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 \(\mathcal{\infty}\)-complete WSTS \(S\). This computes the so-called clover of a state. When \(S\) is the completion of a WSTS \(X\), the clover in \(S\) is a finite description of the downward closure of the reachability set. We show that such completions are \(\mathcal{\infty}\)-complete exactly when \(X\) is an \(\omega^2\)-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 \(s \geq s_1 \rightarrow^* t_1 \geq t\) 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 \(\downarrow \text{Post}^*(\downarrow s)\) of \(s\). While the cover allows one to decide coverability as well, by testing whether \(t \in \downarrow \text{Post}^*(\downarrow s)\), 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 \(s \geq s_1 \rightarrow^* t\). \((U\text{-boundedness generalizes the boundedness problem, which is the instance of } U\text{-boundedness where } U \text{ is the entire set of states}). No backward algorithm can decide this. In fact, \(U\text{-boundedness is undecidable in general, e.g., on lossy channel systems [CFP99]. 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 [ACJT00] 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 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 CloverS 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 closure 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 5 and use them to prove CloverS correct, notably, in Section 5.
Second, we characterize complete WSTS for which Clover_\mathcal{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 5.20), 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 \( \omega^2 \)-WSTS, i.e., those whose state space is \( \omega^2 \)-wqo (a wqo is a well quasi-ordering), as we show in Section 4. All naturally occurring WSTS are in fact \( \omega^2 \)-WSTS. We shall also explain why this study is important: despite the fact that Clover_\mathcal{S} cannot terminate on all inputs, that \( \mathcal{S} \) is an \( \omega^2 \)-WSTS will ensure progress, i.e., that every opportunity of accelerating a loop will eventually be taken by Clover_\mathcal{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 \( \infty \)-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 [ACJT00, 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 \( \{ y \in X \mid \exists x \in E : x \leq y \} \). The downward closure \( \downarrow E \) is \( \{ y \in X \mid \exists x \in E : y \leq x \} \). 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 \( \text{lub}(E) \). An element \( x \) of \( E \) is maximal (resp. minimal) iff \( \uparrow x \cap E = \{ x \} \) (resp. \( \downarrow x \cap E = \{ x \} \)). Write Max \( E \) (resp. 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 $\text{Lub}(E) = \{\text{lub}(D) \mid D \text{ directed subset of } E\}$. Clearly, $E \subseteq \text{Lub}(E)$; $\text{Lub}(E)$ 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 $\text{lub}(D) \leq y$, there is a $z \in D$ such that $x \leq z$. Note that $x \ll y$ implies $x \leq y$, and that $x' \leq x \ll y \leq y'$ 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 $\leq$ is a well partial ordering that also turns $X$ into a dcpo, we say that $X$ is a directed complete well order, or dcpo. We shall be particularly interested in continuous dcpos.

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 $\text{lub}(D) \in U$, 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 $X$, $f(\text{lub}(D)) = f(\text{lub}(D))$. Equivalently, $f$ is continuous in the topological sense, i.e., $f^{-1}(U)$ is open for every open $U$.

A weaker requirement is $\omega$-continuity: $f$ is $\omega$-continuous iff $\text{lub}\{f(x_n) \mid n \in \mathbb{N}\} = f(\text{lub}\{x_n \mid n \in \mathbb{N}\})$, 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 $\text{Lub}(B) = B$. An inductive subset of $X$ is none other than a sub-dcpo of $X$.

The closure $\text{cl}(A)$ of $A \subseteq X$ is the smallest closed set containing $A$. This should not be confused with the inductive closure $\text{Ind}(A)$ of $A$, which is obtained as the smallest inductive subset $B$ containing $A$. In general, $\downarrow A \subseteq \text{Lub}(\downarrow A) \subseteq \text{Ind}(\downarrow A) \subseteq \text{cl}(A)$, 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 $\omega$ added, ordered by $(n_1, n_2, \ldots, n_k) \leq (n_1', n_2', \ldots, n_k')$ iff $(n_1, n_2, \ldots, n_k) = (n_1', n_2', \ldots, n_k')$, or for some $i, 1 \leq i \leq k$, $n_1 = n_1' = n_2' = \ldots = n_{i-1}' = n_{i-1} = \omega$, $n_i \neq \omega$, and either $n'_i = \omega$ or $n_i < n_i' \in \mathbb{N}$. Then take $A = \mathbb{N}_\omega^k \subseteq X$; $\downarrow A = A$, but $\text{Lub}(\downarrow A) = \mathbb{N}_\omega^k \times \mathbb{N}_{k-2}^k$ is even larger, $\ldots$, $\text{Lub}^{i-1}(\downarrow A) = \text{Lub}(\text{Lub}^{i-1}(\downarrow A))$ equals $\mathbb{N}_\omega^i \times \mathbb{N}_{k-i}^k$ for all $i, 2 \leq i \leq k$, and this is a strictly increasing chain of subsets. All of them are contained in $\text{Ind}(\downarrow A) = \mathbb{N}_\omega^k$, which coincides with $\text{cl}(A)$ here. It may also be the case that $\text{Ind}(\downarrow A)$ is strictly contained in $\text{cl}(A)$: consider the set $X$ of all pairs $(i, m)$ with $i \in \{0, 1\}$, $m \in \mathbb{N}$, plus a new element $\omega$, ordered by $(i, m) \leq (j, n)$ iff $i = j$ and $m = n$, and $(i, m) \leq \omega$ for all $(i, m) \in X$, and let $A = \{(0, m) \mid m \in \mathbb{N}\}$; then $\text{Ind}(\downarrow A) = A \cup \{\omega\}$, but the latter is not even downward-closed, so is strictly smaller than $\text{cl}(A)$; in fact $\text{cl}(A)$ 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, $\text{Lub}(A) = \text{Ind}(A) = \text{cl}(A)$. This is well-known, see e.g., [FG00, Proposition 3.5], and will play an important role in our constructions. As a matter in fact, the fact that $\text{Lub}(A) = \text{cl}(A)$, 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 \( \mathcal{G} = (S, \rightarrow) \) 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 \( \text{Post}_\mathcal{G}(s) = \{s' \in S \mid s \rightarrow s'\} \) for the set of immediate successors of the state \( s \). The reachability set of a transition system \( \mathcal{G} = (S, \rightarrow) \) from an initial state \( s_0 \) is \( \text{Post}_\mathcal{G}^*(s_0) = \{s \in S \mid s_0 \overset{*}{\rightarrow} s\} \).

We shall be interested in effective transition systems. Intuitively, a transition system \( (S, \rightarrow) \) is effective iff one can compute the set of successors \( \text{Post}_\mathcal{G}(s) \) of any state \( s \). We shall take this to imply that \( \text{Post}_\mathcal{G}(s) \) is finite, and each of its elements is computable, although one could imagine that \( \text{Post}_\mathcal{G}(s) \) 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 \( (S, \rightarrow, r, \text{post}) \), where \( (S, \rightarrow) \) is a transition system, \( r : E \rightarrow S \) is a representation map, and \( \text{post} : E \rightarrow \mathbb{P}_{\text{fin}}(E) \) is a computable map such that, for every code \( e \), \( r(\text{post}(e)) = \text{Post}_\mathcal{G}(r(e)) \). We write \( r(A) \) 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 \( \text{post} : \mathbb{N} \rightarrow \mathbb{P}_{\text{fin}}(\mathbb{N}) \) that is defined on all elements of \( E \), and such that \( \text{post}(e) \in \mathbb{P}_{\text{fin}}(E) \) 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 \( \text{post} \) function implicit.

An ordered transition system is a triple \( \mathcal{G} = (S, \rightarrow, \leq) \) 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 \( (S, \rightarrow, \leq, r, \text{post}, \preceq) \) where \( (S, \rightarrow, \leq) \) is an ordered transition system, \( (S, \rightarrow, r, \text{post}) \) 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 \( \mathcal{G} = (S, \rightarrow, \leq) \) 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 \( s_1 \rightarrow s'_1 \) and \( s'_1 \geq s' \) (resp. \( s'_1 > s' \)). \( \mathcal{G} \) 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 \( \text{Post}^*_\mathcal{G}(s) \), 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 \( \text{Cover}_\mathcal{G}(s) = \{s \mid \text{Post}_\mathcal{G}(s) \subseteq \mathbb{P}_{\text{fin}}(E) \} \) when \( \mathcal{G} \) is monotonic will be much better behaved. Note that being able to compute the cover allows one to decide coverability: \( s \overset{\geq}{\rightarrow^*} \) \( t \) iff \( t \in \text{Cover}_\mathcal{G}(s) \). 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 \( \text{Post}^*_\mathcal{G}(s) \) finite?) and \( U \)-boundedness (is \( \text{Post}^*_\mathcal{G}(s) \cap U \) finite?) will be decidable, too.

An ordered transition system \( \mathcal{G} = (S, \rightarrow, \leq) \) is a Well Structured Transition System (WSTS) iff \( \mathcal{G} \) 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) \([\text{Fin90}]\). 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
procedure (Section 5), and already the technique of accelerating loops (Definition 3.3) de-

dpends on the considered transition system being functional.

Formally, a functional transition system $(S, \rightarrow_F)$ is a labeled transition system where the
transition relation $\rightarrow_F$ is defined by a finite set $F$ of partial functions $f : S \rightarrow S$, in the
sense that for every $s, s' \in S$, $s \rightarrow_F s'$ iff $s' = f(s)$ for some $f \in F$. If additionally, a partial
ordering $\leq$ is given, a map $f : S \rightarrow S$ is partial monotonic iff $\text{dom}(f)$ is upward-closed and
for all $x, y \in \text{dom}(f)$ with $x \leq y$, $f(x) \leq f(y)$. An ordered functional transition system is an
ordered transition system $\mathcal{G} = (S, \rightarrow_F, \leq)$ 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 $(S, \rightarrow_F)$ is effective if every $f \in F$ is computable: given
a state $s$ and a function $f$, we can decide whether $s \in \text{dom}(f)$ 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 $\text{send}_a$ to add a fixed letter $a$ to
the back of each queue (i.e., $\text{dom}(\text{send}_a) = \Sigma^*$, where $\Sigma$ is the queue alphabet, and
$\text{send}_a(w) = wa$), and functions $\text{recv}_a$ 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., $\text{dom}(\text{recv}_a) = \{waw' \mid w, w' \in \Sigma^*\}$,
$\text{recv}_a(waw') = w'$ 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 $\mathbb{N}_\omega = \mathbb{N} + \text{a new top element } \omega$, with the usual componentwise ordering.
We have defined general completions of $\text{wpos}$, 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:
Definition 3.1. A partial continuous map $f : X \to X$, where $(X, \leq)$ is a dcpo, is a partial map whose domain $\text{dom} f$ is open (not just upward-closed), and such that for every directed subset $D$ in $\text{dom} f$, $\text{lub}(f(D)) = f(\text{lub}(D))$.

