Thin MSO with a probabilistic path quantifier

Mikołaj Bojańczyk[1]

1 University of Warsaw

Abstract

This paper is about a variant of MSO on infinite trees where:

- there is a quantifier "zero probability of choosing a path $\pi \in 2^\omega$ which makes $\varphi(\pi)$ true";
- the monadic quantifiers range over sets with countable topological closure.

We introduce an automaton model, and show that it captures the logic.

1998 ACM Subject Classification F.4.1 Mathematical Logic

Keywords and phrases Automata, MSO, infinite trees, probabilistic temporal logics

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

1 Introduction

The ambient topic of this paper is MSO on infinite binary trees, extended by a quantifier zeroπ $\varphi(\pi)$ which says that there is zero probability of choosing a path π in the tree so that $\varphi(\pi)$ is true. Here we assume that each bit (i.e. turn) in the path is chosen independently at random. This logic was introduced by Michalewski and Mio in [11], where the decidability of satisfiability was left open.

That satisfiability question is not solved here, but we make a small step in its direction. We consider a fragment of the logic, called TMSO+zero, standing for *thin* MSO+zero. In this fragment, the monadic set quantifiers are restricted to sets which are thin in the following sense: a set of nodes is *thin* if there are countably many paths which visit it infinitely often. For example, every path (when seen as a set of nodes) is thin, and every finite set is thin. In the logic TMSO+zero, one has existential and universal quantification over nodes and thin sets of nodes, as well as the probabilistic path quantifier zero. Being thin is definable in MSO, and therefore without the zero quantifier, the logic would be a special case of MSO, and with the zero quantifier it is a special case of the logic from [11].

The contribution of this paper is the definition of an automaton model, called zero automata, and a proof that every formula of TMSO+zero can be effectively translated to an equivalent zero automaton.

Motivation.

The first source of motivation for this paper is the study of probabilistic temporal logics [1, 9, 14, 5]. An important example is the logic PCTL. It is an open problem whether this logic has decidable satisfiability. Much of the difficulty stems from the ability of talking about probabilities like 1/2 or 1/3. If one can only compare probabilities to 0 or 1, which is in the spirit of our logic TMSO+zero, then we get *qualitative* PCTL, whose satisfiability was shown decidable by Brázdil, Forejt, Kretínský and Kucera in [5]. Actually, the qualitative fragment of PCTL, as well as stronger qualitative logics like PCTL*, can be straightforwardly formalised in TMSO+zero, and therefore, by the main result of this paper, translated into zero automata. Another example that we discuss later in the paper is the probabilistic

Conference title on which this volume is based on.
Editors: Billy Editor and Bill Editors; pp. 1–35
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

version of tree automata by Carayol, Haddad and Serre [7]; these are also a special case of zero automata.

The second source of motivation is trying to find a robust classes of languages of infinite words or trees which remains decidable (e.g. with respect to satisfiability). The point of departure is MSO, with its famous decidability results by Büchi [6] and Rabin [12]. One way of departing from that point is to add unary predicates, e.g. extending MSO over ω-words by a predicate "x is a position of the form $n!$", see [13] for a survey. Another way is to add new quantifiers. Due to the strength of MSO, it is not so easy to come up with a quantifier extending MSO that is not obviously undecidable, and yet not already definable in MSO. For example, a nice quantifier is "there exist uncountably many sets with a given property" – but as shown in [2], this quantifier does not add to the expressive power of MSO. A logic that *does* properly extend MSO is MSO+U, which is an extension of MSO by a quantifier which can say that a given property is true for finite sets of unbounded size. The logic is itself undecidable, but has many decidable fragments, typically variants of weak MSO. See [4] for a survey of MSO+U and related logics, including the cost logics of Colcombet [8]. The logic studied in this paper, TMSO+zero, is another example of a logic that is not contained in MSO (and even contains MSO, if the logic is extended by allowing an outermost layer of non-weak existential set quantifiers, which does not affect decidability of satisfiability).

Acknowledgment. I would like to thank Henryk Michalewski and Matteo Mio for introducing me into this area, and for their many valuable comments and suggestions.

2 The logic and the automaton

This section describes the two main models used in the paper: the logic TMSO+zero and zero automata. The following sections discuss how the logic is translated into the automaton.

Tree notation.

The logics and automata of this paper describe properties of possibly infinite binary labelled trees. We treat a node in a tree as a sequence in 2^*, with 2 denoting the set of directions $\{0,1\}$. Define a *tree* over an alphabet Σ to be a partial function $t : 2^* \to \Sigma$ whose domain is closed under prefixes. The special case when function is total is called a *complete tree*, but we do allow incomplete trees, e.g. trees with finite domains. We use standard terminology for trees: node, root, left child, right child, leaf, ancestor and descendant. In our definition, a node might have a right child but not a left child. We write treesΣ for the set of trees over Σ.

Probability measure over paths.

A *path* is defined to be a sequence in 2^ω, which is viewed as an infinite sequence of left or right turns. An equivalent definition is that a path is an ancestor-closed set of nodes that is totally ordered by the ancestor relation. When saying that a path is contained in a set of nodes, or contains a node, the second definition is used. When talking about the probability of a subset of 2^ω we use the *coin-flipping* measure, i.e. we assume that each bit is chosen independently at random, with 0 and 1 having equal probability. The probability is defined at least for all Borel subsets of 2^ω.

▶ **Definition 1.** We say a set $\Pi \subseteq 2^\omega$ has *zero probability* if it is contained in a Borel set with coin-flipping measure zero.

The sets of paths that will appear in the logic TMSO+zero will always be Borel, so the closure under subsets in the above definition will not play much of a role.

2.1 The logic

Before defining the logic TMSO+zero, we discuss the probability-free fragment TMSO.

Thin MSO without the zero quantifier

A set of nodes $X \subseteq 2^*$ is called *thin* if its *closure* defined by

$$\bar{X} \stackrel{\text{def}}{=} \{\pi \in 2^\omega : \pi \text{ passes through infinitely many nodes from } X\}$$

is countable. For example, every finite set is thin, because it has empty closure, and every path is thin, when viewed as a set of nodes, because its closure has one path. Thin sets are closed under arbitrary intersections and finite unions, but not under countable unions, because the countable set of all nodes has all paths in its closure, and is therefore not thin.

The logic *thin* MSO, denoted by TMSO, is the variant of MSO as in Rabin's theorem, except that set quantifiers range only over thin sets. The syntax of the logic is the same as for MSO from Rabin's theorem: there are two types of variable in the logic: *node variables*, which range over nodes in the domain of the input tree, and *(thin) set variables*, which range over thin subsets of the domain of the input tree. There are binary predicates for the left and right child relations, and there is a unary predicate for every label in the input alphabet. By the Cantor-Bendixson theorem, a set of nodes X is thin if and only if one cannot find a subset $Y \subseteq X$ such that Y, when ordered by the descendant relation, is a complete binary tree. Since this alternative characterisation can be formalised in MSO, it follows that TMSO is a fragment of MSO in terms of expressive power. On the other hand, TMSO is at least as expressive as WMSO with path quantifiers.

As far as the author knows, the logic TMSO was not considered explicitly in the literature so far, and it might be interesting to examine its expressive power, e.g. prove that it is strictly weaker than MSO and maybe, in the long run, find an algorithm which inputs a formula of MSO and decides if the formula is equivalent to some formula in TMSO. This investigation, however, is not the topic of the present paper. The present paper is about extending TMSO with a quantifier for zero probability.

Thin MSO with the zero quantifier

We now define the main topic of this paper, i.e. the logic TMSO+zero. First we explain why our point of departure for adding the zero quantifier is TMSO and not some other fragment of MSO. The reason is that TMSO is the strongest logic we could find such that the set quantifiers commute with the probabilistic quantifier in a way which will be made more precise in Section 6. The key observation reason is this: if the domain of the input tree is thin, then it has countably many paths, and therefore the zero quantifier can be eliminated because it always says "yes".

A parameter in the definition of TMSO+zero is a family zero of subsets of 2^ω. The example we have in mind is that zero is the sets with zero probability according to Definition 1, but the results will also work for other choices of zero. The logic TMSO+zero is the extension of the logic TMSO defined above, by adding a quantifier, called zero, which binds a thin set variable π, and such that

$$\mathsf{zero}\pi \ \varphi(\pi)$$

is true if zero contains the set of paths π which are contained in the domain of the input tree and make $\varphi(\pi)$ true, assuming that a path is treated as a set of nodes. (Formally speaking, the path π is seen as a set of nodes when evaluating $\varphi(\pi)$, and as an element of 2^ω when measuring how many paths π make $\varphi(\pi)$ true.)

Example 1. Consider an alphabet $\{a, b\}$. The following formula says that zero contains the set of paths that visit at least one a:

$$\mathsf{zero}\pi\ \exists x\ (x \in \pi \wedge a(x)).$$

If the parameter zero is prefix independent (see Definition 3 for a more precise treatment) and does not contain the set 2^ω of all paths, then the above formula is equivalent to $\forall x\ \neg a(x)$, and therefore the zero quantifier can be avoided. ☐

Example 2. Consider an alphabet $\{a, b\}$. The following formula says that zero contains the set of paths which visit b finitely often:

$$\mathsf{zero}\pi\ \exists x \big(x \in \pi \wedge \forall y\ (y \geq x \wedge y \in \pi \Rightarrow b(x))\big)$$

If zero is our guiding example of zero probability, the negation of the above formula says that the Büchi condition is satisfied with probability one. As shown in [7], Theorem 21, the property above is not definable in MSO. ☐

Example 3. The reduction from qualitative PCTL* in Theorem 5 from [11] produces formulas where set quantification is only used for paths. Therefore, qualitative PCTL* is a special case of TMSO+zero. ☐

Beyond Thin MSO with the zero quantifier.

In the logic TMSO+zero, the set variables are restricted to thin sets. The obvious question is about the more general case, where set variables range over arbitrary sets of nodes, not necessarily thin ones. As mentioned in the introduction, the more general logic was introduced in [11], under the name MSO + $\forall_\pi^{=1}$, and the authors asked about decidability of its satisfiability problem. A long term project for this research is to find out if the satisfiability problem for the more general logic is decidable – or not. In this paper we only begin the project, by studying the thin variant. One scenario is that the thin variant is decidable, but the non-thin variant is undecidable, which would be similar to the situation for MSO+U, where weak variants are decidable, but the full logic is undecidable. However, one should not take the analogy with MSO+U too far: e.g. the thin variant of MSO+U would already be undecidable, because MSO+U is undecidable already for ω-words.

Another natural version of MSO with probability would be to choose a subset of 2^* at random, with each node chosen independently, and then have a quantifier that says there is zero probability of finding a subset with a given property. This logic was proved undecidable in [11], already for ω-words (which can be seen as a special case of TMSO), and the undecidability proof works also for formulas of the form

there is zero probability of choosing a set $X \subseteq \mathbb{N}$ which makes $\varphi(X)$ true,

where $\varphi(X)$ is a formula of first-order logic that defines a set of ω-words over alphabet 2. Therefore, it seems that this kind of probabilistic quantifier is doomed to undecidability.

2.2 The automaton

Having defined the logic TMSO+zero, we define our main automaton model, which is called a zero *automaton*. Like in the logic TMSO+zero, a parameter of the semantics for the automaton is a family zero of subsets of 2^ω. The idea is that the automaton extends a nondeterministic parity automaton with the ability to say that the set of paths satisfying the parity condition belongs, or does not belong, to zero.

▶ **Definition 2.** The syntax of a zero automaton is a tuple

$$\underbrace{Q}_{\text{states}} \qquad \underbrace{\Sigma}_{\text{input alphabet}} \qquad \underbrace{I \subseteq Q}_{\text{initial states}} \qquad \underbrace{\bigcup_{C \subseteq 2} \delta_C \subseteq Q \times \Sigma \times Q^{|C|}}_{\text{transitions}},$$

with all components finite, together with a total order on Q and four subsets

$$Q_{\text{all}}, Q_{\text{zero}}, Q_{\text{nonzero}}, Q_{\text{seed}} \subseteq Q.$$

The idea behind the transitions is that $\delta_{\{0,1\}}$ is used for those nodes which have both children defined, but e.g. $\delta_{\{1\}}$ is used for nodes where only the right child is defined, and $\delta_\emptyset$ is used for leaves.

The semantics are defined as follows. The automaton is run on a tree over the input alphabet, which might not necessarily be complete. A *run* of the automaton is a tree labelled by states with the same domain as the input tree, which is consistent with the transition relation in the following sense: if a node x is in the domain, and we define

$$C \stackrel{\text{def}}{=} \{i \in 2 : xi \text{ is in the domain}\}$$

then there must be a transition in δ_C which relates the state in x, the label of x in the input tree, and the states in the children of x that are in the domain. A tree is accepted if it admits a run which has the initial state in the root and is accepting in the following sense. Define the *maxinf state* on a path in a run to be the maximal state that appears infinitely often on the path. When talking about a maximal state, we refer to the total order on states that is given in the syntax of the automaton. A run ρ is accepting if all of the following conditions hold, assuming that $\mathsf{paths}\,\rho \subseteq 2^\omega$ denotes the set of paths contained in ρ:

1. **all paths acceptance condition:** every path from $\mathsf{paths}\,\rho$ has maxinf in Q_{all}; and
2. **zero acceptance condition:** zero contains the set of paths from $\mathsf{paths}\,\rho$ which have maxinf state in Q_{zero}; and
3. **nonzero acceptance condition:** for every node x in the run with state $q \in Q_{\text{seed}}$:

$$\mathsf{zero} \not\ni \{\pi \in \mathsf{paths}\,\rho : \begin{cases} \pi \text{ passes through } x, \text{ and} \\ \pi \text{ sees only states} < q \text{ after } x, \text{ and} \\ \pi \text{ has maxinf state in } Q_{\text{nonzero}} \end{cases} \}$$

An automaton is called *zeroless* if Q_{zero} is empty (which makes the zero condition vacuously true) and *seedless* if there are no seed states, i.e. Q_{seed} is empty (which makes the nonzero condition vacuously true). In particular, a zeroless and seedless automaton is the same thing as a parity automaton, which proves the zero automata are at least as powerful as MSO.

