Parikh Membership Problems

Oscar H. Ibarra

Department of Computer Science
University of California at Santa Barbara

(Joint work with B. Ravikumar, Sonoma State Univ.)

Supported in part by NSF Grants CCF-1143892 and CCF-1117708.

Problems

- Determine if a string w belongs to a language L specified by an automaton (NFA, PDA, etc.) where the string w is specified by its Parikh vector.
- Investigate the complexity of this problem under various scenarios: the Parikh vector is given in unary, binary, the automaton is fixed (i.e., not part of the input), the automaton is not fixed and part of the input.
- Related problems.

Outline

Motivation

- Membership problems are the most fundamental problems in computation theory.
- We study a variation in which the input string is specified by its Parikh vector, i.e., by a vector $(n_1, n_2, ..., n_k)$ where k is the alphabet size and n_i is the number of occurrences of the i-th letter.
- A potential application area involves pattern matching in which some symbols are allowed to commute [Kopczynski and To].

- Our study was also motivated by a tiling problem (posed by D. Scott) for which a polynomial time algorithm follows from the membership problem studied here.
- Membership problem and other problems (such as equivalence, containment etc.) involving Parikh vectors have been studied before.
- Prior studies have generally assumed that the language is also represented as a semilinear set. In this setting, the complexity of membership and equivalence problems have been investigated, e.g., by [Hyunh].

- Similar membership problems have been studied [Esparza; Kopczynski and To]. Main difference is here, we present:
  – NP-hardness result even when restricted to NFA's accepting a bounded language.
  – The positive result (e.g., PTIME algorithm) for a wider class of languages (those that are accepted by PDA's augmented by reversal-bounded counter machines).
  – An application to a class of tiling problems.

Definitions

- Let N be the set of non-negative integers and $k \geq 1$.
- $Q \subseteq N^k$ is a *linear set* if there is a vector c in N^k (the constant vector) and a set of periodic vectors $V = \{v_1, \ldots, v_r\}$, $r \geq 0$, each v_i in N^k such that $Q = \{c + t_1 v_1 + \cdots + t_r v_r \mid t_1, \ldots, t_r \in N\}$. We denote this set as $Q(c, V)$.

- A finite union of linear sets is called a *semilinear set*.
- Let $\Sigma = \{a_1, \ldots, a_k\}$. For $w \in \Sigma^*$, let $|w|$ be the number of letters (symbols) in w, and $|w|_{a_i}$ denote the number of occurrences of a_i in w.
  – *Parikh map* $P(w)$ of w is the vector $(|w|_{a_1}, \ldots, |w|_{a_k})$.
  – *Parikh image* of a language L is $P(L) = \{P(w) \mid w \in L\}$.
- $L \subseteq w_1^* \cdots w_k^*$ (the w_i's non-null, not necessarily distinct) is a *semilinear language* if the set $\{(n_1, \ldots, n_k) \mid w_1^{n_1} \cdots w_k^{n_k}\}$ is a semilinear set.

- NPDA = nondeterministic pushdown automaton
- DPDA = deterministic pushdown automaton
- NFA = nondeterministic finite automaton
- DFA = deterministic finite automaton
- DLOGSPACE = languages accepted by log-space DTMs
- NLOGSPACE = languages accepted by log-space NTMs
- PTIME = languages accepted by polytime DTMs
- PSPACE = languages accepted by polyspace DTMs

- A *counter* is an integer variable that can be incremented by 1, decremented by 1, left unchanged, and tested for zero. It starts at zero and cannot store negative values.
- An automaton (NFA, NPDA, etc.) can be augmented with counters, where the "move" of the machine also now depends on the status (zero or non-zero) of the counters, and the move can update the counters.
- Well-known: A DFA augmented with two counters is equivalent to a TM [Minsky].
- 1-reversal counter: can only "reverse" once, i.e., once it decrements, it can no longer increment.

