# COMPLETENESS RESULTS FOR CONFLICT-FREE VECTOR REPLACEMENT SYSTEMS

Rodney R. Howell and Louis E. Rosier

Department of Computer Sciences
The University of Texas at Austin
Austin, Texas 78712-1188

TR-86-21 September 1986

# Completeness Results for Conflict-Free Vector Replacement Systems

Rodney R. Howell and Louis E. Rosier

Department of Computer Sciences
University of Texas at Austin
Austin, Texas 78712-1188

9 September 1986

## Abstract

In this paper, we give completeness results for the reachability, containment, equivalence, liveness, and various fair nontermination problems for conflict-free vector replacement systems (VRSs). We first give an NP algorithm for deciding reachability. Since Jones, Landweber, and Lien have shown this problem to be NP-hard, it follows that the problem is NP-complete. Next, we show as our main result that the containment and equivalence problems are $\Pi_2^P$-complete, where $\Pi_2^P$ is the set of all languages whose complements are in the second level of the polynomial-time hierarchy. In showing the upper bound, we first show that the reachability set has a semilinear set (SLS) representation that is exponential in the size of the problem description, but which has a certain degree of symmetry. We are then able to modify a proof given by Huynh (concerning SLSs) to complete our upper bound proof. We then show that the liveness problem is PTIME-complete and has an $O(n^{1.5})$ upper bound. Finally, we show various fair nontermination problems to be complete for NLOGSPACE, PTIME, and NP.

## 1. Introduction

The reachability, containment, and equivalence problems for vector replacement systems (VRSs) (or equivalently vector addition systems (VASs), vector addition systems with states (VASSs), or Petri nets) are the subject of many unanswered questions concerning computational complexity. The containment and equivalence problems are, in general, undecidable [1, 10]. However, the reachability problem is decidable [21, 28], and, for classes of VRSs (VASs, VASSs, Petri nets) whose reachability sets are effectively computable semilinear sets (SLSs), so are the containment and equivalence problems. Classes whose reachability sets are effectively computable SLSs include finite VRSs [20], 5-dimensional VRSs (or, equivalently, 2-dimensional VASSs) [11], conflict-free VRSs [6], persistent VRSs [9, 23, 27, 31], and regular VRSs [8, 39]. The best known lower bound for the general reachability problem is exponential space [26]. For finite VRSs, tight non-primitive recursive upper and lower bounds have been shown for the containment and equivalence problems [5, 12, 30, 32]. For 2-dimensional VASSs, the reachability, containment, and equivalence problems can be solved in DTIME($2^{2^{c^*n}}$) [12]. The reachability problem for conflict-free VRSs has been shown to be NP-hard [19]. The perhaps best studied class is that of symmetric VRSs. For this class, the reachability and equivalence problems have been shown to be exponential space complete [3, 16, 29]. Few other complexity results appear to be known concerning these problems.

In this paper, we show completeness results concerning conflict-free VRSs for these three problems, as well as for liveness and several fairness problems. Conflict-free VASs were introduced by Crespi-Reghizzi and Mandrioli [6], who showed the reachability problem to be decidable for this class. Conflict-free Petri nets were later introduced by Landweber and Robertson [23], who showed that the reachability set of a conflict-free Petri net was semilinear, and that the boundedness problem for this class could be solved in exponential time. Howell, Rosier, and Yen [13] then introduced conflict-free VRSs as a formalism that contains both conflict-free VASs and conflict-free Petri nets as special cases, but for which the boundedness problem retains the same computational complexity; i.e., the boundedness problem was shown to be PTIME-complete for all three classes. (As was pointed out in [13], even though there are translations between the three classes, these translations do not always preserve sharp complexity bounds.) In this paper, we follow the precedent established in [13] of showing upper bounds for VRSs, the most general of the three models, and showing lower bounds for systems which satisfy the definitions of all three models.

The main result of this paper is to show the equivalence and containment problems for conflict-free VRSs to be $\Pi_2^P$-complete, where $\Pi_2^P$ is the second level of the polynomial-time hierarchy (see Stockmeyer [36]). In showing this, we use a strategy developed by Huynh [17] in showing the equivalence problem for semilinear sets to be in

$\Pi_2^P$. Given this result and the fact that conflict-free VRSs have semilinear reachability sets [23], one might attempt to solve the problem by translating the VRSs to SLSs and applying Huynh's results directly. However, it can be shown that such a translation must be exponential. Hence, we prove additional properties concerning the resulting SLS representations that enable us to obtain our results via a modified version of Huynh's proof. Now in his proof, Huynh used the fact that the membership problem for semilinear sets is NP-complete [15]. We therefore first show that the reachability problem for conflict-free VRSs is in NP. (Since Jones, Landweber, and Lien [19] have shown the problem to be NP-hard, it follows that the problem is NP-complete.) In order to show this, we give some properties of conflict-free VRSs that allow us to show that there is an instance of integer linear programming that has a solution iff a given reachability problem has a solution; furthermore, this instance of integer linear programming can be "guessed" in polynomial time. Our next step is to give a SLS representation of the reachability set. We have already mentioned that this representation is exponential in the size of the problem description. On the other hand, we are able to show a certain amount of symmetry in the SLS representation. It is this symmetry that allows us to alter Huynh's proof to give our upper bound. Finally, we show a matching lower bound to complete the proof of our main result.

The remainder of our completeness results have to do with the liveness and fair nontermination problems. The concept of liveness for Petri nets was introduced by Jones, Landweber, and Lien [19]. It follows from results in [19] that the liveness problem for conflict-free Petri nets can be solved in NP. We are able to show an $O(n^{1.5})$ upper bound for the liveness problem for conflict-free VRSs, and to show the problem to be PTIME-complete. Various notions of fairness for Petri nets were introduced by Carstensen and Valk [4]. In addition, we adapt to the VRS model several notions of fairness introduced by Lehman, Pnueli, and Stavi [24]. We are able to show the fair nontermination problems for each of these definitions of fairness to be complete for either NLOGSPACE, PTIME, or NP. Since a number of these problems have efficient solutions, we conjecture that the algorithms may be useful in the verification of parallel systems which can be modelled using conflict-free VRSs.

The remainder of the paper is organized as follows. In Section 2, definitions of the terminology used in this paper are given. In Section 3, we give our results concerning the reachability, containment, and equivalence problems. Finally, in Section 4, we give our results concerning the liveness and fair nontermination problems.

## 2. Definitions

Let Z (N, R) denote the set of integers (nonnegative integers, rational numbers, respectively), and let $Z^k$ ($N^k$, $R^k$) be the set of vectors of k integers (nonnegative integers, rational numbers), and $Z^{k\times m}$ ($N^{k\times m}$, $R^{k\times m}$) be the set of $k\times m$ matrices of integers (nonnegative integers, rational numbers). For a vector $v\in Z^k$, let $v(i)$, $1\leq i\leq k$, denote the $i^{th}$ component of v. For a matrix $V\in Z^{k\times m}$, let $V(i,j)$, $1\leq i\leq k$, $1\leq j\leq m$, denote the element in the $i^{th}$ row and $j^{th}$ column of V, and let $v_j$ denote the $j^{th}$ column of V. For a given value of k, let $\mathbf{0}$ in $Z^k$ denote the vector of k zeros (i.e., $\mathbf{0}(i)=0$ for $i=1,...,k$). Now given vectors u,v, and w in $Z^k$ we say:

- $v=w$ iff $v(i)=w(i)$ for $i=1,...,k$;
- $v\geq w$ iff $v(i)\geq w(i)$ for $i=1,...,k$;
- $v>w$ iff $v\geq w$ and $v\neq w$;
- $u=v+w$ iff $u(i)=v(i)+w(i)$ for $i=1,...,k$.

A $k\times m$ *vector replacement system* (VRS), is a triple $(v_0,U,V)$, where $v_0\in N^k$, $U\in N^{k\times m}$, and $V\in Z^{k\times m}$, such that for any i,j, $1\leq i\leq k$, $1\leq j\leq m$, $U(i,j)+V(i,j)\geq 0$. $v_0$ is known as the *start vector*, U is known as the *check matrix*, and V is known as the *addition matrix*. A column $u_j$ of U is called a *check vector*, and a column $v_j$ of V is called an *addition rule*. For any $x\in N^k$, we say addition rule $v_j$ is *enabled* at x iff $x\geq u_j$. A sequence $<y_1,...,y_n>$ of rules in V is *enabled* at a vector x iff for each j, $1\leq j\leq n$, $y_j$ is enabled at $x+y_1+\cdots+y_{j-1}$. The *reachability set* of the VRS $\mathcal{V}=(v_0,U,V)$, denoted by $R(v_0,U,V)$ (or $R(\mathcal{V})$), is the set of all vectors z, such that $z=v_0+y_1+\cdots+y_n$ for some $n\geq 0$, where each $y_j$ $(1\leq j\leq n)$ is a column of $V$, and $<y_1,...,y_n>$ is enabled at $v_0$. Let $\sigma=<w_0,...,w_t>$ be a sequence of vectors in $N^k$. If $w_0=v_0$, and for every r, $1\leq r\leq t$, there is a j such that $w_r=w_{r-1}+v_j$ and $w_{r-1}\geq u_j$, then we say $<w_0,\ldots,w_t>$ is a *path* in $(v_0,U,V)$. If there exist r and s, $1 \leq r < s \leq t$, such that $w_r \leq w_s$ ($w_r < w_s$), then we say that $\pi = <w_r,\ldots,w_s>$ is a *loop* (*positive loop*), and that $\pi$ is *enabled* at $w_{r-1}$. Let $\Psi$ denote the *Parikh mapping*, such that if $\theta$ is a sequence of rules in V, then $\Psi(\theta) \in N^m$, and $\Psi(\theta)(j)$ is the number of occurrences of $v_j$ in $\theta$. Let $\delta(\theta)$ denote the displacement of $\theta$. We also define an extended Parikh mapping (see also [23]) $\Psi^+$ such that $\Psi^+(\theta) = <\Psi(\theta),\delta(\theta)>$.

