FORWARD ANALYSIS FOR WSTS, PART II: COMPLETE WSTS

ALAIN FINKEL [a] AND JEAN GOUBAULT-LARRECQ [b]

[a] LSV, ENS Cachan, CNRS, France
 e-mail address: finkel@lsv.ens-cachan.fr

[b] LSV, ENS Cachan, CNRS, France and INRIA Saclay, France
 e-mail address: goubault@lsv.ens-cachan.fr

ABSTRACT. We describe a simple, conceptual forward analysis procedure for ∞-complete WSTS $\mathfrak{S}$. This computes the so-called *clover* of a state. When $\mathfrak{S}$ is the completion of a WSTS $\mathfrak{X}$, the clover in $\mathfrak{S}$ is a finite description of the downward closure of the reachability set. We show that such completions are ∞-complete 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 and on lossy channel systems. 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 $(\geq; \to^*; \geq)$ t, i.e., whether $s \geq s_1 \to^* t_1 \geq t$ for some s_1, t_1—is decidable, using a simple algorithm that works backwards [Fin87, Fin90, FS01, AČJT00].

 The starting point of this paper and of its first part [FG09] is our desire to derive similar algorithms working *forwards*, 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 $(\geq; \to^*)$ t. (U-boundedness generalizes the boundedness problem, which is the instance of U-boundedness where U is the entire set of states). No backward algorithm can decide this. In fact, U-boundedness is undecidable in general, e.g., on lossy channel systems [CFP96]. 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 a (necessarily non-terminating) forward procedure is implemented in the TREX tool [ABJ98]. Another argument in favor of forward procedures is the following: for depth-bounded processes, a fragment of the π-calculus, the backward algorithm of [AČJT00] is not applicable when the maximal depth of configurations is not known in advance because, in this case, the predecessor configurations are not effectively computable [WZH10]. But the Expand, Enlarge and Check forward algorithm of [GRvB07], which operates on complete WSTS, solves coverability even though the depth of the process is not known a priori [WZH10].

State of the Art. Karp and Miller [KM69] 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 [Fin87, Fin90] introduced the framework of WSTS 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. In [EN98, Fin90] a variant of this generalization of the Karp-Miller procedure was studied; but no guarantee was given that the cover could be represented finitely. In fact, no effective finite representations of downward-closed sets were given in [Fin90]. Finkel [Fin93] modified the Karp-Miller algorithm to reduce the size of the intermediate computed trees. Geeraerts *et al.* [GRvB07] recently proposed a weaker acceleration, which avoids some possible underapproximations in [Fin93]. Emerson and Namjoshi [EN98] take into account the labeling of WSTS and consequently adapt the generalized Karp-Miller algorithm to model-checking. They assume the existence of a compatible dcpo, and generalize the Karp-Miller procedure to the case of broadcast protocols (which are equivalent to transfer Petri nets). However, termination is then not guaranteed [EFM99], and in fact neither is the existence of a finite representation of the cover. We solved the latter problem in [FG09].

Abdulla, Collomb-Annichini, Bouajjani and Jonsson proposed a forward procedure for lossy channel systems [ACABJ04] using downward-closed regular languages as symbolic representations. Ganty, Geeraerts, Raskin and Van Begin [GRvB06b, GRvB06a] 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 ensure that every downward-closed set has a finite representation. Abdulla *et al.* [ACABJ04] 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 [ADMN04a].

Our Contribution. First, we define a *complete WSTS* as a WSTS $\mathfrak{S}$ whose well-ordering is also a continuous dcpo (a dcpo is a directed complete partial ordering). This allows us to design a conceptual procedure $\mathbf{Clover}_{\mathfrak{S}}$ that looks for a finite representation of the downward closure of the reachability set, i.e., of the cover [Fin90]. 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 [EN98, Fin90]. We establish the main properties of clovers in Section 3 and use them to prove $\mathbf{Clover}_{\mathfrak{S}}$ correct, notably, in Section 5.

Second, we characterize complete WSTS for which $\mathbf{Clover}_{\mathfrak{S}}$ terminates. These are the ones that have a (continuous) flattening with the same clover. This establishes a surprising relationship with the theory of flattening [BFLS05]. The result (Theorem 5.21), together with its corollary on covers, rather than clovers (Theorem 5.26), is the main achievement of this paper.

Third, and building on our theory of completions [FG09], 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 (a wqo is a well quasi-ordering), as we show in Section 4. All naturally occurring WSTS are in fact ω^2-WSTS. We shall also explain why this study is important: despite the fact that $\mathbf{Clover}_{\mathfrak{S}}$ cannot terminate on all inputs, that $\mathfrak{S}$ is an ω^2-WSTS will ensure *progress*, i.e., that every opportunity of accelerating a loop will eventually be taken by $\mathbf{Clover}_{\mathfrak{S}}$.

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 monotonic affine functions are upward-closed.

2. Preliminaries

2.1. **Posets, Dcpos.** We borrow from theories of order, as used in model-checking [AČJT00, FS01], and also from domain theory [AJ94, GHK$^+$03]. 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 $\geq$ for the converse quasi-ordering, $<$ for the associated strict ordering ($\leq \setminus \geq$), and $>$ the converse ($\geq \setminus \leq$) of $<$. There is also an associated equivalence relation $\equiv$, defined as $\leq \cap \geq$.

A set X with a partial ordering $\leq$ is a *poset* $(X, \leq)$, or just X when $\leq$ is clear. If X is merely quasi-ordered by $\leq$, then the quotient $X/\equiv$ is ordered by the relation induced by $\leq$ on equivalence classes. So there is not much difference in dealing with quasi-orderings or partial orderings, and we shall essentially be concerned with the latter.

The *upward closure* $\uparrow E$ of a set E in X 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 *basis* of a downward-closed (resp. upward-closed) set E is a subset A such that $E = \downarrow A$ (resp. $E = \uparrow A$); E has a *finite basis* iff A can be chosen to be 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; equivalently, from any infinite sequence $x_0, x_1, \ldots, x_i, \ldots$, one can extract an infinite ascending chain $x_{i_0} \leq x_{i_1} \leq \ldots \leq x_{i_k} \leq \ldots$, with $i_0 < i_1 < \ldots < i_k < \ldots$. While *wqo* stands for well-quasi-ordered set, 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 $\mathrm{lub}(E)$. 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. $\mathrm{Min}\,E$) for 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 sets 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 $\mathrm{Lub}(E) = \{\mathrm{lub}(D) \mid D \text{ directed subset of } E\}$. 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 $\mathrm{lub}(D) \le y$, there is a $z \in D$ such that $x \le z$. Note that $x \ll y$ implies $x \le y$, and that $x' \le x \ll y \le y'$ implies $x' \ll y'$. Write $\Downarrow E = \{y \in X \mid \exists x \in E \cdot y \ll x\}$, and $\Downarrow x = \Downarrow \{x\}$. X is *continuous* iff, for every $x \in X$, $\Downarrow x$ is a directed subset, and has x as least upper bound.

When $\le$ is a well partial ordering that also turns X into a dcpo, we say that X is a *directed complete well order*, or *dcwo*. We shall be particularly interested in continuous dcwos.

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

A weaker requirement is ω-continuity: f is ω-*continuous* iff $\mathrm{lub}\{f(x_n) \mid n \in \mathbb{N}\} = f(\mathrm{lub}\{x_n \mid n \in \mathbb{N}\})$, for every countable chain $(x_n)_{n \in \mathbb{N}}$. This is all we require when we define accelerations, but general continuity is more natural in proofs. We won't discuss this any further: the two notions coincide when X is countable, which will always be the case of the state spaces X we are interested in, where the states should be representable on a Turing machine, hence at most countably many.

The *closed* sets are the complements of open sets. Every closed set is downward-closed. On a dcpo, the closed subsets are the subsets B that are both downward-closed and *inductive*, i.e., such that $\mathrm{Lub}(B) = B$. An inductive subset of X is none other than a sub-dcpo of X.

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 smallest inductive subset B containing A. In general, $\downarrow A \subseteq \mathrm{Lub}(\downarrow A) \subseteq \mathrm{Ind}(\downarrow A) \subseteq cl(A)$, and all inclusions can be strict. Consider $X = \mathbb{N}_\omega^k$, where $k \in \mathbb{N}$, and $\mathbb{N}_\omega$ denotes $\mathbb{N}$ with a new element ω added, ordered by $(n_1, n_2, \ldots, n_k) \le (n_1', n_2', \ldots, n_k')$ iff $(n_1, n_2, \ldots, n_k) = (n_1', n_2', \ldots, n_k')$, or for some i, $1 \le i \le k$, $n_1 = n_1' = n_2 = n_2' = \ldots = n_{i-1} = n_{i-1}' = \omega$, $n_i \ne \omega$, and either $n_i' = \omega$ or $n_i < n_i' \in \mathbb{N}$. Then take $A = \mathbb{N}^k \subseteq X$: $\downarrow A = A$, but $\mathrm{Lub}(\downarrow A) = \mathbb{N}_\omega \times \mathbb{N}^{k-1}$ is strictly larger; in fact $\mathrm{Lub}(\mathrm{Lub}(\downarrow A)) = \mathbb{N}_\omega^2 \times \mathbb{N}^{k-2}$ is even larger, ..., $\mathrm{Lub}^i(\downarrow A) = \mathrm{Lub}(\mathrm{Lub}^{i-1}(\downarrow A))$ equals $\mathbb{N}_\omega^i \times \mathbb{N}^{k-i}$ for all i, $2 \le i \le k$, and this is a strictly increasing chain of subsets. All of them are contained in $Ind(\downarrow A) = \mathbb{N}_\omega^k$, which coincides with $cl(A)$ here. It may also be the case that $\mathrm{Ind}(\downarrow A)$ is strictly contained in $cl(A)$: consider the set X of all pairs (i, m) with $i \in \{0, 1\}$, $m \in \mathbb{N}$, plus a new element ω, ordered by $(i, m) \le (j, n)$ iff $i = j$ and $m = n$, and $(i, m) \le \omega$ for all $(i, m) \in X$, and let $A = \{(0, m) \mid m \in \mathbb{N}\}$; Then $\mathrm{Ind}(\downarrow A) = A \cup \{\omega\}$, but the latter is not even downward-closed, so is strictly smaller than $cl(A)$; in fact $cl(A)$ is the whole of X.

All this nitpicking is irrelevant when X is a *continuous* dcpo, and A is downward-closed in X. In this case indeed, $\mathrm{Lub}(A) = \mathrm{Ind}(A) = cl(A)$. This is well-known, see e.g., [FG09, Proposition 3.5], and will play an important role in our constructions. As a matter in fact, the fact that $\mathrm{Lub}(A) = cl(A)$, in the particular case of continuous dcpos, is required for lub-accelerations to ever reach the closure of the set of states that are reachable in a transition system.

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

We shall be interested in *effective* transition systems. Intuitively, a transition system $(S, \rightarrow)$ is effective iff one can compute the set of successors $Post_{\mathfrak{S}}(s)$ of any state s. We shall take this to imply that $Post_{\mathfrak{S}}(s)$ is finite, and each of its elements is computable, although one could imagine that $Post_{\mathfrak{S}}(s)$ be described differently, say as a regular expression.

Formally, one needs to find a representation of the states $s \in S$. A *representation map* is any surjective map $r : E \rightarrow S$ from some subset E of $\mathbb{N}$ to S. If $e \in E$ is such that $r(e) = s$, then one says that e is a *code* for the state s.

An *effective transition system* is a 4-tuple $(S, \rightarrow, r, post)$, where $(S, \rightarrow)$ is a transition system, $r : E \rightarrow S$ is a representation map, and $post : E \rightarrow \mathbb{P}_{\text{fin}}(E)$ is a computable map such that, for every code e, $r\langle post(e) \rangle = Post_{\mathfrak{S}}(r(e))$. We write $r\langle A \rangle$ the image $\{r(a) \mid a \in A\}$ of the set A by r, and $\mathbb{P}_{\text{fin}}(E)$ is the set of finite subsets of E. A *computable map* from E to $\mathbb{P}_{\text{fin}}(E)$ is by definition a partial recursive map $post : \mathbb{N} \rightarrow \mathbb{P}_{\text{fin}}(\mathbb{N})$ that is defined on all elements of E, and such that $post(e) \in \mathbb{P}_{\text{fin}}(E)$ for all $e \in E$.

For reasons of readability, we shall make an abuse of language, and say that the pair $(S, \rightarrow)$ is itself an effective transition system in this case, leaving the representation map r and the *post* function implicit.

An *ordered* transition system is a triple $\mathfrak{S} = (S, \rightarrow, \leq)$ where $(S, \rightarrow)$ is a transition system and $\leq$ is a partial ordering on S. We say that $(S, \rightarrow, \leq)$ is *effective* if $(S, \rightarrow)$ is effective and if $\leq$ is decidable.

This is again an abuse of language: formally, an *effective ordered* transition system is a 6-tuple $(S, \rightarrow, \leq, r, post, \preceq)$ where $(S, \rightarrow, \leq)$ is an ordered transition system, $(S, \rightarrow, r, post)$ is an effective transition system, and $\preceq$ is a decidable relation on E such that $e \preceq e'$ iff $r(e) \leq r(e')$. By *decidable on E*, we mean that $\preceq$ is a partial recursive map from $\mathbb{N} \times \mathbb{N}$ to the set of Booleans, which is defined on $E \times E$ at least.

We say that $\mathfrak{S} = (S, \rightarrow, \leq)$ is *monotonic* (resp. *strictly monotonic*) 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 $s_1 \overset{*}{\rightarrow} s_1'$ and $s_1' \geq s'$ (resp. $s_1' > s'$). $\mathfrak{S}$ is *strongly monotonic* 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 monotonic transition systems (not even speaking of being computable). However, the *cover* $Cover_{\mathfrak{S}}(s) = \downarrow Post_{\mathfrak{S}}^*(\downarrow s)$ $(= \downarrow Post_{\mathfrak{S}}^*(s)$ when $\mathfrak{S}$ is monotonic) will be much 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 monotonic and $(S, \leq)$ is wpo. This is our object of study.

For strictly monotonic WSTS, it is also possible to decide the boundedness problem, with the help of the Finite Reachability Tree (FRT) [Fin90]. However, the place-boundedness

problem (i.e., to decide whether a place can contain an unbounded number of tokens) remains undecidable for transfer Petri nets [DFS98], which are strictly monotonic WSTS, 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 that are *functional*, i.e., defined by a finite set of transition functions. This is, as in [FG09], for reasons of simplicity. However, our **Clover**$_\mathfrak{S}$ procedure (Section 5), and already the technique of *accelerating loops* (Definition 3.3) depends on the considered transition system being functional.

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 $f \in F$. If additionally, a partial ordering $\leq$ is given, a map $f : S \to S$ is *partial monotonic* iff dom f is upward-closed and for all $x, y \in$ 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 a well-ordering.

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 can decide whether $s \in$ dom f and in this case, one can also compute $f(s)$.

For example, every Petri net, every reset/transfer Petri net, and in fact every affine counter system (see Definition 6.2) is an effective, functional WSTS.

Lossy channel systems [ACABJ04] are not functional: any channel can lose a letter at any position, and although one may think of encoding this as a functional transition system defined by functions f_i for each i, where f_i would lose the letter at position i, this would require an unbounded number of functions. However, for the purpose of computing covers, lossy channel systems are equivalent [Sch01] ("equivalent" means that the decidability status of the usual properties is the same for both models) to *functional-lossy* channel systems, which are functional [FG09]. In the latter, there are functions $\mathtt{send}_a$ to add a fixed letter a to the back of each queue (i.e., dom$(\mathtt{send}_a) = \Sigma^*$, where Σ is the queue alphabet, and $\mathtt{send}_a(w) = wa$), and functions $\mathtt{recv}_a$ to read a fixed letter a from the front of each queue, where reading is only defined when there is an a in the queue, and means removing all letters up to and including the first a from the queue (i.e., dom$(\mathtt{recv}_a) = \{waw' \mid w, w' \in \Sigma^*\}$, $\mathtt{recv}_a(waw') = w'$ where a does not occur in w).

3. Clovers of Complete WSTS

3.1. **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 $\mathbb{N}_\omega^k$, where $\mathbb{N}_\omega$ is $\mathbb{N}$ plus a new top element ω, with the usual componentwise ordering. We have defined general completions of wpos, serving as state spaces, and have briefly described completions of (functional) WSTS in [FG09]. We temporarily abstract away from this, and consider *complete* WSTS directly.

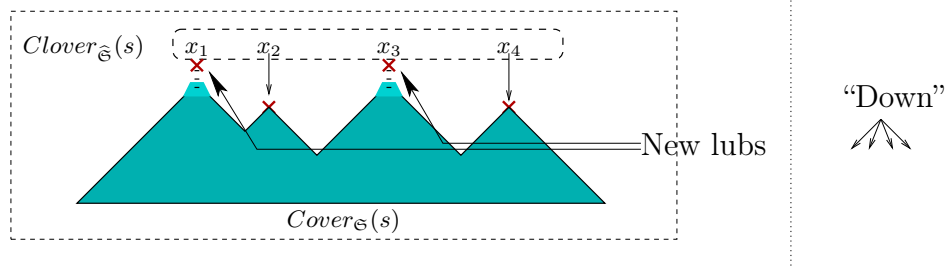Generalizing the notion of continuity to partial maps, we define:

Figure 1: The clover and the cover, in a complete space

Definition 3.1. A *partial continuous* map $f : X \to X$, where $(X, \leq)$ is a dcpo, is a partial map whose domain $\mathrm{dom}\, f$ is open (not just upward-closed), and such that for every directed subset D *in* $\mathrm{dom}\, f$, $\mathrm{lub}(f(D)) = f(\mathrm{lub}(D))$.

This is the special case of a more topological definition: in general, a partial continuous map $f : X \to Y$ is a partial map whose domain is open in X, and such that $f^{-1}(U)$ is open (in X, or equivalently here, in $\mathrm{dom}\, f$) for any open U of Y.

The composition of two partial continuous maps again yields a partial continuous map.

Definition 3.2 (Complete WSTS)**.** A *complete* transition system is a functional transition system $\mathfrak{S} = (S, \xrightarrow{F}, \leq)$ where $(S, \leq)$ is a continuous dcwo and every function in F is partial continuous.

A *complete WSTS* is a functional WSTS that is complete as a functional transition system.

