Note

Fine covers of a VAS language

Sylviane R. Schwer

C.N.A.M., 292 rue Saint-Martin, 75141 Paris, France

Communicated by M. Nivat
Received June 1990

1. Introduction

In recent years much attention has been paid to parallelism and concurrent systems. Petri nets are a formalism which is commonly used for these studies [8]. The theory of Petri nets can be described in the mathematical frame of vector addition systems (VAS). Although these two approaches are equivalent, we formulate our results here in the VAS formalism.

Karp and Miller provided in [5] a tool which is adopted by everyone who wants to study Petri nets or VAS: the coverability tree. It is usual to study the language associated to Petri nets or VAS. The coverability tree of Karp and Miller allows us to give a rational approximation of this language: from the coverability tree it is easy to derive a finite automaton, called the coverability automaton, accepting a language that contains the VAS language. Unfortunately this inclusion is not always a strict one.

The aim of this note is to provide a constructible refinement of the Karp and Miller automaton: the covering automaton. This automaton recognizes a language included in the one recognized by the coverability automaton, and still containing the language associated to the VAS. This last approximation is the best possible in the sense that if one substitutes to a cycle any finite set of elementary paths then some words of the language are no more accepted by the new automaton.

2. Preliminaries

This paragraph is devoted to the definition of rational cover and of fine cover graph of a language. These notions will be applied to vector addition system (VAS)
languages. Therefore, we give our own definition of a vector addition system which is slightly more precise than the usual one (cf. [3, 6]) in a sense explained below.

Then, we recall that from the coverability tree of Karp and Miller [5], one gets a rational cover of the VAS language associated, which is not a fine cover.

We assume that the reader is familiar with usual notations of formal languages theory [2]. However we give some notations: The empty word is denoted by e; if u is a prefix of v we note \( u \preceq v \); \( \text{Pref}(L) \) is the set of all prefixes of words of the language L.

If \( B = (\mathcal{T}, Q, a, \delta, F) \) is a finite automaton, \( \delta^* \) is the extension of \( \delta \) to \( Q \times \mathcal{T}^* \) by setting: \( \delta^*(q, xf) = \delta^*(\delta(q, x), f) \) for \( f \) in \( \mathcal{T}^* \) and \( x \) in \( \mathcal{T} \), and \( \delta^*(q, e) = q \). For any state \( q \) and any subset \( G \) of \( Q \), the set \( \{ f \in \mathcal{T}^* | \delta^*(q, f) \subseteq G \} \) is denoted \( L(B, q, G) \). \( L(B) = L(B, a, F) \) is the language accepted by B. We omit \( F \) when \( F = Q \). When no final states set is mentioned, an automaton is nothing more than a pointed labelled graph, and we freely use the terminology of graphs.

\( \mathbb{Z} \) and \( \mathbb{N} \) are completed with an element \( \omega \) verifying: for all \( n \in \mathbb{Z} \) [resp. \( \mathbb{N} \)] \( \omega > n \). We use the extensions of + and − from \( \mathbb{Z} \times \mathbb{Z} \) [resp. \( \mathbb{N} \times \mathbb{N} \)] to \( (\mathbb{Z} \cup \{ \omega \} \times \mathbb{Z}) \) [resp. \( \mathbb{N} \cup \{ \omega \} \times \mathbb{N} \)] defined by: for all \( n \in \mathbb{Z} \) [resp. \( \mathbb{N} \)] \( \omega + n = \omega \) and \( \omega - n = \omega \).

2.1. Rational cover of a language and iterable factors

Recall [2] that a rational language over an alphabet \( \mathcal{T} \) is an element of the smallest family of subsets of \( \mathcal{T}^* \) containing all finite subsets and closed under union, product and star operation. Remark that only the star operation enables to get infinite languages.

Definition 2.1.1. Let \( L \) and \( R \) be two languages over \( \mathcal{T} \). \( R \) is a rational cover of \( L \) if and only if

(i) \( R \) is a rational language,
(ii) \( L \subseteq R \).

So a rational cover is a rational approximation of the language \( L \) (from the superior side). If \( L \) is not itself a rational language, there are infinitely many rational covers of \( L \), among which \( \mathcal{T}^* \) is the biggest. It is interesting to try to obtain a better rational approximation, even if we know that there is no best approximation in this sense.

Definition 2.1.2. Let \( L \) be a language over \( \mathcal{T} \) and \( R \) be a rational cover of \( L \). A rational language \( C \) is a refinement of \( R \) if and only if \( L \subseteq C \subseteq R \).

