Satisfiability of ECTL* with constraints

Claudia Carapelle[a,1], Alexander Kartzow[a,b,2,*], Markus Lohrey[b]

[a] *Institut für Informatik, Universität Leipzig, Germany*
[b] *Department für Elektrotechnik und Informatik, Universität Siegen, Germany*

Abstract

We show that satisfiability and finite satisfiability for ECTL^* with equality-, order-, and modulo-constraints over $\mathbb{Z}$ are decidable. Since ECTL^* is a proper extension of CTL^* this greatly improves the previously known decidability results for certain fragments of CTL^*, e.g., the existential and positive fragments and EF. We also show that our choice of local constraints is necessary for the result in the sense that if we add the possibility to state non-local constraints over $\mathbb{Z}$ the resulting logic becomes undecidable.

Keywords: temporal logics with integer constraints, ECTL*, monadic second-order logic with the bounding quantifier

1. Introduction

Temporal logics like LTL, CTL or CTL^* are nowadays standard languages for specifying system properties in model-checking. They are interpreted over node labeled graphs (Kripke structures), where the node labels (also called atomic propositions) represent abstract properties of a system. Clearly, such an abstracted system state does in general not contain all the information of the original system state. Consider for instance a program that manipulates two integer variables x and y. A useful abstraction might be to introduce atomic propositions $v_{-2^{32}}, \ldots, v_{2^{32}}$ for $v \in \{x, y\}$, where the meaning of v_k for $-2^{32} < k < 2^{32}$ is that the variable $v \in \{x, y\}$ currently holds the value k, and $v_{-2^{32}}$ (respectively, $v_{2^{32}}$) means that the current value of v is at most -2^{32} (respectively, at least 2^{32}). It is evident that such an abstraction might lead to incorrect results in model-checking.

To overcome these problems, extensions of temporal logics with constraints have been studied. Let us explain the idea in the context of LTL. For a fixed relational structure $\mathcal{A}$ (typical examples for $\mathcal{A}$ are number domains like the integers or rationals extended with certain relations) one adds atomic formulas of the form $R(\mathsf{X}^{i_1}x_1, \ldots, \mathsf{X}^{i_k}x_k)$ (so called constraints) to standard LTL. Here, R is (a name of) one of the relations of the structure $\mathcal{A}$, $i_1, \ldots, i_k \geq 0$, and $x_1, \ldots, x_k$ are variables that range over the universe of $\mathcal{A}$. An LTL-formula containing such constraints is interpreted over (infinite) paths of a standard Kripke structure, where in addition every node (state) associates with each of the variables $x_1, \ldots, x_k$ an element of $\mathcal{A}$ (one can think of $\mathcal{A}$-registers attached to the system states). A constraint $R(\mathsf{X}^{i_1}x_1, \ldots, \mathsf{X}^{i_k}x_k)$ holds in a path $s_0 \to s_1 \to s_2 \to \cdots$ if the tuple $(a_1, \ldots, a_k)$, where a_j is the value of variable x_j at state s_{i_j}, belongs to the $\mathcal{A}$-relation R. In this way, the values of variables at different system states can be compared. In our example from the first paragraph, one might choose for $\mathcal{A}$ the structure $(\mathbb{Z}, <, \equiv, (\equiv_a)_{a \in \mathbb{Z}})$, where $<$ is the usual order on $\mathbb{Z}$, $\equiv$ is the equality relation,[3] $\equiv_a$ is the unary predicate that only holds for a. This

*Corresponding author
Email addresses: `carapelle@informatik.uni-leipzig.de` (Claudia Carapelle), `kartzow@eti.uni-siegen.de` (Alexander Kartzow), `lohrey@eti.uni-siegen.de` (Markus Lohrey)
[1]Supported by the DFG Research Training Group 1763 (QuantLA).
[2]Supported by the DFG research project GELO.
[3]The reader might be confused by the fact we denote the equality relation with $\equiv$. The reason is that later we have to consider relational structures over the same signature, where $\equiv$ is not necessarily the equality relation. To avoid confusion, we have decided to use the symbol $\equiv$ for the equality relation as part of relational structures.

structure has infinitely many predicates, which is not a problem with respect to satisfiability because any formula can only use finitely many of those predicates. Our main result actually is about an expansion of $(\mathbb{Z}, <, \equiv, (\equiv_a)_{a\in\mathbb{Z}})$. Then, one might for instance write down a formula $(<(x, \mathsf{X}^1 y))\,\mathsf{U}(\equiv_{100}(y))$ which holds on a path if and only if there is a point of time where variable y holds the value 100 and for all previous points of time t, the value of x at time t is strictly smaller than the value of y at time $t+1$.

In [13], Demri and Gascon studied LTL extended with constraints from a language IPC*. If we disregard succinctness aspects, these constraints are equivalent to constraints over the structure

$$\mathcal{Z} = (\mathbb{Z}, <, \equiv, (\equiv_a)_{a\in\mathbb{Z}}, (\equiv_{a,b})_{0\leq a<b}), \tag{1}$$

where $\equiv_{a,b}$ denotes the unary relation $\{a + xb \mid x \in \mathbb{Z}\}$ (expressing that an integer is congruent to a modulo b). The main result from [13] states that satisfiability of LTL with constraints from $\mathcal{Z}$ is decidable and in fact PSPACE-complete, and hence has the same complexity as satisfiability for LTL without constraints. We should remark that the PSPACE upper bound from [13] even holds for the succinct IPC*-representation of constraints used in [13].

In the same way as outlined for LTL above, constraints can be also added to logics as CTL, CTL* and even its extension ECTL* (then, constraints $R(\mathsf{X}^{i_1}x_1, \ldots, \mathsf{X}^{i_k}x_k)$ are path formulas). A weak form of CTL* with constraints from $\mathcal{Z}$ (where only integer variables at the same state can be compared) was first introduced in [7], where it is used to describe properties of infinite transition systems, represented by relational automata. It is shown in [7] that the model checking problem for CTL* over relational automata is undecidable.

Demri and Gascon [13] asked whether satisfiability of CTL* with constraints from $\mathcal{Z}$ over Kripke structures is decidable. This problem was investigated in [5, 16], where several partial results where shown: If we replace in $\mathcal{Z}$ the binary predicate $<$ by unary predicates $<_c = \{x \mid x < c\}$ for $c \in \mathbb{Z}$, then satisfiability for CTL* has been shown decidable by [16]. For the full structure $\mathcal{Z}$ satisfiability has been shown decidable for CEF$^+$, the fragment of CTL* which contains the existential and universal fragment of CTL* as well as EF, see [5] .

In this paper we deal with ECTL* [26, 27], which is a proper extension of CTL*, where the CTL* path formulas are replaced by the set of all regular properties of paths (represented by Büchi-automata or MSO-formulas). We prove that ECTL* with constraints over $\mathcal{Z}$ is decidable. Our proof is divided into two steps. The first step provides a tool to prove decidability of ECTL* with constraints over any structure $\mathcal{A}$ (called a concrete domain) over a countable signature σ which satisfies the property that the complement of any of its relations has to be definable in positive existential first-order logic over $\mathcal{A}$ (in this case we call $\mathcal{A}$ negation closed). Let $\mathcal{L}$ be a logic that satisfies the following three properties:

1. Satisfiability of a given $\mathcal{L}$-sentence over the class of infinite node-labeled trees is decidable.
2. $\mathcal{L}$ is closed under boolean combinations with monadic second-order formulas (MSO).
3. $\mathcal{L}$ is compatible with one dimensional first-order interpretations and with the k-copy operation.

A typical such logic is MSO itself. By Rabin's seminal tree theorem [25], satisfiability of MSO-sentences over infinite node-labeled trees is decidable, and Muchnik's theorem (cf. [28]) implies compatability of MSO with k-copying.

Assuming $\mathcal{L}$ has these properties, we prove that satisfiability of ECTL* with constraints over $\mathcal{A}$ is decidable if one can compute from a given finite subsignature $\tau \subseteq \sigma$ an $\mathcal{L}$-sentence ψ_τ (over the signature τ) such that for every countable τ-structure $\mathcal{B}$, $\mathcal{B} \models \psi_\tau$ if and only if there is a homomorphism from $\mathcal{B}$ to $\mathcal{A}$ (i.e., a mapping from the domain of $\mathcal{B}$ to the domain of $\mathcal{A}$ that preserves all relations from τ). We say that the structure $\mathcal{A}$ has the property EHD($\mathcal{L}$) if such a computable function $\tau \mapsto \psi_\tau$ exists. EHD stands for "existence of homomorphism is definable". For instance, the structure $(\mathbb{Q}, <, \equiv)$ has the property EHD(MSO) (cf. Example 10).

It is not clear whether $\mathcal{Z}$ from (1) has the property EHD(MSO) (we conjecture that it does not). Hence, we need a different logic. It turns out that $\mathcal{Z}$ has the property EHD(Bool(MSO, WMSO+B)), where WMSO+B is the extension of weak monadic second-order logic (where only quantification over finite subsets is allowed) with the bounding quantifier B and Bool(MSO, WMSO+B) stands for all Boolean combinations of MSO and WMSO+B sentences. A formula $\mathsf{B}X\,\varphi$ holds in a structure $\mathcal{A}$ if and only if there exists a bound $b \in \mathbb{N}$ such that for every finite subset B of the domain of $\mathcal{A}$ with $\mathcal{A} \models \varphi(B)$ we have $|B| \leq b$. Recently, Bojańczyk and Toruńczyk have shown that satisfiability of WMSO+B over infinite node-labeled trees is decidable [3]. Thus, WMSO+B is a candidate logic for our method. Unfortunately, WMSO+B is not closed under Boolean combinations with MSO-sentences. Thus, we consider the logic $\mathcal{L} = \mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B})$ that consists of all Boolean combinations of MSO and WMSO+B-sentences. Fortunately, the decidability proof for WMSO+B can be extended to $\mathcal{L}$ (cf. Section 2.3). Moreover, $\mathcal{L}$ is compatible with one-dimensional first-order interpretations and with the k-copy operation (cf. Proposition 8). Thus, $\mathcal{L}$ is a suitable logic for our approach that allows to show that satisfiability of ECTL^* with constraints from $\mathcal{Z}$ is decidable. Our proof that $\mathcal{Z}$ from (1) has the property $\mathsf{EHD}(\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B}))$ actually only needs rather weak assumption on the unary predicates (which are satisfied for the unary relations $\equiv_a$ and $\equiv_{a,b}$), see Section 5.3.

While it would be extremely useful to add successor constraints ($y = x + 1$) to $\mathcal{Z}$, this would lead to undecidability even for LTL [12] and the very basic description logic $\mathcal{ALC}$ [22], which is basically multi-modal logic. Nonetheless $\mathcal{Z}$ allows qualitative representation of increment, for example $x = y + 1$ can be abstracted by $(y > x) \wedge \bigvee_{i=-2^k}^{2^k-1} (\equiv_{i,2^k}(x) \wedge \equiv_{i+1,2^k}(y))$ where k is a large natural number. This is why temporal logics extended with constraints over $\mathcal{Z}$ seem to be a good compromise between (unexpressive) total abstraction and (undecidable) high concretion.

Since satisfiability of pure ECTL^* (without constraints) is non-elementary (which follows from the fact that MSO over infinite words is non-elementary), the same lower complexity bound also holds for ECTL^* with constraints from $\mathcal{Z}$. On the other hand, satisfiability of CTL^* is 2EXPTIME-complete [15], but unfortunately, our proof does not yield any complexity bound for satisfiability of CTL^* with constraints from $\mathcal{Z}$. The boolean combinations of (WMSO+B)-sentences and MSO sentences that have to be checked for satisfiability (over infinite trees) are of a simple structure, in particular their quantifier depth is not high, but no complexity statement for satisfiability of WMSO+B is made in [3], and it seems to be difficult to analyze the algorithm from [3] (although it seems to be elementary for a fixed quantifier depth). It is based on a construction for cost functions over finite trees from [8], where the authors only note that their construction seems to have very high complexity.

Let us stress that the approach to decide satisfiability of ECTL^* with constraints via property $\mathsf{EHD}(\mathcal{L})$ is rather general and not restricted to be used for the integers. For instance, we are currently working on the application of this method on concrete domains that are treelike. Moreover, with a slight variation the approach can also deal with finite satisfiability, i.e., with the problem whether a given formula has a model whose underlying Kripke structure is finite. Recall that property $\mathsf{EHD}(\mathcal{L})$ requires that we can compute for each finite subsignature τ a formula in $\mathcal{L}$ that describes the existence of a homomorphism to the τ-reduct of the concrete domain. Analogously, let a σ-structure $\mathcal{A}$ have the property $\mathsf{EHD}_{\mathsf{fin}}(\mathcal{L})$ if we can compute for each finite subsignature τ of σ a formula φ_τ in $\mathcal{L}$ such that φ_τ is satisfied by some structure $\mathcal{B}$ if and only if there is a homomorphism from $\mathcal{B}$ to $\mathcal{A}$ whose image is finite. From the results in this paper it follows that finite satisfiability of ECTL^* with constraints over a concrete domain $\mathcal{A}$ is decidable if $\mathcal{A}$ is negation-closed and has property $\mathsf{EHD}_{\mathsf{fin}}(\mathcal{L})$ for some logic $\mathcal{L}$ with the same properties as required in the result for satisfiability. We use this fact to show that finite satisfiability of ECTL^*-formulas with constraints over the integers is also decidable. Moreover, this result applies to finite satisfiability of ECTL^*-formulas with constraints in any linear order.

A short conference version of this paper appeared in [6]. There, we only consider the fragment CTL^* of ECTL^* and do not consider finite satisfiability.

1.1. Related work

In the area of knowledge representation, extensions of description logics with constraints from concrete domains have been intensively studied, see [20] for a survey. In [21], it was shown that the extension of the description logic $\mathcal{ALC}$ with constraints from $(\mathbb{Q}, <, \equiv)$ has a decidable (EXPTIME-complete) satisfiability problem with respect to general TBoxes (also known as general concept inclusions). Such a TBox can be seen as a second $\mathcal{ALC}$-formula that has to hold in all nodes of a model. Our decidability proof is partly inspired by the construction from [21], which in contrast to our proof is purely automata-theoretic. Further results for description logics and concrete domains can be found in [22, 23].

There are other extensions of temporal logics that allow to reason about structures with added data values. In particular in the linear time setting several extensions that allow to specify properties of so called data words like MTL [18], freeze LTL [14] and its extension TPTL [1]. In general, satisfiability for these logics is undecidable and researchers have concentrated on fragments, see [14] for an overview.

1.2. Outline of the Paper

In the next section we recall basic fact, introduce our notation and present basic definitions concerning structures, constraint structures and MSO logic with its variants WMSO+B and and constraint-path MSO. Section 3 introduces syntax and semantics of our extension of ECTL^* with constraints over an arbitrary relational concrete domain. Section 4 contains the main technical core of our paper. It presents the relation between satisfiability of ECTL^* with constraints over some structure $\mathcal{A}$ and the question whether $\mathcal{A}$ has property $\mathsf{EHD}(\mathcal{L})$ (as a byproduct, we also show that ECTL^* with constraints has the tree model property). We then use this result in Section 5 to prove that satisfiability of ECTL^* with constraints in the integers with order, equality, constants and modulo-predicates is decidable. We then turn to finite satisfiability in Section 7 where we show that a formula has a finite model if and only if it has a model that only uses a finite substructure of the concrete domain. This allows to reuse our results from the satisfiability case in order to prove that finite satisfiability for ECTL^* with constraints in any linear order or in $\mathcal{Z}$ from (1) is decidable. The last section then shows that our results crucially rely on the fact that we only added local constraints to ECTL^*. Even for the far weaker logic LTL adding a kind of future operator in constraints leads to undecidability for every concrete domain that is an infinite linear order.

2. Basic Notions

We abbreviate the set $\{1, \ldots, d\}$ by $[1, d]$. For a function $\eta : A \to B$ and elements $a \in A$ and $b \in B$, $\eta[a \mapsto b]$ indicates the function which maps a to b and otherwise coincides with η.

2.1. Structures

Throughout this paper, we fix a countably infinite sets of atomic propositions $\mathbb{P}$ and function symbols $\mathbb{F}$. Function symbols are usually denoted by $f_1, f_2, \ldots, g_1, g_2, \ldots$. A Kripke structure (over $\mathbb{P}$) is a triple $\mathcal{K} = (D, \to, \rho)$, where

(i) D is an arbitrary set of nodes,
(ii) $\to \subseteq D \times D$ is a binary relation such that for each $u \in D$ there is a $v \in D$ with $u \to v$, i.e., $(D, \to)$ is a directed graph without dead ends, and
(iii) $\rho : D \to 2^{\mathbb{P}}$ is a labeling function that assigns to every node a finite set of atomic propositions such that $\bigcup_{v \in D} \rho(v)$ is finite, i.e., only finitely many propositions appear in $\mathcal{K}$.

A (relational) *signature* is a countable (finite or infinite) set σ of relation symbols. Every relation symbol $R \in \sigma$ has an associated arity $\mathsf{ar}(R) \geq 1$. A *σ-structure* is a pair $\mathcal{A} = (A, I)$, where A is a non-empty set (the universe of the structure) and I maps every $R \in \sigma$ to an $\mathsf{ar}(R)$-ary relation over A.

Quite often, we identify the relation $I(R)$ with the relation symbol R, and we specify a σ-structure as $(A, R_1, R_2, \ldots)$ where $\sigma = \{R_1, R_2, \ldots\}$.

Given $\mathcal{A} = (A, R_1, R_2, \ldots)$ and given a subset $B \subseteq A$, for each R_i we define $R_i{\restriction_B} = R_i \cap B^{\mathsf{ar}(R)}$ to be the restriction of R_i to $B^{\mathsf{ar}(R_i)}$. We write $\mathcal{A}{\restriction_B}$ for the induced substructure $(B, R_1{\restriction_B}, R_2{\restriction_B}, \ldots)$.

We now introduce *constraint graphs*. These are two-sorted structures where one part is a Kripke-Structure and the other part is some σ-structure called the *concrete domain*. The two are connected by (interpretations of) the functions from $\mathbb{F}$ that send nodes from the Kripke-Structure to elements of the concrete domain. Constraint graphs are the structures in which we evaluate the new logic ECTL^* with constraints.

Definition 1. Let σ be a relational signature. An *$\mathcal{A}$-constraint graph* $\mathfrak{C}$ is a tuple $(\mathcal{A}, \mathcal{K}, \gamma)$ where:

- $\mathcal{A} = (A, I)$ is a σ-structure (the *concrete domain*),

- $\mathcal{K} = (D, \to, \rho)$ is a Kripke structure (called the underlying Kripke structure of $\mathfrak{C}$), and
- for each $f \in \mathbb{F}$, $\gamma(f) : D \to A$ is the interpretation of the function symbol f connecting elements of the Kripke structure with element from the concrete domain.

We will also write $f^{\mathfrak{C}}$ for the function $\gamma(f)$. Moreover, we write constraint graph instead of $\mathcal{A}$-constraint graph if no confusion arises.

2.2. Trees and Paths

We first introduce the notion of *Kripke trees* and *Kripke n-trees*. Then we naturally lift the concept of unfolding of Kripke structures into trees to the setting of constraint graphs and finally introduce our notation on paths and their induced Kripke paths and constraint Kripke paths.

A *Kripke tree* is a Kripke structure of the form $\mathcal{T} = (D, \to, \rho)$, where $(D, \to)$ is a rooted tree. Formally, $\mathcal{T}$ is a Kripke tree if it is isomorphic to a Kripke structure of the form $(D, \to, \rho)$, where $D \subseteq \Gamma^*$ is a prefix-closed set of strings over an alphabet Γ (of arbitrary cardinality) and $u \to v$ if and only if $v = ua$ for some $a \in \Gamma$. In case $D = [1, n]^*$ for $n \in \mathbb{N}$, we say that $\mathcal{T}$ is a *Kripke n-tree*. If moreover $n = 1$ then $\mathcal{T}$ is a *Kripke path*. We call a constraint graph $\mathfrak{T} = (\mathcal{A}, \mathcal{T}, \gamma)$ an *$\mathcal{A}$-constraint tree* (respectively, *$\mathcal{A}$-constraint n-tree*, *$\mathcal{A}$-constraint path*) if $\mathcal{T}$ is a Kripke tree (respectively, Kripke n-tree, Kripke path).

Fix a Kripke structure $\mathcal{K} = (D, \to, \rho)$. An *infinite $\mathcal{K}$-path* is an infinite sequence $P = d_0 d_1 d_2 \cdots$ such that $d_i \in D$ and $d_i \to d_{i+1}$ for all $i \geq 0$. For $i \geq 0$ we define the node $P(i) = d_i$. A *finite $\mathcal{K}$-path* is a finite non-empty prefix of an infinite $\mathcal{K}$-path.