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

Definition 3.3 (Lub-acceleration)**.** 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 $f^\infty(x) = \mathrm{lub}\{f^n(x) \mid n \in \mathbb{N}\}$, 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 monotonic. One cannot decide, in general, whether a recursive function f is monotonic [FMP04] or continuous, whether an ordered set $(S, \leq)$ with a decidable ordering $\leq$, is a dcpo or whether it is a wpo. To show the latter claim for example, fix a finite alphabet Σ, and consider subsets S of Σ^* specified by a Turing machine $\mathcal{M}$ with tape alphabet Σ, so that S is the language accepted by $\mathcal{M}$. Let $\leq$ be, say, the prefix ordering on Σ^*. The property that $(S, \leq)$ is a dcpo, resp. a wpo, is non-trivial and extensional, hence undecidable by Rice's Theorem.

We can also prove that given an effective ordered functional transition system, one cannot decide whether it is a WSTS, or a complete WSTS, in a similar way. However, the completion of *any* functional ω^2-WSTS is complete, as we shall see in Theorem 4.4.

In a complete WSTS, there is a *canonical* finite representation of the cover: the *clover* (a succinct description of the *cl*osure of the c*over*).

Definition 3.4 (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 $\mathrm{Max}\,\mathrm{Lub}(Cover_{\mathfrak{S}}(s_0))$.

This is illustrated in Figure 1. The "down" part on the right is meant to illustrate in which directions one should travel to go down in the chosen ordering. The cover $Cover_{\mathfrak{S}}(s_0)$ is a downward-closed subset, illustrated in blue (grey if you read this in black and white). $\mathrm{Lub}(Cover_{\mathfrak{S}}(s_0))$ has some new least upper bounds of directed subsets, here x_1 and x_3. The clover is given by just the maximal points in $\mathrm{Lub}(Cover_{\mathfrak{S}}(s_0))$, here x_1, x_2, x_3, x_4.

The fact that the clover is indeed a representation of the cover follows from the following.

Lemma 3.5. *Let $(S, \leq)$ be a continuous dcwo. For any closed subset F of S, $\mathrm{Max}\, F$ is finite and $F = {\downarrow}\mathrm{Max}\, F$.*

Proof. As F is closed, it is inductive (i.e., $\mathrm{Lub}(F) = F$). In particular, every element x of F is below some maximal element of F. This is a well-known, and an easy application of Zorn's Lemma. Since F is downward-closed, $F = {\downarrow}\mathrm{Max}\, F$. Now every two elements of $\mathrm{Max}\, F$ are incomparable, i.e., $\mathrm{Max}\, F$ is an antichain: since S is wpo, $\mathrm{Max}\, F$ is finite. $\square$

Remark 3.6. *Lemma 3.5 generalizes to Noetherian spaces, which extend wqos [Gou07]: every closed subset F of a sober Noetherian space S is of the form ${\downarrow}\mathrm{Max}\, F$, with $\mathrm{Max}\, F$ finite [Gou07, Corollary 6.5]. Wpos are sober, and every continuous dcpo is sober in its Scott topology [AJ94, Proposition 7.2.27].*

Proposition 3.7. *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. Now use Lemma 3.5 on the closed set $\mathrm{Lub}(Cover_{\mathfrak{S}}(s_0))$. $\square$

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 $A, B \subseteq S$ such that ${\downarrow}A = {\downarrow}B$, $\mathrm{Max}\, A = \mathrm{Max}\, B$. 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 [Fin93].

Despite the fact that the clover is always finite, it is non-computable in general (see Proposition 4.6 below). Nonetheless, it is computable on *flat* complete WSTS, and even on the larger class of *clover-flattable* complete WSTS (Theorem 5.21 below).

3.2. **Completions.** Many WSTS 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 [FG09], a construction 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* $\mathrm{Idl}(X)$ of X, i.e., the set of ideals of X, ordered by inclusion, where an *ideal* is a downward-closed directed subset of X. The least upper bound of a directed family of ideals $(D_i)_{i \in I}$ is their union. $\widehat{X}$ can also be described as the sobrification $\mathcal{S}(X_a)$ of the Noetherian space X_a, but this is probably harder to understand.

There is an embedding $\eta_X : X \to \widehat{X}$, i.e., an injective map such that $x \leq x'$ in X iff $\eta_X(x) \leq \eta_X(x')$ in $\widehat{X}$. This is defined by $\eta_X(x) = {\downarrow}x$. This allows us to consider X as a subset of $\widehat{X}$, by equating X with its image $\eta_X\langle X \rangle$, i.e., by equating each element $x \in X$ with ${\downarrow}x \in \widehat{X}$. However, we shall only do this in informal discussions, as this tends to make proofs messier.

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 [FG09], itself a slight generalization of [GRvB06b]. In fact, $\widehat{X}$ (minus X) is the *smallest* WADL [FG09, Theorem 3.4].

$\widehat{X} = \mathrm{Idl}(X)$ is always a continuous dcpo. In fact, it is even algebraic [AJ94, Proposition 2.2.22]. It may however fail to be well, hence to be a continuous dcwo, see Proposition 4.2 below.

We have also described a hierarchy of datatypes on which completions are effective [FG09, 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 *word-products* on X. These are the products, as defined in [ABJ98], i.e., regular expressions that are products of *atomic expressions* A^* ($A \in \mathbb{P}_{\mathrm{fin}}(\widehat{X})$, $A \neq \emptyset$) 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 [FG09, 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 $\mathcal{S}F = \{\mathcal{S}f \mid f \in F\}$ [FG09, Section 6]. For each partial monotonic map $f \in F$, the partial continuous map $\mathcal{S}f : \widehat{X} \to \widehat{X}$ is such that $\mathrm{dom}\,\mathcal{S}f = \{C \in \widehat{X} \mid C \cap \mathrm{dom}\,f \neq \emptyset\}$, and $\mathcal{S}f(C) = \downarrow f\langle C \rangle$ for every $C \in \widehat{X}$. In the cases of Petri nets or functional-lossy channel systems, the completed WSTS is effective [FG09, Section 6].

The important fact, which assesses the importance of the clover, is Proposition 3.9 below. We first require a useful lemma. Up to the identification of X with its image $\eta_X\langle X \rangle$, this states that for any downward-closed subset F of $\widehat{X}$, $cl(F) \cap X = F \cap X$, i.e., taking the closure of F only adds new limits, no proper elements of X.

Lemma 3.8. *Let X be a wpo. For any downward-closed subset F of $\widehat{X}$, $\eta_X^{-1}(cl(F)) = \eta_X^{-1}(F)$.*

Proof. We show that $\eta_X^{-1}(cl(F)) \subseteq \eta_X^{-1}(F)$; the converse inclusion is obvious. Since $\widehat{X} = \mathrm{Idl}(X)$ is a continuous dcpo, $cl(F) = \mathrm{Lub}(F)$. Take any $x \in \eta_X^{-1}(cl(F))$: then $\eta_X(x) = \downarrow x$ is the least upper bound of a directed family of ideals D_i in F, $i \in I$: $\downarrow x = \bigcup_{i \in I} D_i$. So x is in D_i for some $i \in I$, hence $\eta_X(x) = \downarrow x \subseteq D_i$, i.e., $\eta_X(x)$ is below D_i in $\widehat{X}$. Since F is downward-closed and $D_i \in F$, $\eta_X(x)$ is also in F, i.e., $x \in \eta_X^{-1}(F)$. $\qquad\square$

Up to the identification of X with $\eta_X\langle X \rangle$, the next proposition states that $Cover_{\mathfrak{X}}(s_0) = Cover_{\mathfrak{S}}(s_0) \cap X = \downarrow Clover_{\mathfrak{S}}(s_0) \cap X$. In other words, to compute the cover of s_0 in the WSTS $\mathfrak{X}$ on the state space X, one can equivalently compute the cover s_0 in the completed WSTS $\widehat{\mathfrak{X}}$, and keep only those non-limit elements (first equality of Proposition 3.9). Or one can equivalently compute the *closure* of the cover in the completed WSTS $\widehat{\mathfrak{X}}$, in the form of the downward closure $\downarrow Clover_{\mathfrak{S}}(s_0)$ of its clover. The closure of the cover will include
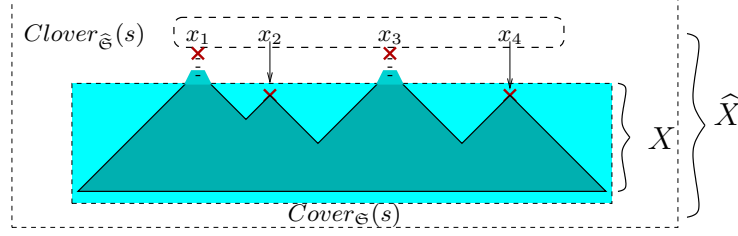
Figure 2: The clover and the cover, in a completed space

extra limit elements, compared to the cover, but no non-limit element by Lemma 3.8. This is illustrated in Figure 2.

Proposition 3.9. *Let $\mathfrak{S} = \widehat{\mathfrak{X}}$ be the completion of the functional WSTS $\mathfrak{X} = (X, \overset{F}{\to}, \leq)$. For every state $s_0 \in X$, $Cover_{\mathfrak{X}}(s_0) = \eta_X^{-1}(Cover_{\mathfrak{S}}(\eta_X(s_0))) = \eta_X^{-1}(\downarrow Clover_{\mathfrak{S}}(\eta_X(s_0)))$.*

Proof. The first equality actually follows from Proposition 6.1 of [FG09]. To be self-contained, we give a direct proof: this will be a consequence of (1) and (2) below. The second equality is a consequence of Proposition 3.7 and Lemma 3.8.

First, we show that: (1) $\eta_X^{-1}(Cover_{\mathfrak{S}}(\eta_X(s_0))) \subseteq Cover_{\mathfrak{X}}(s_0)$. Let x be any element of $\eta_X^{-1}(Cover_{\mathfrak{S}}(\eta_X(s_0)))$, i.e., $\downarrow x$ is in $Cover_{\mathfrak{S}}(\eta_X(s_0))$. By definition, there is a natural number k, and $k+1$ elements $C_0 = \eta_X(s_0)$, C_1, ..., C_k in $\widehat{X}$, and k partial monotonic maps f_1, ..., f_k in F such that $\downarrow x \subseteq C_k$, and $C_i = \mathcal{S}f_i(C_{i-1})$ for every i, $1 \leq i \leq k$.

Since $\downarrow x \subseteq C_k = \mathcal{S}f_k(C_{k-1}) = \downarrow f_k\langle C_{k-1}\rangle$, there is an element $x_{k-1} \in C_{k-1} \cap \operatorname{dom} f_k$ such that $x \leq f_k(x_{k-1})$. Similarly, there is an $x_{k-2} \in C_{k-2} \cap \operatorname{dom} f_{k-1}$ such that $x_{k-1} \leq f_{k-1}(x_{k-2})$, ..., an $x_1 \in C_1 \cap \operatorname{dom} f_2$ such that $x_2 \leq f_2(x_1)$, and an $x_0 \in C_0 \cap \operatorname{dom} f_1$ such that $x_1 \leq f_1(x_0)$. Since $C_0 = \eta_X(s_0) = \downarrow s_0$, we have $x_0 \leq s_0$. Using the fact that f_1, ..., f_k are partial monotonic, $x \leq f_k(f_{k-1}(\ldots(f_2(f_1(s_0)))))$, so $x \in Cover_{\mathfrak{X}}(s_0)$.

Conversely, we show: (2) $Cover_{\mathfrak{X}}(s_0) \subseteq \eta_X^{-1}(Cover_{\mathfrak{S}}(\eta_X(s_0)))$. Let $x \in Cover_{\mathfrak{X}}(s_0)$. So there is a natural number $k \in \mathbb{N}$, and there are k maps f_1, ..., f_k in F such that $x \leq f_k(f_{k-1}(\ldots(f_2(f_1(s_0)))))$; the latter notation in particular implies that $f_i(\ldots(f_2(f_1(s_0))))$ is defined for all i, $0 \leq i \leq k$. For every i, $0 \leq i \leq k$, define C_i as $\downarrow f_i(f_{i-1}(\ldots(f_2(f_1(s_0)))))$. We claim that whenever $i \geq 1$, $C_i = \mathcal{S}f_i(C_{i-1})$. Indeed, $\mathcal{S}f_i(C_{i-1}) = \downarrow f_i\langle C_{i-1}\rangle = \downarrow f_i\langle\downarrow f_{i-1}(\ldots(f_2(f_1(s_0))))\rangle$. Since f_i is partial monotonic, $\downarrow f_i(\downarrow y) = \downarrow f_i(y)$ for every y. So $\mathcal{S}f_i(C_{i-1}) = C_i$. Next, $C_0 = \downarrow s_0$; and $\downarrow x \subseteq C_k$, since $x \in C_k$ and C_k is downward-closed. So $\downarrow x$ is in $Cover_{\mathcal{S}}(\downarrow s_0)$, i.e., $\eta_X(x)$ is in $Cover_{\mathcal{S}}(\eta_X(s_0))$. ☐

$Cover_{\mathfrak{S}}(s_0)$ 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 may leave $Cover_{\mathfrak{S}}(s_0)$, but is always contained in $\downarrow Clover_{\mathfrak{S}}(s_0) = cl(Cover_{\mathfrak{S}}(s_0))$. 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 $Cover_{\mathfrak{X}}(s_0)$.

In more informal terms, the cover is the set of states reachable by either following the transitions in F, or going down. The closure of the cover $\downarrow Clover_{\mathfrak{S}}(s_0)$ contains not just states that are reachable in the above sense, but also the limits of chains of such states. One may think of the elements of $\downarrow Clover_{\mathfrak{S}}(s_0)$ as being those states that are "reachable

in infinitely many steps" from s_0. And we hope to find the finitely many elements of $Clover_{\mathfrak{S}}(s_0)$ by doing enough lub-accelerations.

4. A Robust Class of WSTS: ω^2-WSTS

It would seem clear that the construction of the completion $\mathfrak{S} = \widehat{\mathfrak{X}}$ of a WSTS $\mathfrak{X} = (X, \xrightarrow{F}, \leq)$ be, again, a WSTS. We shall show that this is not the case. 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 $f \in F$.

Next, we shall concern ourselves with the question: under what condition on $\mathfrak{X}$ is $\mathfrak{S} = \widehat{\mathfrak{X}}$ again a WSTS? Equivalently, when is $\widehat{X}$ well-ordered by inclusion? We shall see that there is a definite answer: when X is ω^2-wqo.

4.1. **Motivation.** The question may seem mostly of academic interest. Instead, we illustrate that it is crucial to establish a *progress* property described below.

Let us imagine a procedure in the style of the Karp-Miller tree construction. We shall provide an abstract version of one, **Clover**$_{\mathfrak{S}}$, in Section 5. However, to make things clearer, we shall use a direct imitation of the Karp-Miller procedure for Petri nets for now, generalized to arbitrary WSTS. This is a slight variant of the *generalized Karp-Miller procedure* of [Fin87, Fin90], and we shall therefore call it as such.

We build a tree, with nodes labeled by elements of the completion $\widehat{X}$, and edges labelled by transitions $f \in F$. During the procedure, nodes can be marked extensible or non-extensible. We start with the tree with only one node labeled s_0, and mark it extensible. At each step of the procedure, we pick an extensible leaf node N, labeled with $s \in \widehat{X}$, say, and add new children to N. For each $f \in F$ such that $s \in \text{dom}\,\mathcal{S}f$, let $s' = \mathcal{S}f(s)$, and add a new child N' to N. The edge from N to N' is labeled f. If s' already labels some ancestor of N', then we label N' with s' and mark it non-extensible. If $s'' \leq s'$ for no label s'' of an ancestor of N', then we label N' with s' and mark it extensible. Finally, if $s'' < s'$ for some label s'' of an ancestor N_0 of N' (what we shall refer to as case (*) below), then the path from N_0 to N' is labeled with a sequence of functions $f_1, \ldots, f_p$ from F, and we label N' with the lub-acceleration $(f_p \circ \ldots \circ f_1)^{\infty}(s'')$. (There is a subtle issue here: if there are several such ancestors N_0, then we possibly have to lub-accelerate several sequences $f_1, \ldots, f_p$ from the label s'' of N_0: in this case, we must create several successor nodes N', one for each value of $(f_p \circ \ldots \circ f_1)^{\infty}(s'')$.) When $X = \mathbb{N}^k$ and each $f \in F$ is a Petri net transition, this is the Karp-Miller procedure, up to the subtle issue just mentioned, which we shall ignore.

Let us recall that the Karp-Miller tree (and also the reachability tree) is *finitely branching*, since the set F of functions is finite. This will allow us to use König's Lemma, which states that any finitely branching, infinite tree has at least one infinite branch.

The reasons why the original Karp-Miller procedure terminates on (ordinary) Petri nets are two-fold. First, when $\widehat{X} = \mathbb{N}_\omega^k$, one cannot lub-accelerate more than k times, because each lub-acceleration introduces a new ω component to the label of the produced state, which will not disappear in later node extensions. This is specific to Petri nets, and already fails for reset Petri nets, where ω components do disappear.

The second reason is of more general applicability: $\widehat{X} = \mathbb{N}_\omega^k$ is wpo, and this implies that along every infinite branch of the tree thus constructed, case (*) will eventually happen,
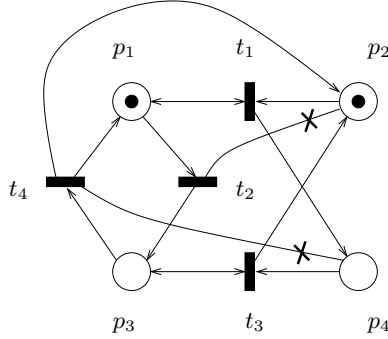
Figure 3: The reset Petri net from [DFS98]

and in fact will happen infinitely many times. Call this *progress*: along any infinite path, one will lub-accelerate infinitely often. In the original Karp-Miller procedure for Petri nets, this will entail termination.

As we have already announced, for WSTS other than Petri nets, termination cannot be ensured. But at least we would like to ensure progress. The argument above shows that progress is obtained provided $\widehat{X}$ is wpo (or even just wqo). *This* is our main motivation in characterizing those wpos X such that $\widehat{X}$ is wpo again.