Example 4. Assume that zero is probability zero as in Definition 1. Consider the special case of a zero automaton where Q_{all} is all states and Q_{seed} is empty. A run is accepting if and only if there is zero probability of having maxinf state in Q_{zero}. Equivalently, the probability of having maxinf state outside Q_{zero} is one. Languages recognised by such automata are the *qualitative tree languages* from [7]. The class of *positive tree languages* from [7] is obtained when Q_{all} and Q_{zero} are empty, and the initial state is used only once in the root, is maximal in the total order, and is the unique seed state. ☐

3 Fat Cantor

In this section, we illustrate the logic and automaton with an extended example. Let us fix zero to be probability zero according to Definition 1. Define the *fat Cantor language* to be the set of complete trees over the alphabet $\{a, b\}$ which satisfy the following property:

$$\underbrace{\neg\mathsf{zero}\pi\big(\forall x\ x \in \pi \wedge b(x)\big)}_{\text{nonzero probability of avoiding } a} \qquad \wedge \qquad \underbrace{\forall x \exists y\ y \geq x \wedge a(y)}_{a\text{'s are dense}}$$

Note that "avoiding a" is a Borel property of paths, and therefore "nonzero probability of avoiding a" means that the sets of paths avoiding a have defined positive probability. This argument will be true in general for our logic – for every fixed input tree, any property of paths definable in the logic will be Borel, and therefore not belonging to zero will mean that it there is defined and positive probability.

The fat Cantor language is nonempty. To construct a tree in the fat Cantor language, choose a fast growing sequence of natural numbers

$$n_1 < n_2 < n_3 < \cdots$$

and then choose a tree (which is unique up to reordering siblings) where a labels are found only at depths from the sequence above, and every node at depth n_i has a unique descendant at depth n_{i+1} with label a. If the sequence (n_i) grows fast enough, then there is nonzero probability of avoiding a. Let us now argue that the fat Cantor language contains no regular tree, i.e. no tree with finitely many nonisomorphic subtrees. Suppose then that t is a regular tree, with n distinct subtrees. If a's are dense in this tree, it follows from regularity that every node has a descendant at distance at most n that has label a. This means there is some constant $\epsilon > 0$ such that for every interval $I \subseteq \mathbb{N}$ of n consecutive positions, the probability of a path visiting a at depth from I is at least ϵ. These events are independent for disjoint intervals, and therefore the probability of seeing a at least once, and even infinitely often, is 1. Summing up: the fat Cantor language is nonempty but contains no regular trees.

Fat Cantor automaton

We now show a zero automaton which recognises the fat Cantor language described above. To simplify notation, we define an automaton which works only on complete trees, i.e. it recognises the intersection of the fat Cantor language with the set of complete trees. In particular, when talking about transitions, we only consider transitions δ_C for $C = \{0, 1\}$.

The input alphabet is $\{a, b\}$. The automaton has four states, totally ordered as follows:

$$\underbrace{q_a}_{\text{already saw } a} \quad < \quad \underbrace{q_1}_{\text{searching for } a} \quad < \quad \underbrace{q_2}_{\text{not searching for } a} \quad < \quad \underbrace{q_0}_{\text{initial state}}$$

The automaton begins in state q_0 in the root, this state will not be visited again during the run. When the automaton is in state q_i with $i \in \{0, 1, 2\}$ and it reads a node with label

b, then it sends q_1 to some child and q_2 to the other child, as witnessed by the following transitions:

$$(q_i, b, q_j, q_k) \quad \text{for } i \in \{0,1,2\} \text{ and } \{j,k\} = \{1,2\}.$$

Choosing which child gets q_1 and which child gets q_2 is the only source of nondeterminism in this automaton. When the automaton sees letter a, it sends q_a to both children regardless of its current state, and q_a is a sink state that cannot be left, as witnessed by the following transitions:

$$(q, a, q_a, q_a) \quad \text{for all } q \in Q \qquad (q_a, a, q_a, q_a) \qquad (q_a, b, q_a, q_a)$$

Since q_0 is used only once in the root, and q_a is a sink state, it follows that on every path either q_a is seen from some point on, or q_a is never seen and the maxinf state is one of q_1, q_2. The acceptance condition is defined by the following sets:

$$Q_{\text{all}} = \{q_a, q_2\} \qquad Q_{\text{zero}} = \emptyset \qquad Q_{\text{nonzero}} = \{q_1, q_2\} \qquad Q_{\text{seed}} = \{q_0\}$$

Because Q_{zero} is empty, every run satisfies the zero acceptance condition. The state q_0 appears only once in the root, and therefore it is never used as a maxinf state. By choice of Q_{all}, the state q_1 is forbidden as a maxinf state, which means that in an accepting run, every path eventually stabilises on either q_a or q_2. Since the only way of leaving q_1 is by seeing an a letter, it follows that a's must be dense. The only seed state is the initial state, which is used only once in the root, and is also the most important state. Therefore, a run satisfies the nonzero acceptance condition if and only if its there is nonzero probability of having maxinf state in $\{q_1, q_2\}$, which means there is nonzero probability of avoiding a.

4 From logic to automata

The main technical result of this paper is that every formula of TMSO+zero can be effectively translated to an equivalent zero automaton. The result works not just for zero probability, but for other choices of zero, as described in the following definition.

▶ **Definition 3.** For a family zero of subsets of 2^ω, consider the following properties:

1. **σ-ideal:** zero is closed under subsets and countable union;
2. **atomless:** zero contains all singletons;
3. **prefix independence:** every set $\Pi \subseteq 2^\omega$ satisfies

$$\Pi \in \mathsf{zero} \Leftrightarrow i\Pi \in \mathsf{zero} \qquad \text{for every } i \in 2$$

4. **recurrent nonzero:** there is a zero automaton which recognises the language

$$\{t \in \mathsf{trees}\{1,2,3\} : \text{for every subtree, the set of paths with maxinf 2 is} \notin \mathsf{zero}\}$$

In the recurrent nonzero condition, it is important that the trees are not necessarily complete. For such a tree, a subtree is obtained by shifting the root to some node in the domain. In particular, if a tree belongs to the language from the recurrent nonzero condition, then it cannot have any leaves.

Here is the main result of this paper.

▶ **Theorem 4.** *Let* zero *be a family of subsets of 2^ω satisfying conditions 1-4 in Definition 3. Then for every formula of* TMSO+zero *one can compute an equivalent* zero *automaton.*

The proof has three steps. In Section 5, we show closure properties of languages recognised by zero automata, of which the most interesting is closure under intersection. In Section 6, we show that the logic TMSO+zero has the same expressive power as a certain transducer model. In Section 7, we complete the proof of the theorem, by translating transducers into zero automata. The results in Section 5 and 6 only use properties 1-3 in Definition 3, while Section 7 uses also property 4.

The following corollary shows the main application of Theorem 4.

▶ **Corollary 5.** *Let* zero *be the subsets of* 2^ω *that have zero probability in the sense of Definition 1. Then for every formula of* TMSO+zero *one can compute an equivalent* zero *automaton.*

Other examples of zero which can be shown to satisfy the assumptions of Theorem 4 include "countable sets of paths [2]" and "meagre sets of paths [10]". These other examples are less interesting because they can already be formalised in MSO alone, i.e. parity automata are sufficient. Theorem 4 can be seen as an alternative way of recovering the results from [2, 10]: one only needs to check that the assumptions of Theorem 4 are satisfied for a particular choice of zero, and that zero automata can be captured by MSO. In view of the results from [2, 10], we have only one example of zero that satisfies the assumptions of Theorem 4, and which strictly extends MSO, namely probability zero.

5 Closure properties of zero automata

This section is about closure properties of the class of languages recognised by zero automata. We show that this class is closed under positive Boolean operations – with intersection being by far the more interesting case. We do not know if languages recognised by zero automata are closed under complementation. If they would be, then zero automata would have exactly the same expressive power as full MSO+zero.

Define a *Mealy machine* to be a deterministic finite automaton on words over some input alphabet Σ, where every transition is labelled by a letter from some output alphabet Γ. Such a machine can be run on a finite word, yielding a length preserving function $\Sigma^* \to \Gamma^*$, it can also be run on an ω-word, yielding a function $\Sigma^\omega \to \Gamma^\omega$, or finally it can be run on all paths in a tree, yielding a function $\mathsf{trees}\Sigma \to \mathsf{trees}\Gamma$ which does not change the domain of the tree. The last case will be called a *tree transducer recognised by a Mealy machine.*

▶ **Lemma 6.** *Languages recognised by* zero *automata are closed under union, as well as images and inverse images under tree transducers recognised by Mealy machines.*

Proof sketch. The lemma does not require any closure properties from the set zero. For union, we use disjoint union of automata (and gluing the initial state). For images use nondeterminism, and for both images and inverse images use a Cartesian product construction to simulate the Mealy machine in the state space of the zero automaton. Note that state spaces in zero automata are ordered. Therefore we impose some random total order on a Mealy machine, and in the Cartesian product we use a lexicographic ordering, with the order on the original zero automaton being more important. ◀

We now show another closure property, which is closure under factorisations, as described below. Define a *factor* to be a set of nodes that is connected with respect to the child relation. In particular, a factor has a unique *root*, i.e. a unique node which is least with respect to the descendant ordering. If X is a factor, then define the *restriction to* X of a tree t to be the tree obtained from t by keeping only the nodes from X. We now show that if L is a language recognised by a zero automaton, then there is a zero automaton which inputs a tree together with a decomposition into disjoint factors, and checks that L contains every tree obtained by restricting the input tree to one of the factors in the partition.

We begin by describing how a decomposition into factors is given on the input. If X is a set of nodes, then define an X-factor to be a set of nodes obtained by taking some $x \in X$ and adding all descendants y such that $(x..y]$ is disjoint with X, where $(x..y]$ denotes proper descendants of x that are (not necessarily proper) ancestors of y. By abuse of notation, we define an X-factor of a tree t to be any tree obtained from t by restricting it to some X-factor. Finally, if X is a set of nodes in a tree $t \in \mathsf{trees}\Sigma$, then define $t \otimes X \in \mathsf{trees}(\Sigma \times 2)$ to be the tree obtained from t by extending the label of each node by a bit indicating membership in X.

▶ **Lemma 7** (Factorisation Lemma). *Assume that* zero *satisfies conditions 1-3 in Definition 3. If $L \subseteq \mathsf{trees}\Sigma$ is recognised by a* zero *automaton, then so is*

$$\{t \otimes X : t \in \mathsf{trees}\Sigma \text{ and } X \text{ is a set of nodes in } t \text{ such that } L \text{ contains every } X\text{-factor of } t\}$$

The main idea in the proof is that to use the "nested" character of the nonzero acceptance condition; here by nesting we mean that the paths contributing to the nonzero condition are cut off whenever a more important state is seen.

We finish this section by stating the most challenging result, which is closure under intersection, as stated in the following lemma.

▶ **Lemma 8** (Intersection Lemma). *Assume that* zero *satisfies conditions 1-3 in Definition 3. Then languages recognised by* zero *automata are closed under intersection.*

The proof has several steps. One of these steps, namely the first step, is showing that languages recognised by zero automata are closed under intersection with languages recognised by zero automata which do not use the nonzero acceptance condition. The first step uses McNaughton's Latest Appearance Record construction.

6 Transducers

To prove Theorem 4, we use a transducer characterisation of the logic TMSO+zero. The transducer characterisation is an "if and only if" characterisation, unlike the translation in the main Theorem 4.

Transducers

Define a *tree transducer* to be any function $\mathsf{trees}\Sigma \to \mathsf{trees}\Gamma$ which does not change the domain of the input tree. Our goal is to show each language definable MSO+zero can be described by composing transducers of certain basic types. To model a language as a transducer, we use the following definition.

▶ **Definition 9.** For a tree language $L \subseteq \mathsf{trees}\Sigma$, define

$$\mathsf{trans}L : \mathsf{trees}\Sigma \to \mathsf{trees}2,$$

called the *characteristic transducer of* L, to be the transducer which labels each node of the input tree by a bit saying whether or not the subtree rooted in that node belongs to L.

We define the *combination* $t_0 \otimes t_1$ of two trees t_0, t_1 over possibly different alphabets Σ_0, Σ_1 but with equal domains, to be the unique tree over $\Sigma_0 \times \Sigma_1$ which projects to each

t_i on the i-th coordinate. In the following theorem, composition of transducers is defined as for functions, while the combination of two transducers f_1, f_2 with the same input alphabet but possibly different output alphabets is the transducer $t \mapsto f_1(t) \otimes f_2(t)$.

▶ **Theorem 10.** *Assume that* zero *has the closure properties 1-3 from Theorem 4. Then a tree language is definable in* TMSO+zero *if and only if its characteristic transducer belongs to the smallest class of transducers which is closed under composition and combination, and which contains the following transducers:*

1. **Zero base.** *The characteristic transducers of all languages of the form:*

$$\mathsf{Z}_n \stackrel{def}{=} \{t \in \mathsf{trees}\{1, \ldots, n, \bot\} : \mathsf{zero} \ni \{\pi \in \mathsf{paths}\, t : \begin{cases} \pi \text{ does not visit } \bot\text{, and} \\ \pi \text{ has even maxinf} \end{cases}\}\}$$

2. **Zeroless base.** *The characteristic transducers of all languages definable in* TMSO*.*
3. **Child number transducer.** *Transducers of the form* $\mathsf{trees}\Sigma \to \mathsf{trees}2$ *which map each node to its child number, with the convention that the root gets label* 0*.*
4. **Mealy machine on trees.** *Transducers recognised by Mealy machines.*

The difficult implication is from logic to transducers; here we use the composition method. Intuitively speaking, the above theorem shows that formula of TMSO+zero can be decomposed into parts that do not talk about zero at all, and into the very basic property Z_n.