A VRS $(v_0,U,V)$ is said to be *conflict-free* iff (1) no number in U is greater than 1; (2) no number in V is less than -1; (3) no row in V has more than one -1; and (4) if $V(i,j)=-1$, then $U(i,j) = 1$, and all other elements in row i of U are 0. For a given $k \times m$ addition matrix V, the *minimal check matrix* is a $k \times m$ matrix U in which $U(i,j)=1$ if $V(i,j)=-1$, and $U(i,j)=0$ otherwise. It is easy to see that the set of $k \times m$ conflict-free VRSs with minimal check matrices is equivalent to the set of $k \times m$ conflict-free VASs (see [6]). Furthermore, there is an obvious translation from a conflict-free Petri net (see [23]) with k places and m transitions to a $k \times m$ conflict-free VRS whose addition rules and check vectors have no elements larger than 1. Thus, our definition is general enough to include both previous definitions. In addition, all lower bounds shown in this paper are shown using VRSs having minimal check matrices and no elements larger than 1. Thus, all of our completeness results hold for conflict-free VRSs, conflict-free VASs, and conflict-free Petri nets.

The *reachability problem* for VRSs is to determine, for a given VRS $\mathcal{V}$ and a vector v, whether $v \in R(\mathcal{V})$. The *containment* and *equivalence problems* are to determine, for two given VRSs $\mathcal{V}$ and $\mathcal{V}'$, whether $R(\mathcal{V}) \subseteq R(\mathcal{V}')$ and whether $R(\mathcal{V}) = R(\mathcal{V}')$, respectively. A VRS $\mathcal{V}$ is said to be *bounded* iff for each row i, there is a constant c such that if $v \in R(\mathcal{V})$, then $v(i) < c$. The *boundedness problem* for VRSs is the problem of determining whether a given VRS is bounded. An addition rule $v_j \in V$ is said to be *live* in $(v_0,U,V)$ if for any $w \in R(v_0,U,V)$, there is a path $\sigma$ in $(w,U,V)$ that enables $v_j$. The *transition liveness problem* for VRSs is to determine, for a given VRS $(v_0,U,V)$ and an addition rule $v_j \in V$, whether $v_j$ is live in $(v_0,U,V)$. The VRS $(v_0,U,V)$ is said to be *live* if every transition $v_j \in V$ is live in $(v_0,U,V)$. The *liveness problem* for VRSs is to determine whether a given VRS is live.

The remaining problems studied in this paper have to do with various notions of fairness. The first three types of fairness we consider were introduced by Lehman, Pnueli, and Stavi [24]. Let $\sigma$ be an infinite path in $(v_0,U,V)$. $\sigma$ is said to be *impartial* if every addition rule $v_j \in V$ is executed infinitely often. $\sigma$ is said to be *just* if every addition rule $v_j \in V$ that is enabled continuously after some point in $\sigma$ is executed infinitely often. $\sigma$ is said to be *fair* if every addition rule $v_j \in V$ that is enabled infinitely often in $\sigma$ is executed infinitely often. The remaining definitions of fairness come from Landweber [22] and Carstensen and Valk [4]. Let $\mathcal{A}$ be a finite set of finite nonempty subsets of $N^k$. $\sigma$ is said to be

- *1-fair* for $\mathcal{A}$ if there is an $A \in \mathcal{A}$ such that some vector reached by $\sigma$ is in A.
- *1′-fair* for $\mathcal{A}$ if there is an $A \in \mathcal{A}$ such that every vector reached by $\sigma$ is in A.
- *2-fair* for $\mathcal{A}$ if there is an $A \in \mathcal{A}$ such that some vector reached infinitely often by $\sigma$ is in A.
- *2′-fair* for $\mathcal{A}$ if there is an $A \in \mathcal{A}$ such that every vector reached infinitely often by $\sigma$ is in A.
- *3-fair* for $\mathcal{A}$ if the set of vectors reached infinitely often by $\sigma$ is an element of $\mathcal{A}$.
- *3′-fair* for $\mathcal{A}$ if there is an $A \in \mathcal{A}$ such that every vector in A is reached infinitely often by $\sigma$.

We refer to these six types of fairness collectively as *i-fairness*, where i is understood to be an element of $\{1,1',2,2',3,3'\}$. The *impartial (just, fair, i-fair) nontermination problem* is the problem of determining whether there is an infinite impartial (just, fair, i-fair, respectively) path in a given VRS for a given set $\mathcal{A}$ (if applicable).

Part of our analysis involves notions from the theories of semilinear sets and convex polyhedra. We define here the terms used in this paper. For a detailed treatment of the theory of convex polyhedra, see Stoer and Witzgall [37].

For any vector $v_0 \in N^k$ and any finite set $P(=\{v_1,...,v_m\}) \subseteq N^k$, the set $\mathcal{L}(v_0,P)=\{x : \exists k_1,...,k_m \in N^k$ and $x=v_0+\sum_{i=1}^{m} k_i v_i\}$ is called the *linear set* with *base* $v_0$ over the set of *periods* P. A finite union of linear sets is called a *semilinear set* (SLS for short). The *cone* generated by $v_0$ and P, denoted by $\mathcal{C}(v_0,P)$, is the set $\{x : \exists k_1,...,k_m \in R$, $k_1,...,k_m \geq 0$, and $x=v_0+\sum_{i=1}^{m} k_i v_i\}$. If $v_0=0$ and we restrict $k_1,...,k_m$ in the above set so that $\sum_{i=1}^{m} k_i=1$, then that set is called the *convex hull* of P, and is denoted conv(P). Let $A \in Z^{k\times m}$ and $b \in Z^k$. If $A \neq \mathbf{0}$, then the solution set of the linear equation $A_j x^T = b(j)$ over $x \in R^m$ is a *hyperplane*, and the solution set of the linear inequality $A_j x^T \leq b(j)$ over $x \in R^m$ is a *halfspace*. Finally, the solution set of $Ax \leq b$ is called a *convex polyhedron* (or polyhedron, for short).

## 3. Reachability, Containment, and Equivalence

The first problem we would like to examine is the reachability problem. Jones, Landweber, and Lien [19] have shown this problem to be NP-hard. Although the problem is known to be decidable [6], no upper bound on its complexity has yet been shown. In order to tighten this gap, we will show the problem to be NP-complete. Our strategy is to guess an instance of integer linear programming whose solutions give Parikh maps of sequences of addition rules that lead to the desired vector. The following two lemmas will give sufficient conditions to guarantee that for every solution $\bar{x}$, there is an enabled sequence $\theta$ such that $\Psi(\theta)=\bar{x}$.

**Lemma 3.1 (from [13]):** For any $k\times m$ conflict-free VRS $\mathcal{V}=(v_0,U,V)$ that is described by n bits, we can construct in time $O(n^{1.5})$ a path $\sigma$ in which no rule in V is used more than once, such that if some rule $v_r$ is not used in $\sigma$, then there is no path in which $v_r$ is used.

**Proof:** We construct $\sigma$ as follows. First, we execute all rules enabled at $v_0$. Then we repeatedly cycle through U, executing all those rules which are enabled but have not yet been executed. We continue until a complete pass is made through U, during which no position increases in value. (Note that this is a sufficient condition to conclude that no new rules are enabled.) Clearly, no more than m+1 passes are made through U. On each pass except the last, there is at least one rule (say $v_j$) enabled that was not enabled the previous pass; i.e., some position (say p) which was zero the previous pass is now positive. Furthermore, since $\mathcal{V}$ is conflict-free, if some rule subtracts from position p, that rule must be $v_j$. Therefore, position p must have never previously been positive. Thus, on each pass except the last some position becomes positive for the first time, so the number of passes is no more than $\min(k,m)+1=O(n^{0.5})$. Therefore, the entire procedure operates in time $O(n^{1.5})$.

Now suppose there is a path $\sigma'$ using rules not in $\sigma$. Let $v_r$ be the first such rule executed in $\sigma'$. Then all rules used before $v_r$ in $\sigma'$ are used in $\sigma$. Since $v_r$ is not executed in $\sigma$, no position from which $v_r$ subtracts ever decreases in value in $\sigma$; hence, these positions are at least as large as they are at the point at which $v_r$ is executed in $\sigma'$. Then $v_r$ is enabled by $\sigma$, a contradiction. Therefore, if $v_r$ is not used in $\sigma$, then there is no path in which $v_r$ is used. $\square$

**Lemma 3.2:** Let $(v_0,U,V)$ be a $k\times m$ conflict-free VRS, and let $\theta$ be an arbitrary sequence of rules from V. If every rule in $\theta$ appears in some path that uses only rules from $\theta$, and if $\delta(\theta)+v_0 \geq \mathbf{0}$, then there is some sequence $\theta'$ enabled at $v_0$ such that $\Psi(\theta')=\Psi(\theta)$.

**Proof:** We will construct a path $\sigma$ consisting of a sequence of n *segments*, $\sigma_1,\ldots,\sigma_n$, where n is the maximum number of times any rule appears in $\theta$. Each segment will be a sequence containing at most one occurrence of each rule in $\theta$. Furthermore, $\sigma$ will be such that if some segment contains no occurrence of some rule, then no succeeding segment will contain an occurrence of that rule. Now, if we restrict our VRS to contain only the rules used in $\theta$, then from Lemma 3.1, some sequence containing every rule in $\theta$ exactly once is valid at $v_0$. This sequence will be segment $\sigma_1$ of $\sigma$. We construct segment $\sigma_r$, $2\leq r\leq n$, as follows: while there is an enabled rule $v_j$ which occurs at least r times in $\theta$ and has not yet been used in $\sigma_r$, execute $v_j$. We claim that according to this construction, segment $\sigma_r$ $(1\leq r\leq n)$ uses exactly one occurrence of each rule that appears at least r times in $\theta$. Suppose, to the contrary, that at some point in the construction of $\sigma_r$, there are no enabled rules in the nonempty set S of rules that appear at least r times in $\theta$ but which have not been used in $\sigma_r$. Without loss of generality, assume $\sigma_r$ is the first segment for which this happens. Let $v_j$ be the first rule used in $\sigma$ that also appears in S, and let w be the vector produced by the first r segments of $\sigma$. Now there must exist an i, $1 \leq i \leq k$, such that w(i)=0 and U(i,j)=1. If V(i,j)$\neq$-1, then from the definition of conflict-freedom, no rule can subtract from position i, so position i would have had to have been 0 throughout $\sigma$. But this would mean $v_j$ could not have been executed even once--a contradiction. Therefore, V(i,j)=-1. Since $\delta(\theta)+v_0 \geq \mathbf{0}$, some rule $v_{j'}$ used in $\sigma_r$ or occurring in S must add to position i. Since $v_j$ is the only rule that can subtract from position i, $v_{j'}$ can not have been executed since the last time $v_j$ was executed; otherwise, $v_j$ would be enabled by w. Thus, $v_{j'} \in S$. Now clearly, $v_j$ and $v_{j'}$ have been executed the same number of times in the first r segments, so $v_0(i) \leq w(i)=0$. Since $v_j$ is the first rule from S used in $\sigma$, some other rule (not $v_{j'}$) which adds to position i must have been executed before $v_j$ was first used. But this forces w(i)>0--a contradiction. Therefore, segment $\sigma_r$ contains exactly one occurrence of each rule that appears at least r times in $\theta$, for $1 \leq r \leq n$. Thus, the sequence $\theta'$ of rules comprising $\sigma$ satisfies the lemma. □

The following is a corollary to the proof of Lemma 3.2; it will be used in obtaining later results.