- A counter that makes r reversals can be simulated by $\lceil \frac{r+1}{2} \rceil$ 1-reversal counters.
- Automata with 1- reversal counters can "count", e,g,. $L = \{xx^r \mid x \in (a+b)^+, |x|_a = |x_b|\}$ can be accepted by an NPDA with two 1-reversal counters. $L_k = \{x_1\#\cdots\#x_k \mid , x_i \in (a+b)^+, x_j \neq x_k \text{ for } j \neq k\}$ can be accepted by an NFA with $k(k+1)/2$ 1-reversal counters.
- Known [Ibarra]:
  – Emptiness and infiniteness problems for NPDAs with 1-reversal counters are decidable.
  – Disjointness is decidable for NFAs with 1-reversal counters.
  – Containment and equivalence are decidable for DFAs with 1-reversal counters.
- Universality is undecidable for NFAs with a single 1-reversal counter.

M Fixed. Input: Vector $(n_1, \ldots, n_k)$ in Unary

– Complexity will be a function of $n_1 + \cdots + n_k$.
– We will show that the problem is in DLOGSPACE.

Theorem

Let M be an NPDA with 1-reversal counters such that $L(M) \subseteq \{a_1, ..., a_k\}^*$. Let $L_M = \{a_1^{n_1} \cdots a_k^{n_k} \mid$ there exists w in $L(M)$ such that for $1 \leq i \leq k$, w has exactly n_i occurrences of a_i $\}$. Then L_M can be accepted by a DFA with 1-reversal counters that runs in linear time.

– Note that L_M is the Parikh image of the language $L(M)$.
– Obviously the above holds for NFA with 1-reversal counters.

Proof

- Construct an NPDA M' with 1-reversal counters that accepts L_M as follows:
  – M' has k new 1-reversal counters $C_1, .., C_k$.
  – M' on input $a_1^{n_1} \cdots a_k^{n_k}$ first reads the input and stores $n_1, ..., n_k$ in counters $C_1, C_2, ..., C_k$.
  – Then M' guesses an input w to M symbol-by-symbol and simulates M. It also decrements counter C_i whenever it guesses symbol a_i in w.
  – When all the counters become zero, M' accepts if and only if w is accepted by M.
- It is known [Ibarra] that any language $B \subseteq w_1^* \cdots w_k^*$, where $k \geq 1$ and $w_1, ..., w_k$ are non-null strings, accepted by an NPDA with 1-reversal counters is a semilinear language. Hence, L_M is a semilinear language.

- It is known that $B \subseteq w_1^* \cdots w_k^*$ is a semilinear language iff B is accepted by a DFA augmented with 1-reversal counters [Ibarra and Seki].
- For every NFA M with 1-reversal counters, there is a constant c such that every string x of length n in $L(M)$ can be accepted by M within cn time (even if x is non-bounded) [Baker and Book].
- Hence, L_M can be accepted by a DFA augmented with 1-reversal counters that runs in linear time. The theorem follows. □

Corollary

L_M is in DLOGSPACE and, hence, in PTIME.

Generalization:

Theorem

Let M be an NPDA augmented with 1-reversal counters, $k \geq 1$, and $w_1, w_2, \ldots, w_k$ non-null strings. Let $L_M = \{w_1^{n_1} w_2^{n_2} \cdots w_k^{n_k} \mid$ there exists w in $L(M)$ such that for each i, w has exactly n_i occurrences of w_i, and all the occurrences of w_1's, ... , w_k's are not overlapping (hence $|w| = n_1|w_1| + \cdots + n_k|w_k|$) $\}$. Then L_M can be accepted by a DFA augmented with 1-reversal counters that runs in linear time. Hence, L_M is in DLOGSPACE and in PTIME.