Before we proceed, let us explain why termination cannot be ensured. Generally, this will follow from undecidability arguments (e.g., Proposition 4.6 below). Here is a concrete case of non-termination. Consider the reset Petri net of [DFS98, Example 3], see Figure 3. This net has 4 places and 4 transitions, hence defines an transition system on $\mathbb{N}^4$. Its transitions are: $t_1(n_1, n_2, n_3, n_4) = (n_1, n_2 - 1, n_3, n_4 + 1)$ if $n_1, n_2 \geq 1$, $t_2(n_1, n_2, n_3, n_4) = (n_1 - 1, 0, n_3 + 1, n_4)$ if $n_1 \geq 1$, $t_3(n_1, n_2, n_3, n_4) = (n_1, n_2 + 1, n_3, n_4 - 1)$ if $n_3, n_4 \geq 1$, and $t_4(n_1, n_2, n_3, n_4) = (n_1 + 1, n_2 + 1, n_3 - 1, 0)$ if $n_3 \geq 1$. Note that $t_4(t_3^{n_2}(t_2(t_1^{n_2}(1, n_2, 0, 0)))) = (1, n_2 + 1, 0, 0)$ whenever $n_2 \geq 1$. The generalized Karp-Miller tree procedure, starting from $s_0 = (1, 1, 0, 0)$, will produce a child labeled $(1, 0, 0, 1)$ through t_1, then $(0, 0, 1, 1)$ through t_2, then $(0, 1, 1, 0)$ through t_3. Using t_4 leads us to case (*) with $s' = (1, 2, 0, 0)$. So the procedure will lub-accelerate the sequence $t_1 t_2 t_3 t_4$, starting from $s_0 = (1, 1, 0, 0)$. However $(t_4 \circ t_3 \circ t_2 \circ t_1)(s') = (1, 1, 0, 0) = s'$ again, so the sequence of iterates $(t_4 \circ t_3 \circ t_2 \circ t_1)^n(s_0)$ stabilizes at s', and $(t_4 \circ t_3 \circ t_2 \circ t_1)^\infty(s_0) = s'$. So the procedure adds a node labeled $s' = (1, 2, 0, 0)$. Similarly, starting from the latter, the procedure will eventually lub-accelerate the sequence $t_1^2 t_2 t_3^2 t_4$, producing a node labeled $(1, 3, 0, 0)$, and in general produce nodes labeled $(1, i + 1, 0, 0)$ for any $i \geq 1$ after having lub-accelerated the sequence $t_1^i t_2 t_3^i t_4$ from a node labeled $(1, i, 0, 0)$. In particular, the generalized Karp-Miller tree procedure will generate infinitely many nodes, and therefore fail to terminate.

This example also illustrates the following: progress does *not* mean that we shall eventually compute limits $g^\infty(s)$ that could not be reached in finitely many steps. In the example above, we do lub-accelerate infinitely often, and compute $(t_4 \circ t_3^i \circ t_2 \circ t_1^i)^\infty(1, i, 0, 0)$, but none of these lub-accelerations actually serve any purpose, since $(t_4 \circ t_3^i \circ t_2 \circ t_1^i)^\infty(1, i, 0, 0) = (1, i + 1, 0, 0)$ is already equal to $(t_4 \circ t_3^i \circ t_2 \circ t_1^i)(1, i, 0, 0)$.

Progress will take a slightly different form in the actual procedure $\mathbf{Clover}_\mathfrak{S}$ of Section 5. In fact, the latter will not build a tree, as the tree is in fact only algorithmic support for ensuring a fair choice of a state in $\widehat{X}$, and essentially acts as a distraction. However, progress will be crucial (Proposition 5.4 states that if the set of values computed by the
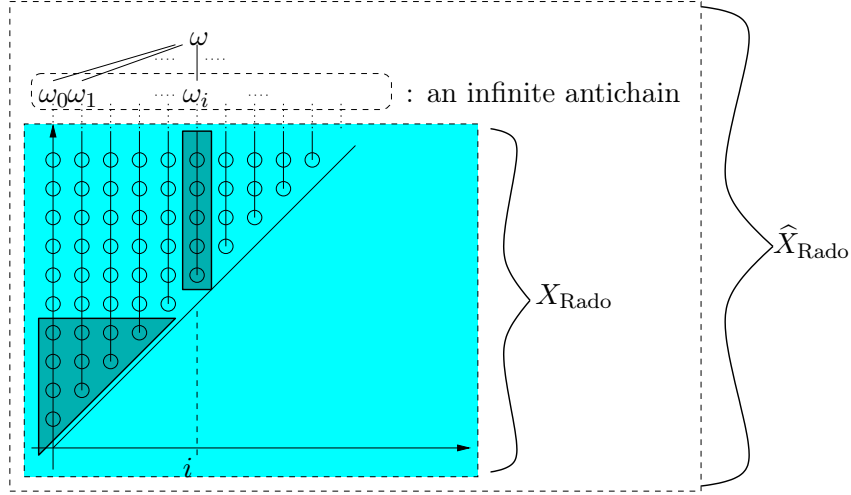
Figure 4: Ideals in Rado's Structure

procedure $\mathbf{Clover}_{\mathfrak{S}}$ is finite then $\mathbf{Clover}_{\mathfrak{S}}$ terminates) in our characterization of the cases where $\mathbf{Clover}_{\mathfrak{S}}$ terminates (Theorem 5.21), as those states that are clover-flattable (see Section 5). Without it, $\mathbf{Clover}_{\mathfrak{S}}$ would terminate in strictly less cases.

4.2. **The Rado Structure.** We now return to the purpose of this section: showing that $\widehat{X}$ is well-ordered iff X is ω^2-wqo. We start by showing that , in some cases, $\widehat{X}$ is indeed *not* well-ordered.

Take X to be Rado's structure X_{Rado} [Rad54], i.e., $\{(m,n) \in \mathbb{N}^2 \mid m < n\}$, ordered by $\leq_{\mathrm{Rado}}$: $(m,n) \leq_{\mathrm{Rado}} (m',n')$ iff $m = m'$ and $n \leq n'$, or $n < m'$. It is well-known that $\leq_{\mathrm{Rado}}$ is a well quasi-ordering, and that $\mathbb{P}(X_{\mathrm{Rado}})$ is not well-quasi-ordered by $\leq^{\sharp}_{\mathrm{Rado}}$, defined as $A \leq^{\sharp}_{\mathrm{Rado}} B$ iff for every $y \in B$, there is a $x \in A$ such that $x \leq_{\mathrm{Rado}} y$ [Jan99]. (Equivalently, $A \leq^{\sharp}_{\mathrm{Rado}} B$ iff $\uparrow B \subseteq \uparrow A$.)

Consider indeed $\omega_i = \{(i,n) \mid n \geq i+1\} \cup \{(m,n) \in X_{\mathrm{Rado}} \mid n \leq i-1\}$, for each $i \in \mathbb{N}$. This is pictured as the dark blue (or dark grey) region in Figure 4, and arises naturally in Lemma 4.1 below. Note that ω_i is downward-closed in $\leq_{\mathrm{Rado}}$. Consider the complement $\overline{\omega}_i$ of ω_i, and note that $\overline{\omega}_i \leq^{\sharp}_{\mathrm{Rado}} \overline{\omega}_j$ iff $\uparrow \overline{\omega}_j \subseteq \uparrow \overline{\omega}_i$, iff $\overline{\omega}_j \subseteq \overline{\omega}_i$ (since $\overline{\omega}_i$ is upward-closed), iff $\omega_i \subseteq \omega_j$. However, when $i < j$, (i,j) is in ω_i but not in ω_j, so $\overline{\omega}_i \not\leq^{\sharp}_{\mathrm{Rado}} \overline{\omega}_j$. So $(\overline{\omega}_i)_{i \in \mathbb{N}}$ is an infinite sequence of $\mathbb{P}(X_{\mathrm{Rado}})$ from which one cannot extract any infinite ascending chain. Hence $\mathbb{P}(X_{\mathrm{Rado}})$ is indeed not wqo.

Let us characterize $\widehat{X_{\mathrm{Rado}}}$. To this end, we exploit the fact that $\widehat{X_{\mathrm{Rado}}} = \mathrm{Idl}(X_{\mathrm{Rado}})$, and examine the structure of directed subsets of X_{Rado}.

Lemma 4.1. *The downward-closed directed subsets of X_{Rado}, apart from those of the form $\downarrow(m,n)$, are of the form $\omega_i = \{(i,n) \mid n \geq i+1\} \cup \{(m,n) \in X_{Rado} \mid n \leq i-1\}$, or $\omega = X_{Rado}$.*

Proof. Take any downward-closed directed subset D of X_{Rado}. Consider the set I of all integers i such that some (i,n) is in D. If I is not bounded, then $D = X_{\mathrm{Rado}}$. Indeed, for

every $(m, n) \in X_{\mathrm{Rado}}$, since I is not bounded, there is an $(i, n') \in D$ with $i > n$. Then $(m, n) < (i, n')$, so $(m, n) \in D$.

If I is bounded, on the other hand, let i be the largest element of I. Then $(i, i + 1)$ is in D: by assumption (i, n) is in D for some $n \geq i + 1$, hence $(i, i + 1)$ also, since D is downward-closed.

There cannot be any $(i', j') \in D$ with $i' < i$ and $j' \geq i$. That is, the rectangular area above the lower triangle of ω_i, as shown in Figure 4, must be entirely outside D. Otherwise, since D is directed, there would be an $(i'', j'') \in D$ with $(i, i + 1), (i', j') \leq_{\mathrm{Rado}} (i'', j'')$; the case $i'' = i$ is impossible, since then $(i', j') \leq_{\mathrm{Rado}} (i'', j'')$ would imply $i' = i''$ and $j' \leq j''$ (impossible since $i' < i$), or $j' < i''$ (impossible since then $i \leq j' < i'' = i$); since $i'' \neq i$ and $(i, i + 1) \leq_{\mathrm{Rado}} (i'', j'')$, $i'' > i + 1$, contradicting the maximality of i in I.

On the other hand, since $(i, i + 1)$ is in D, then the lower triangle of ω_i, as shown in Figure 4, must be in D: these are the points (m, n) with $n < i$.

If the set of natural numbers n such that (i, n) is in D is bounded, say by $n_{\max}$, then the only elements in D are those of the form (i, j) with $j \leq n_{\max}$, and those of the form (m, n) with $n < i$. One checks easily that this is $\downarrow(i, n_{\max})$ in X_{Rado}. Otherwise, D contains every (i, n) with $n \geq i + 1$, and therefore D contains ω_i. It cannot contain more, so $D = \omega_i$. Then one checks that ω_i is indeed directed and downward-closed. $\qquad\square$

So $\widehat{X_{\mathrm{Rado}}} = \mathrm{Idl}(X_{\mathrm{Rado}})$ is obtained by adjoining infinitely many elements ω_0, ω_1, ..., ω_i, ..., and ω to X_{Rado}. They are ordered so that $(i, n) \leq \omega_i$ for all $n \geq i + 1$, $\omega_i \leq \omega$ for all $i \in \mathbb{N}$, and no other ordering relationship exists that involves one of the fresh elements. In particular, note that $\{\omega_i \mid i \in \mathbb{N}\}$ is an infinite antichain, whence $\widehat{X_{\mathrm{Rado}}} = \mathrm{Idl}(X_{\mathrm{Rado}})$ is not wqo:

Proposition 4.2. $\widehat{X_{Rado}}$ *contains an infinite chain, and is therefore not well-ordered by inclusion.* $\qquad\square$

4.3. ω^2-**WSTS.** Recall here the working definition in [Jan99]: a well-quasi-order X is ω^2-*wqo* if and only if it does not contain an (isomorphic copy of) X_{Rado}; here we use Jančar's definition, as it is more tractable than the complex definition of [Mar94]. Jančar proved that X is ω^2-*wqo* iff $(\mathbb{P}(X), \leq^{\sharp})$ is wqo, see e.g. [Jan99]. We show that the above is the only case that can go bad:

Proposition 4.3. *Let S be a well-quasi-order. Then $\widehat{S}$ is well-quasi-ordered by inclusion iff S is ω^2-wqo.*

Proof. Recall that $B_1 \leq^{\sharp}_{\mathrm{Rado}} B_2$ if and only if for every $y_2 \in B_2$, there is $y_1 \in B_1$ with $y_1 \leq_{\mathrm{Rado}} y_2$. Note that $B_1 \leq^{\sharp}_{\mathrm{Rado}} B_2$ if and only if $\uparrow B_1 \supseteq \uparrow B_2$. Reformulate the previous result of Jančar [Jan99] by using the ordering $\leq^{\sharp}_{\mathrm{Rado}}$: S is ω^2-wqo if and only if $\mathbb{P}(S)$ is well-ordered by $\leq^{\sharp}_{\mathrm{Rado}}$.

Recall that the Alexandroff topology on a poset is the collection of its upward-closed subsets; i.e., a subset is Alexandroff-open if and only if it is upward-closed. Write S_a for S with its Alexandroff topology. Any set of the form $\uparrow B$ in S is Alexandroff-open (i.e., upward-closed), and any Alexandroff-open is of this form, with B finite, because S is well. In other words, the set $\mathcal{O}(S_a)$ of all opens (upward-closed subsets) of S is well-ordered by reverse inclusion $\supseteq$ if and only if S is ω^2-wqo.

Recall that the *Hoare powerdomain* $\mathcal{H}(S_a)$ of S_a is the set of all non-empty closed subsets of S_a (the downward-closed subsets of S), ordered by inclusion. It follows that $\mathcal{H}(S_a)$ is well-ordered by inclusion $\supseteq$ if and only if S is ω^2-wqo. Then we recall that $\widehat{S} = \mathcal{S}(S_a)$ is the subspace of $\mathcal{H}(S)$ consisting of all irreducible closed subsets [Gou07].

When S is ω^2-wqo, since $\mathcal{H}(S_a)$ is well-ordered by inclusion, the smaller set $\widehat{S} = \mathcal{S}(S_a)$ is also well-ordered by inclusion.

Conversely, assume that $\widehat{S} = \mathcal{S}(S_a)$ is well-ordered by inclusion. If S was not ω^2-wqo, then it would contain a subset Y that is order-isomorphic to X_{Rado}. Hence $\widehat{S} = \mathcal{S}(S_a) = \mathrm{Idl}(S)$ would contain $\widehat{Y} = \mathrm{Idl}(Y)$. However by Proposition 4.2 $\mathrm{Idl}(Y)$ contains an infinite antichain: contradiction. $\qquad\square$

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

Theorem 4.4. *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.* $\qquad\square$

4.4. **Are ω^2-wqos Ubiquitous?** X_{Rado} is an example of a wqo that is not ω^2-wqo. It is natural to ask whether this is the norm or an exception. We claim that all wpos used in the verification literature are in fact ω^2-wpo.

Consider the following grammar of datatypes, which extends that of [FG09, Section 5] with the case of finite trees (last line):

$$
\begin{array}{lll}
D & ::= & \mathbb{N} \qquad\qquad\qquad\ \text{natural numbers} \\
& | & A_{\leq} \qquad\qquad\qquad \text{finite set } A, \text{ ordered by } \leq \\
& | & D_1 \times \ldots \times D_k \quad \text{finite product} \\
& | & D_1 + \ldots + D_k \quad \text{finite, disjoint sum} \\
& | & D^* \qquad\qquad\qquad\ \text{finite words} \\
& | & D^{\circledast} \qquad\qquad\qquad \text{finite multisets} \\
& | & \mathcal{T}(D) \qquad\qquad\quad\ \text{finite trees}
\end{array}
\tag{4.1}
$$

$\mathbb{N}$ is ordered with its usual ordering; the ordering $\leq$ on the arbitrary finite set A is itself arbitrary. Finite products are ordered componentwise: given that each D_i is ordered by $\leq_i$, then the ordering $\leq$ on $D = D_1 \times \ldots \times D_k$ is defined by $(x_1, \ldots, x_k) \leq (y_1, \ldots, y_k)$ iff $x_1 \leq_1 y_1$ and $\ldots$ and $x_k \leq y_k$. Finite sums are ordered in the obvious way: the elements of $D_1 + \ldots + D_k$ are pairs (i, x) where $1 \leq i \leq k$ and $x \in D_i$, and $(i, x) \leq (j, y)$ iff $i = j$ and $x \leq y$.

D^* is the set of finite words over the (possibly infinite) alphabet D, and given that the ordering on D is $\leq$, D^* is ordered by the *divisibility ordering* $\leq^*$, defined by $w \leq^* w'$ iff, writing w as the sequence of letters $a_1 a_2 \ldots a_n$, then w' is of the form $w_0 a_1' w_1 a_2' \ldots a_n' w_n$, for some words $w_0, w_1, \ldots, w_n$, and some letters a_i', $1 \leq i \leq n$, such that $a_i \leq a_i'$.

$D^{\circledast}$ is the set of finite multisets $\{\!| x_1, \ldots, x_n |\!\}$ of elements of D. Write again $\leq$ the ordering on D. Then $D^{\circledast}$ is ordered by $\leq^{\circledast}$ defined as: $\{\!| x_1, x_2, \ldots, x_m |\!\} \leq^{\circledast} \{\!| y_1, y_2, \ldots, y_n |\!\}$ iff there is an injective map $r : \{1, \ldots, m\} \rightarrow \{1, \ldots, n\}$ such that $x_i \leq y_{r(i)}$ for all i, $1 \leq i \leq m$.

Note that $\leq^{\circledast}$ is not the usual multiset extension $\leq^{mul}$ of $\leq$. However, for one, this is the $\leq^m$ quasi-ordering considered, on finite sets X, by Abdulla *et al.* [ADMN04b, Section 2] for example. Then, it turns out that $m \leq^{\circledast} m'$ entails $m \leq^{mul} m'$. In particular, the fact that $\leq^{\circledast}$ is well, whenever $\leq$ is, *entails* that $\leq^{mul}$ is well: given any sequence of multisets

$(m_i)_{i \in \mathbb{N}}$, one can extract an infinite ascending chain with respect to $\leq^{\circledast}$, hence also with respect to $\leq^{mul}$. Similarly, when $(D^{\circledast}, \leq^{\circledast})$ is an ω^2-wqo, then so is $(D^{\circledast}, \leq^{mul})$, using the fact that X is ω^2-wqo iff both X and $\mathbb{P}(X)$ are wqo (the latter, equipped with $\leq^{\sharp}$).

Finally, $\mathcal{T}(D)$ is the set of all finite (unranked, ordered) trees over function symbols taken from D. This is the smallest set X such that, for every $f \in D$, for every $\vec{t} \in D^*$, the pair $(f, \vec{t})$ is in X. When $\vec{t}$ is the word consisting of the terms $t_1 t_2 \dots t_m$, we usually write $(f, \vec{t})$ as the *term* $f(t_1, t_2, \dots, t_m)$. Given an ordering $\leq$ on D, the *embedding ordering* $\leq^{emb}$ on $\mathcal{T}(D)$ is defined by induction on the sum of the sizes of the terms to compare by: $t = f(t_1, t_2, \dots, t_m) \leq^{emb} g(u_1, u_2, \dots, u_n)$ iff $t \leq^{emb} u_j$ for some j, $1 \leq j \leq n$, or $f \leq g$ and $t_1 t_2 \dots t_m \ (\leq^{emb})^* u_1 u_2 \dots u_n$.