7 From transducers to zero automata.

In this section we complete the proof of Theorem 4, by showing that the transducers from the previous section can be compiled into zero automata. We say that a tree transducer f is *recognised* by a zero automaton if there is a zero automaton recognising the set of trees $t \otimes f(t)$ where t ranges over all input trees for the tree transducer.

▶ **Lemma 11.** *Transducers recognised by* zero *automata are closed under composition, combination and include the child number transducers, transducers induced by Mealy machines, and the characteristic transducers of all languages definable in* TMSO*.*

Proof sketch. For composition, the automaton guesses the intermediate result, and checks both underlying transducers in parallel, using the Intersection Lemma. Combination also uses intersection. For the child-number transducers, Mealy machines and characteristic transducers of languages definable in TMSO, one observes that their corresponding languages are definable in MSO, and zero automata generalise nondeterministic parity tree automata. ◀

By Theorem 10 and the above lemma, in order to prove Theorem 4 it suffices to show that zero automata recognise the characteristic transducers of the languages of the form Z_n as used in Theorem 10. By unraveling the definitions, we need to show the following lemma.

▶ **Lemma 12.** *For every* $n \in \mathbb{N}$ *there is a* zero *automaton recognising the set of trees*

$$t \otimes s \qquad \text{with } t \in \mathsf{trees}\{1, \ldots, n, \bot\},\ s \in \mathsf{trees}2$$

such that for every node x*, its label in* s *is* 1 *iff* Z_n *contains the subtree of* t *rooted in* x*.*

Proof. Let L be the language in the statement of the lemma. For a tree $t \otimes s$, define a $\bot$-factor to be a maximal factor contained in the domain of the tree that does not use label

$\perp$ in t. It is not difficult to see that $t \otimes s$ belongs to L if and only if: (a) every node with label $\perp$ in t has label 1 in s; and (b) every $\perp$-factor belongs to L. Condition (a) can be easily checked by a parity automaton, so thanks to the Intersection Lemma it suffices to produce a zero automaton which checks (b). By the Factorisation Lemma, it suffices to find a zero automaton which tests memberhip in L for individual $\perp$-factors.

Summing up, we can assume without loss of generality that t does not use label $\perp$ at all. Therefore, in the rest of the proof, we show a zero automaton which recognises the language L restricted to the case where $t \in \{1, \ldots, n\}$. For $i \in \{1, \ldots, n\}$, consider the function

$$f_i : \mathsf{trees}\{1, \ldots, n\} \to \mathsf{trees}\{1, 2, 3\} \qquad \text{label of } x \text{ in } f_i(t) = \begin{cases} 1 & \text{if label of } x \text{ in } t \text{ is } < i \\ 2 & \text{if label of } x \text{ in } t \text{ is } = i \\ 3 & \text{if label of } x \text{ in } t \text{ is } > i \end{cases}.$$

We will only use this function for even i. For $t \in \mathsf{trees}\{1, \ldots, n\}$, define nonzero$(t)$ to be the set of nodes in t whose subtree does not belong to Z_n. In terms of this definition, a tree $t \otimes s$ belongs to L if and only if nonzero(t) is exactly the nodes that have label 0 in s. Also, condition 4 from Definition 3 says that there is a zero automaton recognising the language

$$\mathsf{N} \stackrel{\text{def}}{=} \{t \in \mathsf{trees}\{1, 2, 3\} : \text{nonzero}(t) \text{ is all nodes of } t\}$$

▶ Claim 13. Let $t \in \mathsf{trees}\{1, \ldots, n\}$ and $s \in \mathsf{trees}2$ be trees with the same domain. Then $t \otimes s \in L$ if and only if one can find an ancestor closed set of nodes $\{X_i\}_i$, with i ranging over even numbers in $\{1, \ldots, n\}$, such that the following conditions hold:

1. a node has label 0 in s if and only if it belongs to some X_i;
2. for every even $i \in \{1, \ldots, n\}$, restricting $f_i(t)$ to the nodes from X_i yields a tree in N;
3. $\mathsf{zero} \ni \{\pi \in \mathsf{paths}\, t : \pi$ has even t-maxinf and sees 0 finitely often in $s\}$

Before proving the claim, let us observe how it implies the lemma. Since a zero automaton can nondeterministically guess the sets X_i, it suffices to show that there is a zero automaton which checks conditions 1, 2, 3 in the claim. By the Intersection Lemma, it suffices to check each condition individually. Condition 1 is definable in MSO. Condition 2, for any fixed i, follows from the assumption that N is recognised by a zero automaton and the Factorisation Lemma. For condition 3, it is straightforward to construct a zero automaton – it essentially copies the labels from t into its states, except that nodes with label 0 in s trigger a state which is maximal in the total order. It remains to prove the claim.

Proof. We begin with the following observation, which follows from the assumption that zero satisfies conditions 1-3 in Definition 3. For every $t \in \mathsf{trees}\{1, \ldots, n\}$, the set nonzero$(t)$ is closed under ancestors and a node x belongs to nonzero(t) if and only if

$$\mathsf{zero} \not\ni \{\pi \in \mathsf{paths}\, t : \begin{cases} \pi \text{ is contained in nonzero}(t), \text{ and} \\ \pi \text{ passes through } x, \text{ and} \\ \pi \text{ has even } t\text{-maxinf} \end{cases} \} \tag{1}$$

By definition of the tree transducers f_i, a path has even t-maxinf if and only if it has even $f_i(t)$-maxinf for some even i. Therefore, by closure of zero under countable – and therefore also finite – unions, we see that

$$\text{nonzero}(t) = \bigcup_i \text{nonzero}(f_i(t)), \tag{2}$$

where i ranges over even numbers in $\{1, \dots, n\}$.

Let us now prove the claim.

Let us begin with the bottom-up implication. From condition 2 it follows that every node in X_i belongs to nonzero(t). From condition 1 it follows that all nodes with label 0 are in nonzero(t). From condition 1, it follows that the set of nodes with label 0 in s is closed under ancestors. Therefore, condition 3 implies that for every node with label 1 in s is outside nonzero(t). Thus nonzero(t) is exactly the nodes which have label 0 in s, which means that $t \otimes s \in L$.

Consider the top-down implication. Our assumption is that nonzero(t) is exactly the nodes which have label 0 in s. Define X_i to be nonzero$(f_i(t))$. By (2), we see that condition 1 in the statement of the claim holds. From (1) applied to the trees $f_i(t)$, we get condition 2. To prove condition 3, by definition of nonzero(t) and prefix independence of zero, we know that every node $x \notin$ nonzero(t) satisfies

$$\mathsf{zero} \ni \{\pi \in \mathsf{paths}\, t : \pi \text{ passes through } x \text{ and has even } t\text{-maxinf}\}$$

Since nonzero(t) is ancestor closed, it follows that a path passes through some $x \notin$ nonzero(t) if and only if it sees 0 in s finitely often. Therefore, by closure of zero under countable unions, we get condition 3 in the statement of the claim. ◀

◀

8 Conclusion

We have proved that, under certain conditions on zero, every formula of the logic TMSO+zero is recognised by a zero automaton. Therefore, in order to decide satisfiability of TMSO+zero, it suffices to decide emptiness for zero automata. Unlike the logic, zero automata involve no nesting, which makes the emptiness check easier. A planned followup paper will show that emptiness is indeed decidable for zero automata, assuming that zero is the sets of probability zero.

Apart from the emptiness question for zero automata, the main open problem is decidability for the full logic MSO+zero, and not just the thin variant considered in this paper. It is not at all clear if zero automata are closed under complement, and therefore it is quite possible that zero automata are not the right model for MSO+zero. There is another logic, which sits between TMSO+zero and MSO+zero, and which might still admit a translation to zero automata. In this intermediate logic, the condition on sets $X \subseteq 2^*$ is relaxed: instead of thin sets, we consider sets which satisfy $\bar{X} \in$ zero. We leave open the question whether this intermediate logic admits a translation to zero automata.

References

1 Christel Baier, Marcus Größer, and Nathalie Bertrand. Probabilistic ω-automata. *J. ACM*, 59(1):1, 2012.

2 Vince Bárány, Łukasz Kaiser, and Alexander Rabinovich. Cardinality quantifiers in MLO over trees. In *Proc. of CSL*, 2009.

3 Mikolaj Bojanczyk. Weak MSO with the unbounding quantifier. *Theory Comput. Syst.*, 48(3):554–576, 2011.

4 Mikolaj Bojanczyk. U. *ACM SIGLOG News*, 2(4):2–15, 2015.

5 Tomás; Brázdil, Vojtech Forejt, Jan Kretínský, and Antonín Kucera. The satisfiability problem for probabilistic CTL. In *Proc. of LICS*, pages 391–402, 2008.

6 Julius R. Büchi. On a decision method in restricted second-order arithmetic. In *Proc. 1960 Int. Congr. for Logic, Methodology and Philosophy of Science*, pages 1–11, 1962.

7 Arnaud Carayol, Axel Haddad, and Olivier Serre. Randomization in automata on infinite trees. *ACM Trans. Comput. Log.*, 15(3):24:1–24:33, 2014.

8 Thomas Colcombet. Regular cost functions, part I: logic and algebra over words. *Logical Methods in Computer Science*, 9(3), 2013.

9 Daniel Lehmann and Saharon Shelah. Reasoning with time and chance. *Information and Control*, 53(3):165–1983, 1982.

10 Henryk Michalewski and Matteo Mio. Baire category quantifier in monadic second order logic. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part II*, pages 362–374, 2015.

11 Henryk Michalewski and Matteo Mio. Measure quantifier in monadic second order logic. In *Logical Foundations of Computer Science - International Symposium, LFCS 2016, Deerfield Beach, FL, USA, January 4-7, 2016. Proceedings*, pages 267–282, 2016.

12 Michael O. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of American Mathematical Society*, 141:1–35, 1969.

13 Alexander Rabinovich. On decidability of monadic logic of order over the naturals extended by monadic predicates. *Inf. Comput.*, 205(6):870–889, 2007.

14 Sergiu Hart Micha Sharir. Probabilistic propositional temporal logics. *Information and Control*, 70(2–3):97–155, 1986.

A Probability zero satisfies the assumptions of Theorem 4

In this part of the appendix, we prove Corollary 5. Fix zero to be the subsets of 2^ω that have zero probability in the sense of Definition 1. To prove Corollary 5, it suffices to show that zero satisfies the assumptions of Theorem 4, namely conditions 1-4 in Definition 3. Conditions 1-3 in Definition 3 are easy to check, so we focus on condition 4. We need to show that that there is a zero automaton which recognises the language

$$\mathsf{N} \stackrel{\text{def}}{=} \{t \in \mathsf{trees}\{1,2,3\} : \text{for every subtree, the set of paths with maxinf 2 is} \notin \mathsf{zero}\}.$$

We will use the following characterisation of the language N, which can be captured by a zero automaton.

▶ **Lemma 14.** *A tree t belongs to N if and only if there exists a set of nodes S in t such that*

1. *every node in t has a descendant in S; and*
2. *for every node $x \in S$,*

$$\mathsf{zero} \not\ni \{\pi \in \mathsf{paths}\, t : \begin{cases} \pi \text{ passes through } x\text{; and} \\ \pi \text{ does not pass through } S \text{ after } x\text{; and} \\ \pi \text{ has } t\text{-maxinf } 2 \end{cases} \}$$

Before proving the Lemma 14, we observe that the characterisation of N in it can be recognised by a zero automaton which inputs $t \otimes S$, and therefore N is recognised by a zero automaton by guessing S nondeterministically. Condition 1 in Lemma 14 is recognised by a parity automaton, and therefore thanks to the Intersection Lemma from Section 5, it suffices to show that Condition 2 is recognised by a zero automaton. For condition 2, the automaton has four states:

$$1 < 2 < 3 < s.$$

When the automaton sees a node from S, it uses state s. Otherwise, it copies the input letter to its state. The set Q_{all} is all states and the set Q_{zero} is empty, which means that the all paths acceptance condition and the zero acceptance condition are vacuously true. For the nonzero acceptance condition, the unique seed state is s, and the set Q_{nonzero} is $\{2\}$. For this automaton, the nonzero acceptance condition is the same thing as item 2 in Lemma 14. Therefore, Lemma 14 is sufficient to prove Corollary 5.

Proof of Lemma 14. The bottom-up implication is straightforward, and therefore we concentrate on the top-down implication. The key is the following observation, which says that one can find dense sets of nodes with arbitrarily small probability.

▶ Claim 15. Let $X \subseteq 2^*$ be a factor such that every node in X has at least one child in X. For every $\epsilon > 0$, there exists a set $Y \subseteq X$ such that every node from X has a descendant in Y, and the following set has probability $< \epsilon$:

$$\{\pi \in \bar{X} : \pi \text{ passes through } Y \text{ at least once}\}$$

Proof. By assumption on X, every node from X has a descendant from X at arbitrarily large depth. Take some enumeration of the nodes in X, and for the i-th node choose some descendant in X that is sufficiently deep down in the tree so that reaching the descendant has probability at most $\epsilon/2^{i+1}$. Put all of these descendants into Y; their combined probability is at most ϵ. ◀

To prove the top-down implication, let $t \in \mathsf{N}$. Choose some enumeration $x_1, x_2, \ldots$ of the nodes in t. We define families of disjoint factors

$$\emptyset = \mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \mathcal{F}_2 \subseteq \cdots$$

in the tree t, each one containing finitely many factors, such that every x_i in the enumeration has a descendant that is a root of some factor in $\mathcal{F}_i$ and every factor $F \in \mathcal{F}_i$ satisfies:

(a) every node in F has a descendant that is in t but not in F;
(b) the set of paths $\{\pi \in \bar{F} : \pi \text{ has maxinf } 2\}$ is not in zero.

