Noetherian Spaces in Verification

Jean Goubault-Larrecq[*,†]

[*] Preuves, Programmes et Systèmes, UMR 7126, CNRS and University Paris Diderot
[†] LSV, ENS Cachan, CNRS, INRIA

Abstract. Noetherian spaces are a topological concept that generalizes well quasi-orderings. We explore applications to infinite-state verification problems, and show how this stimulated the search for infinite procedures à la Karp-Miller.

1 Introduction

The purpose of this paper is to given a gentle introduction to the theory of Noetherian spaces, in the context of verification of infinite-state systems.

Now such a statement can be intimidating, all the more so as Noetherian spaces originate in algebraic geometry [20, chapitre 0]. Their use there lies in the fact that the Zariski topology of a Noetherian ring is Noetherian.

My purpose is to stress the fact that Noetherian spaces are merely a topological generalization of the well-known concept of *well quasi-orderings*, a remark that I made in [19] for the first time. Until now, this led me into two avenues of research.

The first avenue consists in adapting, in the most straightforward way, the theory of *well-structured transition systems* (WSTS) [1, 4, 16, 21] to more general spaces. WSTS include such important examples as Petri nets and extensions, and lossy channel systems. After some technical preliminaries in Section 2, I will describe the basic theory of Noetherian spaces in Section 3. This leads to a natural generalization of WSTS called *topological WSTS*, which I will describe in Section 4.

In [19], I described a few constructions that preserve Noetherianness. We shall give a more complete catalog in Section 3: $\mathbb{N}^k$, Σ^* and in general every well-quasi-ordered set, but several others as well, including some that *do not* arise from well-quasi-orders.

We apply this to the verification of two kinds of systems that are *not* WSTS. We do not mean these to be any more than toy applications, where decidability occurs as a natural byproduct of our constructions. I certainly do not mean to prove any new, sophisticated decidability result for some realistic application in verification, for which we should probably exert some more effort. I only hope to convince the reader that the theory of Noetherian spaces shows some potential.

The first application, *oblivious stack systems*, are k-stack pushdown automata in which one cannot remember which letter was popped from a stack: see Section 5. The second one, *polynomial games*, is an extension of Seidl and Müller-Olm's static analysis of so-called polynomial programs [29] to games played between two players that can compute on real and complex numbers using addition, subtraction, multiplication, and (dis)equality tests: see Section 6, where we also consider the case of lossy *concurrent* polynomial games, i.e., networks of machines running polynomial programs and which communicate through lossy signaling channels.

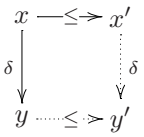
The second avenue of research has led Alain Finkel and myself to make significant progress in designing extensions of the Karp-Miller coverability algorithm to other WSTS than just Petri nets or even counter machines. I will say a bit more in Section 7. This line of research stemmed from the remarkable relationship between the concepts of Noetherianness and of sobriety, which I will explain. It is fair to say that the results obtained with A. Finkel could be proved without any recourse to Noetherian spaces. But the decisive ideas come from topology, and in particular from the important role played by *irreducible* closed sets in Noetherian spaces.

2 Technical Preliminaries

A well quasi-ordering (*wqo*) is a quasi-ordering (a reflexive and transitive relation) that is not only well-founded, i.e., has no infinite descending chain, but also has no infinite antichain (a set of incomparable elements). An alternative definition is: $\leq$ is a wqo on X iff every sequence $(x_n)_{n\in\mathbb{N}}$ in X contains a pair of elements such that $x_i \leq x_j$, $i < j$. Yet another equivalent definition is: $\leq$ is wqo iff every sequence $(x_n)_{n\in\mathbb{N}}$ has a non-decreasing subsequence $x_{i_0} \leq x_{i_1} \leq \ldots \leq x_{i_k} \leq \ldots$, $i_0 < i_1 < \ldots < i_k < \ldots$

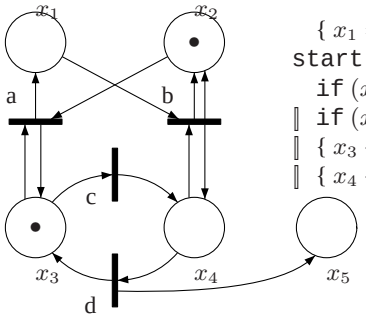
WSTS. One use of well quasi-orderings is in verifying *well-structured transition systems*, a.k.a. *WSTS* [1, 4, 16, 21]. These are transition systems, usually infinite-state, with two ingredients. (For simplicity, we shall consider *strongly monotonic* well-structured transition systems only.)

First, there is a *well* quasi-ordering $\leq$ on the set X of states. Second, the transition relation δ commutes with $\leq$, i.e., if $x\ \delta\ y$ and $x \leq x'$, then there is a state y' such that $x'\ \delta\ y'$ and $y \leq y'$:



(1)

Examples include Petri nets [34] and their extensions, reset/transfer Petri nets for example, in general all affine counter systems [15], the close concept of VASS [23], BVASS [37, 11], lossy channel systems [3], datanets [26], certain process algebras [7]; and some problems, such as those related to timed Petri nets [5] admit elegant solutions by reduction to an underlying WSTS.



$\{\ x_1 = x_4 = x_5 = 0;\ x_2 = x_3 = 1;\ \}$; // init
`start:`
`if` $(x_3 \geq 1)\ \{\ x_2 - -;\ x_1 + +;$ `goto start;` $\}$ // a
$\|$ `if` $(x_4 \geq 1)\ \{\ x_1 - -;\ x_2 += 2;$ `goto start;` $\}$ // b
$\|\ \{\ x_3 - -;\ x_4 + +;$ `goto start;` $\}$ // c
$\|\ \{\ x_4 - -;\ x_3 + +;\ x_5 + +;$ `goto start;` $\}$ // d

Fig. 1. A Petri Net

We illustrate the concept using Petri nets. I won't define what Petri nets are exactly. Look at Figure 1, left, for an example. This is a net with 5 *places* $x_1, \ldots, x_5$, each containing a certain number of *tokens* (shown as bullets): the initial state is shown, with one token in places x_2 and x_3, and none anywhere else. Petri nets run by firing *transitions*, shown as black bars |. Doing so means, for each incoming arrow $\bigcirc\!\!\rightarrow\!|$, remove one token from the source place $\bigcirc$, and for each outgoing arrow $|\!\rightarrow\!\bigcirc$, add one token to the target place. This can only be done provided there are enough tokens in the source places. E.g., transition a can only fire provided there is at least one token in x_2 and at least one token in x_3 (which is the case in the initial state shown in the figure), will remove one from x_2 and one from x_3, then put back one into x_3 and put one into x_1. The net effect of transition a is therefore to *move* one token from x_2 to x_1, *provided* there is at least one token in x_3. If we agree to use a variable x_i to hold the current number of tokens in place x_i, a C-like notation for this is `if` $(x_3 \geq 1)$ { $x_2 - -;\ x_1 + +;$ }, as we have shown on the right-hand side of the figure.

In general, Petri nets are just another way of writing *counter machines without zero test*, i.e., programs that operate on finitely many variables $x_1, \ldots, x_k$ containing natural numbers; the allowed operations are adding and subtracting constants from them, as well as testing whether $x_i \geq c$ for some constants c. The general counter machines also offer the possibility of testing whether x_i equals 0. Crucially, Petri nets do not allow for such zero tests. This makes a difference, as reachability and coverability (see later) is undecidable for general counter machines [28], but decidable for Petri nets [34].

Let us check that Petri nets define WSTS. Consider a Petri net with k places x_1, $\ldots$, x_k. The states, also called *markings*, are tuples $\boldsymbol{n} = (n_1, \ldots, n_k) \in \mathbb{N}^k$, where n_i counts the number of tokens in place x_i. The state space is $\mathbb{N}^k$. Order this by the canonical, pointwise ordering: $(n_1, \ldots, n_k) \leq (n'_1, \ldots, n_k)$ iff $n_1 \leq n'_1$ and $\ldots$ and $n_k \leq n'_k$. This is wqo by *Dickson's Lemma*, whose proof can be safely left to the reader (reason componentwise, observing that $\mathbb{N}$ is itself well-quasi-ordered).

The transitions are each given by a pair of constant vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{N}^k$: we have $\boldsymbol{n}\ \delta\ \boldsymbol{n}'$ iff $\boldsymbol{a} \leq \boldsymbol{n}$ and $\boldsymbol{n}' = \boldsymbol{n} - \boldsymbol{a} + \boldsymbol{b}$ for one of the transitions. For example, transition a in Figure 1 can be specified by taking $\boldsymbol{a} = (0, 1, 1, 0, 0)$ and $\boldsymbol{b} = (1, 0, 1, 0, 0)$. It is easy to see that Diagram (1) holds. Indeed, if some transition is firable from $\boldsymbol{n}$, then it will remain firable even if we add some tokens to some places, and triggering it will produce a new state with more tokens as well.

The standard backward algorithm for WSTS [4, 16]. The *coverability* problem is: given two states x, y, can we reach some state z from x such that $y \leq z$? This is a form of reachability, where we require, not to reach x exactly, but some state in the upward closure $\uparrow x$ of x.

For any subset A of X, let $\mathrm{Pre}^{\exists}\delta(A)$ be the preimage $\{x \in X \mid \exists y \in A \cdot x\ \delta\ y\}$. The commutation property (1) of strongly monotonic systems ensures that the preimage $\mathrm{Pre}^{\exists}\delta(V)$ of any upward closed subset V is again upward closed (V is upward closed iff whenever $x \in V$ and $x \leq x'$, then $x' \in V$). One can then compute $\mathrm{Pre}^{\exists *}\delta(V)$, the set of states in X from which we can reach some state in V in finitely many steps, assuming that upward closed subsets are representable and $\mathrm{Pre}^{\exists}(A)$ is computable from any upward closed subset A: Compute the set V_i of states from which we can reach

some state in V in at most i steps, backwards, by $V_0 = V$, $V_{i+1} = V_i \cup \mathrm{Pre}^{\exists}\delta(V_i)$: this stabilizes at some stage i, where $V_i = \mathrm{Pre}^{\exists *}\delta(V)$.

