Separability and Non-Determinizability of WSTS

Eren Keskin ✉
TU Braunschweig, Germany

Roland Meyer ✉
TU Braunschweig, Germany

Abstract

There is a recent separability result for the languages of well-structured transition systems (WSTS) that is surprisingly general: disjoint WSTS languages are always separated by a regular language. The result assumes that one of the languages is accepted by a deterministic WSTS, and it is not known whether this assumption is needed. There are two ways to get rid of the assumption, none of which has led to conclusions so far: (i) show that WSTS can be determinized or (ii) generalize the separability result to non-deterministic WSTS languages. Our contribution is to show that (i) does not work but (ii) does. As for (i), we give a non-deterministic WSTS language that we prove cannot be accepted by a deterministic WSTS. The proof relies on a novel characterization of the languages accepted by deterministic WSTS. As for (ii), we show how to find finitely represented inductive invariants without having the tool of ideal decompositions at hand. Instead, we work with closures under converging sequences. Our results hold for upward- and downward-compatible WSTS.

2012 ACM Subject Classification Theory of computation → Formal languages and automata theory; Theory of computation → Regular languages

Keywords and phrases WSTS, regular separability, determinization

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2023.8

Related Version *Full Version*: `https://arxiv.org/abs/2305.02736`

Funding This work was supported by the DFG project EDS@SYN: Effective Denotational Semantics for Synthesis.

1 Introduction

Czerwiński et al. [16, Theorems 6 and 7] have recently established a separability result for the languages of well-structured transition systems (WSTS) [20, 4, 2, 23] that is surprisingly general. Disjoint WSTS languages are always separated by a regular language: whenever we have $L(U) \cap L(V) = \emptyset$, then there is a regular language R with $L(U) \subseteq R$ and $R \cap L(V) = \emptyset$. The result says that WSTS languages either intersect, or they are far apart in that a finite amount of information is sufficient to distinguish them. Applications abound, we elaborate on this in the related work. Unfortunately, the result comes with a grain of salt: it assumes that one of the WSTS, U or V, is deterministic. All attempts to remove the assumption have failed so far. The assumption is used for a central argument in the proof, namely that inductive invariants can be represented in a finite way. With determinism, these invariants are downward-closed sets in a WQO, and hence decompose into finitely many ideals [31, 21, 22]. This is precisely the finite amount of information needed for regularity.

A strategy to circumvent the assumption would be to show that WSTS can be determinized. Czerwiński et al. already argue in this direction. In [16, Theorem 5], they show that both finitely-branching WSTS and WSTS over so-called ω^2-WQOs can be determinized. Unfortunately, this does not cover all WSTS. To sum up, it is still open whether the regular separability result holds for all WSTS languages, and we do not understand the impact of non-determinism on the expressiveness of the WSTS model.

34th International Conference on Concurrency Theory (CONCUR 2023).
Editors: Guillermo A. Pérez and Jean-François Raskin; Article No. 8; pp. 8:1–8:17
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Our first contribution is to prove the regular separability result for all WSTS languages, without the assumption of determinism. We accept the fact that determinizing a WSTS no longer yields a WSTS, and carefully study the resulting class of transition systems. They are formed over a lattice in which sequences have subsequences that converge in a natural sense. This leads us to define the closure of a set by adding the limits of all converging sequences. The key insight is that the closure of an inductive invariant is again an inductive invariant. Together with the fact that closed sets have finitely many maximal elements, we arrive at the desired finite representation. In short, when moving from WQOs to converging lattices, maximal elements of closed sets form an alternative to ideal decompositions of downward-closed sets. We call the new transition systems converging.

Our second contribution is to show that WSTS cannot be determinized in general. We give a WSTS language T that we prove cannot be accepted by a deterministic WSTS. The proof relies on a novel characterization of the deterministic WSTS languages: they are precisely the languages whose Nerode (right) quasi order is a WQO. The characterization provides a first hint on how to construct T. The language should have an infinite antichain in the Nerode quasi order, for then this cannot be a WQO. The second hint stems from the determinizability result [16, Theorem 5]. The accepting WSTS should be infinitely branching and the WQO should be no ω^2-WQO. Such WQOs embed the so-called Rado WQO [8, Section 2]. Moreover, the Rado WQO is known to have an infinite antichain when constructing downward-closed sets [22, Proposition 4.2]. The definition of T is thus guided by the idea of translating the Rado antichain into an antichain in the Nerode quasi order. Interestingly, the underlying WSTS is deterministic except for the choice of the initial state.

We develop these results for upward-compatible WSTS [23]. Our third contribution is to show that they also hold for downward-compatible WSTS. We achieve this by proving general relationships between the models. A key insight is that the complement of a deterministic upward-compatible WSTS is a deterministic downward-compatible WSTS. Moreover, the reversal of an upward-compatible WSTS language is a downward-compatible WSTS language.

Details and proofs missing here can be found in the full version of this article [32].

Related Work. The converging transition systems (CTS) we use to generalize the regular separability result [16] have a topological flavor, and indeed are inspired by Goubault-Larrecq's Noetherian transition systems [26, 27]. One difference is that we had to formulate CTS in lattice-theoretic terms to be able to import a theorem from [16] that links regular separability to the existence of finitely represented inductive invariants. Another difference is the study of such invariants (we prove stability under closure) that has no analogue in [26, 27].

We show that deterministic WSTS accept a strictly weaker class of languages than their non-deterministic counterparts. The work [3] also compares classes of WSTS languages, but for fixed models (extended Petri nets). We allow the determinization to freely select the WQO and the transitions, meaning we have considerably less syntactic constraints to work with. There are also pumping lemmas to distinguish WSTS languages from (among others) context-free languages [24]. Our characterization of the deterministic WSTS languages is stronger than the necessary conditions in pumping lemmas. Our language witnessing the weakness of deterministic WSTS is accepted by an infinitely-branching WSTS, a class of systems studied in [7]. That work concentrates on decidability results and pays attention to effectiveness, while we prove a statement of existence and do not need such assumptions.

There is recent interest in separability problems for infinite-state systems [17, 40, 14, 39, 12]. One reason is that standard algorithms rarely apply to separability problems, but these problems tend to call for new approaches. With the basic separator technique [18], Czerwiński and Zetzsche have shown that there is hope for general methods that apply to a range of separability problems [10, 11, 15]. With the closure of inductive invariants under converging sequences, we hope to also have contributed a versatile tool.

Another reason for the popularity of separability problems is their usefulness in verification. In [1], separators act as interpolants in abstraction-guided verification [9]. In [6], separators are advocated as interfaces in rely-guarantee reasoning [30]. In this context, our result implies that regular interfaces yield a complete proof method, provided the system is well-structured.

2 Well-Structured Transition Systems

We recall well-structured transition systems (WSTS) with upward compatibility [20, 4, 2, 23]. Downward compatibility will be addressed in Section 5.

Orders. Let $(Q, \leq)$ be a quasi order and $P \subseteq Q$. We call P a chain, if $\leq$ restricted to P is a total order. We call P an antichain, if the elements in P are pairwise incomparable. The upward closure of P is $\uparrow P = \{q \in Q \mid \exists p \in P.\ p \leq q\}$. We call P upward closed, if $P = \uparrow P$. The powerset of Q restricted to the upward-closed sets is $\mathbb{U}(Q)$. The downward closure is defined similarly and we use $\mathbb{D}(Q)$ for the downward-closed sets. We call $(Q, \leq)$ a well quasi order (WQO), if for every infinite sequence $[p_i]_{i \in \mathbb{N}}$ in Q there are indices $i < j$ with $p_i \leq p_j$.

