Forward Analysis for WSTS, Part II: Complete WSTS

Alain Finkel[1] and Jean Goubault-Larrecq[1,2]

[1] LSV, ENS Cachan, CNRS, France
finkel@lsv.ens-cachan.fr
[2] INRIA Saclay, France
goubault@lsv.ens-cachan.fr

Abstract. We describe a simple, conceptual forward analysis procedure for ∞-complete WSTS $\mathfrak{S}$. This computes the *clover* of a state s_0, i.e., a finite description of the closure of the cover of s_0. When $\mathfrak{S}$ is the completion of a WSTS $\mathfrak{X}$, the clover in $\mathfrak{S}$ is a finite description of the cover in $\mathfrak{X}$. We show that this applies exactly when $\mathfrak{X}$ is an ω^2-*WSTS*, a new robust class of WSTS. We show that our procedure terminates in more cases than the generalized Karp-Miller procedure on extensions of Petri nets. We characterize the WSTS where our procedure terminates as those that are *clover-flattable*. Finally, we apply this to well-structured counter systems.

1 Introduction

Context. Well-structured transition systems (WSTS) are a general class of infinite-state systems where coverability—given states s, t, decide whether $s\ (\geq; \rightarrow^*; \geq)\ t$, i.e., whether $s \geq s_1 \rightarrow^* t_1 \geq t$ for some s_1, t_1—is decidable, using a simple backward algorithm [14,15,19,2].

The starting point of this paper and of its first part [17] is our desire to derive similar *forward* algorithms, namely algorithms computing the *cover* $\downarrow Post^*(\downarrow s)$ of s. While the cover allows one to decide coverability as well, by testing whether $t \in \downarrow Post^*(\downarrow s)$, it can also be used to decide U-boundedness, i.e., to decide whether there are only finitely many states t in the upward-closed set U and such that $s\ (\geq; \rightarrow^*)\ t$. No backward algorithm can decide this. In fact, U-boundedness is undecidable in general, e.g., on lossy channel systems [9]. So the reader should be warned that computing the cover is not possible for general WSTS. Despite this, the known forward algorithms are felt to be more efficient than backward procedures in general: e.g., for lossy channel systems, although the backward procedure always terminates, only the non-terminating forward procedure is implemented in the tool TREX [1].

State of the art. Karp and Miller [27] proposed an algorithm, for Petri nets, which computes a finite representation of the *cover*, i.e., of the downward closure of the reachability set of a Petri net. Finkel [14,15] introduced the WSTS framework and generalized the Karp-Miller procedure to a class of WSTS. This was achieved by building a non-effective completion of the set of states, and replacing ω-accelerations of increasing sequences of states (in Petri nets) by least upper bounds (lub). In [12,15] a variant of this generalization of the Karp-Miller procedure was studied; but no guarantee was given

S. Albers et al. (Eds.): ICALP 2009, Part II, LNCS 5556, pp. 188–199, 2009.

that the cover could be represented finitely. There were no effective finite representations of downward closed sets in [15]. Finkel [16] modified the Karp-Miller algorithm to reduce the size of the intermediate computed trees. [22] recently proposed a weaker acceleration, which avoid some possible underapproximations in [16]. Emerson and Namjoshi [12] took into account the labeling of WSTS for adapting the generalised Karp-Miller algorithm to model-checking. They assume the existence of a compatible cpo, and proved that for broadcast protocols (which are equivalent to transfer Petri nets), the Karp-Miller procedure can be generalized. However, termination is then not guaranteed [13], and in fact neither is the existence of a finite representation of the cover. Abdulla, Colomb-Annichini, Bouajjani and Jonsson proposed a forward procedure for lossy channel systems [3] using downward closed regular languages as symbolic representations. Ganty, Geeraerts, Raskin and Van Begin [21,20] proposed a forward procedure for solving the coverability problem for WSTS equipped with an effective adequate domain of limits, or equipped with a finite set D used as a parameter to tune the precision of an abstract domain. Both solutions insure that every downward closed set has a finite representation. Abdulla *et al.* [3] applied this framework to Petri nets and lossy channel systems. Abdulla, Deneux, Mahata and Nylén proposed a symbolic framework for dealing with downward closed sets for Timed Petri nets [4].

Our contribution. First, we define *complete WSTS* as WSTS whose well-ordering is also a continuous dcpo. This allows us to design a conceptual procedure **Clover**$_{\mathfrak{S}}$ that looks for a finite representation of the downward closure of the reachability set, i.e., of the cover [15]. We call such a finite representation a *clover* (for *clo*sure of *cover*). This clearly separates the fundamental ideas from the data structures used in implementing Karp-Miller-like algorithms. Our procedure also terminates in more cases than the well-known (generalized) Karp-Miller procedure [12,15]. We establish the main properties of clovers in Section 3 and use them to prove **Clover**$_{\mathfrak{S}}$ correct, notably, in Section 5.

Second, we characterize complete WSTS for which **Clover**$_{\mathfrak{S}}$ terminates. These are the ones that have a (continuous) flattening with the same clover. This establishes a surprising relationship with the theory of flattening [8].

Third, and building on our theory of completions [17], we characterize those WSTS whose completion is a complete WSTS in the sense above. They are exactly the ω^2-*WSTS*, i.e., those whose state space is ω^2-wqo, as we show in Section 4.

Finally, we apply our framework of complete WSTS to counter systems in Section 6. We show that affine counter systems may be completed into ∞-complete WSTS iff the domains of the monotone affine functions are upward closed.

2 Preliminaries

Posets, dcpos. We borrow from theories of order, as used in model-checking [2,19], and also from domain theory [6,23]. A *quasi-ordering* $\leq$ is a reflexive and transitive relation on a set X. It is a (partial) *ordering* iff it is antisymmetric.

We write $\geq$ the converse quasi-ordering, $<$ the associated strict ordering ($\leq \setminus \geq$), and $>$ the converse ($\geq \setminus \leq$) of $<$. A set X with a partial ordering $\leq$ is a *poset* $(X, \leq)$, or just X when $\leq$ is clear. The *upward closure* $\uparrow E$ of a set E is $\{y \in X \mid \exists x \in E \cdot x \leq y\}$. The *downward closure* $\downarrow E$ is $\{y \in X \mid \exists x \in E \cdot y \leq x\}$. A subset E of X is *upward closed* if and only if $E = \uparrow E$. *Downward closed* sets are defined similarly. A downward closed (resp. upward closed) set E has a *basis* A iff $E = \downarrow A$ (resp. $E = \uparrow A$); E has a *finite basis* iff A can be chosen finite.