**Corollary 3.1:** If $\theta$ is a sequence of rules enabled at v such that $\delta(\theta) \geq \mathbf{0}$, then there exists a vector $v' \leq v$ with no element larger than 1 and a sequence of rules $\theta'$ with $\Psi(\theta)=\Psi(\theta')$ such that $\theta'$ is enabled at v'.

**Proof:** Let v'(i)=0 if v(i)=0, v'(i)=1 otherwise. Consider the first segment constructed in the proof of Lemma 3.2. Since each rule used in $\theta$ is used exactly once in this segment, no more than 1 is subtracted from any position during the execution of the segment. Thus, this segment is clearly enabled by v'. Now from Lemma 3.2, there is some sequence $\theta'$ enabled at v' such that $\Psi(\theta)=\Psi(\theta')$. □

We are now ready to show the reachability problem to be NP-complete. Recall that the problem was shown to be NP-hard in [19]. An inspection of the construction used in that proof reveals that it holds for both conflict-free Petri nets and conflict-free VASs. Hence, we only need to show the upper bound.

**Theorem 3.1:** The reachability problem for conflict-free VRSs is NP-complete.

**Proof:** Let $(v_0,U,V)$ be a $k\times m$ conflict-free VRS, and let $w\in N^k$ be an arbitrary vector. Our algorithm assumes the existence of some path that results in w, and guesses the set of rules used in that path. It then verifies whether there is some path which uses exactly this set of rules. By Lemma 3.1, this can be verified in polynomial time. Let the set of guessed rules be the $k\times n$ matrix $V'$. Our algorithm now verifies that there is some $x\in N^n$, $x(i)\geq 1$ for $1\leq i\leq n$, such that $V'x+v_0=w$. From Borosh and Treybig [2], this can be verified in NP. Now from Lemma 3.2, if such an x exists, then $w\in R(v_0,U,V)$. □

We now turn to the containment and equivalence problems. We will show that these problems are $\Pi_2^P$-complete, where $\Pi_2^P$ is the set of complements of all languages that can be recognized by a polynomial-time-bounded nondeterministic Turing machine with an NP oracle (see Stockmeyer [36]). In showing the upper bound, we follow a technique used first by Huynh [17] (see also [12]). In [17], Huynh gave a proof that the containment and equivalence problems for semilinear sets are in $\Pi_2^P$. Landweber and Robertson [23] have shown that the reachability set of a conflict-free Petri net is semilinear; it is easy to verify that this also holds for VRSs. In what follows, we give an upper bound on the size of the SLS representation of the reachability set. In particular, we give an SLS representation in which no integer is larger than $(c*k*m*n)^{d*k*m}$, where k and m are the dimensions of the VRS, n is the largest absolute value of any integer in the VRS, and c and d are fixed constants independent of k, m, and n. Now the technique used in [17] is to show that if the two SLSs are not equal, then there is a "small" witness to that fact. Unfortunately, applying our derived bounds to the result in [17] yields a bound of $O((k*m*n)^{(k*m)^{c'*k^2*m}})$ for the largest integer in the smallest witness. This is clearly too large to guess in polynomial time. Furthermore, we cannot improve our bounds enough to make a direct application of Huynh's results work. To see this, observe that there is a bounded $k \times (k-1)$ conflict-free VRS with start vector $(1,0,\ldots,0)$ which has, for any position i, $2 \leq i \leq k$, a rule which will subtract 1 from position i-1 and add 2 to position i. The reachability set of this VRS has at least $2^k$ bases, and even for SLS representations of this size, Huynh's results yield a bound of $O((k^*m^*n)^{(k^*m)^{c'^*k}})$. In [12], a variation of the proof in [17] was given in which a small enough bound was placed on the sizes of the periods to allow some degree of improvement to be made. However, even if a bound of n could be placed on the largest integer in any period, this proof does not not yield a polynomial bound on the binary representation of the smallest witness. What we are able to do, however, is to give an SLS representation with a high degree of symmetry. It is this symmetry, together with our bound on the size of the SLS representation, that allows us to alter the proof in [17] to give us our tight bound on the size of the smallest witness. The following lemma gives the SLS representation of the reachability set of an arbitrary conflict-free VRS.

**Lemma 3.3:** Let $(v_0,U,V)$ be a $k \times m$ conflict-free VRS in which n is the largest absolute value of any integer. Then there exist constants $c_1$, $c_2$, $d_1$, and $d_2$, independent of k, m, and n, such that $R(v_0,U,V)=\cup_{v \in B}\mathcal{L}(v,P_v)$, where B is the set of all reachable vectors with no element larger than $(c_1{}^*k^*m^*n)^{c_2{}^*k^*m}$, and $P_v$ is the set of all displacements of loops enabled at v such that if $p \in P_v$, then

1. p has no element larger than $(d_1{}^*k^*m^*n)^{d_2{}^*k^*m}$, and

2. if $v(i)=0$, then $p(i)=0$.

**Proof:**

$(\cup_{v \in B}\mathcal{L}(v,P_v) \subseteq R(v_0,U,V))$: Clear.

$(R(v_0,U,V) \subseteq \cup_{v \in B}\mathcal{L}(v,P_v))$: Let $G=\{\theta_1,\ldots,\theta_r\}$ be the set of all nonempty sequences

$\theta$ of rules in V for which $\Psi^+(\theta)\geq\mathbf{0}$, and there is no nonempty sequence $\theta'$ such that $\mathbf{0}\leq\Psi^+(\theta')<\Psi^+(\theta)$. (The fact that G is finite follows from results in [20]). For each $G_s=\{\theta_1^s,\ldots,\theta_{r_s}^s\}\subseteq G$, let $W'_s$ be the set of vectors w such that (1) there exists a sequence $\theta$ of rules in V which, when applied to $v_0$, yields w, (2) all sequences in $G_s$ are enabled at w, and (3) $<\Psi(\theta),w>$ is minimal over the set of all $<\Psi(\theta),w>$ satisfying conditions (1) and (2). Also, let $W_s$ be defined in the same manner as $W'_s$, except that condition (2) should read "for each sequence $\theta_i^s\in G_s$, there is a sequence $\theta_j\in G$ such that $\delta(\theta_i^s)=\delta(\theta_j)$ and $\theta_j$ is enabled at w." From Theorem 4.1 in [23], $R(v_0,U,V)=\cup_{G_s\subseteq G}\cup_{w\in W'_s}\mathcal{L}(w,\delta(G_s))$, where $\delta(G_s)$ denotes the set of all displacements of loops in $G_s$. (The proof in [23] deals with persistent Petri nets; however, the same proof technique works for conflict-free VRSs.) An inspection of the proof of Theorem 4.1 in [23] reveals that $R(v_0,U,V)=\cup_{G_s\subseteq G}\cup_{w\in W_s}\mathcal{L}(w,\delta(G_s))$ as well. We will show that $\cup_{G_s\subseteq G}\cup_{w\in W_s}\mathcal{L}(w,\delta(G_s))\subseteq\cup_{v\in B}\mathcal{L}(v,P_v)$.

Let $y\in\cup_{G_s\subseteq G}\cup_{w\in W_s}\mathcal{L}(w,\delta(G_s))$. In particular, let $y=w+\sum_{j=1}^{r_s}a_j\delta(\theta_j^s)$, where $G_s=\{\theta_1^s,\ldots,\theta_{r_s}^s\}\subseteq G$, $w\in W_s$, $a_j\geq0$. We will first give a bound for $\delta(\theta)$ for each $\theta\in G$. Let $x=\Psi(\theta)$, and $v=\delta(\theta)$. Then the following system of Diophantine inequalities holds (recall that Vx gives the displacement of $\theta$):

$Vx\geq\mathbf{0}$
$Vx=v$
$x(j)\geq1$ if $v_j$ is used in $\theta$, $1\leq j\leq m$
$x(j)=0$ if $v_j$ is not used in $\theta$, $1\leq j\leq m$

We now claim that $\Psi^+(\theta)$ is a minimal solution of the above system over the variables $<x(1),\ldots,x(m),v(1),\ldots,v(k)>$. To see this, observe that any smaller solution than $\Psi(\theta)$ would describe a sequence $\theta'$ using the same rules as $\theta$ such that $\Psi^+(\theta')<\Psi^+(\theta)$--a contradiction of the definition of G. From results of von zur Gathen and Sieveking [40] and Huynh [15], all variables in any minimal solution are bounded above by $(d_1*k*m*n)^{d_2*k*m}$ for some constants $d_1$ and $d_2$. Thus, $\delta(\theta)(i)\leq(d_1*k*m*n)^{d_2*k*m}$, $1\leq i\leq k$.