Let $(Q, \leq)$ be a partially-ordered set. We write $\max P$ for the set of maximal elements in the subset $P \subseteq Q$. They may not exist, in which case the set is empty. We call $(Q, \leq)$ a complete lattice, if all $P \subseteq Q$ have a greatest lower bound in Q, also called meet and denoted by $\bigsqcap P \in Q$, and a least upper bound in Q, also called join and denoted by $\bigsqcup P \in Q$. A function $f : Q \to Q$ on a complete lattice is join preserving [13, Section 11.4], if it distributes over arbitrary joins in that $f(\bigsqcup P) = \bigsqcup f(P)$ for all $P \subseteq Q$, where $f(P) = \{f(p) \mid p \in P\}$. We call $(Q, \leq)$ a completely distributive lattice, if it is a complete lattice where arbitrary meets distribute over arbitrary joins, and vice versa:

$$\bigsqcap_{a \in A} \bigsqcup_{b \in B_a} p_{a,b} = \bigsqcup_{f \in C_{A,B}} \bigsqcap_{a \in A} p_{a,f(a)} \qquad \bigsqcup_{a \in A} \bigsqcap_{b \in B_a} p_{a,b} = \bigsqcap_{f \in C_{A,B}} \bigsqcup_{a \in A} p_{a,f(a)}\ .$$

The definition makes use of the Axiom of Choice: $C_{A,B}$ denotes the set of choice functions that map each $a \in A$ to a choice $b \in B_a$. It is also important to note that, for any quasi order $(Q, \leq)$, $(\mathbb{D}(Q), \subseteq)$ is a completely distributive lattice.

Labeled Transition Systems. A labeled transition system (LTS) is a tuple $U = (Q, I, \Sigma, \delta, F)$ that consists of a set of states Q, in our setting typically infinite, a set of inital states $I \subseteq Q$, a set of final states $F \subseteq Q$, a finite alphabet Σ, and a set of labeled transitions $\delta : Q \times \Sigma \to \mathbb{P}(Q)$. The LTS is deterministic, if $|I| = |\delta(p, a)| = 1$ for all $p \in Q$ and $a \in \Sigma$.

Its language is the set of words that can reach a final state from an initial state:

$$L(U) = \{w \in \Sigma^* \mid \delta(I, w) \cap F \neq \emptyset\}\ .$$

Here, we extend the transition relation to sets of states and words: $\delta(P, w.a) = \delta(\delta(P, w), a)$ and $\delta(P, a) = \bigcup_{p \in P} \delta(p, a)$. Finally, if the LTS is deterministic, we write $(Q, y, \Sigma, \delta, F)$ and $\delta(p, a) = q$ instead of $(Q, \{y\}, \Sigma, \delta, F)$ and $\delta(p, a) = \{q\}$.

Let U_1 and U_2 be LTS with $U_i = (Q_i, I_i, \Sigma, \delta_i, F_i)$. We define their synchronized product to be the LTS $U_1 \times U_2 = (Q_1 \times Q_2, I_1 \times I_2, \Sigma, \delta, F_1 \times F_2)$ where $(q_1, q_2) \in \delta((p_1, p_2), a)$, if $q_1 \in \delta_1(p_1, a)$ and $q_2 \in \delta_2(p_2, a)$. Then $L(U_1 \times U_2) = L(U_1) \cap L(U_2)$.

Compatibility. We work with LTS $U = (Q, I, \Sigma, \delta, F)$ whose states form a quasi order $(Q, \leq)$ that is compatible with the remaining components as follows. We have $F = \uparrow F$, the final states are upward closed wrt. $\leq$. Moreover, $\leq$ is a simulation relation [36]: for all pairs of related states $p_1 \leq q_1$ and for all letters $a \in \Sigma$ we have:

for all $p_2 \in \delta(p_1, a)$ there is $q_2 \in \delta(q_1, a)$ with $p_2 \leq q_2$.

We also make the quasi order explicit and call $U = (Q, \leq, I, \Sigma, \delta, F)$ an *upward-compatible* LTS (ULTS).

ULTS can be determinized, in the case of U this yields

$$U^{det} \quad = \quad (\mathbb{D}(Q), \subseteq, \downarrow I, \Sigma, \delta^{det}, F^{det}) \ .$$

The states are the downward-closed sets ordered by inclusion, the transition relation is defined by closing the result of the original transition relation downwards, $\delta^{det}(D, a) = \downarrow\delta(D, a)$ for all $D \in \mathbb{D}(Q)$ and $a \in \Sigma$, and the set of final states consists of all downward-closed sets that contain a final state in the original ULTS, $F^{det} = \{D \in \mathbb{D}(Q) \mid D \cap F \neq \emptyset\}$.

▶ **Lemma 1.** *Let U be an ULTS. Then U^{det} is a deterministic ULTS with $L(U^{det}) = L(U)$.*

We write detULTS for the class of deterministic ULTS. The synchronized product of ULTS is again an ULTS (with the product order).

Well-Structuredness. An *upward-compatible well-structured transition system* (WSTS) is an ULTS U whose states $(Q, \leq)$ form a WQO. The synchronized product of WSTS is again a WSTS. We are interested in L(WSTS), the class of all languages accepted by WSTS. We also study L(detWSTS) $\subseteq L$(WSTS), the class of languages accepted by deterministic WSTS.

We observe that we can focus on WSTS with a countable number of states.

▶ **Lemma 2.** *For every $L \in L(WSTS)$ there is a WSTS U with a countable number of states so that $L = L(U)$.*

The lemma needs two arguments: the language consists of a countable number of words, and we can assume the transition relation to yield downward-closed sets.

3 Regular Separability of WSTS Languages

Two languages $L_1, L_2 \subseteq \Sigma^*$ are *separable by a regular language*, denoted by $L_1 \,|\, L_2$, if there is a regular language $R \subseteq \Sigma^*$ with $L_1 \subseteq R$ and $R \cap L_2 = \emptyset$. Our main result is that disjoint WSTS languages are always separable in this sense.

▶ **Theorem 3.** *For $L_1, L_2 \in L(WSTS)$, we have $L_1 \,|\, L_2$ if and only if $L_1 \cap L_2 = \emptyset$.*

The conclusion is the same as in the main theorem of [16], but we do not need the premise that one of the languages is accepted by a deterministic WSTS. The implication from left to right is trivial, the implication from right to left is our first contribution.

We summarize the arguments. The plan is to invoke the proof principle for regular separability in [16, Theorem 11] and show that the product system has a finitely represented inductive invariant. The principle holds for general ULTS but needs one of them deterministic. Therefore, our first step is to determinize the given WSTS. Determinizing a WSTS will yield an ULTS, but may ruin the WQO property. We show that the set of states of the resulting ULTS still has a rich structure: it is a powerset lattice in which every infinite sequence contains a subsequence that converges in a natural sense. We call such ULTS converging transition systems (CTS). We only define CTS as deterministic models, which is why we determinize both WSTS. CTS are closed under products. Moreover, since the initial languages are disjoint by the assumption, the product trivially has an inductive invariant. It thus remains to turn this invariant of the product into an invariant that can be represented in a finite way. The idea is to add the limits of all converging sequences in the invariant. Since the CTS transitions are compatible with limits, the resulting set of states is again an inductive invariant. By Zorn's lemma, every set can contain only finitely many maximal elements. The maximal elements thus form the finite representation that was needed to conclude the proof.

It would be possible to give the proof at a set-theoretic level, by explicitly working with products of powerset lattices. CTS allow us to abstract away the product structure and highlight the key arguments in the limit construction. We turn to the details.

3.1 Proof Principle for Regular Separability

To establish regular separability, we rely on a proof principle introduced in [16]. The notion of an inductive invariant will be recalled in a moment.

▶ **Theorem 4** (Proof principle for regular separability, [16, Theorem 11]). *Consider ULTS U, V, one deterministic. If $U \times V$ has a finitely represented inductive invariant, then $L(U) \mid L(V)$.*

Interestingly, the proof principle does not need the WQO assumption of WSTS but holds for general ULTS. It does assume one of the ULTS to be deterministic, though. Recall that an *inductive invariant* for an ULTS $(Q, \leq, I, \Sigma, \delta, F)$ is a downward-closed set of states $S \subseteq Q$ that includes all initial states, excludes all final states, and is closed under taking transitions:

$$I \subseteq S \qquad S \cap F = \emptyset \qquad \delta(S, a) \subseteq S\ .$$

The inductive invariant is *finitely represented*, if there is a finite set $C \subseteq S$ with $S = \downarrow C$. We refer to a set C that satisfies this as a *cover* of S.