If we manage to define such families, then the claim will be proved, by taking S to be the roots of the factors that appear in some set $\mathcal{F}_i$. The construction of $\mathcal{F}_i$ is by induction on i. Suppose we have already defined $\mathcal{F}_{i-1}$ and consider the node x_i. By condition (a) and finiteness of $\mathcal{F}_{i-1}$, there must be a descendant of x_i which is in none of the factors from $\mathcal{F}_{i-1}$. Call this descendant y, and let X be the descendants of y that are in t. By the assumption that $t \in \mathsf{N}$, the set

$$\{\pi \in \bar{X} : \pi \text{ has maxinf } 2\}$$

is not in zero. Since the above set is Borel, it has some defined probability $\epsilon > 0$. Apply Claim 15 to X and ϵ, yielding some $Y \subseteq X$. Define F to be the factor obtained from X by removing all nodes which have an ancestor in Y. By Claim 15, conditions (a) and (b) hold for F, and therefore we can add it to $\mathcal{F}_{i-1}$ thus creating $\mathcal{F}_i$. ◀

B Closure of zero automata under intersection

In this part of the appendix, we prove the Factorisation and Intersection Lemmas, which establish closure properties of languages recognised by zero automata. Here is the plan. In Section B.1, we show that languages recognised by zero automata are closed under intersection with languages recognised by zero automata that do not use the nonzero acceptance condition at all. In Section B.2, we prve the Factorisation Lemma. In Section B.3, we show that languages recognised by zero automata are closed under intersection with the most basic languages that refer to nonzero, i.e. with languages of the form

$$\{t \in \mathsf{trees}\{1, \ldots, n\} : \mathsf{zero} \text{ does not contain the set of paths with even maxinf}\}.$$

Finally, in Section B.4 we put all the pieces together and prove closure under intersection for general zero automata.

In this part of the appendix, we use the following notation for intefvals of nodes in a tree. If x, y are nodes such that x is an ancestor of y, we define

$$(x..y) \stackrel{\text{def}}{=} \{z : x \text{ is a proper ancestor of } z \text{ and } z \text{ is a proper ancestor of } y\}.$$

Likewise we define $[x..y]$, $[x..y)$ and $(x..y]$, with round brackets meaning "proper ancestor" and square brackets meaning "not necessarily proper ancestor".

B.1 Intersection with seedless automata

The main difficulty in the Intersection Lemma is going to be intersecting two instances of the nonzero acceptance condition. Let us therefore begin by proving a special case where this difficulty is avoided, i.e. where one of the automata is *seedless*, i.e. it does not have any seed states. In a seedless automaton, the nonzero acceptance condition is vacuously true. In this section, we prove that languages recognised by zero automata are closed under intersection with languages recognised by seedless zero automata.

Let us first explain what is gained in the Intersection Lemma by assuming that one of the automata is seedless. When doing intersection, the all paths acceptance condition and the zero acceptance condition are easier to deal with, because we can use the following reductions, for sets of paths $\Pi_0, \Pi_1 \subseteq 2^\omega$:

$$\Pi_0 = 2^\omega \wedge \Pi_1 = 2^\omega \quad \text{iff} \quad \Pi_0 \cap \Pi_1 = 2^\omega \tag{3}$$

$$\Pi_0 \in \mathsf{zero} \wedge \Pi_1 \in \mathsf{zero} \quad \text{iff} \quad \Pi_0 \cup \Pi_1 \in \mathsf{zero} \tag{4}$$

For the nonzero acceptance condition, we do not see such a reduction, and this is why it is easier to assume that one of the automata is seedless.

Intersection with safety automata

As a warmup, let us consider an even more special case of zero automata, where the entire acceptance condition is not used at all. Define a *safety automaton* to be the very special case of zero automata, where the acceptance condition is trivial, i.e. every run with the initial state in the root is accepting. On the syntactic level this corresponds to Q_{all} being all states, Q_{zero} being empty, and having no seed states.

▶ **Lemma 16.** *Languages recognised by* zero *automata are closed under intersection with safety automata.*

Proof sketch. Using a product construction, i.e. the states are pairs of states. The only challenging question is the order: we use the lexicographic ordering, with the less important coordinate being the safety automaton. ◀

Define a *nondeterministic transducer* to be a partial function

$$f : \mathsf{trees}\Sigma \to \mathsf{trees}\Gamma$$

whose graph is equal to $\{(s, t) : s \otimes t \in L\}$ for some L recognised by a safety automaton. The name *nondeterministic* is chosen because the safety automaton can use nondeterminism, however f itself must be a partial function, i.e. every input produces at most one output. A corollary of Lemma 16 is that languages recognised by zero automata are closed under inverse images of nondeterministic transducers.

McNaughton's Latest Appearance Record

Having proved intersection with safety automata, let us move toward intersection with seedless automata. In the proof, we use a product construction and the reductions from (3) and (4). There is one difficulty to overcome: how to define the total order on the states in a product automaton? For this, use McNaughton's Latest Appearance Record (LAR).

If t is a tree and π is a path contained in its domain, we use the name t-inf of π for the set of labels seen infinitely often by π in the tree. If the alphabet is equipped with a total order, we write t-maxinf of π for the maximal element of its t-inf. The following lemma is proved using McNaugton's LAR construction.

▶ **Lemma 17** (LAR lemma)**.** *Let Σ be a finite set. There exist a tree transducer*

$$f : \mathsf{trees}\Sigma \to \mathsf{trees}\{1, \ldots, n\}$$

recognised by a Mealy machine such that for every tree $t \in \mathsf{trees}\Sigma$ and every path π, the t-inf of π is uniquely determined by the $f(t)$-maxinf of π.

We will use the following corollary of the LAR Lemma.

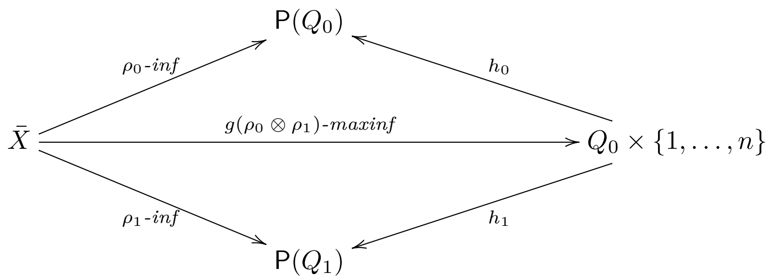
▶ **Corollary 18.** *Let Q_0, Q_1 be ordered sets. There exists some n and functions*

$$g : \mathsf{trees}(Q_0 \times Q_1) \to \mathsf{trees}(Q_1 \times \{1, \ldots, n\}) \qquad h_i : Q_0 \times \{1, \ldots, n\} \to Q_i \quad \textit{for } i \in 2$$

such that g is recognised by a Mealy machine and, assuming that $Q_0 \times \{1, \ldots, n\}$ is ordered lexicographically with Q_0 being more important, the following property holds. Let

$$\rho_0 \in \mathsf{trees}Q_0, \rho_1 \in \mathsf{trees}Q_1$$

be trees with the same domain $X \subseteq 2^*$. Then the following diagram commutes



Proof. Apply the LAR lemma to $Q_0 \times Q_1$, yielding some

$$f : \mathsf{trees}(Q_0 \times Q_1) \to \mathsf{trees}\{1, \ldots, n\}$$

recognised by a Mealy machine. Define g as follows. Suppose that the input is $\rho_0 \otimes \rho_1$. The label of a node x is the pair (q, i) where q is the label of x in ρ_0 and i is the biggest number that appears in $f(\rho_0 \otimes \rho_1)$ on nodes from the following set:

$$\{y : y \leq x \text{ and all nodes in } [y..x) \text{ have label} < q \text{ in } \rho_1\}.$$

For every path π in $\rho_0 \otimes \rho_1$, the $g(\rho_0 \otimes \rho_1)$-maxinf of π is the pair (q, i) such that q is the ρ_0-maxinf of π and i is the $f(\rho_0 \otimes \rho_1)$-maxinf of π. In particular, thanks to the LAR lemma, i can be used to recover the labels that appear infinitely often in ρ_0 and ρ_1. ◀

Intersection with seedless automata

We are now ready to show how intersection with seedless automata.

▶ **Lemma 19** (Seedless Intersection Lemma)**.** *Languages recognised by* zero *automata are closed under intersection with languages recognised by seedless* zero *automata.*

Proof. Consider two zero automata $\mathcal{A}_0, \mathcal{A}_1$ over a common input alphabet Σ such that $\mathcal{A}_1$ is seedless. Let Q_0, Q_1 be their states spaces. By using nondeterminism of zero automata, it suffices to give a zero automaton which recognises the language

$$\{\rho_0 \otimes \rho_1 \in \mathsf{trees}(Q_0 \times Q_1) : \rho_i \text{ is an accepting run of } \mathcal{A}_i \text{ for every } i \in 2\}$$

Consistency with the transitions is recognised by a safety automaton, and thus by Lemma 16 it suffices to check the acceptance conditions.

Apply Corollary 18 to Q_0, Q_1, yielding some n, g and h_0, h_1. By again applying Lemma 16, it suffices to show that there is zero-automaton which recognises the set

$$\{g(\rho_0 \otimes \rho_1) : \rho_i \text{ satisfies the acceptance condition for every } i \in 2\}$$

The automaton which recognises the above language has states $Q_0 \times \{1, \ldots, n\}$, which copy the appropriate coordinates of the input tree. The state space is ordered in the same way as the input alphabet, i.e. lexicographically with Q_0 being more important. The seed states and Q_{nonzero} are inherited from Q_1. For Q_{all} and Q_{zero} we refer to the functions h_0, h_1 and use the reductions from (3) and (4). ◀

Since seedless zero automata generalise parity automata, the Seedless Intersection Lemma implies that languages recognised by zero automata are closed under intersection with MSO definable languages.

B.2 Closure under factorisations

We now prove the Factorisation Lemma. Recall that this lemma says that if $L \subseteq \mathsf{trees}\Sigma$ is recognised by a zero automaton, then so is

$$\{t \otimes X : t \in \mathsf{trees}\Sigma \text{ and } X \text{ is a set of nodes in } t \text{ such that } L \text{ contains every } X\text{-factor of } t\}$$

Proof of the Factorisation Lemma. Let Q be the states of the automaton recognising L. By using nondeterminism of zero automata, it suffices to find a zero automaton which recognises the set of trees

$$\rho \otimes X \in \mathsf{trees}(Q \times 2)$$

such that every X-factor of ρ is an accepting run. The property "every X-factor satisfies the all paths acceptance condition" can be formalised in MSO, and is therefore recognised by a seedless zero automaton. The property "every X-factor satisfies the all nonzero acceptance condition" can be recognised by a zero automaton, which is simply copies the states from the input trees, and uses a special state that is maximal in the order whenever it sees a node from X. This maximal state ensures that the nonzero sets of paths are measured inside X-factors.

Therefore, by closure under intersection with seedless zero automata, it suffices to find a seedless zero automaton, which recognises the property "every X-factor satisfies the zero acceptance condition". In other words, we want a seedless zero automaton which checks that every X-factor Y satisfies:

$$\mathsf{zero} \quad \ni \quad \{\pi \in \bar{Y} : \text{the } \rho\text{-maxinf of } \pi \text{ belongs to } Q_{\text{zero}}\}$$

Because there are countably many X-factors, and zero is closed under countable union and subsets, the above is equivalent to saying the the following set of paths is in zero:

$$\mathsf{zero} \quad \ni \quad \bigcup_Y \{\pi \in \bar{Y} : \text{the } \rho\text{-maxinf of } \pi \text{ belongs to } Q_{\text{zero}}\}$$

where Y ranges over X-factors. By definition of X-factors, a path is eventually contained in some X-factor if and only if it sees nodes from X finitely often. Therefore, the above condition is the same as:

$$\mathsf{zero} \quad \ni \quad \{\pi \in 2^\omega : \pi \text{ sees } X \text{ finitely often and the } \rho\text{-maxinf of } \pi \text{ belongs to } Q_{\text{zero}}\}$$

Using the LAR lemma, we can find a transducer

$$f : \mathsf{trees}(Q \times 2) \to \mathsf{trees}\{1, \dots, n\}$$

such that a path π sees X infinitely often and has ρ-maxinf in Q_{zero} if and only if it has even $f(t \otimes X)$-maxinf. This condition can be expressed by a seedless zero automaton. ◀

B.3 Products with basic nonzero languages

We say $t \in \mathsf{trees}\{1, \dots, n\}$ satisfies the *basic nonzero condition* if

$$\mathsf{zero} \not\ni \{\pi : \pi \text{ is contained in } t \text{ and has even } t\text{-maxinf}\}.$$

This the same as the language Z_n from Theorem 10, except that label $\bot$ is not used. The basic nonzero condition can be seen as the special case of the nonzero automaton where a seed state is used only once and in the root. The goal of this section is to show Lemma 20, which says that languages recognised by zero automata are closed under products with basic nonzero conditions. The *product* of two languages L_0, L_1 is defined as:

$$\{t_0 \otimes t_1 : t_0 \in L_0 \text{ and } t_1 \in L_1\}.$$

Recall that in order for $t_0 \otimes t_1$ to be defined, both trees need to have the same domain.