We will prove that every datatype defined in (4.1) is not only ω^2-wqo but a better quasi-ordering (bqo). Better quasi-orderings were invented by Nash-Williams to overcome certain limitations of wqo theory [NW65]. Their definition is complex, and we shall omit it. For short, X is bqo iff $\mathbb{P}^{\omega_1}(X)$ is wqo, where ω_1 is the first uncountable ordinal, $\mathbb{P}^{\alpha}(X)$ is defined for every ordinal α by $\mathbb{P}^0(X) = X$, $\mathbb{P}^{\alpha+1} = \mathbb{P}(\mathbb{P}^{\alpha}(X))$, $\mathbb{P}^{\alpha}(X) = \bigcup_{\beta < \alpha} \mathbb{P}^{\beta}(X)$ for every limit ordinal α, and where powersets are quasi-ordered by $\leq^{\sharp}$. Abdulla and Nylén give a gentle introduction to the theory of bqos [AN00].

Then:

Proposition 4.5. *Every datatype defined in (4.1) is ω^2-wqo, and in fact bqo.*

Proof. Every bqo is ω^2-wqo, as the above characterization shows ($\mathbb{P}^{\alpha}(X)$ is wqo for all $\alpha \leq \omega_1$, hence certainly for $\alpha = 0$ and $\alpha = 1$). Any finite ordered set, any finite union of bqos, any finite product of bqos is bqo [Mil85]. When D is bqo, the set of all ordinal-indexed sequences over D is again bqo under an obvious extension of the divisibility ordering, see [NW65] or [Mil85, 2.22]. Since any subset of a bqo is again bqo, we deduce that D^* is bqo whenever D is (this is also mentioned in [AN00, Theorem 3.1 (3)]). When D is bqo, $D^{\circledast}$ is proved to be a bqo in [AN00, Theorem 3.1 (4)]. Finally, D is bqo implies that $\mathcal{T}(D)$ is bqo by [Lav71, Theorem 2.2]; Laver in fact shows that the class of so-called Q-trees is bqo under tree embedding as soon as Q is, where a Q-tree is a possibly infinitely branching tree with branches of length at most ω whose nodes are labeled with elements of Q. $\qquad \square$

In fact, all naturally occurring wqos are bqos, perhaps to the notable exception of finite graphs quasi-ordered by the graph minor relation, which are wqo [RS04] but not known to be bqo.

4.5. **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 $\mathcal{S}f$ is recursive for all $f \in F$. $\widehat{S}$ is effective for all the data types of [FG09, Section 5][1]. Also, $\mathcal{S}f$ is indeed recursive for all $f \in F$, whether in Petri nets, functional-lossy channel systems, and reset/transfer Petri nets notably.

In the case of ordinary or reset/transfer Petri nets, and in general for all affine counter systems (which we shall investigate from Definition 6.2 on), $\mathcal{S}f$ coincides with the extension $\overline{f}$ defined in [FMP04, Section 2]: whenever dom f is upward-closed and $f : \mathbb{N}^k \to \mathbb{N}^k$ is defined by $f(\vec{s}) = A\vec{s} + \vec{a}$, for some matrix $A \in \mathbb{N}^{k \times k}$ and vector $\vec{a} \in \mathbb{Z}^k$, then dom $\mathcal{S}f = \uparrow_S$ dom f,

---

[1]That is, of Section 4.4 of this paper, see (4.1), to the exception of the finite tree constructor. We have a proof that $\widehat{S}$ is in fact effective for all the data types of (4.1) [FG12], but this is not published yet.

and $\mathcal{S}(f)(\vec{s})$ is again defined as $A\vec{s} + \vec{a}$, this time for all $\vec{s} \in \mathbb{N}_\omega^k$, and using the convention that $0 \times \omega = 0$ when computing the matrix product $A\vec{s}$ [FMP04, Theorem 7.9].

In the case of functional-lossy channel systems, it is easy to see that $\mathrm{dom}\,\mathcal{S}(\mathtt{send}_a) = \widehat{S}$, $\mathcal{S}(\mathtt{send}_a)(P) = Pa^?$ for every word-product P; and that $\mathrm{dom}\,\mathcal{S}(\mathtt{recv}_a) = \uparrow_S a^?$, and:

$$
\begin{aligned}
\mathcal{S}(\mathtt{recv}_a)(a^?P) &= P \\
\mathcal{S}(\mathtt{recv}_a)(b^?P) &= \mathcal{S}(\mathtt{recv}_a)(P) \quad (b \neq a) \\
\mathcal{S}(\mathtt{recv}_a)(A^*P) &= A^*P \quad \text{if } a \in A \\
\mathcal{S}(\mathtt{recv}_a)(A^*P) &= \mathcal{S}(\mathtt{recv}_a)(P) \quad \text{otherwise}
\end{aligned}
$$

These formulae in fact work whenever letters are taken from an alphabet that is wqo; for example, any of the data types D of (4.1). We retrieve the formulae of [ABJ98, Lemma 6], which were proved in the case where the alphabet D is finite, with $=$ as ordering. This also generalizes the algorithms on the so-called word language generators of [ADMN04a], which are elements of $(A^\circledast)^*$ with A finite.

As promised, we can now show:

Proposition 4.6. *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 [FG09, Section 6] on the message alphabet Σ. By Theorem 4.4, $\mathfrak{S}$ is a complete WSTS. It is effective, too, see above or [ABJ98, Lemma 6]. $Clover_{\mathfrak{S}}(s_0)$ can be written as a finite set of tuples, consisting of control states q_i (one for each of the communicating automata) and of word-products P_j (one for each channel). Each P_j is a product of atomic expressions A^* ($A \in \mathbb{P}_{fin}(\Sigma)$, $A \neq \emptyset$) 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, if we could compute $Clover_{\mathfrak{S}}(s_0)$, this would allow us 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 [CFP96]. We could have played the same argument with reset Petri nets [DFS98] instead as well. $\qquad\square$

5. A Conceptual Karp-Miller Procedure

There are some advantages in using a forward procedure to compute (part of) the clover for solving coverability. For depth-bounded processes, a fragment of the π-calculus, the simple algorithm that works backward (computing the set of predecessors of an upward-closed initial set) of [AČJT00] is not applicable when the maximal depth of configurations is not known in advance because, in this case, the predecessor configurations are not effectively computable [WZH10]. It has been also proved that, unlike backward algorithms (which solve coverability without computing the clover), the Expand, Enlarge and Check forward algorithm of [GRvB07], which operates on complete WSTS, solves coverability by computing a *sufficient* part of the clover, even though the depth of the process is not known a priori [WZH10]. Recently, Zufferey, Wies and Henzinger proposed to compute a part of the clover by using a particular widening, called a *set-widening operator* [ZWH12], which loses some information, but always terminates and seems sufficiently precise to compute the clover in various case studies.

The Petri net case also gives complexity-theoretic insights. Solving coverability in Petri nets can be done by using Rackoff's forward procedure [Rac78], or the backward procedure

[BG11]. Both work in EXPSPACE—the complexity of the forward coverability procedure of [GRvB07] is not known. On the other hand, the complexity of computing the clover is not primitive recursive for Petri nets [MM81].

Model-checking safety properties of WSTS can be reduced to coverability, but there are other properties, such as *boundedness* (is $Post_{\mathfrak{S}}^*(s)$ finite?) and *U-boundedness* (is $Post_{\mathfrak{S}}^*(s) \cap U$ finite?) that cannot be reduced to coverability: U-boundedness is decidable for Petri nets and for Vector Addition Systems but undecidable for Reset Vector Addition Systems [DFS98], and for Lossy Channel Systems [May03a], hence for general WSTS.

Recall that being able to compute the clover allows one to decide not only *coverability* since s $(\geq; \to^*; \geq)$ t iff $t \in Cover_{\mathfrak{S}}(s)$ iff $\exists t' \in Clover_{\mathfrak{S}}(s)$ such that $t \leq t'$ but also boundedness, U-boundedness and place-boundedness. To the best of our knowledge, the only known algorithms that decide place-boundedness (and also some formal language properties such as regularity and context-freeness of Petri net languages) *require one to compute the clover*.

Another argument in favor of computing clovers is Emerson and Namjoshi's [EN98] approach to model-checking *liveness* properties of WSTS, which uses a finite (coverability) graph based on the clover. Since WSTS enjoy the finite path property ([EN98], Definition 7), model-checking liveness properties is decidable for complete WSTS for which the clover is computable.

All these reasons motivate us to *try* to compute the clover for classes of complete WSTS, even though it is not computable in general.

The key to designing some form of a Karp-Miller procedure, such as the generalized Karp-Miller tree procedure (Section 4.1) or the **Clover$_{\mathfrak{S}}$** procedure below is being able to *compute* lub-accelerations. Hence:

Definition 5.1 (∞-Effective). 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 maps 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 $g(\vec{x}) = \vec{x} + \delta$, where $\delta \in \mathbb{Z}^k$. If $\vec{x} < g(\vec{x})$ then $\delta \in \mathbb{N}^k \setminus \{0\}$. Write $\vec{x}_i$ the ith component of $\vec{x}$, it follows that $g^\infty(\vec{x})$ is the tuple whose ith component is $\vec{x}_i$ if $\delta_i = 0$, ω otherwise.

Let $\mathfrak{S}$ be an ∞-effective WSTS, and write $A \leq^\flat B$ iff $\downarrow A \subseteq \downarrow B$, i.e., iff every element of A is below some element of B. This is the *Hoare quasi-ordering*, also known as the *domination* quasi-ordering. The following is a simple procedure which computes the clover of its input $s_0 \in S$ (when it terminates):

> **Procedure Clover$_{\mathfrak{S}}(s_0)$:**
> 1. $A \leftarrow \{s_0\}$;
> 2. **while** $Post_{\mathfrak{S}}(A) \not\leq^\flat A$ **do**
>   (a) Choose fairly (see below) $(g, a) \in F^* \times A$ such that $a \in dom\, g$;
>   (b) $A \leftarrow A \cup \{g^\infty(a)\}$;
> 3. **return** Max A;

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 $n \geq m$.

A possible implementation of this fair choice is the generalized Karp-Miller tree construction of Section 4.1: organize the states of A as labeling nodes of a tree that we grow. At step m, A_m is the set of leaves of the tree, and case (*) of the generalized Karp-Miller tree construction ensures that all pairs $(g, a) \in F^* \times A_m$ will eventually be picked for consideration. However, the generalized Karp-Miller tree construction does some useless work, e.g., when two nodes of the tree bear the same label.

Most existing proposals for generalizing the Karp-Miller construction do build such a tree [KM69, Fin90, Fin93, GRvB07], or a graph [EN98]. We claim that this is mere algorithmic support for ensuring fairness, and that the goal of such procedures is to compute a finite representation of the cover. Our **Clover**$_\mathfrak{S}$ procedure computes the clover, which is the minimal such representation, and isolates algorithmic details from the core construction.

We shall also see that termination of **Clover**$_\mathfrak{S}$ has strong ties with the theory of *flattening* [BFLS05]. However, Bardin *et al.* require one to enumerate sets of the form $g^*(\vec{x})$, which is sometimes harder than computing the single element $g^\infty(\vec{x})$. For example, if $g : \mathbb{N}^k \to \mathbb{N}^k$ is an affine map $g(\vec{x}) = A\vec{x} + \vec{b} - \vec{a}$ for some matrix $A \in \mathbb{N}^{k \times k}$ and vectors $\vec{a}, \vec{b} \in \mathbb{N}^k$, then $g^\infty(\vec{x})$ is computable as a vector in $\mathbb{N}_\omega^k$, as we have seen in Section 4.5. But $g^*(\vec{x})$ is not even definable by a Presburger formula in general, in fact even when g is a composition of Petri net transitions; this is because reachability sets of Petri nets are not semi-linear in general [HP79].

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 $Post_\mathfrak{S}(A) \leq^\flat A$. The various Karp-Miller procedures only test each branch of a tree separately, to the partial exception of the minimal coverability tree algorithm [Fin90] and Geeraerts *et al.*'s recent coverability algorithm [GRvB07], which compare nodes across branches. That the simple test $Post_\mathfrak{S}(A) \leq^\flat A$ does all this at once does not seem to have been observed until now.

5.1. **Correctness and Termination of the Clover Procedure.** By Proposition 4.6, we cannot hope to have **Clover**$_\mathfrak{S}$ terminate on all inputs. But we can at least start by showing that it is correct, whenever it terminates. This will be Theorem 5.5 below.

We first show that if **Clover**$_\mathfrak{S}$ terminates then the computed set A is contained in $\mathrm{Lub}(Post_\mathfrak{S}^*(s_0))$. It is crucial that $\mathrm{Lub}(F) = cl(F)$ for any downward-closed set F, which holds because the state space S is a continuous dcpo. We use this through invocations to Proposition 3.7.

Lemma 5.2. *Let* $\mathfrak{S} = (S, \xrightarrow{F}, \leq)$ *be a complete (functional) WSTS. For any subset A of states,* $Post_\mathfrak{S}^*(cl(A)) \subseteq cl(Post_\mathfrak{S}^*(A))$.

Proof. We first observe that $Post_\mathfrak{S}(cl(A)) \subseteq cl(Post_\mathfrak{S}(A))$. Indeed, for any $s \in Post_\mathfrak{S}(cl(A))$, there is an $f \in F$ and some $t \in \mathrm{dom}\, f \cap cl(A)$ such that $f(t) = s$. Let U be the complement of $cl(Post_\mathfrak{S}(A))$: U is open by definition. Since f is partial continuous, $f^{-1}(U)$ is open. If s were in U, then t would be in $f^{-1}(U)$, and in $cl(A)$. It is a general property of

topological spaces that an open (here $f^{-1}(U)$) meets $cl(A)$ iff it meets A. So there is also a state t' in $f^{-1}(U) \cap A$. That is, $t' \in \mathrm{dom}\, f$, $f(t') \in U$ and $t' \in A$. But $t' \in A$ implies $f(t') \in Post_{\mathfrak{S}}(A) \subseteq cl(Post_{\mathfrak{S}}(A))$, contradicting the fact that $f(t') \in U$. So s cannot be in U, i.e., $s \in cl(Post_{\mathfrak{S}}(A))$.

By an easy induction on $k \in \mathbb{N}$, it follows that $Post_{\mathfrak{S}}^k(cl(A)) \subseteq cl(Post_{\mathfrak{S}}^k(A))$, hence that $Post_{\mathfrak{S}}^*(cl(A)) \subseteq cl(Post_{\mathfrak{S}}^*(A))$. $\qquad\square$

Proposition 5.3. *Let $\mathfrak{S}$ be an ∞-effective complete functional transition system and A_n be the value of the set A, computed by the procedure $\mathbf{Clover}_{\mathfrak{S}}$ on input s_0, after n iterations of the while statement at line 2. Then A_n is finite, and $A_n \leq^\flat A_{n+1} \leq^\flat Clover_{\mathfrak{S}}(s_0)$, for every $n \in \mathbb{N}$.*

Proof. It is obvious that A_n is finite. Also, the inclusion $A_n \subseteq\, \downarrow A_{n+1}$ is clear, and entails $A_n \leq^\flat A_{n+1}$.

We show that $A_n \leq^\flat Clover_{\mathfrak{S}}(s_0)$, i.e., that $A_n \subseteq\, \downarrow Clover_{\mathfrak{S}}(s_0)$, by induction on n. By Proposition 3.7, it is equivalent to show that $A_n \subseteq cl(Cover_{\mathfrak{S}}(s_0))$.

If $n = 0$, $A_0 = \{s_0\}$, so $A_0 \subseteq Cover_{\mathfrak{S}}(s_0) \subseteq cl(Cover_{\mathfrak{S}}(s_0))$.

Assume $A_n \subseteq cl(Cover_{\mathfrak{S}}(s_0))$, and let us prove that $A_{n+1} \subseteq cl(Cover_{\mathfrak{S}}(s_0))$. Let (g, a) be the selected pair at line (a). We must show that $g^\infty(a) \in cl(Cover_{\mathfrak{S}}(s_0))$.

If $a \not< g(a)$, then $g^\infty(a) = g(a)$ is in $Post_{\mathfrak{S}}^*(a)$, and since $a \in A_n$ and $A_n \subseteq cl(Cover_{\mathfrak{S}}(s_0))$ by induction hypothesis, $g(a)$ is in $Post_{\mathfrak{S}}^*(cl(Cover_{\mathfrak{S}}(s_0)))$. The latter is contained in $cl(Post_{\mathfrak{S}}^*(Cover_{\mathfrak{S}}(s_0)))$ by Lemma 5.2, i.e., in $cl(Cover_{\mathfrak{S}}(s_0))$ by monotonicity.

If $a < g(a)$, then $g^\infty(a) = \mathrm{lub}\{g^n(a) \mid n \in \mathbb{N}\}$ is a least upper bound of a directed chain of elements in $Post_{\mathfrak{S}}^*(a)$. So $g^\infty(a) \in \mathrm{Lub}(Post_{\mathfrak{S}}^*(a)) \subseteq cl(Post_{\mathfrak{S}}^*(a))$. Since $a \in A_n$ and $A_n \subseteq cl(Cover_{\mathfrak{S}}(s_0))$ by induction hypothesis, $g^\infty(a)$ is in $cl(Post_{\mathfrak{S}}^*(cl(Cover_{\mathfrak{S}}(s_0))))$. The latter is contained in $cl(cl(Post_{\mathfrak{S}}^*(Cover_{\mathfrak{S}}(s_0)))) = cl(Post_{\mathfrak{S}}^*(Cover_{\mathfrak{S}}(s_0)))$ by Lemma 5.2, i.e., in $cl(Cover_{\mathfrak{S}}(s_0))$ by monotonicity. $\qquad\square$

If the procedure $\mathbf{Clover}_{\mathfrak{S}}$ does not stop, it will compute an infinite sequence of sets of states. In other words, $\mathbf{Clover}_{\mathfrak{S}}$ does not deadlock. This is the progress property mentioned in Section 4.1.