When trying to invoke Theorem 4, finding an inductive invariant for $U \times V$ is easy: the invariant is guaranteed to exist as soon as the language $L(U \times V) = L(U) \cap L(V)$ is empty, which is precisely the hypothesis we start from.

▶ **Lemma 5** ([16, Lemma 10]). *An ULTS U admits an inductive invariant iff $L(U) = \emptyset$.*

The difficult part is to find an inductive invariant that can be represented in a finite way. In [16], this was addressed with ideal decompositions [31, 21, 22]. The ideal decompositions, however, needed the WQO assumption, which lead to the requirement in the main theorem that one WSTS had to be deterministic. As we show in Section 4, this is a real restriction: there are WSTS languages that cannot be accepted by a deterministic WSTS.

Our contribution is to find finitely represented inductive invariants without making use of ideal decompositions. Our approach is to determinize the given WSTS with the construction in Lemma 1, and accept that we can no longer guarantee the result to be a WSTS.

3.2 Converging Transition Systems: WSTS in Disguise

We propose converging transition systems (CTS), a new class of ULTS that is general enough to capture determinized WSTS and retains enough structure to establish the existence of finitely represented inductive invariants. CTS are inspired by Noetherian transition systems [26, 27], but are formulated in a lattice-theoretic rather than in a topological way.

Recall that determinized WSTS have as state space $(\mathbb{D}(Q), \subseteq)$, where $(Q, \leq)$ is a WQO. In a WQO, every infinite sequence admits an increasing subsequence. It is well known [38] that this may not hold for $(\mathbb{D}(Q), \subseteq)$. However, a natural relaxation holds: every infinite sequence $[X_i]_{i\in\mathbb{N}}$ admits an infinite subsequence $[X_{\varphi(i)}]_{i\in\mathbb{N}}$, where any element that is present in one set is present in almost every set. A similar property, defined for complete lattices, is called convergence in the literature [25]. Our definition differs from the citation in two ways. We restrict ourselves to sequences (as opposed to nets), and we require convergence to the join (as opposed to $\limsup = \liminf$). This suffices for our setting.

▶ **Definition 6.** *A* converging lattice $(Q, \leq)$ *is a completely distributive lattice, where every sequence* $[p_i]_{i\in\mathbb{N}}$ *has a converging subsequence* $[p_{\varphi(i)}]_{i\in\mathbb{N}}$*. A* converging sequence $[q_i]_{i\in\mathbb{N}}$ *is an infinite sequence with*

$$\bigsqcup_{i\in\mathbb{N}} \bigsqcap_{j\geq i} q_j = \bigsqcup_{i\in\mathbb{N}} q_i \ .$$

The equality formalizes our explaination from before. In the context of sets, where join and meet are respectively union and intersection, the right-hand side of the equation contains all elements that appear in any set in the sequence. The left side iterates over every finite initial segment, and includes every element that appears in all sets outside of this segment. Every element that is missing in only finitely many sets will eventually be included.

Converging lattices not only generalize downward-closed subsets of WQOs, they are also a sufficient condition for them. The backward direction is by [38, Proof of Theorem 3]. The forward direction is by an application of the following fact [38], also [33, Fact III.3]: $(\mathbb{D}(Q), \subseteq)$ is well-founded if and only if the order is a WQO. The details are given in [32].

▶ **Lemma 7.** $(\mathbb{D}(Q), \subseteq)$ *is a converging lattice if and only if* $(Q, \leq)$ *is a WQO.*

The space of converging sequences is closed under the application of join preserving functions as formulated next. While we would expect this result to be known, we have not found a reference. The lemma is central to our argument, therefore we give the proof.

▶ **Lemma 8.** *Let* $(Q, \leq)$ *be a lattice,* $[p_i]_{i\in\mathbb{N}}$ *a converging sequence in* Q*, and* $f : Q \to Q$ *a join preserving function. Then also* $[f(p_i)]_{i\in\mathbb{N}}$ *is converging.*

Proof. Due to convergence of the given sequence, we have $\bigsqcup_{i\in\mathbb{N}} \bigsqcap_{j\geq i} p_j = \bigsqcup_{i\in\mathbb{N}} p_i$. This equality yields $f(\bigsqcup_{i\in\mathbb{N}} \bigsqcap_{j\geq i} p_j) = f(\bigsqcup_{i\in\mathbb{N}} p_i)$. By join preservation of f, we get

$$\bigsqcup_{i\in\mathbb{N}} f(\bigsqcap_{j\geq i} p_j) = \bigsqcup_{i\in\mathbb{N}} f(p_i) \ .$$

Function f is not assumed to be meet preserving. But we can show an inequality that is sufficient for our needs. For all $S \subseteq Q$ and $s \in S$, we have $f(\bigsqcap S) \leq f(s) \sqcup f(\bigsqcap S)$. Join preservation and the fact that $s \in S$ yield $f(s) \sqcup f(\bigsqcap S) = f(s \sqcup \bigsqcap S) = f(s)$. We have thus shown $f(\bigsqcap S) \leq f(s)$ for all $s \in S$. This means $f(\bigsqcap S) \leq \bigsqcap_{s\in S} f(s)$.

We apply this inequality to the previous equality:

$$\bigsqcup_{i\in\mathbb{N}} f(p_i) = \bigsqcup_{i\in\mathbb{N}} f(\bigsqcap_{j\geq i} p_j) \leq \bigsqcup_{i\in\mathbb{N}} \bigsqcap_{j\geq i} f(p_j) \leq \bigsqcup_{i\in\mathbb{N}} f(p_i) \ .$$

This is $\bigsqcup_{i\in\mathbb{N}} \bigsqcap_{j\geq i} f(p_j) = \bigsqcup_{i\in\mathbb{N}} f(p_i)$, as desired. ◀

We explain the considerations that lead us to the definition of CTS given below. In the light of Lemma 7, the states of a CTS should form a converging lattice. This, however, was not enough to guarantee the existence of finitely represented inductive invariants. One requirement of invariants is that they are closed under taking transitions. To understand which sets satisfy this, we had to restrict the transition relation. We define CTS only as a deterministic model. Then the transitions form a function $\delta(-, a)$ for every letter $a \in \Sigma$. Upward compatibility of these functions is not very informative. Consider determinized WSTS: upward compatibility gives us $\delta(S_0 \cup S_1, a) \supseteq \delta(S_0, a)$, while we expect $\delta(S_0 \cup S_1, a) = \delta(S_0, a) \cup \delta(S_1, a)$. In lattice-theoretic terms, we expect the transition functions $\delta(-, a)$ to be join preserving. A benefit of this requirement is of course that it makes Lemma 8 available. An invariant should also be disjoint from the final states so that we had to control this set as well. When determinizing WSTS, a set $D \in \mathbb{D}(Q)$ is final as soon as it contains a single final state. Given the definition of convergence, we relax this to containing a finite set of final states.

▶ **Definition 9.** *A* converging transition system *(CTS) is an ULTS $U = (Q, \leq, y, \Sigma, \delta, F)$ that is deterministic, where $(Q, \leq)$ is a converging lattice, the functions $\delta(-, a)$ are join preserving for all $a \in \Sigma$, and the final states satisfy*

finite acceptance: for every $\bigsqcup K \in F$ there is a finite set $N \subseteq K$ with $\bigsqcup N \in F$.

The determinization of a WSTS yields a CTS, as it was one of the goals of the CTS definition. Somewhat surprisingly, CTS do not add expressiveness but their languages are already accepted by (non-deterministic) WSTS. The construction is via join prime elements and can be found in the full version [32]. Together, the CTS languages are precisely the WSTS languages, and one may see Definition 9 as a reformulation of the WSTS model.

▶ **Proposition 10.** *If U is a WSTS, then U^{det} is a CTS. For every CTS V, there is a WSTS U with $L(V) = L(U)$. Together, $L(WSTS) = L(CTS)$.*

The correspondence allows us to import the countability assumption from Lemma 2. Indeed, if the WQO $(Q, \leq)$ is countable, then there is only a countable number of downward-closed sets in $(\mathbb{D}(Q), \subseteq)$. This is by a standard argument for WSTS: each downward-closed set can be characterized by its complement, the complement is upward closed, and is therefore characterized by its finite set of minimal elements.

