Stably Computable Predicates are Semilinear

Dana Angluin
Department of Computer Science, Yale University
angluin@cs.yale.edu

James Aspnes*
Department of Computer Science, Yale University
aspnes@cs.yale.edu

David Eisenstat
Department of Computer Science, University of Rochester
eisen@cs.rochester.edu

ABSTRACT

We consider the model of **population protocols** introduced by Angluin *et al.* [2], in which anonymous finite-state agents stably compute a predicate of their inputs via two-way interactions in the all-pairs family of communication networks. We prove that all predicates stably computable in this model (and certain generalizations of it) are semilinear, answering a central open question about the power of the model.

Categories and Subject Descriptors

F.1.1 [**Computation by Abstract Devices**]: Models of Computation—*Unbounded-action devices*; F.1.2 [**Computation by Abstract Devices**]: Modes of Computation—*Parallelism and concurrency*

General Terms

Theory

Keywords

Population protocols, semilinear sets, stable computation.

1. INTRODUCTION

In 2004, Angluin *et al.* [2] proposed a new model of distributed computation by very limited agents called a **population protocol**. In this model, finite-state agents interact in pairs chosen by an adversary, with both agents updating their state according to a joint transition function. For each such transition function, the resulting population protocol is said to **stably compute** a predicate on the initial states of the agents if, after sufficiently many interactions, all agents converge to having the correct value of the predicate. Angluin *et al.* showed that many common predicates such as parity of the number of agents, whether there were more agents in some particular initial state a than in another initial state b, and so forth, could be stably computed by simple population protocols. They further showed that population protocols could in fact compute any **semilinear** predicate, which are precisely those predicates definable in first-order Presburger arithmetic [9]. But it was not known whether there were other, stronger predicates that could also be computed by a population protocol, at least in the simplest case where any agent was allowed to interact with any other.

We show that this is not the case: that the semilinear predicates are precisely the predicates that can be computed by a population protocol if there is no restriction on which agents can interact with each other. This gives an exact characterization of the predicates stably computable by population protocols, answering the major open question of [2] as well as giving an exact characterization of the power of several variants of the model, resolving open problems in [1, 3].

Semilinearity is strongly tied to the notion of stable computation, where a correct and stable output state must always be reachable at any step of the computation. Mere reachability of a desired final state is not enough: the reachability sets of population protocols are not in general semilinear. An example of this phenomenon may be derived from the construction given by Hopcroft and Pansiot of a non-semilinear reachability set for a six-dimensional vector addition system [5].

1.1 Population protocols and related models

There are now several variants in the literature of the original population protocol model, which itself can be viewed as a special case of previous models of finite-state distributed systems. The central feature of all of these models is that interactions occur between finite-state agents asynchronously under the control of an adversary, and the goal in each case is to reach a global configuration in which every agent correctly proclaims the desired output of the protocol. To keep the adversary from blocking progress by partitioning the agent population or by allowing interactions only at inconvenient times, a strong global fairness condition is generally assumed, which requires that any global configuration that is continuously reachable is eventually reached.

In each of these models, there is a **state space** Q for individual agents, an **input function** that maps some **input alphabet** Σ to Q, and an **output function** that maps Q to some **output alphabet** (typically just $\{0, 1\}$ when computing predicates). A **configuration** of the protocol describes the states of all the individual agents.

*Supported in part by NSF grants CNS-0305258 and CNS-0435201.

PODC'06, July 22-26, 2006, Denver, Colorado, USA.

There may also be an **interaction graph** that limits which agents can interact with each other. Paradoxically, restricting the interaction graph usually increases the power of the model, since the main difficulty in a dense graph is that the finite-state agents can't distinguish between different neighbors that happen to be in the same state. In the present work the interaction graph is always a complete graph, which (as shown in [2]) gives the weakest model.

A protocol **stably computes** a predicate if it converges to an **output-stable** configuration, in which all agents agree on the correct output and from which any further transitions do not change the mapped output values (though they may change the underlying states).

The differences between the various population protocol models depend on how communication is structured and when the input is provided:

- In the original model of [2], communication is simultaneous and bidirectional: there is **joint transition function** $\delta : Q \times Q \to Q \times Q$ such that the result of an interaction between two agents in states (q_1, q_2) is to replace them with a pair of agents in states $(q_1', q_2') = \delta(q_1, q_2)$. We give a complete characterization of the predicates stably computable in this model, answering the open question from [2].

- In the **stabilizing inputs** variant [1], the input is not provided in the initial configuration; instead, each agent in the population has an input field that can change over time, and convergence to a stable common output value is only required after the inputs stop changing. In [1], it was shown that all semilinear predicates can be computed with stabilizing inputs, but left open whether there existed some (necessarily non-semilinear) predicates that could be computed only with fixed inputs. Our results show that the semilinear predicates are all that can be computed in either model, implying that any population protocol for fixed inputs can be adapted to work with stabilizing inputs.

- In the **one-way** variants of [3], the bidirectional transition function of the original model is replaced by some form of one-way communication, in which only one of the two agents involved in an interaction learns anything about the state of the other. The simplest form of this model is the **immediate transmission** model, where the joint transition function δ is required to update the state of only one of the agents. The more complex **delayed transmission** and **queued transmission** models incorporate explicit asynchronous message-passing, and allow for the possibility of increasing the power of the model by using an unboundedly large buffer of undelivered messages as extra storage. Our results rule out this possibility: the strongest model (queued transmission) can compute exactly the semilinear predicates, while the others can compute precisely the semilinear predicates with a regular k-core as defined in [3].

In each case, we can represent configurations of the original model as nonempty multisets of states (or, equivalently, as nonzero non-negative vectors indexes by states). The only requirement that a model must satisfy for our results to apply is that its **reachability relation** $x \to y$, which is true if there is some sequence of transitions that transforms x to y, must be **additive**: $x \to y$ implies $x + z \to y + z$ for any multiset of sets z. Effectively, this says that a partitioned subpopulation can still run on its own: the agents do not have any mechanism to detect if additional agents are present but not interacting.

Other models that have this property include vector addition systems [5], some forms of Petri nets, and 1-cell catalytic P-systems. In each case our results apply. The relationship between semilinear sets and catalytic P-systems, which can be represented as population protocols in which all state-changing interactions occur between an unmodified catalyst and at most one non-catalyst agent, has previously been considered by Ibarra *et al.* [6], who show that the set accepted by a 1-cell catalytic P-system are semilinear. Our results extend this fact to predicates stably computable in this model, even without the catalytic restriction.