To decide coverability, then, compute $\mathrm{Pre}^{\exists *}\delta(\uparrow y)$, and check whether x is in it.

This is a very simple algorithm. The only subtle point has to do with termination. One notices indeed that $V_0 \subseteq V_1 \subseteq \ldots \subseteq V_i \subseteq \ldots$, and that if the sequence ever stabilizes at stage i, we can detect it by testing whether $V_{i+1} \subseteq V_i$. Now it must stabilize, *because $\leq$ is wqo*. Indeed, in a wqo every upward closed subset U must be the upward closure $\uparrow E = \{x \in X \mid \exists y \in E \cdot y \leq x\}$ of some finite set E. (Proof: any element $x \in U$ is above a minimal element in U: start from x and go down until you cannot go further—which must eventually happen since $\leq$ is well-founded. The set E of all minimal elements of U must then be finite since there is no infinite antichain.) In particular, $V_\omega = \bigcup_{i \in \mathbb{N}} V_i$ can be written $\uparrow\{x_1, \ldots, x_n\}$. Each x_j, $1 \leq j \leq n$, must be in some V_{i_j}, so they are all in V_i, where $i = \max(i_1, \ldots, i_n)$: it follows that $V_\omega = V_i$, and we are done.

Noetherian spaces. The idea of [19] lies in replacing order theory by topology, noticing that the role of wqos will be played by Noetherian spaces.

Indeed, topology *generalizes* order theory. (To do so, we shall require topological spaces that are definitely non-Hausdorff, even non-T_1, hence very far from metric spaces or other topological spaces commonly used in mathematics.) Any topological space X indeed carries a quasi-ordering $\leq$ called the *specialization quasi-ordering* of X: $x \leq y$ iff every open neighborhood U of x also contains y. It is fruitful, from a computer science perspective, to understand opens U as *tests*; then $x \leq y$ iff y *simulates* x, i.e., passes all the tests that x passes.

Note that in particular every open U is upward closed in $\leq$, and every closed subset F is downward closed. Similarly, continuous map $f : X \to Y$ are in particular monotonic (the converse fails).

In the opposite direction, there are several topologies on X with a given specialization quasi-ordering $\leq$. The finest one (with the most opens) is the *Alexandroff topology*: its opens are all the upward closed subsets. The coarsest one (with the fewest opens) is the *upper topology*: its closed subsets are all unions of subsets of the form $\downarrow E$ (the downward closure of E), E finite. In between, there are other interesting topologies such as the Scott topology, of great use in domain theory [6].

3 The Basic Theory of Noetherian Spaces

A topological space X is *Noetherian* iff every open subset of X is compact. (I.e., one can extract a finite subcover from any open cover.) Equivalently:

Definition 1. *X is Noetherian iff there is no infinite ascending chain $U_0 \subsetneq U_1 \subsetneq \ldots \subsetneq U_n \subsetneq \ldots$ of opens in X.*

The key fact, showing how Noetherian spaces generalize wqos, is the following [19, Proposition 3.1]: $\leq$ is wqo on the set X iff X, equipped with the Alexandroff topology of $\leq$, is Noetherian. This provides plenty of Noetherian spaces.

It turns out that there are also Noetherian spaces that do not arise from wqos, thus Noetherian spaces provide a strict generalization of wqos. The prime example is $\mathbb{P}(X)$, the infinite powerset of X, with the *lower Vietoris topology*, defined as the coarsest that makes $\Diamond U = \{\{A \in \mathbb{P}(X) \text{ resp., } \in \mathbb{P}^*(X) \mid A \cap U \neq \emptyset\}$ open for every open subset U of X. When X is a poset, $\mathbb{P}(X)$ is quasi-ordered by the *Hoare quasi-ordering* $\leq^\flat$: $A \leq^\flat B$ iff for every $a \in A$, there is a $b \in B$ such that $a \leq b$. Assuming X wqo, $\mathbb{P}(X)$ is not wqo in general, however it *is* Noetherian in the lower Vietoris topology—which turns out to be the upper topology of $\leq^\flat$ [19, Corollary 7.4]. A related example, in fact almost the same one, is the *Hoare powerdomain* $\mathcal{H}(X)$ of a space X: this is the space of all non-empty (even infinite) closed subsets F of X, with the lower Vietoris topology, namely the upper topology of $\subseteq$. This is one of the powerdomains used in domain theory, and is a model of the so-called *angelic* variant of non-deterministic choice [6]. Then $\mathcal{H}(X)$ is Noetherian as soon as X is [19, Theorem 7.2].

We don't have any practical applications of $\mathbb{P}(X)$ or $\mathcal{H}(X)$ in verification today. We shall give an application of Noetherian spaces in Section 6, where the underlying space is Noetherian, but is not the Alexandroff topology of a wqo. A simpler example is given by Σ^*, the space of all finite words over a finite alphabet Σ, with the upper topology of the prefix ordering $\leq^{\text{pref}}$. We shall see below why this is Noetherian, and shall use it in Section 5. Note that $\leq^{\text{pref}}$ is certainly *not* wqo, as soon as Σ contains at least two letters a and b: b, ab, aab, $\ldots$, $a^n b$, $\ldots$, is an infinite antichain.

The key insight in the theory of Noetherian spaces is how Noetherianness interacts with *sobriety*. A topological space X is *sober* if and only if every irreducible closed subset C is the closure $\downarrow x$ of a unique point $x \in X$. The closure of a point x is always the downward closure $\downarrow x$ with respect to the specialization quasi-ordering. A closed subset C is *irreducible* iff $C \neq \emptyset$, and whenever C is included in the union of two closed subset, then C must be contained in one of them. For every $x \in X$, it is clear that $\downarrow x$ is irreducible. A sober space has no other irreducible closed subset.

Sober spaces are important in topology and domain theory [6], and are the cornerstone of Stone duality. We refer the reader to [6, Section 7] or to [18, Chapter V] for further information. We shall be content with the following intuitions, which show that sobriety is a form of *completeness*. A space is T_0 iff its specialization quasi-ordering $\leq$ is an ordering, i.e., any two distinct points x, y, can be separated by some open U (think of it as a test that one point passes but not the other one). So a space is T_0 if it has *enough opens* to separate points. A sober space is a T_0 space that also has *enough points*, in the sense that any closed set C that looks like the closure of a point (in the sense that it is irreducible) really is so: $C = \downarrow x$, where necessarily $x = \max C$. Another indication is that, if X is sober, then X is a *dcpo* [6, Proposition 7.2.13]: for every directed family $(x_i)_{i \in I}$, in particular for every chain, the *limit* $\sup_{i \in I} x_i$ exists. So a sober space is complete also in this sense.

Any topological space X can be completed to obtain a sober space $\mathcal{S}(X)$, the *sobrification* of X, which has the same lattice of open subsets (up to isomorphism), and possibly more points. In a sense, we add all missing limits $\sup_{i \in I} x_i$ to X. $\mathcal{S}(X)$ is defined as the collection of all irreducible closed subsets C of X, with the upper topology of $\subseteq$. X is then embedded in $\mathcal{S}(X)$, by equating each point $x \in X$ with $\downarrow x \in \mathcal{S}(X)$.

The first key point about the interaction between sobriety and Noetherianness is that for any space X, X is Noetherian iff $\mathcal{S}(X)$ is Noetherian [19, Proposition 6.2]. This is obvious: X and $\mathcal{S}(X)$ have isomorphic lattices of open sets. Thus, to show that X is Noetherian, it is enough to show that $\mathcal{S}(X)$ is. The following is the cornerstone of the whole theory, and allows one to check that a sober space is Noetherian by checking simple properties of its specialization quasi-ordering [19, Theorem 6.11]:

Theorem 1 (Fundamental Theorem of Sober Noetherian Spaces). *The sober Noetherian spaces are exactly the spaces whose topology is the upper topology of a well-founded partial order $\leq$ that has properties W and T.*

We say that X has *property W* iff, for every $x, y \in X$, there is a finite subset E of maximal lower bounds of x and y, such that every lower bound of x and y is less than or equal to some element of E; i.e., $\downarrow x \cap \downarrow y = \downarrow E$. Similarly, it has *property T* iff the space X itself is of the form $\downarrow E$, E finite.

This allows us to prove that the product of two Noetherian spaces X, Y is again Noetherian [19, Theorem 6.13]. The specialization quasi-ordering on $\mathcal{S}(X \times Y) \cong \mathcal{S}(X) \times \mathcal{S}(Y)$ is the product ordering, and it is easy to see that the product of two well-founded orderings is again well-founded, and similarly for properties T and W.

Since every wqo is Noetherian, classical spaces such as $\mathbb{N}^k$, or Σ^* with the (Alexandroff topology of the) divisibility ordering (Higman's Lemma [22]), or the set of all ground first-order terms $\mathcal{T}(X)$ (a.k.a., vertex-labeled, finite rooted trees) with tree embedding (Kruskal's Theorem [25]), are Noetherian.

It is natural to ask ourselves whether there are topological version of Higman's Lemma and Kruskal's Theorem. There are indeed, and at least the former case was alluded to in [13, Theorem 5.3].

Theorem 2 (Topological Higman Lemma). *Let X be a topological space, X^* the space of all finite words on the alphabet X with the* subword topology*, defined as the coarsest one such that $X^*U_1X^*U_2X^* \ldots X^*U_nX^*$ is open for every sequence of open subsets $U_1, U_2, \ldots, U_n$ of X.*

The specialization quasi-ordering of X^* is the embedding quasi-ordering $\leq^*$, where $w \leq^* w'$ iff one obtains w' from w by increasing some letters and inserting some others.

If X is Noetherian, then so is X^*.

One also observes that if X is Alexandroff, then so is X^*. One therefore obtains Higman's Lemma, that $\leq^*$ is wqo as soon as $\leq$ is wqo on X, as a consequence. Thinking of opens as tests, a word passes the test $X^*U_1X^*U_2X^* \ldots X^*U_nX^*$ iff it has a length n subword whose letters pass the tests $U_1, \ldots, U_n$.