▶ **Lemma 11.** *For every $L \in L(CTS)$, there is a CTS U over a countable number of states so that $L = L(U)$*

We will also need that CTS are closed under synchronized products.

▶ **Lemma 12.** *If U and V are CTS, so is $U \times V$.*

We summarize the findings so far. Given disjoint WSTS languages $L(V_1) \cap L(V_2) = \emptyset$, the goal is to show regular separability $L(V_1) \mid L(V_2)$. We first determinize both WSTS. By Proposition 10, V_1^{det} and V_2^{det} are CTS. Moreover, by Lemma 1, determinization preserves the language. We use Lemma 11 to obtain countable CTS U_1 and U_2 that accept the same languages. To show regular separability, we now intend to invoke Theorem 4 on U_1 and U_2. CTS are already deterministic. It thus remains to show that $U_1 \times U_2$ has a finitely represented inductive invariant. With Lemma 12, $U_1 \times U_2$ is another CTS U. Moreover, the product corresponds to language intersection, so $L(U) = \emptyset$. By Lemma 5, we know that U has an inductive invariant. We now show how to turn this invariant into a finitely represented one.

3.3 Inductive Invariants in CTS

We show the following surprising property for countable CTS: every inductive invariant S can be generalized to an inductive invariant $cl(S)$ that is finitely represented. The closure operator is defined by adding to S the joins of all converging sequences:

$$cl(S) \quad = \quad \{ \bigsqcup_{i\in\mathbb{N}} p_i \mid [p_i]_{i\in\mathbb{N}} \text{ a converging sequence in } S \} \ .$$

▶ **Proposition 13.** *Let U be a countable CTS and S an inductive invariant of U. Then also $cl(S)$ is an inductive invariant of U and it is finitely represented.*

The proposition concludes the proof of Theorem 3. We simply invoke it on the inductive invariant that exists by Lemma 5 as discussed above. The rest of the section is devoted to the proof. We fix a countable CTS $U = (Q, \leq, y, \Sigma, \delta, F)$ and an inductive invariant $S \subseteq Q$.

As Lemma 14 states, the closure is expansive and idempotent. This means further applications do not add new limits. Here, we need the fact that we have a completely distributive lattice. Moreover, the closure yields a downward-closed set. The closure is also trivially monotonic, and hence an upper closure operator indeed [13, Section 11.7], but we will not need monotonicity. The proof of Lemma 14 is given in the full version [32].

▶ **Lemma 14.** $S \subseteq cl(S) = cl(cl(S)) = \downarrow cl(S)$.

Towards showing Proposition 13, we first argue for invariance.

▶ **Lemma 15.** *$cl(S)$ is an inductive invariant.*

Proof. To prove that $cl(S)$ is an inductive invariant, we must show two properties for the joins $\bigsqcup_{i\in\mathbb{N}} p_i = p$ of converging sequences $[p_i]_{i\in\mathbb{N}}$ in S that we added. First, we must show that we do not leave $cl(S)$ when taking transitions, $\delta(p, a) \in cl(S)$ for all $a \in \Sigma$. Second, we must show that the join is not a final state. We begin with the latter. Towards a contradiction, suppose $p \in F$. Convergence yields $\bigsqcup_{i\in\mathbb{N}} \bigsqcap_{j\geq i} p_j \in F$. By the finite acceptance property of CTS, there must be a finite set $K \subseteq \mathbb{N}$ with $k = \max K$ so that

$$\bigsqcup_{i\in K} \bigsqcap_{j\geq i} p_j = \bigsqcap_{j\geq k} p_j \in F \ .$$

Since $\bigsqcap_{j\geq k} p_j \leq p_k$ and F is upward closed, we obtain $p_k \in F$. This is a contradiction: p_k belongs to the inductive invariant S and the invariant does not intersect the final states.

To show $\delta(p, a) \in cl(S)$, we first note that $\delta(p_i, a) \in S$ for all $i \in \mathbb{N}$. This holds as S is an invariant and $p_i \in S$. We now argue that not only the sequence $[\delta(p_i, a)]_{i\in\mathbb{N}}$ is in S, but also its join is in the closure. We use that the transition function $\delta(-, a)$ is join preserving. This allows us to apply Lemma 8 showing that $[\delta(p_i, a)]_{i\in\mathbb{N}}$ coverges. Since the sequence belongs to S, we obtain $\bigsqcup_{i\in\mathbb{N}} \delta(p_i, a) \in cl(S)$. We conclude by applying join preservation:

$$\delta(p, a) = \delta(\bigsqcup_{i\in\mathbb{N}} p_i, a) = \bigsqcup_{i\in\mathbb{N}} \delta(p_i, a) \in cl(S) \ . \qquad ◀$$

It only remains to show that $cl(S)$ is finitely represented.

▶ **Proposition 16.** *There is a finite set $C \subseteq cl(S)$ so that $\downarrow C = cl(S)$.*

We break down the proof of Proposition 16 into two steps. First, we show that $cl(S)$ can be covered by an antichain. Then, we show that infinite antichain covers do not exist. This implies that there must be a finite antichain cover. The proofs reasons over *closed* sets, sets that contain the limits of their converging sequences. We rely on the fact that closed sets have at least one maximal element.

▶ **Lemma 17.** *Consider $G \subseteq Q$ closed and non-empty. Then $\max G \neq \emptyset$.*

Moreover, closedness remains intact after certain removals.

▶ **Lemma 18.** *Consider $G, H \subseteq Q$ where G is closed. Then $G \setminus {\downarrow}H$ is closed.*

We postpone the proofs of these lemmas until after the proof of Proposition 16.

▶ **Lemma 19.** *There is an antichain cover of $cl(S)$.*

Proof. We claim that the maximal elements $\max cl(S)$ form an antichain cover of $cl(S)$. It is clear that $\max cl(S)$ is an antichain. Since $cl(S)$ is downward closed by Lemma 14, we also have ${\downarrow}(\max cl(S)) \subseteq cl(S)$. To see that $\max cl(S)$ is a cover, let $G = cl(S) \setminus {\downarrow}(\max cl(S))$ and suppose $G \neq \emptyset$. Lemma 18 tells us that G is closed. By Lemma 17, we get $\max G \neq \emptyset$. Consider $p \in \max G$. By the definition of G, we have $p \notin \max cl(S)$. Then, however, there must be $q \in cl(S)$ with $p \leq q$ and $p \neq q$. If $q \in {\downarrow}(\max cl(S))$, then $p \in {\downarrow}(\max cl(S))$ as well, which is a contradiction to $p \in G$. If conversely $q \in cl(S) \setminus {\downarrow}(\max cl(S)) = G$, then we have a contradiction to $p \in \max G$. ◀

Now we prove the second part of Proposition 16, which states that there can be no infinite antichain cover.

▶ **Lemma 20.** *There is no infinite antichain cover of $cl(S)$.*

Proof. Suppose there is an infinite antichain cover $C \subseteq cl(S)$. Then, there is an infinite sequence $[p_i]_{i\in\mathbb{N}}$ in C. By Definition 6, it has an infinite converging subsequence $[p_{\varphi(i)}]_{i\in\mathbb{N}}$. The closure operator adds $\bigsqcup_{i\in\mathbb{N}} p_{\varphi(i)}$ to $cl(S)$. Since C is a cover of $cl(S)$, there must be $q \in C$ with $\bigsqcup_{i\in\mathbb{N}} p_{\varphi(i)} \leq q$. Because $p_{\varphi(i)} \sqcup p_{\varphi(0)} \leq q$ and $p_{\varphi(i)}, p_{\varphi(0)}$ are incomparable, we have $p_{\varphi(i)} < q$ for all $i \in \mathbb{N}$. So $p_{\varphi(i)} < q$ for all $i \in \mathbb{N}$, while at the same time $q, p_{\varphi(i)} \in C$. This contradicts the antichain property. ◀

We conclude by showing Lemma 17 and 18.