M Fixed. Input: Vector $(n_1, \ldots, n_k)$ in Binary

The complexity will be a function of $log(n_1) + \cdots + log(n_k)$.
Need the following result [Lenstra]:

Theorem

Let S be a system of linear constraints:

$$v_{11}x_1 + v_{12}x_2 + \cdots + v_{1m}x_m \leq n_1$$

$$.....$$

$$v_{k1}x_1 + v_{k2}x_2 + \cdots + v_{km}x_m \leq n_k$$

where $k, m \geq 1$ *and the* n_i*'s, and the* v_{ij}*'s are integers (+, -, 0), represented in binary. When m (the number of variables) or k (the number of equations) is fixed, deciding if the system has an integer solution (+, -, 0) for* $x_1, ..., x_m$ *is in PTIME.*

When the n_i's and the v_{ij}'s are non-negative and the inequalities become equalities:

Corollary

Let S_1 be a system of linear equations:

$$v_{11}x_1 + v_{12}x_2 + \cdots + v_{1m}x_m = n_1$$

$$.....$$

$$v_{k1}x_1 + v_{k2}x_2 + \cdots + v_{km}x_m = n_k$$

where $k, m \geq 1$ and the n_i's, and the v_{ij}'s are non-negative integers, represented in binary. When m is fixed, deciding if the system S_1 has a non-negative integer solution for $x_1, ..., x_m$ is in PTIME.

Proof (idea): Transform S_1 to Lenstra's system S with m variables and $2k + m$ inequalities. □

When m (number of variables) in the system S_1 above is not fixed, the corollary is no longer valid, even when the number of variables $k = 1$:

Theorem (Lueker)

Deciding, given an equation of the form,

$$v_1 x_1 + v_2 x_2 + \cdots + v_m x_m = n$$

(where $m \geq 1$, v_i's and n are non-negative integers), whether it has a non-negative integer solution is NP-hard.

When the coefficients are bounded by a fixed positive integer:

Theorem

Let S_2 be a system of linear equations:

$$v_{11}x_1 + v_{12}x_2 + \cdots + v_{1m}x_m = n_1$$

$$.....$$

$$v_{k1}x_1 + v_{k2}x_2 + \cdots + v_{km}x_m = n_k$$

where $k, m \geq 1$ and the n_i's, and the v_{ij}'s are non-negative integers (represented in binary) such that the v_{ij}'s are bounded by a **fixed** *positive integer d. When (the number of equations) k is fixed, deciding if the system S_2 has a non-negative integer solution for $x_1, ..., x_m$ is in PTIME.*

Proof (idea): Transform S_2 to a system S_1 with $(d+1)^k$ variables. Since k, d are fixed, number of variables is fixed. □

Theorem

Let M be an NPDA augmented with 1-reversal counters such that $L(M) \subseteq \{a_1, ..., a_k\}^*$. The problem of deciding, given $n_1, ..., n_k$, whether there exists a string w in $L(M)$ with exactly n_i occurrences of a_i (for $1 \leq i \leq k$) is in PTIME. (Note that the time complexity is a function of $log(n_1) + \cdots + log(n_k)$.)

Proof (idea): Reduce the problem to solving the system S_1 with a fixed number, m, of variables. □

OPEN: Does the above result hold when PTIME is replaced with DLOGSPACE or NLOGSPACE?

Input: M, and Vector $(n_1, \ldots, n_k)$ in Unary

Need the followinmg result [Gurari and Ibarra]:

Lemma

Let m be a fixed positive integer. The emptiness problem for NFAs with m 1-reversal counters is in NLOGSPACE (in the size of M).

Theorem

Let m and k be fixed positive integers. The problem of deciding, given an NFA M with m 1-reversal counters over input alphabet $\Sigma = \{a_1, \ldots, a_k\}$ and $n_1, \ldots, n_k$, whether there exists a string w in $L(M) \subseteq \Sigma^*$ with exactly n_i occurrences of a_i (for $1 \leq i \leq k$) is in NLOGSPACE (hence, also in PTIME). (Note that the complexity is with respect to $|M| + n_1 + \cdots + n_k$.)

Open: What if $n_1, \ldots, n_k$ are in binary?

