to decide whether a non-cooperative signaling system has a reachable configuration satisfying a given Presburger formula:

Given: a non-cooperative signaling system M, an initial configuration C_{init}, and a Presburger formula P,

Question: is there a reachable configuration satisfying P?

In contrast to Theorem 2, we can show that the Presburger-reachability problem is undecidable. The undecidability holds even when M has only one signal (i.e., $|Sig| = 1$) and P is an equality formula (i.e., the *equality-reachability problem*). In fact, what we will show is a more general result that characterizes the set of reachable configurations in M satisfying

P exactly as r.e. sets. Notice that, for P systems that deal with symbol objects, proofs for universality almost always use the theoretical tool through matrix grammar with appearance checking [17]. Here, we employ a new tool using Diophantine equations. Before we proceed further, we recall some known results on Diophantine equations (the Hilbert's Tenth Problem).

Let $m \in \mathbf{N}$, $Q \subseteq \mathbf{N}^m$ be a set of natural number tuples, and $E(z_1, \ldots, z_m, y_1, \ldots, y_n)$ be a Diophantine equation system. The set Q is *definable by* E if Q is exactly the solution set of $\exists y_1, \ldots, y_n . E(z_1, \ldots, z_m, y_1, \ldots, y_n)$; i.e., $Q = \{(z_1, \ldots, z_m) : E(z_1, \ldots, z_m, y_1, \ldots, y_n)$ holds for some $y_1, \ldots, y_n\}$. An *atomic* Diophantine equation is in one of the following three forms: $z = xy + \frac{1}{2}x(x+1), z = x + y, z = 1$, where x, y, z are three distinct variables over $\mathbf{N}$. A conjunction of these atomic equations is called a Diophantine equation system of atomic Diophantine equations. It is well known that Q is r.e. iff Q is definable by some Diophantine equation system [18]. From here, it is not hard to show the following:

Lemma 1. *For any set $Q \subseteq \mathbf{N}^m$, Q is r.e. iff Q is definable by a Diophantine equation system of atomic Diophantine equations.*

We now build a relationship between Diophantine equations and non-cooperative signaling systems. Recall that Q is a subset of $\mathbf{N}^m$. We say that Q is (M, C_{init}, P)-*definable* if there are designated symbols $Z_1, \ldots, Z_m$ in M such that, for any numbers $\#(Z_1), \ldots, \#(Z_m)$,

> $(\#(Z_1), \ldots, \#(Z_m))$ is in Q iff there is a reachable configuration from C_{init} in M satisfying P and, for each i, the number of Z_i-objects in the configuration is $\#(Z_i)$.

When P is $true$ and C_{init} is understood, we simply say that Q is definable by M. The non-cooperative signaling system M is *lazy* if, for any reachable configuration and any number n, if the configuration is reachable from C_{init} in n maximally parallel moves, then it is reachable in t maximally parallel moves for any $t \geq n$. We first show that solutions to each atomic Diophantine equation can be defined with a lazy non-cooperative signaling system M with only one signal.

Lemma 2. *The solution set to each atomic Diophantine equation is definable by some lazy non-cooperative signaling system M (starting from some C_{init}) with only one signal.*

Now, we are ready to show the following characterization.

Theorem 5. *For any set $Q \subseteq \mathbf{N}^m$, Q is r.e. iff Q is (M, C_{init}, P)-definable for some non-cooperative signaling system M with one signal, some configuration C_{init}, and some equality formula P.*

From Theorem 5, we immediately have

Theorem 6. *The equality-reachability problem for non-cooperative signaling systems with only one signal is undecidable. Therefore, the Presburger-reachability problem for non-cooperative signaling systems is undecidable as well.*

All the decidable/undecidable results presented so far can be generalized to the case when non-cooperative signaling systems are augmented with rules in the following forms: $s, a \rightarrow v, s'$, where v is a multiset. From now on, we let non-cooperative signaling systems contain these rules by default.

The results in Theorem 5 and Theorem 6 can be used to obtain a new result on non-cooperative P systems $\hat{M}$ where $\hat{M}$ has only one membrane and each rule is in the form of $a \rightarrow v$, where v is a multiset. Notice that $\hat{M}$ is very similar to a non-cooperative signaling system M with only one signal. Indeed, one can easily show that they are effectively equivalent in the following sense:

Lemma 3. *For any set $Q \subseteq \mathbf{N}^m$, Q is definable by some non-cooperative P system $\hat{M}$ iff Q is definable by some non-cooperative signaling system M with only one signal.*

It is known that $\hat{M}$ is not a universal P system model; multisets generated from $\hat{M}$ form the Parikh map of an ET0L language [16]. We now augment $\hat{M}$ with a *Presburger tester* that, nondeterministically at some maximally parallel move during a run of $\hat{M}$, tests (*for only once*) whether the current multiset satisfies a given Presburger formula P. When P is an equality formula, the tester is called an *equality tester*. If yes, the tester outputs the multiset and $\hat{M}$ shuts down. Otherwise, $\hat{M}$ crashes (with no output). Let $X_1, \ldots, X_m$ be designated symbols in $\hat{M}$. We say that $Q \subseteq \mathbf{N}^m$ is *output-definable* by $\hat{M}$ if Q is exactly the set of tuples $(\#(X_1), \ldots, \#(X_m))$ in the output multisets. Directly from Lemma 3 and Theorem 5, one can show that non-cooperative P systems (as well as non-cooperative signaling systems with only one signal) with an equality tester are universal:

Theorem 7. *For any set $Q \subseteq \mathbf{N}^m$, Q is r.e. iff Q is the output-definable by a non-cooperative P system (as well as a non-cooperative signaling system with only one signal) with an equality (and hence Presburger) tester.*

Hence,

Corollary 1. *The equality-reachability problem for non-cooperative P systems is undecidable. Therefore, the Presburger-reachability problem is undecidable as well.*

With the current technology, it might be difficult to implement the equality tester device to achieve the universality, which requires, e.g., external multiset evaluation during an almost instantaneous chemical reaction process. As we already know, a more natural way to perform the evaluation is to wait until the system *halts*; i.e., none of the objects in the current configuration is enabled. In this way, one can similarly formulate the halting-definability and the Presburger/equality-halting-reachability problems for non-cooperative signaling systems as well as for non-cooperative P systems, which concern halting and reachable configurations (instead of reachable configurations). We first show that non-cooperative signaling systems with only one signal has semilinear halting-definable reachability sets. This result essentially tells us that the number of signals matters, as far as halting configurations are considered: non-cooperative signaling systems with multiple signals are strictly stronger than non-cooperative signaling systems with only one signal (as well as non-cooperative P systems). This is because a non-semilinear set like $\{(n, 2^n) : n \geq 0\}$ can be easily halting-definable by a non-cooperative signaling system.

Theorem 8. *For any $Q \subseteq \mathbf{N}^m$, Q is a semilinear set iff Q is halting-definable by a non-cooperative signaling system with only one signal (as well as by a non-cooperative P system).*

One can similarly augment $\hat{M}$ as well as $\hat{M}$ with a Presburger tester but only test and output when a halting configuration is reached; i.e., a *Presburger halting tester*. The following result shows that non-cooperative signaling systems with only one signal and with a Presburger halting tester are not universal, while non-cooperative signaling systems with two signals and with an equality halting tester are universal. That is, again, the number of signals matters.

Theorem 9. *For any $Q \subseteq \mathbf{N}^m$, (1). Q is a semilinear set iff Q is output-definable by a non-cooperative signaling system with only one signal (as well as a non-cooperative P system) and with a Presburger halting tester. (2). Q is r.e. iff Q is output-definable by a non-cooperative signaling system with two signals and with an equality (and hence Presburger) halting tester.*

From Theorem 9, we immediately have:

Theorem 10. *(1). The halting Presburger reachability problem for non-cooperative signaling systems with two signals is undecidable. (2). The halting Presburger reachability problem for non-cooperative signaling systems with only one signal is decidable.*

5 Discussions and Future Work

In this section, we will discuss a number of possible extensions/modifications to the original definition of signaling systems, in order to better understand the computing power of the model.

In our set-up, a signal in a non-cooperative signaling system M has unbounded strength; i.e., it can trigger an unbounded number of instances of an enabled rule. If we restrict the strength of each signal in M to be B (where B is a constant), the resulting M is called a B-bounded non-cooperative signaling system. A move in such M is still maximally parallel. However, each signal can fire at most B instances of rules. From Theorem 2, we know that (unbounded) non-cooperative signaling systems are not universal. In contrast to this fact, we will show that bounded non-cooperative signaling systems are universal.