As a corollary, the space $X^{\circledast}$ of all *multisets* of elements of X, with the *sub-multiset* topology, is Noetherian whenever X is. This is the coarsest one that makes open the subsets $X^{\circledast} \odot U_1 \odot U_2 \odot \ldots \odot U_n$ of all multisets containing at least one element from U_1, one from U_2, ..., one from U_n, where $U_1, U_2, \ldots, U_n$ are open in X. This follows from Theorem 2 because $X^{\circledast}$ is the image of X^* by the *Parikh mapping* $\Psi : X^* \to X^{\circledast}$ that sends each word to its multiset of letters, and because of the easy result that the continuous image of any Noetherian space is again Noetherian.

The way I initially proved Theorem 2 [13, full version, available on the Web, Theorem E.20] is interesting. One first characterizes the irreducible closed subsets of X^*

exactly: they are the *word-products* $P = e_1 e_2 \dots e_n$, where each e_i is an *atomic expression*, either of the form F^* with F non-empty and closed in X, or $C^?$ (denoting sequences of at most one letter taken from C), where C is irreducible closed in X. Note how close this is from the definition of products and SREs [2]. In fact, the latter are the special case one obtains when considering X finite, in which case irreducible closed sets C are single letters a, and non-empty closed sets F are just non-empty subsets of X.

Properties T and W are easy, and one realizes that there is no infinite descending chain of word-products $P_0 \supsetneq P_1 \supsetneq \dots \supsetneq P_k \supsetneq \dots$, as soon as X is Noetherian. This may seem surprising, however one can characterize inclusion of word-products in an algorithmic way (see [13, Definition 5.1]), and from this definition it is clear that if $P \supsetneq P'$, where $P = e_1 e_2 \dots e_m$ and $P' = e'_1 e'_2 \dots e'_n$, then the multiset $\{\!|e_1, e_2, \dots, e_m|\!\}$ is strictly larger than $\{\!|e'_1, e'_2, \dots, e'_n|\!\}$ in the multiset extension $\sqsupset^{mul}$ of $\sqsupset$, defined by: $C'^? \sqsupset C^?$ iff $C' \supsetneq C$; $F'^* \sqsupset F^*$ iff $F' \supsetneq F$; $F'^* \sqsupset C^?$ iff $F' \supseteq C$; and $C'^? \not\sqsupset F^*$. When X is Noetherian, $\supsetneq$ is well-founded, and we conclude by Theorem 1.

This argument is very similar to Murthy and Russell's constructive proof of Higman's Lemma [30], a paper I only discovered very recently (April 2010).

Using a similar line of proof, we obtain an analogous result on finite trees. We equate finite trees on X with ground, unranked first-order terms with function symbols taken from X, which we simply call *terms* on X.

Theorem 3 (Topological Kruskal Theorem). *Let X be a topological space, $\mathcal{T}(X)$ be the set of all terms on X, defined by the grammar $s, t, \dots ::= f(t_1, \dots, t_n)$ ($f \in X$, $n \in \mathbb{N}$). Write $\boldsymbol{t}$ for the sequence $t_1 \dots t_n$. Define the* simple tree expressions *by the grammar $\pi ::= \Diamond U(\pi_1 \mid \dots \mid \pi_n)$ (U open in X, $n \in \mathbb{N}$), and let $\Diamond U(\pi_1 \mid \dots \mid \pi_n)$ denote the collection of all terms that have a subterm $f(\boldsymbol{t})$ with $\boldsymbol{t}$ in the word-product $\mathcal{T}(X)^* \pi_1 \mathcal{T}(X)^* \dots \mathcal{T}(X)^* \pi_n \mathcal{T}(X)^*$. We equip $\mathcal{T}(X)$ with the* tree topology, *defined as the coarsest one that makes every simple tree expression open in $\mathcal{T}(X)$.*

The specialization quasi-ordering of $\mathcal{T}(X)$ is the usual tree embedding *quasi-ordering $\preceq_\leq$, defined inductively by $s = f(\boldsymbol{s}) \preceq_\leq t = g(\boldsymbol{t})$ iff either $s \preceq_\leq t_j$ for some j, $1 \leq j \leq n$ (where $\boldsymbol{t} = t_1 t_2 \dots t_n$), or $f \leq g$ and $\boldsymbol{s} \preceq^*_\leq \boldsymbol{t}$.*

If X is Noetherian, then so is $\mathcal{T}(X)$.

Simple tree expressions are best explained as tests. A simple tree expression $\pi ::= \Diamond U(\pi_1 \mid \dots \mid \pi_n)$ is, syntactically, just a finite tree whose root is labeled U and with subtrees π_1, ..., π_n. Then a term t passes the test π iff it has an embedded term of the same shape as π and whose symbol functions f are all in the opens U labeling the corresponding nodes of π. E.g., whenever $f \in U$, $a \in V$, $b \in W$, $t = g(h(f(g(a, c, c), b), h(g(c))))$ is in $\Diamond U(\Diamond V() \mid \Diamond W())$, because it embeds the term $f(a, b)$, and $f \in U$, $a \in V$, $b \in W$.

We have already dealt with trees, in the special case of *ranked* terms on a *finite* space X in [13, Definition 4.3, Theorem 4.4]. However, these were flawed: the tree-products defined there are irreducible, but not closed. The characterization of irreducible closed subsets of $\mathcal{T}(X)$ is in fact significantly more complicated than for words, although they are still a form of regular expression. This will be published elsewhere.

By the way, I am, at the time I write this, discontent with the above proofs of Theorem 2 and Theorem 3, as they are arguably long and complex. I have found much

simpler proofs, which escape the need for characterizing the irreducible closed subsets, and are in fact closer to Nash-Williams celebrated minimal bad sequence argument [31]. This, too, will be published elsewhere.

We have already mentioned that some Noetherian spaces did not arise from wqos. Theorem 1 makes it easy to show that Σ^* with the upper topology of the prefix ordering (where Σ is finite, with the discrete topology) is Noetherian. Consider indeed $\Sigma^* \cup \{\top\}$, where $\top$ is a new elements, and posit that $w \leq^{\text{pref}} \top$ for every $w \in \Sigma^*$. Equip $\Sigma^* \cup \{\top\}$ with the upper topology of $\leq^{\text{pref}}$. The purpose of adding $\top$ is to enforce property T. Property W is obvious, as well as well-foundedness. So $\Sigma^* \cup \{\top\}$ is sober Noetherian. One now concludes using the easy result that any subspace of a Noetherian space is Noetherian.

One can generalize this to cases where we replace Σ by an arbitrary Noetherian space X, defining an adequate *prefix topology* on X^*. We omit this here. We write $X^{*,\text{pref}}$ the resulting space. We return to $\Sigma^{*,\text{pref}}$ in Section 5.

Let us summarize these results by the grammar of Figure 2: every space D shown there is Noetherian. We have not yet dealt with the case of polynomials; we shall touch upon them in Section 6. The constructions marked with a star are those that have no equivalent in the theory of well-quasi-orderings.

$D ::= A$		(finite, Alexandroff topology of some quasi-ordering $\leq$)	
$\mid$	$\mathbb{N}$	(Alexandroff topology of the natural ordering $\leq$)	
$\mid$	$\mathbb{C}^k$	(with the Zariski topology, see Section 6)	$*$
$\mid$	$\mathrm{Spec}(R)$	(with the Zariski topology, R a Noetherian ring, Section 6)	$*$
$\mid$	$D_1 \times D_2 \times \ldots \times D_n$	(with the product topology)	
$\mid$	$D_1 + D_2 + \ldots + D_n$	(disjoint union)	
$\mid$	D^*	(with the subword topology, Theorem 2)	
$\mid$	$D^{\circledast}$	(with the submultiset topology)	
$\mid$	$\mathcal{T}(D)$	(with the tree topology, Theorem 3)	
$\mid$	$D^{*,\text{pref}}$	(with the prefix topology)	$*$
$\mid$	$\mathcal{H}(D)$	(with the upper topology of $\subseteq$)	$*$
$\mid$	$\mathbb{P}(D)$	(with the lower Vietoris topology)	$*$
$\mid$	$\mathcal{S}(D)$	(with the lower Vietoris topology)	$*$

Fig. 2. An algebra of Noetherian datatypes

4 Effective TopWSTS

It is easy to extend the notion of WSTS to the topological case. Say that a *topological WSTS* (topWSTS) is a pair (X, δ), where X, the *state space*, is Noetherian, and δ, the *transition relation*, is lower semi-continuous. The former is the topological analogue of a wqo, and the latter generalizes strong monotonicity (1). Formally, δ is *lower semi-continuous* iff $\mathrm{Pre}^{\exists}\delta(V) = \{x \in X \mid \exists y \in V \cdot x\ \delta\ y\}$ is open whenever V is.

Modulo a few assumptions on effectiveness, one can then compute $\mathrm{Pre}^{\exists *}\delta(V)$ for any open V: since X is Noetherian, the sequence of opens $V_0 = V$, $V_{i+1} = V_i \cup \mathrm{Pre}^{\exists}\delta(V_i)$ eventually stabilizes. So we can decide, given V and $x \in X$, whether some element in V is reachable from the state x: just test whether $x \in \mathrm{Pre}^{\exists *}\delta(V)$. This is a general form of the standard backward algorithm for WSTS.

Let us make the effectiveness assumptions explicit. We need codes for opens, and ways of computing $\mathrm{Pre}^{\exists}\delta$. The following definition is inspired from Smyth [35] and Taylor [36, Definition 1.15], taking into account simplifications due to Noetherianness.