For example, if \( \mathcal{T} = \{a, b, c\} \), \( L = \{a^nb^m | n, m \in \mathbb{N}\} \), \( R_1 = \{a, b, c\}^* \), \( R_2 = a^*b^*c^* \), and \( R_3 = a^*bc^* \), then \( R_3 \) is a refinement of \( R_2 \), itself refinement of \( R_1 \).

The well-known Kleene theorem asserts that a language \( R \) is rational if and only if it is the language recognized by a finite automaton. So we now link the notion of rational cover to finite automata.
Definition 2.1.3. For a language $L$ of $T^*$, a finite automaton $B = (T, Q, a, \delta, F)$ is said to be a cover graph of $L$ if and only if $L(B)$ is a rational cover of $L$, i.e. $L \subseteq L(B)$. A finite automaton $C$ is a refinement of the cover graph $B$ of $L$ if and only if $L \subseteq L(C) \subseteq L(B)$.

Definition 2.1.4. For a language $L$ of $T^*$ and a word $u$ of $T^+$, $u$ is an iterable factor of $L$ if and only if $(\forall n \in \mathbb{N}) (T^* u^n T^* \cap L \neq \emptyset)$.

Definition 2.1.5. Let $B$ be a finite automaton $B = (T, Q, a, \delta, F)$, and $(q, w) \in Q \times T^+$, $(q, w)$ is a loop of $B$ if and only if $\delta^*(q, w) = q$. It is an elementary loop if the cycle going from $q$ to $q$ labelled by $w$ is elementary.

Definition 2.1.6. For a language $L$ of $T^*$, a finite automaton $B = (T, Q, a, \delta, F)$, a state $q$ of $Q$ and a word $u$ of $T^+$, $(q, u)$ is said to be a strong loop of $B$ related to $L$ if and only if $(q, u)$ is a loop, and

$$(\forall n \in \mathbb{N}) \ ((L(B, a, \{q\}) u^n L(B, q, F)) \cap L \neq \emptyset).$$

If $(q, u)$ is a strong loop, $u$ is an iterable factor for $L$, but note that $(q, u)$ may be a loop and $u$ an iterable factor of $L$, without $(q, u)$ being a strong loop. However, we have the following result.

Lemma 2.1.7. If $B = (T, Q, a, \delta, F)$ is a cover graph of $L$, for every iterable factor $u$ of $L$, there is a state $q$ of $Q$ and a positive integer $n$ such that $(q, u^n)$ is a strong loop of $B$ related to $L$.

Definition 2.1.8. $B$ is a fine cover (graph) of $L$ if and only if every loop of $B$ is a strong loop related to $L$.

We also say that $B = (T, Q, a, \delta, F)$ has the loop accessibility property for $L$, i.e.

$$(\forall (q, w) \text{ loop of } B)(\forall n \in \mathbb{N})(\exists f_n \in T^*)(\delta^*(a, f_n) = q)(\delta^*(q, w) = q \text{ and } f_n w^n \in L).$$

Remarks. For the same rational cover $R$ of $L$, there are infinitely many finite automata recognizing $R$. Some of them may be fine covers, and some others may be not. But a fine cover graph is a more accurate description of the approximation of $L$.

For some rational cover $R$ of $L$, there may be no fine cover graph recognizing $R$. In that case, we can say that, in an intuitive way, $R$ cannot be regarded as a good approximation of $L$. It is however important to note that to have a fine cover does not guarantee at all that the rational language recognized is such a good approximation.
Remark. Let \( X = \{x_1, \ldots, x_n\} \). \( B_0 = (X, \{q\}, q, \{(q, x_1, q), \ldots, (q, x_n, q)\}, \{q\}) \), is a finite automaton such that \( L(B_0) = X^* \). An iterable factor \( u \) of \( L \) corresponds to a strong loop \((q, u)\) of \( B_0 \) related to \( L \). So, the notion of strong loop of an automaton and related to a language is a generalization of the notion of iterable factor.

2.2. Vector addition systems and associated languages

We give here a mathematical definition of VAS which allows to have a good distinction between the set of transitions, regarded as the alphabet of the associated language, and the set of vectors, regarded as the set of generators of the VAS. With our definition, two transitions do not have the same label, even if they are associated with the same vector.