Consider a catalytic P system (with one membrane) $\hat{M}$ where the alphabet is the union of two disjoint sets: Γ (catalytic symbols) and Σ (object symbols). Each rule in $\hat{M}$ is in the form of $Ca \rightarrow Cv$, where $C \in \Gamma$ is a catalyst, and $a \in \Sigma$ is an object and v is a multiset of objects over Σ. The system starts with an initial multiset consisting of a number of catalysts and objects. Notice that, when $\hat{M}$ runs, the number of catalysts remains unchanged. It is known that catalytic P systems can generate any r.e. sets and hence are a universal model [22, 23, 10]. We now show a construction that simulates $\hat{M}$ with a bounded non-cooperative signaling system M. Therefore, bounded non-cooperative signaling systems are universal as well. Without loss of generality, we assume that the initial multiset of $\hat{M}$ contains exactly B copies of each catalyst C. Each rule $Ca \rightarrow Cv$ in $\hat{M}$ is now a rule $s_C, a \rightarrow v, s_C$ in M. Additionally, M has a new symbol d_C for each C and a rule $s_C, d_C \rightarrow d_C, s_C$. Initially, M starts with the same object multiset (without catalysts) in $\hat{M}$ and one instance of d_C-object for each C. It is left to the reader to verify that M, when the signals are with strength $B+1$, simulates $\hat{M}$: an object multiset is reachable in $\hat{M}$ iff the same object multiset augmented with one instance of d_C-object for each C is reachable in M. In fact, the result still holds when $B = 1$. To do this, one needs only to create B distinguished signals $s_C^1, \ldots, s_C^B$ and distinguished symbols $d_C^1, \ldots, d_C^B$, for each symbol C. Hence, 2-bounded non-cooperative signaling systems are universal.

Currently, we do not know whether 1-bounded non-cooperative signaling systems M are universal as well. We say that M is *single* if any object can only be triggered by at most one signal (i.e., whenever $s_1, a \rightarrow v_1, s_1'$ and $s_2, a \rightarrow v_2, s_2'$ are rules in M, then $s_1 = s_2$.). We can show that 1-bounded and single non-cooperative signaling systems M are not universal; they can be simulated by Petri nets (VASS) N. Each maximally parallel move in M is straightforwardly simulated by a transition in N; we omit the details in here.

There is an intimate relationship between some classes of P systems and VASS [14, 15]. Though non-cooperative signaling systems as well as VASS are not universal, they are incomparable in terms of the computing power. This is because, the Presburger-reachability problem of VASS is decidable [9] while, as we have shown, the same problem for non-cooperative signaling systems is undecidable. On the other hand, the Pre^*-image of a non-cooperative signaling system is always upper-closed while this is not true for VASS.

As we have mentioned earlier, in a one-membrane non-cooperative P system with promoters, a rule is in the form of $a \rightarrow v|p$ where p is a multiset called a promoter (a is not necessarily contained in p). The rule fires as usual in a maximally parallel manner but only when objects in the promoter all appear in the current configuration. It is not too difficult to construct such a P system to simulate a non-cooperative signaling system. Therefore, our signaling mechanism is not stronger than promoters. In fact, ours is strictly weaker. This is because one-membrane non-cooperative P systems with promoters are already universal [4]; however, as shown earlier, non-cooperative signaling systems are not universal. In contrast to these, signaling systems (which are not necessarily non-cooperative) are universal as well.

In the definition of a non-cooperative signaling system, a rule is in the form of $s, a \rightarrow v, s'$, where s and s' are signals. Now, we generalize the definition by allowing rules in the form of $S, a \rightarrow v, S'$ where S and S' are sets of signals (instead of signals). The maximally parallel semantics of the rules can be defined similarly. The differences are that the rule is enabled when every signal in S is in the current configuration and, after the rule is fired, every signal in S' is emitted. Hence, the rule now is triggered by exactly all of the signals in S. Such a rule is called a *multi-signal rule*. Let M be such a non-cooperative signaling system with multi-signal rules. Carefully looking at the proof of Theorem 1, we find that the proof can be adapted easily for such an M. Therefore, Theorem 2 and Theorem 4 still hold for non-cooperative signaling system with multi-signal rules. In fact, the results can be further generalized as follows.

Our study of non-cooperative signaling system was restricted to one membrane. We can generalize the model to work on multiple membranes (as in the P system), where each membrane has a set of rules, and in each rule $S, a \rightarrow v, S'$ (we are using multi-signal rules) we specify the "target" membranes where each object in v as well as each signal in S' are transported to. Notice that we do not use priority rules nor membrane dissolving rules. We call this generalized model as a multimembrane non-cooperative signaling system with multi-signal rules. Observe that multimembranes can be equivalently collapsed into one membrane through properly renaming (signal and object) symbols in a membrane. That is, each membrane is associated with a distinguished set of symbols. Of course, in doing so, the number of distinct symbols and signals in the reduced one-membrane system will increase as a function of the number of membranes in the original system. Therefore, Theorem 2 and Theorem 4 can be further generalized:

Theorem 11. *The configuration-reachability problem and the Region-CTL$^\diamond$ model-checking problem for multimembranes non-cooperative signaling systems with multi-signal rules are decidable.*

It is known that there are nonuniversal P systems where the number of membranes induces an infinite hierarchy in terms of computing power [13]. However, Theorem 11 says that the hierarchy collapses for non-cooperative signaling systems. Is there a hierarchy in terms of the number of membranes for a restricted and nonuniversal form of signaling systems (which is stronger than non-cooperative signaling systems)? We might also ask whether for one-membrane signaling systems, there is a hierarchy in terms of the numbers of symbols and signals used (since the conversion described above from multimembrane to one membrane increases the number of symbols and signals). As defined, a non-cooperative signaling system is a "generator" of multisets. For a given configuration C, there may be many configurations C' that satisfy $C \rightarrow_M C'$. Hence, a (maximally parallel) move is nondeterministic. Can we define an appropriate model of non-cooperative signaling system e.g., an "acceptor" of multisets (rather than a generator) such that the next move is unique, i.e., the system is deterministic? Deterministic P systems have been found to have some very nice properties

[11]. Finally, as in P systems, we would like to investigate the case when each move in the non-cooperative signaling system is "sequential", i.e., at each step, we nondeterministically choose a single rule to apply (instead of maximal parallelism). Sequential P systems have been found to be weaker than maximal-parallel P systems. We believe the situation is the same for non-cooperative signaling systems.