For $d \in D$, the *unfolding* of $\mathcal{K}$ from d, denoted by $\mathsf{Unf}(\mathcal{K}, d)$, is the Kripke tree $\mathcal{T} = (T, \to', \rho')$ where

- T is the set of finite $\mathcal{K}$-paths P with $P(0) = d$,
- $\to'$ is defined to be the extension of paths by a single edge, i.e., for finite paths P_1 and P_2 from T we have $P_1 \to' P_2$ iff $P_2 = P_1 d'$ for a node $d' \in D$, and
- ρ' is given by "last-node semantics", i.e., for every $d_0 d_1 \cdots d_n \in T$ we set $\rho'(d_0 d_1 \cdots d_n) = \rho(d_n)$.

The unfolding of a Kripke structure naturally lifts to constraint graphs. If $\mathfrak{C}$ is an $\mathcal{A}$-constraint graph with underlying Kripke structure $\mathcal{K} = (D, \to, \rho)$ and $d \in D$, then we denote by $\mathsf{Unf}(\mathfrak{C}, d)$ the $\mathcal{A}$-constraint tree with underlying Kripke tree $\mathsf{Unf}(\mathcal{K}, d)$, where $f^{\mathsf{Unf}(\mathfrak{C},d)}(d_0 d_1 \cdots d_n) = f^{\mathfrak{C}}(d_n)$ for all $f \in \mathbb{F}$ and all finite paths $d_0 d_1 \cdots d_n$ $(d_0 = d)$.

Given an infinite $\mathcal{K}$-path $P = d_0 d_1 d_2 \cdots$, we naturally identify P with the substructure of $\mathcal{T} = \mathsf{Unf}(\mathcal{K}, d_0)$ induced by the finite non-empty prefixes of P. Thus, P naturally induces a *Kripke path* $\mathcal{T}\restriction_P$, which we usually denote by $\mathcal{P}$. Moreover, for $\mathfrak{C}$ a constraint graph with underlying Kripke structure $\mathcal{K}$, P also induces a *constraint path* $\mathfrak{P}$ in $\mathfrak{C}$, whose underlying Kripke structure is $\mathcal{P}$ and where $f^{\mathfrak{P}}$ $(f \in \mathbb{F})$ is obtained by restricting $f^{\mathsf{Unf}(\mathfrak{C},d_0)}$ to the non-empty finite prefixes of P. We call it the constraint path corresponding to P and occasionally denote it with $\mathfrak{C}\restriction_P$. Note that every constraint path in $\mathfrak{C}$ is an induced subgraph of an unfolding of $\mathfrak{C}$ from some node d. We lift the position notation for paths to Kripke paths and constraint paths by setting $\mathfrak{P}(i) = \mathcal{P}(i) = d_i$ for all $i \geq 0$.

2.3. MSO*,* WMSO+B*, and Constraint-Path-*MSO

Throughout the paper, we fix countably infinite sets $\mathbb{V}^0$ and $\mathbb{V}^1$ of element variables and set variables, respectively. We usually denote variables of $\mathbb{V}^0$ by $x, y, z, x_0, x_1, \ldots, y_0, y_1, \ldots$ and variables of $\mathbb{V}^1$ by $X, Y, Z, X_0, X_1, \ldots, Y_0, Y_1, \ldots$.

Monadic second-order logic (MSO) is the extension of first-order logic where also quantification over subsets of the underlying structure is allowed. Let us fix a signature σ. We define MSO-formulas over the signature σ by the following grammar, where $R \in \sigma$:

$$\varphi ::= R(x_1, \ldots, x_{\mathsf{ar}(R)}) \mid x = y \mid x \in X \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \exists x\, \varphi \mid \exists X\, \varphi. \tag{2}$$

MSO-formulas are evaluated in σ-structures in the usual way, where element variables range over elements of the structure and set variables range over subsets of the universe. *Weak monadic second-order logic* (WMSO) has the same syntax as MSO but second-order variables only range over finite subsets of the underlying universe.

Finally, WMSO+B is the extension of WMSO by the additional quantifier $\mathsf{B}X\,\varphi$ (the *bounding quantifier*) for $X \in \mathbb{V}^1$. The semantics of $\mathsf{B}X\,\varphi$ in the structure $\mathcal{A} = (A, I)$ is defined as follows: $\mathcal{A} \models \mathsf{B}X\,\varphi(X)$ if and only if there is a bound $b \in \mathbb{N}$ such that whenever $\mathcal{A} \models \varphi(B)$ for some finite subset $B \subseteq A$ then $|B| \leq b$. The dual quantifier is denoted by U. It is called the *unbounding quantifier* and $\mathsf{U}X\,\varphi = \neg\mathsf{B}X\,\varphi$ expresses that there are arbitrarily large finite sets that satisfy φ.

Example 2. For later use, we state some example formulas. Let $\varphi(x, y)$ be a WMSO-formula with two free first-order variables x and y. Let $\mathcal{A} = (A, I)$ be a structure and let $E_\varphi = \{(a, b) \in A \times A \mid \mathcal{A} \models \varphi(a, b)\}$ be the binary relation defined by $\varphi(x, y)$. We define the WMSO-formula $\mathsf{reach}_\varphi(x_1, x_2)$ to be

$$\exists Z\,\forall Y \Big[\Big(x_1 \in Y \wedge (\forall y\,\forall z\,(y \in Y \wedge z \in Z \wedge \varphi(y, z)) \to z \in Y)\Big) \to x_2 \in Y\Big]. \tag{3}$$

It is straightforward to prove that $\mathcal{A} \models \mathsf{reach}_\varphi(a, b)$ if and only if $(a, b) \in E_\varphi^*$, i.e., if we consider E_φ as the edge relation of the graph $\mathcal{G}_\varphi = (A, E_\varphi)$, then b is reachable from a in $\mathcal{G}_\varphi$. Note that reach_φ is the standard MSO-formula for reachability but restricted to some finite induced subgraph. Thus, its semantic seen as an MSO-formula is the same because b is reachable from a in the graph $\mathcal{G}_\varphi$ if and only if it is in some finite subgraph of $\mathcal{G}_\varphi$.

Given a set variable Z, we define $\mathsf{reach}^Z_\varphi(x_1, x_2)$ to be

$$x_1 \in Z \wedge \forall Y \subseteq Z\,\big[\big(x_1 \in Y \wedge \forall y\,\forall z\,(y \in Y \wedge z \in Z \wedge \varphi(y, z)) \to z \in Y\big) \to x_2 \in Y\big].$$

For every finite subset $B \subseteq A$, we have $\mathcal{A} \models \mathsf{reach}^B_\varphi(a, b)$ iff b is reachable from a in the subgraph $\mathcal{G}_\varphi{\restriction_B}$. Note that $\mathcal{A} \models \mathsf{reach}^B_\varphi(a, b)$ implies that $\{a, b\} \subseteq B$.

Let $\mathsf{ECycle}_\varphi = \exists x\,\exists y\,(\mathsf{reach}_\varphi(x, y) \wedge \varphi(y, x))$ be the WMSO-formula expressing that there is a cycle in $\mathcal{G}_\varphi$. We now restrict our attention to the case that the graph $\mathcal{G}_\varphi$ defined by $\varphi(x, y)$ is acyclic. Hence, the reflexive transitive closure E^*_φ is a partial order on A. Note that a finite set $F \subseteq A$ is an E_φ-path from $a \in F$ to $b \in F$ if and only if $(F, (E_\varphi \cap (F \times F))^*)$ is a finite linear order with minimal element a and maximal element b. Define the WMSO-formula $\mathsf{Path}_\varphi(x_1, x_2, Z)$ as

$$\forall x \in Z\,\forall y \in Z\,(\mathsf{reach}^Z_\varphi(x, y) \vee \mathsf{reach}^Z_\varphi(y, x))\ \wedge \mathsf{reach}^Z_\varphi(x_1, x) \wedge \mathsf{reach}^Z_\varphi(x, x_2).$$

For every structure $\mathcal{A}$ such that the graph $\mathcal{G}_\varphi$ is acyclic, we have $\mathcal{A} \models \mathsf{Path}_\varphi(a, b, P)$ if and only if P contains exactly the nodes that form an E_φ-path from a to b.

We finally define the WMSO+B-formula

$$\mathsf{BPaths}_\varphi(x, y) = \mathsf{B}Z\,\mathsf{Path}_\varphi(x, y, Z). \tag{4}$$

Under the assumption that $\mathcal{G}_\varphi$ is acyclic, $\mathcal{A} \models \mathsf{BPaths}_\varphi(a, b)$ if and only if there is a bound $k \in \mathbb{N}$ on the length of any E_φ-path from a to b.

Next, let Bool(MSO, WMSO+B) be the set of all Boolean combinations of MSO-formulas and (WMSO+B)-formulas. We use the following result.

Theorem 3 (cf. [3]). *One can decide whether for a given $n \in \mathbb{N}$ and a formula $\varphi \in$ Bool(MSO, WMSO+B) there is a Kripke n-tree $\mathcal{K}$ such that $\mathcal{K} \models \varphi$.*

Proof. This theorem follows from results of Bojańczyk and Toruńczyk [3, 4]. They introduced puzzles which can be seen as pairs $P = (A, C)$, where A is a parity tree automaton and C is an unboundedness condition C which specifies a certain set of infinite paths labeled by states of A. A puzzle accepts a tree $\mathcal{T}$ if there is an accepting run ρ of A on $\mathcal{T}$ such that for each infinite path π occurring in ρ, $\pi \in C$ holds. In particular, ordinary parity tree automata can be seen as puzzles with the trivial unboundedness condition. The proof of Theorem 3 combines the following results.

Lemma 4 ([3]). *From a given* (WMSO+B)*-formula φ and $n \in \mathbb{N}$ one can construct a puzzle P_φ such that φ is satisfied by some Kripke n-tree iff P_φ is nonempty.*

Lemma 5 ([3]). *Emptiness of puzzles is decidable.*

Lemma 6 (Lemma 17 of [4]). *Puzzles are effectively closed under intersection.*

Using these results, it is easy to prove Theorem 3: Let $\varphi \in \mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B})$. First, φ can be effectively transformed into a disjunction $\bigvee_{i=1}^{n}(\varphi_i \wedge \psi_i)$ where $\varphi_i \in \mathsf{MSO}$ and $\psi_i \in \mathsf{WMSO{+}B}$ for all i. By Lemma 4, we can construct a puzzle P_i for ψ_i. The MSO-formula φ_i can be translated into a parity tree automaton A_i [25]. Using Lemma 6 we compute a puzzle P_i' recognizing the intersection of P_i and A_i. Clearly, φ is satisfiable over Kripke n-trees if and only if there is an i such that $\varphi_i \wedge \psi_i$ is satisfiable over Kripke n-trees, if and only if there is an i such that P_i' is nonempty. By Lemma 5, the latter condition is decidable which concludes the proof of the theorem. $\square$

Besides classical MSO we use also a constraint version of MSO, denoted as $\mathsf{MSO}(\sigma)$ for some relational signature σ. $\mathsf{MSO}(\sigma)$ is the usual MSO for (colored) infinite paths with the successor function S extended by atomic formulas that describe local constraints over the concrete domain. Thus, given a σ-structure $\mathcal{A}$, $\mathsf{MSO}(\sigma)$ can be evaluated over the class of $\mathcal{A}$-constraint paths. The set of $\mathsf{MSO}(\sigma)$-formulas is defined by the following grammar:

$$\psi ::= p(x) \mid x_1 = S(x_2) \mid x \in X \mid \neg\psi \mid (\psi \wedge \psi) \mid \exists x\, \psi \mid \exists X\, \psi \mid R(f_1 S^{i_1}(x), \ldots, f_k S^{i_k}(x)) \tag{5}$$

where $p \in \mathbb{P}$, $x, x_1, x_2 \in \mathbb{V}^0$ are element variables, $X \in \mathbb{V}^1$ is a set variable, $R \in \sigma$ is a relation symbol of arity k, $i_1, \ldots, i_k \in \mathbb{N}$ and $f_1, \ldots, f_k \in \mathbb{F}$. We call $\theta = R(f_1 S^{i_1}(x), \ldots, f_k S^{i_k}(x))$ an *atomic constraint*, and we define $d(\theta) = \max\{i_1, \ldots, i_k\}$ to be the depth of θ. Note that the same element variable x is used in all arguments of a constraint. We use a different syntax for constraints than the one we indicated in the introduction of the paper (which, in turn, coincides with the one used in [6, 13, 16]). We choose this different presentation to adapt the constraints to the MSO setting as opposed to temporal logic. Therefore the successor function S takes the place of the next operator X, and we introduce a variable x to indicate the position at which we are evaluating the constraint, which was not needed in the old setting.

Remark 7. Setting $\tau = \{S\} \cup \mathbb{P}$, where S is a unary function symbol and all elements of $\mathbb{P}$ are considered to be unary predicates, $\mathsf{MSO}(\sigma)$ is MSO over τ extended by atomic constraints over σ.

As already mentioned $\mathsf{MSO}(\sigma)$-formulas are interpreted over $\mathcal{A}$-constraint paths for some σ-structure $\mathcal{A}$. Let σ be some signature, $\mathcal{A} = (A, I)$ a σ-structure, $\mathfrak{P}$ an $\mathcal{A}$-constraint path with underlying Kripke path $\mathcal{P} = (D, \to, \rho)$, and let $\eta : (\mathbb{V}^0 \cup \mathbb{V}^1) \to (D \cup 2^D)$ be a valuation function mapping element variables to elements and set variables to sets. The satisfaction relation $\models_{\mathsf{MSO}(\sigma)}$ is mostly defined as expected by structural induction interpreting S by the successor function in $\mathcal{P}$ (to keep notation simple, we write S also for the successor function in $\mathcal{P}$ induced by $\to$).

- $(\mathfrak{P}, \eta) \models_{\mathsf{MSO}(\sigma)} p(x)$ iff $p \in \rho(\eta(x))$.
- $(\mathfrak{P}, \eta) \models_{\mathsf{MSO}(\sigma)} x_1 = S(x_2)$ iff $\eta(x_1) = S(\eta(x_2))$.
- $(\mathfrak{P}, \eta) \models_{\mathsf{MSO}(\sigma)} x \in X$ iff $\eta(x) \in \eta(X)$.
- $(\mathfrak{P}, \eta) \models_{\mathsf{MSO}(\sigma)} \neg\psi$ iff it is not the case that $(\mathfrak{P}, \eta) \models_{\mathsf{MSO}(\sigma)} \psi$.
- $(\mathfrak{P}, \eta) \models_{\mathsf{MSO}(\sigma)} (\psi_1 \wedge \psi_2)$ iff $(\mathfrak{P}, \eta) \models_{\mathsf{MSO}(\sigma)} \psi_1$ and $(\mathfrak{P}, \eta) \models_{\mathsf{MSO}(\sigma)} \psi_2$.
- $(\mathfrak{P}, \eta) \models_{\mathsf{MSO}(\sigma)} \exists x\, \psi$ iff there is a $d \in D$ such that $(\mathfrak{P}, \eta[x \mapsto d]) \models_{\mathsf{MSO}(\sigma)} \psi$.
- $(\mathfrak{P}, \eta) \models_{\mathsf{MSO}(\sigma)} \exists X\, \psi$ iff there is an $E \subseteq D$ such that $(\mathfrak{P}, \eta[X \mapsto E]) \models_{\mathsf{MSO}(\sigma)} \psi$.

- $(\mathfrak{P}, \eta) \models_{\mathsf{MSO}(\sigma)} R(f_1 S^{i_1}(x), \ldots, f_k S^{i_k}(x))$ iff $(f_1^{\mathfrak{P}}(S^{i_1}(\eta(x))), \ldots, f_k^{\mathfrak{P}}(S^{i_k}(\eta(x)))) \in I(R)$.

For an $\mathsf{MSO}(\sigma)$-formula ψ the satisfaction relation only depends on the variables occurring freely in ψ. This motivates the following notation. If $\psi(X_1, X_2, \ldots, X_m)$ is an $\mathsf{MSO}(\sigma)$-formula where $X_1, \ldots, X_m$ are the only free variables, we write $\mathfrak{P} \models_{\mathsf{MSO}(\sigma)} \psi(A_1, \ldots, A_m)$ if and only if, for every valuation function η such that $\eta(X_i) = A_i$, we have $(\mathfrak{P}, \eta) \models_{\mathsf{MSO}(\sigma)} \psi$. Moreover, we write $\models$ instead of $\models_{\mathsf{MSO}(\sigma)}$ if no confusion arises.

We will use some abbreviations in $\mathsf{MSO}(\sigma)$ with the obvious semantics. In particular, we will use formulas of the form $p(S^i(x))$ for $i \geq 0$ and $p \in \mathbb{P}$, stating that the node $S^i(x)$ satisfies the proposition p.

2.4. $\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO} + \mathsf{B})$ *and the k-Copy Operation*

In this section we show a technical result stating that $\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO}{+}\mathsf{B})$ is compatible with the k-copy operation. The proof basically copies the known proofs for MSO and WMSO extended by a translation of bounding quantifiers. Readers that are not interested in the proof details can safely skip them. We will need this result later in Section 4.

We first define the k-copy operation. Let $k \in \mathbb{N}$ be some number and $\mathcal{A} = (A, I)$ some structure over the signature σ that does not contain relation symbols $\sim, P_1, P_2, \ldots, P_k$ ($\sim$ is binary and all P_i are unary). The *k-copy of $\mathcal{A}$*, denoted by $\mathsf{copy}_k(\mathcal{A})$, is the $(\sigma \cup \{\sim, P_1, P_2, \ldots, P_k\})$-structure $(A \times \{1, 2, \ldots, k\}, J)$ where

- for all $R \in \sigma$ of arity m, $J(R) = \{((a_1, i), (a_2, i), \ldots (a_m, i)) \mid (a_1, a_2, \ldots, a_m) \in I(R), 1 \leq i \leq k\}$,
- $J(\sim) = \{((a, i_1), (a, i_2)) \mid a \in A, 1 \leq i_1, i_2 \leq k\}$, and
- for each $1 \leq m \leq k$, $J(P_m) = \{(a, m) \mid a \in A\}$.

Proposition 8. *Let $k \in \mathbb{N}$ be some number, $\mathcal{A} = (A, I)$ some infinite structure over the signature σ, and $\tau = \sigma \cup \{\sim, P_1, P_2, \ldots, P_k\}$ an extension of σ by one fresh binary relation symbol $\sim$ and k fresh unary relation symbols $P_1, \ldots, P_k$. Given a* $\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO}{+}\mathsf{B})$ *sentence φ over τ, we can compute a* $\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO}{+}\mathsf{B})$ *sentence φ^k over σ such that $\mathsf{copy}_k(\mathcal{A}) \models \varphi$ if and only if $\mathcal{A} \models \varphi^k$.*

Proof. The proof is in 3 steps. We only do it for $\mathsf{WMSO}{+}\mathsf{B}$ in order to avoid handling a finite and an infinite version of existential set quantification. The extension to $\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO}{+}\mathsf{B})$ is straightforward. Instead of dealing with the bounding quantifier B directly, we deal with the unbounding quantifier U. This suffices since a bounding quantifier is equivalent to a negated unbounding quantifier. First we define a formula $\hat{\varphi}$. It uses element variables x, x' (respectively, set variables $X^1, \ldots, X^k$) for every element variable x (respectively, set variable X) used in φ. In addition, $\hat{\varphi}$ uses element variables $y_1, \ldots, y_k$ that identify the k different copies of $\mathcal{A}$ from the k-copy of $\mathcal{A}$ (for this purpose $y_1, \ldots, y_k$ are always assigned pairwise different values). Then we prove a strong connection between evaluations of φ on $\mathsf{copy}_k(\mathcal{A})$ and of $\hat{\varphi}$ on $\mathcal{A}$. Finally, we create φ^k from $\hat{\varphi}$ by quantification over the parameters $y_1, y_2, \ldots, y_k$ and show that φ^k has the desired property.

Step 1. We define $\hat{\varphi}$ from φ by case distinction on the structure of φ.