We will now show a bound for w. Recall that there is some sequence $\theta$ that, when applied to $v_0$, yields w. Let $v' \leq w$ be some minimal vector such that for any sequence $\theta_i^s \in G_s$, there is a sequence $\theta_j \in G$ such that $\delta(\theta_i^s)=\delta(\theta_j)$ and $\theta_j$ is enabled at $v'$. By Corollary 3.1, no element in $v'$ is larger than 1. If x and v are again defined as above, the following system of Diophantine inequalities holds (recall that $Vx+v_0=w$):

$Vx+v_0 \geq v'$
$Vx=v$
$x(j) \geq 1$ if $v_j$ is used in $\theta$, $1 \leq j \leq m$
$x(j)=0$ if $v_j$ is not used in $\theta$, $1 \leq j \leq m$

Now clearly, all solutions to the above system over the variables $<x(1),\ldots,x(m),v(1),\ldots,v(k)>$ are pairs $<\Psi(\theta'),w'-v_0>$ such that $w'$ satisfies conditions (1) and (2) of the definition of $W_s$. Since $w \in W_s$ and $<\Psi(\theta),w-v_0>$ is clearly a solution of the above system, in order to satisfy condition (3) of the definition of $W_s$, $<\Psi(\theta),w-v_0>$ must be a minimal solution. Thus, $w(i) \leq (d_1{*}k{*}m{*}n)^{d_2{*}k{*}m}$, $1 \leq i \leq k$.

We are now ready to show $y \in \cup_{v \in B} \mathcal{L}(v,P_v)$. Suppose for some i, $y(i) \neq 0$ and $w(i)=0$. Since $y=w+\sum_{j=1}^{r_s} a_j \delta(\theta_j^s)$, there must be some j, $1 \leq j \leq r_s$, such that $a_j \neq 0$ and $\delta(\theta_j^s)(i) \neq 0$. Thus, if $\theta_j^s$ applied to w (recall that by definition $\theta_j^s$ is enabled by w) yields $w'$, then $w'(i) \neq 0$. Since each $\theta \in G$ has nonnegative displacement, after no more than k applications of this technique, we generate a z such that $y=z+\sum_{j=1}^{r_{s'}} a'_j{*}\delta(\theta_j^{s'})$, where $z(i) \leq (c_1{*}k{*}m{*}n)^{c_2{*}k{*}m}$ for some $c_1$ and $c_2$, $z(i)=0$ iff $y(i)=0$, $a'_j \geq 0$, $G_{s'}=\{\theta_1^{s'},\ldots,\theta_{r_{s'}}^{s'}\} \subseteq G_s$, and $\theta_j^{s'}$ is enabled at z, for all $1 \leq i \leq k$, $1 \leq j \leq r_{s'}$. Clearly, $z \in B$, and if $a'_j \geq 1$, then $\delta(\theta_j^{s'}) \in P_z$. Thus, $y \in \cup_{v \in B} \mathcal{L}(v,P_v)$. □

We will now show that if some vector w is reachable in a conflict-free VRS $\mathcal{V}_1$ but not in another conflict-free VRS $\mathcal{V}_2$, then there is some vector $w'$ whose binary representation is polynomial in the size of the representations of $\mathcal{V}_1$ and $\mathcal{V}_2$, such that $w'$ is reachable in $\mathcal{V}_1$ but not in $\mathcal{V}_2$. In so doing, we closely follow the technique developed in [17]. An important point to remember is that for any linear set $\mathcal{L}(v,P_v)$ given in the SLS representation of $\mathcal{V}$ (from Lemma 3.3), $v(i)=0$ only if for all $p \in P_v$, $p(i)=0$. It is precisely this fact that gives us our improvement over a direct application of the results from [17]. Since the proof is rather lengthy, we omit many of the details that may be found in [17].

**Lemma 3.4:** There exist constants c and d such that for any two $k \times m$ conflict-free VRSs $\mathcal{V}_1$ and $\mathcal{V}_2$ in which n is the largest absolute value of any integer, if $w \in R(\mathcal{V}_1) \setminus R(\mathcal{V}_2)$, then there exists a $w' \in R(\mathcal{V}_1) \setminus R(\mathcal{V}_2)$ such that $w'(i) \leq (c*k*m*n)^{d*k^5*m}$.

**Proof:** From Lemma 3.3, $R(\mathcal{V}_1) = \cup_{v \in B_1} \mathcal{L}(v, P_v^1)$ and $R(\mathcal{V}_2) = \cup_{v \in B_2} \mathcal{L}(v, P_v^2)$, where $B_1$, $B_2$, $P_v^1$, and $P_v^2$ are as defined in Lemma 3.3. If we let $|S|$ denote the number of elements in a set S, then $|B_2| \leq (c_1*k*m*n)^{d_1*k^{2*m}}$ for some constants $c_1$ and $d_1$ independent of k, m, and n. Let $w \in SL_1 \setminus SL_2$, where $SL_1 = \cup_{v \in B_1} \mathcal{L}(v, P_v^1)$, $SL_2 = \cup_{v \in B_2} \mathcal{L}(v, P_v^2)$. Without loss of generality, assume that $w \in L \setminus SL_2$, where $L = \mathcal{L}(x, P_x^1)$ such that $x \in B_1$. For convenience, let $B_2 = \{x_1, \ldots, x_{|B_2|}\}$, and let $C = \mathcal{C}(x, P_x^1)$ and $C_i = \mathcal{C}(x_i, P_{x_i}^2)$. Without loss of generality, let $w \in C_1 \cap \ldots \cap C_r$ and $w \notin C_{r+1} \cup \ldots \cup C_{|B_2|}$, where $1 \leq r \leq |B_2|$. From Lemma A.5 in [17], for each $C_j$, $r+1 \leq j \leq |B_2|$, there is a halfspace $H_j$ defined by a linear inequality $a_j x^T \geq b_j$, whose entries are no larger in absolute value than $(c_2*k*m*n)^{d_2*k^{2*m}}$ for some constants $c_2$, $d_2$, independent of k, m, and n, such that $w \in H_j$ and $H_j \cap C_j = \emptyset$. Then let $C_w = C \cap C_1 \cap \ldots \cap C_r \cap H_{r+1} \cap \ldots \cap H_{|B_2|}$. From Lemma 2.1 in [17], $C_w = \text{conv}(E) + \mathcal{C}(\mathbf{0}, F)$, where $E \subseteq R^k$, $F \subseteq N^k$ are finite sets of nonnegative vectors. (For any $U, V \subseteq R^k$, $U+V$ denotes $\{u+v : u \in U, v \in V\}$.) Furthermore, E and F can be chosen so that the integers in $E \cup F$ are no larger than $(c_3*k*m*n)^{d_3*k^{4*m}}$ for some constants $c_3$ and $d_3$ independent of k, m, and n.

We will now show that $P_{x_1}^2 = P_{x_2}^2 = \ldots = P_{x_r}^2$. It is in showing this fact that we depart from the proof given in [17]. Consider arbitrary $P_{x_j}^2$ and $P_{x_{j'}}^2$, $1 \leq j, j' \leq r$. We will show that $P_{x_j}^2 = P_{x_{j'}}^2$. If $x_j(i)$ ($x_{j'}(i)$) = 0, then for all $p \in P_{x_j}^2$ ($P_{x_{j'}}^2$), $p(i) = 0$ (by the definition given in Lemma 3.3). Now since $w \in \mathcal{C}(x_j, P_{x_j}^2) \cap \mathcal{C}(x_{j'}, P_{x_{j'}}^2)$, if $x_j(i)$ ($x_{j'}(i)$) = 0, then $w(i) = 0$; i.e., if $w(i) \neq 0$, then $x_j(i)$ ($x_{j'}(i)$) $\neq 0$. Conversely, if $w(i) = 0$, clearly $x_j(i)$ ($x_{j'}(i)$) = 0. Thus $x_j(i) = 0$ iff $x_{j'}(i) = 0$. Now consider a sequence of addition rules $\theta$ enabled at $x_j$. From Corollary 3.1, there is a sequence $\theta'$, $\Psi(\theta) = \Psi(\theta')$, such that $\theta'$ is enabled at some point $x' \leq x_j$, where $x'$ has no element greater than 1. Since $x_j(i)=0$ iff $x_{j'}(i)=0$, $x' \leq x_{j'}$, and, hence, $\theta'$ is enabled at $x_{j'}$. In particular, any loop enabled at $x_j$ is also enabled at $x_{j'}$, and by symmetry, the converse also holds. Thus by the definition given in Lemma 3.3, $P^2_{x_j} = P^2_{x_{j'}}$.

We now continue with the reasoning from [17] using the fact shown above. Consider w and $C_w$. We have $w \in C_w$. In what follows we will show that in $C_w$ if w has an element that is too large, i.e., w is too "far away" from conv(E), then we can find a "small" witness w′ for the fact that $SL_1 \not\subseteq SL_2$. To this end, consider the linear sets $\mathcal{L}(\mathbf{0},P^1_x)$, $\mathcal{L}(\mathbf{0},P^2_{x_1})$, and the cone $\mathcal{C}(\mathbf{0},F)$. Huynh [17] points out that $\mathcal{C}(\mathbf{0},F) \subseteq \mathcal{C}(\mathbf{0},P^1_x) \cap \mathcal{C}(\mathbf{0},P^2_{x_1}) \cap ... \cap \mathcal{C}(\mathbf{0},P^2_{x_r})$. Since $P^2_{x_1} = P^2_{x_2} = ... = P^2_{x_r}$, $\mathcal{C}(\mathbf{0},F) \subseteq \mathcal{C}(\mathbf{0},P^1_x) \cap \mathcal{C}(\mathbf{0},P^2_{x_1})$. Therefore, from Caratheodory's theorem for cones (see [37]), each $v \in F$ may be expressed as a linear combination of not more than k linearly independent vectors in $P^1_x$ ($P^2_{x_1}$, respectively). Hence, there are nonnegative integers $\lambda_1$, $\lambda_2$ such that $\lambda_1 v \in \mathcal{L}(\mathbf{0},P^1_x)$ and $\lambda_2 v \in \mathcal{L}(\mathbf{0},P^2_{x_1})$, where $\lambda_1$, $\lambda_2$ may be chosen, by Cramer's rule, as some subdeterminants of the matrices formed by vectors in $P^1_x$, $P^2_{x_1}$, respectively. Thus, $\lambda_1,\lambda_2 \leq (c_4 * k * m * n)^{d_4 * k^2 * m}$ for some constants $c_4$ and $d_4$ independent of k, m, and n. Hence, the least common multiple $\lambda_v$ of $\lambda_1,\lambda_2$ is no more than $(c_5 * k * m * n)^{d_5 * k^2 * m}$. Thus, for each $v \in F$, there exists an integer $\lambda_v \leq (c_5 * k * m * n)^{d_5 * k^2 * m}$ such that $\lambda_v v \in \mathcal{L}(\mathbf{0},P^1_x) \cap \mathcal{L}(\mathbf{0},P^2_{x_1}) = \mathcal{L}(\mathbf{0},P^1_x) \cap \mathcal{L}(\mathbf{0},P^2_{x_1}) \cap ... \cap \mathcal{L}(\mathbf{0},P^2_{x_r})$. (Note that in [17], $\lambda_v$ is the least common multiple of r+1 integers; if we had not been able to reduce the number of factors, our bound for $\lambda_v$ would have been double-exponential in k and m.)