Definition 2 (Computably Noetherian Basis). *Let X be a Noetherian space. A* computably Noetherian basis *(resp.,* subbasis*) on X is a tuple $(N, \mathcal{O}[\![\_]\!], 0, 1, +, \prec\!\!\!\prec)$ (resp., $(N, \mathcal{O}[\![\_]\!], 0, 1, +, \star, \prec\!\!\!\prec)$) where:*

- *N is a recursively enumerable set of so-called* codes*,*
- *$\mathcal{O}[\![\_]\!] : N \to \mathcal{O}(X)$ is a surjective map, $\mathcal{O}[\![0]\!] = \emptyset$, $\mathcal{O}[\![1]\!] = X$, $\mathcal{O}[\![u+v]\!] = \mathcal{O}[\![u]\!] \cup \mathcal{O}[\![v]\!]$ (and $\mathcal{O}[\![u \star v]\!] = \mathcal{O}[\![u]\!] \cap \mathcal{O}[\![v]\!]$ in the case of subbases);*
- *finally, $\prec\!\!\!\prec$ is a decidable relation satisfying: if $u \prec\!\!\!\prec v$ then $\mathcal{O}[\![u]\!] \subseteq \mathcal{O}[\![v]\!]$ (*soundness*), and for any family $(v_i)_{i\in I}$ of codes, there are finitely many elements $i_1, \ldots, i_k \in I$ such that $v_i \prec\!\!\!\prec v_{i_1} + \ldots + v_{i_k}$ for all $i \in I$ (*syntactic compactness*).*

If in addition $u \prec\!\!\!\prec v$ iff $\mathcal{O}[\![u]\!] \subseteq \mathcal{O}[\![v]\!]$ for all codes $u, v \in N$, then we say that $(N, \mathcal{O}[\![\_]\!], 0, 1, +, \prec\!\!\!\prec)$ is a strongly computably Noetherian *basis.*

It is important to notice that such bases describe codes for open subsets, but that we don't care to even represent points, i.e., states, themselves.

The condition $u \prec\!\!\!\prec v$ iff $\mathcal{O}[\![u]\!] \subseteq \mathcal{O}[\![v]\!]$ for all codes $u, v \in N$ trivially entails both soundness and syntactic compactness. The latter follows from the fact that the open $\bigcup_{i\in I} \mathcal{O}[\![v_i]\!]$ is compact, since X is Noetherian. It is an easy exercise to show that *all* the spaces of Figure 2 have a strongly computably Noetherian basis. E.g., for $\mathbb{N}$, the codes are $\underline{n}$, $n \in \mathbb{N}$, plus 0; we take $1 = \underline{0}$, $\mathcal{O}[\![\underline{n}]\!] = \uparrow n$. If $(N_i, \mathcal{O}[\![\_]\!]_i, 0_i, 1_i, +_i, \star_i, \prec\!\!\!\prec_i)$ are strongly computably Noetherian bases of X_i, $1 \leq i \leq n$, then $(N', \mathcal{O}'[\![\_]\!], 0', 1', +', (\star',) \prec\!\!\!\prec')$ defines one again for $X_1 \times \ldots \times X_n$, where $N' = \mathbb{P}_{\text{fin}}(N_1 \times \ldots \times N_n)$ and $\mathcal{O}'[\![u]\!] = \bigcup_{(u_1,\ldots,u_n)\in u} \mathcal{O}[\![u_1]\!] \times \ldots \times \mathcal{O}[\![u_n]\!]$. If $(N, \mathcal{O}[\![\_]\!], 0, 1, +, \star, \prec\!\!\!\prec)$ is a strongly computably Noetherian basis for X, where $N' = \mathbb{P}_{\text{fin}}(N^*)$ and for every $u' \in N'$, $\mathcal{O}'[\![u']\!]$ is the union, over each word $w = u_1 u_2 \ldots u_n$ in u', of the basic open set $\mathcal{O}'[\![w]\!] = X^*\mathcal{O}[\![u_1]\!]X^*\mathcal{O}[\![u_2]\!]X^* \ldots X^*\mathcal{O}[\![u_n]\!]X^*$.

This also works for infinite constructions such as $\mathbb{P}(X)$ or $\mathcal{H}(X)$: if $(N, \mathcal{O}[\![\_]\!], 0, 1, +, \prec\!\!\!\prec)$ is a strongly computably Noetherian basis for X, then $(N', \mathcal{O}'[\![\_]\!], 0', 1', +', \star', \prec\!\!\!\prec')$ is a strongly computably Noetherian subbasis for $\mathbb{P}(X)$, where $N' = \mathbb{P}_{\text{fin}}(N)$, and for every $u \in N'$, $\mathcal{O}'[\![u]\!] = \bigcap_{a\in u} \Diamond\mathcal{O}[\![a]\!]$ (this is X' itself when $u = \emptyset$).

One sometimes also needs a *representation of points*, which we define as some subset P of some r.e. set, with a map $X[\![\_]\!] : P \to X$, and a decidable relation ε on $P \times N$ such that $p \;\varepsilon\; u$ iff $X[\![p]\!] \in \mathcal{O}[\![u]\!]$. If X is T_0, there is always a surjective, canonical representation of points derived from a strongly computable Noetherian subbasis $(N, \mathcal{O}[\![\_]\!], 0, 1, +, \prec\!\!\!\prec)$: take P to be the subset of all codes $u \in N$ such that $\mathcal{O}[\![u]\!]$ is the complement of some set of the form $\downarrow x$, then let $X[\![u]\!] = x$. So we don't formally need another structure to represent points: any computably Noetherian basis already

cares for that. But some other representations of points may come in handy in specific cases.

Definition 3 (Effective TopWSTS). *An* effective topWSTS *is a tuple* $(X, \delta, N, \mathcal{O}[\![\_]\!], 0, 1, +, \prec\!\!\prec, R_\exists)$*, where* (X, δ) *is a topWSTS,* $(N, \mathcal{O}[\![\_]\!], 0, 1, +, \prec\!\!\prec)$ *is an effective basis on* X*,* $R_\exists : N \to N$ *is computable, and* $\mathrm{Pre}^\exists \delta(\mathcal{O}[\![u]\!]) = \mathcal{O}[\![R_\exists(u)]\!]$ *for every* $u \in N$*.*

In other words, one may compute a code of $\mathrm{Pre}^\exists \delta(U)$, given any code u of U, as $R_\exists(u)$. The following is then clear.

Proposition 1. *Let* $(X, \delta, N, \mathcal{O}[\![\_]\!], 0, 1, +, \prec\!\!\prec, R_\exists)$ *be an effective topWSTS. One can effectively compute a code of* $\mathrm{Pre}^{\exists *} \delta(U)$ *from any given code* u *of the open subset* U*.*

Assume additionally a representation $(P, X[\![\_]\!], \varepsilon)$ *of points. Given any code* p *for a point* $x \in X$*, and any code* u *for an open subset* U *of* X*, one can decide whether there is a trace* $x = x_0\ \delta\ x_1\ \delta \ldots \delta\ x_k$ *such that* $x_k \in U$*.*

One can in fact go further and model-check some infinite two-player games. We consider a *may player*, who will play along lower semi-continuous transition relations, and a *must player*, who will play along upper semi-continuous transition relations: δ is *upper semi-continuous* iff $\mathrm{Pre}^\forall \delta(F)$ is *closed* whenever F is.

Formally, one posits a finite set $L = L_{\text{must}} \cup L_{\text{may}}$ of *transition labels*, taken as the (not necessarily disjoint) union of two subsets of *must labels* and *may labels*, and calls a *topological Kripke structure* any tuple $I = (X, (\delta_\ell)_{\ell \in L}, (U_A)_{A \in \mathcal{A}})$, where X is a topological space, δ_ℓ is a binary relation on X, which is lower semi-continuous when $\ell \in L_{\text{may}}$ and upper semi-continuous when $\ell \in L_{\text{must}}$, and U_A is an open of X for every atomic formula A. An *environment* ρ maps variables ξ to opens of X, and serves to interpret formulae in some modal logic. In [19, Section 3], we defined the logic L_μ as follows. The formulae F are inductively defined as atomic formulae A, variables ξ, true $\top$, conjunction $F \wedge F'$, false $\bot$, disjunction $F \vee F'$, must-modal formulae $[\ell]F$, may-modal formulae $\langle \ell \rangle F$, and least fixed points $\mu\xi \cdot F$. The semantics of L_μ is standard: the set $I[\![F]\!]_\delta \rho$ of states $x \in X$ such that x satisfies F is in particular defined so that $I[\![\langle \ell \rangle F]\!]_\delta \rho = \mathrm{Pre}^\exists \delta_\ell(I[\![F]\!]_\delta \rho)$, $I[\![[\ell]F]\!]_\delta \rho = \mathrm{Pre}^\forall \delta_\ell(I[\![F]\!]_\delta \rho)$ (where, if F is the complement of V, $\mathrm{Pre}^\forall(V)$ is the complement of $\mathrm{Pre}^\exists(F)$), and $I[\![\mu\xi \cdot F]\!]_\delta \rho = \bigcup_{i=0}^{+\infty} U_i$, where $U_0 = \emptyset$ and $U_{i+1} = I[\![F]\!]_\delta (\rho[\xi := U_i])$. When X is Noetherian, the latter will in fact arise as a *finite* union $\bigcup_{i=0}^{n} U_i$. We define effective topological Kripke structures in the obvious way, imitating Definition 3: just require computable maps $R^\exists_\ell : N \to N$ representing δ_ℓ for each $\ell \in L_{\text{may}}$, $R^\forall_\ell : N \to N$ representing δ_ℓ for each $\ell \in L_{\text{must}}$, and codes u_A of U_A for each atomic formula A. Computing (a code for) $I[\![F]\!]_\delta \rho$ by recursion on F yields the following decision result.

Proposition 2. *Given an effective topological Kripke structure, any formula* F *of* L_μ*, and any sequence of codes* v_ξ*, one for each variable* ξ*, one can effectively compute a code of* $I[\![F]\!]_\delta \rho$*, where* ρ *is the environment mapping each* ξ *to* $\mathcal{O}[\![v_\xi]\!]$*.*

Given any representation of points, and any code for a point $x \in X$*, one can decide whether* x *satisfies* F*.*

5 Oblivious Stack Systems

Let Σ be a finite alphabet. Reachability and coverability in k-stack pushdown automata are undecidable as soon as $k \geq 2$: encode each half of the tape of a Turing machine by a stack. Here is relaxation of this model that will enjoy a decidable form of coverability.

Define *oblivious* k-stack systems just as pushdown automata, except they cannot check what letter is popped from any stack. Formally, they are automata on a finite set Q of control states, and where transitions are labeled with k-tuples $(\alpha_1, \ldots, \alpha_k)$ of actions. Each action α_i is of the form $\texttt{push}_a$, for each $a \in \Sigma$ (push a onto stack number i) `pop` (pop the top letter from stack i, if any, else block), and `skip` (leave stack i unchanged), and all actions are performed in parallel.