A quasi-ordering is *well-founded* iff it has no infinite strictly descending chain $x_0 > x_1 > \ldots > x_i > \ldots$. An *antichain* is a set of pairwise incomparable elements. A quasi-ordering is *well* iff it is well-founded and has no infinite antichain. We abbreviate well posets as *wpos*.

An *upper bound* $x \in X$ of $E \subseteq X$ is such that $y \leq x$ for every $y \in E$. The *least upper bound (lub)* of a set E, if it exists, is written $lub(E)$. An element x of E is *maximal* (resp. minimal) iff $\uparrow x \cap E = \{x\}$ (resp. $\downarrow x \cap E = \{x\}$). Write $\mathrm{Max}\, E$ (resp. $MinE$) the set of maximal (resp. minimal) elements of E.

A *directed subset* of X is any non-empty subset D such that every pair of elements of D has an upper bound in D. Chains, i.e., totally ordered subsets, and one-element set are examples of directed subsets. A *dcpo* is a poset in which every directed subset has a least upper bound. For any subset E of a dcpo X, let $\mathrm{Lub}(E) = \{lub(D) \mid D \text{ directed subset of } E\}$. Clearly, $E \subseteq \mathrm{Lub}(E)$; $\mathrm{Lub}(E)$ can be thought of E plus all limits from elements of E.

The *way below* relation $\ll$ on a dcpo X is defined by $x \ll y$ iff, for every directed subset D such that $lub(D) \leq y$, there is a $z \in D$ such that $x \leq z$. Write $\twoheaddownarrow E = \{y \in X \mid \exists x \in E \cdot y \ll x\}$. X is *continuous* iff, for every $x \in X$, $\twoheaddownarrow x$ is a directed subset, and has x as least upper bound.

When $\leq$ is a well partial ordering that also turns X into a dcpo, we say that X is a *directed complete well order*, or *dcwo*. If additionally X is continuous, we say that X is a *cdcwo*.

A subset U of a dcpo X is (Scott-)*open* iff U is upward-closed, and for any directed subset D of X such that $lub(D) \in U$, some element of D is already in U. A map $f : X \to X$ is (Scott-)*continuous* iff f is monotonic ($x \leq y$ implies $f(x) \leq f(y)$) and for every directed subset D of X, $lub(f(D)) = f(lub(D))$. Equivalently, f is continuous in the topological sense, i.e., $f^{-1}(U)$ is open for every open U.

A *closed* set is the complement of an open set. Every closed set is downward closed. The *closure* $cl(A)$ of $A \subseteq X$ is the smallest closed set containing A. This should not be confused with the *inductive closure* $\mathrm{Ind}(A)$ of A, which is obtained as the least set B containing A and such that $\mathrm{Lub}(B) = B$. In general, $\downarrow A \subseteq \mathrm{Lub}(\downarrow A) \subseteq \mathrm{Ind}(\downarrow A) \subseteq cl(A)$, and all inclusions can be strict. However, when X is a *continuous* dcpo, and A is downward closed in X, $\mathrm{Lub}(A) = \mathrm{Ind}(A) = cl(A)$. (See, e.g., [17, Proposition 3.5].)

Well-Structured Transition Systems. A *transition system* is a pair $\mathfrak{S} = (S, \to)$ of a set S, whose elements are called *states*, and a *transition relation* $\to \subseteq S \times S$. We write $s \to s'$ for $(s, s') \in \to$. Let $\xrightarrow{*}$ be the transitive and reflexive closure of the relation $\to$. We write $Post_{\mathfrak{S}}(s) = \{s' \in S \mid s \to s'\}$ for the set of immediate successors of the state s. The *reachability set* of a transition system $\mathfrak{S} = (S, \to)$ from an initial state s_0 is $Post^*_{\mathfrak{S}}(s_0) = \{s \in S \mid s_0 \xrightarrow{*} s\}$.

A transition system $(S, \to)$ is *effective* iff S is r.e., and for every state s, $Post_{\mathfrak{S}}(s)$ is finite and computable. An *ordered* transition system is a triple $\mathfrak{S} = (S, \to, \leq)$ where $(S, \to)$ is a transition system and $\leq$ is a quasi-ordering on S. We say that $(S, \to, \leq)$ is *effective* if $(S, \to)$ is effective and if $\leq$ is decidable.

$\mathfrak{S} = (S, \rightarrow, \leq)$ is *monotone* (resp. *strictly monotone*) iff for every $s, s', s_1 \in S$ such that $s \rightarrow s'$ and $s_1 \geq s$ (resp. $s_1 > s$), there exists an $s'_1 \in S$ such that $s_1 \xrightarrow{*} s'_1$ and $s'_1 \geq s'$ (resp. $s'_1 > s'$). $\mathfrak{S}$ is *strongly monotone* iff for every $s, s', s_1 \in S$ such that $s \rightarrow s'$ and $s_1 \geq s$, there exists an $s'_1 \in S$ such that $s_1 \rightarrow s'_1$ and $s'_1 \geq s'$.

Finite representations of $Post_{\mathfrak{S}}(s)$,e.g., as Presburger formulae or finite automata, usually don't exist even for monotone transition systems (not even speaking of being computable). The *cover* $Cover_{\mathfrak{S}}(s) = \downarrow Post^*_{\mathfrak{S}}(\downarrow s)$ $(= \downarrow Post^*_{\mathfrak{S}}(s)$ when $\mathfrak{S}$ is monotone) is better behaved. Note that being able to compute the cover allows one to decide *coverability*: $s\ (\geq; \rightarrow^*; \geq)\ t$ iff $t \in Cover_{\mathfrak{S}}(s)$. In most cases we shall encounter, it will also be decidable whether a finitely represented cover is finite, or whether it meets a given upward closed set U in only finitely many points. Therefore *boundedness* (is $Post^*_{\mathfrak{S}}(s)$ finite?) and *U-boundedness* (is $Post^*_{\mathfrak{S}}(s) \cap U$ finite?) will be decidable, too.

An ordered transition system $\mathfrak{S} = (S, \rightarrow, \leq)$ is a *Well Structured Transition System* (*WSTS*) iff $\mathfrak{S}$ is monotone and $(S, \leq)$ is wpo. This is our object of study.