1. If $\varphi = P_i(x)$ for some $1 \leq i \leq k$, then $\hat{\varphi} := (x' = y_i)$.
2. If $\varphi = x_1 \sim x_2$ then $\hat{\varphi} := (x_1 = x_2)$.
3. If $\varphi = R(x_1, x_2, \ldots, x_r)$ for some $R \in \sigma$, then $\hat{\varphi} := R(x_1, x_2, \ldots, x_r) \wedge (x'_1 = x'_2 = \cdots = x'_r)$.
4. If $\varphi = x \in X$, then $\hat{\varphi} := \bigvee_{i=1}^{k} (x' = y_i \wedge x \in X^i)$
5. If $\varphi = \psi \wedge \chi$, then $\hat{\varphi} := \hat{\psi} \wedge \hat{\chi}$.
6. If $\varphi = \neg\psi$ then $\hat{\varphi} := \neg\hat{\psi}$.
7. If $\varphi = \exists x\, \psi$ then $\hat{\varphi} = \exists x\, \exists x' \, (\bigvee_{i=1}^{k} x' = y_i \wedge \hat{\psi})$.
8. If $\varphi = \exists X\, \psi$ then $\hat{\varphi} = \exists X^1\, \exists X^2 \cdots \exists X^k\, \hat{\psi}$.
9. If $\varphi = \mathsf{U} X\, \psi$ then $\hat{\varphi} = \bigvee_{i=1}^{k} \mathsf{U} X^i\, \exists X^1\, \exists X^2 \ldots \exists X^{i-1}\, \exists X^{i+1} \cdots \exists X^k\, \hat{\psi}$.

Step 2. Let $\varphi(x_1,\dots,x_n,X_1,\dots,X_m)$ be a $\mathsf{WMSO{+}B}$ formula. Fix some $\hat{a}_1,\dots,\hat{a}_k \in A$ such that $\hat{a}_i \neq \hat{a}_j$ for $i \neq j$ (recall that we assume A to be infinite), $a_1,\dots,a_n \in A$, $k_1,\dots,k_n \in \{1,\dots,k\}$, and finite subsets $A_1^1,\dots,A_1^k,A_2^1,\dots,A_2^k,\dots,A_m^k \subseteq A$. Fix a variable assignment η_k (in $\mathsf{copy}_k(\mathcal{A})$) such that $\eta_k(x_i) = (a_i,k_i)$ and $\eta_k(X_i) = \bigcup_{j=1}^k A_i^j \times \{j\}$. Fix another variable assignment η (in $\mathcal{A}$) such that $\eta(y_i) = \hat{a}_i$, $\eta(x_i) = a_i$, $\eta(x_i') = \hat{a}_{k_i}$ and $\eta(X_i^j) = A_i^j$ We claim that $(\mathsf{copy}_k(\mathcal{A}),\eta_k) \models \varphi$ if and only if $(\mathcal{A},\eta) \models \hat{\varphi}$.

The proof is by structural induction. Most cases are straightforward and can be copied from compatability proofs of (W)MSO with the k-copy operation (see [9]). The new case is the unbounding quantifier. For this case assume that $\varphi = \mathsf{U}X\,\psi$. By definition $(\mathsf{copy}_k(\mathcal{A}),\eta_k) \models \varphi$ if and only if for all $n \in \mathbb{N}$ there is a finite set $S \subseteq A \times \{1,\dots,k\}$ such that $|S| \geq n$ and $(\mathsf{copy}_k(\mathcal{A}),\eta_k[X \mapsto S]) \models \psi$. By induction hypothesis this is the case if and only if for all $n \in \mathbb{N}$ there are finite sets $S^1,\dots,S^k \subseteq A$ such that $|S^1| + \cdots + |S^k| \geq n$ and

$$(\mathcal{A},\eta[X^1 \mapsto S^1,\dots,X^k \mapsto S^k]) \models \hat{\psi}.$$

Noting that this means that one of the sets has size at least $\frac{n}{k}$, this statement is equivalent to the statement that for all $n' \in \mathbb{N}$ there are a $1 \leq j \leq k$ and finite sets $S^1,\dots,S^k$ such that $|S^j| \geq n'$ and

$$(\mathcal{A},\eta[X^1 \mapsto S^1,\dots,X^k \mapsto S^k]) \models \hat{\psi}.$$

By the pigeon hole principle, we can rewrite this to the statement that there is a $1 \leq j \leq k$ such that

$$(\mathcal{A},\eta) \models \mathsf{U}X^j\,\exists X^1\,\dots\exists X^{j-1}\,\exists X^{j+1}\,\dots\exists X^k\,\hat{\psi}.$$

This is evidently equivalent to

$$(\mathcal{A},\eta) \models \bigvee_{i=1}^{k} \mathsf{U}X^i\,\exists X^1\,\exists X^2\,\dots\exists X^{i-1}\,\exists X^{i+1}\,\dots\exists X^k\,\hat{\psi},$$

i.e., $(\mathcal{A},\eta) \models \hat{\varphi}$.

Step 3. Finally, for a sentence φ set $\varphi^k = \exists y_1 \exists y_2 \cdots \exists y_k \bigwedge_{1\leq i<j\leq k} y_i \neq y_j \wedge \hat{\varphi}$. Using the claim from Step 2, it is clear that for all structures $\mathcal{A}$ with at least k elements we have

$$\mathsf{copy}_k(\mathcal{A}) \models \varphi \text{ if and only if } \mathcal{A} \models \varphi^k.$$

This concludes the proof. □

2.5. Existence of Homomorphisms

Recall from the introduction that our decidability proof for ECTL^* with constraints over a structure $\mathcal{A}$ is based on the fact we can express in a suitable logic the existence of a homomorphism into $\mathcal{A}$. In this subsection we introduce some formal terminology related to homomorphisms.

For a subsignature $\tau \subseteq \sigma$, a τ-structure $\mathcal{B} = (B,J)$ and a σ-structure $\mathcal{A} = (A,I)$, a *homomorphism* from $\mathcal{B}$ to $\mathcal{A}$ is a mapping $h : B \to A$ such that for all $R \in \tau$ and all tuples $(b_1,\dots,b_{\mathsf{ar}(R)}) \in J(R)$ we have $(h(b_1),\dots,h(b_{\mathsf{ar}(R)})) \in I(R)$. We write $\mathcal{B} \preceq \mathcal{A}$ if there is a homomorphism from $\mathcal{B}$ to $\mathcal{A}$.

Definition 9. Let $\mathcal{L}$ be a logic (e.g. MSO or Bool(MSO, WMSO+B)). A σ-structure $\mathcal{A}$ has the property $\mathsf{EHD}(\mathcal{L})$ (existence of homomorphisms to $\mathcal{A}$ is $\mathcal{L}$-definable) if there is a computable function that maps every finite subsignature $\tau \subseteq \sigma$ to an $\mathcal{L}$-sentence φ_τ such that for every countable τ-structure $\mathcal{B}$ we have: $\mathcal{B} \preceq \mathcal{A}$ if and only if $\mathcal{B} \models \varphi_\tau$.

Example 10. , The structure $\mathcal{Q} = (\mathbb{Q},<,\equiv)$, where $\equiv$ is equality, has the property EHD(WMSO) (and EHD(MSO)). In [21] it is implicitly shown that for a countable $\{<,\equiv\}$-structure $\mathcal{B} = (B,J)$,[4] $\mathcal{B} \preceq \mathcal{Q}$ if and only if there does not exist $(a,b) \in J(<)$ such that $(b,a) \in (J(<) \cup J(\equiv) \cup J(\equiv)^{-1})^*$. This condition can be easily expressed in WMSO using the reach-construction from Example 2.

[4]A remark concerning the equality relation should be made at this point. In the structure $\mathcal{Q}$, we mean with $\equiv$ the equality relation, whereas in $\mathcal{B}$, the relation $J(\equiv)$ can be any binary relation. Nevertheless, in MSO we have a built-in equality, see the MSO-syntax from (2). This is one more reason, why we decided, to denote the equality relation as part of a structure with $\equiv$ instead of $\equiv$. In the structure $\mathcal{B}$ the MSO-formulas $x = y$ and $x \equiv y$ have, in general, different semantics.

3. ECTL^* with constraints

Extended computation tree logic (ECTL^*) is a branching time temporal logic first introduced in [26, 27] as an extension of CTL^*. As the latter, ECTL^* is interpreted on Kripke structures, but while CTL^* allows to specify LTL properties of infinite paths of such models, ECTL^* can describe regular (i.e., MSO-definable) properties of paths. In its original formulation, ECTL^* uses Büchiautomata to replace the classical CTL^* path formulas. In this work, instead of automata, we use MSO-formulas. Given the famous result of Büchithat MSO and Büchiautomata are equi-expressive on paths, we obtain an expressively equivalent logic. We use the formulation using MSO because it provides a simpler framework to add constraints. What we present below is an enhanced version of ECTL^*, which we call σ-constraint ECTL^*, or in short $\mathsf{ECTL}^*(\sigma)$. In $\mathsf{ECTL}^*(\sigma)$ the path-formulas come from $\mathsf{MSO}(\sigma)$ defined in (5). Let σ be a signature. We define $\mathsf{ECTL}^*(\sigma)$-formulas by the following grammar:

$$\varphi ::= \mathsf{E}\,\psi(\underbrace{\varphi,\dots,\varphi}_{\text{m times}}) \mid (\varphi \wedge \varphi) \mid \neg\varphi \tag{6}$$

where $\psi(X_1,\dots,X_m)$ is an $\mathsf{MSO}(\sigma)$-formula over the signature $\mathbb{P} \cup \{S\}$ in which only the set variables $X_1,\dots,X_m \in \mathbb{V}^1$ are allowed to occur freely.

As anticipated above, $\mathsf{ECTL}^*(\sigma)$-formulas are evaluated over some node of an $\mathcal{A}$-constraint graph. Let $\mathfrak{C}$ be an $\mathcal{A}$-constraint graph with underlying Kripke structure $\mathcal{K} = (D, \to, \rho)$. Given $d \in D$, for an $\mathsf{ECTL}^*(\sigma)$-formula φ, we define $(\mathfrak{C}, d) \models \varphi$ inductively by:

- $(\mathfrak{C}, d) \models \varphi_1 \wedge \varphi_2$ iff $(\mathfrak{C}, d) \models \varphi_1$ and $(\mathfrak{C}, d) \models \varphi_2$.
- $(\mathfrak{C}, d) \models \neg\varphi$ iff it is not the case that $(\mathfrak{C}, d) \models \varphi$.
- $(\mathfrak{C}, d) \models \mathsf{E}\,\psi(\varphi_1,\dots,\varphi_m)$ iff there is an infinite $\mathcal{K}$-path $P = d_0d_1d_2\cdots$ with $d_0 = d$, whose corresponding $\mathcal{A}$-constraint path $\mathfrak{P}$ satisfies $\mathfrak{P} \models_{\mathsf{MSO}(\sigma)} \psi(A_1,\dots,A_m)$ where $A_i = \{d_0\cdots d_n \mid n \geq 0, (\mathfrak{C}, d_n) \models \varphi_i\}$ for $1 \leq i \leq m$.

Note that for checking $(\mathfrak{C}, d) \models \varphi$ we may ignore all propositions $p \in \mathbb{P}$ and all function symbols $f \in \mathbb{F}$ that do not occur in φ.

Remark 11. The reader might miss atomic propositions $p \in \mathbb{P}$ in (6). They can be obtained using $\mathsf{MSO}(\sigma)$. More precisely, MSO can express the fact that a position x is the initial position of a path using the formula $\mathsf{pos}_0(x) = \forall y\,(x \neq S(y))$, then the $\mathsf{ECTL}^*(\sigma)$-formula $\mathsf{E}\,\exists x\,(\mathsf{pos}_0(x) \wedge p(x))$ states that in the current node there starts a path whose first node satisfies p, i.e., the current node satisfies p.

Looking back at the semantics for $\mathsf{MSO}(\sigma)$, see (5), note that the role of the concrete domain $\mathcal{A}$ and of the functions $f^{\mathfrak{C}}$ ($f \in \mathbb{F}$), for both $\mathsf{MSO}(\sigma)$ and $\mathsf{ECTL}^*(\sigma)$ are restricted to the semantics of atomic constraints. Ordinary ECTL^*-formulas ($\mathsf{ECTL}^*(\sigma)$-formulas without atomic constraints) are interpreted over a pair $(\mathcal{K}, d)$, where $\mathcal{K}$ is a Kripke structure, and the rules are the same as above (just ignoring the concrete domain and the function symbols from $\mathbb{F}$).

We define the usual abbreviations:

$$\begin{aligned}
\theta_1 \vee \theta_2 &:= \neg(\neg\theta_1 \wedge \neg\theta_2) \text{ (for both } \mathsf{ECTL}^* \text{ and } \mathsf{MSO} \text{ formulas)}\\
\theta_1 \to \theta_2 &:= \neg\theta_1 \vee \theta_2\\
\mathsf{A}\,\psi &:= \neg\mathsf{E}\,\neg\psi \text{ (universal path quantifier)}\\
\forall x\,\psi &:= \neg\exists x\,\neg\psi\\
\forall X\,\psi &:= \neg\exists X\,\neg\psi.
\end{aligned}$$

Note that $(\mathfrak{C}, d) \models \mathsf{A}\,\psi(\varphi_1,\dots,\varphi_m)$ if and only if for all infinite $\mathcal{K}$-paths $P = d_0d_1d_2\cdots$ with $d_0 = d$, we have for the corresponding constraint path $\mathfrak{P}$:

$$\mathfrak{P} \models_{\mathsf{MSO}(\sigma)} \psi(A_1,\dots,A_m) \text{ where } A_i = \{d_0d_1\cdots d_n \mid n \in \mathbb{N} \text{ and } (\mathfrak{C}, d_n) \models \varphi_i\} \text{ for } 1 \leq i \leq m.$$

Using this extended set of operators we can put every formula into a semantically equivalent *negation normal form*, where $\neg$ only occurs in front of atomic $\mathsf{MSO}(\sigma)$-formulas (i.e., formulas of the form $p(x)$, $x = S(y)$, $x \in X$ or atomic constraints). We additionally eliminate subformulas of the form $\neg(x \in X_i)$ where X_i is one of the set variables $X_1, \dots, X_m$ that occurs freely in φ as follows: Suppose $\theta = \mathsf{E}\,\psi(\varphi_1, \dots, \varphi_m)$ is a subformula of φ. Then we replace θ with the equivalent formula $\mathsf{E}\,\psi'(\varphi_1, \dots, \varphi_m, \neg\varphi_1, \dots, \neg\varphi_m)$, where $\psi'(X_1, \dots, X_m, Y_1, \dots, Y_m)$ is obtained from ψ by replacing all occurrences of $\neg(x \in X_i)$ by $x \in Y_i$ for $1 \le i \le m$.

We give in the following some examples of classical CTL^* expressible specifications formulated in ECTL^*. Recall that in monadic second-order logic the binary predicate $<$ can be derived from the successor function.

Example 12. Response to an impulse: In all computations, every occurrence of p is eventually followed by an occurrence of q.

$$\mathsf{CTL}^* : \mathsf{A\,G}\,(p \to \mathsf{F}\,q) \qquad \mathsf{ECTL}^* : \mathsf{A}\,[\forall x\,(p(x) \to \exists y\,(x < y \wedge q(y)))].$$

Absence of unsolicited responses: In all computations q does not occur unless preceded by p.

$$\mathsf{CTL}^* : \mathsf{A}\,(\mathsf{F}\,q \to (\neg q)\,\mathsf{U}\,p) \qquad \mathsf{ECTL}^* : \mathsf{A}\,[\forall x\,(q(x) \to \exists y\,(y \le x \wedge p(y)))].$$

Existence of a stabilizing computation: There is a computation where eventually p holds in every state.

$$\mathsf{CTL}^* : \mathsf{E\,F\,G}\,p \qquad \mathsf{ECTL}^* : \mathsf{E}\,[\exists x\,\forall y\,(x < y \to p(y))].$$

We illustrate in the following example that the nesting of path quantifiers in a CTL^*-formula results in the nesting of MSO-formulas inside the corresponding ECTL^*-formula.

Example 13. The CTL^*-formula $\mathsf{E\,G}\,(p \to \mathsf{A\,X}q)$ expresses the existence of a path P such that every successor of a p-labeled node on P is labeled with q. Let φ be the ECTL^*-formula stating that on all paths q holds in the next state: $\varphi = \mathsf{A}\,\exists x\ (\mathsf{pos}_0(x) \wedge q(S(x)))$, where we use pos_0 to denote the first position of a path (see Remark 11). Then the required property is expressed by the formula $\mathsf{E}\,\psi(\varphi)$, where $\psi(X) = \forall z\,(p(z) \to z \in X)$. All together we obtain the formula

$$\mathsf{E}\,\forall z\ \big(p(z) \to z \in \big[\mathsf{A}\,\exists x\ (\mathsf{pos}_0(x) \wedge q(S(x)))\,\big]\big)\,.$$

In the following example we exploit the higher expressive power of ECTL^* to express a system requirement which cannot be formulated in CTL^*.

Example 14. There is a computation path where p holds in all even positions. The following MSO-formula describes the set X of even positions of a path:

$$\mathsf{even}(X) \coloneqq \exists x\ (\mathsf{pos}_0(x) \wedge x \in X \wedge S(x) \notin X) \wedge \forall x\ (x \in X \leftrightarrow S(S(x)) \in X)\,.$$

The following ECTL^*-formula describes the required property:

$$\mathsf{E}\,[\exists X\,\mathsf{even}(X) \wedge \forall z\,(z \in X \to p(z))].$$

Wolper [29] proved that no CTL^*-formula expresses this property.

Example 15. We show that it is possible, using constraints over $(\mathbb{Z}, <)$, to write an $\mathsf{ECTL}^*(\{<\})$-formula which can only be satisfied by an infinite $(\mathbb{Z}, <)$-constraint graph (we use the infix notation for $<$):

$$\varphi = \mathsf{E}\,[\forall x\,f_1 x < f_1 S(x)]. \tag{7}$$

We are forcing the existence of a path $\mathfrak{P}$ on which $f_1^{\mathfrak{P}}(x)$ is strictly smaller than $f_1^{\mathfrak{P}}(y)$ whenever $x \to y$ in $\mathfrak{P}$. This ensures that the domain of $\mathfrak{P}$ is infinite.

We remark that the last example shows that $\mathsf{ECTL}^*(\sigma)$ is strictly more expressive than ECTL^* in the following sense: Let us denote with $L(\varphi)$ the set of all underlying Kripke structures of $(\mathbb{Z}, <)$-constraint graphs, which satisfy φ. Then $L(\varphi)$ for φ from (7) is not empty and it does not contain any finite Kripke structure. On the other hand it is well known that ECTL^* enjoys the finite model property, and therefore cannot define $L(\varphi)$.

4. Satisfiability of constraint ECTL^* over a concrete domain

Let us now introduce the central notion of satisfiability: We say that an $\mathsf{ECTL}^*(\sigma)$-formula φ is $\mathcal{A}$-*satisfiable* if there is an $\mathcal{A}$-constraint graph $\mathfrak{C}$ with underlying Kripke structure $\mathcal{K} = (D, \rightarrow, \rho)$ and a node $v \in D$ such that $(\mathfrak{C}, v) \models \varphi$. In the following we first show that every satisfiable $\mathsf{ECTL}^*(\sigma)$-formula always has a nice model, namely a tree-model, where the branching degree is bounded by a constant that can be computed from the formula. The proof of this property is analogous to the proof of the tree model property for ECTL^* or CTL^*. Readers that are familiar with one of them can safely skip Lemmas 16 and 17 as well as the proof of Theorem 18.

Lemma 16. *Let $\mathfrak{C} = (\mathcal{A}, \mathcal{K}, \gamma)$ be a constraint graph, d_0 a node of $\mathcal{K}$ and φ an $\mathsf{ECTL}^*(\sigma)$-formula. If $P = d_0 d_1 \cdots d_n$ is an element of $\mathsf{Unf}(\mathfrak{C}, d_0)$, then $(\mathfrak{C}, d_n) \models \varphi$ if and only if $(\mathsf{Unf}(\mathfrak{C}, d_0), P) \models \varphi$.*

Proof. The proof is an easy induction on the structure of the formula using the fact that any constraint path in $\mathfrak{C}$ starting at a node reachable from d_0 corresponds to a constraint path in $\mathsf{Unf}(\mathfrak{C}, d_0)$ and vice versa. □

For similar reasons, we can duplicate subtrees of a constraint tree $\mathfrak{T}$ without affecting the set of satisfied formulas which allows to increase the branching degree of the model arbitrarily.