1.2 Semilinearity

A set of vectors in $\mathbb{N}^k$ is **linear** if it is of the form $\{v_0 + \sum_{i=1}^{m} a_i v_i \mid a_i \in \mathbb{N}\}$ where v_0 is some non-negative **base point**, the v_i are non-negative period vectors, and the a_i are nonnegative integer coefficients on the v_i (a more compact version of this definition using monoid cosets is given in Section 2). It is possible for the set of v_i to be empty, in which case the set consists of a single point. A set of vectors is **semilinear** if it is a finite union of linear sets. Thus a linear set consists of a single base point with a cone of points emanating from it, while a semilinear set consists of finitely many such base points, each with its own (possibly trivial) cone.

Semilinear sets are also precisely the sets definable by first-order formulas in **Presburger arithmetic** [9], which are formulas in arithmetic that use only $<$, $+$, 0, 1, and the standard logical quantifiers and connectives. Here the set consists of all satisfying assignments of the free variables; for example, the semilinear set $S = \{(1,0)+a_1(1,0)+a_2(0,2)\} \cup \{(0,2)+a_3(2,0)\}$, depicted in Figure 1, consists precisely of the satisfying assignments (x, y) of the formula

$$\begin{aligned}&(\exists z : (x = z + z + 1) \wedge (y \geq z))\\ \vee\, &(\exists z : (x = z + z) \wedge (y = 1 + 1))\,,\end{aligned} \tag{1}$$

where $a = b$ abbreviates $\neg(a < b \vee b < a)$ and $a \geq b$ abbreviates $\neg(b < a)$.

A curious and useful property of Presburger formulas is that all quantifiers (and their bound variables) can be eliminated by the addition of binary relations $\equiv_m$ that test for equality modulo m for any integer m [9]. For example, the formula (1) defining S can be rewritten without quantifiers as

$$((x \equiv_2 1) \wedge (x \leq y + y + 1)) \vee ((x \equiv_2 0) \wedge (y = 1 + 1)).$$

This provides an easy way to show that some model *can* stably compute all semilinear predicates: show that it can stably compute both $<$ and $\equiv_m$ for any fixed m, and then show that conjunctions (or disjunctions) and negations of stably-computable predicates are also stably computable. These tasks are not especially difficult in the standard population protocol model; for details the interested reader is directed to [2].

For the converse claim, that only semilinear predicates are stably computable in population protocols and similar models, we require substantially more machinery. Here the

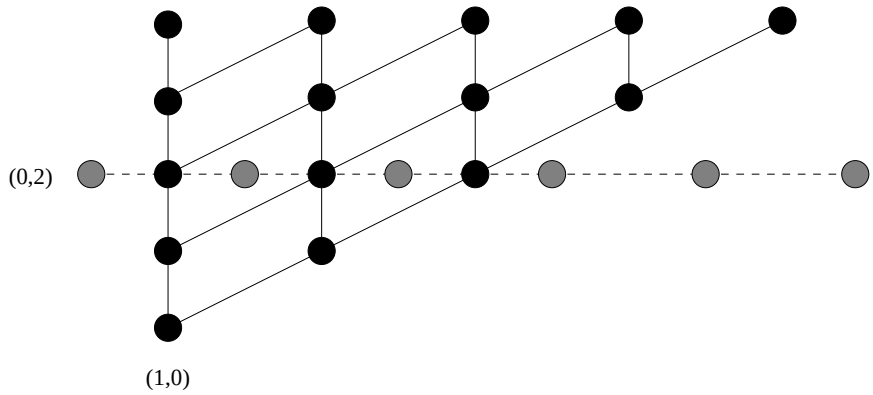

Figure 1: A semilinear set S, equal to the union of the linear set of all points $\{(1,0) + a_1(1,0) + a_2(0,2)\}$ (dark circles) and the linear set $\{(0,2) + a_3(2,0)\}$ (shaded circles).

algebraic characterization of semilinear sets turns out to be more useful.

2. DEFINITIONS

Here we give a formal definition of our model, of stable computation, and of semilinear sets.

2.1 Model

A **protocol** is a 4-tuple $(Q, -\!\!\rightarrow, -\Sigma, \gamma)$, with the following components. Q is a finite set of **states**, which generalize both states and messages from previous models. A **configuration** is a nonempty multiset of states. We identify $C := \mathbb{N}^Q \setminus \{0\}$, the set of nonzero $|Q|$-tuples of nonnegative integers indexed by Q, with the set of all configurations. The usual inclusion ordering on C is denoted $\leq$.

The **reachability relation** $\rightarrow$ is an ordering on C. Specifically, we require that $\rightarrow$ be a reflexive, transitive relation that respects addition in $\mathbb{N}^Q$; that is $c \rightarrow c'$ implies $c + d \rightarrow c' + d$ for all $c, c' \in C$ and $d \in \mathbb{N}^Q$. Intuitively, that $\rightarrow$ respects addition implies that interactions can take place in the presence of other agents. Note that unlike in the standard population protocol model, a configuration may reach configurations with different cardinality.

$\Sigma \subseteq Q$ is a finite set of **input symbols**. An **input** is a nonempty multiset of input symbols. Analogously, we identify $W := \mathbb{N}^\Sigma \setminus \{0\}$ with the set of all inputs. Finally, γ is an **output function**, mapping Q to $\{0, 1\}$. We extend γ to map C to nonempty subsets of $\{0, 1\}$ in the obvious manner by defining

$$\gamma(c) := \{j \in \{0, 1\} \mid \exists q \in Q : \gamma(q) = j \text{ and } c[q] \geq 1\}.$$

2.2 Stable computation

We now define what it means for a protocol to stably compute a predicate. For $j = 0, 1$ we define

$$S_j := \{c \in C \mid \text{if } c \rightarrow c' \text{ then } \gamma(c') = \{j\}\}.$$

Then $S := S_0 \cup S_1$ is the set of **output stable** configurations. Let ψ be a predicate on W, that is, a mapping from W to $\{0, 1\}$. The **support** of ψ is $\psi^{-1}(1)$, that is, the set of inputs that ψ maps to 1. The given protocol **stably computes** ψ if for all $x \in W$ and for all $c \in C$, if $x \rightarrow c$ there exists $c' \in S_{\psi(x)}$ such that $c \rightarrow c'$. That is, if input x reaches any configuration c, then there is a configuration c' reachable from c that is output stable and has output $\psi(x)$. This incorporates the notion of a fair computation[2]. A predicate ψ is **stably computable** if there exists a protocol that stably computes ψ.