For strictly monotone WSTS, it is also possible to decide the boundedness problem, with the help of the Finite Reachability Tree (FRT) [15]. However, the U-Boundedness problem (called the place-boundedness problem for Petri nets) remains undecidable for strictly monotone WSTS (for instance, for transfer Petri nets), but it is decidable for Petri nets. It is decided with the help of a richer structure than the FRT, the Karp-Miller tree. The set of labels of the Karp-Miller tree is a finite representation of the cover.

We will consider transition systems defined by a finite set of transition functions for simplicity. This is as in [17]. Formally, a *functional transition system* $(S, \xrightarrow{F})$ is a labeled transition system where the transition relation $\xrightarrow{F}$ is defined by a finite set F of partial functions $f : S \longrightarrow S$, in the sense that for every $s, s' \in S$, $s \xrightarrow{F} s'$ iff $s' = f(s)$ for some $f \in F$. A map $f : S \rightarrow S$ is *partial monotonic* iff $\operatorname{dom} f$ is upward-closed and for all $x, y \in \operatorname{dom} f$ with $x \leq y$, $f(x) \leq f(y)$. An *ordered functional transition system* is an ordered transition system $\mathfrak{S} = (S, \xrightarrow{F}, \leq)$ where F consists of partial monotonic functions. This is always strongly monotonic. A *functional WSTS* is an ordered functional transition system where $\leq$ is well.

A functional transition system $(S, \xrightarrow{F})$ is *effective* if every $f \in F$ is computable: given a state s and a function f, we may decide whether $s \in \operatorname{dom} f$ and in this case, one may also compute $f(s)$.

3 Clovers of Complete WSTS

Complete WSTS and their clovers. All forward procedures for WSTS rest on completing the given WSTS to one that includes all limits. E.g., the state space of Petri nets is $\mathbb{N}^k$, the set of all markings on k places, but the Karp-Miller algorithm works on $\mathbb{N}^k_\omega$, where $\mathbb{N}_\omega$ is $\mathbb{N}$ plus a new top element ω. We have defined general completions of wpos, serving as state spaces, and have briefly described completions of (functional) WSTS in [17]. We temporarily abstract away from this, and consider *complete* WSTS directly.

Generalizing the notion of continuity to partial maps, a *partial continuous* map $f : X \rightarrow X$, where $(X, \leq)$ is a dcpo, is such that $\operatorname{dom} f$ is open (not just upward-closed), and for every directed subset D *in* $\operatorname{dom} f$, $lub(f(D)) = f(lub(D))$.

Equivalently, $\mathrm{dom}\, f$ is open and $f^{-1}(U)$ is open for any open U. The composite of two partial continuous maps is again partial continuous.

Definition 1. *A* complete *WSTS is a (functional) WSTS $\mathfrak{S} = (S, \xrightarrow{F}, \leq)$ where $(S, \leq)$ is a cdcwo and every function in F is partial continuous.*

The point in complete WSTS is that one can *accelerate* loops:

Definition 2. *Let $(X, \leq)$ be a dcpo, $f : X \to X$ be partial continuous. The* lub-acceleration *$f^\infty : X \to X$ is defined by: $\mathrm{dom}\, f^\infty = \mathrm{dom}\, f$, and for any $x \in \mathrm{dom}\, f$, if $x < f(x)$ then $f^\infty(x) = lub\{f^n(x) \mid n \in \mathbb{N}\}$, else $f^\infty(x) = f(x)$.*

Note that if $x \leq f(x)$, then $f(x) \in \mathrm{dom}\, f$, and $f(x) \leq f^2(x)$. By induction, we can show that $\{f^n(x) \mid n \in \mathbb{N}\}$ is an increasing sequence, so that the definition makes sense.

Complete WSTS are strongly monotone. One may not decide, in general, whether a recursive function f is monotone [18] or continuous, whether an ordered set $(S, \leq)$ with a decidable ordering $\leq$, is a dcpo or whether it is a wpo. We may prove that given an effective ordered functional transition system, one cannot decide whether it is a WSTS, or a complete WSTS. However, the completion of *any* functional ω^2-WSTS is complete, as we shall see in Theorem 1.

In a complete WSTS, there is a *canonical* finite representation of the cover:

Definition 3 (Clover). *Let $\mathfrak{S} = (S, \xrightarrow{F}, \leq)$ be a complete WSTS. The* clover *$Clover_{\mathfrak{S}}(s_0)$ of the state $s_0 \in S$ is $\mathrm{Max}\,\mathrm{Lub}(Cover_{\mathfrak{S}}(s_0))$.*

Proposition 1. *Let $\mathfrak{S} = (S, \xrightarrow{F}, \leq)$ be a complete WSTS, and $s_0 \in S$. Then $Clover_{\mathfrak{S}}(s_0)$ is finite, and $cl(Cover_{\mathfrak{S}}(s_0)) = \downarrow Clover_{\mathfrak{S}}(s_0)$.*

Proof. $\mathrm{Lub}(Cover_{\mathfrak{S}}(s_0)) = cl(Cover_{\mathfrak{S}}(s_0))$ since $Cover_{\mathfrak{S}}(s_0)$ is downward closed, and S is a continuous dcpo. Since S is a wpo, it is Noetherian in its Scott topology [25, Proposition 3.1]. Since S is a continuous dcpo, S is also sober [6, Proposition 7.2.27], so Corollary 6.5 of [25] applies: every closed subset F of S is such that $\mathrm{Max}\, F$ is finite and $F = \downarrow \mathrm{Max}\, F$. Now let $F = \mathrm{Lub}(Cover_{\mathfrak{S}}(s_0))$. □

For any other representative, i.e., for any finite set R such that $\downarrow R = \downarrow Clover_{\mathfrak{S}}(s_0)$, $Clover_{\mathfrak{S}}(s_0) = \mathrm{Max}\, R$. Indeed, for any two finite sets $F, G \subseteq S$ such that $\downarrow F = \downarrow G$, $\mathrm{Max}\, F = \mathrm{Max}\, G$. So *Clover* is the *minimal representative* of the cover, i.e., there is no representative R with $|R| < |Clover_{\mathfrak{S}}(s_0)|$. The clover was called the minimal coverability set in [16].

Despite the fact that the clover is always finite, it is non-computable in general (see Proposition 4 below). Nonetheless, it is computable on *flat* complete WSTS, and even on the larger class of *clover-flattable* complete WSTS (Theorem 3 below).

