Forward Analysis for WSTS, Part II: Complete WSTS

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

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

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

1 Introduction

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

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

State of the art. Karp and Miller [27] proposed an algorithm, for Petri nets, which computes a finite representation of the *cover*, i.e., of the downward closure of the reachability set of a Petri net. Finkel [14,15] introduced the WSTS framework and generalized the Karp-Miller procedure to a class of WSTS. This was achieved by building a non-effective completion of the set of states, and replacing ω-accelerations of increasing sequences of states (in Petri nets) by least upper bounds (lub). In [12,15] a variant of this generalization of the Karp-Miller procedure was studied; but no guarantee was given that the cover could be represented finitely. There were no effective finite representations of downward closed sets in [15]. Finkel [16] modified the Karp-Miller algorithm to reduce the size of the intermediate computed trees. [22] recently proposed a weaker acceleration, which avoid some possible underapproximations in [16]. Emerson and Namjoshi [12] took into account the labeling of WSTS for adapting the generalised Karp-Miller algorithm to model-checking. They assume the existence of a compatible cpo, and proved that for broadcast protocols (which are equivalent to transfer Petri nets), the Karp-Miller procedure can be generalized. However, termination is then not guaranteed [13], and in fact neither is the existence of a finite representation of the cover. Abdulla, Colomb-Annichini, Bouajjani and Jonsson proposed a forward procedure for lossy channel systems [3] using downward closed regular languages as symbolic representations. Ganty, Geeraerts, Raskin and Van Begin [21,20] proposed a forward procedure for solving the coverability problem for WSTS equipped with an effective adequate domain of limits, or equipped with a finite set D used as a parameter to tune the precision of an abstract domain. Both solutions insure that every downward closed set has a finite representation. Abdulla *et al.* [3] applied this framework to Petri nets and lossy channel systems. Abdulla, Deneux, Mahata and Nylén proposed a symbolic framework for dealing with downward closed sets for Timed Petri nets [4].

Our contribution. First, we define *complete WSTS* as WSTS whose well-ordering is also a continuous dcpo. This allows us to design a conceptual procedure $\mathbf{Clover}_{\mathfrak{S}}$ that looks for a finite representation of the downward closure of the reachability set, i.e., of the cover [15]. We call such a finite representation a *clover* (for *clo*sure of *cover*). This clearly separates the fundamental ideas from the data structures used in implementing Karp-Miller-like algorithms. Our procedure also terminates in more cases than the well-known (generalized) Karp-Miller procedure [12,15]. We establish the main properties of clovers in Section 3 and use them to prove $\mathbf{Clover}_{\mathfrak{S}}$ 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 [8].

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

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

2 Preliminaries

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

3 Clovers of Complete WSTS

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

Generalizing the notion of continuity to partial maps, a *partial continuous* map $f : X \rightarrow X$, where $(X, \leq)$ is a dcpo, is such that $\operatorname{dom} f$ is open (not just upward-closed), and for every directed subset D *in* $\operatorname{dom} f$, $lub(f(D)) = f(lub(D))$. Equivalently, $\operatorname{dom} f$ is open and $f^{-1}(U)$ is open for any open U. The composite of two partial continuous maps is again partial continuous.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

The important fact, which assesses the importance of the clover, is the following (see Appendix A for a proof).

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

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

4 A Robust Class of WSTS: ω^2-WSTS

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

We show that, in some cases, $\widehat{X}$ is indeed *not* well-ordered. Take X to be Rado's structure X_{Rado} [29], i.e., $\{(m,n) \in \mathbb{N}^2 \mid m < n\}$, ordered by $\leq_{\text{Rado}}$: $(m,n) \leq_{\text{Rado}} (m',n')$ iff $m = m'$ and $n \leq n'$, or $n < m'$. It is well-known that $\leq_{\text{Rado}}$ is a well quasi-ordering, and that $\mathbb{P}(X_{\text{Rado}})$ is not well-quasi-ordered by $\leq^\sharp_{\text{Rado}}$, defined as $A \leq^\sharp_{\text{Rado}} B$ iff for every $y \in B$, there is a $x \in A$ such that $x \leq_{\text{Rado}} y$ [26]; see for example [5, Example 3.2] for a readable reference. One can show (see Appendix B) that $\widehat{X_{\text{Rado}}} = Idl(X_{\text{Rado}})$ is comprised of all elements of X_{Rado}, plus infinitely many elements ω_0, ω_1, $\ldots$, ω_i, $\ldots$, and ω, so that $(i,n) \leq \omega_i$ for all $n \geq i+1$, $\omega_i \leq \omega$ for all $i \in \mathbb{N}$, and $\{\omega_i \mid i \in \mathbb{N}\}$ is an antichain. We note that the latter is infinite. So:

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

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

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

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

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

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

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

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

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

5 A Conceptual Karp-Miller Procedure

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

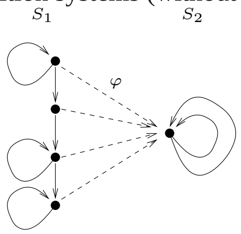

Fig. 2: Flattening

We obtain the following. The (non-trivial) proof appears in Appendix E.