2.3 Monoids, groups, and semilinearity

A subset M of $\mathbb{Z}^d$ is a **monoid** if it contains the zero and is closed under addition; if it is also closed under subtraction, M is a **group**. A monoid $M \subseteq \mathbb{Z}^d$ is **finitely generated** if there exists a finite subset $A \subseteq M$ such that every element of M is a sum of elements from A. It is a classic result in abstract algebra that every subgroup of $\mathbb{Z}^d$ is finitely generated [7], but submonoids are not always finitely generated. For example, the following monoid is not finitely generated.

$$M_{\sqrt{2}} = \{(i, j) \in \mathbb{N}^2 : i \leq \sqrt{2}j\}.$$

A subset H of $\mathbb{Z}^d$ is a **group coset** (resp., **monoid coset**) if there exists an element $v \in \mathbb{Z}^d$ such that $H = v + G$ and G is a group (resp., monoid).

A subset L of $\mathbb{Z}^d$ is **linear** if it is a coset of a finitely generated monoid in $\mathbb{Z}^d$, and is **semilinear** if it is a finite union of linear sets. It follows immediately from the correspondence between semilinear sets and Presburger formulas that the semilinear sets are closed under complement, finite intersection and finite union. A predicate ψ on W is semilinear if and only if $\psi^{-1}(1)$ is semilinear; by closure under complement, this is equivalent to $\psi^{-1}(0)$ being semilinear.

3. MAIN RESULT

THEOREM 1. *Every stably computable predicate is semilinear.*

The proof is by combining Lemma 15, which states that stably computable predicates admit finite coset coverings (can be pumped), and Theorem 18, which states that predicates that admit finite coset coverings are semilinear. Corollaries give exact characterizations settling some open problems related to population protocols.

COROLLARY 2. *The predicates stably computable by population protocols are exactly the semilinear predicates.*

PROOF. The definition of stable computation of a predicate by a population protocol is an instance of the more general definition above. In [2] it was shown that every semilinear predicate is stably computable by a population protocol. □

COROLLARY 3. *The predicates stably computable by population protocols with stabilizing inputs are exactly the semilinear predicates.*

PROOF. In [1] it was shown that the semilinear predicates are stably computable with stabilizing inputs by population protocols. □

Considering the models of 1-way population protocols introduced in [3], we have the following results. (Please refer to that paper for definitions of the models.)

COROLLARY 4. *Queued transmission protocols stably compute exactly the semilinear predicates, and are therefore equal in power to population protocols in the standard 2-way model.*

PROOF. The definition of stable computation of a predicate in the queued model is an instance of the more general definition above. (In fact, this was one of the motivations for its generality.) The queued model was shown to be at least as powerful as the standard 2-way model. □

COROLLARY 5. *Predicates stably computable by immediate or by delayed transmission protocols are exactly the semilinear predicates that have a regular k-core.*

PROOF. In [3] it was shown that predicates stably computable by immediate or delayed transmission protocols have a regular k-core, and are stably computable in the queued transmission model. It was also shown that semilinear predicates with a regular k-core are stably computable in the immediate [resp., delayed] transmission model. □

Further corollaries for reversible computation, Petri nets and vector addition systems will appear in the full paper.

4. OVERVIEW OF THE PROOF

To prove the main result, we must show that the support of any stably-computable predicate is a semilinear set: in particular, that there is a finite set of base points each attached to a finitely-generated cone such that the support is precisely the elements of these cones. The first step, which is requires the development of the machinery of Section 5 and is completed in Section 6, is to show that the support can be decomposed into a finite collection of monoid cosets that are *not necessarily finitely generated*. We then proceed, in Sections 7 and 8, to show that any such decomposition can be further decomposed into a finite covering by cosets of finitely generated monoids, which gives us the full result.

5. GROUNDWORK

We assume ψ is a predicate stably computed by a protocol $(Q, -\!\!\rightarrow, -\Sigma, \gamma)$ and establish some basic results. These will lead up to the Pumping Lemma of Section 6.

5.1 Higman's Lemma

We will make extensive use of some corollaries to Higman's Lemma [4], a fundamental tool in well-quasi-order theory.

LEMMA 6. *Every subset of $\mathbb{N}^d$ under the inclusion ordering $\leq$ has finitely many minimal elements.*

LEMMA 7. *Every infinite subset of $\mathbb{N}^d$ contains an infinite chain (i.e., an infinite totally ordered sequence).*

These both follow from the fact that Higman's Lemma implies that $\mathbb{N}^d$ is a **well-quasi-order**, a set in which any infinite sequence $a_1, a_2, \ldots$ contains elements a_i, a_j with $i < j$ and $a_i \leq a_j$.

5.2 Truncation maps and their properties

For each $k \geq 1$, we define a map τ_k from C to C by

$$\tau_k(c)[q] := \min(k, c[q]) \text{ for all } q \in Q.$$

This map truncates each component of its input to be at most k; clearly $\tau_k(c) \leq c$ for all $c \in C$. Two useful properties of τ_k are that it respects both inclusion and addition.

LEMMA 8. *For all $c, d \in C$ and $k \geq 1$, if $c \leq d$ then $\tau_k(c) \leq \tau_k(d)$.*

PROOF. For each $q \in Q$, we have $c[q] \leq d[q]$, so $\min(k, c[q]) \leq \min(k, d[q])$. Thus $\tau_k(c) \leq \tau_k(d)$. □

LEMMA 9. *For all $c, c', d \in C$ and $k \geq 1$, if $\tau_k(c) = \tau_k(c')$, then $\tau_k(c + d) = \tau_k(c' + d)$.*

PROOF. For each $q \in Q$, either $c[q] = c'[q]$ or both are at least k. In either case, $\min(k, c[q] + d[q]) = \min(k, c'[q] + d[q])$, so $\tau_k(c + d) = \tau_k(c' + d)$. □

5.3 Truncation and stability

Truncation is important because membership of a configuration c in S can be determined from a truncate of fixed size. Let $U := C \setminus S$, the set of **output unstable** configurations.