Completions. There are numberous WSTS which are not complete: the set $\mathbb{N}^k$ of states of a Petri net with k places is not even a dcpo. The set of states of a lossy channel system with k channels, $(\Sigma^*)^k$, is not a dcpo for the subword ordering either. We have defined general completions of wpos, and of WSTS, in [17], which we recall quickly.

The *completion* $\widehat{X}$ of a wpo $(X, \leq)$ is defined in any of two equivalent ways. First, $\widehat{X}$ is the *ideal completion* $Idl(X)$ of X, i.e., the set of ideals of X, ordered by inclusion, where an *ideal* is a downward-closed directed subset of X. This can also be described as the sobrification $\mathcal{S}(X_a)$ of the Noetherian space X_a, but this is probably harder to understand (although it makes proofs simpler). We consider X as a subset of $\widehat{X}$, by equating each element $x \in X$ with $\downarrow x \in \mathit{Idl}(X)$. For instance, if $X = \mathbb{N}^k$, e.g., with $k = 3$, then $(1, 3, 2)$ is equated with the ideal $\downarrow(1, 3, 2)$, while $\{(1, m, n) \mid m, n \in \mathbb{N}\}$ is a *limit*, i.e. an element of $\widehat{X} \setminus X$; the latter is usually written $(1, \omega, \omega)$, and is the least upper bound of all $(1, m, n)$, $m, n \in \mathbb{N}$. The downward-closure of $(1, \omega, \omega)$ in $\widehat{X}$, intersected with X, gives back the set of non-limit elements $\{(1, m, n) \mid m, n \in \mathbb{N}\}$.

This is a general situation: one can always write $\widehat{X}$ as the disjoint union $X \cup L$, so that any downward closed subset D of X can be written as $X \cap \downarrow A$, where A is a *finite* subset of $X \cup L$. Then L, the set of limits, is a *weak adequate domain of limits* (WADL) for X—we slightly simplify Definition 3.1 of [17], itself a slight generalization of [21]. In fact, $\widehat{X}$ (minus X) is the *smallest* WADL [17, Theorem 3.4].

$\widehat{X} = \mathit{Idl}(X)$ is always a continuous dcpo. In fact, it is even algebraic [6, Proposition 2.2.22]. It may however fail to be well, hence to be a cdcwo, see Lemma 1 below.

We have also described a hierarchy of datatypes on which completions are effective [17, Section 5]. Notably, $\widehat{\mathbb{N}} = \mathbb{N}_\omega$, $\widehat{A} = A$ for any finite poset, and $\widehat{\prod_{i=1}^k X_i} = \prod_{i=1}^k \widehat{X_i}$. Also, $\widehat{X^*}$ is the space of *products* on X, as defined in [1], i.e., regular expressions that are products of *atomic expressions* A^* ($A \in \mathbb{P}_{\text{fin}}(\widehat{X})$, where $\mathbb{P}_{\text{fin}}$ denotes the set of *finite* subsets) or $a^?$ ($a \in \widehat{X}$). In any case, elements of completions $\widehat{X}$ have a finite description, and the ordering $\subseteq$ on elements of $\widehat{X}$ is decidable [17, Theorem 5.3].

Having defined the completion $\widehat{X}$ of a wpo X, we can define the completion $\mathfrak{S} = \widehat{\mathfrak{X}}$ of a (functional) WSTS $\mathfrak{X} = (X, \xrightarrow{F}, \leq)$ as $(\widehat{X}, \xrightarrow{\mathcal{S}F}, \subseteq)$, where $\mathcal{S}F = \{\mathcal{S}f \mid f \in F\}$ [17, Section 6]. For each partial monotonic map $f \in F$, the partial continuous map $\mathcal{S}f : \widehat{S} \to \widehat{S}$ is such that $\operatorname{dom} \mathcal{S}f = \{C \in \widehat{X} \mid C \cap \operatorname{dom} f \neq \emptyset\}$, and $\mathcal{S}f(C) = \downarrow f(C)$ for every $C \in \widehat{X}$. In the cases of Petri nets or functional-lossy channel systems, the completed WSTS is effective [17, Section 6].

The important fact, which assesses the importance of the clover, is the following:

Proposition 2. *Let $\mathfrak{S} = \widehat{\mathfrak{X}}$ be the completion of the functional WSTS $\mathfrak{X} = (X, \xrightarrow{F}, \leq)$. For every state $s_0 \in X$, $Cover_{\mathfrak{X}}(s_0) = Cover_{\mathfrak{S}}(s_0) \cap X = \downarrow Clover_{\mathfrak{S}}(s_0) \cap X$.*

$Cover_{\mathfrak{S}}(s_0)$ is contained, usually strictly, in $\downarrow Clover_{\mathfrak{S}}(s_0)$. The above states that, when restricted to non-limit elements (in X), both contain the same elements. Taking lub-accelerations $(\mathcal{S}f)^\infty$ of any composition f of maps in F leaves $Cover_{\mathfrak{S}}(s_0)$, but is always contained in $\downarrow Clover_{\mathfrak{S}}(s_0) = \mathit{cl}(Cover_{\mathfrak{S}}(s_0))$. So we can safely lub-accelerate in $\mathfrak{S} = \widehat{\mathfrak{X}}$ to compute the clover in $\mathfrak{S}$. While the clover is larger than the cover, taking the intersection back with X will produce exactly the cover $Cover_{\mathfrak{X}}(s_0)$.

4 A Robust Class of WSTS: ω^2-WSTS

The construction of the completion $\mathfrak{S} = \widehat{\mathfrak{X}}$ of a WSTS $\mathfrak{X} = (X, \xrightarrow{F}, \leq)$ is almost perfect: the only missing ingredient to show that $\mathfrak{S}$ is a complete WSTS is to check that $\widehat{X}$ is well-ordered by inclusion. We have indeed seen that $\widehat{X}$ is a continuous dcpo; and $\mathfrak{S}$ is strongly monotonic, because $\mathcal{S}f$ is continuous, hence monotonic, for every $f \in F$.