Lemma 17. *Let $\mathfrak{T}$ be a constraint tree. There is a constraint tree $\mathfrak{T}_\omega$ such that*

- *every node of $\mathfrak{T}_\omega$ has infinitely many successors,*
- *$\mathfrak{T}$ and $\mathfrak{T}_\omega$ satisfy the same $\mathsf{ECTL}^*(\sigma)$-formulas at their roots, and*
- *if d is a node and $\varphi = \mathsf{E}\,\psi(\varphi_1, \ldots, \varphi_k)$ is a formula such that $(\mathfrak{T}_\omega, d) \models \varphi$, then there are infinitely many paths starting at d which witness the path quantifier, i.e., there are infinitely many paths $P = d_0 d_1 d_2 \cdots$ in $\mathfrak{T}_\omega$ with $d_0 = d$ and $\mathfrak{P} \models \psi(A_1, \ldots, A_k)$ for $A_i = \{d_0 \cdots d_n \mid n \geq 0, (\mathfrak{T}_\omega, d_n) \models \varphi_i\}$ $(1 \leq i \leq m)$.*

Proof. Let $\mathfrak{T} = (\mathcal{A}, \mathcal{T}, \gamma)$ be some constraint tree where $\mathcal{T} = (T, \rightarrow, \rho)$. Without loss of generality we assume that $T \subseteq D^*$ for some set D such that $\rightarrow$ is the extension of words over D by one letter. We define a Kripke tree $\mathcal{T}_\omega = (T_\omega, \rightarrow_\omega, \rho_\omega)$ and a constraint tree $\mathfrak{T}_\omega = (\mathcal{A}, \mathcal{T}_\omega, \gamma_\omega)$ where

- $T_\omega \subseteq (D \times \mathbb{N})^*$ such that $\bar{d} = (d_1, n_1)(d_2, n_2) \ldots (d_i, n_i) \in T_\omega$ if and only if $\pi_1(\bar{d}) \in T$, where π_1 denotes element-wise projection to the first component,
- $\rightarrow_\omega$ is extension by one element from $D \times \mathbb{N}$,
- $\rho_\omega = \rho \circ \pi_1$, and
- $\gamma_\omega(f) = \gamma(f) \circ \pi_1$ for all $f \in \mathbb{F}$.

Note that the mappings ρ_ω and $\gamma_\omega(f)$ apply the mappings ρ and $\gamma(f)$, respectively, to the path of first components of a path from T_ω.

Since $\mathfrak{T}_\omega$ is basically an infinite copy of $\mathfrak{T}$ everywhere, where projection to the first component translates between the two structures, rather simple inductions prove the following facts.

1. Let P_ω be an infinite path in $\mathfrak{T}_\omega$, $A_1, A_2, \ldots, A_k$ subsets of P_ω and $\varphi \in \mathsf{MSO}(\sigma)$. Let $\mathfrak{P}_\omega$ be the constraint path corresponding to P_ω and let $P = \pi_1(P_\omega)$ be the path in $\mathfrak{T}$ obtained by element-wise projection of P_ω to the first component. Finally, let $\mathfrak{P}$ be the constraint path corresponding to P. Then we have
$$\mathfrak{P}_\omega \models \varphi(A_1, \ldots, A_k) \iff \mathfrak{P} \models \varphi(\pi_1(A_1), \ldots, \pi_1(A_k)).$$
2. For $\bar{d} = (d_1, n_1)(d_2, n_2) \ldots (d_k, n_k) \in T_\omega$ and φ some $\mathsf{ECTL}^*(\sigma)$-formula we have
$$(\mathfrak{T}_\omega, \bar{d}) \models \varphi \iff (\mathfrak{T}, \pi_1(\bar{d})) \models \varphi.$$

The second part particularly implies that every path quantifier that is satisfied at some node in $\mathfrak{T}_\omega$ is witnessed by infinitely many paths in $\mathfrak{T}_\omega$ starting at this node. $\square$

From now on, $\#_{\mathsf{E}}(\varphi)$ denotes the number of different subformulas of the form $\mathsf{E}\,\psi$ in the $\mathsf{ECTL}^*(\sigma)$-formula φ. Then $\mathsf{ECTL}^*(\sigma)$ has the following tree model property:

Theorem 18. *Let φ be an $\mathsf{ECTL}^*(\sigma)$-formula in negation normal form and let $\mathcal{A} = (A, I)$ be a σ-structure. Then φ is $\mathcal{A}$-satisfiable if and only if there is an $\mathcal{A}$-*constraint $(\#_{\mathsf{E}}(\varphi)+1)$-tree *$\mathfrak{T}$ with root r and $(\mathfrak{T}, r) \models \varphi$.*

Proof. Let $e = \#_{\mathsf{E}}(\varphi) + 1$. Due to the previous lemmas, we can assume that $\mathfrak{C} = (\mathcal{A}, \mathcal{K}, \gamma)$ is an $\mathcal{A}$-constraint tree with $\mathcal{K} = (D, \rightarrow, \rho)$ a Kripke tree over $\mathbb{P}$ with root r where every node $d \in D$ has infinitely many successors such that for every formula $\mathsf{E}\,\psi$ with $(\mathfrak{C}, d) \models \mathsf{E}\,\psi$ there are infinitely many pairwise disjoint (except for node d) paths starting at d that witness this path quantifier. We prune $\mathfrak{C}$ such that it is isomorphic to an $\mathcal{A}$-constraint e-tree model of φ.

We inductively define the domain D' of our new tree. For the initial step, choose a constraint path $\mathfrak{P}$ of $\mathfrak{C}$ arbitrarily and add its domain to D'.

For the inductive step we repeat the following procedure until every node has e successors. Let $d \in D'$ be a node with less than e successors (in D'). Our inductive definition ensures that d then has exactly 1 successor. Let $\mathsf{E}\,\psi_1(\varphi_1^1, \ldots, \varphi_{m_1}^1), \ldots, \mathsf{E}\,\psi_k(\varphi_1^k, \ldots, \varphi_{m_k}^k)$ be the existential subformulas of φ which hold true in $(\mathfrak{C}, d)$. Then for each $1 \le j \le k$ there is a constraint path $\mathfrak{P}_j$ in $\mathfrak{C}$ with $\mathfrak{P}_j(0) = d$ and disjoint from $D' \setminus \{d\}$ such that $\mathfrak{P}_j \models_{\mathsf{MSO}(\sigma)} \psi(A_1^j, \ldots, A_{m_j}^j)$, where $A_i^j = \{\mathfrak{P}_j(n) \mid n \ge 0, (\mathfrak{C}, \mathfrak{P}_j(n)) \models \varphi_i^j\}$. For each $j \in \{k+1, k+2, \ldots, e-1\}$ choose further constraint paths $\mathfrak{P}_j$ of $\mathfrak{C}$ with $\mathfrak{P}_{\mathrm{j}}(0) = d$ that are disjoint from $D' \setminus \{d\}$ and the other paths (except for their origin d). Add the domains of $\mathfrak{P}_1, \mathfrak{P}_2, \ldots, \mathfrak{P}_{e-1}$ to D' and continue the construction with the next node of the resulting D' that has only one successor.

The limit of this process results in a subset $D' \subseteq D$ such that D' induces a Kripke e-subtree $\mathcal{T} = \mathcal{K}\restriction_{D'}$ and a constraint subtree $\mathfrak{T} = (\mathcal{A}, \mathcal{T}, \gamma')$, where $\gamma'(f) = \gamma(f)\restriction_{D'}$ for all $f \in \mathbb{F}$. We prove $(\mathfrak{T}, r) \models \varphi$ by showing the following stronger claim using structural induction on the formula φ.

(1) Given a subformula $\theta \in \mathsf{ECTL}^*(\sigma)$ of φ and $d \in D'$ such that $(\mathfrak{C}, d) \models \theta$, then $(\mathfrak{T}, d) \models \theta$.

(2) For all $\mathsf{MSO}(\sigma)$-formulas ψ that are subformulas of φ, all constraint paths $\mathfrak{P}$ in $\mathfrak{T}$ and all subsets $A_1, \ldots, A_m, B_1, \ldots, B_m$ of the domain of $\mathfrak{P}$ such that $A_j \subseteq B_j$ for all $1 \le j \le m$, and all valuation functions η,

$$(\mathfrak{P}, \eta[(X_j \to A_j)_{1 \le j \le m}]) \models_{\mathsf{MSO}(\sigma)} \psi \implies (\mathfrak{P}, \eta[(X_j \to B_j)_{1 \le j \le m}]) \models_{\mathsf{MSO}(\sigma)} \psi, \tag{8}$$

where we assume that X_i only occurs freely in ψ (we can rename bounded occurrences).

Recall that φ is in negation normal form. Hence, the proof only needs to consider the following cases.

- The cases $\theta = \varphi_1 \wedge \varphi_2$ and $\theta = \varphi_1 \vee \varphi_2$ in (1) are straightforward by induction.

- Let $\theta = \mathsf{E}\,\psi(\varphi_1, \ldots, \varphi_m)$ in (1). Note that only $X_1, \ldots, X_m$ are allowed to occur freely in ψ. Let $d \in D'$ be such that $(\mathfrak{C}, d) \models \mathsf{E}\,\psi(\varphi_1, \ldots, \varphi_m)$. By construction of $\mathfrak{T}$, there is a constraint path $\mathfrak{P}$ in $\mathfrak{T}$ (which is simultaneously in $\mathfrak{C}$) with $\mathfrak{P}(0) = d$ such that $\mathfrak{P} \models_{\mathsf{MSO}(\sigma)} \psi(A_1, \ldots, A_m)$, where $A_i = \{\mathfrak{P}(n) \mid n \ge 0, (\mathfrak{C}, \mathfrak{P}(n)) \models \varphi_i\}$. Let $B_i = \{\mathfrak{P}(n) \mid n \ge 0, (\mathfrak{T}, \mathfrak{P}(n)) \models \varphi_i\}$. By the inductive hypothesis for the φ_i (point (1)), we have $A_i \subseteq B_i$. Thus, using the inductive hypothesis (point (2)) for ψ, we obtain $\mathfrak{P} \models_{\mathsf{MSO}(\sigma)} \psi(B_1, \ldots, B_m)$. Hence, $(\mathfrak{T}, d) \models \mathsf{E}\,\psi(\varphi_1, \ldots, \varphi_m)$ as desired.

- Let $\theta = \mathsf{A}\,\psi(\varphi_1, \ldots, \varphi_m)$ in (1). In order to get $(\mathfrak{T}, d) \models \theta$, we need to show that every constraint path $\mathfrak{P}$ in $\mathfrak{T}$ that starts in d satisfies $\mathfrak{P} \models_{\mathsf{MSO}(\sigma)} \psi(B_1, \ldots, B_m)$, where $B_i = \{\mathfrak{P}(n) \mid n \ge 0, (\mathfrak{T}, \mathfrak{P}(n)) \models \varphi_i\}$. So let $\mathfrak{P}$ be a constraint path in $\mathfrak{T}$ (and hence in $\mathfrak{C}$ as well) with $\mathfrak{P}(0) = d$. Since $(\mathfrak{C}, d) \models \theta$, we can deduce that $\mathfrak{P} \models_{\mathsf{MSO}(\sigma)} \psi(A_1, \ldots, A_m)$, where $A_i = \{\mathfrak{P}(n) \mid n \ge 0, (\mathfrak{C}, \mathfrak{P}(n)) \models \varphi_i\}$. By inductive hypothesis for point (1), we conclude that $A_i \subseteq B_i$ and by the inductive hypothesis for point (2) we conclude that $\mathfrak{P} \models_{\mathsf{MSO}(\sigma)} \psi(B_1, \ldots, B_m)$. Hence, we get $(\mathfrak{T}, d) \models \theta$.

This completes the inductive step for point (1). We continue with the inductive step for point (2), i.e., for $\mathsf{MSO}(\sigma)$-subformulas ψ of φ. To simplify notation we write η_A for $\eta[(X_j \to A_j)_{1\le j\le m}]$ and η_B for $\eta[(X_j \to B_j)_{1\le j\le m}]$.

- If $\psi = p(x)$ ($\neg p(x)$, respectively) for some $p \in \mathbb{P}$, then by definition we have: $(\mathfrak{P}, \eta_A) \models_{\mathsf{MSO}(\sigma)} p(x)$ if and only if $p \in \rho(\eta_A(x))$ if and only if $p \in \rho(\eta_B(x))$ if and only if $(\mathfrak{P}, \eta_B) \models_{\mathsf{MSO}(\sigma)} p(x)$.
- Similarly, if ψ is of the form $x = S(y)$, $x \neq S(y)$, $x \in X$, or $x \notin X$ for $X \in \mathbb{V}^1 \setminus \{X_1, \ldots, X_m\}$, we have $(\mathfrak{P}, \eta_A) \models_{\mathsf{MSO}(\sigma)} \psi$ if and only if $(\mathfrak{P}, \eta_B) \models_{\mathsf{MSO}(\sigma)} \psi$ because ψ does not depend on the interpretations of $X_1, \ldots, X_m$.
- Let $\psi = (x \in X_i)$ for some $1 \le i \le m$. Then $(\mathfrak{P}, \eta_A) \models_{\mathsf{MSO}(\sigma)} x \in X_i$ implies $\eta_A(x) \in A_i$ and whence (using $A_i \subseteq B_i$) $\eta_B(x) \in B_i$, i.e., $(\mathfrak{P}, \eta_B) \models_{\mathsf{MSO}(\sigma)} x \in X_i$.
- The cases $\psi = \psi_1 \vee \psi_2$ and $\psi = \psi_1 \wedge \psi_2$ follow from inductive hypothesis.
- Assume that $\psi = \exists x\, \psi_1$ and that $(\mathfrak{P}, \eta_A) \models \psi$. Then there is a d in the domain of $\mathfrak{P}$ such that $(\mathfrak{P}, \eta_A[x \mapsto a]) \models_{\mathsf{MSO}(\sigma)} \psi_1$. By the inductive hypothesis this implies $(\mathfrak{P}, \eta_B[x \mapsto a]) \models_{\mathsf{MSO}(\sigma)} \psi_1$ and therefore $(\mathfrak{P}, \eta_B) \models \exists x\, \psi_1$.
- The cases $\psi = \forall x\, \psi_1$, $\psi = \exists X\, \psi_1$ and $\psi = \forall X\, \psi_1$ (note that X must be different from $X_1, \ldots, X_m$ since X only occurs freely in ψ) are proven analogously to the previous case.
- Let $\psi = R(f_1 S^{i_1}(x), \ldots, f_k S^{i_k}(x))$ or its negation. Since η_A and η_B agree on x, this case is trivial.

This concludes the proof of the theorem. $\square$

Given a concrete domain $\mathcal{A} = (A, I)$, with $\mathsf{SAT}_{\mathsf{ECTL}^*}(\mathcal{A})$ we denote the following computational problem: *Is a given formula* $\varphi \in \mathsf{ECTL}^*(\sigma)$ $\mathcal{A}$*-satisfiable?* The main result of this section gives a criterion on the concrete domain $\mathcal{A}$ that implies decidability of the problem $\mathsf{SAT}_{\mathsf{ECTL}^*}(\mathcal{A})$. To state this criterion, we need one further technical condition:

Definition 19. The σ-structure $\mathcal{A} = (A, I)$ is *negation-closed* if the complement of every relation $I(R)$ is effectively definable by a positive existential first-order formula. Formally, $\mathcal{A}$ is negation-closed, if there is a computable function that maps each relation symbol $R \in \sigma$ to a positive existential first-order formula $\varphi_R(x_1, \ldots, x_{\mathsf{ar}(R)})$ (i.e., a formula that is built up from atomic formulas using $\wedge$, $\vee$, and $\exists$) such that

$$A^{\mathsf{ar}(R)} \setminus I(R) = \{(a_1, \ldots, a_{\mathsf{ar}(R)}) \mid \mathcal{A} \models \varphi_R(a_1, \ldots, a_{\mathsf{ar}(R)})\}.$$

Example 20. The structure $\mathcal{Z} = (\mathbb{Z}, <, \equiv, (\equiv_a)_{a\in\mathbb{Z}}, (\equiv_{a,b})_{0\le a<b})$ from (1) is negation-closed (we write $x = a$ instead of $\equiv_a(x)$ and similarly for $\equiv_{a,b}$). We have for instance:

- $x \neq y$ if and only if $x < y \vee y < x$.
- $x \neq a$ if and only if $\exists y \in \mathbb{Z}(y = a \wedge (x < y \vee y < x))$.
- $x \not\equiv a \bmod b$ if and only if $x \equiv c \bmod b$ for some $0 \le c < b$ with $a \neq c$, i.e., $\bigvee_{\substack{0\le c<b \\ a\neq c}} x \equiv c \bmod b$.

Now we can state the main result of this section:

Theorem 21. *Let $\mathcal{A}$ be a σ-structure that has property* $\mathsf{EHD}(\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B}))$ *(see Definition 9) and is negation-closed. Then the problem* $\mathsf{SAT}_{\mathsf{ECTL}^*}(\mathcal{A})$ *is decidable.*

We say that an $\mathsf{ECTL}^*(\sigma)$-formula φ is in *strong negation normal form* if it is in negation normal form and there is no subformula $\neg\theta$ where θ is an atomic constraint.

Let us fix an $\mathsf{ECTL}^*(\sigma)$-formula φ in negation normal form and a negation-closed σ-structure $\mathcal{A}$ for the rest of this section. We want to check whether φ is $\mathcal{A}$-satisfiable. First, we reduce our problem to formulas in strong negation normal form:

Lemma 22. *Let $\mathcal{A} = (A, I)$ be a negation-closed σ-structure with universe A. From a given $\mathsf{ECTL}^*(\sigma)$-formula φ one can compute an $\mathsf{ECTL}^*(\sigma)$-formula $\hat{\varphi}$ in strong negation normal form such that φ is $\mathcal{A}$-satisfiable if and only if $\hat{\varphi}$ is $\mathcal{A}$-satisfiable.*

Proof. We can assume that φ is in negation normal form. Using induction, it suffices to eliminate a single negated atomic constraint $\theta = \neg R(f_1 S^{i_1}(x), \ldots, f_k S^{i_k}(x))$ in φ, where $k = \mathsf{ar}(R)$. Let $d = \max\{i_1, \ldots, i_k\}$ be the depth of the constraint θ. Since $\mathcal{A}$ is negation-closed, we can compute a positive quantifier-free first-order formula $\psi(y_1, y_2, \ldots, y_k, z_1, z_2, \ldots, z_m)$ such that $\mathcal{A} \models \neg R(a_1, \ldots, a_k)$ if and only if $\mathcal{A} \models \exists z_1 \cdots \exists z_m\, \psi(a_1, \ldots, a_k, z_1, \ldots, z_m)$. Let $g_1, \ldots, g_m \in \mathbb{F}$ be fresh function symbols not occurring in φ. We define the $\mathsf{ECTL}^*(\sigma)$-formula $\hat{\varphi}$ by replacing in φ every occurrence of the negated constraint θ by the formula

$$\psi(f_1 S^{i_1}(x), \ldots, f_k S^{i_k}(x), g_1 S^d(x), \ldots, g_m S^d(x)),$$

i.e., we replace in the positive quantifier-free formula $\psi(y_1, y_2, \ldots, y_k, z_1, z_2, \ldots, z_m)$ every occurrence of a variable y_j (respectively, z_j) by $f_j S^{i_j}(x)$ (respectively, $g_j S^d(x)$).

We have to prove that

$$\hat{\varphi} \text{ is } \mathcal{A}\text{-satisfiable} \iff \varphi \text{ is } \mathcal{A}\text{-satisfiable}.$$

Proof of $\Longrightarrow$. If φ is $\mathcal{A}$-satisfiable, then by Theorem 18 there is an $\mathcal{A}$-constraint e-tree $\mathfrak{T}$ with $(\mathfrak{T}, \varepsilon) \models \varphi$ and underlying Kripke e-tree $\mathcal{T} = ([1, e]^*, \to, \rho)$. We modify $\mathfrak{T}$ and obtain a new constraint tree $\mathfrak{S}$ by replacing the interpretations of the fresh function symbols $g_1, \ldots, g_m$ as follows. Consider $w, v \in [1, e]^*$ such that $|v| = d$. Let v_p be the prefix of v of length i_p for $1 \leq p \leq k$.

- If $(f_1^{\mathfrak{T}}(wv_1), \ldots, f_k^{\mathfrak{T}}(wv_k)) \notin I(R)$, we can fix values $a_1, \ldots, a_m \in A$ such that

  $$\mathcal{A} \models \psi(f_1^{\mathfrak{T}}(wv_1), \ldots, f_k^{\mathfrak{T}}(wv_k), a_1, \ldots, a_m).$$

  In this case we set $g_q(wv) = a_q$ for all $1 \leq q \leq m$.
- If $(f_1^{\mathfrak{T}}(wv_1), \ldots, f_k^{\mathfrak{T}}(wv_k)) \in I(R)$, we choose $g_q(wv) \in A$ arbitrarily for all $1 \leq q \leq m$.
- Finally, for all $w \in [1, e]^*$ such that $|w| < d$ we choose $g_q(w) \in A$ arbitrarily for all $1 \leq q \leq m$.