References

1. R. Alur and D. L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, April 1994.
2. Ioan I. Ardelean, Matteo Cavaliere, and Dragos Sburlan. Computing using signals: From cells to P Systems. In *Second Brainstorming Week on Membrane Computing, Sevilla, Spain, February 2-7 2004*, pages 60–73, Sevilla, Spain, February 2-7 2004.
3. W. M. Becker, L. J. Kleinsmith, and J. Hardin. *The World of the Cell (5th Edition)*. Benjamin Cummings (San Francisco), 2003.
4. P. Bottoni, C. Martin-Vide, Gh. Paun, and G. Rozenberg. Membrane systems with promoters/inhibitors. *Acta Informatica*, 38(10):695–720, 2002.
5. T. Bultan, R. Gerber, and W. Pugh. Model-checking concurrent systems with unbounded integer variables: symbolic representations, approximations, and experimental results. *ACM Trans. Program. Lang. Syst.*, 21(4):747–789, 1999.
6. E. M. Clarke, E. A. Emerson, and A. P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, April 1986.
7. E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking*. The MIT Press, 1999.
8. Z. Dang, O. H. Ibarra, C. Li, and G. Xie. On model-checking of P systems. 2005 (submitted).
9. Javier Esparza. Decidability and complexity of Petri net problems - an introduction. In *Petri Nets*, pages 374–428, 1996.
10. R. Freund, L. Kari, M. Oswald, and P. Sosik. Computationally universal P systems without priorities: two catalysts are sufficient. Available at *http://psystems.disco.unimib.it*, 2003.
11. R. Freund and Gh. Paun. On deterministic P systems. Available at http://psystems.disco.unimib.it. 2003.
12. R. Howell, D. Huynh, L. Rosier, and H. Yen. Some complexity bounds for problems concerning finite and 2-dimensional vasss. *Theoretical Computer Science*, 46:107–140, 1986.
13. O. H. Ibarra. The number of membranes matters. In *Proceedings of the 2003 Workshop on Membrane Computing (WMC 2003)*, Lecture Notes in Computer Science, 2003.
14. O. H. Ibarra, Z. Dang, and O. Egecioglu. Catalytic P systems, semilinear sets, and vector addition systems. *Theor. Comput. Sci.*, 312(2-3):379–399, 2004.
15. O. H. Ibarra, H. Yen, and Z. Dang. The power of maximal parallelism in P systems. In *DLT'04*, pages 212–224, 2004.
16. M. Ito, C. Martin-Vide, and Gh. Paun. A characterization of Parikh sets of ET0L languages in terms of P systems. In *Words, Semigroups, and Transducers*, pages 239–254. World Scientific.
17. C. Martin-Vide and Gh. Paun. Computing with membranes (P systems): universality results. In *MCU*, volume 2055 of *Lecture Notes in Computer Science*, pages 82–101. Springer, 2001.
18. Y. V. Matiyasevich. *Hilbert's Tenth Problem*. MIT Press, 1993.
19. Gh. Paun. Computing with membranes. *Journal of Computer and System Sciences*, 61(1):108–143, 2000.
20. Gh. Paun. *Membrane Computing: An Introduction*. Springer-Verlag, 2002.
21. Gh. Paun and G. Rozenberg. A guide to membrane computing. *TCS*, 287(1):73–100, 2002.
22. P. Sosik. P systems versus register machines: two universality proofs. In *Pre-Proceedings of Workshop on Membrane Computing (WMC-CdeA2002), Curtea de Arges, Romania*, pages 371–382, 2002.
23. P. Sosik and R. Freund. P systems without priorities are computationally universal. In *WMC-CdeA2002*, volume 2597 of *Lecture Notes in Computer Science*, pages 400–409. Springer, 2003.

Appendix: Proofs not presented in the paper

The following result will be needed later. For a vector $C = (x_1, ..., x_n) \in \mathbf{N}^n$, $\langle C \rangle$ denotes $\sum x_i$. Let $C' = (x'_1, ..., x'_n)$ be a vector $\in \mathbf{N}^n$. We say that $C \leq C'$ if each $x_i \leq x'_i$. Let $State$ be a finite set. We use $\alpha(m, n, L, C_0)$, where $m, n, L \in \mathbf{N}$ and $C_0 \in \mathbf{N}^n$, to denote a *state-vector* sequence $(state_0, C_0), (state_1, C_1), ..., (state_m, C_m)$ such that each $C_i \in \mathbf{N}^n$, $state_i \in State$, and $\langle C_i \rangle \leq \langle C_0 \rangle + i \cdot L$. The state-vector sequence is called *terminating* if there are some $i < j$ such that $state_i = state_j$ and $C_i \leq C_j$.

Lemma 4. *There exists a nonterminating state-vector sequence*

$$\alpha(m, n, L, C_0)$$

whose length m is not primitive recursive (in n, L, C_0).

Proof of Lemma 4

Proof. Define function f_n inductively as

$$\begin{cases} f_1(x) = 2x \\ f_n(x) = f_{n-1}^{(x)}(1), \text{ where } f_{n-1}^{(x)} \text{ is the } x\text{th-fold composition of } f_{n-1}. \end{cases}$$

Clearly $f_n(n)$ is not primitive recursive in n.

In what follows, we construct a finite n-dimensional VASS V = $\langle x_0, W, q_0, S, \delta \rangle$ such that given the initial vector $x_0 = (0, ..., 0, 0) \in N^n$, vector $x_m = (0, ..., 0, f_n(n))$ is reachable through a nonterminating sequence $(q_0, x_0) \to (q_1, x_1) \to \ldots \to (q_m, x_m)$ such that $\langle x_i \rangle \leq \langle x_0 \rangle + i$, $|S| = n(2n-1) + 2$, and $m \geq f_n(n)$. By letting $L = 1$ and $\langle C_0 \rangle = \langle x_0 \rangle = 0$, there exists a nonterminating sequence whose length m is not bounded by any primitive recursive function in n, L, C_0.

VASS V consists of n copies of module V_n, which is constructed inductively as follows.

1. (Basis) Module V_2:
   V_2 is shown in Fig. 1(a). The addition vectors of V_2 alter only the first two coordinates. It is clear that if the computation starts from $q_{2,1}$ with vector $(0, n, 0, ..., 0) \in N^n$, then it is possible to reach $q_{2,3}$ with vector $(0, 2n, \overbrace{0, ..., 0}^{n-2}) = (0, f_1(n), \overbrace{0, ..., 0}^{n-2})$.
2. (Induction step) We now show that for module V_i (see Fig. 1(b)), $2 < i \leq n$, V_i can produce $(0, ..., 0, f_{i-1}(n), \overbrace{0, ..., 0}^{n-i})$ in state $q_{i,3}$ when starting from $(0, ..., 0, n, \overbrace{0, ..., 0}^{n-i})$ in state $q_{i,1}$.

   Assume that V_{i-1} can produce $(0, ..., 0, f_{i-2}(n), \overbrace{0, ..., 0}^{n-i+1})$ when starting from

   $$(0, ..., 0, n, \overbrace{0, ..., 0}^{n-i+1}).$$

   With respect to V_i, the first time the loop on $q_{i,1} \to \boxed{V_{i-1}} \to q_{i-1}$ is executed, it produces $(0, ..., 0, f_{i-2}(1) - 1, n-1, \overbrace{0, ..., 0}^{n-i})$. By repeating this loop, the input to V_{i-1}

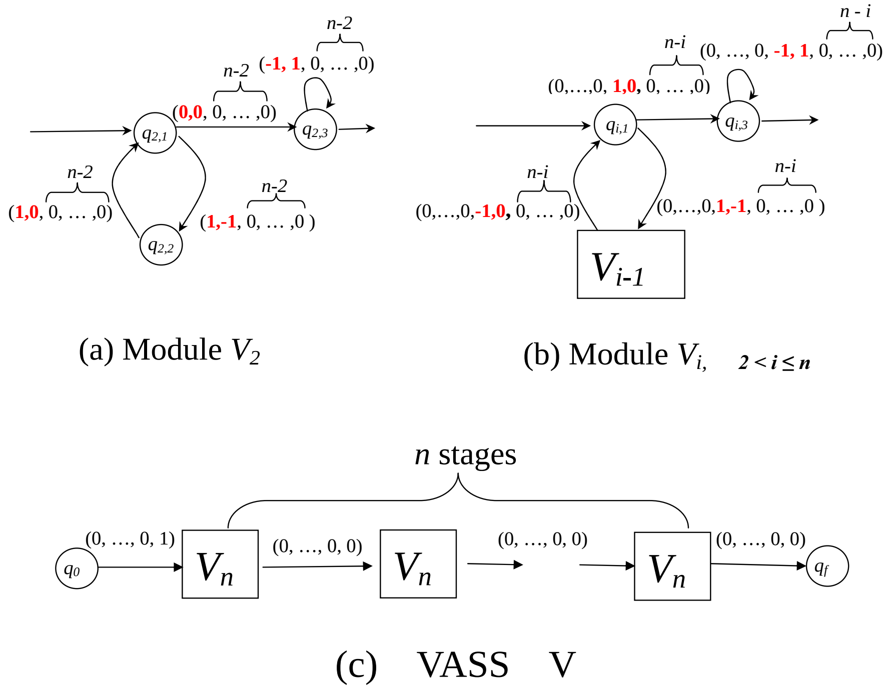

Fig. 1. A VASS witnessing a nonterminating sequence of non-primitive recursive length.

on the j-th iteration is $(0, ..., 0, f_{i-2}^{(j)}(1), n-j, \overbrace{0, ..., 0}^{n-i})$. Hence, when $q_{i,3}$ is reached for the first time after the n-th iterations, the vector becomes

$$(0, ..., 0, f_{i-2}^{(n)}(1), 0, \overbrace{0, ..., 0}^{n-i}) = (0, ..., 0, f_{i-1}(n), 0, \overbrace{0, ..., 0}^{n-i}).$$

Then by applying the self-loop in state $q_{i,3}$, $(0, ..., 0, f_{i-1}(n), \overbrace{0, ..., 0}^{n-i})$ can be produced.