We show that, in some cases, $\widehat{X}$ is indeed *not* well-ordered. Take X to be Rado's structure X_{Rado} [29], i.e., $\{(m, n) \in \mathbb{N}^2 \mid m < n\}$, ordered by $\leq_{\text{Rado}}$: $(m, n) \leq_{\text{Rado}}$

(m', n') iff $m = m'$ and $n \leq n'$, or $n < m'$. It is well-known that $\leq_{\text{Rado}}$ is a well quasi-ordering, and that $\mathbb{P}(X_{\text{Rado}})$ is not well-quasi-ordered by $\leq^\sharp_{\text{Rado}}$, defined as $A \leq^\sharp_{\text{Rado}} B$ iff for every $y \in B$, there is a $x \in A$ such that $x \leq_{\text{Rado}} y$ [26]; see for example [5, Example 3.2] for a readable reference. One can show that $\widehat{X_{\text{Rado}}} = Idl(X_{\text{Rado}})$ is comprised of all elements of X_{Rado}, plus infinitely many elements $\omega_0, \omega_1, \ldots, \omega_i, \ldots,$ and ω, so that $(i, n) \leq \omega_i$ for all $n \geq i + 1$, $\omega_i \leq \omega$ for all $i \in \mathbb{N}$, and $\{\omega_i \mid i \in \mathbb{N}\}$ is an antichain. We note that the latter is infinite. So:

Lemma 1. *$\widehat{X_{Rado}}$ is not well-ordered by inclusion.*

A well-quasi-order X is ω^2-*wqo* if and only if it does not contain an (isomorphic copy of) X_{Rado}, see e.g. [26]. We show that the above is the only case that can go bad:

Proposition 3. *Let S be a well-quasi-order. Then $\widehat{S}$ is well-quasi-ordered by inclusion iff S is ω^2-wqo.*

Let an ω^2-*WSTS* be any WSTS whose underlying poset is ω^2-wqo. It follows:

Theorem 1. *Let $\mathfrak{S} = (S, \xrightarrow{F}, \leq)$ be a functional WSTS. Then $\widehat{\mathfrak{S}}$ is a (complete, functional) WSTS iff $\mathfrak{S}$ is an ω^2-WSTS.*

All wpos used in the literature, and in fact all wpos arising from the hierarchy of data types of [17, Section 5] are ω^2-wqo. This follows from the fact that they are even better-quasi-ordered—see [5] for a gentle introduction to the latter concept.

Effective complete WSTS. The completion $\widehat{\mathfrak{S}}$ of a WSTS $\mathfrak{S}$ is effective iff the completion $\widehat{S}$ of the set of states is effective and if $\mathcal{S}f$ is recursive for all $f \in F$. $\widehat{S}$ is effective for all the data types of [17, Section 5]. Also, $\mathcal{S}f$ is indeed recursive for all $f \in F$, whether in Petri nets, functional-lossy channel systems (a way of recasting lossy channel systems as functional WSTS [17, Section 6]), reset/transfer Petri nets notably. As promised, we can now show:

Proposition 4. *There are effective complete WSTS $\mathfrak{S}$ such that the map $Clover_{\mathfrak{S}} : S \to \mathbb{P}_{fin}(S)$ is not recursive.*

Proof. Let $\mathfrak{S}$ be the completion of a functional-lossy channel system [17, Section 6] on the message alphabet Σ. By Theorem 1, $\mathfrak{S}$ is a complete WSTS. It is effective, too, see op.cit., or [1, Lemma 6]. $Clover_{\mathfrak{S}}(s_0)$ can be written as a tuple of control states and of *simple regular expression* $P_1 + \ldots + P_n$ representing the contents of channels. Each P_i is a product of atomic expressions A^* ($A \in \mathbb{P}_{\text{fin}}(\Sigma)$) or $a^?$ ($a \in \Sigma$). Now $Post^*_{\mathfrak{S}}(s_0)$ is finite iff none of these atomic expressions is of the form A^*. So computing $Clover_{\mathfrak{S}}(s_0)$ would allow one to decide boundedness for functional-lossy channel systems. However functional-lossy channel systems are equivalent to lossy channel systems in this respect, and boundedness is undecidable for the latter [9]. The same argument also applies to reset Petri nets [11]. □

5 A Conceptual Karp-Miller Procedure

We say that an effective complete (functional) WSTS $\mathfrak{S} = (S, \xrightarrow{F}, \leq)$ is ∞-*effective* iff every function g^∞ is computable, for every $g \in F^*$, where F^* is the set of all compositions of map in F. E.g., the completion of a Petri net is ∞-effective: not only is $\mathbb{N}_\omega^k$ a wpo, but every composition of transitions $g \in F^*$ is of the form $g(\boldsymbol{x}) = \boldsymbol{x} + \delta$, where $\delta \in \mathbb{Z}^k$. If $\boldsymbol{x} < g(\boldsymbol{x})$ then $\delta \in \mathbb{N}^k \setminus \{0\}$. Write $\boldsymbol{x}_i$ the ith component of $\boldsymbol{x}$, it follows that $g^\infty(\boldsymbol{x})$ is the tuple whose ith component is $\boldsymbol{x}_i$ if $\delta_i = 0$, ω otherwise.

Let $\mathfrak{S}$ be an ∞-effective WSTS, and write $A \sqsubseteq B$ iff $\downarrow A \subseteq \downarrow B$, i.e., iff every element of A is below some element of B. The following is a simple procedure which computes the clover of its input $s_0 \in S$ (when it terminates):

```
Procedure Clover_S(s_0) :
1. $A \leftarrow \{s_0\}$;
2. while $Post_{\mathfrak{S}}(A) \not\sqsubseteq A$ do
   (a) Choose fairly $(g, a) \in F^* \times A$
       such that $a \in \operatorname{dom} g$;
   (b) $A \leftarrow A \cup \{g^\infty(a)\}$;
3. return $\operatorname{Max} A$;
```

Fig. 1. The **Clover**$_{\mathfrak{S}}$ procedure

Note that **Clover**$_{\mathfrak{S}}$ is well-defined and all its lines are computable by assumption, provided we make clear what we mean by fair choice in line (a). Call A_m the value of A at the start of the $(m-1)$st turn of the loop at step 2 (so in particular $A_0 = \{s_0\}$). The choice at line (a) is *fair* iff, on every infinite execution, every pair $(g, a) \in F^* \times A_m$ will be picked at some later stage $n \geq m$.