This defines an effective topWSTS on the state space $Q \times (\Sigma^{*,\mathrm{pref}})^k$. As we have seen, the latter is Noetherian, although its specialization ordering is certainly not wqo. So the theory of WSTS, as is, does not bring much in solving oblivious k-stack systems. However, Proposition 1 applies: one can decide whether we can reach a given open set V from any state. One observes that one can specify an open set by a finite set $\{p_1, \ldots, p_n\}$ of *forbidden patterns*. A forbidden pattern p is a tuple $(q, w_1, \ldots, w_n)$ where $q \in Q$, and each w_i is either a word in Σ^* or the special symbol $\top$. Such a pattern is *violated* in exactly those states $(q, w'_1, \ldots, w'_n)$ such that for each i such that $w_i \neq \top$, w'_i is a prefix of w_i. It is *satisfied* otherwise. Then $\{p_1, \ldots, p_n\}$ denotes the open subset of all states that satisfy every p_i, $1 \leq i \leq n$. It follows:

Theorem 4. *Given an oblivious k-stack system, any initial configuration and any finite set of forbidden patterns, one can decide whether there is a configuration that is reachable from the initial configuration and satisfies all forbidden patterns.*

In particular, *control-state reachability*, which asks whether one can reach some state $(q, w_1, \ldots, w_n)$, for some fixed q, and arbitrary $w_1, \ldots, w_n$, is decidable for oblivious k-stack systems: take all forbidden patterns of the form $(q', \top, \ldots, \top)$, $q' \in Q \setminus \{q\}$. This much, however, was decidable by WSTS techniques: as S. Schmitz rightly observed, one can reduce this to Petri net control-state reachability by keeping only the lengths of stacks. Theorem 4 is more general, as it allows one to test the contents of the stacks, and comes for free from the theory of topWSTS.

The reader may see a similarity between k-stack pushdown automata and the concurrent pushdown systems of Qadeer and Rehof [32]. However, the latter must push and pop on one stack at a time only. Pushdown automata may require one to *synchronize* push transitions taken on two or more stacks. I.e., if the only transitions available from control state q are labeled $(\texttt{push}_a, \texttt{push}_a, \texttt{skip}, \ldots, \texttt{skip})$ and $(\texttt{push}_b, \texttt{push}_b, \texttt{skip}, \ldots, \texttt{skip})$, then this forces one to push the same letter, a or b, onto the first two stacks when exiting q.

6 Polynomial Games

Let $\mathbb{C}$ be the field of complex numbers, and $k \in \mathbb{N}$. Let R be the ring $\mathbb{Q}[X_1, \ldots, X_k]$ of all polynomials on k variables with coefficients in $\mathbb{Q}$. The *Zariski topology* on $\mathbb{C}^k$ is the one whose opens are $O_I = \{\boldsymbol{x} \in \mathbb{C}^k \mid P(\boldsymbol{x}) \neq 0 \text{ for some } P \in I\}$, where I ranges

over the ideals of R. I.e., its closed subsets are the *algebraic varieties* $F_I = \{\boldsymbol{x} \in \mathbb{C}^k \mid P(\boldsymbol{x}) = 0 \text{ for every } P \in I\}$. This is a much coarser topology that the usual metric topology on $\mathbb{C}^k$, and is always Noetherian.

There is an obvious computably Noetherian subbasis (*not* strongly so) from computable algebraic geometry. The set N of codes is the collection of *Gröbner bases* [9, Section 11], which are finite sets of polynomials $u = \{P_1, \ldots, P_n\}$ over $\mathbb{Q}$, normalized with respect to a form of completion procedure due to Buchberger. Given a so-called admissible ordering of monomials, i.e., a total well-founded ordering $\geq$ on monomials such that $m_1 \geq m_2$ implies that $mm_1 \geq mm_2$ for all monomials m, every non-zero polynomial P can be written as $am + P'$, where $a \in K$, m is the largest monomial of P in $\geq$, and P' only involves smaller monomials. P can then be interpreted as a rewrite rule $m \to -\frac{1}{a}P'$ on polynomials, E.g., if $P = X^2Y - 4X + Y^2$, with X^2Y as leading monomial, one can rewrite X^5Y^2 ($= X^2Y.X^3Y$) to $4X^4Y - X^3Y^3$; the latter ($= X^2Y.(4X^2) - X^3Y^3$) again rewrites, using P, to $-X^3Y^3 + 16X^3 - 4X^2Y^2$, then to $-4X^2Y^2 + XY^4 + 16X^3 - 4X^2Y^2 = 16X^3 - 8X^2Y^2 + XY^4$, and finally to $16X^3 - 32XY + XY^4 + 8Y^3$. Notice that, evaluated on any zero of P, all the polynomials in the rewrite sequence have the same value; e.g., 0 when $X = Y = 0$, or $\frac{9-\sqrt{17}}{2}$ when $X = 1$, $Y = \frac{-1+\sqrt{17}}{2}$.

A Gröbner basis for an ideal I is a finite family w of polynomials such that $I = (w)$ and that is confluent (and necessarily terminating) when interpreted as a rewrite system. Buchberger's algorithm converts any finite set v of polynomials to a Gröbner basis w of (v).

Let then $\mathcal{O}[\![u]\!] = O_{(u)}$, where $(u) = (P_1, \ldots, P_n)$ is the ideal of all linear combinations of P_1, ..., P_n with coefficients in R. One can always compute a Gröbner base for an ideal $I = (P_1, \ldots, P_n)$, given P_1, ..., P_n, by Buchberger's algorithm. The code 0 is then $\{0\}$, 1 is defined as $\{1\}$, $u + v$ is a Gröbner base for $u \cup v$. One can also define $u \star v$ to be a code for $\mathcal{O}[\![u]\!] \cap \mathcal{O}[\![v]\!]$, and compute it in at least two ways [27, Section 4.3]. The simplest algorithm [8, Proposition 4.3.9] consists in computing a Gröbner basis of $I = (YP_1, YP_2, \ldots, YP_m, (1-Y)Q_1, (1-Y)Q_2, \ldots, (1-Y)Q_n)$, where $u = \{P_1, P_2, \ldots, P_m\}$ and $v = \{Q_1, Q_2, \ldots, Q_n\}$ and Y is a fresh variable, and to define $u \star v$ as a Gröbner basis for the *elimination ideal* $\exists Y \cdot I$, defined as those polynomials in I where Y does not occur [8, Theorem 4.3.6]. Given any polynomial P and any Gröbner basis u, one can test whether $P \in (u)$ by a process akin to rewriting: each polynomial in u works as a rewrite rule, and $P \in (u)$ iff the (unique) normal form of P with respect to this rewrite system is 0. One can then test whether $u \preccurlyeq v$ by checking whether, for each $P \in u$, P is in (v). It turns out that $u \preccurlyeq v$ is *not* equivalent to $\mathcal{O}[\![u]\!] \subseteq \mathcal{O}[\![v]\!]$: take $u = \{X\}$, $v = \{X^2\}$, then $u \notin (v)$, although $\mathcal{O}[\![u]\!] = \mathcal{O}[\![v]\!]$. But soundness is obvious, and syntactic compactness (Definition 2) follows since R is a Noetherian ring. We mention in passing that there is also a *strongly* computably Noetherian subbasis, where $u \preccurlyeq v$ iff (u) is included in the radical of (v), and this can be decided using the *Rabinowitch trick* [33].

As a representation of points, we take those u such that (u) is a prime ideal. This is in fact the canonical representation. It contains at least all rational points $(q_1, \ldots, q_k) \in \mathbb{Q}^k$, represented as the Gröbner basis $(X_1 - q_1, \ldots, X_k - q_k)$, but also many more.

One gets natural topWSTS from *polynomial programs*. These are finite automata, on some finite set Q of control states, where transitions are labeled with guards and assignments on k complex-valued variables. The guards g are finite sets $\{P_1, \ldots, P_m\}$ of polynomials in R, interpreted as disjunctions of disequalities $P_1 \neq 0 \vee \ldots \vee P_m \neq 0$. If the guard is satisfied, then the transition can be taken, and the action is triggered. The allowed actions a are parallel assignments $\boldsymbol{x} := P_1(\boldsymbol{x}), \ldots, P_k(\boldsymbol{x})$, where $\boldsymbol{x}$ is the vector of all k variables, and with the obvious semantics. Each P_i is either a polynomial in R, or the special symbol ?, meaning any element of $\mathbb{C}$.

This defines a transition relation δ on $Q \times \mathbb{C}^k$, which happens to be lower semi-continuous, and in fact computably so. As set N' of codes for opens of the state space $Q \times \mathbb{C}^k$, we use $\mathbb{P}_{\text{fin}}(Q \times N)$, and define $\mathcal{O}'[\![u']\!] = \bigcup_{(q,u)\in N'} \{q\} \times \mathcal{O}[\![u]\!]$. Then, $\mathrm{Pre}^{\exists}\delta(\mathcal{O}'[\![u']\!]) = \bigcup_{(q',u)\in u'} \{q\} \times \mathcal{O}[\![g \star \{P[X_i := P_i]_{i\in I_{act}} \mid P \in \forall X_{i_1}, \ldots, X_{i_m} \cdot u\}]\!]$, where a is $\boldsymbol{x} := P_1(\boldsymbol{x}), \ldots, P_k(\boldsymbol{x})$, and i_1, …, i_m are those indices i where P_i is ?, and I_{act} are the others. $P[X_i := P_i]_{i\in I_{act}}$ is parallel substitution of P_i for X_i in P for all $i \in I_{act}$. The $\forall$ operator is defined, and shown to be computable, in [29, Lemma 4].