Proof of Lemma 17. Let $\emptyset \neq G \subseteq Q$ be closed. We prove G chain complete, meaning for every chain $P \subseteq G$ the limit $\bigsqcup P$ is again in G. Then Zorn's lemma [29] applies and yields $\max G \neq \emptyset$. We have Zorn's lemma, because we agreed on the Axiom of Choice. Towards chain completeness, consider an increasing sequence $[p_i]_{i\in\mathbb{N}}$ in G. We prove that $\bigsqcup_{i\in\mathbb{N}} p_i \in G$. For any $i \in \mathbb{N}$, we have $\bigsqcap_{j\geq i} p_i = p_i$. Hence, replacing each meet with the smallest element shows convergence. Since $[p_i]_{i\in\mathbb{N}}$ converges and G is closed, we have $\bigsqcup_{i\in\mathbb{N}} p_i \in G$.

Although we are in a countable setting, the argument for sequences does not yet cover all chains. The problem is that the counting processs may not respect the order. To see this, consider a chain $P \subseteq G$ of ordinal size $|P| = \omega \cdot 2$. The chain is countable, but no counting process can respect the order. We now argue that still $\bigsqcup P \in G$. By [34, Theorem 1], there is a (wrt. inclusion) increasing sequence of subsets $[P_i]_{i\in\mathbb{N}}$ in $\mathbb{P}(P)$, where each P_i is finite and $\bigcup_{i\in\mathbb{N}} P_i = P$. Finite chains contain maximal elements, so let $p_i = \max P_i = \bigsqcup P_i$. Then

$$\bigsqcup P \;=\; \bigsqcup \bigcup_{i\in\mathbb{N}} P_i \;=\; \bigsqcup_{i\in\mathbb{N}} \bigsqcup P_i \;=\; \bigsqcup_{i\in\mathbb{N}} p_i \,.$$

Since $[P_i]_{i\in\mathbb{N}}$ is an increasing sequence, $[p_i]_{i\in\mathbb{N}}$ is also an increasing sequence. As we have shown before, $\bigsqcup_{i\in\mathbb{N}} p_i \in G$. This concludes the proof. ◀

Proof of Lemma 18. Consider $G, H \subseteq Q$ with G closed. We show that $G \setminus \downarrow H$ is closed. Let $[p_i]_{i\in\mathbb{N}}$ be a converging sequence in $G \setminus \downarrow H$. Let $q = \bigsqcup_{i\in\mathbb{N}} p_i$ and suppose $q \notin G \setminus \downarrow H$. Since G is closed, $q \in G$. Then necessarily $q \in \downarrow H$. But by definition, $p_i \leq q$ for all $i \in \mathbb{N}$. So $p_i \in \downarrow H$ as well. This contradicts the fact that the sequence $[p_i]_{i\in\mathbb{N}}$ lives in $G \setminus \downarrow H$. ◀

4 Non-Determinizability of WSTS

We show that the detWSTS languages form a strict subclass of the WSTS languages. To this end, we define a WSTS language T that we prove cannot be accepted by a detWSTS. The proof relies on a novel characterization of the detWSTS languages that may be of independent interest. In the following, we call T the *witness language*. This is our second main result.

▶ **Theorem 21.** *$\mathcal{L}(detWSTS) \neq \mathcal{L}(WSTS)$.*

Towards the definition of T, recall that finitely-branching WSTS and WSTS over so-called ω^2-WQOs can be determinized [16, Theorem 5]. Moreover, it is known that ω^2-WQOs are precisely the WQOs that do not embed the Rado WQO [8, Section 2]. This suggests we should accept the witness language T by an infinitely-branching WSTS over the Rado WQO. We begin with our characterization of the detWSTS languages, as it will provide additional guidance in the definition of the witness language.

4.1 Characterization of the detWSTS Languages

Our characterization is based on a classical concept in formal languages [28, Theorem 3.9]. The *Nerode quasi order* $\leq_L \subseteq \Sigma^* \times \Sigma^*$ of a language $L \subseteq \Sigma^*$ is defined by $w \leq_L v$, if

for all $u \in \Sigma^*$ we have that $w.u \in L$ implies $v.u \in L$.

The characterization says that the detWSTS languages are precisely the languages whose Nerode quasi order is a WQO. Note that this is not the folklore result [5, Proposition 5.1] saying that a language is regular if and only if the syntactic quasi order is a WQO.

▶ **Lemma 22** (Characterization of $\mathcal{L}$(detWSTS))**.** *$L \in \mathcal{L}(detWSTS)$ iff $\leq_L$ is a WQO.*

Proof. $\Rightarrow$ Let $L = L(U)$ with $U = (Q, \leq, i, \Sigma, \delta, F)$ a detWSTS. We extend the order $\leq \subseteq Q \times Q$ on the states to an order $\leq_U \subseteq \Sigma^* \times \Sigma^*$ on words by setting $w \leq_U v$, if $\delta(i, w) = p$ and $\delta(i, v) = q$ with $p \leq q$. Since U is deterministic, p and q are guaranteed to exist and be unique. It is easy to see that $\leq_U$ is a WQO. We now show that $\leq_U$ is included in the Nerode quasi order, and so also $\leq_L$ is a WQO. To this end, we consider $w \leq_U v$ and $u \in \Sigma^*$ with $w.u \in L$, and show that also $v.u \in L$. We have $\delta(i, w.u) = \delta(p_1, u) = p_2$ and $\delta(i, v.u) = \delta(q_1, u) = q_2$ with $p_1 = \delta(i, w)$ and $q_1 = \delta(i, v)$. Since $w \leq_U v$, we have $p_1 \leq q_1$. With the simulation property of WSTS, this implies $p_2 \leq q_2$. Since $w.u \in L$ and $L = L(U)$, we get $p_2 \in F$. Since F is upward closed, also $q_2 \in F$. Hence, $v.u \in L(U) = L$ as desired.

$\Leftarrow$ Consider a language $L \subseteq \Sigma^*$ whose Nerode quasi order $\leq_L$ is a WQO. We define the trivial detWSTS $U_L = (\Sigma^*, \leq_L, \varepsilon, \Sigma, \delta, L)$. The states are all words ordered by the Nerode quasi order. The empty word is the initial state, the language L is the set of final states. Note that L is upward closed wrt. $\leq_L$. The transition relation is defined as expected, $\delta(w, a) = w.a$. It is readily checked that $L(U_L) = L$. ◀

The lemma gives a hint on how to construct the witness language T: we should make sure the associated Nerode quasi order $\leq_T$ has an infinite antichain (then it cannot be a WQO). To obtain such an antichain, remember that T will be accepted by a WSTS over the Rado

WQO $(R, \leq_R)$ [38]. It is known that $(\mathbb{D}(R), \subseteq)$ has an infinite antichain. Our strategy for the definition of T will therefore be to translate the infinite antichain in $(\mathbb{D}(R), \subseteq)$ into an infinite antichain in $(\Sigma^*, \leq_T)$. We turn to the details, starting with the Rado WQO.

4.2 Witness Language

Rado Order. Our presentation of the Rado WQO [38] follows [35]. The *Rado set* is the upper diagonal, $R = \{(c, r) \mid c < r\} \subseteq \mathbb{N}^2$. The Rado WQO $\leq_R \subseteq R \times R$ is defined by:

$$(c_1, r_1) \leq_R (c_2, r_2), \qquad \text{if} \qquad r_1 \leq c_2 \ \vee \ (c_1 = c_2 \ \wedge \ r_1 \leq r_2) \ .$$

Given an element (c, r), we call c the *column* and r the *row*, as suggested by Figure 1(left). Columns will play an important role and we denote column i by $C_i = \{(i, r) \mid i < r\} \subseteq R$. To arrive at a larger element in the Rado WQO, one can increase the row while remaining in the same column, or move to the rightmost column of the current row, and select an element to the right, Figure 1(middle).