Now VASS V consists of n copies of V_n in a way shown in Fig. 1(c). Starting from state q_0, the input to the first copy is $(0, ..., 0, 1)$, so this copy can produce $(0, ..., 0, f_{n-1}(1))$. The input to the j-th copy can thus be $(0, ..., 0, f_{n-1}^{(j-1)}(1))$ and the output can be $(0, ..., 0, f_{n-1}^{(j)}(1))$. Therefore, V can produce $(0, ..., 0, f_{n-1}^{(n)}(1))$ $=(0, ..., 0, f_n(n))$ when q_f is reached. Let σ be such a computation which is from $(q_0, (0, ..., 0))$ to $(q_f, (0, ..., 0, f_n(n)))$.

First notice that the computation σ is of length greater than or equal to $f_n(n)$, since each addition vector in V has at most one "+1" in its coordinates. As for the size of V, it is clear that V_2 has three states, and each V_i has two more states than V_{i-1}. Hence, V_n has $2n-1$ states. V therefore has $n(2n-1)+2$ states. Finally, the computation σ is nonterminating since in each V_i, each loop causes one position to decrease in each iteration. (In fact, the reachability set of VASS V is finite.) This completes the proof. ∎

A proof similar to the above was used in [12] for bounding the sizes of finite VASSs.

Let M be a non-cooperative signaling system. To prove Theorem 1, we need the following lemmas.

Lemma 5. *Let $\mathcal{C} = [S, \beta, \sigma^*]$ be an upper-closed set of configurations. Then, $Pre_M(\mathcal{C})$ can be effectively represented as a finite union of upper-closed sets $\mathcal{C}'$ of configurations such that each $\mathcal{C}'$ is $(|\beta| + |R|)$-bounded, where $|R|$ is the number of rules in the non-cooperative signaling system M.*

Proof of Lemma 5

Proof. Let $C' = (S', \alpha')$ be a configuration in $Pre_M(\mathcal{C})$; i.e., there is a configuration $C \in \mathcal{C}$ such that $C' \to_M C$. Since $\mathcal{C} = [S, \beta, \sigma^*]$, we can represent C as (S, α) where, for some multiset γ in σ^*, the multiset α is the multiset union of β and γ. Let $\mathtt{m}$ be the specific maximally parallel move witnessing $C' \to_M C$, where we use $R^{\mathtt{m}}$, a subset of rules in M, to denote the rules that are actually fired in the move. For the specific move $C' \to_M C$, we would like to know, for each individual object o' in C', the objects in C that o' evolves into. There are a number of disjoint cases to consider:

- C1 A split-rule $split_{o'}$ is fired on the object o' and, as a result, the object o' is split into two objects in C and at least one of the two objects is in β.
- C2 A split-rule $split_{o'}$ is fired on the object o' and, as a result, the object o' is split into two objects in C and both are in γ.
- C3 A die-rule $die_{o'}$ is fired on the object o' and, as a result, the object o' disappears in C.
- C4 The object o' is not enabled under the move and thus, in C, remains as an object in β.
- C5 The object o' is not enabled under the move and thus, in C, remains as an object in γ.

We use α'_{C1} (resp. α'_{C2}, α'_{C3}, α'_{C4}, α'_{C5}) to denote the multiset of all the objects in C' that belong to the category C1 (resp. C2, C3, C4, C5). Clearly, each object o' in C' belongs to exactly one of the above five categories and hence α' is the multiset union of the five multisets: α'_{C1}, α'_{C2}, α'_{C3}, α'_{C4}, and α'_{C5}. Observe that

$$|\alpha'_{\mathrm{C1}}| + |\alpha'_{\mathrm{C4}}| \leq |\beta|. \tag{1}$$

Below, we devise a procedure to extract the *essential* objects o' from each of the five multisets.

Each object o' in category C1 is essential; we use multiset

$$\beta'_{\mathrm{C1}} = \alpha'_{\mathrm{C1}} \tag{2}$$

to denote them. Recall that the split-rule $split_{o'}$ is fired on such an object o' in β'_{C1} and we use $R^{\mathtt{m}}_{\mathrm{C1}}$ to denote the set of all such split-rules; i.e., $R^{\mathtt{m}}_{\mathrm{C1}} = \{split_{o'} : o' \in \beta'_{\mathrm{C1}}\}$. Clearly, $R^{\mathtt{m}}_{\mathrm{C1}} \subseteq R^{\mathtt{m}}$.

For each split-rule $split$ that is in $R^{\mathtt{m}}$ (hence is fired in the move) but not in $R^{\mathtt{m}}_{\mathrm{C1}}$, we choose one object o', which is identified to be essential, in C' that the rule $split$ is fired upon. Notice that the object o' must belong to category C2. We put all such essential objects o' in a multiset, denoted by β'_{C2}. Clearly, the number of objects in the multiset is bounded by the number $|R_{\mathrm{split}}|$ of split-rules in the non-cooperative signaling system

$$|\beta'_{\mathrm{C2}}| \leq |R_{\mathrm{split}}|. \tag{3}$$

Similarly, for each die-rule die in $R^{\mathtt{m}}$, we choose one object o', which is identified to be essential, in C' that the rule die is fired upon. Notice that the object o' must belong to category C3. We put all such essential objects o' in a multiset, denoted by β'_{C3}. Clearly, the number of objects in the multiset is bounded by the number $|R_{\mathrm{die}}|$ of die-rules in the non-cooperative signaling system

$$|\beta'_{\mathrm{C3}}| \le |R_{\mathrm{die}}|. \tag{4}$$

Each object o' in category C4 is essential; we use multiset

$$\beta'_{\mathrm{C4}} = \alpha'_{\mathrm{C4}} \tag{5}$$

to denote them.

Up to now, we define the *essential object multiset* β' to be the multiset of all the essential objects in C'; i.e., β' is the multiset union of β'_{C1}, β'_{C2}, β'_{C3} and β'_{C4}. Using (1), (2), (3), (4), (5) and the fact that $R = R_{\mathrm{split}} \cup R_{\mathrm{die}}$, we have

$$|\beta'| \le |\beta| + |R|. \tag{6}$$

Finally, we define the *unessential symbol set*

$$\sigma' \subseteq \Sigma \tag{7}$$

to be the union of the following symbol-sets:

– $\lfloor \beta'_{\mathrm{C2}} \rfloor$, the symbol-set of multiset β'_{C2} (which is also the symbol-set of multiset α'_{C2}); that is, $\lfloor \beta'_{\mathrm{C2}} \rfloor$ is the set of symbols that are actually appear in multiset β'_{C2},
– $\lfloor \beta'_{\mathrm{C3}} \rfloor$, the symbol-set of multiset β'_{C3} (which is also the symbol-set of multiset α'_{C3}),
– $\lfloor \alpha'_{\mathrm{C5}} \rfloor$, the symbol-set of multiset α'_{C5}.

Now, it is the time to explain why the objects in β' are essential. An object o' in C' is *unessential* if it is in α' but not in β'. Clearly, if one drops one or more unessential objects from C', then the new C' might not reach C. However, from the above constructions, each of these unessential objects can only belong to C2, C3, or C5, and will either disappear or evolve into objects in $\gamma \in \sigma^*$. The new C' can still reach some configuration (which is not necessarily the same as C) in $\mathcal{C} = [S, \beta, \sigma^*]$ through a move in which the fired rules are exactly $R_{\mathtt{m}}$; i.e., the new C' is in $Pre_M(\mathcal{C})$. The same reasoning applies when a new configuration C' is obtained by adding objects o' whose types belong to the unessential symbol set σ' (these newly added objects are still unessential). In summary, from the specific maximally parallel move $\mathtt{m}$ witnessing $C' \to_M C \in \mathcal{C}$, we can construct an upper-closed set

$$\mathcal{C}'_{C',\mathtt{m},C} = [S', \beta', \sigma'^*]. \tag{8}$$

Clearly,

$$C' \in \mathcal{C}'_{C',\mathtt{m},C}. \tag{9}$$

The new C''s satisfying $C' \in Pre_M(\mathcal{C})$ mentioned earlier are exactly those configurations in the upper-closed set. Therefore,

$$\mathcal{C}'_{C',\mathtt{m},C} \subseteq Pre_M(\mathcal{C}). \tag{10}$$

Notice that $\mathcal{C} = [S, \beta, \sigma^*]$ is given. Therefore, among all the possible choices of $C', \mathtt{m}, C$ that makes $C' \to_M C \in \mathcal{C}$, there are only a finite number of distinct upper-closed sets $\mathcal{C}'_{C',\mathtt{m},C}$ because of (8), (6) and (7). We put all these distinct upper-closed sets into a finite class $\mathbf{C}$. From (9) and (10), we conclude that $Pre_M(\mathcal{C})$ is exactly the union of the finitely many upper-closed sets in $\mathbf{C}$. The lemma follows since, due to (6), each upper-closed set in $\mathbf{C}$ is $(|\beta| + |R|)$-bounded.