By induction on the structure of φ we prove that $(\mathfrak{S}, \varepsilon) \models \hat{\varphi}$. All steps are trivial except for the case that the subformula is precisely $\theta = \neg R(f_1 S^{i_1}(x), \ldots, f_k S^{i_k}(x))$. In this case let $\mathcal{P}$ be a Kripke path in $\mathcal{T}$ inducing the constraint path $\mathfrak{P}$ in $\mathfrak{T}$ and the constraint path $\mathfrak{N}$ in $\mathfrak{S}$ and let η be a valuation function such that $(\mathfrak{P}, \eta) \models_{\mathsf{MSO}(\sigma)} \theta$. Thus, setting $a = \eta(x)$, we get $(f_1^{\mathfrak{P}}(S^{i_1}(a)), \ldots, f_k^{\mathfrak{P}}(S^{i_k}(a))) \notin I(R)$. According to our definition of g_q, we have set $g_q(S^d(a)) = a_q$ for all $1 \leq q \leq m$, where $a_1, \ldots, a_m \in A$ such that

$$\mathcal{A} \models \psi(f_1^{\mathfrak{P}}(S^{i_1}(a)), \ldots, f_k^{\mathfrak{P}}(S^{i_1}(a)), a_1, \ldots, a_k).$$

Thus, it follows directly that

$$(\mathfrak{N}, \eta) \models_{\mathsf{MSO}(\sigma)} \psi(f_1 S^{i_1}(x), \ldots, f_k S^{i_k}(x), g_1 S^d(x), \ldots, g_m S^d(x)),$$

which concludes the first direction.

Proof of $\Longleftarrow$. In order to prove that φ is $\mathcal{A}$-satisfiable if $\hat{\varphi}$ is $\mathcal{A}$-satisfiable, let us assume that $\mathfrak{C}$ is an $\mathcal{A}$-constraint graph such that $(\mathfrak{C}, d) \models \hat{\varphi}$ for some node d. In order to show $(\mathfrak{C}, d) \models \varphi$ by induction on the structure of φ, we end up (after several trivial steps) with the following claim: For every constraint path $\mathfrak{P}$ and valuation function η,

$$(\mathfrak{P}, \eta) \models_{\mathsf{MSO}(\sigma)} \psi(f_1 S^{i_1}(x), \ldots, f_k S^{i_k}(x), g_1 S^d(x), \ldots, g_m S^d(x)) \tag{9}$$

implies $(\mathfrak{P}, \eta) \models_{\mathsf{MSO}(\sigma)} \theta$. Assuming (9), there are values, namely $g_i^{\mathfrak{T}}(S^d(\eta(x)))$ $(1 \leq i \leq m)$ witnessing

$$\mathcal{A} \models \exists z_1 \cdots \exists z_m\, \psi(a_1, \ldots, a_k, z_1, \ldots, z_m),$$

where $a_j = f_j^{\mathfrak{T}}(S^{i_j}(\eta(x)))$. By choice of ψ this implies that $\mathcal{A} \models \neg R(a_1, \ldots, a_k)$. Hence, we have $(\mathfrak{P}, \eta) \models_{\mathsf{MSO}(\sigma)} \neg R(f_1 S^{i_1}(x), \ldots, f_k S^{i_k}(x))$, i.e., $(\mathfrak{P}, \eta) \models_{\mathsf{MSO}(\sigma)} \theta$. $\square$

Example 23. Let us consider $\mathcal{Z} = (\mathbb{Z}, <, \equiv, (\equiv_a)_{a\in\mathbb{Z}}, (\equiv_{a,b})_{0\leq a<b})$ from (1). As we saw in Example 20, this is a negation-closed structure over the infinite signature $\sigma = \{<, \equiv\} \cup \{\equiv_a | \ a \in \mathbb{Z}\} \cup \{\equiv_{a,b} | \ 0 \leq a < b\}$. Let $\varphi = \mathsf{E}\,(\forall x\, \neg(f(x) = 3))$ be the $\mathsf{ECTL}^*(\sigma)$-formula expressing the fact that there exists a path on which f never assumes value 3 (we write $f(x) = 3$ instead of $\equiv_3 (fx)$). The strong negation normal form of φ is

$$\hat{\varphi} = \mathsf{E}\left[\forall x \left(g(x) = 3 \wedge \left(f(x) < g(x) \vee g(x) < f(x)\right)\right)\right].$$

Before we start with technical details, let us briefly sketch how we relate satisfiability for formulas in strong negation normal form with the property $\mathsf{EHD}(\mathcal{L})$ where $\mathcal{L}$ is some logic like $\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B})$ that has the properties mentioned in the introduction. The leading idea for solving satisfiability for $\mathsf{ECTL}^*(\sigma)$ formulas is to split the search for a model into two steps.

The first step is to describe all the Kripke trees that satisfy the structural requirements of a given $\mathsf{ECTL}^*(\sigma)$ formula φ. With structural requirements we mean, roughly speaking, all those parts of φ that can also be expressed in pure ECTL^*. The second step is to find interpretations of the functions $f \in \mathbb{F}$ such that also the constraints from φ are satisfied by the resulting constraint graph.

In order to accomplish the first step, we define from φ a pure ECTL^*-formula φ^a which we call the abstraction of φ. This formula φ^a results from φ by replacing each atomic constraint with a fresh atomic proposition not occurring in φ so far. Every Kripke tree $\mathcal{T}$ that satisfies φ^a then satisfies all the structural requirements of φ and is marked by new propositions at those positions where a model of φ would have to satisfy certain requirements with respect to the values assigned by the functions from $\mathbb{F}$.

For the second step we can use the new propositions and extract from every tree model $\mathcal{T}$ of φ^a a σ-structure $\mathcal{B}$ such that $\mathcal{B}$ encodes all the constraints imposed by φ in the sense that we can equip $\mathcal{T}$ with interpretations of the functions from $\mathbb{F}$ such that the resulting $\mathcal{A}$-constraint tree satisfies φ if and only if there is a homomorphism from $\mathcal{B}$ to $\mathcal{A}$.

If $\mathcal{A}$ has property $\mathsf{EHD}(\mathcal{L})$ for some logic $\mathcal{L}$ satisfying the requirements mentioned in the introduction, we can compile our two steps into the question whether a certain $\mathcal{L}$-formula has a tree model. This $\mathcal{L}$-formula requires that all its models encode the Kripke tree $\mathcal{T}$ satisfying φ^a as well as the the corresponding σ-structure $\mathcal{B}$ allowing a homomorphism to $\mathcal{A}$.

For the following definitions let us fix an $\mathsf{ECTL}^*(\sigma)$-formula φ in which only the atomic constraints $\theta_1, \dots, \theta_n$ occur. Let d_i be the depth of θ_i. Moreover, let $\mathbb{F}_\varphi$ be the set of function symbols from $\mathbb{F}$ appearing in φ.

Definition 24. We define φ^a as the ordinary ECTL^*-formula that is obtained from φ by replacing every occurrence of a constraint θ_i by the $\mathsf{MSO}(\sigma)$-formula $p_i(S^{d_i}(x))$. The same definition is also used for an $\mathsf{MSO}(\sigma)$-subformula of φ.

Example 25. Given the $\mathsf{ECTL}^*(\{<, \equiv\})$-formula

$$\varphi = \mathsf{E}\,[\forall x\, (q_1(x) \to (f_1(x) < f_2(S^2(x))))] \wedge \mathsf{A}\,[\exists x\, (q_2(x) \wedge f_1(S(x)) = f_2(x))]$$

we replace the atomic constraints with the propositional variables p_1 and p_2 to obtain the abstracted ECTL^*-formula

$$\varphi^a = \mathsf{E}\,[\forall x\, (q_1(x) \to p_1(S^2(x)))] \wedge \mathsf{A}\,[\exists x\, (q_2(x) \wedge p_2(S(x)))].$$

Definition 26. Given an constraint e-tree $\mathfrak{T}$ with underlying Kripke e-tree $\mathcal{T} = ([1, e]^*, \to, \rho)$ and $\rho(u) \cap \{p_1, \dots, p_n\} = \emptyset$ for all $u \in [1, e]^*$, we define the Kripke e-tree $\mathfrak{T}^a = ([1, e]^*, \to, \rho^a)$, where $\rho^a(u)$ contains

- all propositions from $\rho(u)$ and
- all propositions p_j $(1 \leq j \leq n)$ such that, assuming $\theta_j = R(f_1 S^{i_1}(x), \dots, f_k S^{i_k}(x))$ and $d_j = \max\{i_1, \dots, i_k\}$, we have:
  - $u = wv$ with $|v| = d_j$, and
  - $(f_1^{\mathfrak{T}}(wv_1), \dots, f_k^{\mathfrak{T}}(wv_k)) \in I(R)$, where v_l denotes the prefix of v of length i_l.

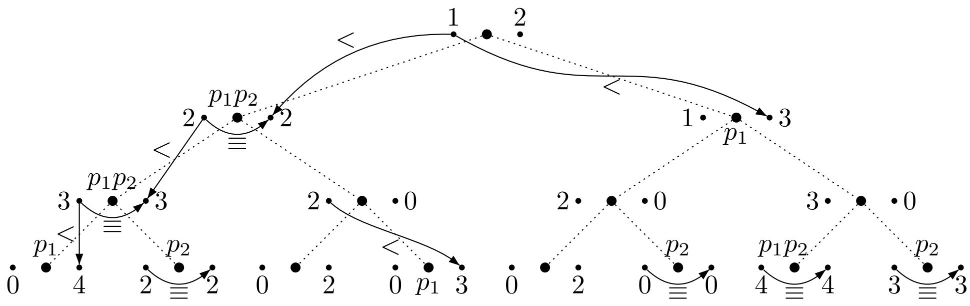

Figure 1: The $(\mathbb{N}, <, \equiv)$-constraint 2-tree $\mathfrak{T}$ from Example 29, the Kripke 2-tree $\mathfrak{T}^a$, and the structure $\mathcal{G}_{\mathfrak{T}^a}$.

Hence, the fact that proposition p_j labels node wv with $|v| = d_j$ means that the constraint θ_j holds along every path that starts in node w and descends in the tree down via node wv.

Definition 27. Given a Kripke e-tree $\mathcal{T} = ([1,e]^*, \rightarrow, \rho)$ (where the new propositions $p_1, \ldots, p_n$ are allowed to occur in $\mathcal{T}$) we define a countable σ-structure $\mathcal{G}_\mathcal{T} = ([1,e]^* \times \mathbb{F}_\varphi, J)$ as follows: The interpretation $J(R)$ of the relation symbol $R \in \sigma$ contains all k-tuples (where $k = \mathsf{ar}(r)$) $((wv_1, f_1), \ldots, (wv_k, f_k))$ for which there are $1 \le j \le n$ and $v \in [1,e]^{d_j}$ such that $p_j \in \rho(wv)$, and $\theta_j = R(f_1 S^{i_1}(x), \ldots, f_k S^{i_k}(x))$, where v_l still denotes the prefix of v of length i_l.

Remark 28. Setting $k = |\mathbb{F}_\varphi|$ one can easily write down a one-dimensional first-order interpretations which interprets $\mathcal{G}_\mathcal{T}$ in $\mathsf{copy}_k(\mathcal{T})$.

Example 29. Figure 1 shows an example, where we assume that $e = 2$ and $n = 2$, $\theta_1 = [f_1 x < f_2 S(x)]$, and $\theta_2 = [f_1 S(x) = f_2 S(x)]$. The figure shows a portion of an $(\mathbb{N}, <, \equiv)$-constraint 2-tree $\mathfrak{T} = ((\mathbb{N}, <, \equiv), \mathcal{T}, \mathbb{F})$. The edges of the Kripke 2-tree $\mathcal{T}$ are dotted. We assume that $\mathcal{T}$ is defined over the empty set of propositions. The node to the left (respectively, right) of a tree node w is labeled by the value $f_1^\mathfrak{T}(w)$ (respectively, $f_2^\mathfrak{T}(w)$). The figure shows the labeling of tree nodes with the two new propositions p_1 and p_2 (corresponding to θ_1 and θ_2) as well as the $\{<, \equiv\}$-structure $\mathcal{G}_{\mathfrak{T}^a}$.

Lemma 30. *Let φ be an $\mathsf{ECTL}^*(\sigma)$-formula in strong negation normal form. The formula φ is $\mathcal{A}$-satisfiable if and only if there is a Kripke $(\#_\mathsf{E}(\varphi) + 1)$-tree $\mathcal{T}$ such that $(\mathcal{T}, \varepsilon) \models \varphi^a$ and $\mathcal{G}_\mathcal{T} \preceq \mathcal{A}$.*

Proof. Let $\mathcal{A} = (A, I)$, $e = (\#_\mathsf{E}(\varphi) + 1)$, and let $\mathbb{F}_\varphi$, n, θ_j, and d_j $(1 \le j \le n)$ be defined as above.

Proof of $\Rightarrow$. Assume that φ is $\mathcal{A}$-satisfiable. By Theorem 18 there is a constraint e-tree $\mathfrak{T} = (\mathcal{A}, \mathcal{T}, \gamma)$ with $\mathcal{T} = ([1,e]^*, \rightarrow, \rho)$ such that $(\mathfrak{T}, \varepsilon) \models \varphi$. Take the Kripke e-tree $\mathfrak{T}^a = ([1,e]^*, \rightarrow, \rho^a)$.

We claim that $h : [1,e]^* \times \mathbb{F}_\varphi \rightarrow A$ given by $h(w,f) = f^\mathfrak{T}(w)$ is a homomorphism from $\mathcal{G}_{\mathfrak{T}^a}$ to $\mathcal{A}$. For this, assume that the tuple $((wv_1, f_1), \ldots, (wv_k, f_k))$ belongs to the interpretation of R in $\mathcal{G}_{\mathfrak{T}^a}$. By Definition 27 there are $1 \le j \le n$ and $v \in [1,e]^{d_j}$ such that $p_j \in \rho^a(wv)$, $\theta_j = R(f_1 S^{i_1}(x), \ldots, f_k S^{i_k}(x))$, and v_l is the prefix of v of length j_l for each $1 \le l \le k$. Since $p_j \in \rho^a(wv)$, Definition 26 implies

$$(h(wv_1, f_1), \ldots, h(wv_k, f_k)) = (f_1^\mathfrak{T}(wv_1), \ldots, f_k^\mathfrak{T}(wv_k)) \in I(R).$$

Hence, h is indeed a homomorphism.

In order to show $(\mathfrak{T}^a, \varepsilon) \models \varphi^a$ we prove simultaneously by structural induction on the formula that

(1) For all $\mathsf{ECTL}^*(\sigma)$-subformulas χ of φ and $v \in [1,e]^*$, if $(\mathfrak{T}, v) \models \chi$, then $(\mathfrak{T}^a, v) \models \chi^a$, and
(2) for all $\mathsf{MSO}(\sigma)$-subformulas ψ of φ, all paths P in $\mathcal{T}$, all valuation functions η, and all subsets $A_1, \ldots, A_k$, $B_1, \ldots, B_k \subseteq P$ such that $A_i \subseteq B_i$ for all $1 \le i \le k$,

$$\text{if } (\mathfrak{P}, \eta[(X_i \rightarrow A_i)_{1 \le i \le k}]) \models_{\mathsf{MSO}(\sigma)} \psi \text{ then } (\mathfrak{P}^a, \eta[(X_i \rightarrow B_i)_{1 \le i \le k}]) \models_{\mathsf{MSO}(\sigma)} \psi^a,$$

where $\mathfrak{P}$ is the constraint path induced by P in $\mathfrak{T}$, $\mathfrak{P}^a$ is the Kripke path induced by P in $\mathfrak{T}^a$, and we assume that $X_1, \ldots, X_k$ only appear freely in φ.

We have to consider the following cases, where we write η_A and η_B for the valuations $\eta[(X_j \to A_j)_{1 \leq j \leq m]}]$ and $\eta[(X_j \to B_j)_{1 \leq j \leq m]}]$, respectively.

- Assume that $\chi = \mathsf{E}\,\psi(\varphi_1, \ldots, \varphi_m)$, whence $\chi^a = \mathsf{E}\,\psi^a(\varphi_1^a, \ldots, \varphi_m^a)$. Since $(\mathfrak{T}, v) \models \theta$, we know that there is an infinite constraint path $\mathfrak{P}$ with $\mathfrak{P}(0) = v$ such that $\mathfrak{P} \models \psi(A_1, \ldots, A_m)$, where $A_i = \{\mathfrak{P}(n) \mid n \geq 0, (\mathfrak{T}, \mathfrak{P}(n)) \models \varphi_i\}$. We can use the induction hypothesis (point (1)) on $\varphi_1, \ldots, \varphi_m$ to obtain that, for all $n \in \mathbb{N}$ and $1 \leq i \leq m$, if $(\mathfrak{T}, \mathfrak{P}(n)) \models \varphi_i$ then $(\mathfrak{T}^a, \mathfrak{P}(n)) \models \varphi_i^a$. So, if we define $B_i = \{\mathfrak{P}(n) \mid n \geq 0, (\mathfrak{T}^a, \mathfrak{P}(n)) \models \varphi_i^a\}$, we can deduce that $A_i \subseteq B_i$. Applying point (2) of the induction hypothesis we conclude that $\mathfrak{P}^a \models \psi^a(B_1, \ldots, B_m)$, where $\mathfrak{P}^a$ denotes the Kripke path in $\mathfrak{T}^a$ that corresponds to $\mathfrak{P}$.
- The case $\chi = \mathsf{A}\,\psi(\varphi_1, \ldots, \varphi_m)$ is treated analogously to the previous one replacing "there is" by "for all".
- Assume that $\psi = \theta_j$ for some atomic constraint $\theta_j = R(f_1 S^{i_1}(x), \ldots, f_k S^{i_k}(x))$ of depth $d_j = \max\{i_1, \ldots, i_k\}$. In this case we want to show that $(\mathfrak{P}, \eta_A) \models_{\mathsf{MSO}(\sigma)} \theta_j$ implies $(\mathfrak{P}^a, \eta_B) \models_{\mathsf{MSO}(\sigma)} p_j(S^{d_j}(x))$. Let $a = \eta(x)$ and $a' = S^{d_j}(a)$. Note that $\eta_A(x) = \eta_B(x) = a$. If $(\mathfrak{P}, \eta_A) \models_{\mathsf{MSO}(\sigma)} R(f_1(S^{i_1}(x)), \ldots, f_k(S^{i_k}(x)))$, then $(f_1^{\mathfrak{P}} S^{i_1}(a), \ldots, f_k^{\mathfrak{P}} S^{i_k}(a)) \in I(R)$, and this, by Definition 26 implies that $p_j \in \rho^a(a')$ which implies $(\mathfrak{P}^a, \eta_B) \models_{\mathsf{MSO}(\sigma)} p_j(S^{d_j}(x))$.
- All other steps are trivial.

Proof of $\Leftarrow$. For the other direction, assume that there are a Kripke e-tree $\mathcal{T} = ([1, e]^*, \to, \rho_{\mathcal{T}})$ such that $(\mathcal{T}, \varepsilon) \models \varphi^a$, and a homomorphism $h : [1, e]^* \times \mathbb{F}_\varphi \to A$ from $\mathcal{G}_{\mathcal{T}}$ to $\mathcal{A} = (A, I)$. Define the $\mathcal{A}$-constraint graph $\mathfrak{T} = (\mathcal{A}, \mathcal{T}', \gamma)$, where

- $\mathcal{T}'$ is obtained from $\mathcal{T}$ by removing the propositions corresponding to atomic constraints, i.e., $\mathcal{T}' = ([1, e]^*, \to, \rho)$ with $\rho(v) = \rho_{\mathcal{T}}(v) \setminus \{p_1, \ldots, p_n\}$ for all $v \in [1, e]^*$,
- $f^{\mathfrak{T}}(v) = h(v, f)$ for all $f \in \mathbb{F}_\varphi$ and $v \in [1, e]^*$, and
- $f^{\mathfrak{T}}$ is an arbitrary function for all $f \in \mathbb{F} \setminus \mathbb{F}_\varphi$.

We claim that $(\mathfrak{T}, \varepsilon) \models \varphi$. Again by structural induction, we prove the following claim.

1. For all $\mathsf{ECTL}^*(\sigma)$-subformulas χ of φ and $v \in [1, e]^*$, if $(\mathcal{T}, v) \models \chi^a$, then $(\mathfrak{T}, v) \models \chi$, and
2. for all $\mathsf{MSO}(\sigma)$-subformulas ψ of φ, all paths P in $\mathcal{T}$, all valuation functions η, and all subsets $A_1, \ldots, A_k$, $B_1, \ldots, B_k \subseteq P$ such that $B_i \subseteq A_i$ for all $1 \leq i \leq k$,

$$(\mathcal{P}, \eta[(X_i \to B_i)_{1 \leq i \leq k}]) \models_{\mathsf{MSO}(\sigma)} \psi^a \text{ then } (\mathfrak{P}, \eta[(X_i \to A_i)_{1 \leq i \leq k}]) \models_{\mathsf{MSO}(\sigma)} \psi,$$

where $\mathcal{P}$ is the Kripke path induced by P in $\mathcal{T}$, $\mathfrak{P}$ is the constraint path induced by P in $\mathfrak{T}$, and we assume that $X_1, \ldots, X_k$ only appear freely in φ.