Proof (idea)

- Given M and $v = (n_1, \ldots, n_k)$, we construct an NFA M_v with k additonal 1-reversal counters, $C_1, \ldots, C_k$. M_v will accept either $\{\epsilon\}$ or $\emptyset$.
- On input different from ϵ, M_v rejects. On input ϵ, M_v first stores n_i into C_i (for each i). Then M_v guesses an input w symbol-by-symbol and simulates M on w, decrementing C_i when it guesses an a_i.
- M_v accepts iff the C_i's become zero and M accepts.
- M_v has size polynomial in $|M| + n_1 + \cdots + n_k$ and can be constructed in DLOGSPACE.
- $L(M_v) \neq \emptyset$ iff there is w such that $P(w) = (n_1, \ldots, n_k)$.
- The result follows from the lemma, since M_v has a fixed number of 1-reversal counters (= $m + k$). □

Input: M, and Vector $(n_1, \cdots, n_k)$ in Binary

The complexity is a function of $|M|$ + $log(n_1) + \cdots + log(n_k)$.

Consider the case when the NFA accepting a bounded language is part of the input. We will show that in this case, the problem becomes NP-complete.

The membership problem for linear sets is the following:

Given: A specification of a linear set $Q(c, \{v_1, \ldots, v_m\})$, where the vectors c, v_1, ..., v_m are k-dimensional vectors of non-negative integers, and a target k-dimensional vector v (all represented in binary).

Question: Are there non-negative integers t_1, ..., t_m such that $v = c + t_1 v_1 + \cdots + t_m v_m$?

Lemma

The linear set membership is NP-hard even when all the components of the vectors in the specifications of the linear sets and the target vector are bounded by 4.

Proof.

The problem without the bounded condition was shown NP-hard by Hyunh using 3-SAT. The proof can be modified so that all components of the vectors in the specifications of the linear sets are bounded by 4. Details are in the proceedings. □

Theorem

The problem of deciding, given an NFA M and $w_1, ..., w_m$ such that $L(M) \subseteq w_1^* ... w_m^*$, where each $w_i \in \{a_1, ..., a_k\}^+$, and $n_1, ..., n_k$, whether there exists a string w in $L(M)$ with exactly n_i occurrences of a_i (for $1 \leq i \leq k$) is NP-complete.

Note that in this version, in addition to the NFA M and the vector $(n_1, n_2, ..., n_k)$, the strings w_1, ..., w_m are given as input. Thus the input size for the problem is $N = |M| + \Sigma_j log_2 n_j + \Sigma_j |w_j|$.

Proof (idea): NP-hardness uses the NP-hardness of the membership problem for linear sets whose vectors have components bounded by 4 (Lemma). For proof that the problem is in NP, see the proceedings. □

The following result was shown in [Esparza] with applications to Petri nets:

- The problem of deciding, given an CFG G over terminal alphabet $\Sigma = \{a_1, \ldots, a_k\}$ and $n_1, \ldots, n_k$, whether there exists a string in $L(M) \subseteq \Sigma^*$ with exactly n_i occurrences of a_i (for $1 \leq i \leq k$) is NP-complete.

When the NFA M accepts a letter-bounded language, we have:

Corollary

The problem of deciding, given an NFA M such that $L(M) \subseteq a_1^* ... a_k^*$ (where the a_i's are distinct symbols), and $n_1, ..., n_k$, whether there exists a string w in $L(M)$ with exactly n_i occurrences of a_i (for $1 \leq i \leq k$) is in PTIME.

- In the case of DFA accepting an unbounded language, the Parikh membership problem is already NP-hard by a reduction from Hamilton cycle problem [To].
- It can be shown that the upper-bound for the general (unbounded) NFA remains in NP [Hyunh].
- Hence, the Parikh membership problem for DFA and NFA are NP-complete.
- When the alphabet size is fixed, the Parikh membership problem for NFA is in PTIME [Kopczynski and To].
- **Open:** When the NFA is augmented by 1-reversal counters, is the Parikh membership problem in NP? (It can be shown that it is in PSPACE.) If the number of counters is one, then it is known to be in NP.