▶ **Lemma 20.** *Languages recognised by* zero *automata are closed under products with languages of the form*

$$\{t \in \mathsf{trees}\{1, \dots, n\} : t \textit{ satisfies the basic nonzero condition}\}.$$

The rest of Section B.3 is devoted to proving the above lemma. Let then $\mathcal{A}$ be a zero automaton with states Q and let $n \in \mathbb{N}$. Let us begin with the following simple observation, which shows that the nonzero acceptance condition works well with factorisations.

▶ **Lemma 21.** *A tree $\rho \in \mathsf{trees}Q$ satisfies the nonzero acceptance condition if and only if there exists a subset X of its domain which contains the root and such that every X-factor of ρ satisfies the nonzero acceptance condition.*

Proof. For the left-to-right implication, we take to contain only the root, and therefore there is only one X-factor, namely the entire tree itself. Consider the converse right-to-left implication. By definition, the nonzero acceptance condition for ρ says that for every node x in ρ with state $q \in Q_{\text{seed}}$:

$$\mathsf{zero} \not\ni \{\pi \in \mathsf{paths}\,\rho : \begin{cases} \pi \text{ passes through } x\text{, and} \\ \pi \text{ sees only states } < q \text{ after } x\text{, and} \\ \pi \text{ has maxinf state in } Q_{\text{nonzero}} \end{cases} \}$$

By prefix independence of zero and the assumption that all X-factors of ρ satisfy the nonzero acceptance condition, we know that

$$\mathsf{zero} \not\ni \{\pi \in \mathsf{paths}\,\rho : \begin{cases} \pi \text{ passes through } x\text{, and} \\ \pi \text{ sees only states } < q \text{ after } x\text{, and} \\ \pi \text{ does not pass through } X \text{ after } x\text{, and} \\ \pi \text{ has maxinf state in } Q_{\text{nonzero}} \end{cases} \}$$

Therefore, we can conclude by closure of zero under taking subsets. ◀

The key to the proof of Lemma 20 is the following characterisation.

▶ **Lemma 22.** *The following conditions are equivalent for every*

$\rho \otimes t \in \mathsf{trees}(Q \times \{1, \dots, n\})$

1. *ρ satisfies the nonzero acceptance condition and $t \in \mathsf{Z}_n$;*
2. *there exists a set X of nodes in the domain of $\rho \otimes t$, which contains the root, such that:*
   a. *every X-factor of ρ satisfies the nonzero acceptance condition; and*
   b. *there is some X-factor Y such that*

$$\{\pi \in \bar{Y} : \begin{cases} \pi \text{ has } \rho\text{-maxinf } q, \text{ and} \\ \pi \text{ has even } t\text{-maxinf} \end{cases} \} \tag{5}$$

holds for some q such that every seed state appearing in Y is $\leq q$.

Proof. Let us begin with the simpler bottom-up implication. By Lemma 21, Condition 2a implies that ρ satisfies the nonzero acceptance condition. Condition 2b implies that $t \in \mathsf{Z}_n$ because zero is prefix independent and closed under subsets.

The top-down implication remains. Let $\rho \otimes t$ satisfy condition 1. Choose some state q such that

$$\mathsf{zero} \not\ni \{\pi \in \mathsf{paths}\, \rho : \begin{cases} \pi \text{ has } \rho\text{-maxinf } q, \text{ and} \\ \pi \text{ has even } t\text{-maxinf} \end{cases} \}.$$

Such a state must exist, by closure of zero under countable – and therefore also finite – unions. Again by closure of zero under countable unions, there must be some z_0 such that

$$\mathsf{zero} \not\ni \{\pi \in \mathsf{paths}\, \rho : \begin{cases} \pi \text{ has } \rho\text{-maxinf } q, \text{ and} \\ \pi \text{ passes through } z_0, \text{ and} \\ \pi \text{ sees only states } \leq q \text{ after } z_0, \text{ and} \\ \pi \text{ has even } t\text{-maxinf} \end{cases} \}.$$

Define Z to be those descendants z of z_0 such that $[z..z_0]$ has only labels $\leq q$ in ρ. The above condition then becomes

$$\mathsf{zero} \not\ni \{\pi \in \bar{Z} : \begin{cases} \pi \text{ has } \rho\text{-maxinf } q, \text{ and} \\ \pi \text{ has even } t\text{-maxinf} \end{cases} \}.$$

Define $A \subseteq Z$ to be those nodes $z \in Z$ such that

$$\mathsf{zero} \not\ni \{\pi \in \bar{Z} : \begin{cases} \pi \text{ passes through } z, \text{ and} \\ \pi \text{ has } \rho\text{-maxinf } q, \text{ and} \\ \pi \text{ has even } t\text{-maxinf} \end{cases} \}.$$

By definition $z_0 \in A$.

▶ Claim 23. There is some $x \in Z \cap A$ such that if we define

$X = \{root, x\} \cup \{z \in \partial Z : z \text{ is a descendant of } x\}$

then for every $z \in X$, the nonzero acceptance condition is satisfied by the tree ρ_z obtained from ρ by removing z and its subtree.

Proof. Because zero is prefix independent and closed under countable unions, every element of A has some descendant in A which has label q. For a state p, define A_p to be the set of descendants z of z_0, not necessarily belonging to Z, such that

$$\mathsf{zero} \not\ni \{\pi \in \mathsf{paths}\,\rho : \begin{cases} \pi \text{ passes through } z, \text{ and} \\ \pi \text{ sees only states} \leq p \text{ in } \rho \text{ after } z, \text{ and} \\ \pi \text{ has } \rho\text{-maxinf in the set } Q_{\text{nonzero}} \text{ for } \mathcal{A} \end{cases} \}.$$

Because zero contains all singletons, if A_p is nonempty then it is not contained in a single path. Note that the mapping $p \mapsto A_p$ is monotone with respect to inclusion on the right. In particular, there are two distinct paths that are contained in all of the sets A_p which are nonempty. Therefore, or there is some $x \in A$ which has label q and such that there is some node which is in all nonempty sets A_p and is not an ancestor of x.

Choose this x.

Let z be as in the statement of the claim. Let us check the nonzero acceptance condition for ρ_z. Let then y be a node in ρ_z that is labelled by a seed state, say p. If y is not an ancestor of z then the subtree of y is the same in ρ and ρ_z, and therefore we are done. Suppose that y is an ancestor of z. If $p \leq q$ then, because z has label $\geq q$, it follows that the nonzero acceptance condition is not affected in in y by removing z. The last case is when $p > q$, which means necessarily that y is a proper ancestor of z_0, since all nodes in Z have label $\leq q$. By choice of x, if there is some descendant of y in some A_p, then there is also a descendant that is not in the removed subtree of z. Therefore removing the subtree of z does not affect the nonzero acceptance condition in y. ◀

Apply the above claim, yielding some x and X. We show below that conditions 2a and 2b from the statement of the lemma are satisfied for this choice of X.

(2a) Take an X-factor of ρ, call it σ. By definition of X, the root of this X-factor is either the root of the entire tree, or it is x, or it is a descendant of x in ∂Z. If the root of the X-factor is the root of the entire tree, then σ is equal to ρ_x, according to the notation from the above claim, and therefore the nonzero acceptance condition is satisfied. If the root is ∂Z, then σ is a subtree of ρ, and the nonzero acceptance condition is easily seen to be preserved under taking subtrees. The last case is when σ is obtained from ρ by moving the root ot x, and removing all subtrees in ∂Z. Since nodes in ∂Z have state q, it follows that the nonzero acceptance condition is satisfied for all seed states $\leq q$. However, by definition of Z, σ has only states $\leq q$, and thus the nonzero acceptance condition holds.

(2b) Let Y be the X-factor which has root x. In other words, Y is the intersection of Z with the descendants of x. Because Y is contained in Z, it has only states $\leq q$. By the assumption that $x \in Z \cap A$, we get (5).

◀

Before proving Lemma 20, we present one final closure property of zero automata.

▶ **Lemma 24.** *Let $q \in Q$. There is a* zero *automaton which accepts*

$$\rho \otimes t \in \mathsf{trees}(Q \times \{1, \ldots, n\})$$

if and only if ρ satisfies the nonzero acceptance condition in $\mathcal{A}$, only states $\leq q$ appear in ρ, and

$$\mathsf{zero} \not\ni \{\pi \in \mathsf{paths}\,\rho : \begin{cases} \pi \textit{ has } \rho\textit{-maxinf } q, \textit{ and} \\ \pi \textit{ has even } t\textit{-maxinf} \end{cases} \}$$

Proof sketch. Let us consider the special case of the lemma where the language is restricted to those trees where the root of ρ not to have a seed state. Here we use Corollary 18 in the same way as in Lemma 19, with a the root being a special seed state of maximal rank, which tests the condition on t-maxinf. For the general case, we observe that if the root is seed state in ρ, then this seed can be delayed to one of the children in the root. ◀

Proof of Lemma 20. It suffices to show that a zero automaton can recognise the property in item 2 of Lemma 22. The set X and the factor Y can be guessed nondeterministically. By the Factorisation Lemma, it suffices to show that there is zero-automaton which checks the property in item 2 for each X-factor individually. Here we use Lemma 24. ◀

B.4 The general case of intersection

In this section we complete the proof of the Intersection Lemma. The key lemma is that languages recognised by zero automata are closed under intersection with languages recognised by zero automata that have only one seed state.

▶ **Lemma 25.** *Let $\mathcal{A}_0, \mathcal{A}_1$ be* zero *automata such that $\mathcal{A}_0$ has only one seed state. Then the intersection of the languages recognised by these automata is recognised by a* zero *automaton.*

Before proving the above lemma, we show how it completes the proof of the Intersection Lemma.

Proof of the Intersection Lemma. Consider zero automata $\mathcal{A}_0, \mathcal{A}_1$ with states Q_0, Q_1. By induction on the number of seed states in $\mathcal{A}_1$, we prove that there is a zero automaton recognising the intersection of the two languages. The induction base is when $\mathcal{A}_0$ has no seed states, which is Lemma 19. Let us do the induction step.

Because languages recognised by zero automata are closed under projections, it suffices to show that there is a zero automaton which recognises the set of trees

$$\rho_0 \otimes \rho_1 \in \mathsf{trees}(Q_0 \times Q_1)$$

such that for every $i \in 2$, the run ρ_i is an accepting run of $\mathcal{A}_i$ over t. Let $\mathcal{A}_1'$ be the automaton obtained from $\mathcal{A}_1$ be taking some seed state q and removing it from the set of seed states. By induction assumption, there is a zero automaton which checks if $\rho_0 \otimes \rho_1$ is a combination of runs for $\mathcal{A}_0$ and $\mathcal{A}_1'$. Adding the state q to the seed states corresponds to intersecting with an automaton that has only one seed state, which is where we use Lemma 25. ◀

Using the same proof as for Lemma 24, one shows the following lemma.

▶ **Lemma 26.** *Let $\mathcal{A}$ be a* zero *automaton with states Q. Let $q \in Q$ be such that the maximal seed state in $\mathcal{A}$ is $\leq q$. For every $n \in \mathbb{N}$ there is a* zero *automaton which recognises the set of trees*

$$\rho \otimes t \in \mathsf{trees}(Q \times \{1, \ldots, n\})$$

such that ρ satisfies the nonzero acceptance condition in $\mathcal{A}$ and for every node x with label q in ρ, if i is the label of x in t then

$$\mathsf{zero} \not\ni \{\pi \in \mathsf{paths}\,\rho : \begin{cases} \pi \textit{ passes through } x, \textit{ and} \\ \pi \textit{ sees } q \textit{ infinitely often, and} \\ \pi \textit{ sees only labels} < i \textit{ in } t \textit{ after } x, \textit{ and} \\ \pi \textit{ has even } t\textit{-maxinf} \end{cases} \}$$

Proof of Lemma 25. Fix zero automata $\mathcal{A}_0, \mathcal{A}_1$ such that $\mathcal{A}_0$ has only one seed state. Let Q_0, Q_1 be their state spaces. By nondeterminism of zero automata and closure under intersection with seedless zero automata, it suffices to prove that there is a zero automaton recognising the language:

$$\{\rho_0 \otimes \rho_1 \in \mathsf{trees}(Q_0 \times Q_1) : \rho_i \text{ satisfies the nonzero acceptance condition for } i \in 2\}.$$

The key is the following characterisation of the above language.