All steps are trivial except for the case that χ is the constraint $\theta_j = R(f_1 S^{i_1}(x), \ldots, f_k S^{i_k}(x))$ of depth d_j. In this case, we have $\chi^a = p_j(S^{d_j}(x))$. Assume that $(\mathcal{P}, \eta[(X_i \to B_i)_{1 \leq i \leq k}]) \models_{\mathsf{MSO}(\sigma)} p_j(S^{d_j}(x))$. By definition of $\mathcal{G}_{\mathcal{T}}$ (Definition 27), this implies that the tuple $((S^{i_1}(\eta(x)), f_1), \ldots, (S^{i_k}(\eta(x)), f_k))$ belongs to the interpretation of R in $\mathcal{G}_{\mathcal{T}}$. Now, since h is a homomorphism we conclude that

$$(h(S^{i_1}(\eta(x)), f_1), \ldots, h(S^{i_k}(\eta(x)), f_k)) = (f_1^{\mathfrak{T}}(S^{i_1}(\eta(x))), \ldots, f_k^{\mathfrak{T}}(S^{i_k}(\eta(x)))) \in I(R),$$

and thus

$$(\mathfrak{P}, \eta[(X_i \to A_i)_{1 \leq i \leq k}]) \models_{\mathsf{MSO}(\sigma)} R(f_1 S^{i_1}(x), \ldots, f_k S^{i_k}(x))$$

as desired. □

Let $\theta = \varphi^a$ for the further discussion. Hence, θ is an ordinary ECTL^*-formula, where negations only occur in front of propositions from $\mathbb{P} \setminus \{p_1, \ldots, p_n\}$ or atomic MSO-formulas, and $e = \#_{\mathsf{E}}(\theta) + 1$. By Lemma 30, we have to check, whether there is a Kripke e-tree $\mathcal{T}$ such that

$$(\mathcal{T}, \varepsilon) \models \theta \text{ and } \mathcal{G}_{\mathcal{T}} \preceq \mathcal{A}.$$

Let $\tau \subseteq \sigma$ be the finite subsignature consisting of all predicate symbols that occur in our initial $\mathsf{ECTL}^*(\sigma)$-formula φ. Note that $\mathcal{G}_{\mathcal{T}}$ is actually a τ-structure. Since the concrete domain $\mathcal{A}$ has the property $\mathsf{EHD}(\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B}))$, one can compute from τ a $\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B})$-sentence α such that for every countable τ-structure $\mathcal{B}$ we have $\mathcal{B} \models \alpha$ if and only if $\mathcal{B} \preceq \mathcal{A}$. Since $\mathcal{G}_{\mathcal{T}}$ is countable, our new goal is to decide whether there is a Kripke e-tree $\mathcal{T}$ such that

$$(\mathcal{T}, \varepsilon) \models \theta \text{ and } \mathcal{G}_{\mathcal{T}} \models \alpha$$

Given the fact that every ECTL^*-formula can be effectively transformed into an equivalent MSO-formula with a single free first-order variable [11, 17], and since the root ε of a tree is first-order definable, we get an MSO-sentence ψ such that $(\mathcal{T}, \varepsilon) \models \theta$ if and only if $\mathcal{T} \models \psi$. Hence, we have to check whether there is a Kripke e-tree $\mathcal{T}$ such that

$$\mathcal{T} \models \psi \text{ and } \mathcal{G}_{\mathcal{T}} \models \alpha.$$

Since $\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B})$ is compatible with first-order interpretations (which is trivial) and with the k-copy operation (Proposition 8), we can compute from α a $\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B})$-sentence α' and from α' another $\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B})$-sentence β such that for $k = |\mathbb{F}_\varphi|$ we have:

$$\mathcal{G}_{\mathcal{T}} \models \alpha \overset{\text{Rmk. 28}}{\iff} \mathsf{copy}_k(\mathcal{T}) \models \alpha' \overset{\text{Prop. 8}}{\iff} \mathcal{T} \models \beta.$$

Thus, we have to check whether there is a Kripke e-tree $\mathcal{T}$ such that $\mathcal{T} \models \psi \wedge \beta$, where $\psi \wedge \beta$ is a $\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B})$-sentence. By Theorem 3 this is decidable, which completes the proof of Theorem 21.

5. Concrete domains over the integers

The main technical result of this section is:

Proposition 31. *The concrete domain* $\mathcal{Z} = (\mathbb{Z}, <, \equiv, (\equiv_a)_{a\in\mathbb{Z}}, (\equiv_{a,b})_{0\le a<b})$ *from* (1) *has the property* $\mathsf{EHD}(\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B}))$.

Since $\mathcal{Z}$ is negation-closed (see Example 20), the following result (our main result) follows by Theorem 21:

Theorem 32. $\mathsf{SAT}_{\mathsf{ECTL}^*}(\mathcal{Z})$ *is decidable.*

We prove Proposition 31 in three steps. First, we show that $(\mathbb{Z}, <)$ has the property $\mathsf{EHD}(\mathsf{WMSO{+}B})$. In a second step we extend this result to the structure $(\mathbb{Z}, <, \equiv)$. Finally we add a countable set of unary predicates satisfying certain computability requirements, and show that $\mathcal{Z}$ is an instance of this case.

5.1. $\mathbb{Z}$ with order-constraints

Our first goal is to prove:

Proposition 33. $(\mathbb{Z}, <)$, $(\mathbb{N}, <)$ *and* $(\mathbb{Z} \setminus \mathbb{N}, <)$ *have the property* $\mathsf{EHD}(\mathsf{WMSO{+}B})$.

As a preparation of the proof, we first define some terminology and then we characterize structures that allow homomorphisms to $(\mathbb{Z}, <)$ in terms of their paths. Let $\mathcal{A} = (A, I)$ be a countable $\{<\}$-structure and set $E = I(<)$. When talking about paths, we always refer to finite directed E-paths. The length of a path $(a_0, a_1, \ldots, a_n)$ is n. For $S \subseteq A$ and $x \in A \setminus S$, a path from x to S is a path from x to some node $y \in S$. A path from S to x is defined symmetrically.

Lemma 34. *We have $\mathcal{A} \preceq (\mathbb{Z}, <)$ if and only if*

(H1) *$\mathcal{A}$ does not contain cycles, and*

(H2) *for all $a, b \in A$ there is an $n \in \mathbb{N}$ such that the length of each path from a to b is bounded by n.*

PROOF.

($\Rightarrow$) Let us first show the "only if" direction of the lemma. Suppose h is a homomorphism from $\mathcal{A}$ to $(\mathbb{Z}, <)$. Heading for a contradiction, suppose that there is a cycle $(a_0, \ldots, a_k)$ in $\mathcal{A}$, i.e., $(a_0, \ldots, a_k)$ is a path such that $(a_k, a_0) \in E$. Setting $z_i = h(a_i)$, this implies $z_i < z_{i+1}$ for $0 \leq i \leq k-1$ and $z_k < z_0$ which is a contradiction. Hence, (H1) holds.

Suppose now that $a, b \in A$ are such that for every $n \in \mathbb{N}$ there is a path of length at least n from a to b. If $d = h(b) - h(a)$, we can find a path $(a_0, a_1 \ldots, a_k)$ with $a_0 = a$, $a_k = b$ and $k > d$. Since h is a homomorphism, this path is mapped to an increasing sequence of integers $h(a) = h(a_0) < h(a_1) < \cdots < h(a_k) = h(b)$. But this contradicts $h(b) - h(a) = d < k$. Hence, (H2) holds.

($\Leftarrow$) For the "if" direction of the lemma assume that $\mathcal{A}$ is acyclic (property (H1)) and that (H2) holds. Fix an enumeration $a_0, a_1, a_2, \ldots$ of the countable set A. For $n \geq 0$ let

$$S_n = \{a \in A \mid \exists i, j \leq n : (a_i, a), (a, a_j) \in E^*\}.$$

We claim that S_n has the following properties.

(P1) S_n is convex w.r.t. the partial order E^*: If $a, c \in S_n$ and $(a, b), (b, c) \in E^*$, then $b \in S_n$.

(P2) For $a \in A \setminus S_n$ all paths between a and S_n are "one-way", i.e., there are not $b, c \in S_n$ such that $(b, a), (a, c) \in E^*$.

(P3) For all $a \in A \setminus S_n$ there is a bound $c \in \mathbb{N}$ such that all paths between a and S_n have length at most c. Let $c_n^a \in \mathbb{N}$ be the smallest such bound (hence, we have $c_n^a = 0$ if there is no path between a and S_n).

(P1) is obvious and moreover implies (P2). To see (P3), assume that there are only paths from S_n to a but not the other way round (see (P2)); the other case is symmetric. If there is no bound on the length of paths from S_n to a, then by definition of S_n, there is no bound on the length of paths from $\{a_0, \ldots, a_n\}$ to a. By the pigeon principle, there is a $0 \leq i \leq n$ such that there is no bound on the length of paths from a_i to a. But this contradicts property (H2).

We build the homomorphism h inductively. For every $n \geq 0$ we define functions $h_n : S_n \to \mathbb{Z}$ such that the following invariants hold for all $n \geq 0$.

(I1) If $n > 0$ then $h_n(a) = h_{n-1}(a)$ for all $a \in S_{n-1}$.

(I2) $h_n(S_n)$ is bounded in $\mathbb{Z}$, i.e., there are $z_1, z_2 \in \mathbb{Z}$ such that $h_n(S_n) \subseteq [z_1, z_2]$.

(I3) h_n is a homomorphism from the induced subgraph $\mathcal{A}\restriction_{S_n}$ to $(\mathbb{Z}, <)$.

For $n = 0$ we have $S_0 = \{a_0\}$. We set $h_0(a_0) = 0$ (any other integer would be also fine). Properties (I1)–(I3) are easily verified. For $n > 0$, there are four cases.

1. $a_n \in S_{n-1}$, thus $S_n = S_{n-1}$. We set $h_n = h_{n-1}$. Clearly, (I1)–(I3) hold for n.
2. $a_n \notin S_{n-1}$ and there is no path from a_n to S_{n-1} or vice versa. We set $h_n(a_n) := 0$. Since $S_n = S_{n-1} \cup \{a_n\}$, (I1)–(I3) follow easily from the induction hypothesis.
3. $a_n \notin S_{n-1}$ and there is a path from a_n to S_{n-1}. Then, by (P2) there are no paths from S_{n-1} to a_n. Hence, we have

$$S_n = S_{n-1} \cup \{a \in A \mid \exists b \in S_{n-1} : (a_n, a), (a, b) \in E^*\}.$$

We have to define the value $h_n(a)$ for all $a \in A \setminus S_{n-1}$ that lie along a path from a_n to S_{n-1}. By (I2) there are $z_1, z_2 \in \mathbb{Z}$ with $h_{n-1}(S_{n-1}) \subseteq [z_1, z_2]$. Recall the definition of c_{n-1}^a from (P3). For

all $a \in A \setminus S_{n-1}$ that lie on a path from a_n to S_{n-1}, we set $h_n(a) = z_1 - c^a_{n-1}$. Since there are paths from a to S_{n-1}, we have $c^a_{n-1} > 0$. Hence, for all $a \in S_n \setminus S_{n-1}$, $h_n(a) < z_1$. Let us check that $h_n : S_n \to \mathbb{Z}$ satisfy (I1)– (I3): Invariant (I1) holds by definition of h_n. For (I2) note that $h_n(S_n) \subseteq [z_1 - c^{a_n}_{n-1}, z_2]$.
It remains to show (I3), i.e., that h_n is a homomorphism from $\mathcal{A}{\restriction}_{S_n}$ to $(\mathbb{Z}, <)$. Hence, we have to show that $h(b_1) < h(b_2)$ for all $(b_1, b_2) \in E \cap (S_n \times S_n)$.
   - If $b_1, b_2 \in S_{n-1}$, then $h_n(b_1) = h_{n-1}(b_1) < h_{n-1}(b_2) = h_n(b_2)$ by the induction hypothesis.
   - If $b_1 \in S_n \setminus S_{n-1}$ and $b_2 \in S_{n-1}$, we know that $h_n(b_2) = h_{n-1}(b_2) \geq z_1$ while $h_n(b_1) < z_1$ by construction. Hence, we have $h_n(b_1) < h_n(b_2)$.
   - If $b_2 \in S_n \setminus S_{n-1}$ and $b_1 \in S_{n-1}$, then $(b_1, b_2) \in E$ contradicts (P2) because b_2 is on a path from a_n to S_{n-1} and (b_1, b_2) is a path in the opposite direction.
   - If both b_1 and b_2 belong to $S_n \setminus S_{n-1}$ then $h_n(b_i) = z_1 - c^{b_i}_{n-1}$ for $i \in \{1, 2\}$ Since $(b_1, b_2) \in E$, we have $c^{b_1}_{n-1} > c^{b_2}_{n-1}$. This implies $h_n(b_1) < h_n(b_2)$.

4. $a_n \notin S_{n-1}$ and there is a path from S_{n-1} to a_n. For all

$$a \in S_n \setminus S_{n-1} = \{a \in A \setminus S_{n-1} \mid a \text{ belongs to a path from } S_{n-1} \text{ to } a_n\},$$

set $h_n(a) = z_2 + c^a_{n-1}$. The rest of the argument is analogous to the previous case.

This concludes the construction of h_n. Thanks to (I1), the limit function $h = \bigcup_{i \in \mathbb{N}} h_i$ exists. By (I3) and $A = \bigcup_{i \in \mathbb{N}} S_i$, h is a homomorphism from $\mathcal{A}$ to $(\mathbb{Z}, <)$. □

A result similar to Lemma 34 holds for $(\mathbb{N}, <)$. Here the characterization of homomorphisms relies on the fact that if some element a is mapped to $n \in \mathbb{N}$ by some homomorphism, then a path leading to a is at most of length n.

Lemma 35. *We have $\mathcal{A} \preceq (\mathbb{N}, <)$ if and only if*

(H1) *$\mathcal{A}$ does not contain cycles, and*
(H2) *for all $a \in A$ there is an $n \in \mathbb{N}$ such that the length of each path ending in a is bounded by n.*

Proof. If $h : A \to \mathbb{N}$ is a homomorphism and $a \in A$ then every path ending in a can be of length at most $h(a)$. Moreover, $\mathcal{A}$ must be acyclic by the same argument that we used for $(\mathbb{Z}, <)$.

For the other direction assume that $\mathcal{A}$ is acyclic and for each $a \in A$ there is some $c_a \in \mathbb{N}$ such that the longest path leading to a has length c_a. Define $h(a) = c_a$. It is rather straightforward to show that h is a homomorphism. □

Proof (Proposition 33). For $(\mathbb{Z}, <)$, we translate the conditions (H1) and (H2) from Lemma 34 into WMSO+B. Cycles are excluded by the sentence $\neg\mathsf{ECycle}_<$ (Example 2). Moreover, for an acyclic $\{<\}$-structure $\mathcal{A}$ we have $\mathcal{A} \models \forall x \forall y\, \mathsf{BPaths}_<(x, y)$ (see also Example 2) if and only if for all $a, b \in A$ there is a bound $n \in \mathbb{N}$ on the length of paths from a to b. Thus,

$$\mathcal{A} \preceq (\mathbb{Z}, <) \text{ if and only if } \mathcal{A} \models \neg\mathsf{ECycle}_< \wedge \forall x \forall y\, \mathsf{BPaths}_<(x, y).$$

Similarly, using Lemma 35, we obtain that

$$\mathcal{A} \preceq (\mathbb{N}, <) \text{ if and only if } \mathcal{A} \models \neg\mathsf{ECycle}_< \wedge \forall y\, \mathsf{B}Z\, \exists x\, \mathsf{Path}_<(x, y, Z).$$

Since $(\mathbb{Z} \setminus \mathbb{N}, <)$ is $(\mathbb{N}, <)$ with reversed order, one proves analogously that

$$\mathcal{A} \preceq (\mathbb{Z} \setminus N, <) \text{ if and only if } \mathcal{A} \models \neg\mathsf{ECycle}_< \wedge \forall x\, \mathsf{B}Z\, \exists y\, \mathsf{Path}_<(x, y, Z).$$

□

5.2. $\mathbb{Z}$ with order- and equality-constraints

In this section, we extend Proposition 33 to the negation-closed structure $(\mathbb{Z}, <, \equiv)$. For this purpose, given a structure $\mathcal{A} = (A, I)$ over the signature $\{\equiv\} \uplus \sigma$ we define *the quotient of* $\mathcal{A}$, obtained by *contracting* all the $\equiv$-paths (note that $I(\equiv)$ is usually not the identity relation on A).

Definition 36. Let $\sim\, = (I(\equiv) \cup I(\equiv)^{-1})^*$ be the smallest equivalence relation on A that contains $I(\equiv)$. We denote the $\sim$-quotient of $\mathcal{A}$ by $\tilde{\mathcal{A}} = (\tilde{A}, \tilde{I})$: It is a σ-structure with domain $\tilde{A} = \{[a] \mid a \in A\}$ the set of all $\sim$-equivalence classes. For all $R \in \sigma$ of arity k, we define $\tilde{I}(R)$ as the set of k-tuples $([a_1], \ldots, [a_k])$ for which there are $b_1 \in [a_1], \ldots, b_k \in [a_k]$ such that $(b_1, \ldots, b_k) \in I(R)$.

Remark 37. Let $\mathcal{A} = (A, I)$ be a $\{<, \equiv\}$-structure and $\tilde{\mathcal{A}} = (\tilde{A}, \tilde{I})$ its quotient. In this case we have that $([a], [b]) \in \tilde{I}(<)$ iff there are $a' \sim a$ and $b' \sim b$ such that $(a', b') \in I(<)$. Since $\sim$ is the reflexive and transitive closure of the first-order definable relation $I(\equiv) \cup I(\equiv)^{-1}$, we can construct a WMSO-formula $\tilde{\varphi}(x, y)$ (using the reach-construction from Example 2) that defines $\sim$. That is, $\sim$ is WMSO-definable (and MSO-definable as well).

Let $\mathcal{C} = (C, J)$ be a structure over the signature $\{\equiv\} \cup \sigma$ where $J(\equiv)$ is real equality, i.e., $J(\equiv) = \{(c, c) \mid c \in C\}$. In this case the quotient $\tilde{\mathcal{C}} = (\tilde{C}, \tilde{J})$ is isomorphic to the reduct of $\mathcal{C}$ with signature σ. Whenever $\equiv$ is interpreted as real equality in the target structure, taking the quotient of a structure is compatible with the existence of homomorphisms in the following sense.

Lemma 38. *Let $\mathcal{C} = (C, J)$ be a concrete domain over $\{\equiv\} \uplus \sigma$ where $J(\equiv)$ is real equality. Then, for every $\tau \subseteq \sigma$, and every $(\{\equiv\} \cup \tau)$-structure $\mathcal{A} = (A, I)$,*

1. *$\mathcal{A} \preceq \mathcal{C}$ if and only if $\tilde{\mathcal{A}} \preceq \tilde{\mathcal{C}}$.*
2. *$\mathcal{C}$ has the property $\mathsf{EHD}(\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B}))$ if and only if $\tilde{\mathcal{C}}$ does.*

Proof.