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

6 Application: Well Structured Counter Systems

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

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

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

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

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

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

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

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

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

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

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

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

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

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

7 Conclusion and Perspectives

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

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

References

1. P. Abdulla, A. Bouajjani, and B. Jonsson. On-the-fly analysis of systems with unbounded, lossy Fifo channels. In *10th CAV*, pages 305–318. Springer Verlag LNCS 1427, 1998.
2. P. A. Abdulla, K. Čerāns, B. Jonsson, and Y.-K. Tsay. Algorithmic analysis of programs with well quasi-ordered domains. *Information and Computation*, 160(1–2):109–127, 2000.
3. P. A. Abdulla, A. Collomb-Annichini, A. Bouajjani, and B. Jonsson. Using forward reachability analysis for verification of lossy channel systems. *Formal Methods in System Design*, 25(1):39–65, 2004.
4. P. A. Abdulla, J. Deneux, P. Mahata, and A. Nylén. Forward reachability analysis of timed Petri nets. In *FORMATS/FTRTFT*, pages 343–362. Springer Verlag LNCS 3253, 2004.
5. P. A. Abdulla and A. Nylén. Better is better than well: On efficient verification of infinite-state systems. In *14th LICS*, pages 132–140, 2000.

6. S. Abramsky and A. 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.
7. A. Antonik. Presburger monotone affine functions can be lub-accelerated. Personal communication, 2009.
8. S. Bardin, A. Finkel, J. Leroux, and Ph. Schnoebelen. Flat acceleration in symbolic model checking. In *3rd ATVA*, pages 474–488. Springer Verlag LNCS 3707, 2005.
9. G. Cécé, A. Finkel, and S. Purushothaman Iyer. Unreliable channels are easier to verify than perfect channels. *Information and Computation*, 124(1):20–31, Jan. 1996.
10. S. Demri, A. Finkel, V. Goranko, and G. van Drimmelen. Towards a model-checker for counter systems. In *4th ATVA*, pages 493–507. Springer Verlag LNCS 4218, 2006.
11. C. Dufourd, A. Finkel, and Ph. Schnoebelen. Reset nets between decidability and undecidability. In *25th ICALP*, pages 103–115. Springer Verlag LNCS 1443, 1998.
12. E. A. Emerson and K. S. Namjoshi. On model-checking for non-deterministic infinite-state systems. In *13th LICS*, pages 70–80, 1998.
13. J. Esparza, A. Finkel, and R. Mayr. On the verification of broadcast protocols. In *14th LICS*, pages 352–359, 1999.
14. A. Finkel. A generalization of the procedure of Karp and Miller to well structured transition systems. In *13th ICALP*, pages 499–508. Springer Verlag LNCS 267, 1987.
15. A. Finkel. Reduction and covering of infinite reachability trees. *Information and Computation*, 89(2):144–179, 1990.
16. A. Finkel. The minimal coverability graph for Petri nets. In *12th Intl. Conf. Advances in Petri Nets*, pages 210–243. Springer Verlag LNCS 674, 1993.
17. A. Finkel and J. Goubault-Larrecq. Forward analysis for WSTS, part I: Completions. In *26th STACS*, Freiburg, Germany, 2009. Springer Verlag. To appear.
18. A. Finkel, P. McKenzie, and C. Picaronny. A well-structured framework for analysing Petri net extensions. *Information and Computation*, 195(1-2):1–29, 2004.
19. A. Finkel and Ph. Schnoebelen. Well-structured transition systems everywhere! *Theoretical Computer Science*, 256(1–2):63–92, 2001.
20. P. Ganty, J.-F. Raskin, and L. van Begin. A complete abstract interpretation framework for coverability properties of WSTS. In *7th VMCAI*, pages 49–64. Springer Verlag LNCS 3855, 2006.
21. G. Geeraerts, J.-F. Raskin, and L. van Begin. Expand, enlarge and check: New algorithms for the coverability problem of WSTS. *J. Comp. and System Sciences*, 72(1):180–203, 2006.
22. G. Geeraerts, J.-F. Raskin, and L. van Begin. On the efficient computation of the minimal coverability set for Petri nets. In *5th ATVA*, pages 98–113. Springer LNCS 4762, 2007.
23. G. Gierz, K. H. Hofmann, K. Keimel, J. D. Lawson, M. Mislove, and D. S. Scott. Continuous lattices and domains. In *Encyclopedia of Mathematics and its Applications*, volume 93. Cambridge University Press, 2003.
24. S. Ginsburg and E. H. Spanier. Bounded Algol-like languages. *Trans. American Mathematical Society*, 113(2):333–368, 1964.
25. J. Goubault-Larrecq. On Noetherian spaces. In *22nd LICS*, pages 453–462, Wrocław, Poland, July 2007. IEEE Computer Society Press.
26. P. Jančar. A note on well quasi-orderings for powersets. *Information Processing Letters*, 72(5–6):155–160, 1999.
27. R. M. Karp and R. E. Miller. Parallel program schemata. *J. Comp. and System Sciences*, 3(2):147–195, 1969.
28. R. Lazič, T. Newcomb, J. Ouaknine, A. W. Roscoe, and J. Worrell. Nets with tokens which carry data. *Fundamenta Informaticae*, 88(3):251–274, 2008.
29. R. Rado. Partial well-ordering of sets of vectors. *Mathematika*, 1:89–95, 1954.