▶ Claim 27. Let ρ_0, ρ_1 be runs of the automata $\mathcal{A}_0, \mathcal{A}_1$ with the same domain, and let X be the set of nodes where the unique seed state of $\mathcal{A}_0$ appears in ρ_0. Then ρ_0, ρ_1 satisfy the nonzero acceptance conditions in their respective automata if and only if there exists families

$$\{\mathrm{Int}_q\}_{q \in Q_1} \qquad \{\mathrm{Ext}_q\}_{q \in Q_1} \tag{6}$$

of sets of nodes in the domain such that

1. if y is a node whose state q in ρ_1 is a seed state, then there is a descendant x of y such that all nodes in $(y..x)$ have label $< q$ and $x \in \mathrm{Int}_p \cup \mathrm{Ext}_p$ for some $p \leq q$.
2. if Y is an X-factor then both conditions below are satisfied:
   a.
   $$\mathsf{zero} \not\ni \{\pi \in \bar{Y} : \text{the } \rho_0\text{-maxinf of } \pi \text{ belongs to } Q_{\text{nonzero}} \text{ from } \mathcal{A}_0\}$$
   b. for every $q \in Q_1$ and $x \in Y \cap \mathrm{Int}_q$,
   $$\mathsf{zero} \not\ni \{\pi \in \bar{Y} : \begin{cases} \pi \text{ passes through } x \text{, and} \\ \pi \text{ sees only states } < q \text{ after } x \text{, and} \\ \text{the } \rho_1\text{-maxinf of } \pi \text{ belongs to } Q_{\text{nonzero}} \text{ from } \mathcal{A}_1. \end{cases}\}$$
3. for every $x \in X \cap \mathrm{Ext}_q$,
   $$\mathsf{zero} \not\ni \{\pi \in \mathsf{paths}\, \rho_0 : \begin{cases} \pi \text{ passes through } x \text{, and} \\ \pi \text{ sees only states } < q \text{ after } x \text{ in } \rho_1 \text{, and} \\ \pi \text{ passes through } X \text{ infinitely often, and} \\ \text{the } \rho_1\text{-maxinf of } \pi \text{ belongs to } Q_{\text{nonzero}} \text{ from } \mathcal{A}_1. \end{cases}\}$$

Proof. We begin with the simpler bottom-up implication. Suppose that ρ_0, ρ_1, X and families as in (6) are such that the above conditions 1, 2, 3 are satisfied. For the automaton $\mathcal{A}_0$, the nonzero condition is exactly the same as saying that condition 2a holds for every X-factor. Consider the automaton $\mathcal{A}_1$. Let y be a node in ρ_1 which has a seed state q. Choose $x \in \mathrm{Int}_p \cup \mathrm{Ext}_p$ as in item 1. Using condition 3 or 2b, depending on whether Int_p or Ext_p contains x, we show that

$$\mathsf{zero} \not\ni \{\pi \in \mathsf{paths}\, \rho_0 : \begin{cases} \pi \text{ passes through } x \text{, and} \\ \pi \text{ sees only states } < p \text{ after } x \text{ in } \rho_1 \text{, and} \\ \text{the } \rho_1\text{-maxinf of } \pi \text{ belongs to } Q_{\text{nonzero}} \text{ from } \mathcal{A}_1. \end{cases}\}.$$

By prefix independence of zero, and the assumption that only states $< q$ appear in $(y..x]$, we see that the nonzero condition is satisfied in y.

We are left with the top-down implication. Let ρ_0, ρ_1 and X be as in the assumptions of the claim. Assume that ρ_0, ρ_1 satisfy the nonzero condition in their respective automata. Condition 2a is satisfied by the assumption that ρ_0 satisfies the nonzero condition. Define the sets from (6) to be the maximal ones which make conditions 2b and 3 hold. It remains to check that condition 1 holds. Suppose then that y is a node which is labelled by a seed sate q in ρ_1. By the assumption that ρ_1 satisfies the nonzero condition, we know that

$$\Pi \stackrel{\text{def}}{=} \{\pi \in \mathsf{paths}\,\rho_0 : \begin{cases} \pi \text{ passes through } y, \text{ and} \\ \pi \text{ sees only states } < q \text{ after } x \text{ in } \rho_1, \text{ and} \\ \pi \text{ has } \rho_1\text{-maxinf state in } Q_{\text{nonzero}} \text{ for } \mathcal{A}_1 \end{cases} \}$$

is not in zero. Let x be a node of ρ_0, or equivalently of ρ_1. Define

$$\Pi_x^{int} \stackrel{\text{def}}{=} \{\pi \in \Pi : \ \pi \text{ passes through } x \text{ and does not visit } X \text{ after } x\}$$
$$\Pi_x^{ext} \stackrel{\text{def}}{=} \{\pi \in \Pi : \ \pi \text{ passes through } x \text{ and visits } X \text{ infinitely often}\}$$

It is easy to see that the above sets cover all of Π. By closure of zero under countable unions, one of the above sets must be outside zero. Condition 1 in the statement of the claim then follows from the following straightforward observations:

- if Π_x^{int} or Π_x^{ext} is nonempty, then $(y..x]$ contains only labels $< q$ in ρ_1;
- if Π_x^{int} is nonempty then $x \in \mathrm{Int}_p$ for some $p \le q$;
- if Π_x^{ext} is nonempty then $x \in \mathrm{Ext}_p$ for some $p \le q$.

◀

Consider sets as in (6). Note that if a node belongs to Int_p, then we can safely add it to Int_q for all $p \le q$ without affecting condition 2b. Therefore, by Claim 27, to prove the lemma it suffices to show that there is a zero automaton which accepts a tree

$$\rho_0 \otimes \rho_1 \otimes int \otimes ext \in \mathsf{trees}(Q_0 \times Q_1 \times (Q_0 \cup \{\bot\}) \times (Q_0 \cup \{\bot\})) \tag{7}$$

if and only if the three conditions from the claim are satisfied assuming that

$$\mathrm{Int}_p \;=\; \{x : int(x) \text{ is not } \bot \text{ and } int(x) \le p\}$$
$$\mathrm{Ext}_p \;=\; \{x : int(x) \text{ is not } \bot \text{ and } ext(x) \le p\}.$$

Condition 1 is definable in MSO, and therefore it suffices to find a zero automaton for conditions 2 and 3. Condition 2a, as a property of the factor in Y, is an instance of Z_n, up to relabeling the alphabet. Condition 2b, as a property of the factor in Y, is recognised by a zero automaton on partial trees. Therefore, by Lemmas 20 and the Factorisation Lemma, Condition 2 is recognised by a zero automaton. From the construction in Lemma 7, we can assume that the automaton recognising Condition 2 uses its maximal state in nodes from X. Therefore, to add condition 3, we use Lemma 26. ◀

C From logic to transducers

In this part of the appendix, we prove Theorem 10. Define $\mathcal{T}$ to be the class of transducers in the statement of the theorem. The theorem says that a language is definable in TMSO+zero if and only if its characteristic transducer belongs to $\mathcal{T}$.

Let us begin with the simpler transducer-to-logic implication. We say that a transducer $f : \mathsf{trees}\Sigma \to \mathsf{trees}\Gamma$ is definable in TMSO+zero if for every $a \in \Gamma$ there is a formula $\varphi_a(x)$ of TMSO+zero with a single free node variable, such that for every $t \in \mathsf{trees}\Sigma$, the formula $\varphi_a(x)$ selects those nodes in the domain of t which have label a in the output $f(t)$.

▶ **Lemma 28.** *Every transducer from $\mathcal{T}$ is definable in* TMSO+zero*.*

Proof sketch. It is not difficult to see that transducers definable in TMSO+zero are closed under composition and combination, and that they contain child number transducers, transducers recognised by Mealy machines, and characteristic transducers of languages definable in TMSO. The only difficulty is the characteristic transducers of the languages

$$\mathsf{Z}_n \stackrel{\text{def}}{=} \{t \in \mathsf{trees}\{1,\ldots,n,\bot\} : \mathsf{zero} \ni \{\pi \in 2^\omega : \begin{cases} \pi \text{ does not visit } \bot\text{, and} \\ \pi \text{ has defined and even maxinf} \end{cases} \}\}$$

To define the characteristic transducer of such a language in TMSO+zero, we need a formula $\varphi_1(x)$ which is true in a node x if and only if the subtree of x satisfies the above property. The formula is

$$\mathsf{zero}\pi \ \big(x \in \pi \wedge \text{“the maxinf is even for } \pi \text{ and there is no appearance of } \bot \text{ after } x\text{”}\big).$$

To prove that the formula is correct, we use prefix independence, i.e. condition 3 in Definition 3. ◀

The above lemma completes the transducer-to-logic implication. Indeed, suppose that the characteristic transducer

$$\mathsf{trans}L : \mathsf{trees}\Sigma \to \mathsf{trees}2$$

of a language $L \subseteq \mathsf{trees}\Sigma$ is definable in TMSO+zero. Apply the above lemma, yielding a formula $\varphi_1(x)$. A tree belongs to L if and only if this formula selects the root.

The rest of Appendix C is devoted to proving the logic-to-transducer implication. Our strategy is as follows. In Section C.1 we prove an important special case, namely that for every ω-regular language L, the class $\mathcal{T}$ contains the characteristic transducer of the language “zero contains the set of paths with labels in L”. In Section C.2, we show that the composition method works for the logic TMSO+zero, i.e. to determine the truth value of a formula on a tree, it suffices to determine the truth value of other – more complicated – formulas on pieces of the tree. Finally, in Section C.3, we use the results from Section C.1 and C.2 to complete the proof of Theorem 10.

C.1 Applying zero to an ω-regular language

The goal of Section C.1 is to prove the following lemma. If $L \subseteq \Sigma^\omega$ is a language of ω-words, then define $\mathsf{zero}L$ to be the set of trees $t \in \mathsf{trees}\Sigma$ such

$$\mathsf{zero} \ni \{\pi \in 2^\omega : \pi \text{ is contained in } t \text{ and the sequence of its labels is in } L\}$$

▶ **Lemma 29.** *If $L \subseteq \Sigma^\omega$ is ω-regular, then the characteristic transducer of* $\mathsf{zero}L$ *is in $\mathcal{T}$.*

What is the difficulty?

To explain the difficulty in proving Lemma 29, we begin by giving a wrong proof. To recognise ω-regular languages, we use deterministic parity automata, but viewed as Mealy machines executed on infinite words. The following is a corollary of the fact that deterministic parity automata recognise all ω-regular languages.

▶ Fact 30. A language $L \subseteq \Sigma^\omega$ is ω-regular if and only if there is some

$$f : \Sigma^\omega \to \{1, \ldots, n\}^\omega$$

recognised by a Mealy machine such that L is the inverse image f^{-1}(parity condition).

Motivated by the above fact, it is tempting to use the following wrong proof of Lemma 29.

Wrong Proof of Lemma 29. Apply Fact 30 to L, yielding a Mealy machine f. View this Mealy machine as a tree transducer

$$f : \mathsf{trees}\Sigma \to \mathsf{trees}\{1, \ldots, n\},$$

which belongs to $\mathcal{T}$, like every tree transducer recognised by a Mealy machine. Our hope would be that the following composition, which belongs to $\mathcal{T}$, is the characteristic transducer of $\mathsf{zero}L$:

$$\mathsf{trees}\Sigma \xrightarrow{\quad f \quad} \mathsf{trees}\{1, \ldots, n\} \xrightarrow{\quad \mathsf{transZ}_n \quad} \mathsf{trees}2$$

By unraveling definitions, to prove this hope, we would need to show that

$$\mathsf{subtree}_x(t) \in \mathsf{zero}L \quad \text{iff} \quad \mathsf{subtree}_x(f(t)) \in \mathsf{Z}_n \qquad \text{for every } t \in \mathsf{trees}\Sigma \text{ and } x \in 2^*$$

The problem with this reasoning is that it would require

$$\mathsf{subtree}_x(f(t)) = f(\mathsf{subtree}_x(t)), \tag{8}$$

because only the right side of the equality describes the run of the Mealy machine f on the subtree of t rooted in x. Such an equality does not hold in general. ◀

Motivated by the reasoning error described in the above proof, we call a tree transducer *memoryless* if it satisfies the equality in (8). More precisely, a transducer $f : \mathsf{trees}\Sigma \to \mathsf{trees}\Gamma$ is called memoryless if the following diagram commutes for every node $x \in 2^*$:

$$\begin{array}{ccc} \mathsf{trees}\Sigma & \xrightarrow{\ \mathsf{subtree}_x\ } & \mathsf{trees}\Sigma \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle f} \\ \mathsf{trees}\Gamma & \xrightarrow{\ \mathsf{subtree}_x\ } & \mathsf{trees}\Gamma \end{array} \tag{9}$$

The idea is that the transducer has no memory about the path leading up to a node x, but it can depend on the subtree of x. For example, the characteristic transducer of every language is memoryless. We now turn to a correct proof of Lemma 29, where memoryless transducers play an important role.

Ordered Mealy machines

Call a Mealy machine *ordered* if there is total order on its state space such that every transition keeps the same state or makes it smaller. Consquently, runs are non-increasing with respect to the total order. We begin by proving the special case of Lemma 29 for ordered Mealy machines.

▶ **Lemma 31.** *Let $f : \Sigma^\omega \to \{1, \ldots, n\}$ be recognised by an ordered Mealy machine. Then $\mathcal{T}$ contains the characteristic transducer of the language*

$$\mathsf{zero}\{w \in \Sigma^\omega : f(w) \text{ satisfies the parity condition}\} = f^{-1}(\mathsf{Z}_n).$$

Proof. The proof is by induction on the number of states in the Mealy machine. Suppose that we have proved Lemma 29 for all ordered Mealy machines with at most k states, and consider an ordered Mealy machine with states

$$q_0 > q_1 > \cdots > q_k.$$

We assume that the initial state is q_0, since otherwise we would be done by the induction assumption. Define

$$f_i : \Sigma^\omega \to \{1, \ldots, n\} \qquad \text{for } i \in \{0, \ldots, k\}$$

to be the function recognised by the Mealy machine obtained from the original one by changing the initial state to q_i, and define L_i to be the language as in the statement of the lemma, except that f_i is used instead of f. In particular, the language from the statement of the lemma is L_0. For $i \in \{0, \ldots, k\}$ define Σ_i to be

$$\Sigma_i = \{a \in \Sigma : \text{when reading } a \text{ in state } q_0, \text{ the Mealy machine goes to state } q_i\}.$$

Because the Mealy machine is deterministic, these sets form a partition of Σ.

Consider a memoryless transducer

$$h : \mathsf{trees}\Sigma \to \mathsf{trees}\{1, \ldots, n, \top, \bot\}$$

defined as follows. Let $t \in \mathsf{trees}\Sigma$ and let x be a node, whose label in t is $a \in \Sigma$. If $a \in \Sigma_0$, then the label of x in $h(t)$ is the output produced by the Mealy machine when executing the transition from state q_0 to back to state q_0 over the input letter a. If $a \in \Sigma_i$ for some $i \in \{1, \ldots, k\}$ then the label of x in $h(t)$ is $\top$ if both child subtrees of t are in L_i, and $\bot$ otherwise.