1. For the direction ($\Rightarrow$) let $h : \mathcal{A} \to \mathcal{C}$ be a homomorphism. We show that $g : \tilde{\mathcal{A}} \to \tilde{\mathcal{C}}$, defined by $g([a]) = h(a)$, is a homomorphism as well. Notice that the mapping g is well defined: $[a] = [b]$ implies $(a, b) \in (I(\equiv) \cup I(\equiv)^{-1})^*$. Since h is a homomorphism, $(h(a), h(b)) \in J(\equiv)$, i.e., $h(a) = h(b)$.
   Then let $([a_1], \ldots, [a_k]) \in \tilde{I}(R)$ for some $R \in \tau$. By definition of $\tilde{I}$, there are $b_1 \in [a_1], \ldots, b_k \in [a_k]$ such that $(b_1, \ldots, b_k) \in I(R)$. Therefore $(g([a_1]), \ldots, g([a_k])) = (h(b_1), \ldots, h(b_k))$, and since h is a homomorphism, $(h(b_1), \ldots, h(b_k)) \in J(R) = \tilde{J}(R)$ as wanted.
   For the direction ($\Leftarrow$) let $h : \tilde{\mathcal{A}} \to \tilde{\mathcal{C}}$ be a homomorphism. We define $g : \mathcal{A} \to \mathcal{C}$ by $g(a) = h([a])$. Then let $R \in \tau$ and $b_1, \ldots, b_k \in A$ such that $(b_1, \ldots, b_k) \in I(R)$. This implies that $([b_1], \ldots, [b_k]) \in \tilde{I}(R)$ and therefore $(h([b_1]), \ldots, h([b_k])) = (g(b_1), \ldots, g(b_k)) \in J(R)$.
   Finally, if $a, b \in A$ are such that $(a, b) \in I(\equiv)$, then $[a] = [b]$. Therefore $g(a) = h([a]) = h([b]) = g(b)$, i.e., $(g(a), g(b)) \in J(\equiv)$. This proves that g is a homomorphism.
2. Let $\mathcal{L} = \mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B}))$ in the following arguments. Since $\tilde{\mathcal{C}}$ is a reduct of $\mathcal{C}$, it is clear that property $\mathsf{EHD}(\mathcal{L})$ for $\mathcal{C}$ implies property $\mathsf{EHD}(\mathcal{L})$ for $\tilde{\mathcal{C}}$. For the other direction, assume that $\tilde{\mathcal{C}}$ has the property $\mathsf{EHD}(\mathcal{L})$. Let $\tau \subseteq \sigma \uplus \{\equiv\}$ be a finite subsignature. If τ does not contain $\equiv$ then, by the property $\mathsf{EHD}(\mathcal{L})$ for $\tilde{\mathcal{C}}$, there exists an $\mathcal{L}$-sentence ψ_τ such that for every τ-structure $\mathcal{A}$ we have $\mathcal{A} \models \psi_\tau$ if and only if $\mathcal{A} \preceq \tilde{\mathcal{C}}$. But the latter is equivalent to $\mathcal{A} \preceq \mathcal{C}$ (since τ does not contain $\equiv$).
   Hence, we can assume that τ contains $\equiv$. Let $\tau' = \tau \setminus \{\equiv\}$. Since $\tilde{\mathcal{C}}$ has property $\mathsf{EHD}(\mathcal{L})$, we can find an $\mathcal{L}$-sentence $\psi_{\tau'}$, such that every τ'-structure $\mathcal{A}$,
   $$\mathcal{A} \models \psi_{\tau'} \iff \mathcal{A} \preceq \tilde{\mathcal{C}}. \tag{10}$$
   By Remark 37 there is an $\mathcal{L}$-formula $\tilde{\varphi}(x, y)$ that defines the equivalence relation $\sim$. Let $\tilde{\theta}_\tau$ be the $\mathcal{L}$-sentence obtained by replacing in $\psi_{\tau'}$ every occurrence of an atomic formula $R(x_1, \ldots, x_k)$ for $R \in \tau'$ by
   $$\tilde{R}(x_1, \ldots, x_k) := \exists z_1 \cdots \exists z_k \, (\tilde{\varphi}(z_1, x_1) \wedge \cdots \wedge \tilde{\varphi}(z_k, x_k) \wedge R(z_1, \ldots, z_k)).$$

We claim that for every τ-structure $\mathcal{B} = (B, I)$,

$$\mathcal{B} \models \tilde{\theta}_\tau \iff \tilde{\mathcal{B}} \models \psi_{\tau'}. \tag{11}$$

Using this claim, we obtain

$$\mathcal{B} \models \tilde{\theta}_\tau \iff \tilde{\mathcal{B}} \models \psi_{\tau'} \overset{(10)}{\iff} \tilde{\mathcal{B}} \preceq \tilde{\mathcal{C}} \overset{1.}{\iff} \mathcal{B} \preceq \mathcal{C},$$

which implies that $\mathcal{B}$ has the property $\mathsf{EHD}(\mathcal{L})$, as wanted.

The proof of (11) is by induction on the structure of the formula, the only non-trivial case being $\mathcal{B} \models \tilde{R}(a_1, \ldots, a_k)$ if and only if $\tilde{\mathcal{B}} \models R([a_1], \ldots, [a_k])$. Note that $\tilde{\mathcal{B}} \models R([a_1], \ldots, [a_k])$ if and only if there are $b_1, \ldots, b_k \in B$ such that $b_j \sim a_j$ and $\mathcal{B} \models R(b_1, \ldots, b_k)$, which is exactly what $\tilde{R}(a_1, \ldots, a_k)$ expresses. $\square$

An application of the previous lemma to Proposition 33 directly yields the following results.

Proposition 39. *$(\mathbb{Z}, <, \equiv)$, $(\mathbb{N}, <, \equiv)$, and $(\mathbb{Z} \setminus \mathbb{N}, <, \equiv)$ have property* $\mathsf{EHD}(\mathsf{WMSO{+}B})$.

5.3. Adding Unary Predicates

We now lift our result to expansions of $(\mathbb{Z}, <, \equiv)$ by unary predicates that satisfy some computability assumptions. For the rest of this section, we fix a signature σ of unary predicates (not containing the symbols $\equiv$ and $<$) and a $(\sigma \cup \{\equiv, <\})$-structure $\mathcal{Z}_\sigma = (\mathbb{Z}, I)$ where $I(\equiv)$ and $I(<)$ are interpreted as expected.

Definition 40. We call a finite subset $\bar{P} \subseteq \sigma$ *bounded below* (*bounded above*, respectively) if $\bigcap_{P \in \bar{P}} I(P)$ is bounded below (bounded above, respectively).

We next define properties (C1) and (C2) that imply property $\mathsf{EHD}(\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B}))$ for $\mathcal{Z}_\sigma$.

(C1) The bounds of $\bar{P} \subseteq \sigma$ are effectively computable in the sense that we can decide, given a finite subset $\bar{P} \subseteq \sigma$, whether $\bar{P}$ is bounded below (above, respectively) and that we can compute, given a finite subset $\bar{P} \subseteq \sigma$ that is bounded below (above, respectively), a bound $\mathsf{l}(\bar{P}) \in \mathbb{Z}$ ($\mathsf{u}(\bar{P}) \in \mathbb{Z}$, respectively) such that $\mathsf{l}(\bar{P}) \leq z$ ($\mathsf{u}(\bar{P}) \geq z$, respectively) for all $z \in \bigcap_{P \in \bar{P}} I(P)$.

(C2) For all finite subsets $\bar{P}_1, \bar{P}_2 \subseteq \sigma$ and all predicates $P \in \sigma$, if $\bar{P}_1$ is bounded below and $\bar{P}_2$ is bounded above, then we can effectively compute the finite set $I(P) \cap [\mathsf{l}(\bar{P}_1), \mathsf{u}(\bar{P}_2)]$.

The main result of this section is the following proposition.

Proposition 41. *If σ and I are chosen in such a way that $\mathcal{Z}_\sigma$ satisfies conditions (C1) and (C2), then $\mathcal{Z}_\sigma$ has property* $\mathsf{EHD}(\mathsf{WMSO{+}B})$. *The analogous result holds for $\mathcal{N}_\sigma = \mathcal{Z}_\sigma{\restriction}_{\mathbb{N}}$.*

We fix a finite subsignature $\tau \subseteq \sigma$. Due to (C1), we can compute $m < M \in \mathbb{Z}$ such that m is a lower bound for all $\bar{P} \subseteq \tau$ that are bounded below and M is an upper bound for all $\bar{P} \subseteq \tau$ that are bounded above. We fix the numbers m and M for the rest of this section.

Let $\mathcal{A} = (A, J)$ be a $(\tau \cup \{<, \equiv\})$-structure. The proof of Proposition 41 uses a decomposition of $\mathcal{A}$ into four parts, called "the bounded part", "the greater part", "the smaller part" and "the rest".

Intuitively, an element $a \in A$ belongs to the bounded part if we know a priori that any homomorphism h from $\mathcal{A}$ to $\mathcal{Z}_\tau$ (we write $\mathcal{Z}_\tau$ for the reduct of $\mathcal{Z}_\sigma$ with signature $\tau \cup \{\equiv, <\}$) maps a to an element in the interval $[m, M]$. Similarly, the greater part consists of all elements $a \in A$ that do not belong the bounded part but any homomorphism to $\mathcal{Z}_\tau$ must map a above m, and the smaller part consists of all elements $a \in A$ that do not belong the bounded part but any homomorphism to $\mathcal{Z}_\tau$ must map a below M.

We then reduce the question whether $\mathcal{A}$ can be embedded into $\mathcal{Z}_\tau$ to the questions whether the bounded part satisfies a certain WMSO-formula and whether the $\{<, \equiv\}$-reducts of everything except for the bounded part, its greater part and its smaller part allow homomorphisms to $(\mathbb{Z}, <, \equiv)$, $(\mathbb{N}, <, \equiv)$, and $(\mathbb{Z} \setminus \mathbb{N}, <, \equiv)$, respectively.

Definition 42. Let $\mathcal{A} = (A, J)$ be a $(\tau \cup \{<, \equiv\})$-structure. We denote by $\tilde{\mathcal{A}} = (\tilde{A}, \tilde{J})$ the $\sim$-quotient of $\mathcal{A}$ (cf. Definition 36).

We call $a \in A$ *bounded below* if there is some $b \in A$, a $<$-path in $\tilde{\mathcal{A}}$ from $[b]$ to $[a]$, and a subset $\bar{P} \subseteq \tau$ which is bounded below such that $[b] \in \tilde{J}(P)$ for all $P \in \bar{P}$.

We call $a \in A$ *bounded above* if there is some $b \in A$, a $<$-path in $\tilde{\mathcal{A}}$ from $[a]$ to $[b]$, and a subset $\bar{P} \subseteq \tau$ which is bounded above such that $[b] \in \tilde{J}(P)$ for all $P \in \bar{P}$.

With these preparations, we can easily define the four substructures mentioned above.

Definition 43. For a $(\tau \cup \{<, \equiv\})$-structure $\mathcal{A} = (A, J)$ we define

- the bounded part $B = \{a \in A \mid a \text{ is bounded below and bounded above}\}$,
- the greater part $G = \{a \in A \mid a \text{ is bounded below but not bounded above}\}$,
- the smaller part $S = \{a \in A \mid a \text{ is bounded above but not bounded below}\}$, and
- the rest $R = \{a \in A \mid a \text{ is neither bounded above nor bounded below}\}$.

Let us start with two simple lemmas.

Lemma 44. *Let $h : \mathcal{A} \to \mathcal{Z}_\tau$ be a homomorphism. Then the following holds:*

- *If $a \in B$ then $m \leq h(a) \leq M$.*
- *If $a \in S$ then $h(a) \leq M$.*
- *If $a \in G$ then $m \leq h(a)$*

Proof. It suffices to show that if a is bounded below (bounded above, respectively), then $m \leq h(a)$ ($h(a) \leq M$, respectively). If a is bounded below, then if there is some $b \in A$, a $<$-path in $\tilde{\mathcal{A}}$ from $[b]$ to $[a]$, and a subset $\bar{P} \subseteq \tau$ which is bounded below such that $[b] \in \tilde{J}(P)$ for all $P \in \bar{P}$. We get $m \leq h(b) \leq h(a)$. If a is bounded above, we can argue in the same way. □

Lemma 45. *The following relations are disjoint from $J(<)$: $B \times S$, $B \times R$, $G \times B$, $R \times B$, $G \times S$, $G \times R$, $R \times S$.*

Proof. Assume for instance $(b, s) \in J(<)$ for some $b \in B$ and $s \in S$. Since $b \in B$, b is bounded from below. Hence, there is some $c \in A$, a $<$-path in $\tilde{\mathcal{A}}$ from $[c]$ to $[b]$, and a subset $\bar{P} \subseteq \tau$ which is bounded below such that $[b] \in \tilde{J}(P)$ for all $P \in \bar{P}$. Hence, there is also a $<$-path in $\tilde{\mathcal{A}}$ from $[c]$ to $[s]$, i.e., s is bounded below, which contradicts $s \in S$. □

Remark 46. The parts B, G, S, and R are all MSO- and WMSO-definable in the sense that there are MSO-formulas $\chi_i(x)$ for $i \in \{B, G, S, R\}$ with one free first-order variable x such that $\mathcal{A} \models \chi_i(a)$ for each $a \in A$ if and only if a belongs to the part i (and the same holds if we interpret $\chi_i(x)$ as a WMSO-formula).

We next state three lemmas that allow to prove Proposition 41.

Lemma 47. *We have $\mathcal{A} \preceq \mathcal{Z}_\tau$ if and only if $\mathcal{A}{\restriction_B} \preceq \mathcal{Z}_\tau{\restriction_{[m,M]}}$ and $\mathcal{A}{\restriction_{G \cup S \cup R}} \preceq \mathcal{Z}_\tau$.*

Lemma 48. *Given a finite $\tau \subseteq \sigma$ we can compute an MSO-sentence ψ_B such that $\mathcal{A}{\restriction_B} \preceq \mathcal{Z}_\tau{\restriction_{[m,M]}}$ if and only if $\mathcal{A}{\restriction_B} \models \psi_B$.*

Lemma 49. *The following four conditions are equivalent:*

1. *There is a homomorphism $h : \mathcal{A}{\restriction_{G \cup S \cup R}} \to \mathcal{Z}_\tau{\restriction_{\mathbb{Z} \setminus [m,M]}}$ with $h(G) \subseteq [M+1, \infty)$, $h(S) \subseteq (-\infty, m-1]$.*
2. *$\mathcal{A}{\restriction_{G \cup S \cup R}} \preceq \mathcal{Z}_\tau$*

3. $(G \cup S \cup R, J(<), J(\equiv)) \preceq (\mathbb{Z}, <, \equiv)$, $(G, J(<), J(\equiv)) \preceq (\mathbb{N}, <, \equiv)$, *and* $(S, J(<), J(\equiv)) \preceq (\mathbb{Z} \setminus \mathbb{N}, <, \equiv)$
4. *There is a homomorphism* $h : (G \cup S \cup R, J(<), J(\equiv)) \to (\mathbb{Z}, <, \equiv)$ *with* $h(G) \subseteq \mathbb{N}$, $h(S) \subseteq \mathbb{Z} \setminus \mathbb{N}$.

Before we prove these lemmas, we show how they imply Proposition 41.

PROOF (PROPOSITION 41). Fix a finite subsignature $\tau \subseteq \sigma$. By Lemma 47 we have $\mathcal{A} \preceq \mathcal{Z}_\tau$ if and only if $\mathcal{A}{\restriction}_B \preceq \mathcal{Z}_\tau{\restriction}_{[m,M]}$ and $\mathcal{A}{\restriction}_{G \cup S \cup R} \preceq \mathcal{Z}_\tau$. By Lemma 48 we can compute from τ an MSO-sentence ψ_B such that $\mathcal{A}{\restriction}_B \models \psi_B$ if and only if $\mathcal{A}{\restriction}_B \preceq \mathcal{Z}_\tau{\restriction}_{[m,M]}$. Moreover, from Lemma 49 we know that $\mathcal{A}{\restriction}_{G \cup S \cup R} \preceq \mathcal{Z}_\tau$ if and only if

- $(G \cup S \cup R, J(<), J(\equiv)) \preceq (\mathbb{Z}, <, \equiv)$,
- $(G, J(<), J(\equiv)) \preceq (\mathbb{N}, <, \equiv)$, and
- $(S, J(<), J(\equiv)) \preceq (\mathbb{Z} \setminus \mathbb{N}, <, \equiv)$.

Each of the structures $(\mathbb{Z}, <, \equiv), (\mathbb{N}, <, \equiv)$, $(\mathbb{Z} \setminus \mathbb{N}, <, \equiv)$ has the property EHD(Bool(MSO, WMSO+B)). Hence, there are Bool(MSO, WMSO+B)-sentences ψ_G, ψ_S, and ψ_R such that $\mathcal{A}{\restriction}_{G \cup S \cup R} \preceq \mathcal{Z}_\tau$ if and only if

- $(G \cup S \cup R, J(<), J(\equiv)) \models \psi_R$,
- $(G, J(<), J(\equiv)) \models \psi_G$, and
- $(S, J(<), J(\equiv)) \models \psi_S$.

Since the subsets B, G, S and R of $\mathcal{A}$ are MSO-definable as well as WMSO-definable (cf. Remark 46), we can compute relativizations of ψ_i to part i for $i \in \{B, G, S, R\}$ and obtain Bool(MSO, WMSO+B)-sentences $\varphi_B, \varphi_R, \varphi_G$ and φ_S such that

- $\mathcal{A} \models \varphi_B$ if and only if $\mathcal{A}{\restriction}_B \models \psi_B$,
- $\mathcal{A} \models \varphi_R$ if and only if $(G \cup S \cup R, J(<), J(\equiv)) \models \psi_R$,
- $\mathcal{A} \models \varphi_G$ if and only if $(G, J(<), J(\equiv)) \models \psi_G$, and
- $\mathcal{A} \models \varphi_S$ if and only if $(S, J(<), J(\equiv)) \models \psi_S$.

Putting everything together, we have $\mathcal{A} \preceq \mathcal{Z}_\tau$ if and only if $\mathcal{A} \models \varphi_B \wedge \varphi_G \wedge \varphi_S \wedge \varphi_R$. □

We now prove the auxiliary lemmas (in a different order).

PROOF (LEMMA 49). The direction $(1 \Rightarrow 2)$ is trivial. Let us prove $(2 \Rightarrow 3)$, $(3 \Rightarrow 4)$, and $(4 \Rightarrow 1)$.

$(2 \Rightarrow 3)$ Let $h : \mathcal{A}{\restriction}_{G \cup S \cup R} \to \mathcal{Z}_\tau$ be a homomorphism. It follows immediately that h is also a homomorphism from the reduct $(G \cup S \cup R, J(<), J(\equiv))$ to $(\mathbb{Z}, <, \equiv)$.

Let $a \in G$. Then $h(a) \geq m$ by Lemma 44. Setting $h' : G \to \mathbb{N}$ with $h'(a) = h(a) - m$ yields a homomorphism from $(G, J(<), J(\equiv))$ to $(\mathbb{N}, <, \equiv)$.

The proof for $(S, J(<), J(\equiv)) \preceq (\mathbb{Z} \setminus \mathbb{N}, <, \equiv)$ is analogous.

$(3 \Rightarrow 4)$ Assume that there are homomorphisms

$$\begin{aligned} h &: (G \cup S \cup R, J(<), J(\equiv)) \to (\mathbb{Z}, <, \equiv), \\ h_G &: (G, J(<), J(\equiv)) \to (\mathbb{N}, <, \equiv), \text{ and} \\ h_S &: (S, J(<), J(\equiv)) \to (\mathbb{Z} \setminus \mathbb{N}, <, \equiv). \end{aligned}$$

Define the mapping $h' : G \cup S \cup R \to \mathbb{Z}$ by

$$h'(a) = \begin{cases} h(a) & \text{if } a \in R, \\ \max(h(a), h_G(a)) & \text{if } a \in G, \\ \min(h(a), h_S(a)) & \text{if } a \in S. \end{cases}$$

With Lemma 45 one easily concludes that this is the desired homomorphism.

(4 $\Rightarrow$ 1) Let $h : (G \cup S \cup R, J(<), J(\equiv)) \to (\mathbb{Z}, <, \equiv)$ be the homomorphism from 4. Let $\mathcal{P}^+$ be the set of subsets of τ that are not bounded above and let $\mathcal{P}^-$ be the set of subsets of τ that are not bounded below. We define a sequence $(\eta_i)_{i\in\mathbb{Z}}$ of integers as follows:

- $\eta_0 = M + 1$,
- η_{-1} is the maximal number such that for each $\bar{P} \in \mathcal{P}^-$ there is a $\eta_{-1} \le z < m$ with $z \in I(P)$ for all $P \in \bar{P}$ (we set $\eta_{-1} = m - 1$ if $\mathcal{P}^- = \emptyset$),
- for $i > 0$ let η_i be minimal such that for each $\bar{P} \in \mathcal{P}^+$ there is a $\eta_{i-1} \le z < \eta_i$ with $z \in I(P)$ for all $P \in \bar{P}$ (we set $\eta_i = \eta_{i-1} + 1$ if $\mathcal{P}^+ = \emptyset$),
- for $i < -1$ let η_i be maximal such that for each $\bar{P} \in \mathcal{P}^-$ there is a $\eta_i \le z < \eta_{i+1}$ with $z \in I(P)$ for all $P \in \bar{P}$ (we set $\eta_i = \eta_{i+1} - 1$ if $\mathcal{P}^- = \emptyset$).

For all $a \in G \cup S \cup R$ let $\bar{P}_a = \{P \in \tau \mid [a] \in \tilde{J}(P)\}$. Note that for all $r \in R$, $\bar{P}_r$ is neither bounded above or below (otherwise r would be bounded above or below, respectively), for all $g \in G$, $\bar{P}_g$ is not bounded above and for all $s \in S$, $\bar{P}_s$ is not bounded below. We conclude that the following map $h' : G \cup S \cup R \to \mathbb{Z}$ is well defined:

$$h'(a) = \min\{z \in \mathbb{Z} \mid \eta_{h(a)} \le z < \eta_{h(a)+1} \text{ and } z \in I(P) \text{ for all } P \in \bar{P}_a\}.$$

Since h preserves $<$ and $\equiv$, h' does the same. Moreover, h' is defined in such a way that it preserves all unary predicates from τ.