Proposition 5.4 (Progress). *Let $\mathfrak{S}$ be an ∞-effective complete functional WSTS and A_n be the value of the set A, computed by the procedure $\mathbf{Clover}_{\mathfrak{S}}$ on input s_0, after n iterations of the while statement at line 2. If $\bigcup_n A_n$ is finite, then the procedure $\mathbf{Clover}_{\mathfrak{S}}$ terminates on input s_0.*

Proof. Assume $\mathbf{Clover}_{\mathfrak{S}}$ does not stop on input s_0, but $A = \bigcup_n A_n$ is finite. Since $A_n \leq^\flat A_{n+1}$, there is an index m such that $A_n = A_m$ for all $n \geq m$; also $A = A_m$. Let $(g, a) \in F^* \times A$ be arbitrary. We shall show that $g(a) \leq^\flat A$, i.e., there is an element $a' \in A$ such that $g(a) \leq a'$. Since $a \in A_m$, by fairness there is an $n \in \mathbb{N}$ with $n \geq m$ such that (g, a) is picked at line (a) after n iterations of the loop. Then $g^\infty(a) \leq^\flat A_{n+1} = A$, so $g(a) \leq g^\infty(a) \leq^\flat A_{n+1} = A$. It follows that $Post_{\mathcal{S}}^*(A) \leq^\flat A$, so $Post_{\mathcal{S}}(A) \leq^\flat A$, hence the procedure must stop after m turns of the loop: contradiction. The converse implication is obvious. $\qquad\square$

While $\mathbf{Clover}_{\mathfrak{S}}$ is non-deterministic, this is *don't care non-determinism*: if one execution does not terminate, then no execution terminates. If $\mathbf{Clover}_{\mathfrak{S}}$ terminates, then it computes the clover, and if it does not terminate, then at each step n, the set A_n is contained in the clover. Let us recall that $A_n \leq^\flat A_{n+1}$. We can now prove:

Theorem 5.5 (Correctness). *If* **Clover**$_\mathfrak{S}(s_0)$ *terminates, then it computes* $Clover_\mathfrak{S}(s_0)$.

Proof. If **Clover**$_\mathfrak{S}$ terminates, then it returns a set $\mathrm{Max}\,A$ such that $Post_\mathfrak{S}(A) \leq^\flat A$, i.e., $Post_\mathfrak{S}(A) \subseteq {\downarrow}A$. By monotonicity, it follows that $Post_\mathfrak{S}({\downarrow}A) \subseteq {\downarrow}A$, hence that ${\downarrow}Post_\mathfrak{S}({\downarrow}A) \subseteq {\downarrow}A$. Note that ${\downarrow}A = {\downarrow}\mathrm{Max}\,A$, since A is finite. It follows that $Cover_\mathfrak{S}(s)$ is contained in ${\downarrow}\mathrm{Max}\,A$ for any $s \in {\downarrow}A$.

However, by Proposition 5.3, $\{s_0\} = A_0 \leq^\flat A_1 \leq^\flat \ldots \leq^\flat A_n \leq^\flat \ldots \leq^\flat A$, so $s_0 \in {\downarrow}A$. So $Cover_\mathfrak{S}(s_0) \subseteq {\downarrow}\mathrm{Max}\,A$.

Since A is finite, $\mathrm{Max}\,A$ is, too, so ${\downarrow}\mathrm{Max}\,A$ is closed. Any closed set containing another set must contain its closure. So ${\downarrow}\mathrm{Max}\,A$ must also contain $cl(Cover_\mathfrak{S}(s_0))$. By Proposition 3.7, ${\downarrow}\mathrm{Max}\,A$ must therefore contain ${\downarrow}Clover_\mathfrak{S}(s_0)$. In other words, $Clover_\mathfrak{S}(s_0) \leq^\flat \mathrm{Max}\,A$. However, using Proposition 5.3 again, $\mathrm{Max}\,A \leq^\flat A \leq^\flat Clover_\mathfrak{S}(s_0)$. So $\mathrm{Max}\,A = Clover_\mathfrak{S}(s_0)$. $\qquad\square$

If the generalized Karp-Miller tree procedure (see Section 4.1) terminates then it has found a finite set $g_1, g_2, ..., g_n$ 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. So **Clover**$_\mathfrak{S}$ terminates on at least all inputs where the generalized Karp-Miller tree procedure terminates. We can say more:

Proposition 5.6. *The procedure* **Clover**$_\mathfrak{S}$ *terminates on strictly more input states* $s_0 \in S$ *than the generalized Karp-Miller tree procedure.*

Proof. Consider the reset Petri net of [DFS98, Example 3] again (Figure 3). Add a new transition $t_5(n_1, n_2, n_3, n_4) = (n_1 + 1, n_2 + 1, n_3 + 1, n_4 + 1)$. The generalized Karp-Miller procedure does not terminate on this modified reset Petri net starting from $s_0 = (1, 1, 0, 0)$, because it already does not terminate on the smaller one of Section 4.1. On the other hand, by fairness, **Clover**$_\mathfrak{S}$ will sooner or later decide to pick a pair of the form (t_5, a) at line (a), and then immediately terminate with the maximal state $(\omega, \omega, \omega, \omega)$, which is the sole element of the clover. $\qquad\square$

Deciding when **Clover**$_\mathfrak{S}$ terminates is itself impossible. We first observe that **Clover**$_\mathfrak{S}$ terminates on each bounded state.

Lemma 5.7. *Let* $\mathfrak{S} = (S, \xrightarrow{F})$ *be an* ∞*-effective complete WSTS, and* $s_0 \in S$ *a state that is* bounded*, i.e., such that the reachability set* $Post^*_\mathfrak{S}(s_0)$ *is finite. Then* **Clover**$_\mathfrak{S}(s_0)$ *terminates.*

Proof. Since $Post^*_\mathfrak{S}(s_0)$ is finite, $g^\infty(s)$ is in $Post^*_\mathfrak{S}(s_0)$ for every $s \in Post^*_\mathfrak{S}(s_0)$ and every $g \in F^*$ with $s \in \mathrm{dom}\,g$. So, defining again A_n as the value of the set A computed by **Clover**$_\mathfrak{S}$ on input s_0, after n iterations of the while statement at line 2, $\bigcup_{n\in\mathbb{N}} A_n$ is contained in $Post^*_\mathfrak{S}(s_0)$, hence finite. By Proposition 5.4, **Clover**$_\mathfrak{S}(s_0)$ terminates. $\qquad\square$

Proposition 5.8. *There is an* ∞*-effective complete WSTS such that we cannot decide, given* $s_0 \in S$*, whether* **Clover**$_\mathfrak{S}(s_0)$ *will terminate.*

Proof. Assume we can decide whether **Clover**$_\mathfrak{S}(s_0)$ terminates.

If **Clover**$_\mathfrak{S}(s_0)$ does not terminate, then $Post^*_\mathfrak{S}(s_0)$ is infinite, by Lemma 5.7.

If on the other hand **Clover**$_\mathfrak{S}(s_0)$ terminates, then it computes the clover $Clover_\mathfrak{S}(s_0)$ by Theorem 5.5, and we can decide boundedness as in the proof of Proposition 4.6, in the case of functional-lossy channel systems: just check whether any of the computed word-products contains a starred atomic expression A^*.
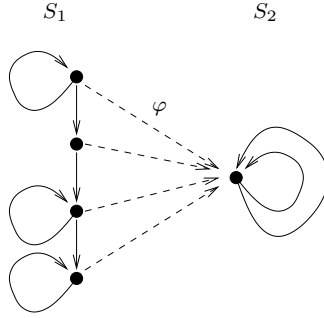
Figure 5: Flattening

In any case, we can decide boundedness, i.e., whether $Post^*_{\mathfrak{G}}(s_0)$ is finite. But this is impossible [CFP96, May03b]. A similar argument works with reset Petri nets, where boundedness is also undecidable [DFS98]. □

5.2. **Clover-Flattable Complete WSTS.** We now characterize those ∞-effective complete WSTS on which **Clover**$_{\mathfrak{G}}$ terminates.

A functional transition system $(\mathfrak{G}, \xrightarrow{F})$ 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.) corresponding composition of maps, i.e., fg denotes $g \circ f$.) Ginsburg and Spanier [GS64] 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. The simplest example of a non-flat system has one state q and two transitions $q \xrightarrow{a} q$ and $q \xrightarrow{b} q$.

For an arbitrary system S, *flattening* [BFLS05] consists in finding a flat system S', equivalent to S with respect to reachability, and in computing on S' instead of S. We adapt the definition in [BFLS05] to functional transition systems, without an explicit finite control graph for now (but see Definition 5.15).

Definition 5.9 (Flattening). A *flattening* of a functional transition system $\mathfrak{G}_2 = (S_2, \xrightarrow{F_2})$ is a pair $(\mathfrak{G}_1, \varphi)$, where:

(1) $\mathfrak{G}_1 = (S_1, \xrightarrow{F_1})$ is a *flat* functional transition system;
(2) and $\varphi : \mathfrak{G}_1 \to \mathfrak{G}_2$ is a *morphism* of transition systems. That is, φ is a pair of two maps, both written φ, from S_1 to S_2 and from F_1 to F_2, such that 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 $\varphi(s') = \varphi(f_1)(\varphi(s))$ (see Figure 5).

Let us recall that a pair $(\mathfrak{G}, s_0)$ of a transition system and a state is *Post**-flattable iff there is a flattening $\mathfrak{G}_1$ of $\mathfrak{G}$ and a state s_1 of $\mathfrak{G}_1$ such that $\varphi(s_1) = s_0$ and $Post^*_{\mathfrak{G}}(s_0) = \varphi(Post^*_{\mathfrak{G}_1}(s_1))$.

Recall that we equate ordered functional transition systems $(S, \xrightarrow{F}, \leq)$ with their underlying function transition system $(S, \xrightarrow{F})$. The notion of flattening then extends to ordered functional transition systems. However, it is then natural to consider *monotonic flattenings*,

where in addition $\varphi : S_1 \to S_2$ is monotonic. In the case of complete transition systems, the natural extension requires φ to be continuous:

Definition 5.10 (Continuous Flattening). Let $\mathfrak{S}_2 = (S_2, \xrightarrow{F_2}, \leq_2)$ be a complete transition system. A flattening $(\mathfrak{S}_1, \varphi)$ of $\mathfrak{S}_2$ is *continuous* iff:

(1) $\mathfrak{S}_1 = (S_1, \xrightarrow{F_1}, \leq_1)$ is a *complete* transition system;
(2) and $\varphi : S_1 \to S_2$ is *continuous*.

Definition 5.11 (Clover-Flattable). Let $\mathfrak{S}$ be a complete transition system, and s_0 be a state. We say that $(\mathfrak{S}, s_0)$ is *clover-flattable* iff there is an continuous flattening $(\mathfrak{S}_1, \varphi)$ of $\mathfrak{S}$, and a state s_1 of $\mathfrak{S}_1$ such that:

(1) $\varphi(s_1) = s_0$ (φ maps initial states to initial states);
(2) $cl(Cover_{\mathfrak{S}}(s_0)) = cl(\varphi\langle cl(Cover_{\mathfrak{S}_1}(s_1))\rangle)$ (φ preserves the closures of the covers of the initial states).

On complete WSTS—our object of study—, the second condition can be simplified to $\downarrow Clover_{\mathfrak{S}}(s_0) = \downarrow \varphi(Clover_{\mathfrak{S}_1}(s_1))$ (using Proposition 3.7 and the fact that φ, as a continuous map, is monotonic), or equivalently to $Clover_{\mathfrak{S}}(s_0) = \mathrm{Max}\,\varphi\langle Clover_{\mathfrak{S}_1}(s_1)\rangle$. Recall also that, when $\mathfrak{S}$ is the completion $\widehat{\mathfrak{X}}$ of a WSTS $\mathfrak{X} = (X, \xrightarrow{F}, \leq)$, the clover of $s_0 \in X$ is a finite description of the *cover* of s_0 in $\mathfrak{X}$ (Proposition 3.9), and this is what φ should preserve, up to taking downward closures.

There are apparently weaker and stronger froms of clover-flattability, which we now introduce. Let us start with the weak form, where equality in the second condition is replaced by inclusion:

Definition 5.12 (Weakly Clover-Flattable). Let $\mathfrak{S}$ be a complete transition system, and s_0 be a state. We say that $(\mathfrak{S}, s_0)$ is *weakly clover-flattable* iff there is an continuous flattening $(\mathfrak{S}_1, \varphi)$ of $\mathfrak{S}$, and a state s_1 of $\mathfrak{S}_1$ such that:

(1) $\varphi(s_1) \leq s_0$;
(2) and $cl(Cover_{\mathfrak{S}}(s_0)) \subseteq cl(\varphi\langle cl(Cover_{\mathfrak{S}_1}(s_1))\rangle)$.

One may simplify the second condition slightly, to: $Cover_{\mathfrak{S}}(s_0) \subseteq cl(\varphi\langle cl(Cover_{\mathfrak{S}_1}(s_1))\rangle)$. In the case of complete WSTS, this is equivalent to $Clover_{\mathfrak{S}}(s_0) \leq^\flat \varphi(Clover_{\mathfrak{S}_1}(s_1))$.

The strong form of clover-flattability uses an explicit finite control graph, as in [BFLS05]. Recall that a *rlre* (restricted linear regular expression) over the alphabet Σ is a regular expression of the form $w_1^* w_2^* ... w_k^*$, where $w_1, w_2, ..., w_k \in \Sigma^*$. The language of an rlre is clearly bounded, and the language $\mathrm{Pfx}(w_1^* w_2^* \ldots w_k^*)$ of prefixes of all words from the latter is then again bounded [GS64].

Recall that a deterministic finite automaton (DFA) is a tuple $\mathcal{A} = (\Sigma, Q, \delta, q_0, Fin)$, where Σ is a finite alphabet, Q is a finite set of so-called *control states*, $q_0 \in Q$ is the *initial* state, $Fin \subseteq Q$ is the set of *final* states, and $\delta : Q \times \Sigma \to Q$ is a partial function called the *transition function*.

One can convert any rlre to a DFA recognizing the same language. For example, Figure 6 displays a DFA for $a^*(bcc)^*(bcaa)^*$ over $\Sigma = \{a, b, c\}$, where final states are circled. The language $\mathrm{Pfx}(a^*(bcc)^*(bcaa)^*)$ is then recognized by the same DFA, except that now all states are final.

This is general: $\mathrm{Pfx}(w_1^* w_2^* \ldots w_k^*)$ is always recognizable by a DFA whose states are all final. Let us therefore call *rl-automaton* any such DFA. Since all states are final, we shall omit the Fin component, and say that $\mathcal{A} = (\Sigma, Q, \delta, q_0)$ itself is an rl-automaton.
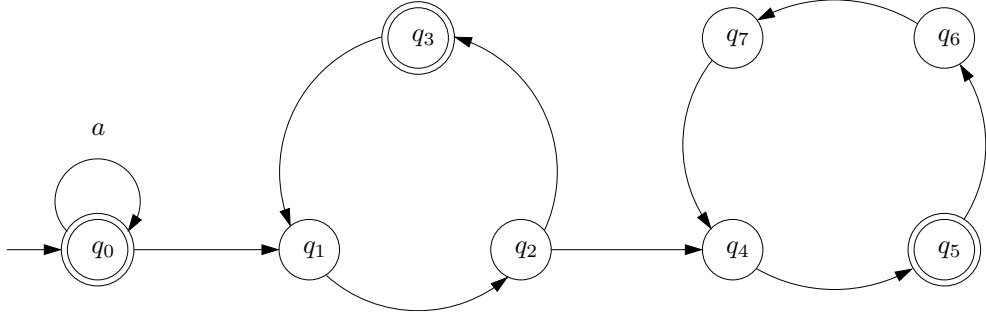
Figure 6: An rl-automaton

Let us define the synchronized product.

Definition 5.13 (Synchronized Product). Let $\mathfrak{S} = (S, \xrightarrow{F}, \leq)$ be a complete functional transition system, and $\mathcal{A} = (F, Q, \delta, q_0)$ be an rl-automaton on the same alphabet F.

Define the *synchronized product* $\mathfrak{S} \times \mathcal{A}$ as the ordered functional transition system $(S \times Q, \xrightarrow{F'}, \leq')$, where F' is the collection of all partial maps $f \bowtie \delta : (s, q) \mapsto (f(s), \delta(q, f))$, for each $f \in F$ such that $\delta(q, f)$ is defined for some $q \in Q$. Let also $(s, q) \leq' (s', q')$ iff $s \leq s'$ and $q = q'$.

Let π_1 be the morphism of transition systems defined as first projection on states; i.e., $\pi_1(s, q) = s$ for all $(s, q) \in S \times Q$, $\pi_1(f \bowtie \delta) = f$ for all $f \in F$.

Lemma 5.14 (Synchronized Product). *Let $\mathfrak{S} = (S, \xrightarrow{F}, \leq)$ be a complete functional transition system, and $\mathcal{A} = (F, Q, \delta, q_0)$ be an rl-automaton on the same alphabet F.*

Then $(\mathfrak{S} \times \mathcal{A}, \pi_1)$ is a continuous flattening of $\mathfrak{S}$.

Proof. First, the technical condition that $\delta(q, f)$ should be defined for some $q \in Q$ only excludes maps $f \bowtie \delta$ with an empty domain, and is therefore benign. This technical condition is needed to define $\pi_1(f \bowtie \delta)$ as f: formally, we define $\pi_1(f')$ for any $f' \in F'$ by letting $\pi_1(f')(s)$ be the first component of the pair $f'(s, q)$, where q is some arbitrary state such that $\delta(q, f)$ is defined, and let $\pi_1(f')(s)$ be undefined otherwise; when $f' = f \bowtie \delta$, such a q exists by the technical condition, and this will yield $f(s)$ when $s \in \mathrm{dom}\, f$, and will be undefined otherwise. So indeed $\pi_1(f \bowtie \delta) = f$.

$(S \times Q, \leq')$ is easily seen to be a dcpo. In fact, it is the disjoint sum of finitely many copies of S, and as such, is a continuous dcpo. It is also well-ordered, as a finite disjoint sum of well-ordered spaces. So $S \times Q$ is a continuous dcwo. Then we check that $f \bowtie \delta$ is partial continuous. Its domain is $\bigcup_{\substack{q \in Q \\ \delta(q,f)\ \text{defined}}} \mathrm{dom}\, f \times \{q\}$, which is open. Moreover $f \bowtie \delta$ is clearly continuous for every $f \in F$: for any directed family $(s_i, q_i)_{i \in I}$ in $\mathrm{dom}(f \bowtie \delta)$, first all q_is must be equal, say $q_i = q \in Q$, and second $(s_i)_{i \in I}$ must be directed in $\mathrm{dom}\, f$. So $f(\mathrm{lub}\{s_i \mid i \in I\}) = \mathrm{lub}\{f(s_i) \mid i \in I\}$, whence $(f \bowtie \delta)(\mathrm{lub}\{(s_i, q) \mid i \in I\}) = (\mathrm{lub}\{f(s_i) \mid i \in I\}, \delta(q, f)) = \mathrm{lub}\{(f \bowtie \delta)(s_i, q) \mid i \in I\}$. That π_1 is continuous is clear as well.