In fact, a close look at the above proof shows that the class $\mathbf{C}$ can be computed effectively. To do this, one first enumerates all the $(|\beta|+|R|)$-bounded upper-closed sets $\mathcal{C}' = [S', \beta', \sigma']$ for all S', β' and σ'. (There are only a finite number of them.) Each $\mathcal{C}'$ is put into the class $\mathbf{C}$ whenever the following statement is true for configuration $C' = (S', \beta')$:

> for some $C \in \mathcal{C}$, $C' \rightarrow_M C$ in a maximally parallel move on which the essential object multiset is exactly β' and the essential symbol set is exactly σ'.

The truth can be obviously verified effectively since there are only finitely many $C \in \mathcal{C}$ that the $C' = (S', \beta')$ can possibly move into. This completes the proof. ∎

Clearly, Lemma 5 can be generalized to the case when $\mathcal{C}$ is a finite union of upper-closed sets.

Lemma 6. *Let $\mathcal{C}$ be, for some $m > 0$, a finite union of m-bounded upper-closed sets of configurations. Then, $Pre_M(\mathcal{C})$ can be effectively represented as a finite union of $(m + |R|)$-bounded upper-closed sets of configurations, where $|R|$ is the number of rules in the non-cooperative signaling system M.*

Proof of Lemma 6

Proof. Suppose that $\mathcal{C}$ is the union of $\mathcal{C}_1, \ldots, \mathcal{C}_n$, for some $n > 0$, each of which is an m-bounded upper-closed set of configurations. Clearly, $Pre_M(\mathcal{C})$ equals the union of

$$Pre_M(\mathcal{C}_1), \ldots, Pre_M(\mathcal{C}_n).$$

The result follows after using Lemma 5 on each $Pre_M(\mathcal{C}_i)$. ∎

Let $\mathcal{C}$ be a finite union of upper-closed sets of configurations. We use $Pre^*_M(\mathcal{C})$ to denote the set of all configurations C' such that $C' \rightsquigarrow C$ for some $C \in \mathcal{C}$. We are going to show that $Pre^*_M(\mathcal{C})$ is a finite union of upper-closed sets of configurations as well. To do this, we need an intermediate result.

We translate M into a different non-cooperative signaling system $\hat{M}$, as follows.

- (Alphabet in $\hat{M}$) The alphabet of $\hat{M}$ is $\dot{\Sigma} \cup \bar{\Sigma}$, where Σ is the alphabet of M and, $\dot{\Sigma} = \{\dot{a} : a \in \Sigma\}$ and $\bar{\Sigma} = \{\bar{a} : a \in \Sigma\}$ are two new and disjoint alphabets. Each symbol $\dot{a} \in \dot{\Sigma}$ is called a *solid symbol* while each symbol $\bar{a} \in \bar{\Sigma}$ is called a *star symbol* (its meaning will be explained in a moment). Correspondingly, for each $a \in \Sigma$, an $\dot{a}$-object is a *solid object* while an $\bar{a}$-object is a *star object*.
- (Signals in $\hat{M}$) The signal set of $\hat{M}$ is exactly the same as the signal set Sig of M.
- (Rules in $\hat{M}$)
  - For each split-rule $s, a \rightarrow bc, s'$ in M, we add the following split-rules in $\hat{M}$:
    $s, \dot{a} \rightarrow \dot{b}\dot{c}, s'$,
    $s, \dot{a} \rightarrow \bar{b}\dot{c}, s'$,
    $s, \dot{a} \rightarrow \dot{b}\bar{c}, s'$,
    $s, \bar{a} \rightarrow \bar{b}\bar{c}, s'$.
    That is, in $\hat{M}$, a solid object can split similarly into two objects as in M, but at least one of them must be a solid object. Additionally, a star object can only be split into two star objects.

- For each die-rule $s, a \to \Lambda$ in M, we add the following die-rule in $\hat{M}$:
  $s, \bar{a} \to \Lambda$.
  That is, in $\hat{M}$, a star object can die similarly as in M. But a solid object can never die.

The non-cooperative signaling system $\hat{M}$ essentially maintains the reachability relation of M in the following sense. Consider a configuration $\hat{C}$ in $\hat{M}$. For every a, we now replace each $\dot{a}$-object as well as each $\bar{a}$-object in $\hat{C}$ with an a-object. The result, denoted by $[\hat{C}]$, is clearly a configuration in M. The following property can be shown directly from the construction of $\hat{M}$:

> For any configuration C in M and configuration $\hat{C}$ in $\hat{M}$ with $C = [\hat{C}]$, the following two statements are true:
> (I). For any configuration C' in M there is a configuration $\hat{C}'$ in $\hat{M}$ with $C' = [\hat{C}']$ such that $C' \to_M C$ implies $\hat{C}' \to_{\hat{M}} \hat{C}$.
> (II). For any configuration $\hat{C}'$ in $\hat{M}$ there is a configuration C in M with $C' = [\hat{C}']$ such that $\hat{C}' \to_{\hat{M}} \hat{C}$ implies $C' \to_M C$.

For a set $\hat{\mathcal{C}}$ of configurations in $\hat{M}$, we use $[\hat{\mathcal{C}}]$ to denote the set $\{[\hat{C}] : \hat{C} \in \hat{\mathcal{C}}\}$ of configurations in M. The first intermediate result shows that Pre^*_M-computation can be realized by $Pre^*_{\hat{M}}$-computation.

Lemma 7. *Let $\mathcal{C}$ be a set of configurations in M and $\hat{\mathcal{C}}$ be a set of configurations in $\hat{M}$, satisfying $\mathcal{C} = [\hat{\mathcal{C}}]$. Then, $Pre^*_M(\mathcal{C}) = [Pre^*_{\hat{M}}(\hat{\mathcal{C}})]$. In particular, if $Pre^*_{\hat{M}}(\hat{\mathcal{C}})$ is a finite union of upper-closed sets, then so is $Pre^*_M(\mathcal{C})$.*

Proof of Lemma 7

Proof. Let C' be a configuration in $Pre^*_M(\mathcal{C})$; i.e., there is a $C \in \mathcal{C}$ such that $C' \leadsto_M C \in \mathcal{C}$. Suppose that the reachability is witnessed by the following execution

$$C' = C_0 \to_M C_1 \to_M \ldots \to_M C_n = C \tag{11}$$

for some n and some $C_1, \ldots, C_{n-1}$. From the condition of $\mathcal{C} = [\hat{\mathcal{C}}]$, there is a configuration, denoted by $\hat{C}$, in $\hat{\mathcal{C}}$ such that $C = [\hat{C}]$. Using the statement (I) presented earlier, we can construct configurations $\hat{C}', \hat{C}_n, \ldots, \hat{C}_0, \hat{C}$ in $\hat{M}$ such that

$$\hat{C}' = \hat{C}_0 \to_{\hat{M}} \hat{C}_1 \to_{\hat{M}} \ldots \to_{\hat{M}} \hat{C}_n = \hat{C} \tag{12}$$

with $C_0 = [\hat{C}_0], \ldots, C_n = [\hat{C}_n]$. Since $\hat{C}_n = \hat{C} \in \hat{\mathcal{C}}$, we have $\hat{C}' = \hat{C}_0 \in Pre^*_{\hat{M}}(\hat{\mathcal{C}})$. Therefore, $C' = C_0 \in [Pre^*_{\hat{M}}(\hat{\mathcal{C}})]$. Hence, $Pre^*_M(\mathcal{C}) \subseteq [Pre^*_{\hat{M}}(\hat{\mathcal{C}})]$.