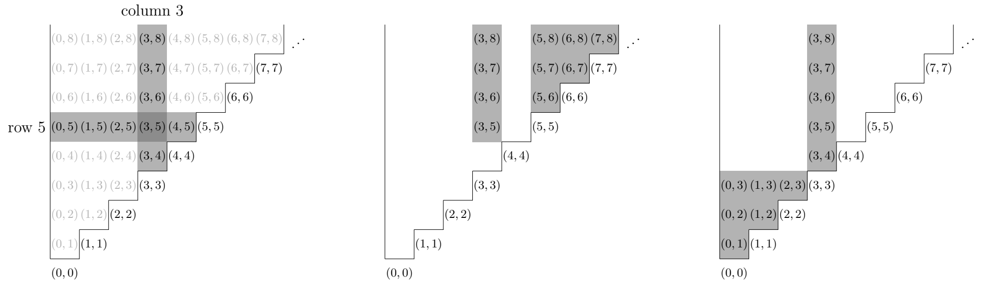


Figure 1 Rado order with the column and row of $(3, 5)$ marked (left), with the elements larger than $(3, 5)$ marked (middle), and with the downward closure of column 3 marked (right).

It is not difficult to see that $(R, \leq_R)$ is a WQO [38]. In an infinite sequence, either the columns eventually plateau out, in which case the rows lead to comparable elements, or the columns grow unboundedly, in which case they eventually exceed the row in the initial pair. The interest in the Rado WQO is that the WQO property is lost when moving to $(\mathbb{D}(R), \subseteq)$. This failure is due to the following well-known fact.

▶ **Lemma 23** ([22], Proposition 4.2). *$\{\downarrow C_i \mid i \in \mathbb{N}\}$ is an infinite antichain in $(\mathbb{D}(R), \subseteq)$.*

To see the lemma, we illustrate the downward closure of a column in Figure 1(right). Inclusion fails to be a WQO as each column C_i forms an infinite set that the downward closure $\downarrow C_j$ with $j > i$ cannot cover. Indeed, $\downarrow C_j$ only has the triangle to the bottom-left of column C_j available to cover C_i, and the triangle is a finite set. We will use exactly this difference between infinite and finite sets in our witness language. It will become clearer as we proceed.

Definition of T. The witness language is the language accepted by $U_R = (R, \leq_R, C_0, \Sigma, \delta, R)$. The set of states is the Rado set, the set of initial states is the first column, and the set of final states is again the entire Rado set. The latter means that a word is accepted as long as it admits a run. The letters in $\Sigma = \{a, \bar{a}, \mathit{zero}\}$ reflect the operation that the transitions $\delta \subseteq R \times \Sigma \times R$ perform on the states:

$$\begin{array}{rclcrcl} \delta((c, r), a) & = & (c+1, r+1) & \qquad & \delta((c+1, r+1), \bar{a}) & = & (c, r) \\ \delta((c+1, r), \mathit{zero}) & = & (0, c) & & \delta((0, r+1), \mathit{zero}) & = & (0, r) \ . \end{array}$$

We will explain the transitions in a moment, but remark that they are designed in a way that makes $\leq_R$ a simulation relation and hence U_R a WSTS.

▶ **Lemma 24.** *$T \in L(\mathit{WSTS})$.*

To develop an intuition to the language, consider

$$T \cap a^*.\bar{a}^*.\mathit{zero}^* \ = \ \{a^n.\bar{a}^n.\mathit{zero}^i \mid n, i \in \mathbb{N}\} \cup \{a^n.\bar{a}^k.\mathit{zero}^i \mid n, k, i \in \mathbb{N}, n - k > i\} \ .$$

Until reading the first *zero* symbol, the language keeps track of the (Dyck) balance of a and $\bar{a}$ symbols in a word. If the balance becomes negative, the word is directly rejected. If the balance is non-negative, it is the task of the *zero* symbols to distinguish a balance of exactly zero from a positive balance. Words with a balance of exactly zero get accepted regardless of how many *zero* symbols follow. Word that have a positive balance of $c > 0$ when reading the first *zero* get rejected after reading c-many *zero* symbols. As we show, this is enough to distinguish words with a balance of $c > 0$ from words with a balance of $d > 0$ for $d \neq c$, and thus obtain infinitely many classes in the Nerode quasi order. We turn to the details.

▶ **Proposition 25.** *$T \notin L(\mathit{detWSTS})$*

To prove $T \notin L(\text{detWSTS})$, we associate with each column C_i in the Rado WQO the *column language* $L_i = \{w \in \Sigma^* \mid \delta(C_0, w) = C_i\}$. It consists of those words that reach *all* states in C_i from the initial column C_0. The column languages are non-empty.

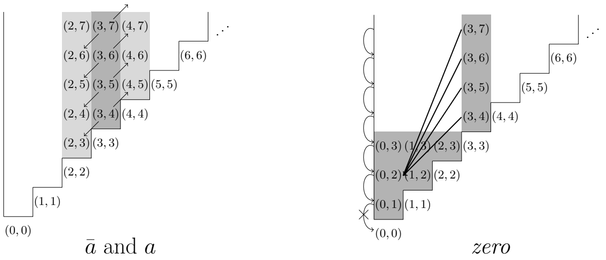
▶ **Lemma 26.** *$L_i \neq \emptyset$ for all $i \in \mathbb{N}$.*

We start from the entire initial column, meaning $\varepsilon \in L_0$. The transitions labeled by a move from all states in one column to all states in the next column, $L_i.a \subseteq L_{i+1}$. This already proves the lemma. The $\bar{a}$-labeled transitions undo the effect of the a-labeled transitions and decrement the column, $L_{i+1}.\bar{a} \subseteq L_i$. In the initial column, this is impossible, $\delta(C_0, \bar{a}) = \emptyset$. We illustrate the behaviour of a and $\bar{a}$ in Figure 2 (left).

By Lemma 23, the columns form an antichain in $(\mathbb{D}(R), \subseteq)$. The languages L_i translate this antichain into (actually several) antichains of the form we need. Combined, Lemmas 26, 27, and 22 conclude the proof of Proposition 25, and therefore Theorem 21.

▶ **Lemma 27.** *Every set $K \subseteq \Sigma^*$ with $|K \cap L_i| = 1$ for all $i \in \mathbb{N}$ is an antichain in $(\Sigma^*, \leq_T)$.*

In the rest of the section, we prove Lemma 27. The lemma claims that entire column languages are incomparable in the Nerode quasi order, so we write $L \not\sim_T K$ if for all $w \in L$ and all $v \in K$ we have $w \not\leq_T v$ and $v \not\leq_T w$. Difficult is the incomparability with L_0 stated in the next lemma. The proof will make formal the idea behind the *zero*-labeled transitions.



$\bar{a}$ and a *zero*

Figure 2 The effect of a and $\bar{a}$-labeled transitions on column 3 (left) and the effect of *zero*-labeled transitions on columns 0 and 3 (right).

▶ **Lemma 28.** *$L_0 \not\sim_T L_k$ for all $k > 0$.*

Proof. Let $w \in L_0$ and $v \in L_k$, meaning w leads to all states in column 0 while v leads to all states in column $k > 0$. It is easy to find a suffix that shows $v \not\leq_T w$, namely $\bar{a}$. Appending $\bar{a}$ to v leads to column C_{k-1}, and so $v.\bar{a} \in T$, while there is no transition on $\bar{a}$ from C_0, and so $w.\bar{a} \notin T$.

For $w \not\leq_T v$, we need the *zero* transitions. The idea is to make them fail in C_k for $k > 0$, and have no effect in C_0. The problem is that the states in C_k must simulate $(0, r)$ for $r \leq k$. The trick is to fail with a delay. Instead of having no effect in C_0, we let the *zero* transitions decrement the row. Instead of failing in C_k, we let the *zero* transitions imitate the behavior from $(0, k)$ and move to $(0, k-1)$. This is illustrated in Figure 2(right).