Finally, the language of fireable transitions in $\mathfrak{S} \times \mathcal{A}$ is contained in the language of $\mathcal{A}$, which is of the form $\mathrm{Pfx}(w_1^* w_2^* \ldots w_k^*)$, hence bounded. So $\mathfrak{S} \times \mathcal{A}$ is flat. $\square$

Strong flattenings are special: the decision to take the next action $f \in F$ from state (s, q) is dictated by the current control state q *only*, while ordinary flattenings allow more complex decisions to be made.

We say that a transition system is strongly clover-flattable iff we can require that the flat system $\mathfrak{S}_1$ is a synchronized product, and the continuous morphism of transition systems φ is first projection π_1:

Definition 5.15 (Strongly Clover-Flattable)**.** Let $\mathfrak{S} = (S, \xrightarrow{F})$ be a complete functional transition system. We say that $(\mathfrak{S}, s_0)$ is *strongly clover-flattable* iff there is an rl-automaton $\mathcal{A}$, say with initial state q_0, such that $cl(Cover_{\mathfrak{S}}(s_0)) = cl(\pi_1 \langle cl(Cover_{\mathfrak{S} \times \mathcal{A}}(s_0, q_0)) \rangle)$.

The following is then obvious.

Lemma 5.16. *On complete functional transition systems, the implications "strongly clover-flattable" $\implies$ "clover-flattable" $\implies$ "weakly clover-flattable" hold.* $\square$

It is also easy to show that "weakly clover-flattable" also implies "clover-flattable". However, we shall show something more general in Theorem 5.21 below.

We show in Proposition 5.18 that $\mathbf{Clover}_{\mathfrak{S}}(s_0)$ can only terminate when $(\mathfrak{S}, s_0)$ is strongly clover-flattable. We shall require the following lemma. For notational simplicity, we equate words $g_1 g_2$ with compositions $g_2 \circ g_1$.

Lemma 5.17. *Let* $\mathfrak{S} = (S, \xrightarrow{F})$ *be a complete functional transition system, and* $s_0 \in F$. *Assume* $g_1^\infty g_2^\infty \ldots g_n^\infty(s_0)$ *is defined, and in some open subset* U *of* S, *for some* $g_1, g_2, \ldots, g_n \in F$. *Then there are natural numbers* $k_1, k_2, \ldots, k_n$ *such that* $g_1^{k_1} g_2^{k_2} \ldots g_n^{k_n}(s_0)$ *is defined, and in* U.

Proof. By induction on n. This is clear if $n = 0$. Otherwise, let $s = g_1^\infty g_2^\infty \ldots g_{n-1}^\infty(s_0)$, so that $g_n^\infty(s)$ is defined and in U. If $s < g_n(s)$, then $g_n^\infty(s) = \text{lub}\{g_n^k(s) \mid k \in \mathbb{N}\}$. That the latter is in the Scott-open U implies that $g_n^{k_n}(s)$ is in U for some $k_n \in \mathbb{N}$. If $s \not< g_n(s)$, then $g_n^\infty(s) = g_n(s)$, and we take $k_n = 1$. Let V be the open $(g_n^{k_n})^{-1}(U)$. (Note that, whereas g_n^∞ is not partial continuous in general, $g_n^{k_n}$ is.) So $s = g_1^\infty g_2^\infty \ldots g_{n-1}^\infty(s_0)$ is in V, in each case. We apply the induction hypothesis and obtain the existence of $k_1, k_2, \ldots, k_{n-1}$ such that $g_1^{k_1} g_2^{k_2} \ldots g_{n-1}^{k_{n-1}}(s_0)$ is defined and in V. Hence $g_1^{k_1} g_2^{k_2} \ldots g_n^{k_n}(s_0)$ is defined, and in U, by definition of V. $\square$

Proposition 5.18. *Let* $\mathfrak{S}$ *be an* ∞-*effective complete WSTS. If* $\mathbf{Clover}_{\mathfrak{S}}$ *terminates on* s_0, *then* $(\mathfrak{S}, s_0)$ *is strongly clover-flattable.*

Proof. Write $\mathfrak{S}$ as $(S, \xrightarrow{F}, \leq)$. Assume that $\mathbf{Clover}_{\mathfrak{S}}$ terminates on s_0. Then it returns some finite set A such that $A = Clover_{\mathfrak{S}}(s_0)$ by Theorem 5.5. Enumerate the elements $a_1, \ldots, a_k$ of A. Each element a_i of A, $1 \leq i \leq k$, is obtained as $g_{i1}^\infty g_{i2}^\infty \ldots g_{in_i}^\infty(s_0)$, where each g_{ij} is in F^*.

Build a DFA for the language $\mathcal{L} = g_{11}^* g_{12}^* \ldots g_{1n_1}^* g_{21}^* g_{22}^* \ldots g_{2n_2}^* \ldots g_{k1}^* g_{k2}^* \ldots g_{kn_k}^*$. Make all its states final, so as to obtain an rl-automaton $\mathcal{A}$, with initial state q_0.

We must show that $cl(Cover_{\mathfrak{S}}(s_0)) = cl(\pi_1 \langle cl(Cover_{\mathfrak{S} \times \mathcal{A}}(s_0, q_0)) \rangle)$, i.e., that $\downarrow A = cl(\pi_1 \langle Cover_{\mathfrak{S} \times \mathcal{A}}(s_0, q_0) \rangle)$.

The inclusion from right to left is obvious: for any state (s, q) that is reachable from $\downarrow(s_0, q_0)$ in $\mathfrak{S} \times \mathcal{A}$, s is reachable from $\downarrow s_0$ in $\mathfrak{S}$. So $\pi_1 \langle Post^*_{\mathfrak{S}}(\downarrow s_0) \rangle \subseteq Post^*_{\mathfrak{S} \times \mathcal{A}}(\downarrow(s_0, q_0))$. Taking downward closures yields $\pi_1 \langle Cover^*_{\mathfrak{S}}(s_0) \rangle \subseteq Cover^*_{\mathfrak{S} \times \mathcal{A}}(s_0, q_0)$, and taking closures

yields $cl(\pi_1\langle Cover_{\mathfrak{S}\times\mathcal{A}}(s_0,q_0))\rangle) \subseteq cl(Cover^*_{\mathfrak{S}\times\mathcal{A}}(s_0,q_0)) = {\downarrow}A$ (using Theorem 5.5 and Proposition 3.7).

The other inclusion reduces to showing that for every i, $1 \leq i \leq k$, the ith element a_i of A is in $cl(\pi_1\langle Cover_{\mathfrak{S}\times\mathcal{A}}(s_0,q_0))\rangle)$. It is equivalent to show that every open subset U containing a_i intersects $\pi_1\langle Cover_{\mathfrak{S}\times\mathcal{A}}(s_0,q_0))\rangle$. By Lemma 5.17, there are natural numbers $k_1, k_2, \ldots, k_{n_i}$ such that $g_{i1}^{k_1} g_{i2}^{k_2} \ldots g_{in}^{k_n}(s_0)$ is defined and in U. Since the word $g_{i1}^{k_1} g_{i2}^{k_2} \ldots g_{in}^{k_n}$ is in the language $\mathcal{L}$, $g_{i1}^{k_1} g_{i2}^{k_2} \ldots g_{in}^{k_n}(s_0)$ is the first component of some pair reachable from (s_0,q_0) in $\mathfrak{S}\times\mathcal{A}$. In particular, $g_{i1}^{k_1} g_{i2}^{k_2} \ldots g_{in}^{k_n}(s_0)$ is in $\pi_1\langle Cover_{\mathfrak{S}\times\mathcal{A}}(s_0,q_0))\rangle$. So U intersects $\pi_1\langle Cover_{\mathfrak{S}\times\mathcal{A}}(s_0,q_0))\rangle$, as claimed. $\square$

We now loop the loop and show that $\mathbf{Clover}_{\mathfrak{S}}$ terminates on s_0 whenever $(\mathfrak{S}, s_0)$ is weakly clover-flattable (Theorem 5.21 below). This may seem obvious. In particular, if $(\mathfrak{S}, s_0)$ is clover-flattable, then accelerate along the loops from $\mathfrak{S}_1$, where $\mathfrak{S}_1, \varphi$ is a continuous flattening of $\mathfrak{S}$. The difficulty is that we *cannot* actually choose to accelerate whenever we want: the $\mathbf{Clover}_{\mathfrak{S}}$ procedure decides by itself when it should accelerate, independently of any flattening whatsoever.

There is an added difficulty, in the sense that one should also check that lub-accelerations, as they are used in $\mathbf{Clover}_{\mathfrak{S}}$, are enough to reach all required least upper bounds. The key point is the following lemma, which asserts the existence of finitely many subsequences $g^{p_j+\ell q_j}(s)$, $\ell \in \mathbb{N}$, whose exponents form infinite arithmetic progressions, and which generate all possible limits of directed families of elements of the form $g^n(s)$, $n \in \mathbb{N}$, except possibly for finitely many isolated points.

This is the point in our study where progress is needed. Indeed, we require S to be wpo to pick k and m in the proof below.

Lemma 5.19. *Let S be a dcwo, $g : S \to S$ a partial monotonic map, and $s \in S$. Consider the family G of all elements of the form $g^n(s)$, for those $n \in \mathbb{N}$ such that this is defined. Then there are finitely many directed subfamilies G_0, G_1, $\ldots$, G_{m-1} of G such that:*

(1) $cl(G) = \bigcup_{j=0}^{m-1} cl(G_j) = {\downarrow}\{\mathrm{lub}(G_0), \mathrm{lub}(G_1), \ldots, \mathrm{lub}(G_{m-1})\}$;
(2) *each G_j is either a one-element set $\{g^{p_j}(s)\}$, where $p_j \in \mathbb{N}$, or is a chain of the form $\{g^{p_j+\ell q_j}(s) \mid \ell \in \mathbb{N}\}$, where $p_j \in \mathbb{N}$, $q_j \in \mathbb{N} \setminus \{0\}$, and $g^{p_j}(s) < g^{p_j+q_j}(s)$;*
(3) *for every j, $0 \leq j < m$, $s \not\leq g^{p_j}(s)$.*

Proof. First, the claim is obvious if G is finite, in which case we just take G_1, $\ldots$, G_m to consist of the sets $\{s_1\}$, $\ldots$, $\{s_m\}$, where $G = \{s_1, \ldots, s_m\}$. Write s_j as $g^{p_j}(s)$, and note that it cannot be the case that $s < g^{p_j}(s)$, otherwise $g^{ip_j}(s)$ would be defined for all $i \in \mathbb{N}$ (an easy induction on i, using the fact that the domain of g^{p_j} is upward-closed), contradicting the fact that G is finite. So condition (3) holds.

So assume G is infinite, i.e., $g^n(s)$ is defined for arbitrarily large values of n. Whenever $g^n(s)$ is defined, $g^m(s)$ is, too, for all $m < n$. So $g^n(s)$ is defined for all $n \in \mathbb{N}$, and $G = \{g^n(s) \mid n \in \mathbb{N}\}$. Since S is wpo, for some $k, m \in \mathbb{N}$ with $k < m$, $g^k(s) \leq g^m(s)$. We pick a minimal k such that $g^k(s) \leq g^m(s)$ for some $m > k$; and given k, we pick a minimal $m > k$ such that $g^k(s) \leq g^m(s)$.

Let $G_0 = \{s\}$, $G_1 = \{g(s)\}$, $\ldots$, $G_{k-1} = \{g^{k-1}(s)\}$, $G_k = \{g^{k+i(m-k)}(s) \mid i \in \mathbb{N}\}$, $G_{k+1} = \{g^{k+1+i(m-k)}(s) \mid i \in \mathbb{N}\}$, $\ldots$, $G_{m-1} = \{g^{m-1+i(m-k)}(s) \mid i \in \mathbb{N}\}$.

Each G_j is directed. This is clear when $j < k$. Otherwise, since $g^k(s) \leq g^m(s)$ and g is partial monotonic, we obtain $g^{j+i(m-k)}(s) = g^{j-k+i(m-k)}(g^k(s)) \leq g^{j-k+i(m-k)}(g^m(s)) = g^{j+(i+1)(m-k)}(s)$. So $G_j = (g^{j+i(m-k)}(s))_{i\in\mathbb{N}}$ is an increasing chain.

Condition (2) is satisfied: G_j is a one-element set when $0 \leq j < k$, or when $k \leq j < m$ and $g^j(s) = g^{j+m-k}(s)$, i.e., when the first two elements of G_j are equal; indeed, in the latter case $g^{j+i(m-k)}(s) = g^{i(m-k)}(g^j(s)) = g^{i(m-k)}(g^{j+m-k}(s)) = g^{j+(i+1)(m-k)}(s)$, so all elements of the sequence coincide. Otherwise, i.e., if $k \leq j < m$ and $g^j(s) \neq g^{j+m-k}(s)$ (in which case $g^j(s) < g^{j+m-k}(s)$, since $g^{j+i(m-k)}(s) \leq g^{j+(i+1)(m-k)}(s)$ for all i), let $p_j = j$ and $q_j = m - k$.

Let us establish condition (1). First, $G = \bigcup_{j=0}^{m-1} G_j$. In particular, $G_j \subseteq G$, so $cl(G_j) \subseteq cl(G)$ for all j, whence $\bigcup_{j=0}^{m-1} cl(G_j) \subseteq cl(G)$.

Next, let $s_j = \mathrm{lub}(G_j)$ for all j, $0 \leq j < m$. This exists because G_j is a chain, hence is directed, and S is a dcpo. The finite union $\bigcup_{j=0}^{m-1} \downarrow s_j$ is closed, and contains $\bigcup_{j=0}^{m-1} G_j = G$, so it contains $cl(G)$. Conversely, the definition of s_j makes it clear that $s_j \in cl(G_j) \subseteq cl(G)$. So $cl(G) = \bigcup_{j=0}^{m-1} \downarrow s_j = \downarrow \{s_0, s_1, \ldots, s_{m-1}\}$.

Take any element x in $cl(G)$. Since $x \in cl(G)$, $x \leq s_j$ for some j, $0 \leq j < m$. However, $s_j \in cl(G_j)$, and $cl(G_j)$ is downward-closed, so $x \in \bigcup_{j=0}^{m-1} cl(G_j)$. So $cl(G) \subseteq \bigcup_{j=0}^{m-1} cl(G_j)$. So condition (1) holds.

Finally, assume condition (3) failed. Then $s < g^j(s)$ for some j, $0 \leq j < m$. Certainly $j \neq 0$, since $g^0(s) = s$. By the minimality of k such that $g^k(s) \leq g^m(s)$ for some $m > k$, $k = 0$. By the minimality of m, $m \leq j$. But this contradicts $j < m$. $\square$

Proposition 5.20. *Let $\mathfrak{S}$ be an ∞-effective complete WSTS. Assume that $(\mathfrak{S}, s_0)$ is weakly clover-flattable. Then $\mathbf{Clover}_{\mathfrak{S}}$ terminates on s_0.*

Proof. Let $\mathfrak{S}_1$, φ be a continuous flattening of $\mathfrak{S}$, and s_1 be a state of $\mathfrak{S}_1$ such that $\varphi(s_1) \leq s_0$ and $Cover_{\mathfrak{S}}(s_0) \subseteq \downarrow \varphi \langle Cover_{\mathfrak{S}_1}(s_1) \rangle$, i.e., $Clover_{\mathfrak{S}}(s_0) \leq^{\flat} \varphi \langle Clover_{\mathfrak{S}_1}(s_1) \rangle$. Write $\mathfrak{S}_1$ as $(S_1, \xrightarrow{F_1}, \leq)$. Since $\mathfrak{S}_1$ is flat, every $g_1 \in F_1$ is in $w_1^* w_2^* \ldots w_m^*$, for some fixed sequence $w_1, w_2, \ldots, w_m \in F_1^*$.

Extend the action of $\varphi : F_1 \to F$ on words by $\varphi(f_1 f_2 \ldots f_p) = \varphi(f_1)\varphi(f_2) \ldots \varphi(f_p)$. Thus $\varphi(w_1)$, ..., $\varphi(w_m)$ are defined.

Consider first $\varphi(w_1)$. Apply Lemma 5.19 with $g = \varphi(w_1)$ and $s = s_0$, and get finitely many subfamilies of G_0, G_1, ..., G_{m-1} of $G = \{\varphi(w_1)^n(s_0) \mid n \in \mathbb{N}, \varphi(w_1)^n(s_0) \text{ is defined}\}$ satisfying the conditions given in the Lemma.

For each j such that G_j is a one-element set, say $G_j = \{\varphi(w_1)^n(s_0)\}$, observe that $\mathbf{Clover}_{\mathfrak{S}}$ will eventually select the pair $(\varphi(w_1)^n, s_0)$ at line 2.(a) by fairness, and add $(\varphi(w_1)^n)^{\infty}(s_0)$ to A. By condition (3), $s_0 \not< \varphi(w_1)^n(s_0)$, so $(\varphi(w_1)^n)^{\infty}(s_0) = \varphi(w_1)^n(s_0)$. So $\mathbf{Clover}_{\mathfrak{S}}$ will eventually add $\varphi(w_1)^n(s_0) = \mathrm{lub}(G_j)$ to A.

Still taking the notations of the Lemma, for every j such that G_j contains more than one element, $\mathbf{Clover}_{\mathfrak{S}}$ will eventually select the pair $(\varphi(w_1)^{p_j}, s_0)$, adding $(\varphi(w_1)^{p_j})^{\infty}(s_0)$ to A. Using condition (3) as above, one sees that $(\varphi(w_1)^{p_j})^{\infty}(s_0) = \varphi(w_1)^{p_j}(s_0)$. Then, by fairness again (and this is the important point in the proof, where lub-acceleration is needed), $\mathbf{Clover}_{\mathfrak{S}}$ will eventually select the pair $(\varphi(w_1)^{q_j}, \varphi(w_1)^{p_j}(s_0))$, and therefore add $(\varphi(w_1)^{q_j})^{\infty}(\varphi(w_1)^{p_j}(s_0))$ to A. By condition (2), $(\varphi(w_1)^{q_j})^{\infty}(\varphi(w_1)^{p_j}(s_0))$ is just $\mathrm{lub}\{\varphi(w_1)^{p_j + \ell q_j}(s_0) \mid \ell \in \mathbb{N}\} = \mathrm{lub}(G_j)$.