Our procedure is more conceptual than the existing proposals, which generally build a tree [27,15,16,22] or a graph [12] for computing the clover. We shall see that termination of **Clover**$_{\mathfrak{S}}$ has strong ties with the theory of *flattening* [8]; but this paper requires one to enumerate sets of the form $g^*(\boldsymbol{x})$, which is sometimes harder than computing just the element $g^\infty(\boldsymbol{x})$. For example, if $g : \mathbb{N}^k \to \mathbb{N}^k$ is an affine map $g(\boldsymbol{x}) = A\boldsymbol{x} + \boldsymbol{b}$ with $A \geq 0$ and $\boldsymbol{b} \geq 0$ then $g^\infty(\boldsymbol{x})$ is computable as a vector in $\mathbb{N}_\omega^k$ [18, Theorem 7.9], but $g^*(\boldsymbol{x})$ is not even definable by a Presburger formula.

Finally, we use a *fixpoint test* (line 2) that is not in the Karp-Miller algorithm; and this improvement allows **Clover**$_{\mathfrak{S}}$ to terminate in *more cases* than the Karp-Miller procedure when it is used for extended Petri nets (for reset Petri nets for instance, which are a special case of the affine maps above), as we shall see. To decide whether the current set A, which is always an under-approximation of $Clover_{\mathfrak{S}}(s_0)$, is the clover, it is enough to decide whether $Post_{\mathfrak{S}}(A) \sqsubseteq A$. The various Karp-Miller procedures only test each branch of a tree separately, to the partial exception of the minimal coverability tree algorithm [15] and the recent coverability algorithm [22], which compare nodes across branches. That the simple test $Post_{\mathfrak{S}}(A) \sqsubseteq A$ does all this at once does not seem to have been observed until now.

By Proposition 4, we cannot hope to have **Clover**$_{\mathfrak{S}}$ terminate on all inputs. But:

Theorem 2 (Correctness). *If* **Clover**$_{\mathfrak{S}}(s_0)$ *terminates, then it computes* $Clover_{\mathfrak{S}}(s_0)$.

If the generalized Karp-Miller Tree procedure [15] terminates then it has found a finite set $g_1, g_2, \ldots, g_n$ of maps to lub-accelerate. These lub-accelerations will also be found by **Clover**$_{\mathfrak{S}}$, by fairness. From the fixpoint test, **Clover**$_{\mathfrak{S}}$ will also stop. The reset Petri net of [11, Example 3], with an extra transition that adds a token to each place, is an example where the generalized Karp-Miller procedure does not terminate, while **Clover**$_{\mathfrak{S}}$ terminates. So:

Proposition 5. *The procedure* **Clover**$_{\mathfrak{S}}$ *terminates in more cases than the generalized Karp-Miller procedure.*

Termination is however undecidable, using Proposition 4 and Theorem 2.

Proposition 6. *There is an ∞-effective complete WSTS such that the termination of* **Clover**$_{\mathfrak{S}}$ *is undecidable.*

We now characterize those transition systems on which **Clover**$_{\mathfrak{S}}$ terminates.

A functional transition system $(\mathfrak{S}, \xrightarrow{F})$ with initial state s_0 is *flat* iff there are finitely many words $w_1, w_2, ..., w_k \in F^*$ such that any fireable sequence of transitions from s_0 is contained in the language $w_1^* w_2^* ... w_k^*$. (We equate functions in F with letters from the alphabet F, and understand words as the corresponding composition of maps.) Ginsburg and Spanier [24] call this a *bounded* language, and show that it is decidable whether any context-free language is flat.

Not all systems of interest are flat. For an arbitrary system S, *flattening* [8] consists in finding a flat system S', equivalent to S w.r.t. reachability, and computing on S' instead of S. We adapt the definition in [8] to functional transition systems (without an explicit finite control graph). A functional transition system $\mathfrak{S}_1 = (S_1, \xrightarrow{F_1})$, together with a map $\varphi : S_1 \to S_2$ and a map, also written φ, from F_1 to F_2, is a *flattening* of a functional transition system $\mathfrak{S}_2 = (S_2, \xrightarrow{F_2})$ iff (1) $\mathfrak{S}_1$ is flat and (2) for all $(s, s') \in S_1^2$, for all $f_1 \in F_1$ such that $s \in \mathrm{dom}\, f_1$ and $s' = f_1(s)$, $\varphi(s) \in \mathrm{dom}\, \varphi(f_1)$ and $\varphi(s') = \varphi(f_1)(\varphi(s))$. (I.e., φ is a morphism of transition systems.) Let us recall that $(\mathfrak{S}, s_0)$ is *Post*-flattable* iff there is a flattening $\mathfrak{S}_1$ of $\mathfrak{S}$ and a state s_1 of $\mathfrak{S}_1$ such that $\varphi(s_1) = s_0$ and $Post^*_{\mathfrak{S}}(s_0) = \varphi(Post^*_{\mathfrak{S}_1}(s_1))$. A flattening is *continuous* iff $\mathfrak{S}_1$ is an complete transition system and $\varphi : S_1 \to S_2$ is continuous. Correspondingly, we say that $(\mathfrak{S}, s_0)$ is *clover-flattable* iff there is an continuous flattening $\mathfrak{S}_1$, φ of $\mathfrak{S}$ and a state s_1 of $\mathfrak{S}_1$ such that $\varphi(s_1) = s_0$ and $Clover_{\mathfrak{S}}(s_0) \sqsubseteq \varphi(Clover_{\mathfrak{S}_1}(s_1))$.

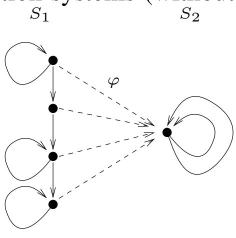

Fig. 2. Flattening

We obtain the following; the proof is non-trivial, and omitted for lack of space.

Theorem 3. *Let $\mathfrak{S}$ be an ∞-effective complete WSTS. The procedure* **Clover**$_{\mathfrak{S}}$ *terminates on s_0 iff $(\mathfrak{S}, s_0)$ is clover-flattable. Then we can even require that the continuous flattening has the same clover up to φ, i.e., $Clover_{\mathfrak{S}}(s_0) = \mathrm{Max}\, \varphi(Clover_{\mathfrak{S}_1}(s_1))$.*

6 Application: Well Structured Counter Systems

We now demonstrate how the fairly large class of counter systems fits with our theory. We show that counter systems composed of affine monotone functions with upward closed definition domains are complete (strongly monotonic) WSTS. This result is obtained by showing that every monotone affine function is continuous and its lub-acceleration f^∞ is computable. Moreover, we prove that it is possible to decide whether a general counter system (given by a finite set of Presburger relations) is a monotone affine counter system, but that one cannot decide whether it is a WSTS.