Regular Expressions

Consider the class of regular expressions of the form $w_1^* \cdots w_m^*$ over the alphabet $\{a_1, ..., a_k\}$, where $k, m \geq 1$. Moreover, for $1 \leq i \leq m$ and $1 \leq j \leq k$, the number of occurrence of a_j in w_i is at most 4.

Denote such a regular expression by R_{km} and the language it denotes by $L(R_{km})$.

Theorem

- *When m or k is fixed, the problem of deciding, given a regular expression R_{km} and $n_1, ..., n_k$, whether there is a string w in $L(R_{km})$ with exactly n_i occurrences of a_i (for $1 \leq i \leq k$) is in PTIME.*
- *When m and k are not fixed, the problem of deciding, given a regular expression R_{km} and $n_1, ..., n_k$, whether there is a string w in $L(R_{km})$ with exactly n_i occurrences of a_i (for $1 \leq i \leq k$) is NP-complete.*

Proof

Given a regular expression $R_{km} = w_1^* \cdots w_m^*$:

- First construct another regular expression $R'_{km} = z_1^* \cdots z_m^*$, such that if in w_i, a_j occurs i_j times, $z_i = a_1^{i_1} a_2^{i_2} \cdots a_k^{i_k}$.
- Clearly, there exists a string w in $L(R_{km})$ with exactly n_i occurrences of a_i if and only if there is string z in $L(R'_{km})$ with exactly n_i occurrences of a_i.

- From R'_{km}, construct in polynomial time a system of k linear equations with m variables with non-negative integer coefficients of the form:

$$v_{11}x_1 + v_{12}x_2 + \cdots + v_{1m}x_m = n_1$$

$$.....$$

$$v_{k1}x_1 + v_{k2}x_2 + \cdots + v_{km}x_m = n_k$$

– When m (resp., k) is fixed, then the system can be solved in polynomial time as shown earlier.

– When m and k are not fixed, given regular expression R_{km}, we construct in polynomial time an NFA M accepting the language $L(R_{km})$. NP-completeness follows from the last theorem. □

Semilinear Set Membership Problem

Given: Specification of a semilinear set $S = Q_1 \cup \cdots \cup Q_r$ where each $Q_i \subseteq N^k$ is a linear set, and a vector $v = (n_1, \ldots, n_k)$ in N^k. (The arity of S is k.)

Question: Is v in S?

Theorem

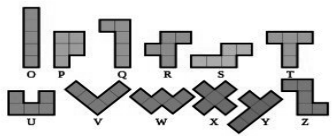
- *For any fixed positive integer m, the membership problem is in PTIME when the number of periodic vectors in each Q_i is at most m.*
- *The membership problem is NP-complete if the number of periodic vectors in each Q_i is not bounded, even when (the arity of S) $k = 1$.*

Theorem

- *The membership problem is NP-complete if the number of periodic vectors in each Q_i is not bounded, even when the components of periodic the vectors in each Q_i have value at most 4 (note that the arity k of S is no longer assumed to be bounded in this case).*
- *For any fixed positive integers k and d, the membership problem is in PTIME when the arity of S is at most k and the components of the periodic vectors in each Q_i have value at most d.*

Polyominoes:

- Polyominoes are a collection of unit squares forming a connected piece in the sense that each square is reachable from any other by going through adjacent squares.
- There are 12 pentominoes listed shown below:



- Many problems have been studied related to covering a board with polyominoes (leaving no holes and with no overlaps).

- Scott has used backtracking to solve the problem of placing one copy each of the 12 different pentominoes in the standard 8 by 8 checker-board with a 2 by 2 hole in the center.
- In a lecture at the University of Pennsylvania in April, 2012, Scott stated a tiling problem:

  Suppose the 12 pentominoes are labeled 1 through 12. Given a sequence of twelve positive integers $(n_1, n_2, ..., n_{12})$ as input, determine if there is a tiling of $5 \times n$ checker-board using exactly n_i copes of tile i (where $n = n_1 + ... + n_{12}$).

  Scott asked whether this problem is in *PTIME*, *NP*-complete or possibly a problem of intermediate complexity (with n_i's given in unary).