LEMMA 10. *For all $c \leq c'$, if $c \in U$, then $c' \in U$ (U is closed upward under inclusion).*

PROOF. Suppose $c \in U$. Then either $\gamma(c) = \{0, 1\}$ or $\gamma(c) = \{j\}$ for some $j \in \{0, 1\}$ and there exists $c' \in C$ such that $c \rightarrow c'$ and $(1 - j) \in \gamma(c')$. In the first case, for all $d \geq c$ we have $\gamma(d) = \{0, 1\}$ and $d \in U$. In the second case, for all $d \geq c$ we have $d \rightarrow d - c + c'$ and $j \in \gamma(d)$ and $(1 - j) \in \gamma(d - c + c')$, so $d \in U$. □

LEMMA 11. *There exists $k \geq 1$ such that $c \in U$ if and only if $\tau_k(c) \in U$.*

PROOF. By Higman's lemma, only finitely many elements $u_1, \ldots, u_n$ are minimal in U, and because U is upwards closed, $c \in U$ if and only if $u_i \leq c$ for some i. Let k be the maximum component $u_i[q]$ for all $i \in \{1, \ldots, n\}$ and $q \in Q$. Then $\tau_k(u_i) = u_i$ for each i.

Suppose $c \in U$. Then $u_i \leq c$ for some i, so $u_i = \tau_k(u_i) \leq \tau_k(c)$ by Lemma 8, and thus $\tau_k(c) \in U$. Conversely, if $\tau_k(c) \in U$, then $u_i \leq \tau_k(c) \leq c$ for some i, and $c \in U$. □

LEMMA 12. *There exists $k \geq 1$ such that for all $c \in C$ and $j \in \{0, 1\}$, we have $c \in S_j$ if and only if $\tau_k(c) \in S_j$.*

PROOF. By Lemma 11, there exists $k \geq 1$ such that for all $c \in C$, we have $c \in U$ if and only if $\tau_k(c) \in U$. Taking the contrapositive, we have $c \in S$ if and only if $\tau_k(c) \in S$. Since truncation does not affect output, the conclusion follows. □

5.4 Extensions

We define a map X from C to subsets of $\mathbb{N}^\Sigma$ as follows.

$$X(c) := \{x \in \mathbb{N}^\Sigma \mid \exists d \geq c : c + x \to d \text{ and } \tau_k(c) = \tau_k(d)\},$$

where k is the constant from the conclusion of Lemma 12. If $c \in S$, then $X(c)$ is the set of inputs by which c can be pumped. We call such inputs the **extensions** of c. We first prove that pumping does not affect stable output.

LEMMA 13. *If $x \in W$ and $c \in S$ and $x \to c$, then ψ is constant on $x + X(c)$.*

PROOF. If $y \in X(c)$, then there exists $d \in C$ such that $c + y \to d$ and $\tau_k(c) = \tau_k(d)$. Since $c \in S$, by Lemma 12 we have $d \in S$, and $\gamma(c) = \gamma(d)$. Thus $\psi(x) = \psi(x + y)$. □

We now prove that pumping operations can be composed, i.e. that $X(c)$ is a monoid.

LEMMA 14. *$X(c)$ is a monoid for all $c \in C$.*

PROOF. We have $0 \in X(c)$, with $d = c$ as a witness. If $x_1, x_2 \in X(c)$, then there exist d_1, d_2 such that $c \leq d_1$ and $c \leq d_2$ and $\tau_k(c) = \tau_k(d_1) = \tau_k(d_2)$ and $c + x_1 \to d_1$ and $c + x_2 \to d_2$. Thus

$$c + x_1 + x_2 \to d_1 + x_2 = (d_1 - c) + c + x_2 \to (d_1 - c) + d_2.$$

Taking $d := d_1 + d_2 - c$, we have $c \leq d$ and $c + x_1 + x_2 \to d$ and $\tau_k(c) = \tau_k(d_2) = \tau_k(c + d_2 - c)$ and by Lemma 9, $\tau_k(c + d_2 - c) = \tau_k(d_1 + d_2 - c) = \tau_k(d)$, since $\tau_k(c) = \tau_k(d_1)$. We conclude that $x_1 + x_2 \in X(c)$. □

6. A PUMPING LEMMA

Given a set of inputs $Y \subseteq W$, a **monoid-coset covering of Y with respect to ψ** is a set $\{(x_i, M_i)\}_{i \in I}$ of pairs of inputs and submonoids of $\mathbb{N}^\Sigma$ such that $Y \subseteq \bigcup_{i \in I}(x_i + M_i)$ and for all $i \in I$, we have $\psi(x_i + M_i) = \{\psi(x_i)\}$. We write ψ **admits finite coset coverings** if for all Y there exists a finite monoid-coset covering of Y with respect to ψ. The following lemma states that every stably computable predicate admits a finite monoid-coset covering. We show later (Theorem 18) that any predicate that admits finite coset coverings is semilinear.

LEMMA 15. *The (stably computable) predicate ψ admits finite coset coverings.*

PROOF. To avoid trivial cases, assume $Y \subseteq W$ is infinite. Let $y_1, y_2, \ldots$ be any enumeration of Y such that $y_i \leq y_j$ implies $i \leq j$. (For example, we might choose some ordering of Σ and enumerate Y in lexicographic order.) We define a family of sets $B_i \subseteq W \times C$ inductively as follows:

- $B_0 := \emptyset$.
- If there exists $(x, c) \in B_{i-1}$ such that $y_i \in x + X(c)$, then $B_i := B_{i-1}$.
- Otherwise,

  $$\begin{aligned} B_i := & B_{i-1} \\ & \cup \{(y_i, \sigma(y_i))\} \\ & \cup \{(y_i, \sigma(c + y_i - x)) \mid (x, c) \in B_{i-1} \text{ and } x \leq y_i\}, \end{aligned}$$

  where $\sigma(d) \in S$ is any stable configuration reachable from d.

The existence of each $\sigma(c)$ is guaranteed by the requirements of stably computing a predicate. Clearly, each B_i is finite. We now show that $B := \bigcup_{i \geq 1} B_i$ is also finite. By Lemmas 13 and 14, it follows that $\{(x, X(c)) \mid (x, c) \in B\}$ is a finite monoid-covering of Y with respect to ψ.