30. R. Valk. Self-modidying nets, a natural extension of Petri nets. In *5th ICALP*, pages 464–476. Springer Verlag LNCS 62, 1978.

31. K. N. Verma and J. Goubault-Larrecq. Karp-Miller trees for a branching extension of VASS. *Discrete Mathematics & Theoretical Computer Science*, 7(1):217–230, Nov. 2005.

A Proof of Proposition 2

We first need a useful lemma.

Lemma 2. *For any downward closed subset F of $\widehat{X}$, $cl(F) \cap X = F \cap X$, where cl is closure in $\widehat{X}$.*

Proof. Since $\widehat{X} = Idl(X)$ is a continuous dcpo, $cl(F) = \mathrm{Lub}(F)$. Take any $x \in cl(F) \cap X$, then x is the least upper bound (in $\widehat{X}$) of a directed subset D of $F \cap X$. That is, $\downarrow x$ is the union of all $\downarrow y$, $y \in D$. Hence there is an $y \in D$ such that $x \in \downarrow y$. In particular, $x \leq y$. Since $y \in F$ and F is downward-closed, x is in F. So $x \in F \cap X$. We have proved $cl(F) \cap X \subseteq F \cap X$. The converse inclusion is obvious. □

We show that (Proposition 2): for every state $s_0 \in X$, $Cover_{\mathfrak{X}}(s_0) = Cover_{\mathfrak{S}}(s_0) \cap X = \downarrow Clover_{\mathfrak{S}}(s_0) \cap X$.

The first equality actually follows from Proposition 6.1 of [17], but a direct proof will be clearer. This will be a consequence of (1) and (2) below. The second equality is a consequence of Proposition 1 and Lemma 2.

First, we show that: (1) $Cover_{\mathfrak{S}}(s_0) \cap X \subseteq Cover_{\mathfrak{X}}(s_0)$. Let x be any element of $Cover_{\mathfrak{S}}(s_0) \cap X$. That is, up to the identification of elements $x \in X$ with $\downarrow x \in \widehat{X}$, $\downarrow x$ is in $Cover_{\mathfrak{S}}(s_0)$. By definition, there is a natural number k, $k+1$ elements $C_0 =\downarrow s_0$, $C_1, \ldots, C_k$ in $\widehat{X}$, and k partial monotonic maps $f_1, \ldots, f_k$ 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 $\downarrow x \subseteq C_k = \mathcal{S}f_k(C_{k-1}) = \downarrow f_k(C_{k-1})$, 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 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 $C_0 = \downarrow s_0$, 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 $x \in Cover^*_{\mathcal{X}}(s_0)$.

Second, we show: (2) $Cover_{\mathcal{X}}(s_0) \subseteq Cover_{\mathcal{S}}(s_0) \cap X$. Let $x \in Cover_{\mathcal{X}}(s_0)$. So there is a natural number $k \in \mathbb{N}$ and k maps f_1, …, f_k in F such that $x \leq f_k(f_{k-1}(\ldots(f_2(f_1(s_0)))))$, where the latter expression is defined. For all 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, $\mathcal{S}f_i(C_{i-1}) = \downarrow f_i(C_{i-1}) = \downarrow f_i(\downarrow f_{i-1}(\ldots(f_2(f_1(s_0)))))$. 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 $Cover_{\mathcal{S}}(\downarrow s_0)$. Equating $\downarrow x$ with $x \in X$ and $\downarrow s_0$ with s_0 as usual, x is in $Cover_{\mathcal{S}}(s_0) \cap X$.

Now the first equality $Cover_{\mathfrak{X}}(s_0) = Cover_{\mathfrak{S}}(s_0) \cap X$ is by (1) and (2). For the second equality, note that $\downarrow Clover_{\mathfrak{S}}(s_0) \cap X = cl(Cover_{\mathfrak{S}}(s_0)) \cap X$ by Proposition 1. By Lemma 2, this is just $Cover_{\mathfrak{S}}(s_0) \cap X$. □

B Proofs of Section 4

Proof of Lemma 1.

We exploit the fact that $\widehat{X_{\text{Rado}}} = Idl(X_{\text{Rado}})$, and examine the structure of directed subsets of X_{Rado}. In fact, we claim that the downward closed directed subsets of X_{Rado}, apart from those of the form $\downarrow(m, n)$, are of the form $\omega_i = \{(i, n) \mid n \geq i + 1\} \cup \{(m, n) \in X_{\text{Rado}} \mid n \leq i - 1\}$, or $\omega = X_{\text{Rado}}$. See Figure 3 for a pictorial representation of ω_i.