Definition 4. *A* relational counter system *(with n counters), for short an R-*counter system, *$\mathcal{C}$ is a tuple $\mathcal{C} = (Q, R, \to)$ where Q is a finite set of control states, $R = \{r_1, r_2, ...r_k\}$ is a finite set of Presburger relations $r_i \subseteq \mathbb{N}^n \times \mathbb{N}^n$ and $\to \subseteq Q \times R \times Q$.*

We will consider a special case of Presburger relations, those which allow to code the graph of affine functions. A (partial) function $f : \mathbb{N}^n \longrightarrow \mathbb{N}^n$ is *non-negative affine*, for short *affine* if there exist a matrix $A \in \mathbb{N}^{n \times n}$ with *non-negative coefficients* and a vector $b \in \mathbb{Z}^n$ such that for all $\boldsymbol{x} \in \operatorname{dom} f, f(\boldsymbol{x}) = A\boldsymbol{x} + \boldsymbol{b}$. When necessary, we will extend affine maps $f : \mathbb{N}^n \longrightarrow \mathbb{N}^n$ by continuity to $f : \mathbb{N}^n_\omega \longrightarrow \mathbb{N}^n_\omega$, by $f(lub_{i \in \mathbb{N}}(\boldsymbol{x}_i)) = lub_{i \in \mathbb{N}}(f(\boldsymbol{x}_i))$ for every countable chain $(\boldsymbol{x}_i)_{i \in \mathbb{N}}$ in $\mathbb{N}^n$.

Definition 5. *An* affine counter system *(with n counters) (ACS) $\mathcal{C} = (Q, R, \rightarrow)$ is a R-counter system where all relations r_i are (partial) affine functions.*

The domain of maps f in an affine counter system ACS are Presburger-definable. A reset/transfer Petri net is an ACS where every line or column of every matrix contains at most one non-zero coefficient equal to 1, and, all domains are upward closed sets. A *Petri net* is an ACS where all affine maps are translations with upward closed domains.

Theorem 4. *One can decide whether an effective relational counter system is an ACS.*

Proof. The formula expressing that a relation is a function is a Presburger formula, hence one can decide whether R is the graph of a function. One can also decide whether the graph G_f of a function f is monotone because monotonicity of a Presburger-definable function can be expressed as a Presburger formula. Finally, one can also decide whether a Presburger formula represents an affine function $f(\boldsymbol{x}) = A\boldsymbol{x} + \boldsymbol{b}$ with $A \in \mathbb{N}^{n \times n}$ and $\boldsymbol{b} \in \mathbb{Z}^n$ from [10]. □

For counter systems (which include Minsky machines), monotonicity is undecidable. Clearly, a counter system $\mathfrak{S}$ is well-structured iff $\mathfrak{S}$ is monotone: so there is no algorithm to decide whether a relational counter system is a WSTS. However, an ACS is strongly monotonic iff each map f is partial monotonic; this is equivalent to requiring that $\operatorname{dom} f$ is upward closed, since all matrices A have non-negative coefficients. This is easily cast as Presburger formula, and therefore decidable.

Proposition 7. *There is an algorithm to decide whether an ACS is a strongly monotonic WSTS.*

We have recalled that Petri net functions ($f(x) = x + b$, $b \in \mathbb{Z}^n$ and $\operatorname{dom}(f)$ upward closed) can be lub-accelerated effectively. This result was generalized to broadcast protocols (equivalent to transfer Petri nets) by Emerson and Namjoshi [12] and to a class of affine functions $f(\boldsymbol{x}) = A\boldsymbol{x} + \boldsymbol{b}$ such that $A \in \mathbb{N}^{n \times n}$, $b \in \mathbb{N}^n$ and $\operatorname{dom}(f)$ is upward closed [18]. Antonik recently extended this result to Presburger monotone affine functions: for every $f(\boldsymbol{x}) = A\boldsymbol{x} + \boldsymbol{b}$ with $A \in \mathbb{N}^{n \times n}$, $b \in \mathbb{Z}^n$ and $\operatorname{dom}(f)$ Presburger-definable, the function f^∞ is recursive [7]. We deduce the following strong relationship between well-structured ACS and complete well-structured ACS.

Theorem 5. *The completion of an ACS S is an ∞-effective complete WSTS iff S is a strongly monotonic WSTS.*

Proof. Strong monotonicity reduces to partial monotonicity of each map f, as discussed above. Well-structured ACS are clearly effective, since $Post(\boldsymbol{s}) = \{\boldsymbol{t} \mid \exists f \in F \cdot f(\boldsymbol{t}) = \boldsymbol{s}\}$ is Presburger-definable. Note also that monotone affine function are continuous, and $\mathbb{N}^n_\omega$ is cdcwo. Finally, for every Presburger monotone affine function f, the function f^∞ is recursive, so the considered ACS is ∞-effective. □

Corollary 1. *One may decide whether the completion of an ACS is an ∞-effective complete WSTS.*

So the completions of reset/transfer Petri nets [11], broadcast protocols [13], self-modifying Petri nets [30] and affine well-structured nets [18] are ∞-effective complete WSTS.

7 Conclusion and Perspectives

We have provided a framework of *complete WSTS*, and of *completions* of WSTS, on which forward reachability analyses can be conducted, using natural finite representations for downward closed sets. The central element of this theory is the *clover*, i.e., the set of maximal elements of the closure of the cover. We have shown that, for complete WSTS, the clover is finite and describes the closure of the cover exactly. When the original WSTS is not complete, we have shown the the general completion of WSTS defined in [17] is still a WSTS, iff the original WSTS is an ω^2-*WSTS*. This charaterize a new, robust class of WSTS. We have also defined a simple procedure for computing the clover for ∞-effective complete WSTS, and we have shown that it terminates iff the WSTS is *clover-flattable*, iff it contains a flat subsystem having the same clover. We have also observed procedure terminates in more cases than the Karp-Miller procedure when applied to extensions of Petri nets.

In the future, we shall explore efficient strategies for choosing sequences $g \in F^*$ to lub-accelerate in the $\mathbf{Clover}_{\mathfrak{S}}$ procedure. We will also analyze whether $\mathbf{Clover}_{\mathfrak{S}}$ terminates in models such as BVASS [31], transfer Data nets [28], reconfigurable nets, timed Petri nets [4], post-self-modifying Petri nets [30] and strongly monotone affine well-structured nets [18]), i.e., whether they are clover-flattable.