Let again A_n be the value of the set A, computed by the procedure $\mathbf{Clover}_{\mathfrak{S}}$ on input s_0, after n iterations of the while statement at line 2. Let $A = \bigcup_{n \in \mathbb{N}} A_n$. We have just shown that at some step, say n_1, $\mathbf{Clover}_{\mathfrak{S}}$ will have added enough elements to A so that every element of the form $\varphi(w_1)^{k_1}(s_0)$, $k_1 \in \mathbb{N}$ (provided this is defined), is below some element of A_{n_1}.

Let us proceed with $\varphi(w_2)$. Fix an arbitrary element s of A_{n_1}, and apply Lemma 5.19 with $g = \varphi(w_2)$. Proceeding as above, we observe that there is an $n_2 \geq n_1$ such that every element of the form $\varphi(w_2)^{k_2}(s)$, $n \in \mathbb{N}$, is below some element of A_{n_2}. Since s is arbitrary in A_{n_1}, we conclude that every element of the form $\varphi(w_2)^{k_2}(\varphi(w_1)^{k_1}(s_0))$, $k_1, k_2 \in \mathbb{N}$, is below some element of A_{n_2}.

We now induct on i, $1 \leq i \leq m$, to show similarly that there is an $n_i \in \mathbb{N}$ such that every element of the form $\varphi(w_i)^{k_i}(\varphi(w_{i-1})^{k_{i-1}}(\ldots \varphi(w_1)^{k_1}(s_0)))$, where $k_1, \ldots, k_i \in \mathbb{N}$, is below some element of A_{n_i}.

In particular, for $i = m$, writing n for n_m: $(*)$ there is an $n \in \mathbb{N}$ such that every element of the form $\varphi(w_m)^{k_m}(\varphi(w_{m-1})^{k_{m-1}}(\ldots \varphi(w_1)^{k_1}(s_0)))$, where $k_1, \ldots, k_m \in \mathbb{N}$, is below some element of A_n. We claim that $\mathbf{Clover}_{\mathfrak{S}}(s_0)$ must stop after step n.

Let U be the (open) complement of the closed set $\downarrow A_n$, and assume that U intersects $\downarrow Clover_{\mathfrak{S}}(s_0)$. Then U must also intersect $\downarrow \varphi \langle Clover_{\mathfrak{S}_1}(s_1) \rangle$, hence $\varphi \langle Clover_{\mathfrak{S}_1}(s_1) \rangle$. (Remember that open subsets are upward-closed.) So $\varphi^{-1}(U)$ intersects $Clover_{\mathfrak{S}_1}(s_1)$, whence $\varphi^{-1}(U)$ intersects $\downarrow Clover_{\mathfrak{S}_1}(s_1)$, since $\varphi^{-1}(U)$ is upward-closed, using the fact that U is and that φ is monotonic. By Proposition 3.7, $\varphi^{-1}(U)$ intersects $cl(Cover_{\mathfrak{S}_1}(s_1))$. Since φ is continuous, $\varphi^{-1}(U)$ is open. We now use the fact that an open intersects the closure of a set iff it intersects that set. So $\varphi^{-1}(U)$ must intersect $Cover_{\mathfrak{S}_1}(s_1)$. So U intersects $\varphi \langle Cover_{\mathfrak{S}_1}(s_1) \rangle$, say at a. In particular, there is an $a_1 \in S_1$ such that $a \leq \varphi(a_1)$, and $a_1 \leq w_1^{k_1} w_2^{k_2} \ldots w_m^{k_m}(s_1)$, for some natural numbers $k_1, k_2, \ldots, k_m$.

Since $a \leq \varphi(a_1) \leq \varphi(w_1^{k_1} w_2^{k_2} \ldots w_m^{k_m})(\varphi(s_1)) \leq \varphi(w_1^{k_1} w_2^{k_2} \ldots w_m^{k_m})(s_0) = \varphi(w_m)^{k_m} (\varphi(w_{m-1})^{k_{m-1}}(\ldots \varphi(w_1)^{k_1}(s_0)))$, a is in $\downarrow A_n$ by $(*)$. But this contradicts the fact that $a \in U$. So the complement U of $\downarrow A_n$ does not intersect $\downarrow Clover_{\mathfrak{S}}(s_0)$, i.e., $\downarrow Clover_{\mathfrak{S}}(s_0) \subseteq \downarrow A_n$.

By Proposition 5.3, the converse inclusion holds. We conclude that the procedure $\mathbf{Clover}_{\mathfrak{S}}$ stops after the nth turn of the loop, because of the fixpoint test at line 2. $\square$

Putting together Lemma 5.16, Proposition 5.18, and Proposition 5.20, we obtain:

Theorem 5.21 (Main Theorem). *Let $\mathfrak{S}$ be an ∞-effective complete WSTS. The following statements are equivalent:*

(1) *$(\mathfrak{S}, s_0)$ is clover-flattable;*
(2) *$(\mathfrak{S}, s_0)$ is weakly clover-flattable;*
(3) *$(\mathfrak{S}, s_0)$ is strongly clover-flattable;*
(4) *$\mathbf{Clover}_{\mathfrak{S}}(s_0)$ terminates.* $\square$

5.3. **Cover-flattability (without the "l" in "Cover").** Turning to non-complete WSTS, we define:

Definition 5.22 (Monotonic Flattening). Let $\mathfrak{X}_2 = (X_2, \xrightarrow{F_2}, \leq_2)$ be an ordered functional transition system. A flattening $(\mathfrak{X}_1, \varphi)$ of $\mathfrak{X}_2$ is *monotonic* iff:

(1) $\mathfrak{X}_1 = (X_1, \xrightarrow{F_1}, \leq_1)$ is an ordered functional transition system;
(2) and $\varphi : X_1 \to X_2$ is *monotonic*.

Definition 5.23 (Cover-Flattable). Let $\mathfrak{X}$ be an ordered functional transition system, and x_0 be a state. We say that $(\mathfrak{X}, x_0)$ is *cover-flattable* iff there is a monotonic flattening $(\mathfrak{X}_1, \varphi)$ of $\mathfrak{X}$, and a state x_1 of $\mathfrak{X}_1$ such that:

(1) $\varphi(x_1) = x_0$;

(2) $Cover_{\mathfrak{X}}(x_0) = \downarrow\varphi\langle Cover_{\mathfrak{X}_1}(x_1)\rangle$.

Definition 5.24 (Weakly Cover-Flattable). Let $\mathfrak{X}$ be an ordered functional transition system, and x_0 be a state. We say that $(\mathfrak{X}, x_0)$ is *weakly cover-flattable* iff there is a monotonic flattening $(\mathfrak{X}_1, \varphi)$ of $\mathfrak{X}$, and a state x_1 of $\mathfrak{X}_1$ such that:

(1) $\varphi(x_1) \leq x_0$;
(2) and $Cover_{\mathfrak{X}}(x_0) \subseteq \downarrow\varphi\langle Cover_{\mathfrak{X}_1}(x_1)\rangle$.

Definition 5.25 (Strongly Cover-Flattable). Let $\mathfrak{X} = (X, \xrightarrow{F})$ be an ordered functional transition system. We say that $(\mathfrak{X}, x_0)$ is *strongly cover-flattable* iff there is an rl-automaton $\mathcal{A}$, say with initial state q_0, such that $Cover_{\mathfrak{X}}(x_0) = \pi_1\langle Cover_{\mathfrak{X}\times\mathcal{A}}(x_0, q_0)\rangle$.

Theorem 5.26. *Let $\mathfrak{X} = (X, \xrightarrow{F}, \leq)$ be an ω^2-WSTS that is ∞-effective, in the sense that $\widehat{\mathfrak{X}}$ is ∞-effective, i.e., that $(\mathcal{S}g)^{\infty}$ is computable for every $g \in F^*$. The following statements are equivalent:*

(1) *$(\mathfrak{X}, x_0)$ is cover-flattable;*
(2) *$(\mathfrak{X}, x_0)$ is weakly cover-flattable;*
(3) *$(\mathfrak{X}, x_0)$ is strongly cover-flattable;*
(4) *$(\widehat{\mathfrak{X}}, \eta_X(x_0))$ is (weakly, strongly) clover-flattable;*
(5) **Clover**$_{\widehat{\mathfrak{X}}}(\eta_X(x_0))$ *terminates.*

In this case, **Clover**$_{\widehat{\mathfrak{X}}}(\eta_X(x_0))$ *returns the clover $A = Clover_{\mathfrak{S}}(s_0)$, and this is a finite description of the cover, in the sense that $Cover_{\mathfrak{X}}(x_0) = \eta_X^{-1}(\downarrow A)$.*

Proof. First, that **Clover**$_{\widehat{\mathfrak{X}}}(\eta_X(x_0))$ computes the clover A is Theorem 5.5, and the fact that $Cover_{\mathfrak{X}}(x_0) = \eta_X^{-1}(\downarrow A)$, by Proposition 3.9. If we equate X with $\eta_X\langle X\rangle$, the latter means that the cover is just $X \cap \downarrow A$.

Next, (4) is equivalent to (5), by Theorem 5.21. Note in particular that $\widehat{\mathfrak{X}}$ is a complete WSTS by Theorem 4.4, and is ∞-effective by assumption.

The implications (1) $\Longrightarrow$ (2) and (3) $\Longrightarrow$ (1) are clear. For the latter, note that, since $Cover_{\mathfrak{X}\times\mathcal{A}}(x_0, q_0)$ is downward-closed, $\pi_1\langle Cover_{\mathfrak{X}\times\mathcal{A}}(x_0, q_0)\rangle = \downarrow\pi_1\langle Cover_{\mathfrak{X}\times\mathcal{A}}(x_0, q_0)\rangle$, and take $\varphi = \pi_1$.

We now show that (2) implies (4), i.e., that if $(\mathfrak{X}, x_0)$ is weakly cover-flattable, then $(\widehat{\mathfrak{X}}, \eta_X(x_0))$ is weakly clover-flattable. So let $\mathfrak{X}_1 = (X_1, \xrightarrow{F_1}, \leq)$, φ and x_1 as in Definition 5.24. In particular, $\varphi(x_1) \leq x_0$ and $Cover_{\mathfrak{X}}(x_0) \subseteq \downarrow\varphi\langle Cover_{\mathfrak{X}_1}(x_1)\rangle$. Let S_1 be the ideal completion $\mathrm{Idl}(\mathfrak{X}_1)$, with inclusion as ordering, and define the complete transition system $\mathfrak{S}_1 = (S_1, \xrightarrow{F_1'}, \subseteq)$, where $F_1' = \{\mathrm{Idl}(f) \mid f \in F_1\}$. $\mathrm{Idl}(f)$ is the partial continuous function that maps every ideal D such that $D \cap \mathrm{dom}\, f \neq \emptyset$ to $\downarrow f\langle D\rangle$. Remember that $\widehat{\mathfrak{X}} = \mathrm{Idl}(\mathfrak{X})$. Define $\varphi' : S_1 \to \widehat{\mathfrak{X}}$ as $\mathrm{Idl}(\varphi)$: this is continuous. On transitions, φ' maps $\mathrm{Idl}(f)$ to $\mathrm{Idl}(\varphi(f))$: this is well-defined, as one can recover f from $\mathrm{Idl}(f)$, by the fact that $f(x) = \mathrm{lub}(\mathrm{Idl}(f)(\downarrow x))$. So $(\mathfrak{S}_1, \varphi')$ is a continuous flattening of $\widehat{\mathfrak{X}}$. Let $s_1 = \downarrow x_1$, $s_0 = \downarrow x_0$. We claim that $\varphi'(s_1) \subseteq s_0$, and that $Cover_{\widehat{\mathfrak{X}}}(s_0) \subseteq cl(\varphi'\langle cl(Cover_{\mathfrak{S}_1}(s_1))\rangle)$. The first inequality is because $\varphi'(s_1) = \mathrm{Idl}(\varphi)(\downarrow x_1) = \downarrow\varphi\langle\downarrow x_1\rangle = \downarrow\varphi(x_1) \subseteq \downarrow x_0 = s_0$, since $\varphi(x_1) \leq x_0$. For the second inequality, let s be any element of $Cover_{\widehat{\mathfrak{X}}}(s_0)$. So $s \subseteq g(s_0)$ for some $g \in F_1'^*$. We observe that Idl is a functor, i.e., that Idl of the identity map is the identity, and that $\mathrm{Idl}(g_1 g_2) = \mathrm{Idl}(g_1)\,\mathrm{Idl}(g_2)$ for all g_1, g_2. So, writing g as a composition $g_1 g_2 \ldots g_k$ of elements $g_i = \mathrm{Idl}(h_i)$ of F_1', g equals $\mathrm{Idl}(h)$, where $h = h_1 h_2 \ldots h_k \in F_1^*$. It follows that $s \subseteq \mathrm{Idl}(h)(\downarrow x_0) = \downarrow h(x_0)$. Observe that $h(x_0) \in Cover_{\mathfrak{X}}(x_0) \subseteq \downarrow\varphi\langle Cover_{\mathfrak{X}_1}(x_1)\rangle$,

so $s \subseteq \downarrow \varphi \langle Cover_{\mathfrak{X}_1}(x_1) \rangle$. In particular, every element x of the ideal s is below some element of the form $\varphi(f(x_1))$, $f \in F_1^*$. We observe that $x \in \varphi'(\mathrm{Idl}(f)(s_1))$: indeed, $\varphi'(\mathrm{Idl}(f)(s_1)) = \mathrm{Idl}(\varphi)(\mathrm{Idl}(f)(s_1)) = \mathrm{Idl}(\varphi \circ f)(s_1) = \downarrow(\varphi \circ f)\langle \downarrow x_1 \rangle = \downarrow \varphi(f(x_1))$, and x is in the latter since $x \leq \varphi(f(x_1))$. >From $x \in \varphi'(\mathrm{Idl}(f)(s_1))$, and since $\mathrm{Idl}(f)(s_1) \in Cover_{\mathfrak{S}_1}(s_1)$, we deduce that $x \in \varphi'\langle Cover_{\mathfrak{S}_1}(s_1) \rangle$. Since x is arbitrary in s, $s \subseteq \varphi'\langle Cover_{\mathfrak{S}_1}(s_1) \rangle$, i.e., $s \in \downarrow \varphi'\langle Cover_{\mathfrak{S}_1}(s_1) \rangle \subseteq cl(\varphi'\langle cl(Cover_{\mathfrak{S}_1}(s_1)) \rangle)$.

Finally, we show that (4) implies (3), i.e., that if $(\widehat{\mathfrak{X}}, \eta_X(x_0))$ is strongly clover-flattable, then $(\mathfrak{X}, x_0)$ is strongly cover-flattable. Let $\mathcal{A}$ be an rl-automaton, with initial state q_0, such that $cl(Cover_{\widehat{\mathfrak{X}}}(\eta_X(x_0))) = cl(\pi_1 \langle cl(Cover_{\widehat{\mathfrak{X}} \times \mathcal{A}}(\eta_X(x_0), q_0)) \rangle)$. We claim that $Cover_{\mathfrak{X}}(x_0) = \pi_1 \langle Cover_{\mathfrak{X} \times \mathcal{A}'}(x_0, q_0) \rangle$, where $\mathcal{A}'$ is the automaton obtained from $\mathcal{A}$ by replacing each $\mathcal{S}g$ transition by a g transition, $g \in F$. (Note by the way that the definition of $\mathcal{S}g$ is the same as that of $\mathrm{Idl}(g)$ above.) The inclusion from right to left is obvious, so let us show that $Cover_{\mathfrak{X}}(x_0) \subseteq \pi_1 \langle Cover_{\mathfrak{X} \times \mathcal{A}'}(x_0, q_0) \rangle$. Let x be any element of $Cover_{\mathfrak{X}}(x_0)$. So $x \leq g(x_0)$ for some $g \in F$. Then $x \in \downarrow g(x_0) = \downarrow g \langle \downarrow x_0 \rangle = \mathrm{Idl}(g)(\eta_X(x_0))$, so $\downarrow x \in Cover_{\widehat{\mathfrak{X}}}(\eta_X(x_0))$. By assumption $\downarrow x$ is in $cl(\pi_1 \langle cl(Cover_{\widehat{\mathfrak{X}} \times \mathcal{A}}(\eta_X(x_0), q_0)) \rangle)$. We may simplify this by observing that $cl(f \langle cl(A) \rangle) = cl(f \langle A \rangle)$ for any continuous map f and any subset A, so that $\downarrow x \in cl(\pi_1 \langle Cover_{\widehat{\mathfrak{X}} \times \mathcal{A}}(\eta_X(x_0), q_0) \rangle)$. In $\widehat{X} = \mathrm{Idl}(X)$, the closure $cl(A)$ of any downward-closed subset A of $\mathrm{Idl}(X)$ equals $\mathrm{Lub}(A)$, since $\mathrm{Idl}(X)$ is continuous. It follows that, if $\downarrow x \in cl(A)$, then $\downarrow x$ is the union of a directed family $(s_i)_{i \in I}$ of elements of A; in particular, x is in some s_i, $i \in I$, i.e., x is in some element (an ideal) of A. Taking $A = \pi_1 \langle Cover_{\widehat{\mathfrak{X}} \times \mathcal{A}}(\eta_X(x_0), q_0) \rangle$, x is in some ideal s such that $(s, q) \in Cover_{\widehat{\mathfrak{X}} \times \mathcal{A}}(\eta_X(x_0), q_0)$ for some state q of $\mathcal{A}$. That is, $s \subseteq \mathcal{S}g(\eta_X(x_0))$ for some $g = g_1 g_2 \ldots g_k$, where $g_1, g_2, \ldots, g_k \in F$, and q is the state obtained by reading the word $\mathcal{S}g_1 \mathcal{S}g_2 \ldots \mathcal{S}g_k$ in $\mathcal{A}$ from q_0. In particular, q is also the state obtained by reading the word $g_1 g_2 \ldots g_k$ in $\mathcal{A}'$ from q_0. And $s \subseteq \mathcal{S}g(\eta_X(x_0))$ means that $s \subseteq \downarrow g \langle \downarrow x_0 \rangle = \downarrow g(x_0)$, so $x \in s$ implies $x \leq g(x_0)$. In particular, $(x, q) \in Cover_{\mathfrak{X} \times \mathcal{A}'}(x_0, q_0)$, so $x \in \pi_1 \langle Cover_{\mathfrak{X} \times \mathcal{A}'}(x_0, q_0) \rangle$. $\qquad \square$

By a slight abuse of language, say that a functional WSTS $\mathfrak{S} = (S, \xrightarrow{F}, \leq)$ is cover-flattable iff $(\mathfrak{S}, s_0)$ is cover-flattable for every initial state $s_0 \in S$.