$q \xrightarrow{g,a} q'$

The polynomial programs above are exactly those of Müller-Olm and Seidl [29]. Proposition 1 then immediately applies. In particular, one can decide whether one can reach some configuration $(q', \boldsymbol{x})$ such that $P_1(\boldsymbol{x}) \neq 0$ or … or $P_m(\boldsymbol{x}) \neq 0$ for some state q' and polynomials $P_1, \ldots, P_m$, from a given configuration, in a given polynomial program. This is a bit more than the polynomial constants problem of [29]. For example, we may want to check whether at q' we always have $Y = X^2 + 2$ and $X^2 + Y^2 = Z^2$, and for this we let $P_1 = Y - X^2 - 2$, $P_2 = Z^2 - X^2 - Y^2$, $m = 2$. Figure 3 is a rendition of an example by Müller-Olm and Seidl, in C-like syntax. The conditions (shown as '?') at lines 1 and 3 are abstracted away: `if` and `while` statements are to be read as non-deterministic choices. One may check that it is always the case that `x` is 0 at line 4. This is a polynomial program with control states 1 through 4, the variables are $X_1 =$ `x`, $X_2 =$ `y`, all the guards are trivial (i.e., the empty set of polynomials), and the actions should be clear; e.g., the only action from state 2 to state 3 is the pair of polynomials $(X_1X_2 - 6, 0)$.

```
1.  if (?) { x = 2; y = 3; } else { x = 3; y = 2; }
2.  x = x * y - 6; y = 0;
3.  while (?) { x = x + 1; y = y - 1; };
3'. x = x^2 + x * y;
4.  return;
```

Fig. 3. Müller-Olm and Seidl's example

Polynomial Games. We can again go further. Define *polynomial games* as a topological Kripke structure where, for each may transition $\ell \in L_{\text{may}}$, δ_ℓ is specified by guards and actions as above. For each must transition $\ell \in L_{\text{must}}$, we specify δ_ℓ by giving ourselves a finite set A_ℓ of triples $(q, q', \alpha) \in Q \times Q \times \mathbb{Q}[X_1, \ldots, X_k, X'_1, \ldots, X'_k]$, and defin-

ing $(q, \boldsymbol{x})\ \delta_\ell(q', \boldsymbol{x}')$ iff there is a triple $(q, q', \alpha) \in A_\ell$ such that $\alpha(\boldsymbol{x}, \boldsymbol{x}') = 0$. So the must player can, in particular, compute polynomial expressions of $\boldsymbol{x}$, test polynomials against 0, and solve polynomial equations. It is easy to see that δ_ℓ is then upper semi-continuous, as $\mathrm{Pre}^\exists \delta_\ell(\{q'\} \times F_{(u)}) = \bigcup_{(q,q',\alpha)\in A_\ell} \{q\} \times F_{\exists X'_1 \cdot \ldots \cdot \exists X'_k \cdot (\alpha \cup \{P[X_i := X'_i]_{i=1}^k | P \in u\})}$. By Proposition 2:

Theorem 5. *The model-checking problem for $\mathcal{L}_\mu$ formulas on polynomial games is decidable.*

We do not know whether the added expressive power of polynomial games, compared to the simpler polynomial programs, will be of any use in verification, however the theorem stands.

Lossy Concurrent Polynomial Programs. All the above can be done without any recourse to topology, and requires one only to work with polynomial ideals. However, the topological approach is modular. If one should someday need to decide games that would involve state spaces such as $\mathbb{N}^m \times \mathbb{C}^k$, or $\mathbb{P}(\mathbb{C}^k)$, the theory of Noetherian spaces would apply right out of the box. Here is a trivial case.

Consider a network of polynomial programs communicating through specific FIFO channels. We shall call such networks *concurrent polynomial programs*. We shall assume these channels to be *lossy*, i.e., messages may disappear from the channels, nondeterministically, at any time. We shall also assume that the messages are taken from some fixed finite set Σ, i.e., the channels are only used for signaling, not for transmitting any actual value. This imitates the behavior of lossy channel systems [3], which are a special case (with no variable from $\mathbb{C}$) of our lossy concurrent polynomial programs. Lossiness is interesting, if only because the non-lossy variants are undecidable [10].

For simplicity, we shall only consider two programs A and B communicating through one FIFO channel from A to B. Dealing with more programs and more channels presents no difficulty, other than notational.

The messages sent over the channel are control signals from Σ. So the data type of the possible contents of the channel is Σ^*, with the subword topology (alternatively, the Alexandroff topology of the usual embedding quasi-ordering $\leq^*$, where $\leq$ is equality on Σ). Let Q_A be the finite set of control states of A, Q_B that of B. Let $\boldsymbol{X} = X_1, \ldots, X_m$ be the vector of the numerical variables of A, $\boldsymbol{Y} = Y_1, \ldots, Y_n$ those of B. The configurations are tuples $(q_A, \boldsymbol{X}, q_B, \boldsymbol{Y}, w)$, where $(q_A, \boldsymbol{X})$ is a configuration of the polynomial program A, $(q_B, \boldsymbol{Y})$ is a configuration of B, and $w \in \Sigma^*$ is the contents of the channel. Compared to the non-concurrent case, the guards and the actions of A (resp., B) can only deal with variables from $\boldsymbol{X}$ (resp., $\boldsymbol{Y}$), except for two new families of actions $\mathtt{recv}_a$ (for B) and $\mathtt{send}_a$ (for A), where a is a constant in Σ.

Formally, given any A-transition from q_A to q'_A with guard g and action $\mathtt{send}_a$, $a \in \Sigma + \mathbb{C}^k$, we define δ so that $(q_A, \boldsymbol{X}, q_B, \boldsymbol{Y}, w)\ \delta\ (q'_A, \boldsymbol{X}, q_B, \boldsymbol{Y}, aw)$ provided g is satisfied (add a in front of w), while the semantics of a $\mathtt{recv}_a$ action, $a \in \Sigma$, from q_B to q'_B with guard g, is given by $(q_A, \boldsymbol{X}, q_B, \boldsymbol{Y}, w_1 a w)\ \delta\ (q_A, \boldsymbol{X}, q'_B, \boldsymbol{Y}, w)$ if g is satisfied (i.e., drop enough letters from the FIFO channel until we reach an a, and pop it). It is an easy exercise to show that this is lower semi-continuous, and computably so. (We could also add transitions that drop letters from the channel, as in lossy channel systems, but this is not needed for the rest of our treatment.)

The opens are finite unions of sets of the form $\{(q_A, \boldsymbol{x}, q_B, \boldsymbol{x}, w) \mid (\boldsymbol{x}, \boldsymbol{y}) \in O_I, w \in \Sigma^* a_1 \Sigma^* \dots \Sigma^* a_q \Sigma^*\}$, where q_A, q_B are fixed control states, $I = (p_1, \dots, p_\ell)$ is a fixed polynomial ideal over $\mathbb{Q}[\boldsymbol{X}, \boldsymbol{Y}]$, and a_1, …, a_q are fixed letters from Σ. In other words, such an open subset is specified by a *forbidden pattern*: a state satisfies the forbidden pattern iff its A is in control state q_A, B is in control state q_B, $p_i(\boldsymbol{x}, \boldsymbol{y}) \neq 0$ for some i, $1 \leq i \leq \ell$, and $a_1 a_2 \dots a_q$ is a subword of the contents w of the channel.

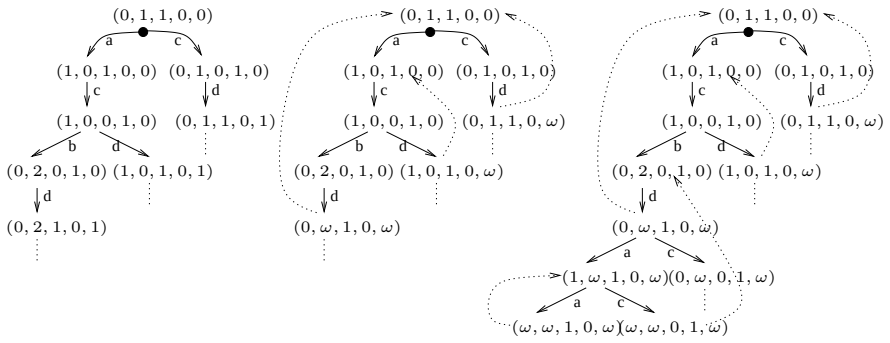
Theorem 6. *Given a lossy concurrent polynomial program, an initial configuration where the values of the variables are given as rational numbers, and a finite set of forbidden patterns, one can decide whether there is a configuration reachable from the initial configuration and that satisfies all forbidden patterns.*

In particular, control-state reachability (can one reach a configuration where A would be in state q_A and B in state q_B?) is decidable.

Algebraic geometry. To end this section, we only mention that $\mathbb{C}^k$ is considered as a special case in algebraic geometry. It turns out that the sobrification $\mathcal{S}(\mathbb{C}^k)$ coincides with $\mathrm{Spec}(\mathbb{Q}[X_1, X_2, \dots, X_k])$, the *spectrum* of the (Noetherian) ring $\mathbb{Q}[X_1, X_2, \dots, X_k]$. The spectrum $\mathrm{Spec}(R)$ of a ring R is the set of all prime ideals of R, and comes with the Zariski topology, whose closed subsets are $F_I = \{p \in \mathrm{Spec}(R) \mid I \subseteq p\}$, where I ranges over the ideals of R. It is well-known that $\mathrm{Spec}(R)$ is a Noetherian space whenever R is a Noetherian ring, see [20, chapitre premier, Section 1.1]. This provides yet another construction of Noetherian spaces, although we currently have no application in computer science that would require the added generality.

7 Completions, and Complete WSTS

The algorithm of Proposition 1 works backwards, by computing iterated sets of predecessors. The Karp-Miller algorithm [24] works *forwards* instead, but only applies to Petri nets. Forward procedures are useful, e.g., to decide boundedness, see [14].



(*a*) Unfolding (beginning) (*b*) Taking some shortcuts (*c*) Finishing some branches

Fig. 4. Running the Karp-Miller procedure on Figure 1

Consider for example the Petri net of Figure 1, and consider it as a transition system over $\mathbb{N}^5$. The initial state is $(0,1,1,0,0)$, and there are four transitions a, b, c, and d. One can then unfold all possible runs of the net in a tree (see Figure 4, (a)). Here, from the initial state, one can trigger transitions a or c, leading to states $(1,0,1,0,0)$ and $(0,1,0,1,0)$ respectively. From the latter we can only trigger d, leading to $(0,1,1,0,1)$, and so on. Doing so would yield an infinite tree.