On the other hand, Let $\hat{C}'$ be a configuration in $Pre^*_{\hat{M}}(\hat{\mathcal{C}})$. That is, there is a $\hat{C} \in \hat{\mathcal{C}}$ such that $\hat{C}' \leadsto_{\hat{M}} \hat{C}$; i.e., (12) holds. Using the statement (II) presented earlier, we can construct configurations $C', C_n, \ldots, C_0, C$ in M such that (11) holds and $[\hat{C}'] = C'$ and $[\hat{C}] = C$. Since $\hat{C} \in \hat{\mathcal{C}}$, we have $C = [\hat{C}] \in [\hat{\mathcal{C}}] = \hat{C}$. So, $[\hat{C}'] = C' \in Pre^*_M(\mathcal{C})$. Therefore, $[Pre^*_{\hat{M}}(\hat{\mathcal{C}})] \subseteq Pre^*_M(\mathcal{C})$.

Hence, $Pre^*_M(\mathcal{C}) = [Pre^*_{\hat{M}}(\hat{\mathcal{C}})]$. The second part of the theorem follows trivially. ∎

The second intermediate result concerns $Pre^*_{\hat{M}}$-computation. To be precise, let $\hat{\mathcal{C}}_0$ be an upper-closed set of configurations in $\hat{M}$. $\hat{\mathcal{C}}_0$ is *separated* if $\hat{\mathcal{C}}_0 = [S, \dot{\beta}, \bar{\sigma}^*]$ for some S, $\dot{\beta} \in \dot{\Sigma}^*$ and $\bar{\sigma} \subseteq \bar{\Sigma}^*$. When this is the case, the result states that $Pre^*_{\hat{M}}(\hat{\mathcal{C}}_0)$ is a finite union of upper-closed sets.

Lemma 8. *Let $\hat{\mathcal{C}}_0$ be a separated upper-closed set of configurations in $\hat{M}$. Then, $Pre^*_{\hat{M}}(\hat{\mathcal{C}}_0)$ is effectively a finite union of upper-closed sets of configurations in $\hat{M}$.*

Proof of Lemma 8

Proof. Let $\hat{\mathcal{C}}_0 = [S_{\rm goal}, \dot{\beta}_{\rm goal}, \bar{\sigma}^*_{\rm goal}]$ be a separated upper-closed set, where $\dot{\beta}_{\rm goal} \in \dot{\Sigma}^*$ and $\bar{\sigma}_{\rm goal} \subseteq \bar{\Sigma}^*$. We use $\hat{\mathcal{C}}_1, \hat{\mathcal{C}}_2, \ldots$ to denote sets of configurations in $\hat{M}$, which are defined as follows: for each $t \geq 0$,

$$\hat{\mathcal{C}}_{t+1} = \hat{\mathcal{C}}_t \cup Pre_{\hat{M}}(\hat{\mathcal{C}}_t). \tag{13}$$

Notice that $Pre^*_{\hat{M}}(\hat{\mathcal{C}}_0)$ is the union of all $\hat{\mathcal{C}}_t$, $t \geq 0$. From Lemma 6, each $\hat{\mathcal{C}}_t$ is also a finite union of upper-closed sets.

From the definition of Pre, the set $\hat{\mathcal{C}}_t$ defines all such configurations in $\hat{M}$ that can reach some configuration in $\hat{\mathcal{C}}_0$ in at most t maximally parallel moves. Notice that the total number of solid objects in each configuration in $\hat{\mathcal{C}}_t$ can not exceed the number $|\dot{\beta}_{\rm goal}|$ of solid objects in all configurations in $\hat{\mathcal{C}}_0 = [S_{\rm goal}, \dot{\beta}_{\rm goal}, \bar{\sigma}^*_{\rm goal}]$. This is because, in $\hat{M}$, a solid object must evolve into at least one solid object. Therefore, for each upper-closed set, say, $\hat{\mathcal{C}} = [S, \hat{\beta}, \hat{\sigma}^*]$, in $\hat{\mathcal{C}}_t$, we have the following:

- We put all the solid objects in $\hat{\beta}$ into a multiset $\dot{\theta}$ while put the remaining (star) objects into a multiset $\bar{\mu}$ (i.e., $\hat{\beta}$ is the multiset union of $\dot{\theta}$ and $\bar{\mu}$). Then, the size of $\dot{\theta}$ is also bounded by $|\dot{\beta}_{\rm goal}|$.
- The part $\hat{\sigma}^*$ in $\hat{\mathcal{C}}$ does not contain any solid objects; i.e., $\hat{\sigma} \subseteq \bar{\Sigma}$. (Otherwise, due to the star, configurations in $\mathcal{C}$ will have unbounded number of solid objects.) To emphasize this fact, we use $\bar{\sigma}^*$ to denote the $\hat{\sigma}^*$ in $\hat{\mathcal{C}}$.
- Furthermore, we can safely assume that $\bar{\mu} \in \bar{\sigma}^*$. That is, for each $\bar{a}$, the existence of an $\bar{a}$-object in $\bar{\mu}$ implies $\bar{a} \in \bar{\sigma}$. To see this, let $\hat{C}$ be a configuration in $\hat{\mathcal{C}}$, where o is some $\bar{a}$-object. By definition, in $\hat{M}$, configuration $\hat{C}$ can reach some configuration $\hat{C}_0$ in $\hat{\mathcal{C}}_0 = [S_{\rm goal}, \dot{\beta}_{\rm goal}, \bar{\sigma}^*_{\rm goal}]$, within at most t maximally parallel moves. According to the construction of $\hat{M}$, the star object o in $\hat{C}$ can only evolve into 0 or more star objects when $\hat{C}_0$ is reached. In particular, since $\hat{C}_0 \in \hat{\mathcal{C}}_0$ and the latter is a separated upper-closed set, we conclude that the evolved star objects must all in $\bar{\sigma}^*_{\rm goal}$. Hence, $\hat{C}'$, the result of adding 0 or more additional copies of o (an $\bar{a}$-object) into $\hat{C}$, can still reach some configuration in $\hat{\mathcal{C}}_0$, in at most t maximally parallel moves. Therefore, the symbol $\bar{a}$ can be safely added to the $\bar{\sigma}$.

Considering the above three facts, we will use $\hat{\mathcal{C}} = [(S, \dot{\theta}), \bar{\mu}, \bar{\sigma}^*]$ with $\bar{\mu} \in \bar{\sigma}^*$ to symbolically represent an upper-closed set $\hat{\mathcal{C}}$ in $\hat{\mathcal{C}}_t$. In particular, we call the pair $(S, \dot{\theta})$ as the *tag* of the set, noticing that there are only a bounded number of distinct tags (since $S \subseteq Sig$ and the size of $\dot{\theta}$ is bounded by the given $|\dot{\beta}_{\rm goal}|$). We use a finite set $\mathcal{T}$ to denote all the distinct tags. Hence, each upper-closed set $\hat{\mathcal{C}}$ in $\hat{\mathcal{C}}_t$ can be written in the form of

$$\hat{\mathcal{C}} = [tag, \bar{\mu}, \bar{\sigma}^*] \tag{14}$$

with $\bar{\mu} \in \bar{\sigma}^* \subseteq \bar{\Sigma}^*$ and $tag \in \mathcal{T}$. Finally, we are ready to show the result. Firstly, we claim that

CLAIM. There is a t_{fix} such that $\hat{\mathcal{C}}_{t_{\text{fix}}+1} = \hat{\mathcal{C}}_{t_{\text{fix}}}$.

Clearly, if the CLAIM holds, the result is immediate, since one can effectively calculate the fixed point $Pre^*_{\hat{M}}(\hat{\mathcal{C}}_0)$ as follows:

$t = -1$;
`repeat`
 $t = t + 1$;
 $\hat{\mathcal{C}}_{t+1} = \hat{\mathcal{C}}_t \cup Pre_{\hat{M}}(\hat{\mathcal{C}}_t)$;
`until` ($\hat{\mathcal{C}}_{t+1}$ equals $\hat{\mathcal{C}}_t$)
Let $t_{\text{fix}} = t$;
`return` $\hat{\mathcal{C}}_{t_{\text{fix}}}$ as $Pre^*_{\hat{M}}(\hat{\mathcal{C}}_0)$.