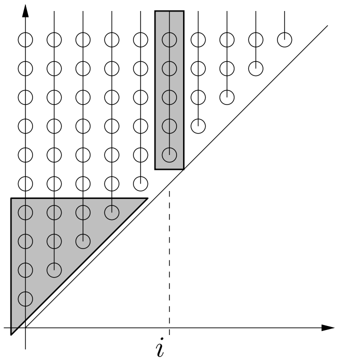

Fig. 3. Ideals in Rado's Structure

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 $D = X_{\text{Rado}}$. Indeed, for every $(m, n) \in X_{\text{Rado}}$, 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 $n \geq i + 1$, hence $(i, i + 1)$ also, since D is downward closed.

There cannot be any $(i', j') \in D$ with $i' < i$ and $j' \geq i$. Otherwise, since D is directed, there would be an $(i'', j'') \in D$ with $(i, i + 1), (i', j') \leq_{\text{Rado}} (i'', j'')$; the case $i'' = i$ is impossible, since then $(i', j') \leq_{\text{Rado}} (i'', j'')$ would imply $i' = i''$ and $j' \leq j''$ (impossible since $i' < i$), or $j' < i''$ (impossible since this would entail $i < i''$, contradicting the maximality of i in I); so, since $i'' \neq i$ and $(i, i + 1) \leq_{\text{Rado}} (i'', j'')$, $i'' > i + 1$, again 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 3, must be in D: these are the points (m, n) with $n < i$.

If the set of integers 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, then D contains all (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.

So $\widehat{X_{\text{Rado}}} = Idl(X_{\text{Rado}})$ is obtained by adjoining infinitely many elements ω_0, ω_1, $\ldots$, ω_i, $\ldots$, 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 $\mathcal{S}(X_{Rado\ a}) = Idl(X_{\text{Rado}})$ is not wqo. □

Proof of Proposition 3.

Jančar [26] shows that S is ω^2-wqo if and only if $\mathbb{P}(S)$ is well-ordered by $\leq^\sharp_{\text{Rado}}$. Recall that $B_1 \leq^\sharp_{\text{Rado}} B_2$ if and only if for every $y_2 \in B_2$, there is $y_1 \in B_1$ with $y_1 \leq_{\text{Rado}} y_2$. Note that $B_1 \leq^\sharp_{\text{Rado}} B_2$ if and only if $\uparrow B_1 \supseteq \uparrow B_2$.

We need to remember that the Alexandroff topology on a poset has all upward closed subsets as opens. 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 [25].

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 $\widehat{S} = \mathcal{S}(S_a)$ would contain $\widehat{Y} = Idl(Y)$. However by Lemma 1 $Idl(Y)$ contains an infinite antichain: contradiction. □

C Cover and Clover of Ordered Transition Systems

Lemma 3. *If $(\mathfrak{S}, s_0)$ be a monotone transition system, then $\downarrow Post^*_{\mathfrak{S}}(\downarrow s_0) = \downarrow Post^*_{\mathfrak{S}}(s_0)$.*

Proof. Let us show $\downarrow Post^*_{\mathfrak{S}}(\downarrow s_0) \subseteq \downarrow Post^*_{\mathfrak{S}}(s_0)$. Let $t_0 \leq s_0$ and $t_0 \xrightarrow{*} t_1$ such that $s \leq t_1$. Then by monotonicity of $\mathfrak{S}$, there is s_1 such that $s_0 \xrightarrow{*} s_1$ and $t_1 \leq s_1$. Hence $s \leq s_1$ and $s \in \downarrow Post^*_{\mathfrak{S}}(s_0)$. □

Proposition 8. *Let $\mathfrak{S}$ be a complete ordered transition system and A a finite set of states. Then $Clover_{\mathfrak{S}}(s_0) \sqsubseteq A$ if and only if $Post^*_{\mathfrak{S}}(s_0) \sqsubseteq A$.*

Proof. Let $\mathcal{F}$ be any closed subset of states. By Proposition 1, $\downarrow Clover_{\mathfrak{S}}(s_0) \subseteq \mathcal{F}$ iff $cl(Cover_{\mathfrak{S}}(s_0)) \subseteq \mathcal{F}$. Now $cl(Cover_{\mathfrak{S}}(s_0)) = cl(\downarrow Post^*_{\mathfrak{S}}(s_0)) = cl(Post^*_{\mathfrak{S}}(s_0))$. So $\downarrow Clover_{\mathfrak{S}}(s_0) \subseteq \mathcal{F}$ iff $cl(Post^*_{\mathfrak{S}}(s_0)) \subseteq \mathcal{F}$, iff $Post^*_{\mathfrak{S}}(s_0) \subseteq \mathcal{F}$, since $\mathcal{F}$ is closed. Now take $\mathcal{F} = \downarrow A$, which is closed because A is finite. □

D Proofs of Section 5

The following two propositions are used to prove the correctness (Theorem 2) of **Clover**$_{\mathfrak{S}}$. 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 1.