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 $\text{dom} f$) 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 $\mathcal{S} = (S, F \to, \leq)$ where $(S, \leq)$ is a continuous dcpo 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: $\text{dom} f^\infty = \text{dom} f$, and for any $x \in \text{dom} f$, if $x < f(x)$ then $f^\infty(x) = \text{lub}\{f^n(x) \mid n \in \mathbb{N}\}$, else $f^\infty(x) = f(x)$.

Note that if $x \leq f(x)$, then $f(x) \in \text{dom} f$, 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 \cite{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 $\Sigma$, and consider subsets $S$ of $\Sigma^*$ specified by a Turing machine $\mathcal{M}$ with tape alphabet $\Sigma$, so that $S$ is the language accepted by $\mathcal{M}$. Let $\leq$ be, say, the prefix ordering on $\Sigma^*$. 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 $\omega^2$-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 closure of the cover).

Definition 3.4 (Clover). Let $\mathcal{S} = (S, F \to, \leq)$ be a complete WSTS. The clover $\text{Clover}_\mathcal{S}(s_0)$ of the state $s_0 \in S$ is $\text{Max Lub}(\text{Cover}_\mathcal{S}(s_0))$. 

Figure 1: The clover and the cover, in a complete space
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 $\text{Cover}\mathcal{S}(s_0)$ is a downward-closed subset, illustrated in blue (grey if you read this in black and white). $\text{Lub}(\text{Cover}\mathcal{S}(s_0))$ 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 $\text{Lub}(\text{Cover}\mathcal{S}(s_0))$, 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 $S$, $\text{Max} F$ is finite and $\text{Max} F = \bigcap \text{Max} F$.

**Proof.** As $F$ is closed, it is inductive (i.e., $\text{Lub}(F) = F$). 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, $\text{Max} F = \bigcap \text{Max} F$. Now every two elements of $\text{Max} F$ are incomparable, i.e., $\text{Max} F$ is an antichain: since $S$ is wpo, $\text{Max} F$ is finite. □

**Remark 3.6.** Lemma 3.5 generalizes to Noetherian spaces, which extend wpos [Gou07]; every closed subset $F$ of a sober Noetherian space $S$ is of the form $\bigcap \text{Max} F$, with $\text{Max} F$ finite [Gou07, Corollary 6.5]. Wpos are sober, and every continuous dcpo is sober in its Scott topology [A.J93, Proposition 7.2.27].

**Proposition 3.7.** Let $\mathcal{S} = (S, \rightarrow, \leq)$ be a complete WSTS, and $s_0 \in S$. Then $\text{Clover}\mathcal{S}(s_0)$ is finite, and $\text{cl}(\text{Cover}\mathcal{S}(s_0)) = \bigcap \text{Clover}\mathcal{S}(s_0)$.

**Proof.** $\text{Lub}(\text{Cover}\mathcal{S}(s_0)) = \text{cl}(\text{Cover}\mathcal{S}(s_0))$ since $\text{Cover}\mathcal{S}(s_0)$ is downward-closed, and $S$ is a continuous dcpo. Now use Lemma 3.5 on the closed set $\text{Lub}(\text{Cover}\mathcal{S}(s_0))$. □

For any other representative, i.e., for any finite set $R$ such that $\bigcap R = \bigcap \text{Clover}\mathcal{S}(s_0)$, $\text{Clover}\mathcal{S}(s_0) = \text{Max} R$. Indeed, for any two finite sets $A, B \subseteq S$ such that $\bigcap A = \bigcap B$, $\text{Max} A = \text{Max} B$. So $\text{Clover}$ is the minimal representative of the cover, i.e., there is no representative $R$ with $|R| < |\text{Clover}\mathcal{S}(s_0)|$. 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 $\hat{X}$ of a wpo $(X, \leq)$ is defined in any of two equivalent ways. First, $\hat{X}$ is the ideal completion $\text{Idl}(X)$ 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. $\hat{X}$ can also be described as the sobrification $S(X_a)$ of the Noetherian space $X_a$, but this is probably harder to understand.

There is an embedding $\eta_X : X \rightarrow \hat{X}$, i.e., an injective map such that $x \leq x'$ in $X$ iff $\eta_X(x) \leq \eta_X(x')$ in $\hat{X}$. This is defined by $\eta_X(x) = \downarrow x$. This allows us to consider $X$ as a subset of $\hat{X}$, by equating $X$ with its image $\eta_X(X)$, i.e., by equating each element $x \in X$ with $\downarrow x \in \hat{X}$. 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 \( X \sqcup L \), so that any downward-closed subset \( D \) of \( X \) can be written as \( X \sqcup \downarrow A \), where \( A \) is a finite subset of \( X \sqcup L \). 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].

\[ \widehat{X} = \text{Idl}(X) \] is always a continuous dcpo. In fact, it is even algebraic [AJS94, 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, \( \widehat{\mathbb{N}} = \mathbb{N}_* \), \( \widehat{A} = A \) for any finite poset, and \( \prod_{i=1}^k X_i = \prod_{i=1}^k \widehat{X}_i \). Also, \( \widehat{X}^\times \) 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 \( A^* \) (\( A \in \mathbb{P}_{\text{fin}}(\widehat{X}) \), \( A \neq \emptyset \)) or \( a^\uparrow \) (\( a \in \widehat{X} \)). 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 \( \widehat{\mathcal{G}} = \widehat{\mathcal{X}} \) of a (functional) WSTS \( \mathcal{X} = (X, F, \preceq) \) as \((\widehat{X}, \widehat{S}F, \preceq)\), where \( \widehat{S}F = \{ \widehat{S}f \mid f \in F \} \) [FG09, Section 6]. For each partial monotonic map \( f \in F \), the partial continuous map \( \widehat{S}f : \widehat{X} \to \widehat{X} \) is such that \( \text{dom} \widehat{S}f = \{ C \in \widehat{X} \mid C \cap \text{dom} f \neq \emptyset \} \), and \( \widehat{S}f(C) = \downarrow f(C) \) for every \( C \in \widehat{X} \). In the cases of Petri nets or functional-lossy channel systems, the completed WSTS is effective [FG09, Section 6].

The important fact, which assesses the importance of the clover, is Proposition below. We first require a useful lemma. Up to the identification of \( X \) with its image \( \eta_X(X) \), this states that for any downward-closed subset \( F \) of \( \widehat{X} \), \( \text{cl}(F) \cap X = F \cap X \), 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 \( \widehat{X} \), \( \eta_X^{-1}(\text{cl}(F)) = \eta_X^{-1}(F) \).

**Proof.** We show that \( \eta_X^{-1}(\text{cl}(F)) \subseteq \eta_X^{-1}(F) \); the converse inclusion is obvious. Since \( \widehat{X} = \text{Idl}(X) \) is a continuous dcpo, \( \text{cl}(F) = \text{Lub}(F) \). Take any \( x \in \eta_X^{-1}(\text{cl}(F)) \): 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., \( x \in \eta_X^{-1}(F) \). \( \square \)

Up to the identification of \( X \) with \( \eta_X(X) \), the next proposition states that \( \text{Cover}_X(s_0) = \text{Cover}_{\mathcal{G}}(s_0) \cap X = \downarrow \text{Clover}_{\mathcal{G}}(s_0) \cap X \). In other words, to compute the cover of \( s_0 \) in the WSTS \( \mathcal{X} \) on the state space \( \widehat{X} \), one can equivalently compute the cover \( s_0 \) in the completed WSTS \( \widehat{X} \), 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 \( \widehat{X} \), in the form of the downward closure \( \downarrow \text{Clover}_{\mathcal{G}}(s_0) \) of its clover. The closure of the cover will include
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 $\mathcal{S} = \hat{X}$ be the completion of the functional WSTS $X = (X, \mathcal{F}, \leq)$. For every state $s_0 \in X$, $\text{Cover}_X(s_0) = \eta_X^{-1}(\text{Clover}_\mathcal{S}(\eta_X(s_0))) = \eta_X^{-1}(\{\text{Clover}_\mathcal{S}(\eta_X(s_0))\}).$

**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: \(1\) \(\eta_X^{-1}(\text{Clover}_\mathcal{S}(\eta_X(s_0))) \subseteq \text{Cover}_X(s_0)\). Let \(x\) be any element of \(\eta_X^{-1}(\text{Clover}_\mathcal{S}(\eta_X(s_0)))\), i.e., \(\downarrow x\) is in \(\text{Cover}_\mathcal{S}(\eta_X(s_0))\). By definition, there is a natural number \(k\), and \(k+1\) elements \(C_0 = \eta_X(s_0), C_1, \ldots, C_k\) in \(\hat{X}\), and \(k\) partial monotonic maps \(f_1, \ldots, f_k\) in \(F\) such that \(\downarrow x \subseteq C_k\), and \(C_i = \mathcal{F}f_i(C_{i-1})\) for every \(i, 1 \leq i \leq k\).

Since \(\downarrow x \subseteq C_k = \mathcal{F}f_k(C_{k-1}) = \downarrow f_k(C_{k-1})\), there is an element \(x_{k-1} \in C_{k-1} \cap \text{dom} f_k\) such that \(x \leq f_k(x_{k-1})\). Similarly, there is an \(x_{k-2} \in C_{k-2} \cap \text{dom} f_{k-1}\) such that \(x_{k-1} \leq f_{k-1}(x_{k-2})\), \ldots, an \(x_1 \in C_1 \cap \text{dom} f_2\) such that \(x_2 \leq f_2(x_1)\), and an \(x_0 \in C_0 \cap \text{dom} f_1\) 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 \(f_1, \ldots, f_k\) are partial monotonic, \(x \leq f_k(f_{k-1}(\ldots(f_2(f_1(s_0)))))), so \(x \in \text{Cover}_X(s_0)\).

Conversely, we show: \(2\) \(\text{Cover}_X(s_0) \subseteq \eta_X^{-1}(\text{Clover}_\mathcal{S}(\eta_X(s_0)))\). Let \(x \in \text{Cover}_X(s_0)\). So there is a natural number \(k \in \mathbb{N}\), and there are \(k\) maps \(f_1, \ldots, f_k\) 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 \(\downarrow f_i(\ldots(f_2(f_1(s_0))))\).