Let $G=\{\lambda_v v : v \in F\}$. Intuitively, each $\lambda_v v$ is a "superperiod" which can be subtracted from w so that a "small" witness w′ can be obtained. We formalize this idea in the following. Suppose w has an element larger than any integer in E. Then $C_w$ is an unbounded polyhedron, and F (or equivalently G) is not empty.

Consider the lattice points in $C_w$, i.e., elements in $C_w \cap N^k$. Let $u \in C_w \cap N^k$. By Caratheodory's theorem for cones (see [37]), u may be expressed as

$u=\sum_{y\in E}\mathcal{G}_y y+\sum_{z\in G'}\alpha_z z$, where $\mathcal{G}_y,\alpha_z\in R$, $\mathcal{G}_y,\alpha_z\geq 0$, $\sum_{y\in E}\mathcal{G}_y=1$, and $G'\subseteq G$ is a linearly independent set. Therefore, $u'=u-\sum_{z\in G'}\lfloor\alpha_z\rfloor z\leq u$ and $u'\in C_w\cap N^k$. Let U denote the set of all such lattice points u′ in $C_w\cap N^k$. It can easily be seen that the largest integer in U is no more than $(c_6*k*m*n)^{d_6*k^4*m}$ for some constants $c_6$ and $d_6$ independent of k, m, and n. In addition, it holds that $C_w\cap N^k=\cup_{u\in U}\cup_{G'\subseteq G}\mathcal{L}(u,G')$, where G′ runs over all subsets of $\leq$ k linearly independent vectors in G. It can now be shown, similar to Lemma 2.2 in [17], that for each $u\in U$, the intersection $\mathcal{L}(u,G')\cap\mathcal{L}(x,P^1_x)$ is a SLS of the form $\cup_{y\in Y}\mathcal{L}(y,G')$ so that the integers in Y are no greater than $(c*k*m*n)^{d*k^5*m}$ for some constants c and d independent of k, m, and n. Clearly, $Y\subseteq\mathcal{L}(u,G')\cap\mathcal{L}(x,P^1_x)$.

We are now in position to conclude the proof. Observe that $w\in(C_w\cap N^k)\cap\mathcal{L}(x,P^1_x)$. So for some $w'\in Y$ and G′ a subset of G containing no more than k linearly independent vectors, $w\in\mathcal{L}(w',G')$. Now w′ is in the set $\mathcal{L}(x,P^1_x)$. On the other hand, it is clear that $w'\notin\mathcal{L}(x_1,P^2_{x_1})\cup\ldots\cup\mathcal{L}(x_r,P^2_{x_r})$, since w would belong to $\mathcal{L}(x_1,P^2_{x_1})\cup\ldots\cup\mathcal{L}(x_r,P^2_{x_r})$ otherwise. Thus $w'\in L\backslash SL_2$. □

We are now ready to show our main result, that the containment and equivalence problems are $\Pi^P_2$-complete. The upper bound follows almost immediately from Lemma 3.4 and Theorem 3.1. Before formally proving the theorem, we will briefly explain the strategy for showing the lower bound. Let X and Y be disjoint sets of Boolean variables, and let F(X,Y) be a Boolean expression in 3DNF. Stockmeyer [36] showed the problem of deciding whether $(\forall X)(\exists Y)$:F(X,Y)=0 is $\Pi^P_2$-complete (the notations $(\forall X)$ and $(\exists Y)$ denote $(\forall x_1\ldots\forall x_{n_1})$ and $(\exists y_1\ldots\exists y_{n_2})$, respectively, where $X=\{x_1,\ldots,x_{n_1}\}$ and $Y=\{y_1,\ldots,y_{n_2}\}$). We will reduce this problem to the containment and equivalence problems. The reduction will consist of constructing two conflict-free VRSs, $\mathcal{V}_1$ and $\mathcal{V}_2$, which are identical except that $\mathcal{V}_2$ has one additional rule. Let us say that a clause in F(X,Y) is *killed* if one of its literals has a value of 0. The function of the VRSs is to simulate an assignment of values to the variables in $X\cup Y$, signifying killed clauses by incrementing certain positions. The additional rule in $\mathcal{V}_2$ will allow it to kill all clauses after a complete assignment is made. Thus, if we record which clauses were killed by assignments to X variables, $R(\mathcal{V}_1)=R(\mathcal{V}_2)$ iff for any assignment of values to X there is an assignment of values to Y that results in killing all clauses. Now the VRSs must be able to record which variables have been assigned values, which clauses have been killed, and which clauses have been killed by X variables. We also wish to make our proof general enough to work for conflict-free VASs and Petri nets as well. To accomodate each of these requirements, we use two positions for each variable and eight positions for each clause.

**Theorem 3.2:** The equivalence and containment problems for conflict-free VRSs are $\Pi_2^P$-complete.

**Proof:** Recall that $\Pi_2^P$ is the set of all complements of languages that can be recognized by a polynomial-time-bounded nondeterministic Turing machine with an NP oracle. We will first briefly describe such a machine which will decide non-containment. Let $\mathcal{V}_1$ and $\mathcal{V}_2$ be $k\times m$ conflict-free VRSs such that the largest absolute value of any integer in either VRS is n. From Lemma 3.4, if $R(\mathcal{V}_1)\not\subseteq R(\mathcal{V}_2)$, then there is a $w\in R(\mathcal{V}_1)\backslash R(\mathcal{V}_2)$ that can be guessed in polynomial time. From Theorem 3.1, we can then check that $w\in R(\mathcal{V}_1)\backslash R(\mathcal{V}_2)$ with an NP oracle. Thus, the containment (and, hence, equivalence) problem is in $\Pi_2^P$.

We will now show the containment and equivalence problems to be $\Pi_2^P$-hard. Let $X=\{x_1,\ldots,x_{n_1}\}$, $Y=\{y_1,\ldots,y_{n_2}\}$, $X\cap Y=\emptyset$, $F(X,Y)=C_1\vee\ldots\vee C_m$, $C_j=\alpha_{1,j}\wedge\alpha_{2,j}\wedge\alpha_{3,j}$, $\alpha_{i,j}\in\{x,\bar{x} : x\in X\cup Y\}$. We will define a $(2n_1+2n_2+8m)\times(3n_1+3n_2+8m)$ conflict-free VRS $\mathcal{V}_1$ and a $(2n_1+2n_2+8m)\times(3n_1+3n_2+8m+1)$ conflict-free VRS $\mathcal{V}_2$ such that $R(\mathcal{V}_1)=R(\mathcal{V}_2)$ iff $(\forall X)(\exists Y):F(X,Y)=0$. The construction will be such that $R(\mathcal{V}_1)\subseteq R(\mathcal{V}_2)$; hence, it will also be the case that $R(\mathcal{V}_2)\subseteq R(\mathcal{V}_1)$ iff $(\forall X)(\exists Y):F(X,Y)=0$. For ease of illustration, we will treat the reachable vectors as a set of assignments to a set of variables. The addition rules will then operate on these variables. The variables we will use are $\{a_i,\bar{a}_i : 1\leq i\leq n_1\}\cup\{b_i,\bar{b}_i : 1\leq i\leq n_2\}\cup\{c_{i,j},\bar{c}_{i,j} : 0\leq i\leq 3, 1\leq j\leq m\}$. $a_i$

and $\bar{a}_i$ will correspond to $x_i$, $b_i$ and $\overline{b}_i$ will correspond to $y_i$, $c_{0,i}$ and $\bar{c}_{0,i}$ will correspond to $C_i$, and $c_{i,j}$ and $\bar{c}_{i,j}$ will correspond to $\alpha_{i,j}$. Both $\mathcal{V}_1$ and $\mathcal{V}_2$ will have start vectors of $\mathbf{0}$.