The Karp-Miller construction builds a finite tree by taking some shortcuts, and abstracting away the values of components of the states that may become unbounded. E.g., in Figure 1, we realize that firing c then d leads to a state $(0,1,1,0,1)$ where the first four components are the same as in the initial state $(0,1,1,0,0)$, but the fifth is larger. Iterating this c-d sequence would lead to a state $(0,1,1,0,N)$ with N arbitrary large, and we abstract this away by *replacing* the state $(0,1,1,0,1)$ by $(0,1,1,0,\omega)$, where ω denotes any, arbitrarily large, number. This also happens along the other two branches shown in Figure 4, (a). The dotted arrows going up in Figure 4, (b), indicate which state we can go back to in order to perform such iterations.

One can see a tuple in $\mathbb{N}_\omega^5$ (where $\mathbb{N}_\omega = \mathbb{N} \uplus \{\omega\}$) such as $(0,1,1,0,\omega)$ as meaning, in particular, that there are infinitely many tokens in place x_5. While this is not quite correct, it certainly gives some further intuitions. In particular, one can continue to simulate the execution of the Petri net from such extended states. The result, apart from some irrelevant parts, is shown in Figure 4, (c). This is the *coverability tree* of the Petri net. (The dotted arrows are not part of this—the coverability *graph* would include them. We also glossed over a few details, notably that the Karp-Miller construction does not proceed to simulate the execution of the Petri net from an extended state that is identical to a state further up the same branching.)

The reason why the resulting tree is twofold. First, $\mathbb{N}_\omega^k$ is wqo. This implies that, along any infinite branch, we must eventually find an extended state that is componentwise larger than another one higher in the tree, giving us an opportunity to take a shortcut. We call this a *progress* property: in any infinite run, we will eventually take a shortcut, subsuming infinitely many iterations.

Second, taking a shortcut adds an ω component to a tuple, and ω components never disappear further down the branch: so any branch must in fact be finite. It follows from König's Lemma that the tree itself is finite, and is built in finite time.

The Karp-Miller construction gives more information about the Petri net than the standard backward algorithm. Letting $A \subseteq \mathbb{N}_\omega^k$ be the set of all extended states labeling the nodes of the tree, one sees that $\mathbb{N}^k \cap {\downarrow} A$ is exactly the *cover* of the Petri net, i.e., the downward closure ${\downarrow}\mathrm{Post}^*\delta(x)$ of the set $\mathrm{Post}^*\delta(x)$ of states y that are reachable from the initial state x by the transition relation δ. In particular, one can decide coverability, by checking whether $y \in {\downarrow}\mathrm{Post}^*\delta(x)$. One can also decide *boundedness*, i.e., test whether $\mathrm{Post}^*\delta(x)$? is finite (check whether any ω component occur anywhere in the coverability tree), and *place-boundedness*, i.e., test whether there is a bound on the number of tokens that can be in any given place. In the example above, and after simplification, the cover is $\mathbb{N}^5 \cap {\downarrow}\{(\omega,\omega,0,1,\omega),(\omega,\omega,1,0,\omega)\}$: the bounded places are x_3 and x_4.

The Karp-Miller construction is fine, but only works on Petri nets. There cannot be any similar, terminating procedure for general WSTS, since this would decide bound-

edness again. But boundedness is already undecidable on lossy channel systems [10] and on reset Petri nets [12].

Even if we drop the requirement for termination, finding a procedure that would compute the cover of a general WSTS (when it terminates) remained elusive for some time. Some important special cases could be handled in the literature, e.g., a large class of affine counter systems generalizing reset/transfer Petri nets [15], or lossy channel systems [2], but a general theory of covers, and of forward procedures à la Karp-Miller for general WSTS was missing. This is what we solved, with A. Finkel, in two recent papers [13, 14].

This involved two tasks: (i) first, much as we needed extended states in $\mathbb{N}_\omega^k$ in the Karp-Miller procedure, we should work in an adequate *completion* $\widehat{X}$ of the state space X; (ii) then, we should understand what a Karp-Miller-like procedure actually computes. We actually cheated here, since we have already given an answer to (ii): the Karp-Miller procedure computes (among other things) the cover of the Petri net.

Completions. In (i), by adequate we mean that X should embed into $\widehat{X}$, in such a way that every closed subset D of X should be representable by a *finite* subset of $\widehat{X}$. (In [13], we wanted to this for every downward closed, not just closed, D. However, if X has the Alexandroff topology of a wqo, this is the same thing.) Formally, D should be the set of those points in X that are below finitely many points in $\widehat{X}$: $D = X \cap {\downarrow_{\widehat{X}}} E$, E finite. We write ${\downarrow_{\widehat{X}}}$ to stress the fact that the downward closure is to be taken in $\widehat{X}$, i.e., E is a subset of $\widehat{X}$, not X. Typically, if $X = \mathbb{N}^k$, then $\widehat{X}$ should be $\mathbb{N}_\omega^k$, and for example, $D = \{(m, n, p) \in \mathbb{N}^3 \mid m + n \leq 3\}$ is representable as $\mathbb{N}^3 \cap {\downarrow_{\widehat{\mathbb{N}^3}}}\{(0, 3, \omega), (1, 2, \omega), (2, 1, \omega), (3, 0, \omega)\}$. It turns out that the sobrification $\mathcal{S}(X)$ is exactly what we need here, as advocated in [13, Proposition 4.2]:

Proposition 3. *Let X be a Noetherian space. Every closed subset F of X is a finite union of irreducible closed subsets $C_1, \ldots, C_m$.*

So let the completion $\widehat{X}$ be $\mathcal{S}(X)$. Proposition 3 states that, modulo the canonical identification of $x \in X$ with $\downarrow x \in \mathcal{S}(X)$, F is $X \cap {\downarrow_{\widehat{X}}}\{C_1, \ldots, C_m\}$. We have stressed the subcase where X was wqo (and equipped with its Alexandroff topology) in [13], and this handles 7 of the 13 constructions in Figure 2. We would like to note that Noetherian spaces X allow us to consider more kinds of state spaces, while ensuring that each closed subset of X is finitely representable, in a canonical way. Moreover, these representations are effective, too.

One might wonder whether there would be other adequate completions $\widehat{X}$. There are, indeed, however $\mathcal{S}(X)$ is canonical in a sense. Adapting Geeraerts *et al.* slightly [17], call *weak adequate domain of limits*, or *WADL*, over the Noetherian space X any space $\widehat{X}$ in which X embeds, and such that the closed (downward closed when X is wqo) subsets of X are exactly the subsets representable as $X \cap {\downarrow_{\widehat{X}}} E$ for some finite subset E of $\widehat{X}$. It is an easy consequence of Theorem 1 that $\mathcal{S}(X)$ is the *smallest* WADL, and $\mathcal{H}(X)$ is the *largest* WADL: up to isomorphism, any WADL $\widehat{X}$ must be such that $\mathcal{S}(X) \subseteq \widehat{X} \subseteq \mathcal{H}(X)$.

It is then natural to ask whether $\widehat{X} = \mathcal{S}(X)$ is effectively presented, in the sense that we have codes for all elements of $\mathcal{S}(X)$ and that the ordering (i.e., $\subseteq$) on $\mathcal{S}(X)$

is decidable. It turns out that the answer is positive for *all* the datatypes of Figure 2. E.g., given codes for elements of $\widehat{X_1}$, $\widehat{X_2}$, the codes for elements of $\widehat{X_1 \times X_2}$ are just pairs of codes (x_1, x_2) for elements of $\widehat{X_1}$, $\widehat{X_2}$. Given codes for elements of $\widehat{X}$, the codes for elements of $\widehat{X^*}$ are the word-products we mentioned in Section 3. It might seem surprising that we could do this even for the infinite powerset $\mathbb{P}(X)$. Notice that $\widehat{\mathbb{P}(X)} = \mathcal{H}(X)$, up to isomorphism, and that every element of $\mathcal{H}(X)$ can be written as $C_1 \cup \ldots \cup C_n$ for finitely many elements $C_1, \ldots, C_n$ of $\widehat{X}$ by Proposition 3. So take as codes for elements of $\widehat{\mathbb{P}(X)}$ the *finite* sets E of codes of elements $C_1, \ldots, C_n$ of $\widehat{X}$.

Clovers. Point (ii) was clarified, among other things, in [14]: Karp-Miller-like procedures compute the *clover* of a state $x \in X$ in a WSTS $\mathfrak{X}$, and this is a finite representative $\{C_1, \ldots, C_m\}$, as defined above, of the topological *closure* (in $\widehat{X}$) of the set of points reachable from x. The clover *always* exists, by Proposition 3. It may fail to be computable: if it were, it would allow us to decide boundedness for reset Petri nets or for lossy channel systems, which are undecidable.

While we investigated this for WSTS, it actually works for any *topological* WSTS. We only need to extend the transition relation $\delta \subseteq X \times X$ to one, $\mathcal{S}\delta$, on $\widehat{X} \times \widehat{X}$. The canonical candidate is such that $C \; \mathcal{S}\delta \; C'$ iff C' is included in the closure of $\mathrm{Post}\delta(C) = \{y \in X \mid \exists x \in C \cdot x \; \delta \; y\}$; and this is representable as a *finitely branching* (because of Proposition 3 again) relation $\widehat{\delta}$. E.g., the minimal such relation is such that $C \; \widehat{\delta} \; C'$ iff C' is maximal such that $C \; \mathcal{S}\delta \; C'$. We then get a completed topWSTS $\widehat{\mathfrak{X}}$.

To do anything with this, we must assume that $\widehat{\delta}$ is effective, and in fact more. We explored this in [14], in the case where $\widehat{X}$ is wqo, and $\widehat{\mathfrak{X}}$ is functional (i.e., $C \; \widehat{\delta} \; C'$ iff $C' = g_i(C)$ for some i, $1 \leq i \leq n$, where $g_1, \ldots, g_n$ is a fixed collection of partial continuous maps from $\widehat{X}$ to $\widehat{X}$) and ∞-effective (see below), and obtained a simple procedure **Clover** that computes the clover of any state $C \in \widehat{X}$ (in particular, any $x \in X$) whenever it terminates.