▶ Claim 32. The transducer h is in $\mathcal{T}$.

Proof. For every Σ, the class $\mathcal{T}$ contains the transducer

$$\mathsf{trees}\Sigma \to \mathsf{trees}(\Sigma \times \Sigma)$$

which maps each node the labels of both of its children; this by using characteristic transducers of languages definable in WMSO with path quantifiers. Therefore, the claim follows by using the induction assumption. ◀

Define the tree language

$$K = \mathsf{zero}\{w \in \{1, \ldots, n, \top, \bot\}^\omega : \begin{cases} w \text{ has a prefix in } \{1, \ldots, n\}^*\top; \text{ or} \\ w \text{ contains neither } \bot \text{ nor } \top \text{ and satisfies the parity condition} \end{cases}\}.$$

▶ Claim 33. The characteristic transducer of K is in $\mathcal{T}$.

Proof. Define

$$g : \mathsf{trees}\{1, \ldots, n, \top, \bot\} \to \mathsf{trees}\{1, \ldots, n, \bot\}$$

to be the transducer which replaces $\top$ by $\bot$, which is memoryless. Assuming that zero is prefix independent and nontrival (i.e. does not contain 2^ω), K is equal to the intersection

$$g^{-1}(\mathsf{Z}_n) \qquad \cap \qquad \underbrace{\text{“every node with label } \top \text{ has an ancestor with label } \bot\text{”}}_{M}.$$

Therefore, the characteristic transducer of K is obtained by taking the pointwise minimum of the following two transducers, which are in $\mathcal{T}$:

$$\mathsf{transZ}_n \circ g, \quad \mathsf{trans}M \qquad : \qquad \{1, \ldots, n, \top, \bot\} \to \mathsf{trees}2$$

◀

▶ Claim 34. For $t \in \mathsf{trees}\Sigma$, $t \in L_0$ if and only if $h(t) \in K$.

Proof. Fix a tree $t \in \mathsf{trees}\Sigma$. Define

$$\Pi = \{\pi \in 2^\omega : \text{the labels of } \pi \text{ in } f(t) \text{ satisfy the parity condition}\}.$$

By definition, t belongs to L_0 if and only if zero contains the set Π. Define $X \subseteq 2^*$ to be those nodes of t which have a label outside Σ_0, but all their proper ancestors have label in Σ_0. For $x \in X$, define Π_x to be those paths in Π which pass through x. The nodes from X form an antichain, and therefore the sets $\{\Pi_x\}_{x \in X}$ are disjoint. Finally, define $\Pi_\bot$ to be those paths in Π which are in none of the sets Π_x, i.e. those paths in t that only visit labels from Σ_0. Because zero is closed under subsets and countable unions, the tree t belongs to L_0 if and only if zero contains all sets Π_x with $x \in X \cup \{\bot\}$. Finally, by prefix-independence, the set Π_x belongs to zero if and only if the label of x in $h(t)$ is $\bot$. ◀

Therefore, by Claim 32 and closure of $\mathcal{T}$ under composition, the lemma will follow once we prove that the characteristic transducer of the language L_0 is the composition $\mathsf{trans}K \circ h$:

$$\mathsf{trees}\Sigma \xrightarrow{\quad h \quad} \mathsf{trees}(\{1, \ldots, n, \top, \bot\}) \xrightarrow{\quad \mathsf{trans}K \quad} \mathsf{trees}2 .$$

By definition of characteristic transducers, we need to show that

$$\mathsf{subtree}_x(t) \in L_0 \quad \text{iff} \quad \mathsf{subtree}_x(h(t)) \in K \qquad \text{for every } t \in \mathsf{trees}\Sigma \text{ and } x \in 2^*$$

Because h is memoryless, the equivalence becomes

$$\mathsf{subtree}_x(t) \in L_0 \quad \text{iff} \quad h(\mathsf{subtree}_x(t)) \in K \qquad \text{for every } t \in \mathsf{trees}\Sigma \text{ and } x \in 2^*,$$

which follows from Claim 32. ◀

Having proved proved the special case of Lemma 29 for ordered Mealy machines, we now move to the general case. We use the following lemma, which factors every Mealy machine through an ordered one in a way that is consistent with taking subtrees.

▶ **Lemma 35.** *Let $f : \mathsf{trees}\Sigma \to \mathsf{trees}\Gamma$ be recognised by a Mealy machine. There exists an alphabet Δ and tree transducers recognised by Mealy machines*

$$\mathsf{trees}\Sigma \xrightarrow{g} \mathsf{trees}\Delta \qquad \mathsf{trees}\Delta \xrightarrow{h} \mathsf{trees}\Gamma ,$$

such that the Mealy machine recognising h is ordered and the following diagram commutes for every $x \in 2^*$

$$\begin{array}{ccccc} \mathsf{trees}\Sigma & \xrightarrow{f} & \mathsf{trees}\Gamma & \xrightarrow{\mathsf{subtree}_x} & \mathsf{trees}\Gamma \\ {\scriptstyle g}\downarrow & & & & \uparrow{\scriptstyle h} \\ \mathsf{trees}\Delta & & \xrightarrow[\mathsf{subtree}_x]{} & & \mathsf{trees}\Delta \end{array}$$

Proof. Using the construction as in Lemma 12 from [3]. ◀

We are now ready to finish Section C.1 and prove Lemma 29.

Proof of Lemma 29. Let $L \subseteq \Sigma^\omega$ be an ω-regular language. We want to prove that $\mathcal{T}$ contains the characteristic transducer of the tree language $\mathsf{zero}L$. By Fact 30, one find a function $f : \mathsf{trees}\Sigma \to \mathsf{trees}\{1, \ldots, n\}$ recognised by a Mealy machine such that

$$t \in \mathsf{zero}L \qquad \text{iff} \qquad f(t) \in \mathsf{Z}_n \qquad \text{for every } t \in \mathsf{trees}\Sigma.$$

Apply Lemma 35 to the Mealy machine recognising f, yielding Δ, g and h. Let $t \in \mathsf{trees}\Sigma$ and let $x \in 2^*$ be a node. We have

$$\mathsf{subtree}_x(t) \in L \quad \underbrace{\text{iff}}_{\text{by definition of } f} \quad f(\mathsf{subtree}_x(t)) \in \mathsf{Z}_n \quad \underbrace{\text{iff}}_{\text{by Lemma 35}} \quad h(\mathsf{subtree}_x(g(t))) \in \mathsf{Z}_n.$$

The above equivalence proves that

$$\mathsf{trans}(h^{-1}(\mathsf{Z}_n)) \circ g.$$

is the characteristic transducer of L. The transducer g belongs to $\mathcal{T}$ because it is recognised by a Mealy machine, while the characteristic transducer of $h^{-1}(\mathsf{Z}_n)$ belongs to $\mathcal{T}$ by Lemma 31. ◀

C.2 Compositionality of TMSO+zero

In this section we show a composition theorem for TMSO+zero, which says that if a tree is cut into many pieces, then evaluating a formula in the whole reduces to evaluating similar formulas in the pieces.

Quantifier rank

We modify the logic TMSO+zero as follows, without changing its expressive power. For every input letter a we add a predicate "root has label a" of arity zero. Using this predicate, we can test the root label using a quantifier-free formula. We eliminate node variables, and keep only thin set variables. We lift the tree predicates to sets as follows:

- $\text{child}_i(X)$ says that all nodes in set X are i-th children;
- $X \leq Y$ says that some node in Y is a descendant of some node in X;
- $X \subseteq a$ says that all nodes in X have label a.

The above logic has the same expressive power as TMSO+zero, and from now on when talking about TMSO+zero we mean the above syntax.

Define the quantifier rank of a formula in TMSO+zero, according the the syntax above, possibly with free variables, to be the nesting depth of quantifiers in the formula, with all quantifiers treated the same. Call two trees over the same alphabet n-equivalent if they satisfy the same sentences of TMSO+zero that have quantifier rank at most n. We write $\equiv_n$ for n-equivalence, and write

$$\mathsf{trees}\Sigma_{/\equiv_n}$$

for the set of n-equivalence classes for trees over an input alphabet Σ. By induction on n one can prove that the above set is finite for every choice of n and Σ.

Truncation

For a prefix-closed set of nodes $X \subseteq 2^*$, define $\partial X \subseteq 2^*$ to be the nodes that are not in X, but have their parent in X. Let $n \in \mathbb{N}$ and $X \subseteq 2^*$ be a prefix-closed set of nodes. For a tree $t \in \mathsf{trees}\Sigma$ whose domain contains X, define the (n, X)-*truncation* of t to be the tree where the domain is $X \cup \partial X$ and the label of a node x is defined by

$$\begin{cases} \text{label of } x \text{ in } t & \text{if } x \in X \\ \mathsf{subtree}_x(t)_{/\equiv_n} & \text{if } x \in \partial X \end{cases}$$

In particular, the alphabet of the (n, X)-truncation is

$$\Sigma_{/n} \stackrel{\text{def}}{=} \Sigma \cup (\mathsf{trees}\Sigma)_{/\equiv_n}.$$

The goal of this Section C.2 is to prove the following lemma, which says that for every TMSO+zero formula φ there is some n such that truth value of φ in a tree is uniquely determined by its (n, X)-truncation, for every prefix closed set X. The lemma is proved by induction on the quantifier rank. To make the induction work, we need deal with free variables. We code free variables in a tree as follows: if $t \in \mathsf{trees}\Sigma$ is and X is a set of nodes, we write $t \otimes X \in \mathsf{trees}(\Sigma \times 2)$ for the tree obtained from t by extending the label of each node with a bit saying whether or not the node belongs to X.

▶ **Lemma 36.** *Let $\varphi(X_1, \ldots, X_k)$ be a formula of* TMSO+zero *over alphabet Σ. There exists some n and a* TMSO+zero *definable language*

$$L \subseteq \mathsf{trees}((\Sigma \times 2^k)_{/n})$$

such that for every prefix closed set $X \subseteq 2^*$, every thin sets $X_1, \ldots, X_k$ and $t \in \mathsf{trees}\Sigma$,

$$t \models \varphi(X_1, \ldots, X_k)$$

if and only if L contains the (n, X)-truncation of $t \otimes X_1 \otimes \cdots \otimes X_k$.

Proof. Induction on the size of φ. We only do the induction step where

$$\varphi(X_1, \ldots, X_k) = \mathsf{zero}\pi\ \psi(X_1, \ldots, X_k, \pi).$$

Because zero is a σ-ideal, a tree t satisfies the above formula if and only if both conditions below are satisfied:

1. **Internal.** The following set of paths is in zero:

$$\{\pi \in 2^\omega : \pi \text{ does not pass through } \partial X \text{ and } t \models \psi(X_1, \ldots, X_k, \pi)\}$$

2. **External.** For every $x \in \partial X$, the following set of paths is in zero:

$$\{\pi \in 2^\omega : \pi \text{ passes through } x \text{ and } t \models \psi(X_1, \ldots, X_k, \pi)\}$$

Lemma 36 will follow once we show that the internal and external conditions defined above can be expressed using TMSO+zero in terms of the (n, X)-truncation of $t \otimes X_1 \otimes \cdots \otimes X_k$, for some n. This is stated in the following lemma.

▶ **Lemma 37.** *There is some $n \in \mathbb{N}$ and language*

$$L_{int}, L_{ext} \subseteq \mathsf{trees}((\Sigma \times 2^k)_{/n})$$

definable in TMSO+zero *such that for every $t \in \mathsf{trees}\Sigma$ and subsets $X, X_1, \ldots, X_k$ of its domain such that X is prefix closed and $X_1, \ldots, X_k$ are thin:*

1. **Internal.** *The internal condition is satisfied if and only if* L_{int} *contains the* (n, X)-*truncation of* $t \otimes X_1 \otimes \cdots \otimes X_k$.
2. **External.** *The external condition is satisfied if and only if* L_{ext} *contains the* (n, X)-*truncation of* $t \otimes X_1 \otimes \cdots \otimes X_k$.

As mentioned above, the above lemma yields the induction step in Lemma 36. The rest of Section C.2 is devoted to proving Lemma 37.

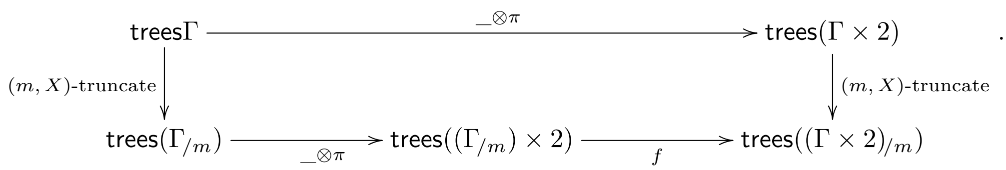
Internal.

Let us begin by proving the internal item in Lemma 37. The key to the proof is the following claim.