$\mathcal{V}_1$ and $\mathcal{V}_2$ will both have the following rules:

- $v_i^1$, $1 \leq i \leq n_1$:
  $a_i \leftarrow a_i+1$
  $c_{0,j} \leftarrow c_{0,j}+1$ $\forall$ j for which $x_i \in C_j$, $1 \leq j \leq m$
- $v_i^2$, $1 \leq i \leq n_1$:
  $a_i \leftarrow a_i+1$
  $c_{0,j} \leftarrow c_{0,j}+1$ $\forall$ j for which $\bar{x}_i \in C_j$, $1 \leq j \leq m$
- $v_i^3$, $1 \leq i \leq n_2$:
  $b_i \leftarrow b_i+1$
  $c_{j,k} \leftarrow c_{j,k}+1$ $\forall$j,k for which $\alpha_{j,k}=y_i$, $1 \leq j \leq 3$, $1 \leq k \leq m$
- $v_i^4$, $1 \leq i \leq n_2$:
  $b_i \leftarrow b_i+1$
  $c_{j,k} \leftarrow c_{j,k}+1$ $\forall$j,k for which $\alpha_{j,k}=\bar{y}_i$, $1 \leq j \leq 3$, $1 \leq k \leq m$
- $v_{i,j}^5$, $0 \leq i \leq 3$, $1 \leq j \leq m$:
  $c_{i,j} \leftarrow c_{i,j}-1$
  $\bar{c}_{i,j} \leftarrow \bar{c}_{i,j}+1$
- $v_{i,j}^6$, $1 \leq i \leq 3$, $1 \leq j \leq m$:
  $c_{i,j} \leftarrow c_{i,j}+1$
  $c_{i-1,j} \leftarrow c_{i-1,j}+1$
  $\bar{c}_{i-1,j} \leftarrow \bar{c}_{i-1,j}-1$
- $v_i^7$, $1 \leq i \leq m$:
  $c_{1,i} \leftarrow c_{1,i}+1$
  $c_{3,i} \leftarrow c_{3,i}+1$
  $\bar{c}_{3,1} \leftarrow \bar{c}_{3,i}-1$
- $v_i^8$, $1 \leq i \leq n_1$:
  $a_i \leftarrow a_i-1$
  $\bar{a}_i \leftarrow \bar{a}_i+1$

- $v_i^9$, $1 \le i \le n_2$:

  $b_i \leftarrow b_i - 1$

  $\overline{b}_i \leftarrow \overline{b}_i + 1$

In addition to the above rules, $\mathcal{V}_2$ has the following rule:

- $v^{10}$:

  $a_i \leftarrow a_i + 1, \forall\ 1 \le i \le n_1$
  $\bar{a}_i \leftarrow \bar{a}_i - 1, \forall\ 1 \le i \le n_1$
  $b_i \leftarrow b_i + 1, \forall\ 1 \le i \le n_2$

  $\overline{b}_i \leftarrow \overline{b}_i - 1, \forall\ 1 \le i \le n_2$
  $c_{i,j} \leftarrow c_{i,j} + 1, \forall\ 1 \le i \le 3,\ 1 \le j \le m$

Clearly, both systems are conflict-free, and $R(\mathcal{V}_1) \subseteq R(\mathcal{V}_2)$. We will call all rules superscripted with i *type i* rules. The type 1 rules correspond to assignments of 0 to X variables, and type 2 rules correspond to assignments of 1 to X variables. Similarly, type 3 rules correspond to assignments of 0 to Y variables, and type 4 rules correspond to assignments of 1 to Y variables. Note that the execution of a type 1 or 2 rule that corresponds with an assignment that kills clause $C_j$ will increment $c_{0,j}$. Likewise, the execution of a type 3 or 4 rule that corresponds with an assignment that makes $\alpha_{i,j}=0$ will increment $c_{i,j}$. Thus, the function of the types 5, 6, and 7 rules is to allow $c_{i,j}$, $1 \le i \le 3$, to reach any positive value if clause $C_j$ is killed. Finally, the types 8 and 9 rules will enable rule $v^{10}$ in $\mathcal{V}_2$, which in turn will allow $c_{i,j}$, $1 \le i \le 3$, to reach any positive value if all $a_k$'s and $b_k$'s have been incremented at least once.

Based on the above comments, we now make the following observations:

1. $a_i + \bar{a}_i$ reflects the number of value assignments made to $x_i$ (where any assignment may be made 0, 1, or more times).

2. $b_i + \overline{b}_i$ reflects the number of value assignments made to $y_i$.

3. $c_{0,j} + \bar{c}_{0,j}$ reflects the number of times clause $C_j$ has been killed by assignments to variables in X.

4. In $\mathcal{V}_1$, $c_{i,j}+\bar{c}_{i,j}$, $1\leq i\leq 3$, can become positive only if clause $C_j$ is killed.

5. In $\mathcal{V}_2$, $c_{i,j}+\bar{c}_{i,j}$, $1\leq i\leq 3$, can become positive only if either clause $C_j$ is killed or every variable in $\{X\cup Y\}$ has been assigned a value at least once.

We are now ready to show that $R(\mathcal{V}_2)\subseteq R(\mathcal{V}_1)$ iff $(\forall X)(\exists Y):F(X,Y)=0$.

($\Rightarrow$): Assume $R(\mathcal{V}_2)\subseteq R(\mathcal{V}_1)$. Let $B:X\mapsto\{0,1\}$ be any assignment of Boolean values to the variables in X. We will show that there is a $B':Y\mapsto\{0,1\}$ such that $F(B(X),B'(Y))=0$. We will first construct a path $\sigma$ in $\mathcal{V}_2$. Clearly, all rules of types 1-4 are always enabled; therefore, we first execute, for each i, $1\leq i\leq n_1$, $v_i^1$ if $B(x_i)=0$, or $v_i^2$ if $B(x_i)=1$. Next, we execute, for each i, $1\leq i\leq n_2$, either $v_i^3$ or $v_i^4$. At this time, $a_i=1$ and $b_j=1$ for $1\leq i\leq n_1$, $1\leq j\leq n_2$. Thus, we can now execute, for each i, $1\leq i\leq n_1$, and each j, $1\leq j\leq n_2$, $v_i^8$ and $v_j^9$. Since now $\bar{a}_i=1$ and $\bar{b}_j=1$, $1\leq i\leq n_1$, $1\leq j\leq n_2$, we can execute $v^{10}$. Note that this leaves $a_i=1$, $\bar{a}_i=0$, $b_j=1$, $\bar{b}_j=0$, $\bar{c}_{0,l}=0$, $1\leq c_{k,l}\leq 2$, and $\bar{c}_{k,l}=0$, for all $1\leq i\leq n_1$, $1\leq j\leq n_2$, $1\leq k\leq 3$, $1\leq l\leq m$. $c_{0,l}$ will be nonzero iff the assignment B kills clause $C_l$ $(1\leq l\leq m)$. Call the resulting vector w.

Now since $w\in R(\mathcal{V}_2)$, $w\in R(\mathcal{V}_1)$ also. Let $\sigma'$ be a path to w in $\mathcal{V}_1$. Since $a_i=1$ and $\bar{a}_i=0$ in w, $\sigma'$ must contain exactly one occurrence of either $v_i^1$ or $v_i^2$, but not both. Furthermore, the rules of types 1 and 2 must clearly produce the same values in all the $c_{0,i}$'s as those in w, $1\leq i\leq m$. Now the remaining rules in $\sigma$ must make all positions $c_{i,j}$, $1\leq i\leq 3$, $1\leq j\leq m$, positive. If clause $C_j$ is not killed by B, then $c_{0,j}=\bar{c}_{0,j}=0$, and the only way for any position $c_{i,j}$, $1\leq i\leq 3$, to become positive is for some type 3 or 4 rule to increment one of them. Since for each i, exactly one of $v_i^3$ or $v_i^4$ must be executed in $\sigma'$, there must be some $B':Y\mapsto\{0,1\}$ such that for all j, $1\leq j\leq m$, if B does not kill $C_j$, then $B'$ does. Hence, $(\forall X)(\exists Y):F(X,Y)=0$.

($\Leftarrow$): Assume $(\forall X)(\exists Y):F(X,Y)=0$. Let w be an arbitrary vector in $R(\mathcal{V}_2)$. We will show that $w\in R(\mathcal{V}_1)$. Let $\sigma$ be a path to w in $R(\mathcal{V}_2)$. If $\sigma$ does not use $v^{10}$, then clearly $w\in R(\mathcal{V}_1)$. Therefore, assume without loss of generality that $\sigma$ uses $v^{10}$. It is clear from the proof of Lemma 3.2 that we can assume without loss of generality that some initial path $\sigma'$ in $\sigma$ uses exactly one occurrence of every rule used by $\sigma$. Furthermore, it is clear from the proof of Lemma 3.1 that we can assume without loss of generality that at any point in $\sigma'$, the next rule to be executed is some arbitrary rule used by $\sigma$, as long as it is enabled and has not yet been executed. Now under these assumptions, before $v^{10}$ can be executed for the first time, it must be the case that $\bar{a}_i=1$ and $\bar{b}_i=1$; i.e., each type 8 and type 9 rule has been executed once. Now before $v_i^8$ ($v_i^9$) can be executed, either $v_i^1$ or $v_i^2$ ($v_i^3$ or $v_i^4$) must have been executed. We will assume without loss of generality that exactly one of these two rules has been executed before $v^{10}$ is first executed. Call the initial portion of $\sigma$ ending with the first execution of $v^{10}$ $\sigma''$, and let $w'$ be the vector produced by $\sigma''$.

We will first show that $w' \in R(\mathcal{V}_1)$; then we will show that there is a path from $w'$ to $w$ in $\mathcal{V}_1$. We first execute in $\mathcal{V}_1$ the types 1 and 2 rules used in $\sigma''$. Note that since exactly one of $v_i^1$ and $v_i^2$ is used in $\sigma''$, this rule represents the assignment of a Boolean value to $x_i$. Let $B{:}X \mapsto \{0,1\}$ represent the assignment induced by these rules. Since $(\forall X)(\exists Y){:}F(X,Y)=0$, there is a $B'{:}Y \mapsto \{0,1\}$ such that for all j, $1 \le j \le m$, if $C_j$ is not killed by B, then it is killed by $B'$. We next execute the rules corresponding to $B'$. Now, the values of $a_i$, $\bar{a}_i$, $b_j$, $\bar{b}_j$, $c_{0,k}$, and $\bar{c}_{0,k}$ match their counterparts in $w'$, $1 \le i \le n_1$, $1 \le j \le n_2$, $1 \le k \le m$, and for every j, $1 \le j \le m$, there is an i, $0 \le i \le 3$, such that $c_{i,j} > 0$. Furthermore, no $c_{i,j}$, $1 \le i \le 3$, $1 \le j \le m$, is greater than 1. Thus, rules of types 5-7 can clearly be used to bring the $c_{i,j}$'s equal to their counterparts in $w'$. Hence, $w'$ is reachable in $\mathcal{V}_1$. We can now simulate the remainder of $\sigma$ as follows. We simulate $\sigma$ until the next occurrence of $v^{10}$ is reached, except that we skip all occurrences of rules of types 8 and 9. Now when $v^{10}$ is reached, at least one of each type 8 and type 9 rule must have been skipped. We therefore simulate one occurrence of each rule of types 8-10 using only rules of types 5-7. We iterate this process until the end of $\sigma$ is reached. We then execute all rules of types 8 and 9 that have not yet been simulated. It is not hard to verify that every rule in this simulation is enabled at the proper time. □

## 4. Liveness and Fairness

The next problems we would like to examine are the liveness and transition liveness problems. In [13] an $O(n^{1.5})$ algorithm was given for determining boundedness for a conflict-free VRS, where n is the number of bits needed to encode the VRS. One portion of this algorithm was devoted to determining which addition rules could be used infinitely often. Call the set of these rules $\mathcal{I}$. In what follows, we will show that the set $\mathcal{L}$ of all live rules is identical to $\mathcal{I}$. From this result, we will be able to show the liveness and transition liveness problems to be PTIME-complete.