We claim that whenever \(i \geq 1\), \(C_i = \mathcal{F}f_i(C_{i-1})\). Indeed, \(\mathcal{F}f_i(C_{i-1}) = \downarrow f_i(C_{i-1}) = \downarrow f_i(\downarrow f_{i-1}(\ldots(f_2(f_1(s_0))))\). Since \(f_i\) is partial monotonic, \(\downarrow f_i(\downarrow y) = \downarrow f_i(y)\) for every \(y\). So \(\mathcal{F}f_i(C_{i-1}) = C_i\). Next, \(C_0 = \downarrow s_0\); and \(\downarrow x \subseteq C_k\), since \(x \subseteq C_k\) and \(C_k\) is downward-closed. So \(\downarrow x\) is in \(\text{Cover}_\mathcal{S}(\downarrow s_0)\), i.e., \(\eta_X(x)\) is in \(\text{Cover}_\mathcal{S}(\eta_X(s_0))\). \(\square\)

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

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 \(\downarrow \text{Clover}_\mathcal{S}(s_0)\) 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 \(\downarrow \text{Clover}_\mathcal{S}(s_0)\) as being those states that are "reachable
in infinitely many steps" from $s_0$. And we hope to find the finitely many elements of $\text{Clover}_G(s_0)$ by doing enough lub-accelerations.

4. A Robust Class of WSTS: $\omega^2$-WSTS

It would seem clear that the construction of the completion $\mathcal{G} = \hat{X}$ of a WSTS $\mathcal{X} = (X, F_\mathcal{X}, \leq)$ be, again, a WSTS. We shall show that this is not the case. The only missing ingredient to show that $\mathcal{G}$ is a complete WSTS is to check that $\hat{X}$ is well-ordered by inclusion. We have indeed seen that $\hat{X}$ is a continuous dcpo; and $\mathcal{G}$ 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 $\mathcal{X}$ is $\mathcal{G} = \hat{X}$ again a WSTS? Equivalently, when is $\hat{X}$ well-ordered by inclusion? We shall see that there is a definite answer: when $X$ is $\omega^2$-wpo.

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, $\text{Clover}_G$, 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 $\hat{X}$, 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 $s \in \hat{X}$, say, and add new children to $N$. For each $f \in F$ such that $s \in \text{dom} Sf$, let $s' = Sf(s)$, 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 $(f_p \circ \ldots \circ f_1)^\omega(s'')$. (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 $(f_p \circ \ldots \circ f_1)^\omega(s'').$) 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 $\hat{X} = \mathbb{N}^k$, one cannot lub-accelerate more than $k$ times, because each lub-acceleration introduces a new $\omega$ 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 $\omega$ components do disappear.

The second reason is of more general applicability: $\hat{X} = \mathbb{N}^k$ is wpo, and this implies that along every infinite branch of the tree thus constructed, case (*) will eventually happen,
and in fact will happen infinitely many times. Call this \textit{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 \(\hat{X}\) is wpo (or even just wqo). \textit{This} is our main motivation in characterizing those wpos \(X\) such that \(\hat{X}\) is wpo again.

Before we proceed, let us explain why termination cannot be ensured. Generally, this\footnote{\label{termin}Clover S.} will follow from undecidability arguments (e.g., Proposition \ref{undec} below). Here is a concrete case of non-termination. Consider the reset Petri net of \cite[Example 3]{DFS98}, see Figure 3.

This net has 4 places and 4 transitions, hence defines a 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 \(n_1 \geq 1, n_2 \geq 1\), \(t_3(n_1, n_2, n_3, n_4) = (n_1, n_2 + 1, n_3, n_4 - 1)\) 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 \(t_3(t_2(t_3^2(1, n_2, 0, 0)))) = (1, n_2 + 1, 0, 0)\) 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 subsequence \(t_1t_2t_3t_4\), 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 \(t_1^2t_2t_3t_4\), 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 \(t_1^it_2t_3t_4\) 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: \textit{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 \((t_4 \circ t_3 \circ t_2 \circ t_1)^\infty(1, i, 0, 0)\), but none of these lub-accelerations actually serve any purpose, since \((t_4 \circ t_3 \circ t_2 \circ t_1)^\infty(1, i, 0, 0) = (1, i + 1, 0, 0)\) is already equal to \((t_4 \circ t_3 \circ t_2 \circ t_1)(1, i, 0, 0)\).

Progress will take a slightly different form in the actual procedure Clover S. of Section \ref{clover}. 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 \(\hat{X}\), and essentially acts as a distraction. However, progress will be crucial (Proposition \ref{fair} states that if the set of values computed by the
procedure \texttt{Clover}$_S$ is finite then \texttt{Clover}$_S$ terminates) in our characterization of the cases where \texttt{Clover}$_S$ terminates (Theorem \ref{thm:clover-termination}), as those states that are clover-flattable (see Section \ref{sec:clover-structure}). Without it, \texttt{Clover}$_S$ would terminate in strictly less cases.

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

Take \(\mathcal{X}\) to be Rado’s structure \(\mathcal{X}_{\text{Rado}}\) \cite{Rad54}, 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 \(\mathcal{P}(\mathcal{X}_{\text{Rado}})\) is not well-quasi-ordered by \(\leq_{\hat{\text{Rado}}}\), defined as \(A \leq_{\hat{\text{Rado}}} B\) iff for every \(y \in B\), there is a \(x \in A\) such that \(x \leq_{\text{Rado}} y\) \cite{Jan99}. (Equivalently, \(A \leq_{\hat{\text{Rado}}} B\) iff \(\uparrow B \subseteq \uparrow A\).)

Consider indeed \(\omega_i = \{(i,n) \mid n \geq i + 1\} \cup \{(m,n) \in \mathcal{X}_{\text{Rado}} \mid n \leq i - 1\}\), for each \(i \in \mathbb{N}\).

This is pictured as the dark blue (or dark grey) region in Figure \ref{fig:ideals}. It arises naturally in Lemma \ref{lem:downward-closed} below. Note that \(\omega_i\) is downward-closed in \(\leq_{\text{Rado}}\). Consider the complement \(\overline{\omega_i}\) of \(\omega_i\), and note that \(\overline{\omega}_i \leq_{\hat{\text{Rado}}} \overline{\omega}_j\) 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 \(\omega_i\) but not in \(\omega_j\), so \(\overline{\omega}_i \not\leq_{\hat{\text{Rado}}} \overline{\omega}_j\). So \((\overline{\omega}_i)_{i \in \mathbb{N}}\) is an infinite sequence of \(\mathcal{P}(\mathcal{X}_{\text{Rado}})\) from which one cannot extract any infinite ascending chain. Hence \(\mathcal{P}(\mathcal{X}_{\text{Rado}})\) is indeed not wqo.

Let us characterize \(\overline{\mathcal{X}_{\text{Rado}}}\). To this end, we exploit the fact that \(\overline{\mathcal{X}_{\text{Rado}}} = \text{Idl}(\mathcal{X}_{\text{Rado}})\), and examine the structure of directed subsets of \(\mathcal{X}_{\text{Rado}}\).

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

Proof. Take any downward-closed directed subset \(D\) of \(\mathcal{X}_{\text{Rado}}\). Consider the set \(I\) of all integers \(i\) such that some \((i,n)\) is in \(D\). If \(I\) is not bounded, then \(D = \mathcal{X}_{\text{Rado}}\). 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 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 \(\omega_i\), as shown in Figure 4, must be entirely outside \(D\). Otherwise, since \(D\) is directed, there would be an \((i'', j'') \in D\) with \((i, i + 1), (i', j') \leq_{\text{Rado}} (i'', j'')\); 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 then \(i \leq j' < i'' = i\)); since \(i'' \neq i\) and \((i, i + 1) \leq_{\text{Rado}} (i'', j''), i'' > i + 1\), contradicting the maximality of \(i\) in \(I\).

On the other hand, since \((i, i + 1)\) is in \(D\), then the lower triangle of \(\omega_i\), 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 \(\Perp(i, n_{\max})\) in \(X_{\text{Rado}}\). Otherwise, \(D\) contains every \((i, n)\) with \(n \geq i + 1\), and therefore \(D\) contains \(\omega_i\). It cannot contain more, so \(D = \omega_i\).

Then one checks that \(\omega_i\) is indeed directed and downward-closed.

So \(\widehat{X}_{\text{Rado}} = \text{Idl}(X_{\text{Rado}})\) is obtained by adjoining infinitely many elements \(\omega_0, \omega_1, \ldots, \omega_i, \ldots\), and \(\omega\) to \(X_{\text{Rado}}\). 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 \(\widehat{X}_{\text{Rado}} = \text{Idl}(X_{\text{Rado}})\) is not wqo:

**Proposition 4.2.** \(\widehat{X}_{\text{Rado}}\) contains an infinite chain, and is therefore not well-ordered by inclusion. \(\square\)

### 4.3. \(\omega^2\)-WSTS

Recall here the working definition in [Jan99]: a well-quasi-order \(X\) is \(\omega^2\)-wqo if and only if it does not contain an (isomorphic copy of) \(X_{\text{Rado}}\); here we use Jančar’s definition, as it is more tractable than the complex definition of [Mar94]. Jančar proved that \(X\) is \(\omega^2\)-wqo iff \((\mathbb{P}(X), \leq^2)\) 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 \(\omega^2\)-wqo.

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

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 \(\omega^2\)-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 $\omega^2$-wqo. Then we recall that $\mathcal{S} = S(S_a)$ is the subspace of $\mathcal{H}(S)$ consisting of all irreducible closed subsets $\text{[Gou07]}
$.

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

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

Let an $\omega^2$-WSTS be any WSTS whose underlying poset is $\omega^2$-wqo. It follows:

**Theorem 4.4.** Let $\mathcal{S} = (S, \rightarrow, \leq)$ be a functional WSTS. Then $\mathcal{S}$ is a (complete, functional) WSTS iff $\mathcal{S}$ is an $\omega^2$-WSTS.

### 4.4. Are $\omega^2$-wqos Ubiquitous? $X_{\text{Rado}}$ is an example of a wqo that is not $\omega^2$-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 $\omega^2$-wpo.

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

$$
\begin{align*}
D & ::= \mathbb{N} & \text{natural numbers} \\
& | A \leq & \text{finite set } A, \text{ ordered by } \leq \\
& | D_1 \times \ldots \times D_k & \text{finite product} \\
& | D_1 + \ldots + D_k & \text{finite, disjoint sum} \\
& | D^* & \text{finite words} \\
& | D^\circ & \text{finite multisets} \\
& | T(D) & \text{finite trees}
\end{align*}
$$

$\mathbb{N}$ 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 $x_1 \leq_i y_1$ 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 $a_1a_2\ldots a_n$, then $w'$ is of the form $w_0a_1'w_1a_2'\ldots a_i'w_n$, for some words $w_0$, $w_1$, $\ldots$, $w_n$, and some letters $a_i'$, $1 \leq i \leq n$, such that $a_i \leq a_i'$

$D^\circ$ is the set of finite multisets $\{x_1, \ldots, x_n\}$ of elements of $D$. Write again $\leq$ the ordering on $D$. Then $D^\circ$ is ordered by $\leq^\circ$ defined as: $\{x_1, x_2, \ldots, x_m\} \leq^\circ \{y_1, y_2, \ldots, y_n\}$ 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 $\leq^\circ$ is not the usual multiset extension $\leq^\text{mul}$ of $\leq$. However, for one, this is the $\leq^m$ quasi-ordering considered, on finite sets $X$, by Abdulla et al. $\text{[ADMN04a] Section 2}$ for example. Then, it turns out that $m \leq^\circ m'$ entails $m \leq^\text{mul} m'$. In particular, the fact that $\leq^\circ$ is well, whenever $\leq$ is, entails that $\leq^\text{mul}$ is well: given any sequence of multisets
(m_i)_{i \in \mathbb{N}}$, one can extract an infinite ascending chain with respect to \( \leq^\circ \), hence also with respect to \( \leq_{mul} \). Similarly, when \( (D^\circ, \leq^\circ) \) is an \( \omega^2\)-wqo, then so is \( (D^\circ, \leq_{mul}) \), using the fact that \( X \) is \( \omega^2\)-wqo iff both \( X \) and \( \mathbb{P}(X) \) are wqo (the latter, equipped with \( \leq^\circ \)).