▶ Claim 38. For every alphabet Γ and $m \in \mathbb{N}$ there is a tree transducer

$$f : \mathsf{trees}((\Gamma_{/m}) \times 2) \to (\Gamma \times 2)_{/m})$$

which is definable in TMSO and makes the following diagram commute for every prefix closed X and every path π not passing through ∂X:



Proof of Claim 38. The claim says that for every tree $s \in \mathsf{trees}\Gamma$ and path π that does not pass through ∂X, the (m, X)-truncation of $s \otimes \pi$ can be computed by a tree transducer based on the (m, X)-truncation of s combined with the path π. Actually, a stronger result holds, namely that the label of a node x in the (m, X)-truncation of $s \otimes \pi$ is uniquely determined by membership $x \in \pi$ and the label of x in the (m, X)-truncation of s. This stronger result is obtained by unraveling the definition of (m, X)-truncation. ◀

Using the claim, show how to define the internal condition. By induction assumption, there is a language K definable in TMSO+zero and some m such that

$$t \models \psi(X_1, \ldots, X_k, \pi)$$

if and only if K contains the (m, X)-truncation of $t \otimes X_1 \otimes \cdots \otimes X_k \otimes \pi$. Therefore, the internal condition is equivalent to saying that zero contains the set of paths $\pi \in 2^\omega$ which satisfy both conditions below:

1. the (m, X)-truncation of $t \otimes X_1 \otimes \cdots \otimes X_k \otimes \pi$ belongs to K.
2. π does not pass through ∂X; and

Apply Claim 38 to m and the alphabet $\Sigma \times 2^k$, yielding a transducer f. The first condition above is rephrased as:

1. if s is the (m, X)-truncation of $t \otimes X_1 \otimes \cdots \otimes X_k$, then $f(s \otimes \pi) \in K$.

Whether or not the above holds can be expressed by a formula of TMSO+zero with a free variable π executed on the (m, X)-truncation of $t \otimes X_1 \otimes \cdots \otimes X_k$. By applying a zero quantifier to it, we get a sentence of TMSO+zero defining the internal condition in terms of the (m, X)-truncation of $t \otimes X_1 \otimes \cdots \otimes X_k$.

External

To deal with the external condition, it suffices to show that there is a language

$$L \subseteq \mathsf{trees}(((\Sigma \times 2^k)_{/n}) \times 2)$$

definable in TMSO+zero such that for every prefix closed set X and every $x \in \partial X$, if s is the (n, X)-truncation of $t \otimes X_1 \otimes \cdots \otimes X_k$ then $s \otimes \{x\} \in L$ if and only if the following set of paths is in zero:

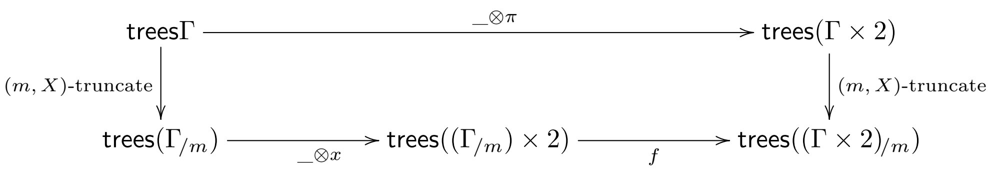
$$\{\pi \in 2^\omega : \pi \text{ passes through } x \text{ and } t \models \psi(X_1, \dots, X_k, \pi)\} \tag{10}$$

In the following, we write $x2^\omega$ for the set of paths which pass through x. A similar proof as for Claim 38 gives us the following claim.

▶ Claim 39. Let $m \in \mathbb{N}$ and let τ be an m-equivalence class of trees over some alphabet $\Gamma \times 2$. There is a transducer

$$f : \mathsf{trees}((\Gamma_{/m}) \times 2) \to (\Gamma \times 2)_{/m})$$

which is definable in TMSO and makes the following diagram commute



for every $x \in \partial X$ and $\pi \in x2^\omega$ such that τ is the m-equivalence class of

$$\mathsf{subtree}_x(t \otimes \pi)$$

where $t \in \mathsf{trees}\Gamma$ is the tree in the upper left corner of the diagram.

Using the above claim, we show how to define the external condition. By induction assumption, there is a language K and some m such that

$$t \models \psi(X_1, \dots, X_k, \pi)$$

holds if and only if K contains the (m, X)-truncation of $t \otimes X_1 \otimes \cdots \otimes X_k \otimes \pi$. In terms of K, condition (10) becomes

$$\{\pi \in 2^\omega : K \text{ contains the } (m, X)\text{-truncation of } t \otimes X_1 \otimes \cdots \otimes X_k \otimes x\pi\}$$

Let T be all the m-equivalence classes of trees over alphabet $\Sigma \times 2^{k+1}$. Because T is finite, in order to check that the above set is in zero, it suffices to check that for every $\tau \in T$ the set zero contains the set of paths π which satisfy both conditions below:

1. the (m, X)-truncation of $t \otimes X_1 \otimes \cdots \otimes X_k \otimes x\pi$ is in K; and
2. the m-equivalence class of $\mathsf{subtree}_x(t \otimes X_1 \otimes \cdots \otimes X_k \otimes x\pi)$ is τ.

By Claim 39, there is a transducer f such that the above two conditions become:

1. the (m, X)-truncation of $t \otimes X_1 \otimes \cdots \otimes X_k$ is in $\{s : f(s \otimes x) \in K\}$; and
2. the m-equivalence class of $\mathsf{subtree}_x(t \otimes X_1 \otimes \cdots \otimes X_k \otimes x\pi)$ is τ.

The first condition can be checked in TMSO+zero given x and the the (m, X)-truncation of $t \otimes X_1 \otimes \cdots \otimes X_k$. For the second condition, we observe that

$$\mathsf{subtree}_x(t \otimes X_1 \otimes \cdots \otimes X_k \otimes x\pi) = \mathsf{subtree}_x(t \otimes X_1 \otimes \cdots \otimes X_k) \otimes \pi.$$

Therefore, whether or not the second condition holds can be uniquely determined by the $(m+3)$-equivalence class of the subtree

$$\mathsf{subtree}_x(t \otimes X_1 \otimes \cdots \otimes X_k).$$

The reason we use $m+3$ instead of $m+1$ is that we need to load the path π from the labels in the tree into a path variable. Summing up, we have shown that there is a formula of TMSO+zero, with a free variable x, such that the set of paths in (10) belongs to zero if and only if the formula is true in $(m+3, X)$-truncation of $t \otimes X_1 \otimes \cdots \otimes X_k$. This completes the external case, and therefore also the proof of Lemma 36. ◀

C.3 Proof of Theorem 10

In this section we complete the proof of the logic-to-transducer implication in Theorem 10. We will prove the following lemma.

▶ **Lemma 40.** *Let Σ be an alphabet and let $n \in \mathbb{N}$. Then $\mathcal{T}$ contains the tree transducer*

$$\tau_n : \mathsf{trees}\Sigma \to \mathsf{trees}(\mathsf{trees}\Sigma_{/\equiv_n})$$

which labels each node by the n-equivalence class of its subtree.

Every sentence of TMSO+zero with quantifier rank n is equivalent to a finite union of n-equivalence classes, and therefore the above lemma gives the logic-to-transducer implication in Theorem 10, and thus completes the proof of the theorem. We prove the lemma by induction on n. The only information stored by 0-equivalence is the root label, and therefore the base case of $n = 0$ is trivial. We turn to the induction step. By definition, the $(n+1)$-equivalence class of a tree $t \in \mathsf{trees}\Sigma$ is uniquely determined by the set of formulas that are true in t and have the form $\mathsf{Q}X\varphi(X)$ where Q is one of the quantifiers "exists thin set" or "zero" and φ has quantifier rank at most n. Therefore, the induction step in Lemma 40 will follow from the following lemma.

▶ **Lemma 41.** *Let Q be one of the quantifiers "exists thin set" or "*zero*", and let $\varphi(X)$ have quantifier rank at most n. Let τ_n be as in Lemma 40. Then $\mathcal{T}$ contains a transducer*

$$f : \mathsf{trees}(\mathsf{trees}\Sigma_{/\equiv_n}) \to \mathsf{trees}2$$

such that $f \circ \tau_n$ is the characteristic transducer of $\mathsf{Q}X\varphi(X)$.

The rest of Section C.3 is devoted to proving the above lemma.

Eliminating zero

The following lemma says that if the domain of a partial tree is small enough, then there is no need to use the zero quantifier. Although very simple, the lemma explains why we use thin sets in the logic.

▶ **Lemma 42.** *Let φ be a sentence of* TMSO+zero*. There is a sentence of* TMSO *which is equivalent to φ over trees whose domain is a thin set.*

Proof. By conditions 1 and 2 in Definition 3, zero contains every countable set of paths. Therefore, for trees with a thin domain, the zero quantifier always returns true. ◀

The following is an immediate corollary of Lemmas 36 and 42.

▶ **Corollary 43.** *Let $\varphi(Y)$ be a formula of* TMSO+zero *with a single free variable Y. There is some $n \in \mathbb{N}$ and a language definable L in* TMSO *such that*

$$t \models \varphi(Y) \wedge Y \subseteq X \qquad \text{iff} \qquad L \text{ contains the } (n, X)\text{-truncation of } t.$$

holds for every tree t and every thin prefix-closed subset of its domain $X \subseteq 2^*$:

Proof of Lemma 41

We now prove Lemma 41. Let Q be one of the quantifiers "exists thin sets" or "zero" and let $\varphi(Y)$ have quantifier rank at most n. We need to show that $\mathcal{T}$ contains a transducer

$$f : \mathsf{trees}(\mathsf{trees}\Sigma_{/\equiv_n}) \to \mathsf{trees}2$$

such that $f \circ \tau_n$ is the characteristic transducer of $\mathsf{Q}X\varphi(Y)$. We deal with each quantifier separately.

- Consider the case when Q is existential quantification over thin sets, i.e. the formula is $\exists Y \varphi(Y)$. Apply Corollary 43 to $\varphi(Y)$ yielding some n and L. If Y is thin, then if we take X to be the closure of Y under prefixes, we also get a thin set. Therefore, by Corollary 43, a tree $t \in \mathsf{trees}\Sigma$ satisfies $\exists Y \varphi(Y)$ if and only if the (n, X)-truncation of t belongs to L for some thin prefix closed X. The above property is can be defined in TMSO using the tree $\tau_n(t)$. Therefore, the characteristic transducer of the property is in $\mathcal{T}$, this is the transducer f in the conclusion of Lemma 41.
- We are left with the case when Q is the zero quantifier. Consider a formula of the form $\mathsf{zero}\pi\varphi(\pi)$. Apply Corollary 43 to $\varphi(\pi)$, yielding a some n and L. By abuse of notation, we define the (n, π)-truncation of a tree to be the (n, X)-truncation where X is the set of nodes on the path π. Since the only path contained in such an X is π, and zero contains all singletons, it follows from Corollary 43 that $t \models \varphi(\pi)$ holds if and only if the (n, π)-truncation of t belongs to L.

  ▶ Claim 44. There is an ω-regular language $K \subseteq \Gamma^\omega$ and memoryless transducer

  $$h : \mathsf{trees}(\mathsf{trees}\Sigma_{/\equiv_n}) \to \Gamma$$

  in the class $\mathcal{T}$ such that for every tree $t \in \mathsf{trees}\Sigma$ and path π, the following conditions are equivalent:
  - the (n, π)-truncation of t belongs to L;
  - the labels on π in the tree $h(\tau_n(t))$ belong to K.

  Proof sketch. The transducer from the statement of the claim has the following type

  $$h : \mathsf{trees}(\mathsf{trees}\Sigma_{/\equiv_n}) \to \mathsf{trees}\Big(\underbrace{\overbrace{(\mathsf{trees}\Sigma_{/\equiv_n})}^{\text{left child}} \times \overbrace{(\mathsf{trees}\Sigma_{/\equiv_n})}^{\text{right child}} \times \overbrace{2}^{\text{child number}} \times \overbrace{\Sigma}^{\text{label}}}_{\Gamma}\Big).$$

  Given an input tree, h produces a tree where each node x is labelled by: the labels of its two children in the input tree, its child number, and the unique root label of trees in the

n-equivalence class that labels the node x in the input tree. This is clearly a memoryless procedure, and it can be implemented using TMSO, and therefore the transducer h belongs to $\mathcal{T}$. By definition, for every tree t and path π, the (n,π)-truncation of t is uniquely determined by the sequence of labels given by π in $h(\tau_n(t))$, and furthermore the function

$$\text{sequence of labels in } h(\tau_n(t)) \text{ given by } \pi \qquad \mapsto \qquad \text{the } (n,\pi)\text{-truncation of } t$$

is realised by an MSO transduction from ω-words to partial trees, in the sense of the book by Courcelle and Engelfriet. In particular, by the Backwards Translation Theorem in that book, every MSO definable property of the (n,π)-trnucation of t can be translated back to an MSO definable property of the sequence of labels in $h(\tau_n(t))$, thus proving the claim. ◀

Let us now complete the proof of Lemma 41. Let K and h be as in the above claim. By the claim, we know that for every tree $t \in \mathsf{trees}\Sigma$, node x and path π,

$$\mathsf{subtree}_x(t) \models \varphi(\pi)$$

is equivalent to saying that the labels on π in the tree $h(\tau_n(\mathsf{subtree}_x(t)))$ belong to K. Because both τ_n and h are memoryless,

$$h(\tau_n(\mathsf{subtree}_x(t))) = \mathsf{subtree}_x(h(\tau_n(t)))$$

It follows that

$$\mathsf{subtree}_x(t) \models \mathsf{zero}\pi\ \varphi(\pi) \qquad \text{iff} \qquad \mathsf{subtree}_x(h(\tau_n(t))) \in \mathsf{zero}K.$$

This means that the characteristic transducer of $\mathsf{zero}\pi\varphi(\pi)$ is equal to

$$\overbrace{\mathsf{trans}(\mathsf{zero}L)}^{\text{in } \mathcal{T} \text{ by Lemma 29}} \circ \overbrace{h}^{\text{in } \mathcal{T} \text{ by Claim 44}} \circ \tau_n.$$

Therefore, taking g to be the composition of the first two transducers above, we get the conclusion of Lemma 41.