- Will show that Scott's problem is in DSPACE (resp., PTIME) when the n_i's are in unary (resp., binary).
- First we code any valid placement P of a $5 \times n$ board using pentominoes using a suitable (finite) alphabet.
- The alphabet is quite large, since there are 63 distinct pentominoes when all possible orientations (rotations and reflections) are counted as distinct, and each square is coded uniquely.
- Call this coding code(P), and define:

  $L_{tile} = \{code(P) \mid P$ is a valid placement of a collection of pentominoes on a $5 \times m$ checkerboard for some $m\}$.

Theorem

L_{tile} is regular.

Proof

Idea: Construct a DFA accepting L_{tile}.
Details:

- The DFA reads the coding from left to right where each symbol is a column of the tiling.
- As each column is read, the DFA remembers the parts of the pentominoes that it has read, and makes sure that the successive columns are consistent.
- For example, if the third row in one of the columns contains the first square of a horizontal pentomino (O), then the DFA will make sure that the next four columns of the third row contain the other squares of this pentomino.

- To check this, the DFA keeps track in a buffer in the finite control all the information about the as yet unseen pieces of the pentominos that are currently being processed.
- The main step performed by the DFA on reading an input (column) is to check that the column is 'consistent' with the buffer data, and update the buffer.
- For example, when the first square of a horizontal pentomino (O) is read, the buffer will contain the remaining 4 squares.
- The accepting states are defined as those in which the buffer is empty.

The resulting DFA accepts the valid codings of pentomino tilings of a $5 \times m$ board. □

To show that Scott's problem is in *DLOGSPACE*, recall the corollary we showed earlier:

Corollary

Let M be an NPDA augmented with 1-reversal counters such that $L(M) \subseteq \{a_1, ..., a_k\}^*$. Let $L_M = \{a_1^{n_1} \cdots a_k^{n_k} \mid$ there exists w in $L(M)$ such that for $1 \leq i \leq k$, w has exactly n_i occurrences of a_i $\}$. Then L_M is in DLOGSPACE and, hence, in PTIME.

Theorem

L_{tile} is in DLOGSPACE.

Proof(idea): Construct from the DFA accepting L_{tile} a DFA that uses 1-reversal counters to "count" and check the n_i's.

Details:

- We map the input $(n_1, n_2, ..., n_{12})$ to the string $a_1^{n_1} ... a_{12}^{n_{12}}$.
- We show that there is a nondeterministic 12 counter machine M (which reverses each counter at most twice) that accepts the language $L = \{a_1^{n_1} ... a_{12}^{n_{12}} \mid$ there is a tiling of $5 \times n$ checker-board using n_j pentominoes of type $j\}$.
- Let M_{tile} be the DFA that accepts the encoded tilings.
- M guesses (symbol by symbol) a string w that represents a tiling of $5 \times n$ board where n is the length of the input string.
- After each symbol is guessed it simulates a single step of the DFA M_{tile} and it also moves the input head exactly once after each symbol is guessed.

- In addition, in finite control, it keeps track of the last five symbols it guessed, and uses it to decrement j for each tile j that was guessed, and that lies completely to the left of the current input position.
- Each counter may be reversed at most twice. This is seen as follows: if a tile of type j is encountered before the input head reads the block a_j, then clearly a symbol will be pushed for each occurrence of tile j in the guessed board, and then when the block of a_j's is reached, a symbol is popped off counter j for each a_j on the input tape.
- This step is repeated until the counter value reaches 0.