Assuming to the contrary that B is infinite, infinitely many different elements of Y appear as first components of elements of B. By Higman's lemma, there exists an infinite chain $z_1 < z_2 < \ldots$ of such elements. Our construction guarantees the existence of associated configurations $\{d_i\}_{i \geq 1}$ such that $(z_i, d_i) \in B$ and $d_i + (z_{i+1} - z_i) \to d_{i+1}$ for all $i \geq 1$.

By Higman's lemma again, there exists an increasing function f such that the sequences $(z_{f(i)})_{i \geq 1}$ and $(d_{f(i)})_{i \geq 1}$ are nondecreasing. Thus the sequence $\tau_k(d_{f(i)})$ reaches a maximum and becomes constant at some index $i = j$. Consequently, we have $z_{f(j+1)} - z_{f(j)} \in X(d_{f(j)})$, which contradicts the membership of $(z_{f(j+1)}, d_{f(j+1)})$ in B. □

Applying this lemma with $Y = \psi^{-1}(1)$ (the support of ψ) we obtain a finite family of monoid cosets $x_i + M_i$ such that

$$\psi^{-1}(1) = \bigcup_i (x_i + M_i).$$

This does not prove semilinearity by itself, since some M_i might not be finitely generated. However, every submonoid of $\mathbb{Z}^1$ is finitely generated, so in the special case of a *unary* alphabet, we have already that ψ is semilinear, and therefore regular.

COROLLARY 16. *Every stably computable predicate over a unary alphabet is semilinear. Thus, over a unary alphabet, the stably computable predicates are exactly the semilinear (in fact, regular) predicates.*

For example, the unary predicate that is true if the number of input symbols is a power of 2 (or a prime, or any other non-regular predicate) is not stably computable. Another easy corollary suffices to show certain other predicates over non-unary alphabets are not stably computable.

COROLLARY 17. *Suppose ψ is a stably computable predicate such that $L = \psi^{-1}(1)$ is infinite. Then L contains an infinite linear subset.*

PROOF. By Lemma 15 there is a finite monoid-coset covering of L. If L is infinite, some $(x + M) \subseteq L$ in the covering must be infinite. □

As an application, consider the set of inputs over the alphabet $\{a, b\}$ such that the number b's is the square of the number of a's. This is an infinite set with no infinite linear subset, and is therefore not stably computable. Using closure results for stably computable predicates we can then show that the set of all inputs over alphabet $\{a, b, c\}$ such that the number of c's is the product of the number of a's and the number of b's is not stably computable. The existence of a pumping lemma and the negative results for these particular predicates were conjectured in [2].

7. PROOF OF THE MAIN RESULT: OUTLINE

We are now in a position to describe how we go from the pumping lemma (Lemma 15) of Section 6 to the main result.

Recall the following set of points in $\mathbb{N}^2$.

$$M_{\sqrt{2}} = \{(i,j) : i \leq \sqrt{2}j\}.$$

This is a monoid but is not stably computable. To see this, suppose the contrary. By the Pumping Lemma, there exist pairs (x_i, c_i) for $1 \leq i \leq m$ such that

$$M_{\sqrt{2}} = \bigcup_{1 \leq i \leq m} (x_i + X(c_i)).$$

Let $v = (-1, \sqrt{2})$; then $x \cdot v > 0$ for all $x \in M_{\sqrt{2}}$. Let $\epsilon = \min_i\{x_i \cdot v\}$. Choose some $y \in M_{\sqrt{2}}$ such that $0 < y \cdot v < \epsilon$. Then for some i, $y \in x_i + X(c_i)$, and $(y - x_i) \cdot v < 0$. Thus for a sufficiently large $m \in \mathbb{N}$, $(x_i + m(y - x_i)) \cdot v < 0$, which contradicts the fact that $x_i + X(c_i)$ is a subset of $M_{\sqrt{2}}$. The issue here is that the line dividing the positive and negative inputs cannot have an irrational slope if the predicate is stably computable. One ingredient of our proof is a generalization of this idea to separating hyperplanes.

However, to be able to use separating hyperplanes, we first must deal with separating "intermixed" positive and negative inputs using their images in a finite group. For example, consider the following set of points in $\mathbb{N}^2$.

$$L = \{(i,j) : i < j, (i+j) \text{ is odd}\}.$$

By first separating the points in $\mathbb{N}^2$ by their images (i mod $2, j$ mod 2) in the group $\mathbb{Z}_2 \times \mathbb{Z}_2$, we get four subproblems in which lines suffice to separate the positive and negative points.

A further issue is that in the most general case, the separating hyperplanes (which we can think of as cosets of finitely-generated groups) may themselves include points that require further handling. This is dealt with by induction on the dimension of the hyperplane, but requires careful handling of the relationship between monoids $X(c)$ and their cosets $x + X(c)$. We shall prove the following theorem by induction on the dimension of the group G; it clearly holds when G has dimension 0, i.e., when G is the trivial group.

THEOREM 18. *If ψ admits finite coset coverings then for any group coset $H = x_0 + G \subseteq \mathbb{Z}^\Sigma$, we have $\psi^{-1}(1) \cap H$ is semilinear.*

Note that together with Lemma 15 this implies Theorem 1, as we can take $H = \mathbb{Z}^\Sigma$.

The details of the proof are quite involved and are given in Section 8. To summarize briefly, the overall strategy is:

1. By dividing the space into residue classes with respect to appropriately chosen moduli, we can arrange for the vectors from the monoids associated with the cover to appear in all-positive and all-negative regions separated by hyperplanes. In this part of the proof we extend $\mathbb{N}^\Sigma$ to $\mathbb{Z}^\Sigma$ to make use of the fact that all subgroups of $\mathbb{Z}^\Sigma$ are finitely generated. (Section 8.1.)

2. We then further map the problem from $\mathbb{Z}^\Sigma$ to $\mathbb{R}^\Sigma$, and use techniques from convex geometry to show that appropriate hyperplanes separating the monoids indeed exist. (Section 8.2.) We obtain a (looser) separation by **slabs** (the space between two parallel hyperplanes) on the inputs by observing that each input is displaced a uniformly bounded amount from its corresponding extension vector.

3. Moving to $\mathbb{R}^\Sigma$ allows for the possibility of separating hyperplanes with irrational coefficients. Applying the Pumping Lemma as described above, we show that the resulting separating hyperplanes are normal to a vector with rational coordinates and thus correspond to cosets of subgroups of $\mathbb{Z}^\Sigma$. (Section 8.3.)