**Lemma 4.1:** For any $k \times m$ conflict-free VRS $(v_0, U, V)$, the set $\mathcal{I}$ of rules that can be used infinitely often is the same as the set $\mathcal{L}$ of live rules.

**Proof:** Clearly $\mathcal{L} \subseteq \mathcal{I}$. Assume $\mathcal{I} \not\subseteq \mathcal{L}$, and let $w \in N^k$ be such that in some path, w is the last point from which there exists for each rule in $\mathcal{I}$, a path that enables that rule. Thus, there are rules $v_1 \in V$ $(u_1 \leq w)$, $v_2 \in \mathcal{I}$, and a path $\sigma$ in (w,U,V) such that $v_2$ is used in $\sigma$, but $v_2$ can never be used after $w+v_1$ is reached. From Lemma 3.1, there is a path $\sigma'$ enabled at w that uses $v_1$ and all rules in $\mathcal{I}$. Note from the proof of Lemma 3.1 that without loss of generality, we can assume that any arbitrary rule enabled at w is used first in $\sigma'$; thus, we may assume $v_1$ is used first in $\sigma'$. Since $v_2 \in \mathcal{I}$ but $v_2$ cannot be used after $w+v_1$ is reached, $v_2=v_1$. Now since $v_2 \in \mathcal{I}$, for every position from which $v_2$ subtracts, there must be a rule in $\mathcal{I}$ that adds to that position. Since every rule in $\mathcal{I}$ is executed in $\sigma'$, $v_2$ is enabled by $\sigma'$--a contradiction. Hence, $\mathcal{L}=\mathcal{I}$. □

**Theorem 4.1:** The liveness and transition liveness problems for conflict-free VRSs are PTIME-complete. Furthermore, there is an $O(n^{1.5})$ algorithm to decide each problem.

**Proof:** Since the set of live rules can be computed in time $O(n^{1.5})$ (see [13]), we can clearly decide both problems in time $O(n^{1.5})$. The PTIME-hard proofs are similar to Lemma 4.1 in [13]. The details are left to the reader. □

We now consider the various fair nontermination problems defined in Section 2. It is often the case that problems in verification of parallel computations can be phrased as fair nontermination problems concerning various formal models of computation. Most work of this sort to date has been concerned with finite-state models (see, e.g., [7, 14, 25, 34, 35, 38]). Hence, the reason that we now examine fair nontermination problems for conflict-free VRSs is that perhaps some of the problems in verification of parallel computations can be modelled as some type of fair nontermination problem for conflict-free VRSs. It turns out that some of the fair nontermination problems we examine have efficient solutions. Therefore, it may be the case that the algorithms presented here will be useful in the verification of parallel systems. The first three of these problems we will consider are the impartial, just, and fair nontermination problems.

**Theorem 4.2:** The impartial, just, and fair nontermination problems for conflict-free VRSs are all PTIME-complete. Furthermore, there is an $O(n^{1.5})$ algorithm to decide each problem.

**Proof:** We first claim that there is an infinite impartial path iff the VRS is live. If there is an infinite impartial path, the VRS is clearly live. Conversely, if the VRS is live, from Lemma 3.1, there is a path from any reachable marking which contains uses one occurrence of each rule; hence, there is an infinite impartial path. Thus, from Theorem 4.1, the impartial nontermination problem for conflict-free VRSs is PTIME-complete, and there is an $O(n^{1.5})$ algorithm to solve the problem.

Next, we claim that there is an infinite just (fair) path iff there is an infinite path. To see this, note that from any reachable marking, we can execute a path using every live rule. Now from Lemma 4.1, the set of live rules is exactly the set of rules which can be enabled infinitely often. Thus, from [13], there is an $O(n^{1.5})$ algorithm to decide just (fair) nontermination. The lower bound follows immediately from the proof of Theorem 4.1 in [13]. □

**Remark:** Prof. Vidal-Naquet has pointed out to us yet another definition of fairness, namely, an infinite path $\sigma$ is fair in $(v_0,U,V)$ if for every rule $v_j \in \{v_j : v_j$ is live in $(w,U,V)$ for all $w \in \sigma\}$, $v_j$ is executed infinitely often. See also Queille and Sifakis [33]. Clearly, the above result holds for this type of fairness as well.

Now of the six remaining fair nontermination problems, five are NP-complete, and one is NLOGSPACE-complete.

**Theorem 4.3:** The i-fair nontermination problem for conflict-free VRSs is NP-complete for $i \in \{1,2,2',3,3'\}$.

**Proof:** We first show that all five of the problems are NP-hard by reducing reachability to each of them. Let $(v_0,U,V)$ be an arbitrary $k \times m$ conflict-free VRS, and let w be an arbitrary vector in $N^k$. Let $V'$ $(U')$ be V (U) with an additional column of zeros, and let $\mathcal{A}=\{\{w\}\}$. It is now easy to see that for each $i \in \{1,2,2',3,3'\}$, there is an i-fair path for $\mathcal{A}$ in $(v_0,U',V')$ iff $w \in R(v_0,U,V)$. Thus, from Lemma 3.1, the five problems are NP-hard.

We now describe an NP algorithm for each problem. Let $(v_0,U,V)$ be a $k \times m$ conflict-free VRS, and let $\mathcal{A}$ be a finite set of finite subsets of $N^k$.

- 1-fair: Guess a vector w in some set $A \in \mathcal{A}$, and verify that $w \in R(v_0,U,V)$.
- 2-fair: Guess a vector w in some set $A \in \mathcal{A}$, verify that $w \in R(v_0,U,V)$, then verify that for some $w'$ reachable in one step from w, $w \in R(w',U,V)$.
- $2'$-fair: Guess some set $A \in \mathcal{A}$, verify that some element of A is reachable, then guess a sequence $\theta$ of $|A|$ rules. If every vector reached in executing $\theta$ is in A, then there is an infinite $2'$-fair path.
- 3-fair: Same as $2'$-fair, except that each element in A must be reached by $\theta$.
- $3'$-fair: Guess some set $A \in \mathcal{A}$, verify that the first element of A is reachable, verify that each successive element of A is reachable from the previous element of A, and verify that the first element of A is reachable from the last element of A.

Clearly, all of the above algorithms operate in NP; therefore, all five problems are NP-complete. □

**Theorem 4.4:** The $1'$-fair nontermination problem for conflict-free VRSs is NLOGSPACE-complete.

**Proof:** We will first show the problem to be in NLOGSPACE. Let the start vector initially be the current vector. We first guess a set $A \in \mathcal{A}$. Next, we verify that the start vector is in A. Then we repeatedly guess a rule $v_j$ and a vector $w \in A$, and verify that the execution of $v_j$ at the current vector produces w. w then becomes the current vector. If this process can be continued for more iterations than there are rules in A, there is an infinite $1'$-fair path. Clearly, this nondeterministic algorithm can be implemented using only logarithmic space.

We will now show the problem to be NLOGSPACE-hard. We will use a reduction from the graph accessibility problem, which is well known to be NLOGSPACE-complete [18]. Let $G=(Q,E)$ be a directed graph in which $Q=\{q_1,\ldots q_n\}$, and $q_1,q_n \in Q$ are the start vertex and final vertex, respectively. We first construct a graph $G'=(Q',E')$, where $Q'=\{p_{i,j} : q_i \in Q \text{ and } 1 \leq j \leq n\}$ and $E'=\{(p_{i,j},p_{i',j+1}) : (q_i,q_{i'}) \in E \text{ or } q_i=q_{i'}=q_n, 1 \leq j \leq n\text{-}1\} \cup \{(p_{n,n},p_{n,n})\}$. Clearly, this construction can be done in deterministic logspace, and there is an infinite path from $p_{1,1}$ in $G'$ iff there is a path from $q_1$ to $q_n$ in G. We will now construct a $(2n^2) \times (3n^2)$ conflict-free VRS $\mathcal{V}$ and a set $A \subseteq N^{2n^2}$ such that there is an infinite path in $\mathcal{V}$ that is $1'$-fair for $\{A\}$ iff there is a path from $q_1$ to $q_n$ in G. We will again use variables to denote the positions in the VRS. The variables we will use are $\{x_{i,j},y_{i,j} : p_{i,j} \in Q'\}$. In the start vector, $x_{1,1}=1$, and all other positions are 0. $\mathcal{V}$ will have the following rules:

- $v^1_{i,j}$, where $p_{i,j} \in Q'$:

  $x_{i,j} \leftarrow x_{i,j}-1$
  $y_{i,j} \leftarrow y_{i,j}+1$

- $v^2_{i,j}$, where $p_{i,j} \in Q'$:

  $x_{i,j} \leftarrow x_{i,j}+1$

- $v^3_{i,j}$, where $p_{i,j} \in Q'$:

  $y_{i,j} \leftarrow y_{i,j}-1$

Again, all rules superscripted with i will be called type i rules. The set A will contain all 0-1 vectors in $N^{2n^2}$ that contain exactly one 1, as well as all 0-1 vectors containing exactly two 1's such that $y_{i,j}=x_{i',j'}=1$ iff $(p_{i,j},p_{i',j'}) \in E'$. Clearly, $\mathcal{V}$ is conflict-free, and the construction can be done in deterministic logspace. We will now show that there is an infinite path in $\mathcal{V}$ that is 1'-fair for {A} iff there is an infinite path from $p_{1,1}$ in G'.