To conclude the proof, we turn now to showing the CLAIM. Suppose that the CLAIM does not hold. That is, one can find an infinite number of configurations

$$\hat{C}_1, \ldots, \hat{C}_t, \ldots$$

in $\hat{M}$ such that each $\hat{C}_t$ ($t \geq 1$) witnesses the fact that $\hat{\mathcal{C}}_{t-1} \neq \hat{\mathcal{C}}_t$; i.e., noticing that $\hat{\mathcal{C}}_{t-1} \subseteq \hat{\mathcal{C}}_t$ by (13),

$$\hat{C}_t \in \hat{\mathcal{C}}_t \text{ but } \hat{C}_t \notin \hat{\mathcal{C}}_{t-1}. \tag{15}$$

Since each $\hat{\mathcal{C}}_t$ is a finite union of upper-closed sets, we use, according to (14), $[tag_t, \bar{\mu}_t, \bar{\sigma}_t^*]$ to denote one of the upper-closed sets that contains $\hat{C}_t$; i.e.,

$$\hat{C}_t \in [tag_t, \bar{\mu}_t, \bar{\sigma}_t^*] \subseteq \hat{\mathcal{C}}_t, \tag{16}$$

with

$$\bar{\mu}_t \in \bar{\sigma}_t^*. \tag{17}$$

Notice that $tag_t \in \mathcal{T}$ and $\bar{\sigma}_t \subseteq \bar{\Sigma}$, where bot $\mathcal{T}$ and $\bar{\Sigma}$ are finite sets. Therefore, there must be an infinite subsequence $0 < t_1 < \ldots < t_n < \ldots$ such that, for some $tag \in \mathcal{T}$ and some $\bar{\sigma} \subseteq \bar{\Sigma}$, we have,

$tag = tag_{t_1} = tag_{t_2} = \ldots$, and, $\bar{\sigma} = \bar{\sigma}_{t_1} = \bar{\sigma}_{t_2} = \ldots$.

Suppose that the tag is in the form of $(S, \dot{\theta})$ for some S and $\dot{\theta}$. Hence, using (16), each configuration $\hat{C}_{t_i}$, $i \geq 1$, can be written as a pair of the signal set S and an object multiset $\hat{\gamma}_i = \dot{\theta} \cup \bar{\mu}_{t_i} \cup \bar{\delta}_i$, for some $\bar{\delta}_i \in \bar{\sigma}^*$. According to (17), $\bar{\mu}_{t_i}$ is also a multiset in $\bar{\sigma}^*$. Therefore, we can use $\bar{\eta}_i \in \bar{\sigma}^*$ to denote $\bar{\mu}_{t_i} \cup \bar{\delta}_i$ and then write $\hat{\gamma}_i = \dot{\theta} \cup \bar{\eta}_i$. Applying Dickson's lemma on the infinite sequence of multisets (over alphabet $\bar{\sigma}$),

$$\bar{\eta}_1, \ldots, \bar{\eta}_i, \ldots,$$

we conclude that, there are numbers $p < q$ such that $\bar{\eta}_p$ is contained in $\bar{\eta}_q$. From the definition of $\bar{\eta}_p$, we have $\bar{\mu}_{t_p}$ is also contained in $\bar{\eta}_q$. Therefore, $\bar{\eta}_q$ can be represented as the union of multiset $\bar{\mu}_{t_p}$ and some multiset in $\bar{\sigma}^*$. Hence, $\hat{C}_{t_q}$ is an element in $[tag, \bar{\mu}_{t_p}, \bar{\sigma}^*]$, which is equal to $\hat{\mathcal{C}}_{t_p} = [tag_{t_p}, \bar{\mu}_{t_p}, \bar{\sigma}^*_{t_p}]$. That is, $\hat{C}_{t_q} \in \hat{\mathcal{C}}_{t_p}$. Since $p < q$ and $t_p < t_q$, we have $\hat{\mathcal{C}}_{t_p} \subseteq \hat{\mathcal{C}}_{t_q-1}$. Therefore, $\hat{C}_{t_q} \in \hat{\mathcal{C}}_{t_q-1}$, which contradicts (15). Hence, the CLAIM holds. ∎

Unfortunately, an accurate complexity bound is difficult to obtain for Lemma 8. The key to the complexity lies in how large is the number t_{fix} of iterations before the fixed point is reached in the CLAIM in the above proof. Closely looking at the proof reveals that the t_{fix} makes every sequence

$$[tag_1, \bar{\mu}_1, \bar{\sigma}_1^*], \ldots, [tag_{t_{\mathrm{fix}}}, \bar{\mu}_{t_{\mathrm{fix}}}, \bar{\sigma}_{t_{\mathrm{fix}}}^*]$$

terminating in the following sense, where each $[tag_t, \bar{\mu}_t, \bar{\sigma}_t^*]$ is an upper-closed set in $\hat{\mathcal{C}}_t$. That is, there are $i < j$ such that $tag_i = tag_j$, $\bar{\sigma}_i = \bar{\sigma}_j$, and $\bar{\mu}_i \subseteq \bar{\mu}_j$. We rewrite the sequence into

$$(state_1, \bar{\mu}_1), \ldots, (state_{t_{\mathrm{fix}}}, \bar{\mu}_{t_{\mathrm{fix}}}) \tag{18}$$

where each $state_t = (tag_t, \bar{\sigma}_t)$, which is drawn from the finite space $\mathcal{T} \times \bar{\Sigma}$. Using (13) and Lemma 6, one can easily conclude that the size of each $\bar{\mu}_t$ is bounded by the size of multiset $\dot{\beta}_{\mathrm{goal}}$ plus $t \cdot L$, where L is the number of rules in $\hat{M}$ (recall that, in Lemma 8, $\hat{\mathcal{C}}_0$ is given as $[S_{\mathrm{goal}}, \dot{\beta}_{\mathrm{goal}}, \bar{\sigma}^*_{\mathrm{goal}}]$). Notice that, multisets $\bar{\mu}_t$ can also be treated as vectors. In this way, the state-vector sequence in (18) is also a terminating sequence in the sense of Lemma 4. From the same lemma, we conclude that the number t_{fix} to compute $Pre^*_{\hat{M}}(\hat{\mathcal{C}}_0)$, as an upper bound for the worst-case, is not primitive recursive on the representation $\hat{\mathcal{C}}_0$ and $\hat{M}$. This estimation also applies to all the decidable results that rely on Lemma 8 and will be obtained in the rest of this section.

Given the above two intermediate results (Lemma 7 and Lemma 8), we are now ready to complete the proof of Theorem 1.

Proof of Theorem 1

Proof. We only prove the result for $\mathcal{C}$ being an upper-closed set of configurations in M. The proof can be easily generalized to the case when $\mathcal{C}$ is a finite union of upper-closed sets of configurations, similar to the proof of Lemma 6.

Suppose that the upper-closed set $\mathcal{C}$ of configurations in M is in the form of $[S, \beta, \sigma^*]$. We use $\dot{\beta}$ to denote the result of, for every a, replacing each a-object in β with an $\dot{a}$-object. Similarly, we use $\bar{\sigma}$ to denote the result of replacing every symbol a in σ with symbol $\bar{a}$. Now, we obtain $\hat{\mathcal{C}} = [S, \dot{\beta}, \bar{\sigma}^*]$, which is clearly a separated upper-closed set of configurations in $\hat{M}$. Notice that $[\hat{\mathcal{C}}] = \mathcal{C}$. Hence, from Lemma 7, we have $Pre^*_M(\mathcal{C}) = [Pre^*_{\hat{M}}(\hat{\mathcal{C}})]$. Additionally, from Lemma 8, $Pre^*_{\hat{M}}(\hat{\mathcal{C}})$ can effectively represented as a finite union of upper-closed set of configurations in $\hat{M}$. From the second part of Lemma 7, the result follows. ∎

Proof of Theorem 4

Proof. Let M be a non-cooperative signaling system and P is a Presburger formula. For a given Region-CTL$^\diamond$ formula f, to check whether $[P] \subseteq [f]$, it suffices to show that the set $[f]$ can be effectively computed as a finite union of upper-closed sets (of configurations in M). We prove this by induction on the definition of f. Notice that such finite unions are closed under Boolean operations and obviously, the configuration set defined by a region formula is one such finite union. Hence, the only non-trivial case is to compute $[\exists \diamond\ f]$, assuming that $[f]$ is already a finite union of upper-closed sets. Notice that $[\exists \diamond\ f] = Pre^*_M([f])$. Hence, $[\exists \diamond\ f]$ is also a finite union of upper-closed sets, from Lemma 1. ∎