Lemma 4. *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 properties 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 $Post^k_{\mathfrak{S}}(cl(A)) \subseteq cl(Post^k_{\mathfrak{S}}(A))$, hence that $Post^*_{\mathfrak{S}}(cl(A)) \subseteq cl(Post^*_{\mathfrak{S}}(A))$. □

Proposition 9. *Let $\mathfrak{S}$ be an ∞-effective complete functional transition system and A_n be the value of the set A, computed by the procedure* **Clover**$_{\mathfrak{S}}$ *on input s_0, after n iterations of the while statement at line 2. Then A_n is finite, and $A_n \sqsubseteq A_{n+1} \sqsubseteq Clover_{\mathfrak{S}}(s_0)$, 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 $A_n \sqsubseteq A_{n+1}$.

We show that $A_n \sqsubseteq Clover_{\mathfrak{S}}(s_0)$, i.e., that $A_n \subseteq \downarrow Clover_{\mathfrak{S}}(s_0)$ by induction on n. By Proposition 1, it is equivalent to show that $A_n \subseteq cl(Cover_{\mathfrak{S}}(s_0))$.

If $n = 0, A_0 = \{s_0\}$, so $A_0 \sqsubseteq Cover_{\mathfrak{S}}(s_0) \sqsubseteq cl(Cover_{\mathfrak{S}}(s_0))$. Assume $A_n \subseteq cl(Cover_{\mathfrak{S}}(s_0))$, and let us prove that $A_{n+1} \sqsubseteq cl(Cover_{\mathfrak{S}}(s_0))$. 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 $A_n \sqsubseteq cl(Cover_{\mathfrak{S}}(s_0))$ 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 4, i.e., in $cl(Cover_{\mathfrak{S}}(s_0))$ by monotonicity.

If $a < g(a)$, then $g^{\infty}(a) = lub\{g^n(a) \mid n \in \mathbb{N}\}$ 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 $A_n \sqsubseteq cl(Cover_{\mathfrak{S}}(s_0))$ 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 4, i.e., in $cl(Cover_{\mathfrak{S}}(s_0))$ by monotonicity. □

If the procedure **Clover**$_{\mathfrak{S}}$ does not stop, it will compute an infinite sequence of sets of states. In other words, **Clover**$_{\mathfrak{S}}$ does not deadlock.

Proposition 10. *Let $\mathfrak{S}$ be an ∞-effective complete functional WSTS. The set $\bigcup_n A_n$ is finite if and only if the procedure* **Clover**$_{\mathfrak{S}}$ *stops on input s_0.*

Proof. Assume **Clover**$_{\mathfrak{S}}$ does not stop on input s_0. Since $A_n \sqsubseteq A_{n+1}$, if $A = \bigcup_n A_n$ is finite then 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 $g(a) \sqsubseteq A$, 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). Then $g^{\infty}(a) \sqsubseteq A_{n+1} = A$, so $g(a) \leq g^{\infty}(a) \sqsubseteq A_{n+1} = A$. It follows that $Post^*_{\mathcal{S}}(A) \sqsubseteq A$, so $Post_{\mathcal{S}}(A) \sqsubseteq A$, hence the procedure must stop after m turns of the loop: contradiction. So A is infinite. The converse implication is obvious. □

While **Clover**$_{\mathfrak{S}}$ is non-deterministic, this is *don't care non-determinism*: if one execution does not terminate, then no execution terminates. If **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 $A_n \sqsubseteq A_{n+1}$. We can now prove:
Theorem 2 : If **Clover**$_{\mathfrak{S}}(s_0)$ terminates, then it computes $Clover_{\mathfrak{S}}(s_0)$.

Proof. If the procedure **Clover**$_{\mathfrak{S}}$ terminates, then it returns a set $\operatorname{Max} A$ such that $Post_{\mathfrak{S}}(A) \sqsubseteq A$, i.e., $\downarrow Post_{\mathfrak{S}}(A) \subseteq \downarrow A$. By Proposition 9, A contains all A_n, hence also s_0. So $\downarrow A$ contains $Cover_{\mathfrak{S}}(s_0)$. Since A is finite, $\downarrow A$ is closed, so $\downarrow A$ must also contain $cl(Cover_{\mathfrak{S}}(s_0))$. By Proposition 1, it follows that $\downarrow A$ must contain $\downarrow Clover_{\mathfrak{S}}(s_0)$, i.e., $Clover_{\mathfrak{S}}(s_0) \sqsubseteq A$. However, by Proposition 9 again, $A \sqsubseteq Clover_{\mathfrak{S}}(s_0)$. So $\operatorname{Max} A = Clover_{\mathfrak{S}}(s_0)$. □

We needed the above theorem so as to establish:
Proposition 6 : There is an ∞-effective complete WSTS such that the termination of **Clover**$_{\mathfrak{S}}$ is undecidable.

Proof. Assume we can decide whether **Clover**$_{\mathfrak{S}}$ terminates. If **Clover**$_{\mathfrak{S}}$ does not terminate on s_0, then the reachability set is infinite. If **Clover**$_{\mathfrak{S}}$ terminates, then it computes the clover by Theorem 2, and we can decide boundedness as in the proof of Proposition 4, in the case of functional-lossy channel systems. A similar argument works with reset Petri nets, where boundedness is also undecidable [11]. □

We may also characterize the cover.