The role of the completion $\widehat{X}$ is again manifest in that **Clover** needs to *lub-accelerate* some infinite sequences of states obtained in a regular fashion as $C_0 < g(C_0) \leq g^2(C_0) \leq \ldots \leq g^n(C_0) \leq \ldots$ by applying one functional transition $g : \widehat{X} \to \widehat{X}$, replacing the sequence by its least upper bound $g^\infty(C_0)$ (which exists: recall that every sober space is a dcpo in its specialization quasi-ordering). This is what we called taking shortcuts until now. If $C_0 \not< g(C_0)$, then define $g^\infty(C_0)$ as just $g(C_0)$. $\widehat{\mathfrak{X}}$ is ∞-*effective* iff g^∞ is computable.

Here is the procedure. $\mathrm{Max}\, A$ denotes the set of all maximal elements of $A \in \mathbb{P}_{\text{fin}}(\widehat{X})$. The procedure takes an initial extended state $s_0 \in \widehat{X}$, and, if it terminates, returns a finite set $\mathrm{Max}\, A$ (the *clover* of s_0) such that $\downarrow_{\widehat{X}} \mathrm{Max}\, A$ is the closure of the cover of the WSTS.

Procedure Clover(s_0) :
1. $A \leftarrow \{s_0\}$;
2. **while** $\mathrm{Post}(\mathcal{S}\delta)(A) \not\leq^\flat A$ **do**
   (a) Choose fairly $(g, C) \in \{g_1, \ldots, g_n\}^* \times A$ such that $C \in \mathrm{dom}\, g$;
   (b) $A \leftarrow A \cup \{g^\infty(a)\}$;
3. **return** $\mathrm{Max}\, A$;

The elements g chosen at line (a) are chosen from $\{g_1, \ldots, g_n\}^*$, the set of compositions $g_{i_1} \circ g_{i_2} \circ \ldots \circ g_{i_k}$ of functions from $\{g_1, \ldots, g_n\}$. A typical implementation of **Clover** would build a tree, as in the Karp-Miller construction. In fact, a tree is a simple

way to ensure that the choice of (g, C) at line (a) is *fair*, i.e., no pair is ignored infinitely long on any infinite branch. Concretely, we would build a tree extending downwards. At each step, A is given by the set of extended states written at the nodes of the current tree. One picks (g, C) as in line (a) by picking a transition g_i to apply from a yet unexplored state C' (at a leaf), and considering all states C higher in the branch (the path from C to C' being given by transitions, say, $g_{i_k}, g_{i_{k-1}}, \ldots, g_{i_2}$), letting $g = g_i \circ g_{i_2} \circ \ldots \circ g_{i_k}$.

The **Clover** procedure extends straightforwardly to topological WSTS, provided they are functional (here, each g_i needs to be continuous). It is however unclear how to dispense with the requirement that it be functional. Moreover, the nice characterization that **Clover** terminates exactly on those systems that are *clover-flattable* [14, Theorem 3] seems to require the fact that X is ω^2-wqo, not even just wqo, for deep reasons.

References

1. P. A. Abdulla, K. Čerāns, B. Jonsson, and T. Yih-Kuen. Algorithmic analysis of programs with well quasi-ordered domains. *Information and Computation*, 160(1/2):109–127, 2000.
2. P. A. Abdulla, A. Collomb-Annichini, A. Bouajjani, and B. Jonsson. Using forward reachability analysis for verification of lossy channel systems. *Formal Methods in System Design*, 25(1):39–65, 2004.
3. P. A. Abdulla and B. Jonsson. Verifying programs with unreliable channels. In *Proc. 8th IEEE Int. Symp. Logic in Computer Science (LICS'93)*, pages 160–170, 1993.
4. P. A. Abdulla and B. Jonsson. Ensuring completeness of symbolic verification methods for infinite-state systems. *Theoretical Computer Science*, 256(1–2):145–167, 2001.
5. P. A. Abdulla and A. Nylén. Timed Petri nets and bqos. In *Proc. 22nd Int. Conf. Application and Theory of Petri Nets (ICATPN'01)*, pages 53–70. Springer Verlag LNCS 2075, 2001.
6. S. Abramsky and A. Jung. Domain theory. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 3, pages 1–168. Oxford University Press, 1994.
7. L. Acciai and M. Boreale. Deciding safety properties in infinite-state pi-calculus via behavioural types. In S. Albers, A. Marchetti-Spaccamela, Y. Matias, S. E. Nikoletseas, and W. Thomas, editors, *Proc. ICALP'09*, pages 31–42. Springer Verlag LNCS 5556, 2009.
8. W. W. Adams and P. Loustaunau. *An introduction to Gröbner bases*, volume 3 of *Graduate Studies in Mathematics*. American Mathematical Society, 1994. 289 pages.
9. B. Buchberger and R. Loos. Algebraic simplification. In B. Buchberger, G. E. Collins, R. Loos, and R. Albrecht, editors, *Computer Algebra, Symbolic and Algebraic Computation*. Springer Verlag, 1982-1983.
10. G. Cécé, A. Finkel, and S. Purushothaman Iyer. Unreliable channels are easier to verify than perfect channels. *Information and Computation*, 124(1):20–31, Jan. 1996.
11. P. de Groote, B. Guillaume, and S. Salvati. Vector addition tree automata. In *Proc. 19th IEEE Int. Symp. Logics in Computer Science*, pages 64–73, 2004.
12. C. Dufourd, A. Finkel, and Ph. Schnoebelen. Reset nets between decidability and undecidability. In *ICALP'98*, pages 103–115. Springer Verlag LNCS 1443, 1998.
13. A. Finkel and J. Goubault-Larrecq. Forward analysis for WSTS, part I: Completions. In S. Albers and J.-Y. Marion, editors, *Proc. STACS'09*, pages 433–444, Freiburg, Germany, 2009.
14. A. Finkel and J. Goubault-Larrecq. Forward analysis for WSTS, part II: Complete WSTS. In S. Albers, A. Marchetti-Spaccamela, Y. Matias, and W. Thomas, editors, *Proc. ICALP'09*, pages 188–199, Rhodes, Greece, 2009. Springer Verlag LNCS 5556.

15. A. Finkel, P. McKenzie, and C. Picaronny. A well-structured framework for analysing Petri net extensions. *Information and Computation*, 195(1-2):1–29, 2004.
16. A. Finkel and P. Schnoebelen. Well-structured transition systems everywhere! *Theoretical Computer Science*, 256(1–2):63–92, 2001.
17. G. Geeraerts, J.-F. Raskin, and L. Van Begin. Expand, enlarge and check: New algorithms for the coverability problem of WSTS. *J. Comp. Sys. Sciences*, 72(1):180–203, 2006.
18. G. Gierz, K. H. Hofmann, K. Keimel, J. D. Lawson, M. Mislove, and D. S. Scott. Continuous lattices and domains. In *Encyclopedia of Mathematics and its Applications*, volume 93. Cambridge University Press, 2003.
19. J. Goubault-Larrecq. On Noetherian spaces. In *Proc. 22nd IEEE Int. Symp. Logic in Computer Science (LICS'07)*, pages 453–462, Wrocław, Poland, 2007.
20. A. Grothendieck. *Éléments de géométrie algébrique (rédigés avec la collaboration de Jean Dieudonné): I. Le langage des schémas*, volume 4. Publications mathématiques de l'I.H.É.S., 1960. pages 5–228.
21. T. A. Henzinger, R. Majumdar, and J.-F. Raskin. A classification of symbolic transition systems. *ACM Trans. Computational Logic*, 6(1):1–32, 2005.
22. G. Higman. Ordering by divisibility in abstract algebras. *Proceedings of the London Mathematical Society*, 2(7):326–336, 1952.
23. J. Hopcroft and J. J. Pansiot. On the reachability problem for 5-dimensional vector addition systems. *Theoretical Computer Science*, 8:135–159, 1979.
24. R. M. Karp and R. E. Miller. Parallel program schemata. *Journal of Computer and System Sciences*, 3(2):147–195, 1969.
25. J. B. Kruskal. Well-quasi-ordering, the tree theorem, and Vazsonyi's conjecture. *Transactions of the American Mathematical Society*, 95(2):210–225, 1960.
26. R. Lazič, T. Newcomb, J. Ouaknine, A. W. Roscoe, and J. Worrell. Nets with tokens which carry data. *Fundamenta Informaticae*, 88(3):251–274, 2008.
27. H. Lombardi and H. Perdry. The Buchberger algorithm as a tool for ideal theory of polynomial rings in constructive mathematics. In *Gröbner Bases and Applications (Proc. of the Conference 33 Years of Gröbner Bases)*, volume 251 of *London Mathematical Society Lecture Notes*, pages 393–407. Cambridge University Press, 1998.
28. M. L. Minsky. Recursive unsolvability of Post's problem of "tag" and other topics in the theory of Turing machines. *Annals of Mathematics, Second Series*, 74(3):437–455, 1961.
29. M. Müller-Olm and H. Seidl. Polynomial constants are decidable. In M. V. Hermenegildo and G. Puebla, editors, *Proc. 9th International Symposium on Static Analysis (SAS'02)*, pages 4–19. Springer-Verlag LNCS 2477, 2002.
30. C. R. Murthy and J. R. Russell. A constructive proof of Higman's lemma. In *Proc. 5th IEEE Symposium on Logic in Computer Science (LICS'90)*, pages 257–267, 1990.
31. C. S.-J. A. Nash-Williams. On better-quasi-ordering transfinite sequences. *Proc. Cambridge Philosophical Society*, 64:273–290, 1968.
32. S. Qadeer and J. Rehof. Context-bounded model checking of concurrent software. In N. Halbwachs and L. Zuck, editors, *Proc. 11th Intl. Symp. Tools and Algorithms for the Construction and Analysis of Systems (TACAS'05)*, pages 93–107. Springer Verlag LNCS 3440, 2005.
33. S. Rabinowitch. Zum Hilbertschen Nullstellensatz. *Mathematische Annalen*, 102:520, 1929.
34. C. Reutenauer. *Aspects Mathématiques des Réseaux de Petri*. Masson, 1993.
35. M. Smyth. Effectively given domains. *Theoretical Computer Science*, 5:257-274, 1977.
36. P. Taylor. Computably based locally compact spaces. *Logical Methods in Computer Science*, 2(1), 2006.
37. K. N. Verma and J. Goubault-Larrecq. Karp-Miller trees for a branching extension of VASS. *Discrete Mathematics & Theoretical Computer Science*, 7(1):217–230, 2005.