($\Rightarrow$): Let $\sigma$ be an infinite path in $\mathcal{V}$ that is 1'-fair for {A}. Then every vector reached by $\sigma$ is in A. We associate with $\sigma$ a sequence of vertices in Q' as follows: with each point reached by $\sigma$ in which one x variable is 1 and all other variables are 0, associate the vertex $p_{i,j}$ such that $x_{i,j}=1$. We will show by induction that for every h>0, (1) there are at least h vertices in the sequence, and (2) the first h vertices in the sequence form a path from $p_{1,1}$ in G'. Clearly, this holds for h=1. Let h>1, and assume the claim holds for h-1. Suppose the last of the h-1 points has $x_{i,j}=1$. Now since $\mathbf{0} \notin A$ and all vectors in A with more than one 1 have a y variable equal to 1, $v^1_{i,j}$ must be executed next in $\sigma$. Now since executing $v^3_{i,j}$ would produce $\mathbf{0} \notin A$, some type 2 rule must be executed. This produces a vector with $x_{i',j'}=y_{i,j}=1$. Since this vector must be in A, $(p_{i,j},p_{i',j'}) \in E'$. Now clearly, the only rule that will produce a vector in A is $v^3_{i,j}$. This produces a vector in which $x_{i',j'}$ is the only position equal to 1. Since $(p_{i,j},p_{i',j'}) \in E'$ and there is a path of h-1 vertices from $p_{1,1}$ to $p_{i,j}$, the claim holds for h. Thus, it is clear that there is an infinite path from $p_{1,1}$ in G'.

($\Leftarrow$): Let $\sigma$ be an infinite path from $p_{1,1}$ in G'. Associate with $\sigma$ a sequence of vectors such that with each vertex $p_{i,j}$ reached by $\sigma$ is associated the vector with $x_{i,j}=1$ and all other variables equal to 0. We will show by induction that for any h>0 there is a path in $\mathcal{V}$ that remains within A and passes in order through the first h points in the sequence associated with $\sigma$. Clearly, this holds for h=1. Let h>1, and assume the claim for h-1. Suppose the $(h\text{-}1)^{st}$ vertex reached by $\sigma$ is $p_{i,j}$. Let $p_{i',j'}$ be the next vertex reached by $\sigma$. Then $\mathcal{V}$ can clearly execute $v^1_{i,j}$, $v^2_{i',j'}$, and $v^3_{i,j}$, producing vectors in A, the last of which has $x_{i',j'}=1$ and all other variables equal to 0. Thus, the claim holds for h. It is now clear that there is an infinite path in $\mathcal{V}$ that is $1'$-fair for $\{A\}$. This completes the proof. $\square$

**Acknowledgment:** We would like to thank Prof. Vidal-Naquet for pointing out the definition of fairness given in [33].

## References

[1] Baker, H., Rabin's Proof of the Undecidability of the Reachability Set Inclusion Problem of Vector Addition Systems, MIT Project MAC. CSGM 79, Cambridge, MA, 1973.

[2] Borosh, I. and Treybig, L., Bounds on Positive Integral Solutions of Linear Diophantine Equations, *Proc. AMS 55*, 2 (March 1976), pp. 299-304.

[3] Cardoza, E., Lipton. R. and Meyer, A., Exponential Space Complete Problem for Petri Nets and Commutative Semigroups, *Proceedings of the 8th Annual ACM Symposium on Theory of Computing* (1976), pp. 50-54.

[4] Carstensen, H., and Valk, R., Infinite Behaviour and Fairness in Petri Nets, *Advances in Petri Nets*, LNCS 188 (1984), 83-100.

[5] Clote, P., The Finite Containment Problem for Petri Nets, *Theoret. Comp. Sci. 43* (1986), 99-106.

[6] Crespi-Reghizzi, S. and Mandrioli, D., A Decidability Theorem for a Class of Vector Addition Systems, *Information Processing Letters 3*, 3 (1975), pp. 78-80.

[7] Emerson, E., and Lei, C., Modalities for Model Checking: Branching Time Logic Strikes Back, to appear in *Science of Computer Programming* (1986). (Some of these results were presented at the *18th Annual Hawaii International Conference on System Sciences* and at the *12th Annual ACM Symposium on Principles of Programming Languages.*).

[8] Ginzburg, A. and Yoeli, M., Vector Addition Systems and Regular Languages, *J. of Computer and System Sciences 20* (1980), pp. 277-284.

[9] Grabowski, J., The Decidability of Persistence for Vector Addition Systems, *Information Processing Letters 11*, 1 (1980), pp. 20-23.

[10] Hack, M., The Equality Problem for Vector Addition Systems is Undecidable, *Theoret. Comp. Sci. 2* (1976), pp. 77-95.

[11] Hopcroft, J. and Pansiot, J., On the Reachability Problem for 5-Dimensional Vector Addition Systems, *Theoret. Comp. Sci. 8* (1979), pp. 135-159.

[12] Howell, R., Huynh, D., Rosier, L., and Yen, H., Some Complexity Bounds for Problems Concerning Finite and 2-Dimensional Vector Addition Systems with States, to appear in *Theoret. Comp. Sci.*

[13] Howell, R., Rosier, L., and Yen, H., An $O(n^{1.5})$ Algorithm to Decide Boundedness for Conflict-Free Vector Replacement Systems, to appear in *Information Processing Letters.*

[14] Hart, S., Sharir, M., and Pnueli, A., Termination of Probabilistic Concurrent Programs, *ACM Transactions on Programming Languages and Systems 5* (1983), 356-380.

[15] Huynh, D., The Complexity of Semilinear Sets, *Elektronische Informationsverarbeitung und Kybernetik 18* (1982), pp. 291-338.

[16] Huynh, D., The Complexity of the Equivalence Problem for Commutative Semigroups and Symmetric Vector Addition Systems, *Proceedings of the 17th Annual ACM Symposium on Theory of Computing* (1985), pp. 405-412.

[17] Huynh, D., A Simple Proof for the $\Sigma_2^P$ Upper Bound of the Inequivalence Problem for Semilinear Sets, *Elektronische Informationsverarbeitung und Kybernetik 22* (1986), pp. 147-156.

[18] Jones, N., Space-Bounded Reducibility Among Combinatorial Problems, *J. of Computer and System Sciences 11* (1975), 68-75.

[19] Jones, N., Landweber, L. and Lien, Y., Complexity of Some Problems in Petri Nets, *Theoret. Comp. Sci. 4* (1977), pp. 277-299.

[20] Karp, R. and Miller, R., Parallel Program Schemata, *J. of Computer and System Sciences 3*, 2 (1969), pp. 147-195.

[21] Kosaraju, R., Decidability of Reachability in Vector Addition Systems, *Proceedings of the 14th Annual ACM Symposium on Theory of Computing* (1982), pp. 267-280.

[22] Landweber, L., Decision Problems for $\omega$-Automata, *Math. Syst. Theory 3* (1969), 376-384.

[23] Landweber, L. and Robertson, E., Properties of Conflict-Free and Persistent Petri Nets, *JACM 25*, 3 (1978), pp. 352-364.

[24] Lehman, D., Pnueli, A., and Stavi, J., Impartiality, Justice, and Fairness: The Ethics of Concurrent Termination, *Proceedings of the 8th International Colloquium on Automata, Languages, and Programming*, LNCS 115 (1981), 264-277.

[25] Lichtenstein, O., and Pnueli, A., Checking that Finite State Concurrent Programs Satisfy their Linear Specification, *Proc. of the 12th Annual ACM Symposium on Principles of Programming Languages* (1985), 97-107.

[26] Lipton, R., The Reachability Problem Requires Exponential Space, Yale University, Dept. of CS., Report No. 62, Jan., 1976.

[27] Mayr, E., Persistence of Vector Replacement Systems is Decidable, *Acta Informatica 15* (1981), pp. 309-318.

[28] Mayr, E., An Algorithm for the General Petri Net Reachability Problem, *SIAM J. Comput. 13*, 3 (1984), pp. 441-460. A preliminary version of this paper was presented at the *13th Annual Symposium on Theory of Computing*, 1981.

[29] Mayr, E. and Meyer, A., The Complexity of the Word Problems for Commutative Semigroups and Polynomial Ideals, *Advances in Mathematics 46* (1982), pp. 305-329.

[30] McAloon, K., Petri Nets and Large Finite Sets, *Theoret. Comp. Sci. 32* (1984), pp. 173-183.

[31] Muller, H., Decidability of Reachability in Persistent Vector Replacement Systems, *Proceedings of the 9th Symposium on Mathematical Foundations of Computer Science* , LNCS 88 (1980), pp. 426-438.

[32] Muller, H., Weak Petri Net Computers for Ackermann Functions, to appear in *Elektronische Informationsverarbeitung und Kybernetik.*

[33] Queille, J., and Sifakis, J., Fariness and Related Properties in Transition Systems--A Temporal Logic to Deal with Fairness, *Acta Informatica 19* (1983), 195-220.

[34] Rosier, L. and Yen, H., Logspace Hierarchies, Polynomial Time and the Complexity of Fairness Problems Concerning $\omega$-Machines, *Proceedings of the 3rd Annual Symposium on Theoretical Aspects of Computer Science,* LNCS 210 (1986), pp. 306-320.

[35] Rosier, L. and Yen, H., On the Complexity of Deciding Fair Termination of Probabilistic Concurrent Finite-State Programs, *Proceedings of the 13th*

*International Colloquium on Automata, Languages and Programming*, LNCS 226 (1986), 334-343.

[36] Stockmeyer, L., The Polynomial-Time Hierarchy, *Theoret. Comp. Sci. 3* (1977), 1-22.

[37] Stoer, J. and Witzgall, C., *Convexity and Optimization in Finite Dimensions I,* (Springer Verlag, 1970).

[38] Vardi, M., Automatic Verification of Probablistic Concurrent Finite-State Programs, *Proceedings of the 26th Annual Symposium on Foundations of Computer Science* (1985), 327-338.

[39] Valk, R. and Vidal-Naquet, G., Petri Nets and Regular Languages, *J. of Computer and System Sciences 23* (1981), pp. 299-325.

[40] Von zur Gathen, J., Sieveking, M., A Bound on Solutions of Linear Integer Equalities and Inequalities, *Proc. Amer. Math. Soc. 72*, 1 (1978), 155-158.