Proposition 11. *Let $\mathfrak{S}$ be an ∞-effective complete WSTS, obtained as the completion of an ω^2-WSTS $\mathfrak{X}$. If the procedure* **Clover**$_{\mathfrak{S}}$ *stops on input s_0 and returns A, then $Cover_{\mathfrak{X}}(s_0) = X \cap \downarrow A$.*

Proof. By Proposition 2 and Theorem 2. □

Proposition 5 : The procedures **Clover**$_{\mathfrak{S}}$ terminates more often than the Generalized Karp-Miller Tree procedure.

Proof. It is clear that if the Generalized Karp-Miller Tree (GKMT) procedure terminates then **Clover**$_{\mathfrak{S}}$, too. As a matter of fact, if the GKMT procedure terminates, it has found the clover (the minimal coverability set) by using finitely many lub-accelerations; these lub-accelerations will be inevitably found by **Clover**$_{\mathfrak{S}}$ and from the fixpoint test at line 2, **Clover**$_{\mathfrak{S}}$ will stop also.

Now, let us consider the reset Petri net of [11, Example 3]. This has 4 places, 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 $n_1 \geq 1$, $t_2(n_1, n_2, n_3, n_4) = (n_1 - 1, 0, n_3 + 1, n_4)$, $t_3(n_1, n_2, n_3, n_4) = (n_1, n_2 + 1, n_3, n_4 - 1)$ if $n_3 \geq 1$, and $t_4(n_1, n_2, n_3, n_4) = (n_1 + 1, n_2 + 1, n_3 - 1, 0)$. 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 $n_2 \geq 1$. 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 Petri net with initial marking $(1, 1, 0, 0)$ because there is a non-regular infinite path: $(1, 1, 0, 0) \overset{t_1 t_2 t_3 t_4}{\longrightarrow} (1, 2, 0, 0) \overset{t_1^2 t_2 t_3^2 t_4}{\longrightarrow} (1, 3, 0, 0) \ldots \overset{t_1^i t_2 t_3^i t_4}{\longrightarrow} (1, i + 1, 0, 0) \ldots$ going towards infinity (on the second component). The GKMT procedure cannot compact it, because the least upper bound of no repeated sequence along this path contains any ω. Hence, the GKMT is infinite because the second component unbounded along this infinite non-regular path. Now it is clear that the procedure **Clover**$_{\mathfrak{S}}$ will compute (by firing transition t_5) the maximal marking $(\omega, \omega, \omega, \omega)$, which is the sole element of the clover; hence **Clover**$_{\mathfrak{S}}$ stops. □

E Proof of Theorem 3

We require the following auxiliary lemma:

Lemma 5. *Let S be well-ordered by $\leq$, and $(s_n)_{n\in\mathbb{N}}$ a sequence of elements in S. Then either $s_k < s_{k'}$ for some $k < k'$, or $(s_n)_{n\in\mathbb{N}}$ only contains finitely many elements.*

Proof. By definition, there are $k < k'$ such that $s_k \leq s_{k'}$ $(*)$. Assume the inequality $(*)$ cannot be made strict. Let $A = \{s_n \mid n \in \mathbb{N}\}$. We note that every strictly increasing chain in A is finite. Indeed, if $s_n < s_{n_1}$ then $n > n_1$ because otherwise $(*)$ could be made strict with $k = n$, $k' = n_1$. (Note that $n = n_1$ is impossible.)

So build a tree as follows. There is a distinguished root, whose sons are decorated with the maximal elements of A. Since they are all incomparable and $\leq$ is well, there are only finitely many sons of the root. Also, because every strictly increasing chain in A is finite, every element of A is below some element decorating a root's son.

For each constructed leaf decorated with $a \in A$, add as many sons to it as there are maximal elements b in A with $b < a$, each decorated with some distinct b. Repeat until no new node can be added.

Since $\leq$ is well-founded, all the branches of the final tree are finite. This is a finitely-branching tree, so by König's Lemma, the tree is finite. However, every element of A ends up decorating some node in the tree. So A is finite. □

We first show:

Proposition 12. *Let $\mathfrak{S}$ be an ∞-effective complete WSTS. Assume $(\mathfrak{S}, s_0)$ clover-flattable. Then* **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 $\varphi(s_1) = s_0$ and $Cover_{\mathfrak{S}}(s_0) = \downarrow \varphi(Cover_{\mathfrak{S}_1}(s_1))$. 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 $w_1^* w_2^* \dots w_m^*$, for some fixed sequence $w_1, w_2, \dots, w_m \in F_1^*$.

Let again A_n be the value of the set A, computed by the procedure **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 shall show that there is a natural number n such that $Clover_{\mathfrak{S}}(s_0) \sqsubseteq A_n$, under the apparently more general assumption that $\varphi(s_1) \leq s_0$ and $Clover_{\mathfrak{S}}(s_0) \sqsubseteq \varphi(Clover_{\mathfrak{S}_1}(s_1))$.