**Definition 2.2.1.** A \( k \)-vector addition system \( (k\text{-VAS}) \) is an ordered triple \( A = (T, \varphi, a) \) where \( T \) is an alphabet, \( \varphi \) a monoid morphism from \((T^*, \cdot)\) to \((\mathbb{Z}^k, +)\) and \( a \) is a \( k \)-tuple of \( \mathbb{N}^k \).

**Example.** \( A = (T, \varphi, a) \) with \( T = \{a, b, c\} \) and \( k = 3 \),

\[ \varphi(a) = [-1, 1, 0], \quad \varphi(b) = [0, -1, 1], \quad \varphi(c) = [2, 0, -1], \]
\[ a = [1, 0, 0]. \]

**Definition 2.2.2.** A string \( w \) of \( T^* \) is said to be legal in \( A = (T, \varphi, a) \) if and only if it satisfies the condition:

\[ (\forall v \in T^*)(v \leq w \Rightarrow (a + \varphi(v) \geq 0)). \]

The set of all legal strings in \( A \) is called the \( A \)-language and is denoted by \( L(A) \).

It follows from the definitions that \( L(A) \) is closed by prefixes, so we can also write

\[ L(A) = \{w \in T^* \mid a + \varphi(\text{Pref}(w)) \in \mathbb{N}^k\}. \]

**Example (continued).** \( abca \) is a legal string in \( A \), but \( abac \) is not a legal string because \( a + \varphi(aba) = [-1, 1, 1] \notin \mathbb{N}^k \). Notice that, however, \( \varphi(abac) = \varphi(abca) \).

In the sequel, we will denote \( \|u\|^- \) (resp. \( \|u\|^+ \)) the set of nonpositive coordinates (resp. nonnegative) of \( \varphi(u) \). We denote by \( 0 \) the \( k \)-tuple with all its coordinates equal to zero.

The coverability tree of Karp and Miller [5] enables to provide a nonambiguous cover graph of \( L(A) \), called coverability automaton of \( A \).
Example (continued). Recall that a cover graph of Karp and Miller has the following properties [8].

Proposition 2.2.3. Let \( A = (T, \varphi, a) \) be a \( k \)-VAS and \( G(A) = (T, Q, a, \delta) \) its coverability automaton, then

(i) \( L(A) \subseteq L(G(A)) \)

(ii) \( (\forall (q, w) \in Q \times T^*)(\delta(q, w) = q \Rightarrow [(\forall n \in \mathbb{N})(\exists f_n \in T^*)(f_n w^n \in L(A))]) \).

(iii) If there is \( q \) in \( Q \) such that \( q[p] = \omega \), then the coordinate \( p \) is not bounded, i.e.
\[
(\forall n \in \mathbb{N})(\exists u \in L(A))(a[p] + \varphi(u)[p] \geq n).
\]

Example (continued). The coverability automaton is not a fine cover of \( L(A) \) because we have \( L(A) \subseteq \text{Pref}(abcaab{a, b, c}^*) \) and \( L(G(A)) = \text{Pref}(abcaab{a, b, c}^*) \).

\([\omega, \omega, 0], a)\) is a loop, but not a strong loop: if \( (\omega, \omega, 0) \) and \( (\omega, \omega, 0) = \omega \omega \omega \), then \( w \in abca^* \), but \( abca^* \cap L(A) = \{abca, abca\} \). Hence for \( n > 1 \), there is no \( f \) in \( L(A) \) such that \( \delta([1, 0, 0], f) = [\omega, \omega, 0] \) and \( fa^n \in L(A) \).

\([\omega, \omega, 0], a)\) is a strong loop because \( (\forall n \in \mathbb{N}), (\delta([1, 0, 0], (abc)^n)) = [\omega, \omega, 0]) \) and \( (abc)^n a^n \in L(A) \).

For the coverability automaton \( G(A) \) of a \( k \)-VAS \( A \), we have a stronger property than in Lemma 2.1.7.

Lemma 2.2.4. If \( (q, w) \) is a loop of \( G(A) \), then there is \( q' \in Q \) such that \( (q', w) \) is a strong loop in \( G(A) \) related to \( L(A) \).

Proof. Let \( (q, w) \) be a loop; from Proposition 2.2.3(ii), we have: \( (\forall n \in \mathbb{N})(\exists f_n \in T^*) \) such that \( f_n w^n \in L(A) \). So there is \( q' \in Q \) such that for infinitely many \( n \), \( \delta(a, f_n) = q' \). There is an integer \( r \) such that \( (q', w') \) is a strong loop related to \( L(A) \).