By working with column languages, the *zero* transitions fail in C_k with a delay as follows. We have $L_0.\mathit{zero} \subseteq L_0$ but $L_k.\mathit{zero} \not\subseteq L_0$, meaning from C_0 we again reach the entire column C_0, while from C_k we only reach the state $(0, k-1)$. The decrement behavior in the initial column allows us to distinguish the cases by exhausting the rows. Certainly, zero^{k-1} is enabled in large enough states of C_0, meaning $w.\mathit{zero}^k \in T$. The state $(0, k-1)$ reached by $v.\mathit{zero}$, however, does not enable corresponding transitions, $v.\mathit{zero}^k \notin T$. ◀

When executed in C_k with $k > 0$, the *zero* transitions resemble reset transitions [19]. An analogue of leaving C_0 unchanged despite decrements does not exist in the classical model. Moreover, reset nets are defined over $\mathbb{N}^k$ (an ω^2-WQO) as opposed to the Rado set. To conclude the proof of Lemma 27, we lift the previous result to arbitrary column languages.

▶ **Lemma 29.** *$L_i \not\sim_T L_j$ for all $i \neq j$.*

Proof. Let $i < j$ and consider $w \in L_i$ and $v \in L_j$. For $v \not\leq_T w$, we append $\bar{a}^j$, which is possible only from the larger column: $v.\bar{a}^j \in T$ but $w.\bar{a}^j \notin T$. For $w \not\leq_T v$, we append $\bar{a}^i$. Then $w.\bar{a}^i \in L_0$ while $v.\bar{a}^i \in L_k$ with $k > 0$. Now Lemma 28 applies and yields a suffix u so that $w.\bar{a}^i.u \in T$ but $v.\bar{a}^i.u \notin T$. ◀

The WSTS accepting the witness language T uses non-determinism only in the choice of the initial state. The transitions are deterministic. Moreover, the Rado WQO is embedded in every non-ω^2-WQO [8, Section 2]. Given the determinizability results from [16, Theorem 5], language T thus shows non-determinizability of WSTS with minimal requirements.

5 Downward-Compatible WSTS

We show that the regular separability and non-determinizability results we have obtained for upward-compatible WSTS so far can be lifted to downward-compatible WSTS (DWSTS). In DWSTS, smaller states simulate larger ones and the set of final states is downward closed. We lift our results by establishing general relations between the language classes $L(\text{WSTS})$, $L(\text{DWSTS})$, $L(\text{detWSTS})$, and $L(\text{detDWSTS})$. Figure 3 summarizes them.

$$\begin{tikzcd}
L(\text{detWSTS}) \arrow[r, "{\subsetneq,\ \text{Theorem 21}}"] \arrow[d, leftrightarrow, "{\not\subseteq_{rev},\ \not\supseteq_{rev},\ \text{Lemma 35}}"', "{=_{cmp},\ \text{Lemma 31}}"'] & L(\text{WSTS}) \arrow[d, leftrightarrow, "{=_{rev},\ \text{Lemma 30}}"] \\
L(\text{detDWSTS}) \arrow[r, "{\subsetneq,\ \text{Theorem 34}}"] & L(\text{DWSTS})
\end{tikzcd}$$

Figure 3 Relations between language classes.

Downward Compatibility. A *downward-compatible LTS* (DLTS) is an LTS $D = (Q, I, \Sigma, \delta, F)$ whose states are equipped with a quasi order $\leq \subseteq Q \times Q$ so that the following holds. The final states are downward closed, $\downarrow F = F$, and $\geq$ is a simulation relation. Recall that this means for all $p_1 \leq q_1$ and for all $q_2 \in \delta(q_1, a)$ there is $p_2 \in \delta(p_1, a)$ with $p_2 \leq q_2$. We denote the class of deterministic DLTS by detDLTS. We use $L(\text{DLTS})$ and $L(\text{detDLTS})$ to refer to the classes of all DLTS resp. detDLTS languages. If $\leq$ is also a WQO, we call D a *downward-compatible WSTS* (DWSTS).

Relations between L(DLTS) and L(ULTS). The languages accepted by DLTS are the reverse of the languages accepted by ULTS, and vice-versa. This is easy to see by reversing the transitions. Let $U = (Q, \leq, I, \Sigma, \delta, F)$ be an ULTS. We define $U^{rev} = (Q, \leq, F, \Sigma, \delta^{rev}, \downarrow I)$ to be its reversal. The initial and final states are swapped and the direction of the transitions is flipped, $\delta^{rev} = \{(p, a, q) \mid (q, a, p') \in \delta, p \leq p'\}$. Note that we close the initial states downwards and add transitions from states smaller than the original target. This corresponds to the assumption that the original transitions relate downward-closed sets. The construction can also be applied in reverse to get an ULTS D^{rev} from a DLTS D.

▶ **Lemma 30.** *If $U \in$ ULTS (WSTS), then $U^{rev} \in$ DLTS (DWSTS) and $L(U^{rev}) = L(U)^{rev}$. If $D \in$ DLTS (DWSTS), then $D^{rev} \in$ ULTS (WSTS) and $L(D^{rev}) = L(D)^{rev}$.*

The detDLTS languages are precisely the complements of the detULTS languages. For a detULTS or detDLTS $U = (Q, \leq, y, \Sigma, \delta, F)$, we define the complement $\overline{U} = (Q, \leq, y, \Sigma, \delta, \overline{F})$ by complementing the set of final states [37, Theorem 5].

▶ **Lemma 31.** *$U \in$ detULTS (detWSTS) iff $\overline{U} \in$ detDLTS (detDWSTS), and $L(\overline{U}) = \overline{L(U)}$.*

Behind this is the observation that, under determinism, $\leq$ is a simulation if and only if $\geq$ is [36, Theorem 3.3(ii)]. The details are in the full version [32].

5.1 Lifting Results

Regular Separability of DWSTS. We obtain the regular separability of disjoint DWSTS languages as a consequence of the previous results. More precisely, we need Lemma 30, Theorem 3, and the closure of the regular languages under reversal.

▶ **Theorem 32.** *Let $L_1, L_2 \in L(\text{DWSTS})$. We have $L_1 \mid L_2$ if and only if $L_1 \cap L_2 = \emptyset$.*

Non-Determinizability of DWSTS. To show that DWSTS cannot be determinized, recall our witness language T from Section 4. Surprisingly, we have the following.

▶ **Lemma 33.** *$T^{rev} \in L(\text{detDWSTS})$ and $\overline{T}^{rev} \in L(\text{detWSTS})$.*

For the first claim, recall that the witness language is accepted by the WSTS U_R. The DWSTS ${U_R}^{rev}$ has one minimal initial state, and transition images $\delta^{rev}(p, b)$ with one minimal element for all $p \in R$ and $b \in \Sigma$. Removing simulated states yields a deterministic DWSTS. The details are in the full version [32]. For the second claim, $\overline{T^{rev}} \in L(\text{detWSTS})$ by Lemma 31. But $\overline{T^{rev}} = \overline{T}^{rev}$, and so $\overline{T}^{rev} \in L(\text{detWSTS})$. Behind this is the fact that bijections commute with complements, and reversal is a bijection.

The lemma allows us to prove non-determinizability for DWSTS. Notably, we do not need a characterization for the languages of deterministic DWSTS.

▶ **Theorem 34.** *$\overline{T} \in L(\text{DWSTS}) \setminus L(\text{detDWSTS})$ and so $L(\text{DWSTS}) \neq L(\text{detDWSTS})$.*

Proof. By Lemma 33, $\overline{T}^{rev} \in L(\text{detWSTS})$. Lemma 30 yields $\overline{T} \in L(\text{DWSTS})$. Suppose $\overline{T} \in L(\text{detDWSTS})$. Then $T \in L(\text{detWSTS})$ by Lemma 31. This contradicts Proposition 25. ◀

5.2 Consequences

We have shown that neither upward- nor downward-compatible WSTS can be determinized. This does not yet rule out the possibility of determinizing an upward-compatible WSTS into a downward-compatible one, and vice versa. Given the correspondence in Lemma 30, we should allow the determinization to reverse the language. We now show that also this form of reverse-determinization is impossible: there are even deterministic languages that cannot be reverse-determinized. This is by Lemma 33, Proposition 25, and Theorem 34.