Finally, \( T(D) \) 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 \( \vec{t} \in D^\star \), the pair \( (f, \vec{t}) \) is in \( X \). When \( \vec{t} \) is the word consisting of the terms \( t_1t_2\ldots t_m \), we usually write \( (f, \vec{t}) \) as the term \( f(t_1, t_2, \ldots, t_m) \). Given an ordering \( \leq \) on \( D \), the embedding ordering \( \leq_{emb} \) on \( T(D) \) is defined by induction on the sum of the sizes of the terms to compare by:
\[
t = f(t_1, t_2, \ldots, t_m) \leq_{emb} g(u_1, u_2, \ldots, u_n) \text{ iff } t \leq_{emb} u_j \text{ for some } j, 1 \leq j \leq n, \text{ or } f \leq g \text{ and } t_1t_2\ldots t_m \leq_{emb} u_1u_2\ldots u_n.
\]

We will prove that every datatype defined in (4.1) is not only \( \omega^2\)-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 \( \mathbb{P}^\omega(X) \) is wqo, where \( \omega_1 \) is the first uncountable ordinal, \( \mathbb{P}^\alpha(X) \) is defined for every ordinal \( \alpha \) by \( \mathbb{P}^\alpha(X) = X, \mathbb{P}^{\alpha+1}(X) = \mathbb{P}(\mathbb{P}^\alpha(X)), \mathbb{P}^\alpha(X) = \bigcup_{\beta<\alpha} \mathbb{P}^\beta(X) \) for every limit ordinal \( \alpha \), and where powersets are quasi-ordered by \( \leq^\circ \). Abdulla and Nylén give a gentle introduction to the theory of bqos [AN00]. Then:

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

**Proof.** Every bqo is \( \omega^2\)-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, 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 \( D^\star \) is bqo whenever \( D \) is (this is also mentioned in [AN00] Theorem 3.1 (3)). When \( D \) is bqo, \( D^\circ \) is proved to be a bqo in [AN00] Theorem 3.1 (4)]. Finally, \( D \) is bqo implies that \( T(D) \) is bqo by [Lav71] Theorem 2.2; 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 \( \omega \) whose nodes are labeled with elements of \( Q \).

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 \( \hat{\mathcal{S}} \) of a WSTS \( \mathcal{S} \) is effective iff the completion \( \hat{S} \) of the set of states is effective and \( \hat{S}f \) is recursive for all \( f \in F \). \( \hat{S} \) is effective for all the data types of [FG09] Section 51. Also, \( \hat{S}f \) 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), \( \hat{S}f \) coincides with the extension \( \vec{f} \) defined in [FMP04] Section 2: whenever \( \text{dom } f \) is upward-closed and \( f : \mathbb{N}^k \rightarrow \mathbb{N}^k \) is defined by \( f(s) = A\overrightarrow{s} + \overrightarrow{a} \), for some matrix \( A \in \mathbb{N}^{k \times k} \) and vector \( \overrightarrow{a} \in \mathbb{Z}^k \), then dom \( \hat{S}f = \uparrow_S \text{dom } f \),

\[\text{That is, of Section 4.4 of this paper, see (4.1), to the exception of the finite tree constructor. We have a proof that } \hat{S} \text{ is in fact effective for all the data types of (4.1) [FG12], but this is not published yet.}\]
and \( S(f) \) is again defined as \( A \bar{s} + \bar{a} \), this time for all \( \bar{s} \in \mathbb{N}^k \), and using the convention that \( 0 \times \omega = 0 \) when computing the matrix product \( A \bar{s} \) \cite[Theorem 7.9]{FMP04}.

In the case of functional-lossy channel systems, it is easy to see that \( \text{dom} S(\text{send}_a) = \hat{S} \), \( S(\text{send}_a)(P) = Pa^\omega \) for every word-product \( P \); and that \( \text{dom} S(\text{recv}_a) = \uparrow S \), and:

\[
\begin{align*}
S(\text{recv}_a)(a^\omega P) &= P \\
S(\text{recv}_a)(b^\omega P) &= S(\text{recv}_a)(P) \quad (b \neq a) \\
S(\text{recv}_a)(A^* P) &= A^* P \quad \text{if } a \in A \\
S(\text{recv}_a)(A^* P) &= S(\text{recv}_a)(P) \quad \text{otherwise}
\end{align*}
\]

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 \cite[Lemma 6]{ABJ98}, 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 \cite{ADMN04a}, which are elements of \( (A^\omega)^* \) with \( A \) finite.

As promised, we can now show:

**Proposition 4.6.** There are effective complete WSTS \( \mathcal{G} \) such that the map \( \text{Clover}_{\mathcal{G}} : S \to \mathbb{P}_{\text{fin}}(S) \) is not recursive.

**Proof.** Let \( \mathcal{G} \) be the completion of a functional-lossy channel system \cite[Section 6]{FG09} on the message alphabet \( \Sigma \). By Theorem 4.4, \( \mathcal{G} \) is a complete WSTS. It is effective, too, see above or \cite[Lemma 6]{ABJ98}. \( \text{Clover}_{\mathcal{G}}(s_0) \) 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 \( A^* \) (\( A \in \mathbb{P}_{\text{fin}}(\Sigma) \), \( A \neq \emptyset \)) or \( a^\omega \) (\( a \in \Sigma \)). Now \( \text{Post}^\omega_{\mathcal{G}}(s_0) \) is finite iff none of these atomic expressions is of the form \( A^* \). So, if we could compute \( \text{Clover}_{\mathcal{G}}(s_0) \), 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 \cite{CFP96}. We could have played the same argument with reset Petri nets \cite{DFS98} instead as well. \( \square \)

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 \( \pi \)-calculus, the simple algorithm that works backward (computing the set of predecessors of an upward-closed initial set) of \cite{ACJT00} 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 \cite{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 \cite{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 \cite{WZH10}. Recently, Zufferey, Wies and Henzinger proposed to compute a part of the clover by using a particular widening, called a set-widening operator \cite{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 \cite{Rac78}, or the backward procedure...
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 $Post^\infty_S(s)$ finite?) and $U$-boundedness (is $Post^\infty_S(s) \cap U$ 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 $s \geq [\rightarrow^*; \geq] t$ iff $t \in \text{Cover}_S(s)$ iff $\exists t' \in \text{Clover}_S(s)$ 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 procedure below is being able to compute lub-accelerations. Hence:

**Definition 5.1** ($\infty$-Effective). An effective complete functional WSTS $\mathcal{S} = (S, F, \leq)$ is $\infty$-effective iff every function $g^\infty$ 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 $\infty$-effective: not only is $N^k_\omega$ 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 $i$th component of $\vec{x}$; it follows that $g^\infty(\vec{x})$ is the tuple whose $i$th component is $\vec{x}_i$ if $\delta_i = 0$, $\omega$ otherwise.

Let $\mathcal{S}$ be an $\infty$-effective WSTS, and write $A \leq^b B$ 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$_S(s_0)$**:
1. $A \leftarrow \{s_0\}$;
2. **while** $Post^b_S(A) \not\leq^b A$ **do**
   (a) Choose fairly (see below) $(g, a) \in F^* \times A$ such that $a \in \text{dom } g$;
   (b) $A \leftarrow A \cup \{g^\infty(a)\}$;
3. **return** Max $A$;

Note that Clover$_S$ 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 some later stage 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 \texttt{Clover} 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 \texttt{Clover} 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 \(\mathbb{N}^k\), as we have seen in Section 4.1. 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 \texttt{Clover} 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 \texttt{Clover}(\(s_0\)), is the clover, it is enough to decide whether \(\text{Post}(A) \leq^* A\). 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 \(\text{Post}(A) \leq^* A\) 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 \texttt{Clover} 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 \texttt{Clover} terminates then the computed set \(A\) is contained in \(\text{Lub}(\text{Post}^*_F(s_0))\). It is crucial that \(\text{Lub}(F) = \text{cl}(F)\) 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 \(\Theta = (S, F, \leq)\) be a complete (functional) WSTS. For any subset \(A\) of states, \(\text{Post}_\Theta(\text{cl}(A)) \subseteq \text{cl}(\text{Post}_\Theta(A))\).

Proof. We first observe that \(\text{Post}_\Theta(\text{cl}(A)) \subseteq \text{cl}(\text{Post}_\Theta(A))\). Indeed, for any \(s \in \text{Post}_\Theta(\text{cl}(A))\), there is an \(f \in F\) and some \(t \in \text{dom} f \cap \text{cl}(A)\) such that \(f(t) = s\). Let \(U\) be the complement of \(\text{cl}(\text{Post}_\Theta(A))\): \(U\) 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 \(\text{cl}(A)\). It is a general property of
topological spaces that an open (here $f^{-1}(U)$) meets $\text{cl}(A)$ iff it meets $A$. So there is also a state $t'$ in $f^{-1}(U) \cap A$. That is, $t' \in \text{dom } f$, $f(t') \in U$ and $t' \in A$. But $t' \in A$ implies $f(t') \in \text{Post}_\mathcal{S}(A) \subseteq \text{cl}(\text{Post}_\mathcal{S}(A))$, contradicting the fact that $f(t') \in U$. So $s$ cannot be in $U$, i.e., $s \in \text{cl}(\text{Post}_\mathcal{S}(A))$.

By an easy induction on $k \in \mathbb{N}$, it follows that $\text{Post}_\mathcal{S}^k(\text{cl}(A)) \subseteq \text{cl}(\text{Post}_\mathcal{S}^k(A))$, hence that $\text{Post}_\mathcal{S}^k(\text{cl}(A)) \subseteq \text{cl}(\text{Post}_\mathcal{S}^k(A))$.