If \( r > 1 \) then there is a state \( q'' \) such that \( q' + \varphi(w') = q'' \) and \( q'' + (r - 1)\varphi(w') = q' \) and \( q'' + r\varphi(w') = q' \). As a bounded coordinate of \( q' \) is necessarily null, \( q'' = q' \).

Corollary 2.2.5. For every iterable factor \( w \) of \( L(A) \), there is a strong loop \( (q, w) \) in \( G(A) \).

3. Iterating systems

In order to prove that a loop is a strong loop, we introduce a new tool: the iterating system's notion. We prove in this section that a loop is a strong loop if and only if there exists an iterating system related to this loop.
Definition 3.1. Let $A = (T, \varphi, a)$ be a $k$-VAS and $G(A) = (T, Q, q_0, \delta)$ its coverability automaton with $q_0 = a$. Let $(q, w)$ be a loop. Then an iterating system of length $p$ related to $(q, w)$ is a finite sequence

$$\mathcal{J}(q, w) = (\alpha_0, q_1, u_1, \alpha_1, q_2, \ldots, \alpha_{p-1}, q_p, u_p, q_p, a_p, q)$$

with:

(i) for $0 \leq i < p$, $\alpha_i \in T^*$, $q_i \in Q$, and for $1 \leq i < p$, $u_i \in T^+$;

(ii) for $0 \leq i < p-1$, $\delta(q_i, \alpha_i) = q_{i+1}$; $\delta(q_p, \alpha_p) = q$, and for $1 \leq i < p$, $\delta(q_i, u_i) = q$;

(iii) $\alpha_0 u_1^+ \alpha_1 u_2^+ \ldots \alpha_{p-1} u_p^+ a_p w \cap L(A) \neq \emptyset$;

satisfying the following property:

$$\|u_i\|^- \leq \bigcup_{0 \leq j < i} \|u_j\|^+ \quad \text{and} \quad \|w\|^- \leq \bigcup_{0 \leq j < p} \|u_j\|^+,$$

where $1 \leq i \leq p$.

The final formulation of this definition took advantage of some remarks of Gilleron [1].

Lemma 3.2. The property involved in Definition 3.1 is equivalent to the following assertion: there are $\tau_1, \ldots, \tau_p \in \mathbb{N}$ such that, for $1 \leq i < p$,

$$\varphi(u_1^+ u_2^+ \ldots u_p^+ u_{i+1}) \geq 0 \quad \text{and} \quad \varphi(u_1^+ u_2^+ \ldots u_p^+ w) \geq 0.$$ 

Lemma 3.3. If $(\alpha_0, q_1, u_1, \alpha_1, q_2, \ldots, \alpha_{p-1}, q_p, u_p, q_p, a_p, q)$ is an iterating system related to $(q, w)$, then

$$(\forall n \geq 0)(\text{Card}(\alpha_0 u_1^+ \alpha_1 u_2^+ \ldots \alpha_{p-1} u_p^+ a_p w^n \cap L(A)) = \infty).$$

Let us first give a characterization of a strong positive loop.

Lemma 3.4. Let $(q, w)$ be a loop, then $(q, w)$ is a strong positive loop if and only if there is an iterating system of length 0 related to it.

Proof. Suppose that $(\alpha_0, q)$ is an iterating system of length 0 related to $w$. By definition, $\varphi(w) \geq 0$ and as $\alpha_0 w \in L(A)$ then $\alpha_0 w^k \subseteq L(A)$ i.e. $(q, w)$ is a positive loop which is a strong loop.

Suppose now that $(q, w)$ is a strong positive loop, then there exists $\alpha_0 \in T^*$ such that $\delta(a, \alpha_0) = q$ and $\alpha_0 w \subseteq L(A)$; as $\varphi(w) \geq 0$, $\alpha_0 w^k \subseteq L(A)$ i.e. $(\alpha_0, \delta(a, \alpha_0))$ is an iterating system of length 0 related to $w$. □

We are now ready to prove the key proposition of our paper.

Proposition 3.5. Let $A = (T, \varphi, a)$ be a $k$-VAS and $G(A) = (T, Q, a, \delta)$ its coverability automaton. Let $(q, w)$ be in $Q \times T^+$. Then the two following statements are equivalent:

(i) $(q, w)$ is a strong loop,

(ii) there exists an iterating system related to $(q, w)$. 

Sketch of the proof. (ii) ⇒ (i): follows from Lemma 3.3.