4. At this stage, we have shown that the positive inputs to the predicate consist of (a) those inputs in the interior of the separated regions, which can by identified first by computing the residue classes of their coordinates and then by identifying which of a finite number of polytopes (given by intersections of half-spaces with rational coordinates) they appear within, and (b) those inputs that lie in some slab. The first class is semilinear as identification of a residue class and identification of membership in a particular polytope are both expressible in Presburger arithmetic. The second class is then shown to be semilinear by induction on dimension, with the base case of dimension 0 being the trivially semilinear case of a single point. (Section 8.4.)

8. PROOF OF THE MAIN RESULT: DETAILS

The proof of Theorem 18 is delayed to Section 8.4. First we describe the process of separating inputs in more detail.

8.1 Separating intermixed inputs

Let $\psi : W \to \{0,1\}$ be a predicate that admits finite coset coverings. Let G be a subgroup of $\mathbb{Z}^\Sigma$ and let $H = h + G$ be a coset of G. Take $\{(a_i, M_i)\}_{i \in I}$ and $\{(b_j, N_j)\}_{j \in J}$ to be finite monoid-coset covers of $\psi^{-1}(0) \cap H$ and $\psi^{-1}(1) \cap H$ respectively. Assume without loss of generality that $M_i, N_j \subseteq G$ by intersecting with G if necessary. Define $K(i,j) := \mathbb{Z}(M_i \cap N_j)$, the group generated by the intersection of M_i and N_j.

LEMMA 19. *For all $i \in I$ and $j \in J$, no coset of $K(i,j)$ intersects both $a_i + M_i$ and $b_j + N_j$.*

PROOF. If we suppose to the contrary, there exist $x \in a_i + M_i$ and $x' \in b_j + N_j$ such that $(x - x') \in K(i,j)$. Because $M_i \cap N_j$ is a monoid, we can rewrite $x - x'$ as a difference $y' - y$ where $y' \in M_i$ and $y \in N_j$. Thus $x + y = x' + y'$. The former is in $a_i + M_i$, whereas the latter is in $b_j + N_j$. This contradicts the fact that $a_i + M_i$ and $b_j + N_j$ are disjoint (they have different predicate values). □

Define

$$K := \bigcap_{K(i,j) \text{ has finitely many cosets in } G} K(i,j).$$

In addition to having finitely many cosets in G, the group K inherits the relevant properties of the groups $K(i,j)$ included in its defining intersection.

LEMMA 20. *For all $i \in I$ and $j \in J$ such that $K(i,j)$ has finitely many cosets in G, no coset of K intersects both $a_i + M_i$ and $b_j + N_j$.*

PROOF. Since $K(i,j) \supseteq K$, each coset of $K(i,j)$ is a union of cosets of K. □

Since all group cosets are semilinear, we can use membership in the cosets of K to separate those pairs of monoid cosets that are intermixed. For the others, we have to take another approach.

8.2 Separating with hyperplanes

Let $g_1 + K, \ldots, g_n + K$ be the cosets of K in G. For each $\ell \in \{1, \ldots, n\}$ define $H_\ell := h + g_\ell + K$ and $I_\ell := \{i \in I \mid (a_i + M_i) \cap H_\ell \neq \emptyset\}$ and $J_\ell := \{j \in J \mid (b_j + N_j) \cap H_\ell \neq \emptyset\}$. Lemma 20 guarantees that $K(i,j)$ does not have finitely many cosets in G for all $i \in I_\ell$ and $j \in J_\ell$.

At this point we turn to the methods of geometry. We pass from groups to vector spaces by working in the vector space closure $\mathbb{R}G$ of G in $\mathbb{R}^\Sigma$. Instead of monoids, we consider the convex cones they generate: the set of nonnegative linear combinations of monoid elements. We connect the geometry to the algebra by observing that $K(i,j)$ has finitely many cosets in G if and only if $\mathbb{R}K(i,j)$, the vector space closure of $K(i,j)$, is all of $\mathbb{R}G$. Thus if two monoid cosets are not intermixed, their intersection of their associated monoids has strictly smaller dimension than G. This allows us to separate these monoids with a hyperplane.

Formally, given sets of vectors U and U', a set of nonzero vectors V **distinguishes** U **and** U' if for all $u \in U$ and $u' \in U'$, there exists $v \in V$ such that $(u \cdot v)(u' \cdot v) \leq 0$; that is, either one dot product is zero or one is negative and the other is positive. Define $\widehat{M}_\ell := \bigcup_{i \in I_\ell} M_i$ and $\widehat{N}_\ell := \bigcup_{j \in J_\ell} N_j$. The goal of this subsection is to show the existence of a finite set of vectors V that distinguishes $\widehat{M}_\ell$ from $\widehat{N}_\ell$. The main tool we use is the Separating Hyperplane Theorem from convex geometry. Note that $\operatorname{int} U$ denotes the interior of U.

THEOREM 21. *(Separating Hyperplane Theorem [8]) If U and U' are convex subsets of $\mathbb{R}^d$ with nonempty interiors such that $\operatorname{int} U \cap \operatorname{int} U' = \emptyset$, then there exists $v \in \mathbb{R}^d$ such that $u \cdot v \leq 0$ for all $u \in U$ and $u' \cdot v \geq 0$ for all $u' \in U'$.*

Unfortunately, M_i and N_j are not convex. Thus we are forced to consider the convex cones that they generate. We need Carathéodory's theorem, another result from convex geometry, to verify the seemingly trivial fact that the intersection of these cones lies in a proper vector subspace of $\mathbb{R}G$.

THEOREM 22. *(Carathéodory's Theorem [8]) For any set of vectors $Y \subseteq \mathbb{R}^d$, if $x \in \mathbb{R}_+ Y$, then there exists a linearly independent subset $Y' \subseteq Y$ such that $x \in \mathbb{R}_+ Y'$.*

LEMMA 23. *For all $i \in I_\ell$ and $j \in J_\ell$, the set $Z = \mathbb{R}_+ M_i \cap \mathbb{R}_+ N_j$ is contained in a proper vector subspace of $\mathbb{R}G$.*

PROOF. Suppose to the contrary. Then the vector subspace $\mathbb{R}Z$ is all of $\mathbb{R}G$ and has a basis $z_1, \ldots, z_d \subseteq Z$. Since $\mathbb{Q}Z$ is dense in $\mathbb{R}Z$, we assume without loss of generality that $z_s \in Z \cap \mathbb{Q}Z$ for all $s \in \{1, \ldots, d\}$.