- From this point, for each a_j on the input tape, the counter will be incremented and each occurrence of tile type j will result in decrementing the counter.
- When the entire input has been read, if all the counters reach value 0, and the DFA M_{tile} reaches an accepting state, it is clear that the input is a yes instance of the problem and is accepted.
- Thus it is clear that a NFA N_1 with 12 counters each of which reverse at most twice can accept the language L. It is easy to see that N_1 can be simulated by a 24-counter machine NFA N_2 with counters reversing once.
- Using the corollary above, L which N_2 accepts is in *DLOGSPACE*. □

- The previous result showed that Scott's problem is in DLOGSPACE (hence in PTIME) when $(n_1, \ldots, n_{12})$ is represented in unary as a string $a_1^{n_1} \cdots a_{12}^{n_{12}}$.
- When the n_i's are given in binary, it seems unlikely that the problem is in DLOGSPACE (or in NLOGSPACE). However, it is still in PTIME.
- This follows from the following result shown earlier:

Theorem

Let M be an NPDA augmented with 1-reversal counters such that $L(M) \subseteq \{a_1, ..., a_k\}^*$. The problem of deciding, given $n_1, ..., n_k$, whether there exists a string w in $L(M)$ with exactly n_i occurrences of a_i (for $1 \leq i \leq k$) is in PTIME. (Note that the time complexity is a function of $log(n_1) + \cdots + log(n_k)$.)

Generalization

Let $k \geq 1$ and $T = \{t_1, \ldots, t_k\}$ be a finite set of tiles. Let $d \geq 1$ and $P(n_1, \ldots, n_k)$ be a Presburger relation.

For *fixed* k, T, d, P, consider the following problem.

Given: $(n_1, \ldots, n_k)$.

Question: Is there a tiling of a $d \times n$ checker-board using exactly n_i copies of tile i (where $n = n_1 + \cdots + n_k$) and the n_i's satisfy the Presburger relation P?

This problem is in DLOGSPACE when the n_i's are in unary, and in PTIME when the n_i's are in binary.

Known: Suppose d is not fixed. It is NP-complete, given d and n, whether a $d \times n$ board can be tiled, even when only one type of tromino is used [Moore and Robson].

Ginsburg's Problem Concerning Semilinear Sets

In his 1966 book, "The Mathematical Theory of Context-Free Languages", S. Ginsburg posed the following open problem: Find a decision procedure for determining if an arbitrary semilinear set is a finite union of stratified linear sets. This problem is still open.

A linear set $S = (c, V)$ is *stratified* if:

- Every $v \in V$ has at most two nonzero components, and
- There exist no integers i, j, k, l with $1 \leq i < j < k < l \leq n$ and no vectors $u, v \in V$ such that none of $u[i]$, $v[j]$, $u[k]$, $v[l]$ is zero.

It turns out that Ginsburg's problem is equivalent to synchronizability of multitape automata.

- Let M be n-tape automaton of a given type (e.g., NPDA, NFA, etc.), with a one-way read-only head per tape and a right end marker \$ on each tape. As usual, an n-tuple $x = (x_1, \ldots, x_n)$ is accepted if M eventually reaches the configuration where all n heads are on \$ in an accepting state.
- M is synchronized (or aligned) if for every n-tuple $x = (x_1, \ldots, x_n)$ that is accepted, there is a computation on x such that at any time during the computation, all heads, except those that have reached the end marker, are on the same position (i.e., aligned). When a head reaches the marker, it can no longer move.
- Synchonization have been studied before.

The following were recently shown in [Ibarra and Seki]:

Theorem

Ginsburg's problem is equivalent to deciding for an arbitrary n-tape NFA (resp., NPDA) M accepting $L(M) \subseteq a_1^* \times \cdots \times a_n^*$ (where $n \geq 1$ and $a_1, \ldots, a_n$ are symbols) whether there exists a synchronized n-tape NPDA M' equivalet to M (i.e., accepts $L(M)$).

Theorem

It is decidable, given an n-tape NFA M whose inputs come from $B_1 \times \cdots \times B_n$ (where each $B_i \subseteq w_1^* \cdots w_k^*$ for some $k \geq 1$ and nonnull words $w_1, \ldots, w_k$), whether there exists a synchronized n-tape NFA M' equivalent to M.

The second result is not true if the B_i's are not bounded, since the problem becomes undecidable.