(i) ⇒ (ii): Suppose that \((q, w)\) is a strong loop, then by Definition 2.1.5

\[(\forall n \in \mathbb{N})(\exists f_n \in T^*)(\delta(a, f_n) = q)(\delta(q, w) = q \text{ and } f_n w^n \in L(A)).\]

We choose for each \(n\) a word \(f_n\) of minimal length.

There are two possibilities for the infinite sequence \((|f_n|)_{n \geq 0}\) of \(\mathbb{N}\): either it is bounded or not.

Case 1: \((|f_n|)_{n \geq 0}\) is bounded. As the words \(f_n\) are written on a finite alphabet \(T\), the same word \(f\) may be chosen infinitely often:

\[(\exists f \in T^*)(\forall n \in \mathbb{N})(\exists m \in \mathbb{N})(m \geq n)(f w^m \in L(A)),\]

which implies that:

\[(\exists f \in T^*)(\forall n \in \mathbb{N})(\exists m \in \mathbb{N})(m \geq n)(a + \varphi(f) + mw(w) \geq 0).\]

This implies \(\varphi(w) \geq 0\), hence there is an iterating system of length 0 related to \((q, w)\) as shown in Lemma 3.4.

Case 2: \((|f_n|)_{n \geq 0}\) is unbounded. Hence there is a strictly increasing subsequence included. Without loss of generality, we can assume that \(|f_n| < |f_{n+1}|\) and \((T\) is a finite set) \(f_n = h_\epsilon e_n\) with the following property:

\[(\forall n \geq 0)(|h_n| = n)((h_{n+1} = h_n x_n + 1, \text{ with } x_{n+1} \in T)).\]

The sequence \((\varphi(h_n))_{n \geq 0}\) of \(\mathbb{N}^k\) is infinite, so there exists \((n_0, n_1)\) with \(n_0 < n_1\) such that \(n_0\) is the minimal integer satisfying \(\varphi(h_{n_0}) \leq \varphi(h_{n_1})\). Let us set \(a_0 = h_{n_0} u_1 = x_{n_0 + 1} \ldots x_{n_1}\). Then \(a_0 u_1 = h_n\) and \(\varphi(u_1) > 0\), hence \((\delta(a, a_0), u_1)\) is a positive loop.

If \(\|w\| \leq \|u_i\|^{+}\), then the proposition is proved with the iterating system \((h_{n_0}, \delta(a, h_{n_0}), u_1, \delta(a, h_{n_0}), e_n, q)\).

If not, we have the following property:

\[f \in L(G(A), \delta(a, h_{n_0})) \quad \forall p \in \|u_i\| \exists r > 0 |\varphi(u_i)[p] > 0|\]

Hence the only problem remains in \((\|k\| \|u_i\|^{+})\) in which there is some negative coordinate of \(\|w\|^{-}\). We can do the same reasoning related to \((\|k\| \|u_i\|^{+})\) and the part of the coverability tree with root \(\delta(a, h_{n_0})\).

As

\[\text{Card}(\prod_{i \in [k] \setminus \|u_i\|^{+}}(\|w\|^{-})) < \text{Card}(\|w\|^{-})\]

we only have to repeat the same argument a finite number of times in order to eliminate \(\|w\|^{-}\). The number of times we repeat the argument gives the length of \(\delta(q, w)\).

4. Fine cover of a VAS language

We show here that we can refine Karp and Miller coverability automaton in an automaton in which all loops are strong loops, i.e. which is a fine cover of the VAS
language. This construction is due to the fact that it is decidable whether or not a loop is strong, a property proved using the notion of iterating system.

**Proposition 4.1.** Let \( A = (T, \varphi, a) \) be a \( k \)-VAS and \( G(A) = (T, Q, a, \delta) \) its coverability automaton. Let \((q, w)\) be in \( Q \times T^+ \). One can decide whether or not \((q, w)\) is a strong loop of \( G(A) \).