**Proposition 5.3.** Let $\mathcal{S}$ be an \(\infty\)-effective complete functional transition system and $A_n$ be the value of the set $A$, computed by the procedure $\text{Clover}_\mathcal{S}$ on input $s_0$, after $n$ iterations of the while statement at line 2. Then $A_n$ is finite, and $A_n \leq^\flat A_{n+1} \leq^\flat \text{Clover}_\mathcal{S}(s_0)$, 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 $A_n \leq^\flat \text{Clover}_\mathcal{S}(s_0)$, i.e., that $A_n \subseteq \downarrow \text{Clover}_\mathcal{S}(s_0)$, by induction on $n$. By Proposition 3.7, it is equivalent to show that $A_n \subseteq \text{cl}(\text{Cover}_\mathcal{S}(s_0))$.

If $n = 0$, $A_0 = \{s_0\}$, so $A_0 \subseteq \text{Cover}_\mathcal{S}(s_0) \subseteq \text{cl}(\text{Cover}_\mathcal{S}(s_0))$.

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

If $a \leq g(a)$, then $g^\infty(a) = g(a)$ is in $\text{Post}_\mathcal{S}^k(a)$, and since $a \in A_n$ and $A_n \subseteq \text{cl}(\text{Cover}_\mathcal{S}(s_0))$ by induction hypothesis, $g(a)$ is in $\text{Post}_\mathcal{S}^k(\text{cl}(\text{Cover}_\mathcal{S}(s_0)))$. The latter is contained in $\text{cl}(\text{Post}_\mathcal{S}^k(\text{Cover}_\mathcal{S}(s_0)))$ by Lemma 5.2 i.e., in $\text{cl}(\text{Cover}_\mathcal{S}(s_0))$ by monotonicity.

If $a < g(a)$, then $g^\infty(a) = \text{lub}(g^n(a) \mid n \in \mathbb{N})$ is a least upper bound of a directed chain of elements in $\text{Post}_\mathcal{S}^k(a)$. So $g^\infty(a) \in \text{Lub}(\text{Post}_\mathcal{S}^k(a)) \subseteq \text{cl}(\text{Post}_\mathcal{S}^k(a))$. Since $a \in A_n$ and $A_n \subseteq \text{cl}(\text{Cover}_\mathcal{S}(s_0))$ by induction hypothesis, $g^\infty(a)$ is in $\text{cl}(\text{Post}_\mathcal{S}^k(\text{Cover}_\mathcal{S}(s_0)))$. The latter is contained in $\text{cl}(\text{cl}(\text{Post}_\mathcal{S}^k(\text{Cover}_\mathcal{S}(s_0)))) = \text{cl}(\text{Post}_\mathcal{S}^k(\text{Cover}_\mathcal{S}(s_0)))$ by Lemma 5.2 i.e., in $\text{cl}(\text{Cover}_\mathcal{S}(s_0))$ by monotonicity.

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

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

**Proof.** Assume $\text{Clover}_\mathcal{S}$ does not stop on input $s_0$, but $A = \bigcup A_n$ 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 $\text{Post}_\mathcal{S}^k(A) \leq^\flat A$, so $\text{Post}_\mathcal{S}^k(A) \leq A$, hence the procedure must stop after $m$ turns of the loop: contradiction. The converse implication is obvious.

While $\text{Clover}_\mathcal{S}$ is non-deterministic, this is don't care non-determinism: if one execution does not terminate, then no execution terminates. If $\text{Clover}_\mathcal{S}$ 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$_\mathcal{E}(s_0)$ terminates, then it computes Clover$_\mathcal{E}(s_0)$.

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

However, by Proposition 5.3 $\{s_0\} = A_0 \leq^\flat A_1 \leq^\flat \cdots \leq^\flat A_n \leq^\flat \cdots \leq^\flat A$, so $s_0 \in \downarrow A$. So Clover$_\mathcal{E}(s_0) \subseteq \downarrow \text{Max } A$.

Since $A$ is finite, Max $A$ is too, so $\downarrow \text{Max } A$ is closed. Any closed set containing another set must contain its closure. So $\downarrow \text{Max } A$ must also contain $cl(\text{Clover}_\mathcal{E}(s_0))$. By Proposition 3.7, $\downarrow \text{Max } A$ must therefore contain $\downarrow \text{Clover}_\mathcal{E}(s_0)$. In other words, Clover$_\mathcal{E}(s_0) \leq^\flat \text{Max } A$. However, using Proposition 5.3 again, Max $A \leq^\flat A \leq^\flat \text{Clover}_\mathcal{E}(s_0)$. So Max $A = \text{Clover}_\mathcal{E}(s_0)$.

If the generalized Karp-Miller tree procedure (see Section 4.1) terminates then it has found a finite set $g_1, g_2, \ldots, g_n$ of maps to lub-accelerate. These lub-accelerations will also be found by Clover$_\mathcal{E}$, by fairness. From the fixpoint test, Clover$_\mathcal{E}$ will also stop. So Clover$_\mathcal{E}$ terminates on at least all inputs where the generalized Karp-Miller tree procedure terminates. We can say more:

Proposition 5.6. The procedure Clover$_\mathcal{E}$ 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$_\mathcal{E}$ 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.

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

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

Proof. Since Post$_\mathcal{E}^*(s_0)$ is finite, $g^\omega(s)$ is in Post$_\mathcal{E}^*(s_0)$ for every $s \in \text{Post}_\mathcal{E}^*(s_0)$ and every $g \in F^*$ with $s \in \text{dom } g$. So, defining again $A_n$ as the value of the set $A$ computed by Clover$_\mathcal{E}$ 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$_\mathcal{E}^*(s_0)$, hence finite. By Proposition 5.4 Clover$_\mathcal{E}(s_0)$ terminates.

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

Proof. Assume we can decide whether Clover$_\mathcal{E}(s_0)$ terminates.

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

If on the other hand Clover$_\mathcal{E}(s_0)$ terminates, then it computes the clover Clover$_\mathcal{E}(s_0)$ by Theorem 5.5 and we can decide boundedness as in the proof of Proposition 4.1 in the case of functional-lossy channel systems: just check whether any of the computed word-products contains a starred atomic expression $A^*$. 
In any case, we can decide boundedness, i.e., whether \( \text{Post}^*_{\mathcal{S}}(s_0) \) is finite. But this is impossible \cite{CFP96, May03b}. A similar argument works with reset Petri nets, where boundedness is also undecidable \cite{DFS98}.

5.2. Clover-Flattable Complete WSTS. We now characterize those \( \infty \)-effective complete WSTS on which Clover terminates.

A functional transition system \((S, F \rightarrow)\) 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 \( F \).) Ginsburg and Spanier \cite{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 \cite{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 \cite{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 \( \mathcal{S}_2 = (S_2, F_2) \) is a pair \((\mathcal{S}_1, \varphi)\), where:

1. \( \mathcal{S}_1 = (S_1, F_1) \) is a flat functional transition system;
2. and \( \varphi : \mathcal{S}_1 \rightarrow \mathcal{S}_2 \) is a morphism of transition systems. That is, \( \varphi \) is a pair of two maps, both written \( \varphi \), 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 \( s \in \text{dom} f_1 \) and \( s' = f_1(s) \), \( \varphi(s) \in \text{dom} \varphi(f_1) \) and \( \varphi(s') = \varphi(f_1)(\varphi(s)) \) (see Figure 5).

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

Recall that we equate ordered functional transition systems \((S, F, \leq)\) with their underlying function transition system \((S, F)\). The notion of flattening then extends to ordered functional transition systems. However, it is then natural to consider monotonic flattenings,
Definition 5.10 (Continuous Flattening). Let $\mathcal{S}_2 = (S_2, F_2, \leq_2)$ be a complete transition system. A flattening $(\mathcal{S}_1, \varphi)$ of $\mathcal{S}_2$ is continuous iff:

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

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

1. $\varphi(s_1) = s_0$ ($\varphi$ maps initial states to initial states);
2. $\text{cl}(\text{Cover}_{\mathcal{S}}(s_0)) = \text{cl}(\varphi(\text{cl}(\text{Cover}_{\mathcal{S}_1}(s_1))))$ ($\varphi$ preserves the closures of the covers of the initial states).

On complete WSTS—our object of study—, the second condition can be simplified to $\downarrow \text{Clover}_{\mathcal{S}}(s_0) = \downarrow \varphi(\text{Clover}_{\mathcal{S}_1}(s_1))$ (using Proposition 3.7 and the fact that $\varphi$, as a continuous map, is monotonic), or equivalently to $\text{Clover}_{\mathcal{S}}(s_0) = \text{Max}_2 \varphi(\text{Clover}_{\mathcal{S}_1}(s_1))$. Recall also that, when $\mathcal{S}$ is the completion $\hat{X}$ of a WSTS $X = (X, E, \leq)$, the clover of $s_0 \in X$ is a finite description of the cover of $s_0$ in $X$ (Proposition 3.9), and this is what $\varphi$ should preserve, up to taking downward closures.

There are apparently weaker and stronger forms 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 $\mathcal{S}$ be a complete transition system, and $s_0$ be a state. We say that $(\mathcal{S}, s_0)$ is weakly clover-flattable iff there is an continuous flattening $(\mathcal{S}_1, \varphi)$ of $\mathcal{S}$, and a state $s_1$ of $\mathcal{S}_1$ such that:

1. $\varphi(s_1) \leq s_0$;
2. $\text{cl}(\text{Cover}_{\mathcal{S}}(s_0)) \subseteq \text{cl}(\varphi(\text{cl}(\text{Cover}_{\mathcal{S}_1}(s_1))))$.

One may simplify the second condition slightly, to: $\text{Cover}_{\mathcal{S}}(s_0) \subseteq \text{cl}(\varphi(\text{cl}(\text{Cover}_{\mathcal{S}_1}(s_1))))$.

In the case of complete WSTS, this is equivalent to $\text{Clover}_{\mathcal{S}}(s_0) \leq^2 \varphi(\text{Clover}_{\mathcal{S}_1}(s_1))$.

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 $\Sigma$ 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 $\text{Pfx}(w_1^*w_2^*...w_k^*)$ of prefixes of all words from the latter is then again bounded [GS64].

Recall that a deterministic finite automaton (DFA) is a tuple $A = (\Sigma, Q, \delta, q_0, \text{Fin})$, where $\Sigma$ is a finite alphabet, $Q$ is a finite set of so-called control states, $q_0 \in Q$ is the initial state, $\text{Fin} \subseteq Q$ 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 $\text{Pfx}(a^*(bcc)^*(bcaa)^*)$ is then recognized by the same DFA, except that now all states are final.

This is general: $\text{Pfx}(w_1^*w_2^*...w_k^*)$ 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 $\text{Fin}$ component, and say that $A = (\Sigma, Q, \delta, q_0)$ itself is an rl-automaton.
Let us define the synchronized product.

**Definition 5.13** (Synchronized Product). Let $\mathcal{S} = (S, \xrightarrow{F}, \leq)$ 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 $\mathcal{S} \times \mathcal{A}$ as the ordered functional transition system $(S \times Q, \xrightarrow{F'}, \leq')$, where $F'$ is the collection of all partial maps $f \gg\delta : (s, q) \mapsto (f(s), \delta(q, f))$, 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 $\pi_1$ be the morphism of transition systems defined as first projection on states; i.e., $\pi_1(s, q) = s$ for all $(s, q) \in S \times Q$, $\pi_1(f \gg\delta) = f$ for all $f \in F$.