Corollary 5.27. *Every Petri net, and every VASS, is cover-flattable.*

Proof. The state space of a Petri net on k places is $\mathbb{N}^k$, that of a VASS [HP79] is $Q \times \mathbb{N}^k$, where Q is a finite set of control states. We deal with the latter, as they are more general. Transitions of the VASS $\mathfrak{X}$ are of the form $f(q, \vec{x}) = (q', \vec{x} + \vec{b} - \vec{a})$, provided $\vec{x} \geq \vec{a}$, and where $\vec{a}, \vec{b}$ are fixed tuples in $\mathbb{N}^k$. It is easy to see that $\mathcal{S}f$ is defined by: $\mathcal{S}f(q, \vec{x}) = (q', \vec{x} + \vec{b} - \vec{a})$, provided $\vec{x} \geq \vec{a}$, this time for all $\vec{x} \in \mathbb{N}_\omega^k$. So the completion $\widehat{\mathfrak{S}}$ of the VASS is ∞-effective. On these, the Karp-Miller algorithm terminates [KM69], hence also the generalized Karp-Miller algorithm of Section 4.1. By Proposition 5.6, **Clover**$_{\widehat{\mathfrak{S}}}$ terminates on any input $s_0 \in Q \times \mathbb{N}_\omega^k$. So $\mathfrak{X}$ is cover-flattable, by Theorem 5.26. $\qquad \square$

Corollary 5.28. *There are reset Petri nets, and functional-lossy channel systems that are not cover-flattable.*

Proof. One can again show that their completions are ∞-effective, see Section 4.5. However the cover is undecidable both for reset Petri nets and (functional-)lossy channel systems $\mathfrak{X}$, so **Clover**$_{\widehat{\mathfrak{X}}}(\eta_X(x_0))$ must fail to terminate for some initial state x_0. We conclude by Theorem 5.26. $\qquad \square$

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 monotonic functions with upward-closed definition domains are complete (strongly monotonic) WSTS. This result is obtained by showing that every monotonic affine function f is continuous and its lub-acceleration f^∞ is computable [CFS11]. Moreover, we prove that it is possible to decide whether a general counter system (given by a finite set of Presburger relations) is a monotonic affine counter system, but that one cannot decide whether it is a WSTS.

Definition 6.1. A *relational counter system* (with n counters), for short an *R-counter system*, $\mathcal{C}$ is a tuple $\mathcal{C} = (Q, R, \rightarrow)$ 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 $\rightarrow \subseteq Q \times R \times Q$.

We will consider a special case of Presburger relations, those which allow us to encode 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 $\vec{x} \in \text{dom} f, f(\vec{x}) = A\vec{x} + \vec{b}$. When necessary, we will extend affine maps $f : \mathbb{N}^n \longrightarrow \mathbb{N}^n$ by continuity to $f : \mathbb{N}_\omega^n \longrightarrow \mathbb{N}_\omega^n$, by $f(\text{lub}_{i \in \mathbb{N}}(\vec{x}_i)) = \text{lub}_{i \in \mathbb{N}}(f(\vec{x}_i))$ for every countable chain $(\vec{x}_i)_{i \in \mathbb{N}}$ in $\mathbb{N}^n$. That is, we just write f instead of $\mathcal{S}f$.

Definition 6.2. An *affine counter system* (with n counters), a.k.a. an *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 6.3. *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 monotonic 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 $f(\vec{x}) = A\vec{x} + \vec{b}$ with $A \in \mathbb{N}^{n \times n}$ and $\vec{b} \in \mathbb{Z}^n$, using results by Demri *et al.* [DFGvD06]. □

For counter systems (which include Minsky machines), monotonicity is undecidable. Clearly, a counter system $\mathfrak{S}$ is well-structured iff $\mathfrak{S}$ is monotonic: 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 $\text{dom} f$ is upward-closed, since all matrices A have non-negative coefficients. This is easily cast as Presburger formula, and therefore decidable.

Proposition 6.4. *There is an algorithm to decide whether an ACS is a strongly monotonic WSTS.*

Proof. The strong monotony of an ACS $\mathcal{C}$ means that every function of $\mathcal{C}$ is monotonic and this can be expressed by a Presburger formula saying that all the (Presburger-definable) definition domains are upward-closed (the matrices are known to be positive). □

We have recalled that the transitions function of Petri nets ($f(x) = x + \vec{b}$, $\vec{b} \in \mathbb{Z}^n$ and 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 [EN98] and to another class of monotonic affine functions $f(\vec{x}) = A\vec{x} + \vec{b}$ such that $A \in \mathbb{N}^{n \times n}$, $b \in \mathbb{N}^n$ (note that b is not in $\mathbb{Z}^n$) and dom(f) is upward closed [FMP04].

[CFS11] recently extended this result to all monotonic affine functions: for every $f(\vec{x}) = A\vec{x} + \vec{b}$ with $A \in \mathbb{N}^{n \times n}$, $\vec{b} \in \mathbb{Z}^n$ and dom(f) upward-closed, the function f^∞ is recursive.

We deduce the following strong relationship between well-structured ACS and complete well-structured ACS.

Theorem 6.5. *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 $Post(\vec{s}) = \{\vec{t} \mid \exists f \in F \cdot f(\vec{t}) = \vec{s}\}$ is Presburger-definable. Note also that monotonic affine function are continuous, and $\mathbb{N}_\omega^n$ is a continuous dcwo. Finally, for every Presburger monotonic affine function f, the function f^∞ is recursive, so the considered *ACS* is ∞-effective. $\qquad\square$

Corollary 6.6. *One can decide whether the completion of an ACS is an ∞-effective complete WSTS.*

So the completions of reset/transfer Petri nets [DFS98], broadcast protocols [EFM99], self-modifying Petri nets [Val78] and affine well-structured nets [FMP04] 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 general completion of WSTS defined in [FG09] is still a WSTS, iff the original WSTS is an ω^2-*WSTS*. This delineates a new, robust class of WSTS: all known WSTS are ω^2-WSTS. The property of being an ω^2-WSTS is also important to ensure progress in Karp-Miller-like procedures.

We have also defined a simple procedure, **Clover**$_\mathfrak{S}$ for computing the clover for ∞-effective complete WSTS $\mathfrak{S}$. This captures the essence of generalized forms of the Karp-Miller procedure, while terminating in more cases. We have shown that that **Clover**$_\mathfrak{S}$ terminates iff the WSTS is *clover-flattable*, i.e., that it is some form of projection of a flat system, with the same clover. We have also shown that several variants of the notion of clover-flattability were in fact equivalent. We believe that this characterization is an important, and non-trivial result.

In the future, we shall explore efficient strategies for choosing sequences $g \in F^*$ to lub-accelerate in the **Clover**$_\mathfrak{S}$ procedure. We will also analyze whether **Clover**$_\mathfrak{S}$ terminates in models such as BVASS [VG05], reconfigurable nets, timed Petri nets [ADMN04a], post-self-modifying Petri nets [Val78] and strongly monotonic affine well-structured nets [FMP04]), i.e., whether they are cover-flattable.

One potential use of the clover is in deciding coverability. But the **Clover**$_\mathfrak{G}$ procedure may fail to terminate. This is in contrast to the Expand, Enlarge and Check forward algorithm of [GRvB07], which always terminates, hence decides coverability. One may want to combine the best of both worlds, and the lub-accelerations of **Clover**$_\mathfrak{G}$ can profitably be used to improve the efficiency of the Expand, Enlarge and Check algorithm. This remains to be explored.

Finally, recall that computing the finite clover is a first step [EN98] in the direction of solving liveness properties (and not only safety properties which reduce to coverability). We plan to clarify the construction of a *cloverability graph* which would be the basis for liveness model checking.

Acknowledgement

References

[ABJ98] Parosh Abdulla, Ahmed Bouajjani, and Bengt Jonsson. On-the-fly analysis of systems with unbounded, lossy Fifo channels. In *Proc. 10th Intl. Conf. Computer Aided Verification (CAV'98)*, pages 305–318, Vancouver, Canada, June 1998. Springer Verlag LNCS 1427.

[ACABJ04] Parosh Aziz Abdulla, Aurore Collomb-Annichini, Ahmed Bouajjani, and Bengt Jonsson. Using forward reachability analysis for verification of lossy channel systems. *Formal Methods in System Design*, 25(1):39–65, 2004.

[AČJT00] Parosh Aziz Abdulla, Karlis Čerāns, Bengt Jonsson, and Yih-Kuen Tsay. Algorithmic analysis of programs with well quasi-ordered domains. *Information and Computation*, 160(1–2):109–127, 2000.

[ADMN04a] Parosh Aziz Abdulla, Johann Deneux, Pritha Mahata, and Aletta Nylén. Forward reachability analysis of timed Petri nets. In *FORMATS/FTRTFT*, pages 343–362. Springer Verlag LNCS 3253, 2004.

[ADMN04b] Parosh Aziz Abdulla, Johann Deneux, Pritha Mahata, and Aletta Nylén. Forward reachability analysis of timed Petri nets. In Yassine Lakhnech and Sergio Yovine, editors, *FORMATS/FTRTFT*, volume 3253 of *Lecture Notes in Computer Science*, pages 343–362. Springer, 2004.

[AJ94] Samson Abramsky and Achim Jung. Domain theory. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 3, pages 1–168. Oxford University Press, 1994.

[AN00] Parosh Aziz Abdulla and Aletta Nylén. Better is better than well: On efficient verification of infinite-state systems. In *Proc. 14th IEEE Symp. Logic in Computer Science (LICS'00)*, pages 132–140, 2000.

[BFLS05] Sébastien Bardin, Alain Finkel, Jérôme Leroux, and Philippe Schnoebelen. Flat acceleration in symbolic model checking. In *Proc. 3rd Intl. Symp. Automated Technology for Verification and Analysis (ATVA'05)*, pages 474–488. Springer Verlag LNCS 3707, 2005.

[BG11] Laura Bozzelli and Pierre Ganty. Complexity analysis of the backward coverability algorithm for vass. In *Proceedings of the 5th international conference on Reachability problems*, RP'11, pages 96–109, Berlin, Heidelberg, 2011. Springer-Verlag.

[CFP96] Gérard Cécé, Alain Finkel, and S. Purushothaman Iyer. Unreliable channels are easier to verify than perfect channels. *Information and Computation*, 124(1):20–31, January 1996.

[CFS11] Pierre Chambart, Alain Finkel, and Sylvain Schmitz. Forward analysis and model checking for trace bounded WSTS. In Lars M. Kristensen and Laure Petrucci, editors, *Proceedings of the 32nd International Conference on Applications and Theory of Petri Nets (ICATPN'11)*,

volume 6709 of *Lecture Notes in Computer Science*, Newcastle upon Tyne, UK, June 2011. Springer.

[DFGvD06] Stéphane Demri, Alain Finkel, Valentin Goranko, and Govert van Drimmelen. Towards a model-checker for counter systems. In *4th ATVA*, pages 493–507. Springer Verlag LNCS 4218, 2006.

[DFS98] Catherine Dufourd, Alain Finkel, and Philippe Schnoebelen. Reset nets between decidability and undecidability. In *Proc. 25th Intl. Coll. Automata, Languages and Programming (ICALP'98)*, pages 103–115. Springer Verlag LNCS 1443, 1998.

[EFM99] Javier Esparza, Alain Finkel, and Richard Mayr. On the verification of broadcast protocols. In *14th LICS*, pages 352–359, 1999.

[EN98] E. Allen Emerson and Kedar S. Namjoshi. On model-checking for non-deterministic infinite-state systems. In *13th LICS*, pages 70–80, 1998.

[FG09] Alain Finkel and Jean Goubault-Larrecq. Forward analysis for WSTS, part I: Completions. In Susanne Albers and Jean-Yves Marion, editors, *Proceedings of the 26th Annual Symposium on Theoretical Aspects of Computer Science (STACS'09)*, volume 3 of *Leibniz International Proceedings in Informatics*, pages 433–444, Freiburg, Germany, February 2009. Leibniz-Zentrum für Informatik.

[FG12] Alain Finkel and Jean Goubault-Larrecq. Forward analysis for WSTS, part I: Completions. In preparation, 2012. Journal version of [FG09].

[Fin87] Alain Finkel. A generalization of the procedure of Karp and Miller to well structured transition systems. In *Proc. 13th Intl. Coll. Automata, Languages and Programming (ICALP'87)*, pages 499–508. Springer Verlag LNCS 267, 1987.

[Fin90] Alain Finkel. Reduction and covering of infinite reachability trees. *Information and Computation*, 89(2):144–179, 1990.

[Fin93] Alain Finkel. The minimal coverability graph for Petri nets. In *12th Intl. Conf. Advances in Petri Nets*, pages 210–243. Springer Verlag LNCS 674, 1993.

[FMP04] Alain Finkel, Pierre McKenzie, and Claudine Picaronny. A well-structured framework for analysing Petri net extensions. *Information and Computation*, 195(1-2):1–29, 2004.

[FS01] Alain Finkel and Philippe Schnoebelen. Well-structured transition systems everywhere! *Theoretical Computer Science*, 256(1–2):63–92, 2001.

[GHK⁺03] Gerhard Gierz, Karl Heinrich Hofmann, Klaus Keimel, Jimmie D. Lawson, Michael Mislove, and Dana S. Scott. Continuous lattices and domains. In *Encyclopedia of Mathematics and its Applications*, volume 93. Cambridge University Press, 2003.

[Gou07] Jean Goubault-Larrecq. On Noetherian spaces. In *Proceedings of the 22nd Annual IEEE Symposium on Logic in Computer Science (LICS'07)*, pages 453–462, Wrocław, Poland, July 2007. IEEE Computer Society Press.

[GRvB06a] Pierre Ganty, Jean-François Raskin, and Laurent van Begin. A complete abstract interpretation framework for coverability properties of WSTS. In E. Allen Emerson and Kedar S. Namjoshi, editors, *Proc. 7th International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI'06)*, pages 49–64. Springer Verlag LNCS 3855, 2006.

[GRvB06b] Gilles Geeraerts, Jean-François Raskin, and Laurent van Begin. Expand, enlarge and check: New algorithms for the coverability problem of WSTS. *J. Comp. and System Sciences*, 72(1):180–203, 2006.

[GRvB07] Gilles Geeraerts, Jean-François Raskin, and Laurent van Begin. On the efficient computation of the minimal coverability set for Petri nets. In *Proc. 5th Intl. Symp. Automated Technology for Verification and Analysis (ATVA'05)*, pages 98–113. Springer LNCS 4762, 2007.

[GS64] Seymour Ginsburg and Edwin H. Spanier. Bounded Algol-like languages. *Trans. American Mathematical Society*, 113(2):333–368, 1964.

[HP79] J. Hopcroft and J. J. Pansiot. On the reachability problem for 5-dimensional vector addition systems. *Theoretical Computer Science*, 8:135–159, 1979.

[Jan99] Petr Jančar. A note on well quasi-orderings for powersets. *Information Processing Letters*, 72(5–6):155–160, 1999.

[KM69] R. M. Karp and R. E. Miller. Parallel program schemata. *J. Comp. and System Sciences*, 3(2):147–195, 1969.

[Lav71] Richard Laver. On Fraïssé's order type conjecture. *Annals of Mathematics (2)*, 93:89–111, 1971.

[Mar94] Alberto Marcone. Foundations of BQO theory. *Trans. Amer. Math. Soc.*, 345(2):641–660, 1994.

[May03a]   Richard Mayr. Undecidable problems in unreliable computations. *Theor. Comput. Sci.*, 297(1-3):337–354, 2003.

[May03b]   Richard Mayr. Undecidable problems in unreliable computations. *Theoretical Computer Science*, 297(1-3):337–354, 2003.

[Mil85]   E. C. Milner. Basic WQO- and BQO-theory. In I. Rival, editor, *Graphs and Order. The Role of Graphs in the Theory of Ordered Sets and Its Applications*, pages 487–502. D. Reidel Publishing Co., 1985.

[MM81]   Ernst W. Mayr and Albert R. Meyer. The complexity of the finite containment problem for petri nets. *J. ACM*, 28(3):561–576, 1981.

[NW65]   Crispin Saint-John Alvah Nash-Williams. On well-quasi-ordering infinite trees. *Proc. Cambridge Philosophical Society*, 61:697–720, 1965.

[Rac78]   Charles Rackoff. The covering and boundedness problems for vector addition systems. *Theor. Comput. Sci.*, 6:223–231, 1978.

[Rad54]   Richard Rado. Partial well-ordering of sets of vectors. *Mathematika*, 1:89–95, 1954.

[RS04]   Neil Robertson and P.D. Seymour. Graph minors. XX. Wagner's conjecture. *Journal of Combinatorial Theory, Series B*, 92(2):325–357, 2004.

[Sch01]   Philippe Schnoebelen. Bisimulation and other undecidable equivalences for lossy channel systems. In Naoki Kobayashi and Benjamin C. Pierce, editors, *Proceedings of the 4th International Workshop on Theoretical Aspects of Computer Software (TACS'01)*, volume 2215 of *Lecture Notes in Computer Science*, pages 385–399, Sendai, Japan, October 2001. Springer.

[Val78]   Rüdiger Valk. Self-modidying nets, a natural extension of Petri nets. In *Proceedings of the 5th International Colloquium on Automata, Languages and Programming (ICALP'78)*, pages 464–476. Springer Verlag LNCS 62, 1978.

[VG05]   Kumar N. Verma and Jean Goubault-Larrecq. Karp-Miller trees for a branching extension of VASS. *Discrete Mathematics & Theoretical Computer Science*, 7(1):217–230, November 2005.

[WZH10]   Thomas Wies, Damien Zufferey, and Thomas A. Henzinger. Forward analysis of depth-bounded processes. In C.-H. Luke Ong, editor, *FOSSACS*, volume 6014 of *Lecture Notes in Computer Science*, pages 94–108. Springer, 2010.

[ZWH12]   Damien Zufferey, Thomas Wies, and Thomas A. Henzinger. Ideal abstractions for well-structured transition systems. In *VMCAI*, pages 445–460, 2012.