Consider any open subset U of S that intersects $\downarrow Clover_{\mathfrak{S}}(s_0)$. Then U must also intersect $\downarrow \varphi(Clover_{\mathfrak{S}_1}(s_1))$. Since $\varphi(Clover_{\mathfrak{S}_1}(s_1))$ is finite, its downward closure is closed, hence equals its closure. By elementary topology, the open U then intersects $\downarrow \varphi(Clover_{\mathfrak{S}_1}(s_1))$, so U intersects $\varphi(Clover_{\mathfrak{S}_1}(s_1))$, so $\varphi^{-1}(U)$ intersects

$Clover_{\mathfrak{S}_1}(s_1))$, so $\varphi^{-1}(U)$ intersects $\downarrow Clover_{\mathfrak{S}_1}(s_1))$ (because $\varphi^{-1}(U)$ is upward closed, since U is and φ is monotonic). By Proposition 1, $\varphi^{-1}(U)$ intersects $cl(Cover_{\mathfrak{S}_1}(s_1))$. Since φ is continuous, $\varphi^{-1}(U)$ is open, so by elementary topology again, $\varphi^{-1}(U)$ must intersect $Cover_{\mathfrak{S}_1}(s_1)$. So U intersects $\varphi(Cover_{\mathfrak{S}_1}(s_1))$, say at a. In particular, there is an $a_1 \in S_1$ such that $a \leq \varphi(a_1)$, and $a_1 \leq w_1^{k_1} w_2^{k_2} \dots w_m^{k_m}(s_1)$, for some natural numbers k_1, k_2, …, k_m. We shall show that a is in A_n for some n that is *independent* of U, a, a_1, k_1, k_2, …, k_m. That is, for any U, a, and a_1, we get the same n.

Extend the action of $\varphi : F_1 \to F$ on words by $\varphi(f_1 f_2 \dots f_k) = \varphi(f_1)\varphi(f_2)\dots\varphi(f_k)$. The basic idea is to let the procedure **Clover**$_{\mathfrak{S}}$ lub-accelerate $\varphi(w_m)$ to simulate the computation of $w_m^{k_m}(s_1)$. But something that easy won't work.

So we require a technical detour. Consider the sequence $(\varphi(w_m)^k(s_0))_{k\in\mathbb{N}}$ in S. Since S is wqo, there are two indices $k < k'$ such that $\varphi(w_m)^k(s_0) \leq \varphi(w_m)^{k'}(s_0)$. Choose k minimal such that for some $k' > k$, the strict inequality $\varphi(w_m)^k(s_0) < \varphi(w_m)^{k'}(s_0)$ holds, if possible (call this case I). Otherwise, the sequence $(\varphi(w_m)^k(s_0))_{k\in\mathbb{N}}$ only contains finitely many distinct elements by Lemma 5. In this case, let k be some number such that s_0, $\varphi(w_m)(s_0)$, $\varphi(w_m)^2(s_0)$, …, $\varphi(w_m)^k(s_0)$ contains every element of the sequence (call this case II).

By application of fairness, the procedure **Clover**$_{\mathfrak{S}}$ will eventually select the pair $(\varphi(w_m), s_0)$; if $k \geq 1$, $s_0 \not\leq \varphi(w_m)(s_0)$ by the minimality of k, so $\varphi(w_m)^\infty(s_0) = \varphi(w_m)(s_0)$. Similarly, the pair $(\varphi(w_m), \varphi(w_m)(s_0))$ will eventually be selected, then $(\varphi(w_m), \varphi(w_m)^2(s_0))$, then … then $(\varphi(w_m), \varphi(w_m)^{k-1}(s_0))$, yielding $\varphi(w_m)^k(s_0)$.

In case II, $\varphi(w_m)^{k_m}(s_0)$ is already among the elements $s_0, \varphi(w_m)(s_0), \varphi(w_m)^2(s_0)$, …, $\varphi(w_m)^k(s_0)$. In case II, another application of fairness shows that the procedure **Clover**$_{\mathfrak{S}}$ will eventually select the pair $(g, a') = (\varphi(w_m)^{k'-k}, \varphi(w_m)^k(s_0))$. Then $a' < g(a')$ by the definition of k, so we add $g^\infty(a')$ to the current value of A, and clearly $\varphi(w_m)^{k_m}(s_0) \leq g^\infty(a')$.

In any case, this whole sequence of k (in case II), resp. $k+1$ (in case I) lub-accelerations eventually forces some element s_0^1 above $\varphi(w_m)^{k_m}(s_0)$ into A_{n_1}, where n_1 only depends on k, which in turns depends on s_0 and w_m, but certainly not on k_m, or U, a, or a_1.