**Sketch of the proof.** From Proposition 3.5, it is the same problem to decide if there exists an iterating system

\[
J(q, w) = (a_0, q_1, u_1, \alpha_1, q_2, \ldots, \alpha_{p-1}, q_p, u_p, \alpha_p, q)
\]

of length \( p \) related to \((q, w)\). It is easy to see that it is enough to check it for iterating systems with \( p \leq k \). This is achieved the following way: Take one state \( q_1 \), the existence of \( u_1 \) such that \( \delta(q_1, u_1) = q_1 \) satisfying \( \|u_1\|^- = 0 \) is the problem of the existence of elementary loops labelled \( x_1, x_2, \ldots, x \), such that \( \sum_{1 \leq i \leq t} \lambda_i \cdot \varphi(x_i) \geq 0 \) with \( u_1 \) in the shuffle of \( \{x_i^+ \mid 1 \leq i \leq t\} \) (and \( \varphi(u_1) = \sum_{1 \leq i \leq t} \lambda_i \cdot \varphi(x_i) \geq 0 \)). As there is a finite number of elementary loops, the decidability of this problem is due to the decidability of the problem of the existence of a positive solution to a finite set of integer linear inequations \([7]\). One has now only to care about the coordinates \( j \) that are not in \( \|u_1\|^+ \). Then take a state \( q_2 \) in \( \text{Acc}(q_1) \), and do the same procedure, and so on. As \( p \leq k \), there is only a finite number of sequences \( q_1, q_2, \ldots, q_p \) to check. \( \square \)

**Proposition 4.2.** Let \( A = (T, \varphi, a) \) be a \( k \)-VAS and \( G(A) \) its coverability automaton. There is an automaton \( C(A) = (T, Q, a, \delta) \), that can be constructed from the coverability automaton \( G(A) \), satisfying the refinement property

\[
[L(A) \subseteq L(C(A)) \subseteq L(G(A))]
\]

and the loop accessibility property

\[
[(\forall (q, w) \subseteq Q \times T^+)(\forall n \in \mathbb{N})(\exists f_n \in T^*)(\delta(a, f_n) = q)(\delta(q, w) = q \text{ and } f_n w^n \in L(A))].
\]

**Sketch of the proof.** If \((q, w)\) is not a strong loop, then there is an elementary loop which is not a strong loop. So it is enough to check only elementary loops.

Let \((q, w)\) be an elementary loop, if \((q, w)\) is not a strong loop, then there is an integer \( I \) such that

\[
I = \min \{ (\forall f \in L(A), \delta(a, f) = q \text{ and } f w^n \in L(A)) \Rightarrow (n \leq s) \}.
\]

For such a \((q, w)\), we substitute to the cycle containing \( q \) and labelled by \( wI + 1 \) simple paths labelled by \( w^0, \ldots, w^I \) with initial vertex \( q_0 \) and final vertex \( q_I \) (\( I + 1 \) duplications of \( q \)). Each state of the cycle will be duplicated \( I + 1 \) times.

For the other paths of the automaton, the transition function will be extended in the obvious way, every time a state of the cycle appears in a transition, then this
transition will be duplicated $I + 1$ times, each duplication will contain a different
duplication of this state; the initial transition will be removed at the end like all the
states of the removed cycle except for those which are included in a simple path
not in the cycle. □

Example (continued). In our example, there is only one elementary loop
$([w, w, 0], a)$ which is not a strong loop.

This finite automaton is a fine cover of $L(A)$.

As a corollary, we get the following theorem.

**Theorem 4.3.** Let $A$ be a $k$-VAS. It is possible to construct a fine cover of $L(A)$,
refinement of Karp and Miller's coverability automaton.

5. Conclusion

In this paper, we have introduced the notion of rational cover of a language,
which was implicit in the literature: bounded languages are by definition languages
covered by a rational with star high one [2] and the coverability automaton of a
VAS recognizes a cover of the associated language [5]. The notion of fine cover is
a natural one which can be studied for its own sake, and not necessarily in the area
of VAS languages. For example, it would be interesting to study fine covers of
context-free languages.

Concerning VAS languages, we proved that we can build an automaton which is
a fine cover of a VAS language, recognizing a language included in the language
recognized by the coverability automaton of Karp and Miller. We propose to call
this automaton the covering automaton of the VAS language.

This has been proved with the notion of loop accessibility property and iterating
systems. This proof is based on a decidability property of integer inequalities. This
problem, which is in general of exponential time complexity, falls down here to a
polynomial complexity since the number of elementary loops and of coordinates
are fixed [7]. Iterating systems can be used for various purposes: it is a notion that
we believed to be very well suited to VAS languages. For instance, the rationality
of a VAS language can be expressed in the following terms: for every loop there is
an iterating system of length 0 related to it (Lemma 5.4) [10, 12, 13]. Other examples
can be found in our proof of the decidability of context-freeness of a VAS language
[9], where several necessary conditions can be easily expressed with this tool [11].
References