By Theorem 22, for each s, there is a linearly independent set $Y_s \subseteq M_i$ such that $z_s \in \mathbb{R}_+ Y_s$. When we write each z_s as its unique linear combination of elements in Y_s, we see by linear algebra over $\mathbb{Q}G$ that each coefficient is rational. Repeating this argument with N_j yields that $z_s \in \mathbb{Q}_+ M_i \cap \mathbb{Q}_+ N_j$. We clear denominators to find numbers m_s such that $m_s z_s \in M_i \cap N_j$. The set $\{m_s z_s\}$, however, is a basis for $\mathbb{R}G$, which contradicts the fact that $\mathbb{R}K(i,j)$ is not all of $\mathbb{R}G$. □

LEMMA 24. *For all $i \in I_\ell$ and $j \in J_\ell$, there exists a nonzero vector v such that $\{v\}$ distinguishes I_ℓ and J_ℓ.*

PROOF. If either M_i or N_j is contained in a proper vector subspace of $\mathbb{R}G$, then take v to be normal to that subspace.

Otherwise, consider the sets $U := \mathbb{R}_+ M_i$ and $U' := \mathbb{R}_+ N_j$. The intersection of their interiors is both open and contained in a proper vector subspace of $\mathbb{R}G$. Thus U and U' have no interior point in common. Therefore, by Theorem 21 there exists $v \in \mathbb{R}G$ such that for all $u \in U$, we have $u \cdot v \leq 0$; and for all $u' \in U'$, we have $u' \cdot v \geq 0$. It follows that $\{v\}$ distinguishes M_i and N_j. □

To obtain a set of vectors that distinguish $\widehat{M}_\ell$ and $\widehat{N}_\ell$, we put together all of the individual distinguishing vectors.

LEMMA 25. *For all $1 \leq \ell \leq n$, there exists a finite set of vectors V that distinguishes $\widehat{M}_\ell$ and $\widehat{N}_\ell$.*

PROOF. The set $V := \{v(i,j) \mid i \in I_\ell \text{ and } j \in J_\ell\}$ distinguishes $\widehat{M}_\ell$ and $\widehat{N}_\ell$, where $\{v(i,j)\}$ is the vector from Lemma 24 such that $\{v(i,j)\}$ distinguishes M_i and N_j. □

Combining results in this section and the previous one, we have some powerful criteria for determining the predicate value of an input.

LEMMA 26. *Let V be a set of vectors that distinguishes $\widehat{M}_\ell$ and $\widehat{N}_\ell$. Suppose $x \in \widehat{M}_\ell$ and $y \in H_\ell$ are such that for all $j \in J_\ell$ and $v \in V$, we have $(x \cdot v)((y - b_i) \cdot v) > 0$. Then $\psi(y) = 0$.*

PROOF. Suppose to the contrary that $\psi(y) = 1$. Then there exists $j \in J_\ell$ such that $(y - b_j) \in N_j$. The set V, however, fails to distinguish x and $y - b_j$, which is a contradiction. □

Clearly, this lemma has an analogous counterpart that establishes sufficient conditions for $\psi(y) = 1$.

8.3 Achieving rationality

The chief obstacle yet to be overcome is that a distinguishing set of vectors might include vectors with irrational coordinates. In order to rule out predicates like $\psi(r,s) := [r < (\sqrt{2})s]$, we need to show that we can always distinguish $\widehat{M}_\ell$ and $\widehat{N}_\ell$ by vectors with integral coordinates. We must use for a second time the fact that ψ admits finite coset covers.

LEMMA 27. *For all ℓ there exists a finite set of vectors $V \subseteq \mathbb{Z}^\Sigma$ that distinguishes $\widehat{M}_\ell$ and $\widehat{N}_\ell$.*

PROOF. By clearing denominators, it is enough to find $V \subseteq \mathbb{Q}^\Sigma$. By Lemma 25 there exists a finite set of vectors V that distinguishes $\widehat{M}_\ell$ and $\widehat{N}_\ell$. Assume that $V \setminus \mathbb{Q}^\Sigma$ has minimum cardinality, that is, as few vectors in V have irrational components as possible. To avoid a special case later, we assume without loss of generality that V contains the standard basis Σ of $\mathbb{R}^\Sigma$.

Suppose to the contrary that there exists $v \in V \setminus \mathbb{Q}^\Sigma$. By assumption, the set $V' := V \setminus \{v\}$ cannot distinguish $\widehat{M}_\ell$ and $\widehat{N}_\ell$. Thus there exist $i \in I_\ell$ and $j \in J_\ell$ and $x \in M_i$ and $y \in N_j$ such that for all $v' \in V'$ we have $(x \cdot v')(y \cdot v') > 0$.

We consider two cases. In the first case, for all choices of x and y we have $(x \cdot v)(y \cdot v) = 0$. Then at least one vector in each problem pair is normal to v. By linear algebra over $\mathbb{Q}^\Sigma$, there exists a vector $v' \in \mathbb{Q}^\Sigma$ such that if $w \in G$ is normal to v, then w is normal to v'. Thus $V' \cup \{v'\}$ distinguishes $\widehat{M}_\ell$ and $\widehat{N}_\ell$, which is a contradiction, since $(V' \cup \{v'\}) \setminus \mathbb{Q}^\Sigma$ has fewer elements than $V \setminus \mathbb{Q}^\Sigma$.

In the second case, we have x and y such that $(x \cdot v')(y \cdot v') > 0$ for all $v' \in V'$ and $(x \cdot v)(y \cdot v) < 0$. Assume without loss of generality that $x \cdot v < 0 < y \cdot v$ by taking $-v$ instead of v if necessary. Let $\Omega := \{w \in \mathbb{R}G \mid v' \cdot w > 0 \text{ for all } v' \in V \text{ and } w[q] > 0 \text{ for all } q \in \Sigma\}$. Clearly Ω is an open set. Also, $y \in \Omega$, since by the assumption that $\Sigma \subseteq V$ we have $y[q] > 0$ for all $q \in \Sigma$. Given that Ω is a nonempty open set, we can extend x, y to a basis $w_1 = x, w_2 = y, \ldots, w_m$ such that $w_s \in \Omega$ for all $s \geq 2$. By perturbing each w_s slightly to have rational coordinates and clearing denominators, we assume without loss of generality that $w_s \in \Omega \cap K$. There exists s such that $(w_s \cdot v)/(w_1 \cdot v)$ is irrational, since otherwise some scalar multiple of v belongs to $\mathbb{Q}^\Sigma$.