Now repeat the argument, replacing s_0 by s_0^1 and s_1 by $s_1^1 = w_m^{k_m}(s_1)$. Note that $a \leq \varphi(a_1)$, and $a_1 \leq w_1^{k_1} w_2^{k_2} \dots w_{m-1}^{k_{m-1}}(s_1^1)$, while $\varphi(s_1^1) \leq s_0^1$. The latter inequality comes from the fact that $\varphi(s_1^1) = \varphi(w_m)^{k_m}(\varphi(s_1)) \leq \varphi(w_m)^{k_m}(s_0)$. Then there is another step $n_2 \geq n_1$ such that A_{n_2} contains some element s_0^2 above $\varphi(w_{m-1})^{k_{m-1}}(s_0^1) = \varphi(w_{m-1}^{k_{m-1}} w_m^{k_m})(s_0)$, and n_2 again depends only on s_0, w_m and w_{m-1}. Iterating, we eventually obtain $n_m \geq n_{m-1} \geq \ldots \geq n_1$ such that A_{n_m} contains some element s_0^m above $\varphi(w_1^{k_1} w_2^{k_2} \dots w_m^{k_m})(s_0)$, and n_m depends only on s_0, w_m, …, w_2, w_1.

Since $a \leq \varphi(a_1) \leq \varphi(w_1^{k_1} w_2^{k_2} \dots w_m^{k_m})(\varphi(s_1)) \leq \varphi(w_1^{k_1} w_2^{k_2} \dots w_m^{k_m})(s_0) \leq s_0^m$, a is in $\downarrow A_{n_m}$. Since n_m is independent of a and U, we may now define U to be the complement of $\downarrow A_{n_m}$ (the latter is closed). If U intersected $\downarrow Clover_{\mathfrak{S}}(s_0) = cl(Cover_{\mathfrak{S}}(s_0))$, then we would find $a \in U \cap \varphi(Cover_{\mathfrak{S}_1}(s_1))$, and a would be in $\downarrow A_{n_m}$; this is what we have just proved above. But a cannot be both in U and in $\downarrow A_{n_m}$, whose intersection is empty. So $U \cap cl(Cover_{\mathfrak{S}}(s_0)) = \emptyset$. So $cl(Cover_{\mathfrak{S}}(s_0)) \subseteq \downarrow A_{n_m}$.

By Proposition 9, the converse inclusion holds. We conclude that the procedure **Clover**$_{\mathfrak{S}}$ stops after the n_mth turn of the loop. □

Proposition 13. *Let $\mathfrak{S}$ be an ∞-effective complete WSTS. If* **Clover**$_{\mathfrak{S}}$ *terminates on s_0, then $(\mathfrak{S}, s_0)$ is clover-flattable. Moreover, we can even require that the continuous flattening has the same clover up to φ, i.e., $Clover_{\mathfrak{S}}(s_0) = \mathrm{Max}\,\varphi(Clover_{\mathfrak{S}_1}(s_1))$.*

Proof. Write $\mathfrak{S}$ as $(S, \xrightarrow{F}, \leq)$. Assume that **Clover**$_{\mathfrak{S}}$ terminates on s_0. Then it returns some finite set A such that $A = Clover_{\mathfrak{S}}(s_0)$ by Theorem 2 Each element a of A is obtained as $g_1^{a\infty} g_2^{a\infty} \dots {g_{n_a}^{a}}^{\infty}(s_0)$, where each g_i^a is in F^*. Enumerate the elements $a_1, \dots, a_k$ of A, and create a fresh control state q_{ij} for every i, j with $1 \leq i \leq k$, $1 \leq j \leq n_a$, plus another one written ϵ. We enumerate $\epsilon < q_{1n_{a_1}} < \dots < q_{12} < q_{11} < q_{2n_{a_2}} < \dots < q_{22} < q_{21} < \dots < q_{k2} < q_{k1}$ (the allowed order of application of each $g_j^{a_i}$). Let $\mathfrak{S}_1$ be the WSTS whose set of states is $Q \times S$, where Q is the set of control states q_{ij}, and let $f_j^{a_i}(q, s)$ be defined iff $q \leq q_{ij}$ and $s \in \mathrm{dom}\, g_j^{a_i}$, as $(q_{ij}, g_j^{a_i})$. The ordering on S_1 is defined as $(q, s) \leq (q', s')$ iff $q = q'$ and $s \leq s'$. The map φ sends (q, s) to s, and is clearly continuous.

By repeated application of Lemma 4, and since each element a of A is obtained as $g_1^{a\infty} g_2^{a\infty} \dots {g_{n_a}^{a}}^{\infty}(s_0)$, (q, a) is also in $cl(Cover_{\mathfrak{S}_1}(s_0))$ for some control state q. So $\varphi(cl(Cover_{\mathfrak{S}_1}(s_0)))$ contains A. By Proposition 1, $A \subseteq \varphi(\downarrow Clover_{\mathfrak{S}_1}(s_0)) = \downarrow \varphi(Clover_{\mathfrak{S}_1}(s_0))$, hence $A = Clover_{\mathfrak{S}}(s_0) \sqsubseteq \varphi(Clover_{\mathfrak{S}_1}(s_0))$. The converse inequality holds by construction, so $Clover_{\mathfrak{S}}(s_0) = \mathrm{Max}\,\varphi(Clover_{\mathfrak{S}_1}(s_1))$. □