▶ **Lemma 35.** *$T^{rev} \in L(detDWSTS)$ but $T \notin L(detWSTS)$. Similarly, $\overline{T}^{rev} \in L(detWSTS)$ but $\overline{T} \notin L(detDWSTS)$*

After reversal, both witness languages T and $\overline{T}$ can be accepted by a deterministic WSTS. When it comes to separability, this means the results from [16] apply to them. A consequence of Lemma 35, however, is that there are WSTS languages that can neither be determinized nor reverse-determinized. An instance is $K = T.\#.\overline{T}^{rev}$ with # a fresh letter.

▶ **Lemma 36.** *$K \in L(WSTS)$, $K \notin L(detWSTS)$, and $K^{rev} \notin L(detDWSTS)$.*

When considering disjoint $K_1, K_2 \in L(\text{WSTS})$ that can neither be determinized nor reverse-determinized, the separability result from [16] does not apply. Theorem 3 is stronger and yields $K_1 \mid K_2$. The situation is similar for downward-compatible WSTS.

6 Conclusion and Future Work

We have shown that disjoint WSTS languages are always separated by a regular language. This strengthens the popular separability result from [16] by showing that the premise in that work (one language had to be accepted by a deterministic WSTS) is not needed. We have also shown that deterministic WSTS accept a strictly weaker class of languages than their non-deterministic counterparts, meaning the premise was a real restriction.

Behind our separability result is a closure of inductive invariants that adds limits of converging sequences, and the fact that the transition relation is compatible with limits. It would be interesting to formulate this in a topological setting [26, 27]. It would also be interesting to apply our invariant closure in settings where separability does not coincide with intersection emptiness and the complexity is open [6]. Finally, it would be interesting to develop compositional verification technology based on separability.

References

1 P. A. Abdulla, M. F. Atig, V. Dave, and S. Narayanan Krishna. On the separability problem of string constraints. In *CONCUR*, volume 171 of *LIPIcs*, pages 16:1–16:19. Dagstuhl, 2020.

2 P. A. Abdulla, K. Cerans, B. Jonsson, and Y.-K. Tsay. General decidability theorems for infinite-state systems. In *LICS*, pages 313–321. IEEE, 1996.

3 P. A. Abdulla, G. Delzanno, and L. Van Begin. Comparing the expressive power of well-structured transition systems. In *CSL*, volume 4646 of *LNCS*, pages 99–114. Springer, 2007.

4 P. A. Abdulla and B. Jonsson. Verifying programs with unreliable channels. In *LICS*, pages 160–170. IEEE, 1993.

5 G.Rozenberg A.Salomaa, editor. *Handbook of Formal Languages*. Springer, 1997. `doi:10.1007/978-3-642-59136-5`.

6 P. Baumann, R. Meyer, and G. Zetzsche. Regular separability in Büchi VASS. In *STACS*, volume 254 of *LIPIcs*, pages 9:1–9:19. Dagstuhl, 2023.

7 M. Blondin, A. Finkel, and P. McKenzie. Handling infinitely branching WSTS. In *ICALP*, volume 8573 of *LNCS*, pages 13–25. Springer, 2014.

8 P. Jančar. A note on well quasi-orderings for powersets. *IPL*, 72(5):155–160, 1999.

9 E. M. Clarke, T. A. Henzinger, H. Veith, and R. Bloem, editors. *Handbook of Model Checking*. Springer, 2018.

10 L. Clemente, W. Czerwiński, S. Lasota, and C. Paperman. Regular separability of Parikh automata. In *ICALP*, volume 80 of *LIPIcs*, pages 117:1–117:13. Dagstuhl, 2017.

11 L. Clemente, W. Czerwiński, S. Lasota, and C. Paperman. Separability of reachability sets of vector addition systems. In *STACS*, volume 66 of *LIPIcs*, pages 24:1–24:14. Dagstuhl, 2017.

12 L. Clemente, S. Lasota, and R. Piórkowski. Timed games and deterministic separability. In *ICALP*, volume 168 of *LIPIcs*, pages 121:1–121:16. Dagstuhl, 2020.

13 P. Cousot. *Principles of abstract interpretation*. MIT Press, 2021.

14 W. Czerwiński, P. Hofman, and G. Zetzsche. Unboundedness problems for languages of vector addition systems. In *ICALP*, volume 107 of *LIPIcs*, pages 119:1–119:15. Dagstuhl, 2018.

15 W. Czerwiński and S. Lasota. Regular separability of one counter automata. In *LICS*, pages 1–12. IEEE, 2017.

16 W. Czerwiński, S. Lasota, R. Meyer, S. Muskalla, K. Narayan Kumar, and P. Saivasan. Regular separability of well-structured transition systems. In *CONCUR*, volume 118 of *LIPIcs*, pages 35:1–35:18. Dagstuhl, 2018.

17 W. Czerwiński, W. Martens, L. van Rooijen, M. Zeitoun, and G. Zetzsche. A characterization for decidable separability by piecewise testable languages. *DMTCS*, 19(4), 2017.

18 W. Czerwiński and G. Zetzsche. An approach to regular separability in vector addition systems. In *LICS*, pages 341–354. ACM, 2020.

19 C. Dufourd, A. Finkel, and Ph. Schnoebelen. Reset nets between decidability and undecidability. In *ICALP*, pages 103–115. Springer, 1998.

20 A. Finkel. A generalization of the procedure of Karp and Miller to well structured transition systems. In *ICALP*, volume 267 of *LNCS*, pages 499–508. Springer, 1987.

21 A. Finkel and J. Goubault-Larrecq. Forward analysis for WSTS, part II: complete WSTS. In *ICALP*, volume 5556 of *LNCS*, pages 188–199. Springer, 2009.

22 A. Finkel and J. Goubault-Larrecq. Forward analysis for WSTS, part II: complete WSTS. *LMCS*, 8(3), 2012.

23 A. Finkel and Ph. Schnoebelen. Well-structured transition systems everywhere! *TCS*, 256(1-2):63–92, 2001.

24 G. Geeraerts, J.-F. Raskin, and L. Van Begin. Well-structured languages. *Acta Informatica*, 44(3-4):249–288, 2007.

25 A. R. Gingras. Convergence lattices. *RMJ*, 6(1):85–104, 1976.

26 J. Goubault-Larrecq. On Noetherian spaces. In *LICS*, pages 453–462. IEEE, 2007.

27 J. Goubault-Larrecq. Noetherian spaces in verification. In *ICALP*, volume 6199 of *LNCS*, pages 2–21. Springer, 2010.

28 J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, 1979.

29 T. Jech. *Set Theory*. Springer, 2002.

30 C. B. Jones. Tentative steps toward a development method for interfering programs. *ACM TOPLAS*, 5(4):596–619, 1983.

31 M. Kabil and M. Pouzet. Une extension d'un théorème de P. Jullien sur les âges de mots. *RAIRO Theor. Informatics Appl.*, 26:449–482, 1992.

32 Eren Keskin and Roland Meyer. Separability and non-determinizability of WSTS (full version). *CoRR*, abs/2305.02736, 2023. `doi:10.48550/arXiv.2305.02736`.

33 J. Leroux and S. Schmitz. Demystifying reachability in vector addition systems. In *LICS*, pages 56–67. IEEE, 2015.

34 G. Markowsky. Chain-complete posets and directed sets with applications. *Algebra Universalis*, 6:53–68, December 1976. `doi:10.1007/BF02485815`.

35 E. C. Milner. *Basic WQO- and BQO-Theory*, pages 487–502. Springer, 1985.

36 R. Milner. An algebraic definition of simulation between programs. In *IJCAI*, pages 481–489. Kaufmann, 1971.

37 M. O. Rabin and D. Scott. Finite automata and their decision problems. *IBM Journal of Research and Development*, 3(2):114–125, 1959.

38 R. Rado. Partial well-ordering of sets of vectors. *Mathematika*, 1(2):89–95, 1954.

39 R. S. Thinniyam and G. Zetzsche. Regular separability and intersection emptiness are independent problems. In *FSTTCS*, volume 150 of *LIPIcs*, pages 51:1–51:15. Dagstuhl, 2019.

40 G. Zetzsche. Separability by piecewise testable languages and downward closures beyond subwords. In *LICS*, pages 929–938. ACM, 2018.