In consequence $\mathbb{N}(x \cdot v) + \mathbb{N}(w_s \cdot v)$ is dense in $\mathbb{R}$, so we can find sequences of positive integers m_t and m'_t such that

$$m_t(x \cdot v) + m'_t(w_s \cdot v)$$

is a negative monotone increasing sequence of real numbers approaching 0 as t approaches infinity. For all $v' \in V$, the points $(m_t x + m'_t w_s)_{t \geq 1}$ lie on the same side of the hyperplane normal to v, and as a sequence they approach the hyperplane normal to v arbitrarily closely. By another application of Higman's lemma, there is an increasing function f such that $(m_{f(t)}, m'_{f(t)})$ is an increasing sequence.

Let $z \in (a_i + M_i) \cap H_\ell$. There exists some constant $c \geq 0$ such that for each r, the points $((z+(c+m_{f(t)})x+m'_{f(t)}w_s)-b_r)_{t\geq 1}$ all lie on the same side of each hyperplane as x. It is easily verified that each of these points belongs to H_ℓ. Thus by Lemma 26, $\psi(z + (c + m_{f(t)})x + m'_{f(t)}w_s)$ is constantly 0. Apply the pumping lemma (Lemma 15) again to these inputs to obtain a finite cover. Then there exist $t_1 < t_2$ such that $(m_{t_2} - m_{t_1})x + (m'_{t_2} - m'_{t_1})w_s \in P$, where the monoid coset $(z+m_{t_1}x+m_{t_2}w_s)+P$ belongs to the cover. Pumping $z + (c + m_{t_1})x + m'_{t_1}w_s$ by a sufficiently large multiple of $(m_{t_2} - m_{t_1})x + (m'_{t_2} - m'_{t_1})w_s$ yields an element z' such that $\psi(z') = 0$, but for each r, we have that $z' - a_r$ is on the same side of each hyperplane as w_s, which contradicts Lemma 26. □

8.4 Proof of Theorem 18

We can now prove Theorem 18 by induction on the dimension of G (the cardinality of the largest linearly independent subset of G.)

PROOF. If the dimension of G is zero, then H is a single point and the result holds. If the dimension of G is greater than zero, then by Lemmas 20 and 27, there exist a group K and finite distinguishers $V_\ell \subseteq \mathbb{Z}^\Sigma$ for all cosets H_ℓ of K in H.

For each distinguishing vector $v \in V_\ell$, consider the set of points $x \in H_\ell$ such that it is not that the case that the following numbers are either all negative or all positive: $(x - a_i) \cdot v$ for $i \in I_\ell$ and $(x - b_j) \cdot v$ for $j \in J_\ell$. This set has finite width in the direction of v. Thus it can be written as the union of finitely many cosets of a group of smaller dimension. The key to the induction is that any point not in the union of these sets over the different x satisfies the hypotheses of one of the variants of Lemma 26.

Define the sets

$$B := \bigcup_\ell \left\{ x \in H_\ell \mid \begin{array}{l} \exists y \in \widehat{N}_\ell : ((x - b_j) \cdot v)(y \cdot v) > 0 \\ \text{for all } j \in J_\ell \text{ and } v \in V_\ell \end{array} \right\}$$

$$B' := \bigcup_\ell \left\{ x \in H_\ell \mid \begin{array}{l} \nexists y \in \widehat{M}_\ell : ((x - a_i) \cdot v)(y \cdot v) > 0 \\ \text{for all } i \in I_\ell \text{ and } v \in V_\ell \end{array} \right\}.$$

By Lemma 26 we have $B \subseteq \psi^{-1}(1) \subseteq B'$. It is not difficult to verify that B and B' are semilinear. Moreover, $B' \setminus B$ is a union of finitely many group cosets of dimension less than G. It follows by inductive hypothesis that $\psi^{-1}(1) \cap (B' \setminus B)$ is semilinear, and thus that ψ itself is semilinear. □

9. REFERENCES

[1] Dana Angluin, James Aspnes, Melody Chan, Michael J. Fischer, Hong Jiang, and René Peralta. Stably computable properties of network graphs. In Viktor K. Prasanna, Sitharama Iyengar, Paul Spirakis, and Matt Welsh, editors, *Distributed Computing in Sensor Systems: First IEEE International Conference, DCOSS 2005, Marina del Rey, CA, USA, June/July, 2005, Proceedings*, volume 3560 of *Lecture Notes in Computer Science*, pages 63–74. Springer-Verlag, June 2005.

[2] Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer, and René Peralta. Computation in networks of passively mobile finite-state sensors. In *PODC '04: Proceedings of the Twenty-Third Annual ACM Symposium on Principles of Distributed Computing*, pages 290–299. ACM Press, 2004.

[3] Dana Angluin, James Aspnes, David Eisenstat, and Eric Ruppert. On the power of anonymous one-way communication. In *Ninth International Conference on Principles of Distributed Systems*, pages 307–318, December 2005.

[4] G. Higman. Ordering by divisibility in abstract algebras. *Proceedings of the London Mathematical Society*, 3(2):326–336, 1952.

[5] J. Hopcroft and J. Pansiot. On the reachability problem for 5-dimensional vector addition systems. *Theoretical Computer Science*, 8(2):135–159, 1978.

[6] Oscar H. Ibarra, Zhe Dang, and Omer Egecioglu. Catalytic p systems, semilinear sets, and vector addition systems. *Theor. Comput. Sci.*, 312(2-3):379–399, 2004.

[7] Serge Lang. *Algebra (Revised Third Edition)*. Springer-Verlag, 2002.

[8] Steven R. Lay. *Convex Sets and their Applications*. Krieger Publishing Company, 1992.

[9] Mojzesz Presburger. Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt. In *Comptes-Rendus du I Congrès de Mathématiciens des Pays Slaves*, pages 92–101, Warszawa, 1929.