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

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

**Proof.** First, the technical condition that $\delta(q, f)$ should be defined for some $q \in Q$ only excludes maps $f \gg\delta$ with an empty domain, and is therefore benign. This technical condition is needed to define $\pi_1(f \gg\delta)$ 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 $f' = f \gg\delta$, such a $q$ exists by the technical condition, and this will yield $f(s)$ when $s \in \text{dom } f$, and will be undefined otherwise. So indeed $\pi_1(f \gg\delta) = f$.

$(S \times Q, \leq')$ 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 $f \gg\delta$ is partial continuous. Its domain is $\bigcup_{\delta(q, f) \text{ defined}} \text{dom } f \times \{q\}$, which is open. Moreover $f \gg\delta$ is clearly continuous for every $f \in F$: for any directed family $(s_i, q_i)_{i \in I}$ in dom$(f \gg\delta)$, first all $q_i$s must be equal, say $q_i = q \in Q$, and second $(s_i)_{i \in I}$ must be directed in dom $f$. So $f(\text{lub}\{s_i \mid i \in I\}) = \text{lub}\{f(s_i) \mid i \in I\}$, whence $(f \gg\delta)(\text{lub}\{(s_i, q) \mid i \in I\}) = \text{lub}\{(f(s_i) \mid i \in I), \delta(q, f)) = \text{lub}\{(f \gg\delta)(s_i, q) \mid i \in I\}$. That $\pi_1$ is continuous is clear as well.

Finally, the language of fireable transitions in $\mathcal{S} \times \mathcal{A}$ is contained in the language of $\mathcal{A}$, which is of the form Pfx$(w_1^*, w_2^* \ldots w_k^*)$, hence bounded. So $\mathcal{S} \times \mathcal{A}$ is flat. \qed
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 \( \mathcal{S}_1 \) is a synchronized product, and the continuous morphism of transition systems \( \varphi \) is first projection \( \pi_1 \):

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

The following is then obvious.

**Lemma 5.16.** On complete functional transition systems, the implications “strongly clover-flattable” \( \Rightarrow \) “clover-flattable” \( \Rightarrow \) “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 \( \text{Clover}_{\mathcal{S}}(s_0) \) can only terminate when \((\mathcal{S}, s_0)\) is strongly clover-flattable. We shall require the following lemma. For notational simplicity, we equate words \( g_1g_2 \) with compositions \( g_2 \circ g_1 \).

**Lemma 5.17.** Let \( \mathcal{S} = (S, F) \) 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 \( g_n^\infty(s) = \operatorname{lub}\{g_k^n(s) \mid k \in \mathbb{N}\} \). That the latter is in the Scott-open \( U \) implies that \( g_k^n(s) \) is in \( U \) for some \( k_n \in \mathbb{N} \). If \( s < g_n(s) \), 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 \( g_n^\infty \) 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 \( \mathcal{S} \) be an \( \infty \)-effective complete WSTS. If \( \text{Clover}_{\mathcal{S}} \) terminates on \( s_0 \), then \((\mathcal{S}, s_0)\) is strongly clover-flattable.

**Proof.** Write \( \mathcal{S} \) as \((S, F, \leq)\). Assume that \( \text{Clover}_{\mathcal{S}} \) terminates on \( s_0 \). Then it returns some finite set \( A \) such that \( A = \text{Clover}_{\mathcal{S}}(s_0) \) by Theorem 5.3. Enumerate the elements \( a_1, \ldots, a_k \) of \( A \). Each element \( a_i \) of \( A \), \( 1 \leq i \leq k \), is obtained as \( g_1^\infty g_2^\infty \ldots g_i^\infty(s_0) \), where each \( g_i \) is in \( F^* \).

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

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

The inclusion from right to left is obvious: for any state \((s, q)\) that is reachable from \( \downarrow(s_0, q_0) \) in \( \mathcal{S} \times \mathcal{A} \), \( s \) is reachable from \( \downarrow s_0 \) in \( \mathcal{S} \). So \( \pi_1(\text{Post}_{\mathcal{S}}^*(\downarrow s_0)) \subseteq \text{Post}_{\mathcal{S} \times \mathcal{A}}^*(\downarrow(s_0, q_0)) \).

Taking downward closures yields \( \pi_1(Cover_{\mathcal{S}}^*(s_0)) \subseteq Cover_{\mathcal{S} \times \mathcal{A}}^*(s_0, q_0) \), and taking closures
yields \( cl(\pi_1(Cover_{E \times A}(s_0, q_0))) \) \( \subseteq cl(Cover_{E \times A}^*(s_0, q_0)) = \downarrow A \) (using Theorem 5.5 and Proposition 3.7).

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

We now loop the loop and show that \( Clover_E \) terminates on \( s_0 \) whenever \( (E, s_0) \) is weakly clover-flattable (Theorem 5.21 below). This may seem obvious. In particular, if \( (E, s_0) \) is clover-flattable, then accelerate along the loops from \( E_1 \), where \( \varphi \) is a continuous flattening of \( E \). The difficulty is that we cannot actually choose to accelerate whenever we want: the \( Clover_E \) 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 \( Clover_E \), 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 dco, \( g: S \to S \) 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. \( cl(G) = \bigcup_{j=0}^{m-1} cl(G_j) = \downarrow \{ lub(G_0), lub(G_1), \ldots, lub(G_{m-1}) \} \);
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 \( \{ g^{p_j+\ell q_j}(s) \}, \ell \in \mathbb{N} \), where \( p_j \in \mathbb{N}, q_j \in \mathbb{N} \setminus \{0\}, \) and \( g^{p_j}(s) < g^{p_j+\ell q_j}(s) \);
3. for every \( j \), \( 0 \leq j < m \), \( s \not\in g^{p_j}(s) \),

**Proof.** First, the claim is obvious if \( S \) 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 \( s \) as \( g^{p_j}(s) \), and note that it cannot be the case that \( s < g^{p_j}(s) \), otherwise \( g^{p_j}(s) \) 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 \( G_0 = \{ s \}, G_1 = \{ g(s) \}, \ldots, G_{k-1} = \{ g^{k-1}(s) \}, G_k = \{ g^{k+i(m-k)}(s) \mid i \in \mathbb{N} \}, G_{k+1} = \{ g^{k+i+(m-k)}(s) \mid i \in \mathbb{N} \}, \ldots, G_{m-1} = \{ g^{m+i+(m-k)}(s) \mid i \in \mathbb{N} \} \).

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 \( g^{k+i(m-k)}(s) = g^{k+i+(m-k)}(g^k(s)) \leq g^{k+i+(m-k)}(g^m(s)) = g^{k+i+(m-k)}(s) \). So \( G_j = \{ g^{k+i(m-k)}(s) \}_{i \in \mathbb{N}} \) 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 $g^{j+i(m-k)}(s) = g^{j(m-k)}(g^j(s)) = g^{j+i(m-k)}(g^j)(s) = g^{j+i(m-k)}(s)$, so all elements of the sequence coincide. Otherwise, i.e., if $k \leq j < m$ and $g^j(s) \neq g^{j+i(m-k)}(s)$ (in which case $g^j(s) < g^{j+m-k}(s)$, since $g^{j+i(m-k)}(s) \leq g^{j+i(m-k)}(s)$ 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 $s_j = lub(G_j)$ 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 $\bigcup_{j=0}^{m-1} s_j$ 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 $cl(G) = \bigcup_{j=0}^{m-1} s_j = \bigcup_{j=0}^{m-1} \bigcup_{j=0}^{m-1} s_j = \bigcup_{j=0}^{m-1} \bigcup_{j=0}^{m-1} s_j$.

Take any element $x \in cl(G)$. Since $x \in cl(G)$, $x \subseteq s_j$ 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 $j < m$.

**Proposition 5.20.** Let $\mathcal{G}$ be an $\infty$-effective complete WSTS. Assume that $(\mathcal{G}, s_0)$ is weakly clover-flattable. Then $\text{Clover}_{\mathcal{G}}$ terminates on $s_0$.

**Proof.** Let $\mathcal{G}_1, \varphi$ be a continuous flattening of $\mathcal{G}$, and $s_1$ be a state of $\mathcal{G}_1$ such that $\varphi(s_1) \leq s_0$ and $\text{Cover}_{\mathcal{G}_1}(s_0) \subseteq \downarrow \varphi(\text{Cover}_{\mathcal{G}_1}(s_1))$, i.e., $\text{Clover}_{\mathcal{G}_1}(s_0) \subseteq \downarrow \varphi(\text{Clover}_{\mathcal{G}_1}(s_1))$. Write $\mathcal{G}_1$ as $(S_1, F_1, \leq)$. Since $\mathcal{G}_1$ is flat, every $g_1 \in F_1$ is in $w_1^* w_2^* \ldots w_n^*$, for some fixed sequence $w_1, w_2, \ldots, w_m \in F_1^*$.

Extend the action of $\varphi : F_1 \rightarrow F$ on words by $\varphi(f_1 f_2 \ldots f_p) = \varphi(f_1) \varphi(f_2) \ldots \varphi(f_p)$. Thus $\varphi(w_1), \ldots, \varphi(w_m)$ 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 $G_0, G_1, \ldots, G_{m-1}$ 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 $\text{Clover}_{\mathcal{G}_1}$ will eventually select the pair $(\varphi(w_1)^n(s_0), s_0)$ at line 2.(a) by fairness, and add $(\varphi(w_1)^n(s_0), s_0)$ to $A$. By condition (3), $s_0 \notin \varphi(w_1)^n(s_0)$, so $(\varphi(w_1)^n(s_0), s_0) = \varphi(w_1)^n(s_0)$.

So $\text{Clover}_{\mathcal{G}_1}$ will eventually add $\varphi(w_1)^n(s_0) = lub(G_j)$ to $A$.

Still taking the notations of the Lemma, for every $j$ such that $G_j$ contains more than one element, $\text{Clover}_{\mathcal{G}_1}$ will eventually select the pair $(\varphi(w_1)^p_j(s_0), s_0)$, adding $(\varphi(w_1)^p_j(s_0), s_0)$ to $A$. Using condition (3) as above, one sees that $(\varphi(w_1)^p_j(s_0), s_0) = \varphi(w_1)^p_j(s_0)$. Then, by fairness again (and this is the important point in the proof, where lub-acceleration is needed), $\text{Clover}_{\mathcal{G}_1}$ will eventually select the pair $(\varphi(w_1)^p_j(s_0), \varphi(w_1)^p_j(s_0))$, and therefore add $(\varphi(w_1)^p_j(s_0), \varphi(w_1)^p_j(s_0))$ to $A$. By condition (2), $(\varphi(w_1)^p_j(s_0), \varphi(w_1)^p_j(s_0))$ is just $\text{lub}(\varphi(w_1)^p_j(s_0), \varphi(w_1)^p_j(s_0)) \in \mathbb{N} = \text{lub}(G_j)$.