References

1. Abdulla, P., Bouajjani, A., Jonsson, B.: On-the-fly analysis of systems with unbounded, lossy Fifo channels. In: Y. Vardi, M. (ed.) CAV 1998. LNCS, vol. 1427, pp. 305–318. Springer, Heidelberg (1998)
2. Abdulla, P.A., Čerāns, K., Jonsson, B., Tsay, Y.-K.: Algorithmic analysis of programs with well quasi-ordered domains. Information and Computation 160(1–2), 109–127 (2000)
3. Abdulla, P.A., Collomb-Annichini, A., Bouajjani, A., Jonsson, B.: Using forward reachability analysis for verification of lossy channel systems. Formal Methods in System Design 25(1), 39–65 (2004)
4. Abdulla, P.A., Deneux, J., Mahata, P., Nylén, A.: Forward reachability analysis of timed petri nets. In: Lakhnech, Y., Yovine, S. (eds.) FORMATS/FTRTFT 2004. LNCS, vol. 3253, pp. 343–362. Springer, Heidelberg (2004)
5. Abdulla, P.A., Nylén, A.: Better is better than well: On efficient verification of infinite-state systems. In: 14th LICS, pp. 132–140 (2000)
6. Abramsky, S., Jung, A.: Domain theory. In: Abramsky, S., Gabbay, D.M., Maibaum, T.S.E. (eds.) Handbook of Logic in Computer Science, vol. 3, pp. 1–168. Oxford University Press, Oxford (1994)
7. Antonik, A.: Presburger monotone affine functions can be lub-accelerated. Personal communication (2009)
8. Bardin, S., Finkel, A., Leroux, J., Schnoebelen, P.: Flat acceleration in symbolic model checking. In: Peled, D.A., Tsay, Y.-K. (eds.) ATVA 2005. LNCS, vol. 3707, pp. 474–488. Springer, Heidelberg (2005)

9. Cécé, G., Finkel, A., Purushothaman Iyer, S.: Unreliable channels are easier to verify than perfect channels. Information and Computation 124(1), 20–31 (1996)
10. Demri, S., Finkel, A., Goranko, V., van Drimmelen, G.: Towards a model-checker for counter systems. In: Graf, S., Zhang, W. (eds.) ATVA 2006. LNCS, vol. 4218, pp. 493–507. Springer, Heidelberg (2006)
11. Dufourd, C., Finkel, A., Schnoebelen, P.: Reset nets between decidability and undecidability. In: Larsen, K.G., Skyum, S., Winskel, G. (eds.) ICALP 1998. LNCS, vol. 1443, pp. 103–115. Springer, Heidelberg (1998)
12. Emerson, E.A., Namjoshi, K.S.: On model-checking for non-deterministic infinite-state systems. In: 13th LICS, pp. 70–80 (1998)
13. Esparza, J., Finkel, A., Mayr, R.: On the verification of broadcast protocols. In: 14th LICS, pp. 352–359 (1999)
14. Finkel, A.: A generalization of the procedure of Karp and Miller to well structured transition systems. In: Ottmann, T. (ed.) ICALP 1987. LNCS, vol. 267, pp. 499–508. Springer, Heidelberg (1987)
15. Finkel, A.: Reduction and covering of infinite reachability trees. Information and Computation 89(2), 144–179 (1990)
16. Finkel, A.: The minimal coverability graph for Petri nets. In: Rozenberg, G. (ed.) APN 1993. LNCS, vol. 674, pp. 210–243. Springer, Heidelberg (1993)
17. Finkel, A., Goubault-Larrecq, J.: Forward analysis for WSTS, part I: Completions. In: 26th STACS, Freiburg, Germany. Springer, Heidelberg (to appear, 2009)
18. Finkel, A., McKenzie, P., Picaronny, C.: A well-structured framework for analysing Petri net extensions. Information and Computation 195(1-2), 1–29 (2004)
19. Finkel, A., Schnoebelen, P.: Well-structured transition systems everywhere! Theoretical Computer Science 256(1–2), 63–92 (2001)
20. Ganty, P., Raskin, J.-F., van Begin, L.: A complete abstract interpretation framework for coverability properties of WSTS. In: Emerson, E.A., Namjoshi, K.S. (eds.) VMCAI 2006. LNCS, vol. 3855, pp. 49–64. Springer, Heidelberg (2005)
21. Geeraerts, G., Raskin, J.-F., van Begin, L.: Expand, enlarge and check: New algorithms for the coverability problem of WSTS. J. Comp. and System Sciences 72(1), 180–203 (2006)
22. Geeraerts, G., Raskin, J.-F., van Begin, L.: On the efficient computation of the minimal coverability set for Petri nets. In: Namjoshi, K.S., Yoneda, T., Higashino, T., Okamura, Y. (eds.) ATVA 2007. LNCS, vol. 4762, pp. 98–113. Springer, Heidelberg (2007)
23. Gierz, G., Hofmann, K.H., Keimel, K., Lawson, J.D., Mislove, M., Scott, D.S.: Continuous lattices and domains. In: Encyclopedia of Mathematics and its Applications, vol. 93. Cambridge University Press, Cambridge (2003)
24. Ginsburg, S., Spanier, E.H.: Bounded Algol-like languages. Trans. American Mathematical Society 113(2), 333–368 (1964)
25. Goubault-Larrecq, J.: On Noetherian spaces. In: 22nd LICS, Wrocław, Poland, pp. 453–462. IEEE Computer Society Press, Los Alamitos (2007)
26. Jančar, P.: A note on well quasi-orderings for powersets. Information Processing Letters 72(5–6), 155–160 (1999)
27. Karp, R.M., Miller, R.E.: Parallel program schemata. J. Comp. and System Sciences 3(2), 147–195 (1969)
28. Lazič, R., Newcomb, T., Ouaknine, J., Roscoe, A.W., Worrell, J.: Nets with tokens which carry data. Fundamenta Informaticae 88(3), 251–274 (2008)
29. Rado, R.: Partial well-ordering of sets of vectors. Mathematika 1, 89–95 (1954)
30. Valk, R.: Self-modidying nets, a natural extension of Petri nets. In: Ausiello, G., Böhm, C. (eds.) ICALP 1978, vol. 62, pp. 464–476. Springer, Heidelberg (1978)
31. Verma, K.N., Goubault-Larrecq, J.: Karp-Miller trees for a branching extension of VASS. Discrete Mathematics & Theoretical Computer Science 7(1), 217–230 (2005)