Conclusion

- We studied some problems related to testing membership for regular and other languages, where we assume that the string is specified by its Parikh vector.
- We studied four versions of this problem in which the Parikh vector is specified in unary or binary, and whether the machine M is fixed or part of the input.
- We showed that the problem can be solved in PTIME in the unary case when the fixed language comes from a very broad class (namely, the class of languages accepted by NPDA augmented by reversal-bounded counter machines).
- When the input vector is specified in binary, the complexity of the problem is already NP-hard in the case of word-bounded regular languages.

- One of the interesting fact we found is the difference between the letter-bounded regular languages and word-bounded regular languages. The membership problem for the former case is in PTIME while in the latter case it is NP-complete.
- Our results imply that a classical tiling problem is in DLOGSPACE (PTIME) when the number of tiles of various types are specified in unary (binary) notation.
- We also studied the complexity of membership for regular expressions and semilinear sets.
- Finally, we gave a charactization of Ginsburg's problem in terms of synchronization of multitape NPDAs.

References

- B. S. Baker and R. V. Book. Reversal-bounded multipushdown machines. *J. Comput. Syst. Sci.* 8(3): 315-332 (1974).
- J. Esparza. Petri nets, commutative context-free grammars and basic parallel processes. *Fundamenta Informaticae* 30 (1997) 23-41.
- S. Ginsburg. *The Mathematical Theory of Context-Free Languages*. McGraw-Hill, New York, 1966.
- S. W. Golomb. Polyominoes (2nd edition). Princeton University Press. ISBN 0-691-02444-8. (1994).

- E. M. Gurari and O. H. Ibarra. The complexity of decision problems for finite-turn multicounter machines. *J. Computer and System Sciences* 22, 220-229 (1981)
- T.-D. Hyunh. The complexity of semilinear sets. *Elektr. Inform.-verarbeitung and Kybern.* 291-338, 6 (1982).
- T.-D. Hyunh. Commutative Grammars: The complexity of uniform word problems *Information and Control* 57, 21-39, (1983).
- O. H. Ibarra. Reversal-bounded multicounter machines and their decision problems. *J. Assoc. Comput. Mach.* 25, 116-133, 1978.

- O. H. Ibarra and B. Ravikumar. On sparseness and ambiguity for acceptors and transducers. *Symposium on Theoretical Aspects of Computer Science, Proceedings, Lecture Notes in Computer Science*, 1986.
- O. H. Ibarra and B. Ravikumar. On bounded languages and reversal-bounded automata. *LATA 2013, Proceedings, Lecture Notes in Computer Science*, 2013, 359-370.
- O. H. Ibarra and S. Seki. Characterizations of bounded semilinear languages by one-way and two-way deterministic machines. *IJFCS* 23, 1291-1306, 2012.
- O. H. Ibarra and H. Yen. On the Containment and equivalence problems for two-way transducers *Theoretical Computer Science* Vol. 429, 155-163, 2012.

- E. Kopczynski and A. W. To. Parikh images of grammars: complexity and applications. *Proc. of 25th Annual IEEE Logic in Computer Science*, 80-89, 2010.
- G. J. Lavado, G. Pighizzini and S. Seki. Converting nondeterministic automata and context-free grammars into Parikh equivalent one-way and two-way automata. *Developments in Language Theory, Proceedings, Lecture Notes in Computer Science*, 7410, pages 284-295, 2012.
- T. Cormen, C. Leiserson, R. Rivest and C. Stein. *Introduction to Algorithms*, 2nd Edition. *MIT Press, 2001.*
- H. W. Lenstra, Jr. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8, 583-548, 1983.

- G. S. Lueker. Two NP-complete problems in nonnegative integer programming. *Report No. 178, Computer Science Laboratory, Princeton University*, 1975.
- R. J. Parikh. On context-free languages. *J. Assoc. Comput. Mach.*, 13, 570-581, 1966.
- D. S. Scott. Programming a combinatorial puzzle. Technical Report No. 1, Department of Electrical Engineering, Princeton University (1958).
- A. W. To. Parikh images of regular languages: complexity and applications (unpublished manuscript).