Let again $A_n$ be the value of the set $A$, computed by the procedure $\text{Clover}_{\mathcal{G}_1}$ 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 $n_1$, $\text{Clover}_{\mathcal{G}_1}$ 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 $A_n$. 


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 $\varphi(w_1)^{k_1}(\varphi(w_{i-1})^{k_{i-1}}(\ldots \varphi(w_1)^{k_1}(s_0)), k_1, \ldots, k_i \in \mathbb{N}$, 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 $\varphi(w_m)^{k_m}(\varphi(w_{m-1})^{k_{m-1}}(\ldots \varphi(w_1)^{k_1}(s_0)), k_1, \ldots, k_m \in \mathbb{N}$, is below some element of $A_n$. We claim that $\text{Cover}_{\mathcal{G}}(s_0)$ must stop after step $n$.

Let $U$ be the (open) complement of the closed set $\downarrow A_n$, and assume that $U$ intersects $\downarrow \text{Cover}_{\mathcal{G}}(s_0)$. Then $U$ must also intersect $\downarrow \varphi(\text{Cover}_{\mathcal{G}_1}(s_1))$, hence $\varphi(\text{Cover}_{\mathcal{G}_1}(s_1))$. (Remember that open subsets are upward-closed.) So $\varphi^{-1}(U)$ intersects $\text{Cover}_{\mathcal{G}_1}(s_1)$, whence $\varphi^{-1}(U)$ intersects $\downarrow \text{Cover}_{\mathcal{G}_1}(s_1)$, since $\varphi^{-1}(U)$ is upward-closed, using the fact that $U$ is and that $\varphi$ is monotonic. By Proposition 3.7, $\varphi^{-1}(U)$ intersects $\text{cl}(\text{Cover}_{\mathcal{G}_1}(s_1))$. Since $\varphi$ 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 $\text{Cover}_{\mathcal{G}_1}(s_1)$. So $U$ intersects $\varphi(\text{Cover}_{\mathcal{G}_1}(s_1))$, say at $a$. In particular, there is an $a_1 \in S_1$ such that $a \leq \varphi(a_1)$, and $a_1 \leq w_1^{k_1} w_2^{k_2} \ldots w_m^{k_m}(s_1)$, for some natural numbers $k_1, k_2, \ldots, k_m$.

Since $a \leq \varphi(a_1) \leq \varphi(w_1^{k_1} w_2^{k_2} \ldots w_m^{k_m})(s_1) \leq \varphi(w_1^{k_1} w_2^{k_2} \ldots w_m^{k_m})(s_0) = \varphi(w_m)^{k_m}(\varphi(w_{m-1})^{k_{m-1}}(\ldots \varphi(w_1)^{k_1}(s_0)), a$ 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 $\downarrow \text{Cover}_{\mathcal{G}}(s_0)$, i.e., $\downarrow \text{Cover}_{\mathcal{G}}(s_0) \subseteq \downarrow A_n$.

By Proposition 5.20 the converse inclusion holds. We conclude that the procedure $\text{Cover}_{\mathcal{G}}$ stops after the $n$th turn of the loop, because of the fixpoint test at line 2.

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

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

1. $(\mathcal{G}, s_0)$ is clover-flattable;
2. $(\mathcal{G}, s_0)$ is weakly clover-flattable;
3. $(\mathcal{G}, s_0)$ is strongly clover-flattable;
4. $\text{Cover}_{\mathcal{G}}(s_0)$ terminates.

---

### 5.3. Cover-flattability (without the “P” in “Cover”)

Turning to non-complete WSTS, we define:

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

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

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

1. $\varphi(x_1) = x_0$;
(2) \(\text{Cover}_\mathcal{X}(x_0) = \downarrow \varphi(\text{Cover}_\mathcal{X}(x_1))\).

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

1. \(\varphi(x_1) \leq x_0\);
2. and \(\text{Cover}_\mathcal{X}(x_0) \subseteq \downarrow \varphi(\text{Cover}_{\mathcal{X}_1}(x_1))\).

**Definition 5.25** (Strongly Cover-Flattable). Let \(\mathcal{X} = (X, \rightarrow, \leq)\) be an ordered functional transition system. We say that \((\mathcal{X}, x_0)\) is *strongly cover-flattable* iff there is an rl-automaton \(\mathcal{A}\), say with initial state \(g_0\), such that \(\text{Cover}_\mathcal{X}(x_0) = \pi_1(\text{Cover}_{\mathcal{X} \times \mathcal{A}}(x_0, g_0))\).

**Theorem 5.26.** Let \(\mathcal{X} = (X, \rightarrow, \leq)\) be an \(\omega^2\)-WSTS that is \(\infty\)-effective, in the sense that \(\hat{\mathcal{X}}\) is \(\infty\)-effective, i.e., that \((Sg)\infty\) is computable for every \(g \in F^*\). The following statements are equivalent:

1. \((\mathcal{X}, x_0)\) is cover-flattable;
2. \((\mathcal{X}, x_0)\) is weakly cover-flattable;
3. \((\mathcal{X}, x_0)\) is strongly cover-flattable;
4. \((\hat{\mathcal{X}}, \eta_X(x_0))\) is (weakly, strongly) clover-flattable;
5. \(\text{Clover}_{\hat{\mathcal{X}}}(\eta_X(x_0))\) terminates.

In this case, \(\text{Clover}_{\hat{\mathcal{X}}}(\eta_X(x_0))\) returns the clover \(A = \text{Clover}_\mathcal{G}(s_0)\), and this is a finite description of the cover, in the sense that \(\text{Cover}_\mathcal{X}(x_0) = \eta_X^{-1}(\downarrow A)\).

**Proof.** First, that \(\text{Clover}_{\hat{\mathcal{X}}}(\eta_X(x_0))\) computes the clover \(A\) is Theorem \ref{thm:clover}, and the fact that \(\text{Cover}_\mathcal{X}(x_0) = \eta_X^{-1}(\downarrow A)\), by Proposition \ref{prop:cover}. If we equate \(X\) with \(\hat{X}\), the latter means that the cover is just \(X \cap \downarrow A\).

Next, \(4\) is equivalent to \(5\), by Theorem \ref{thm:clover}. Note in particular that \(\hat{\mathcal{X}}\) is a complete WSTS by Theorem \ref{thm:complete-wsts}, and is \(\infty\)-effective by assumption.

The implications \(1 \implies 2\) and \(3 \implies 1\) are clear. For the latter, note that, since \(\text{Cover}_{\mathcal{X} \times \mathcal{A}}(x_0, g_0)\) is downward-closed, \(\pi_1(\text{Cover}_{\mathcal{X} \times \mathcal{A}}(x_0, g_0)) = \downarrow \pi_1(\text{Cover}_{\mathcal{X} \times \mathcal{A}}(x_0, g_0))\), and take \(\varphi = \pi_1\).

We now show that \((2)\) implies \((4)\), i.e., that if \((\mathcal{X}, x_0)\) is weakly cover-flattable, then \((\hat{\mathcal{X}}, \eta_X(x_0))\) is weakly cover-flattable. So let \(\mathcal{X}_1 = (X_1, \rightarrow, \leq\), \(\varphi\) and \(x_1\) as in Definition \ref{def:weakly-cover-flattable}. In particular, \(\varphi(x_1) \leq x_0\) and \(\text{Cover}_{\mathcal{X}}(x_0) \subseteq \downarrow \varphi(\text{Cover}_{\mathcal{X}_1}(x_1))\). Let \(S_1\) be the ideal completion Idl(\(\mathcal{X}_1\)), with inclusion as ordering, and define the complete transition system \(\mathcal{G}_1 = (S_1, \rightarrow, \leq\)), where \(F'_{\mathcal{G}} = \{\text{Idl}(f) \mid f \in F_1\}\). \(\text{Idl}(f)\) is the partial continuous function that maps every ideal \(D\) such that \(D \cap \text{dom } f \neq \emptyset\) to \(\downarrow f(D)\). Remember that \(\hat{\mathcal{X}} = \text{Idl}(\mathcal{X})\). Define \(\varphi' : S_1 \to \hat{\mathcal{X}}\) as \(\text{Idl}(\varphi)\); this is continuous. On transitions, \(\varphi'\) maps \(\text{Idl}(f)\) to \(\text{Idl}(\varphi(f))\): this is well-defined, as one can recover \(f\) from \(\text{Idl}(f)\), by the fact that \(f(x) = \text{lub}(\text{Idl}(f)(\downarrow x))\). So \((\mathcal{G}_1, \varphi')\) is a continuous flattening of \(\hat{\mathcal{X}}\). Let \(S_1 = \downarrow x_1, s_0 = \downarrow x_0\). We claim that \(\varphi'(s_1) \subseteq s_0\), and that \(\text{Cover}_{\hat{\mathcal{X}}}(s_0) \subseteq \text{cl}(\varphi'(\text{cl}(\text{Cover}_{\mathcal{G}_1}(s_1))))\). The first inequality is because \(\varphi'(s_1) = \text{Idl}(\varphi)(\downarrow x_1) = \downarrow \varphi(\downarrow x_1) = \downarrow \varphi(x_1) \subseteq \downarrow x_0 = s_0\), since \(\varphi(x_1) \leq x_0\). For the second inequality, let \(s\) be any element of \(\text{Cover}_{\hat{\mathcal{X}}}(s_0)\). So \(s \subseteq g''(s_0)\) for some \(g \in F_{\mathcal{G}}^*\). We observe that \(\text{Idl}\) is a functor, i.e., that \(\text{Idl}\) of the identity map is the identity, and that \(\text{Idl}(g_1g_2) = \text{Idl}(g_1) \text{Idl}(g_2)\) for all \(g_1, g_2\). So, writing \(g\) as a composition \(g_1g_2 \ldots g_k\) of elements \(g_i = \text{Idl}(h_i)\) of \(F_1\), \(g\) equals \(\text{Idl}(h)\), where \(h = h_1h_2 \ldots h_k \in F_{\mathcal{G}}^*\). It follows that \(s \subseteq \text{Idl}(h)(\downarrow x_0) = \downarrow h(x_0)\). Observe that \(h(x_0) \in \text{Cover}_{\hat{\mathcal{X}}}(x_0) \subseteq \downarrow \varphi(\text{Cover}_{\mathcal{X}_1}(x_1))\),
so \( s \subseteq \downarrow \phi(Cover_{\Xi}(x_1)) \). In particular, every element \( x \) of the ideal \( s \) is below some element of the form \( \phi(f(x_1)) \), \( f \in F_k^s \). We observe that \( x \in \phi'(Idl(f)(s_1)) \); indeed, \( \phi'(Idl(f)(s_1)) = Idl(\phi)(Idl(f)(s_1)) = Idl(\phi \circ f)(s_1) = \downarrow (\phi \circ f)(\downarrow x_1) = \downarrow \phi(f(x_1)) \), and \( x \) is in the latter since \( x \leq \phi(f(x_1)) \). From \( x \in \phi'(Idl(f)(s_1)) \), and since \( Idl(f)(s_1) \in Cover_{\Xi}(s_1) \), we deduce that \( x \in \phi'(Cover_{\Xi}(s_1)) \). Since \( x \) is arbitrary in \( s \), \( s \subseteq \phi'(Cover_{\Xi}(s_1)) \), i.e., \( s \in \phi'(Cover_{\Xi}(s_1)) \subseteq cl(\phi'(Idl(Cover_{\Xi}(s_1)))) \).