Proof of Lemma 1

Proof. It is known that Q is definable by some Diophantine equation system iff Q is definable by some Diophantine equation system that is a conjunction of a number of equations, each of which is in one of the following three forms:

$$z = xy, z = x + y, z = 1, \tag{19}$$

where x, y, z are distinct variables. The result follows, since (the solutions to) the equation $z = xy$ in (19) can be definable by the following Diophantine equation system of atomic Diophantine equations, by introducing new variables s, t, u, v, w:

$w = xy + \frac{1}{2}x(x+1) \wedge$
$w = z + u \wedge$
$u = xv + \frac{1}{2}x(x+1) \wedge$
$v + s = t \wedge$
$t = 1 \wedge$
$s = 1.$ ∎

Proof of Lemma 2

Proof. For an atomic Diophantine equation in the form of $z = xy + \frac{1}{2}x(x+1)$, it is easy to verify that the solution set can be defined by the following non-cooperative signaling system M with the signal s_0:

$s_0, a \to aX, s_0,$
$s_0, X \to XZ, s_0,$
$s_0, a \to bc, s_0,$
$s_0, b \to bY, s_0.$

M starts with one a-object, and the number of X-objects (resp. Y-objects, Z-objects) corresponds to the value of variable x (resp. y, z) in the atomic equation. To make the M lazy, one only needs to add the following two rules

$s_0, a_0 \to a_0d, s_0,$
$s_0, a_0 \to ad, s_0,$

and let M starts from one a_0-object (instead of one a-object).

The other two forms of atomic Diophantine equations can be constructed easily. ∎

Proof of Theorem 5

Proof. (⇐). Directly from the definition of (M, C_{init}, P)-definability and Theorem 2 (i.e., the set of reachable configurations of M is recursive).

(⇒). Let E be

$$E_1 \wedge \ldots \wedge E_n, \tag{20}$$

where each E_i is an atomic Diophantine equation. From Lemma 1, it suffices to show that the solution set to E is (M, C_{init}, P)-definable for some desired M, C_{init}, and P, constructed below. For each E_i, using Lemma 2, we can obtain a lazy one-signal system M_i and configuration C^i_{init} such that E_i is definable by M_i. After properly renaming, we may assume that each M_i employs a distinct alphabet and all the M_i's share the only signal s_0. In particular, we assume that each variable x in E_i is uniquely designated to the symbol X^i_x in M_i; i.e., the value of x corresponds to the number $\#(X^i_x)$ of X^i_x-objects in a configuration of M_i. The desired one-signal system M is constructed as follows:

- M's alphabet is the union of the alphabets in all the M_i's;
- M has only one signal s_0;
- M's rules are the union of the rules in all the M_i's;
- M's initial configuration C_{init} is the multiset union of all the C^i_{init}'s.

For each x, E_i, and E_j, whenever x appears in both E_i and E_j, we call $\#(X^i_x) = \#(X^j_c)$ as a *name-constraint*. Let the equality formula P be the conjunction of all the name-constraints. Using the fact that each M_i is lazy, it is not hard to show that the solution set to E is (M, C_{init}, P)-definable. ∎

Proof of Theorem 8

Proof. We only prove the theorem for non-cooperative P systems; we leave it to the reader to generalize the proof to non-cooperative signaling systems with only one signal.

($\Rightarrow$). By definition, let the semilinear set Q be the following finite union of linear sets: $Q_1 \cup \ldots \cup Q_n$, where, without loss of generality, each $Q_i = \{v : v = v_{i0} + v_{i1}t_1 + \ldots + v_{ik}t_k, t_1, \ldots, t_k \in \mathbf{N}\}$, for some $v_{i0}, v_{i1}, \ldots, v_{ik} \in \mathbf{N}^m$ and some k. Let Σ be an alphabet with m symbols, and we still use vector v to denote a multiset on Σ. We now construct a non-cooperative P system $\hat{M}$ as follows. For each $1 \leq i \leq n$, $\hat{M}$ has new symbols $d_{i0}, d_{i1}, \ldots, d_{ik}$, and contains the following rules

$d_{i0} \to v_{i0} d_{i1}$,
$d_{i1} \to v_{i1} d_{i1}$,
$d_{i1} \to d_{i2}$,
$\vdots$
$d_{ik} \to v_{ik} d_{ik}$,
$d_{ik} \to \Lambda$,

which exactly generates from d_{i0}, as halting multisets, all the multisets in E_i. Initially, $\hat{M}$ starts from (one instance of) a new symbol d, and fires one of the following rules: $d \to d_{10}, \ldots, d \to d_{n0}$. Clearly, E is halting-definable by $\hat{M}$.

($\Leftarrow$). Let $\hat{M}$ be a non-cooperative P system over alphabet Σ starting from some given multiset C_{init}. We will show that the set Q of all the reachable and halting configurations in $\hat{M}$ form a semilinear set. We say that $a \in \Sigma$ is a non-terminal symbol if there is a rule $a \to v$ in $\hat{M}$ for some v; we use $N \subseteq \Sigma$ to denote all the non-terminal symbols. Each symbol in $T = \Sigma - N$ is called a terminal symbol. Observe that, when $T = \emptyset$, the result simply follows since the only possible reachable and halting multiset in $\hat{M}$ is the empty multiset. So now, we assume $T \neq \emptyset$. Clearly, a reachable and halting multiset must be a multiset over T. Furthermore, a non-terminal symbol $a \in N$ is *concrete* if there is a multiset u (a halting multiset) over alphabet T such that an a-object can evolve, through a number of maximally parallel moves, into multiset u. We assume that C_{init} does not contain non-concrete objects. Furthermore, we can safely delete rules that contain non-concrete symbols from $\hat{M}$; doing this will not affect the set Q of all the reachable and halting configurations in $\hat{M}$. This is because, during any execution of $\hat{M}$ from C_{init}, when a non-concrete object appears, the execution can never lead to a halting configuration. Therefore, without loss of generality, we assume that every symbol $a \in N$ is concrete.

We now define $\bar{M}$ to be the sequential version of $\hat{M}$. That is, each move in $\bar{M}$ is no longer maximally parallel; the move will nondeterministically choose at most *one* instance of an enabled rule to fire. Here is a key observation: for any halting multiset, it is reachable in $\hat{M}$ iff it is reachable in $\bar{M}$, where both $\hat{M}$ and $\bar{M}$ start from the same C_{init}. For a word w in Σ^*, we use $[w]$ to denote the multiset over alphabet Σ that w corresponds to. Let us fix any word w_0 with $[w_0] = C_{\text{init}}$. Let G be a context-free grammar where the rules are exactly $a \to w$, for some a, w with $a \to [w]$ being a rule in $\hat{M}$. That is, G treats all the rules in $\hat{M}$ as context-free grammar rules. We use Q_G to denote all the words w on T^* such that $w_0 \Rightarrow_G^* w$. It is left to the reader to check that, for any halting multiset v, v is reachable in $\bar{M}$ iff there is a word $w \in Q_G$ with $[w] = v$. Notice that Q_G is a context-free language and hence semilinear. Therefore, Q, which equals the Parikh map of Q_G, is a semilinear set. ∎

Proof of Theorem 9

Proof. Part 1 is directly from Theorem 8. Part 2 (⇐) is similar to the proof of Theorem 5 (⇐).

We now focus on the Part 2 (⇒). Let Q be an r.e. set. In the proof of Theorem 5 (⇒), we demonstrated a non-cooperative signaling system M (with one signal s_0) such that Q is (M, C_{init}, P)-definable for some configuration C_{init}, and some equality formula P. We now modify the non-cooperative signaling system M into M' as follows. M' has two signals s_0, s_1. For each rule $s_0, a \to v, s_0$ in M, we change it into the rule $s_0, a \to v, s_1$. Notice that the signal s_1 will not enable any object (i.e., a "garbage signal"). Additionally, we add a new symbol g and a rule $s_0, g \to g, s_0$. M' starts with the same initial configuration C_{init} but with an additional instance of a g-object. Hence, the g serves as a trigger, upon the s_0 emitted, to active all the other rules like $s_0, a \to v, s_1$. We also add the rule $s_0, g \to g, s_1$. Notice that, once this rule is fired, the signal s_0 disappears and the system enters a halting configuration. At this time, M' fires the equality tester P and output accordingly. Clearly, Q is output-definable by M'. The result follows. ∎