Finally, we show that (4) implies (3), i.e., that if \( \widehat{(\Xi, \eta_X(x_0))} \) is strongly cover-flattable, then \( (\Xi, x_0) \) is strongly cover-flattable. Let \( A \) be an rl-automaton, with initial state \( q_0 \), such that \( cl(Cover_{\widehat{\Xi}}(\eta_X(x_0))) = cl(\pi_1(cl(Cover_{\widehat{\Xi} \times A'}(\eta_X(x_0), q_0)))) \). We claim that \( Cover_{\Xi}(x_0) = \pi_1(Cover_{\Xi \times A'}(x_0, q_0)) \), where \( A' \) is the automaton obtained from \( A \) by replacing each \( Sg \) transition by a \( g \) transition, \( g \in F \). (Note by the way that the definition of \( Sg \) is the same as that of \( Idl(g) \) above.) The inclusion from right to left is obvious, so let us show that \( Cover_{\Xi}(x_0) \subseteq \pi_1(Cover_{\Xi \times A'}(x_0, q_0)) \). Let \( x \) be any element of \( Cover_{\Xi}(x_0) \). So \( x \leq g(x_0) \) for some \( g \in F \). Then \( x \in \downarrow g(x_0) = \downarrow g(\downarrow x_0) = Idl(g)(\eta_X(x_0)) \), so \( x \in Cover_{\widehat{\Xi}}(\eta_X(x_0)) \). By assumption \( \downarrow x \) is in \( cl(\pi_1(cl(Cover_{\widehat{\Xi} \times A'}(\eta_X(x_0), q_0)))) \). We may simplify this by observing that \( cl(f(cl(A))) = cl(f(A)) \) for any continuous map \( f \) and any subset \( A \), so that \( \downarrow x \in cl(\pi_1(Cover_{\widehat{\Xi} \times A'}(\eta_X(x_0), q_0))) \). In \( \widehat{\Xi} = Idl(X) \), the closure \( cl(A) \) of any downward-closed subset \( A \) of \( Idl(X) \) equals \( Lub(A) \), since \( Idl(X) \) 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 \( A = \pi_1(Cover_{\widehat{\Xi} \times A'}(\eta_X(x_0), q_0)) \), \( x \) is in some ideal \( s \) such that \( (s, q) \in Cover_{\Xi \times A'}(\eta_X(x_0), q_0) \) for some state \( q \) of \( A \). That is, \( s \subseteq Sg(\eta_X(x_0)) \) 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 \( Sg_1 Sg_2 \ldots Sg_k \) in \( A \) from \( q_0 \). In particular, \( x \) is also the state obtained by reading the word \( g_1 g_2 \ldots g_k \) in \( A' \) from \( q_0 \). And \( s \subseteq Sg(\eta_X(x_0)) \) means that \( s \subseteq \downarrow g(\downarrow x_0) = \downarrow g(x_0) \), so \( x \in s \) implies \( x \leq g(x_0) \). In particular, \( (x, q) \in Cover_{\Xi \times A'}(x_0, q_0) \), so \( x \in \pi_1(Cover_{\Xi \times A'}(x_0, q_0)) \).

By a slight abuse of language, say that a functional WSTS \( \mathcal{G} = (S, \widehat{\to}, \leq) \) is cover-flattable iff \( (\mathcal{G}, s_0) \) 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 \( \mathbb{N}^k \) 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 \( \Xi \) 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 \( Sf \) is defined by: \( Sf(q, \vec{x}) = (q', \vec{x} + \vec{b} - \vec{a}) \), provided \( \vec{x} \geq \vec{a} \), this time for all \( \vec{x} \in \mathbb{N}^k \). So the completion \( \widehat{\mathcal{G}} \) of the VASS is \( \infty \)-effective. On these, the Karp-Miller algorithm terminates \( \text{[KM69]} \), hence also the generalized Karp-Miller algorithm of Section 4.1. By Proposition 5.6, \( \text{Clover}_{\mathcal{G}} \) terminates on any input \( s_0 \in Q \times \mathbb{N}^k \). So \( \Xi \) is cover-flattable, by Theorem 5.26.

**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 \( \infty \)-effective, see Section 4.5. However the cover is undecidable both for reset Petri nets and (functional-)lossy channel systems \( \Xi \), so \( \text{Clover}_{\widehat{\Xi}}(\eta_X(x_0)) \) must fail to terminate for some initial state \( x_0 \). We conclude by Theorem 5.26.
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 \( f^\infty \)
is computable \cite{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, \( \mathcal{C} = (Q, R, \to) \) where \( Q \) is a finite set of control states, \( R = \{r_1, r_2, \ldots r_k\} \)
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 us to encode
the graph of affine functions. A (partial) function \( f : \mathbb{N}^n \to \mathbb{N}^n \) 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 \( f : \mathbb{N}^n \to \mathbb{N}^n \) by continuity to \( f : \mathbb{N}_\omega^n \to \mathbb{N}_\omega^n \), 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 \( Sf \).

**Definition 6.2.** An affine counter system (with \( n \) counters), a.k.a. an 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 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. \cite{DFGvD06}.

For counter systems (which include Minsky machines), monotonicity is undecidable.
Clearly, a counter system \( \mathcal{S} \) is well-structured iff \( \mathcal{S} \) 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 \( \text{dom}(f) \) 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 \( A \in \mathbb{N}^{n \times n}, \vec{b} \in \mathbb{N}^n \) (note that \( \vec{b} \) is not in \( \mathbb{Z}^n \)) and \( \text{dom}(f) \) 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 \( \text{dom}(f) \) upward-closed, the function \( f^\infty \) 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 \( \infty \)-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 \( \text{Post}(\vec{s}) = \{ \vec{t} \mid \exists f \in F : f(\vec{t}) = \vec{s} \} \) is Presburger-definable. Note also that monotonic affine function are continuous, and \( \mathbb{N}^\omega = \) a continuous dcwo. Finally, for every Presburger monotonic affine function \( f \), the function \( f^\infty \) is recursive, so the considered ACS is \( \infty \)-effective.

**Corollary 6.6.** One can decide whether the completion of an ACS is an \( \infty \)-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 \( \infty \)-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 \( \omega^2 \)-WSTS. This delineates a new, robust class of WSTS: all known WSTS are \( \omega^2 \)-WSTS. The property of being an \( \omega^2 \)-WSTS is also important to ensure progress in Karp-Miller-like procedures.

We have also defined a simple procedure, Clover, for computing the clover for \( \infty \)-effective complete WSTS \( S \). This captures the essence of generalized forms of the Karp-Miller procedure, while terminating in more cases. We have shown that that Clover 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 procedure. We will also analyze whether Clover 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 \texttt{Clover}\textsubscript{S} procedure may fail to terminate. This is in contrast to the Expand, Enlarge and Check forward algorithm of \cite{GRvB07}, which always terminates, hence decides coverability. One may want to combine the best of both worlds, and the lub-accelerations of \texttt{Clover}\textsubscript{S} 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 \cite{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.

\textbf{Acknowledgement}

The authors wish to acknowledge fruitful discussions with Sylvain Schmitz, and to thank the anonymous referees for their comments.

\textbf{References}

\begin{itemize}
  \item \cite{ABJ98} Parosh Abdulla, Ahmed Bouajjani, and Bengt Jonsson. On-the-fly analysis of systems with unbounded, lossy Fifo channels. In \textit{Proc. 10th Intl. Conf. Computer Aided Verification (CAV’98)}, pages 305–318, Vancouver, Canada, June 1998. Springer Verlag LNCS 1427.
  \item \cite{ACABJ04} Parosh Aziz Abdulla, Aurore Collomb-Annichini, Ahmed Bouajjani, and Bengt Jonsson. Using forward reachability analysis for verification of lossy channel systems. \textit{Formal Methods in System Design}, 25(1):39–65, 2004.
  \item \cite{AČJT00} Parosh Aziz Abdulla, Karlis Čerāns, Bengt Jonsson, and Yih-Kuen Tsay. Algorithmic analysis of programs with well quasi-ordered domains. \textit{Information and Computation}, 160(1–2):109–127, 2000.
  \item \cite{ADMN04a} Parosh Aziz Abdulla, Johann Deneux, Pritha Mahata, and Aletta Nylén. Forward reachability analysis of timed Petri nets. In \textit{FORMATS/FTRTFT}, pages 343–362. Springer Verlag LNCS 3253, 2004.
  \item \cite{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, \textit{FORMATS/FTRTFT}, volume 3253 of \textit{Lecture Notes in Computer Science}, pages 343–362. Springer, 2004.
  \item \cite{AJ94} Samson Abramsky and Achim Jung. Domain theory. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, editors, \textit{Handbook of Logic in Computer Science}, volume 3, pages 1–168. Oxford University Press, 1994.
  \item \cite{AN00} Parosh Aziz Abdulla and Aletta Nylén. Better is better than well: On efficient verification of infinite-state systems. In \textit{Proc. 14th IEEE Symp. Logic in Computer Science (LICS’00)}, pages 132–140, 2000.
  \item \cite{BFLS05} Sébastien Bardin, Alain Finkel, Jérôme Leroux, and Philippe Schnoebelen. Flat acceleration in symbolic model checking. In \textit{Proc. 3rd Intl. Symp. Automated Technology for Verification and Analysis (ATVA’05)}, pages 474–488. Springer Verlag LNCS 3707, 2005.
  \item \cite{BG11} Laura Bozzielli and Pierre Ganty. Complexity analysis of the backward coverability algorithm for vass. In \textit{Proceedings of the 5th international conference on Reachability problems, RP’11}, pages 96–109, Berlin, Heidelberg, 2011. Springer-Verlag.
  \item \cite{CFP96} Gérard Cécé, Alain Finkel, and S. Purushothaman Iyer. Unreliable channels are easier to verify than perfect channels. \textit{Information and Computation}, 124(1):20–31, January 1996.
  \item \cite{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, \textit{Proceedings of the 32nd International Conference on Applications and Theory of Petri Nets (ICATPN’11)},
[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-modifying 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.