Next we show that the image of h' has empty intersection with the interval $[m, M]$. By definition of η_{-1}, η_0 and h', $h'(a) \in [m, M]$ would imply $h(a) = -1$ Note that by by our assumptions on h, this implies $a \in R \cup S$. In particular, $\bar{P}_a$ cannot be bounded below, i.e., $\bar{P}_a \in \mathcal{P}^-$. Thus, there is a minimal $\eta_{-1} \le z < m$ such that $z \in I(P)$ for all $P \in \bar{P}_a$. This implies $h'(a) = z < m$ which completes our claim. Thus, h' is a homomorphism from $\mathcal{A}{\restriction}_{G\cup S\cup R}$ to $\mathcal{Z}_\tau{\restriction}_{\mathbb{Z}\setminus[m,M]}$.

To show that $h'(G) \subseteq [M + 1, \infty)$ and $h(S) \subseteq (-\infty, m - 1]$ note that $h(G) \subseteq \mathbb{N}$ and $h(S) \subseteq \mathbb{Z} \setminus \mathbb{N}$. This implies $h'(G) \subseteq [M + 1, \infty)$ and $h'(S) \subseteq (-\infty, M]$. Hence, $h'(S) \subseteq (-\infty, m - 1]$ by the previous paragraph. □

Proof (Lemma 47). If $h : \mathcal{A} \to \mathcal{Z}_\tau$ is a homomorphism, then the restrictions of h to B and $G \cup S \cup R$ witness $\mathcal{A}{\restriction}_B \preceq \mathcal{Z}_\tau{\restriction}_{[m,M]}$ (here we use Lemma 44) and $\mathcal{A}{\restriction}_{G\cup S\cup R} \preceq \mathcal{Z}_\tau$.

Now assume that $h_1 : \mathcal{A}{\restriction}_B \to \mathcal{Z}_\tau{\restriction}_B$ and $h_2 : \mathcal{A}{\restriction}_{G\cup S\cup R} \to \mathcal{Z}_\tau$ are homomorphisms. By Lemma 49 there exists a homomorphism $h_2' : \mathcal{A}{\restriction}_{G\cup S\cup R} \to \mathcal{Z}_\tau$ such that $h(G) \subseteq [M + 1, \infty)$ and $h(S) \subseteq (-\infty, m - 1]$. We define the mapping $h : A \to \mathbb{Z}$ by $h(b) = h_1(b) \in [m, M]$ for $b \in B$ and $h(a) = h_2'(a)$ for $a \in G \cup S \cup R$. This mapping preserves $\equiv$ and all unary predicated. Moreover, using Lemma 45 it follows easily that it preserves also the relation $<$. □

Proof (Lemma 48). A homomorphism $h : \mathcal{A}{\restriction}_B \to \mathcal{Z}_\tau{\restriction}_{[m,M]}$ can be identified with a partition of B into $M - m + 1$ sets $B_m, \ldots, B_M$, where $B_i = \{a \in B \mid h(a) = i\}$. Hence, the MSO-sentence ψ_B from Lemma 48 states that there is a partition of B into $M - m + 1$ sets $B_m, \ldots, B_M$ such that the corresponding mapping $h : B \to [m, M]$ preserves all relations from τ. Fixing a tuple of $M - m + 1$ many set variables $\overline{X} = (X_m, \ldots, X_M)$, we want to define formulas with the following properties:

- $\psi_{\text{part}}(\overline{X})$ expresses that $\overline{X}$ forms a finite partition.
- $\psi_<(\overline{X})$ expresses that the partition preserves the relation $I(<)$.
- $\psi_=(\overline{X})$ expresses that the partition preserves the relation $I(\equiv)$.
- $\psi_\tau(\overline{X})$ expresses that the partition preserves every unary relation $P \in \tau$.

These formulas can be defined as follows:

$$\psi_{\text{part}} = \forall x \bigvee_{i\in[m,M]} \Big(x \in X_i \wedge \bigwedge_{\substack{j\in[m,M]\\ i\neq j}} x \notin X_j \Big),$$

$$\psi_{<} = \forall x\, \forall y \left(x < y \to \bigvee_{\substack{i,j\in[m,M]\\ i<j}} (x \in X_i \wedge y \in X_j) \right),$$

$$\psi_{=} = \forall x\, \forall y \left(x \equiv y \to \bigvee_{i\in[m,M]} (x \in X_i \wedge y \in X_i) \right),$$

$$\psi_{\tau} = \bigwedge_{P\in\tau} \forall x \left(x \in P \to \bigvee_{i\in I(P)\cap[m,M]} x \in X_i \right).$$

Note that the formulas of the last form are all computable due to condition (C2). Now we can define $\psi_B = \psi_{\text{part}} \wedge \psi_< \wedge \psi_= \wedge \psi_\tau$. □

5.4. Expansions of $\mathbb{Z}$ that satisfy Conditions (C1) and (C2)

In this section, we will present concrete examples of unary relations that satisfy the conditions (C1) and (C2) from the previous section.

Definition 50. Define the signature

$$\sigma = \{F_S, C_S \mid S \subseteq \mathbb{Z} \text{ finite}\} \cup \{\equiv_{a,b} \mid a, b \in \mathbb{N}, a < b\},$$

where all symbols are unary. We define the structure $\mathcal{Z}_\sigma = (\mathbb{Z}, I)$ where $I(F_S) = S$, $I(C_S) = \mathbb{Z} \setminus S$, and $\equiv_{a,b}$ holds at all $z \in \mathbb{Z}$ such that $z = a \bmod b$.

Note that the $\mathcal{Z}_\sigma$ (which is defined over the signature $\sigma \cup \{<, \equiv\}$, see the first paragraph of Section 5.3) is an expansion of $\mathcal{Z} = (\mathbb{Z}, <, \equiv, (\equiv_a)_{a\in\mathbb{Z}}, (\equiv_{a,b})_{0\leq a<b})$ from (1) because $\equiv_a$ is the same relation as $F_{\{a\}}$.

Lemma 51. *$\mathcal{Z}_\sigma$ satisfies the conditions (C1) and (C2).*

Proof. The condition (C2) holds trivially because all set $I(P)$ for $P \in \sigma$ are computable sets and the map $P \mapsto I(P)$ is computable.

It remain to show (C1). Let $\bar{P} = \bar{F} \cup \bar{C} \cup \bar{M}$ be a finite set where $\bar{F} \subseteq \{F_S \mid S \subseteq \mathbb{Z} \text{ finite}\}$, $\bar{C} \subseteq \{C_S \mid S \subseteq \mathbb{Z} \text{ finite}\}$, and $\bar{M} \subseteq \{\equiv_{a,b} \mid a, b \in \mathbb{N}, a < b\}$.

Note that $\bar{F} \neq \emptyset$ implies that $\bar{P}$ is bounded above and below. Otherwise $\bar{P}$ is bounded above (below) if and only if $\bar{M}$ is bounded above (below). Let $\bar{M} = \{\equiv_{a_1,b_1}, \ldots, \equiv_{a_k,b_k}\}$ and

$$S = \bigcap_{i=1}^{k} \{a_i + zb_1 \mid z \in \mathbb{Z}\}.$$

The set S is either empty or of the form $\{y + z \cdot \operatorname{lcm}(b_1, b_2, \ldots, b_k) \mid z \in \mathbb{Z}\}$ for some $y \in \mathbb{Z}$ (and hence neither bounded below nor bounded above), where $\operatorname{lcm}(b_1, b_2, \ldots, b_k)$ is the least common multiple of the numbers $b_1, b_2, \ldots, b_k$. The latter holds if and only if $a_i = a_j \bmod \gcd((b_i, b_j)$ for all $i \neq j$, see e.g. [].

For those $\bar{P}$ that are bounded above (bounded below), it is easy to compute an upper bound (a lower bound). If $\bar{F}$ is nonempty, take an $F_S \in \bar{F}$ and use $\min(S)$ and $\max(S)$ as bounds. If $\bar{P}$ is bounded and $\bar{F}$ is empty, then 0 is a lower and upper bound. □

We can add further unary predicates and still have conditions (C1) and (C2). Let Prim be the set of prime numbers. Consider the expansion $\mathcal{Z}_{\sigma\cup\{\pi,\overline{\pi}\}}$ of the structure $\mathcal{Z}_\sigma$ from Definition 50, where $I(\pi) = \mathsf{Prim}$ and $I(\overline{\pi}) = \mathbb{Z} \setminus \mathsf{Prim}$. The following result of Dirichlet relates prime numbers to modulo constraints. Recall that two natural numbers n_1, n_2 are *coprime* if there is no prime p such that $p \mid n_1$ and $p \mid n_2$.

Theorem 52 (Dirichlet's Theorem). *Let $a < b \in \mathbb{N}$. The equation $x = a \bmod b$ has infinitely many solutions that are prime if and only if a and b are coprime.*

If a and b are not coprime, let $p \neq 1$ be a common divisor of both. Every solution of $x = a \bmod b$ is a multiple of p whence there is at most 1 solution that is a prime which can be computed from a and b.

There is an easy observation relating the complement of the primes with the modulo predicates.

Lemma 53. *For all numbers $a < b \in \mathbb{N}$ there are infinitely many solutions of $x = a \bmod b$ that are not prime numbers.*

PROOF. There are three cases:

- If $a = 0$, all solutions above b are not prime.
- If $a = 1$, assume that $n \in \mathbb{N}$ is a solution of $x = 1 \bmod b$. Then $n^k = 1 \bmod b$ for all $k \geq 2$ and we obtain infinitely many non-prime solutions.
- If $a > 1$, then for all $n \in \mathbb{N}$ we have $a + bna = a \bmod b$ and $a + bna$ is not a prime because it is a multiple of a. □

Corollary 54. *The structures $\mathcal{Z}_{\sigma\cup\{\pi,\overline{\pi}\}}$ has property* $\mathsf{EHD}(\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B}))$.

PROOF. Take a subset $\bar{P}$ of the unary relations from $\sigma \cup \{\pi, \overline{\pi}\}$, where σ is from Definition 50. Then, we first determine whether the intersection of all unary relations from $\sigma \cap \bar{P}$ is finite or not, as in the proof of Lemma 51. If the intersection is infinite then it is of the form $S = \{c + z \cdot b \mid z \in \mathbb{Z}\} \setminus F$ for $c < b \in \mathbb{Z}$ and a finite set $F \setminus \mathbb{Z}$, which can be computed. Clearly, $\mathsf{Prim} \cap S$ is bounded below by 0 and by Dirichlet's Theorem it is bounded above if and only if c and b are not coprime, in which case an upper bound can be computed from b and c. The set $(\mathbb{Z} \setminus \mathsf{Prim}) \cap S$ is neither bounded below nor bounded above (by Lemma 53). Since Prim and $\mathbb{Z} \setminus \mathsf{Prim}$ are computable, properties (C1) and (C2) hold. □

Since $\mathcal{Z}_{\sigma\cup\{\pi,\overline{\pi}\}}$ is also negation-closed, we get:

Corollary 55. *The problem $\mathsf{SAT}_{\mathsf{ECTL}^*}(\mathcal{Z}_{\sigma\cup\{\pi,\overline{\pi}\}})$ is decidable.*

At the end of this section, we briefly mention that the expansion of $\mathbb{Z}$ under consideration may contain undecidable unary predicates. Take some undecidable set $H \subseteq \mathbb{N}$, e.g., the halting problem. Consider the structure $\mathcal{Z}' = (\mathbb{Z}, <, \equiv, H, \bar{H})$, where $\bar{H} = \mathbb{Z} \setminus H$. Then $\{H, \bar{H}\}$ satisfies conditions (C1) and (C2). Just note that H is bounded below but not above, $\bar{H}$ is neither bounded below nor above and $H \cap \bar{H} = \emptyset$. Thus, the bounds m and M for the bounded part can be chosen to be $m = 0, M = -1$. The conditions on the bounded part reduce to the fact that it has to be empty. Since $\mathcal{Z}'$ is also negation-closed, we conclude that $\mathsf{SAT}_{\mathsf{ECTL}^*}(\mathcal{Z}')$ is decidable.

6. Extensions, and Applications

A simple adaptation of our proof for the concrete domain $\mathcal{Z}$ shows that the negation closed structure $\mathcal{Q} = (\mathbb{Q}, <, \equiv, (\equiv_q)_{q\in\mathbb{Q}})$ has the property $\mathsf{EHD}(\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B}))$ as well: We have $\mathcal{A} = (A, I) \preceq \mathcal{Q}$ if and only if

- (A, E) is acyclic, where $E = \mathrm{Id} \circ I(<) \circ \mathrm{Id}$ and Id is the relation $(I(\equiv) \cup I(\equiv)^{-1})^*$,
- there is no $(a, b) \in E^+$ with $a \in I(\equiv_p)$, $b \in I(\equiv_q)$ and $q \leq p$, and
- there is no $(a, b) \in \mathrm{Id}^*$ with $a \in I(\equiv_p)$, $b \in I(\equiv_q)$, and $q \neq p$.

It follows that $\mathsf{SAT}_{\mathsf{ECTL}^*}(\mathcal{Q})$ is decidable.

Let us finally state a simple preservation theorem for $\mathsf{SAT}_{\mathsf{ECTL}^*}(\mathcal{A})$. Assume that $\mathcal{A} = (A, I)$ and $\mathcal{B} = (B, J)$ are structures over countable signatures $\sigma_{\mathcal{A}}$ and $\sigma_{\mathcal{B}}$, respectively. We say that $\mathcal{A}$ is *existentially interpretable* in $\mathcal{B}$ if there exist $n \geq 1$ and quantifier-free first-order formulas $\varphi(y_1, \ldots, y_l, x_1, \ldots, x_n)$ and

$$\varphi_r(z_1, \ldots, z_{l_r}, x_{1,1}, \ldots, x_{1,n}, \ldots, x_{\mathsf{ar}(r),1}, \ldots, x_{\mathsf{ar}(r),n}) \text{ for } r \in \sigma_{\mathcal{A}}$$

over the signature $\sigma_{\mathcal{B}}$, where the mapping $r \mapsto \varphi_r$ has to be computable, such that $\mathcal{A}$ is isomorphic to the structure $(\{\overline{b} \in B^n \mid \exists \overline{c} \in B^l \colon \mathcal{B} \models \varphi(\overline{c}, \overline{b})\}, I)$ with

$$I(r) = \{(\overline{b}_1, \ldots, \overline{b}_k) \in B^{kn} \mid \exists \overline{c} \in B^{l_r} \colon \mathcal{B} \models \varphi_r(\overline{c}, \overline{b}_1, \ldots, \overline{b}_k)\} \text{ for } r \in \sigma_{\mathcal{A}} \text{ with } k = \mathsf{ar}(r).$$

Proposition 56. *If $\mathsf{SAT}_{\mathsf{ECTL}^*}(\mathcal{B})$ is decidable and $\mathcal{A}$ is existentially interpretable in $\mathcal{B}$, then $\mathsf{SAT}_{\mathsf{ECTL}^*}(\mathcal{A})$ is decidable too.*

Proof. Let ψ be an $\mathsf{ECTL}^*(\sigma_{\mathcal{A}})$-formula. Let $\mathbb{F}_\psi$ be the set of function symbols that occur in ψ. We use the notations introduced before Proposition 56. Let us choose new functions f_i, $g_{f,j}$, and $h_{r,k}$ for all $1 \leq i \leq n$, $f \in \mathbb{F}_\psi$, $1 \leq j \leq l$, $r \in \sigma_{\mathcal{A}}$, and $1 \leq k \leq l_r$. Define the $\mathsf{ECTL}^*(\sigma_{\mathcal{B}})$-formula

$$\theta = \psi' \wedge \mathsf{A}\forall x \bigwedge_{f \in \mathbb{F}_\psi} \varphi(g_{f,1}(x), \ldots, g_{f,l}(x), f_1(x), \ldots, f_n(x)),$$

where ψ' is obtained from ψ by replacing in ψ every constraint $R(f_1 S^{i_1}(x), \ldots, f_k S^{i_k}(x))$ (where $k = \mathsf{ar}(R)$) by the boolean formula

$$\varphi_r(h_{r,1}S^d(x), \ldots, h_{r,l_r}S^d(x), f_{1,1}S^{i_2}(x), \ldots, f_{1,n}S^{i_1}(x), \ldots, f_{k,1}S^{i_k}(x), \ldots, f_{k,n}S^{i_k}(x)),$$

where $d = \max\{i_1, \ldots, i_{\mathsf{ar}(r)}\}$. Using arguments similar to those from the proof of Lemma 22, one can show that ψ is $\mathcal{A}$-satisfiable if and only if θ is $\mathcal{B}$-satisfiable. $\square$

Examples of structures $\mathcal{A}$ that are existentially interpretable in $(\mathbb{Z}, <, \equiv)$, and hence have a decidable $\mathsf{SAT}_{\mathsf{ECTL}^*}(\mathcal{A})$-problem are:

- $(\mathbb{Z}^n, <_{\text{lex}}, \equiv)$ (for $n \geq 1$), where $<_{\text{lex}}$ denotes the strict lexicographic order on n-tuples of integers, and
- the structure $\text{Allen}_{\mathbb{Z}}$, which consists of all $\mathbb{Z}$-intervals together with Allen's relations b (before), a (after), m (meets), mi (met-by), o (overlaps), oi (overlapped by), d (during), di (contains), s (starts), si (started by), f (ends), fi (ended by). In artificial intelligence, Allen's relations are a popular tool for representing temporal knowledge.

7. Finite Satisfiability

Fix a signature σ and a negation-closed σ-structure $\mathcal{A} = (A, I)$. An $\mathsf{ECTL}^*(\sigma)$-formula φ is *finitely $\mathcal{A}$-satisfiable* if there is an $\mathcal{A}$-constraint graph $\mathfrak{C}$, whose underlying Kripke structure $\mathcal{K}$ is *finite*, and a node v of $\mathcal{K}$ such that $(\mathfrak{C}, v) \models \varphi$. We denote as $\mathsf{FINSAT}_{\mathsf{ECTL}^*}(\mathcal{A})$ the following computational problem: *Is a given formula $\varphi \in \mathsf{ECTL}^*(\sigma)$ finitely $\mathcal{A}$-satisfiable?* The main result of this section is the following.

Proposition 57. *Let ψ be an $\mathsf{ECTL}^*(\sigma)$-formula, and let $F \subseteq \mathbb{F}$ be the set of function symbols occurring in ψ. Then ψ is finitely $\mathcal{A}$-satisfiable if and only if there is an $\mathcal{A}$-constraint structure $\mathfrak{C}$ and a node d such that*

1. *$(\mathfrak{C}, d) \models \psi$ and*
2. *$\bigcup_{f \in F} \mathsf{im}(f^{\mathfrak{C}})$ is finite, i.e., the functions from F map the elements of the Kripke structure to finitely many elements of $\mathcal{A}$.*

PROOF. The "only-if" part is trivial because every finite model of φ satisfies conditions 1. and 2. For the "if" part let us start with an $\mathcal{A}$-constraint graph $\mathfrak{C}$ with underlying Kripke structure $\mathcal{K} = (D, \to, \rho)$ such that

1. $(\mathfrak{C}, d) \models \psi$ and
2. $\bigcup_{f \in \mathbb{F}_\psi} \mathsf{im}(f^{\mathfrak{C}})$ is finite.

We have to find a finite model of ψ. W.l.o.g. we can assume that every node of D is reachable from d.

Let $B = \bigcup_{f \in \mathbb{F}_\psi} \mathsf{im}(f^{\mathfrak{C}})$. We now define an abstracted ECTL^*-formula ψ^a (without constraints) as follows: First take for all $f \in \mathbb{F}_\psi$ and all $a \in B$ a fresh proposition $[f, a]$. Then we construct from ψ the formula ψ_0 by replacing every occurrence of an atomic constraint $R(f_1 S^{i_1}(x), \ldots, f_k S^{i_k}(x))$ by the ECTL^*-path formula

$$\bigvee_{(a_1,\ldots,a_k)\in I(R)\cap B^k} \bigwedge_{j=1}^{k} [f_j, a_j](S^{i_j}(x)).$$

Finally, we define $\psi^a = \psi_0 \wedge \psi_1$, where ψ_1 is defined as

$$\psi_1 = \mathsf{A}\,\forall x \Big(\bigwedge_{f \in \mathbb{F}_\psi} \bigvee_{a \in B} ([f, a](x) \wedge \bigwedge_{b \in B\setminus\{a\}} \neg[f, b](x))\Big).$$

It states that for every node x that is reachable from the current node and every $f \in \mathbb{F}_\psi$ there is exactly one $a \in B$ such that x is labeled with $[f, a]$.

In a first step, we construct from the model $\mathfrak{C}$ for ψ a Kripke structure $\mathfrak{C}^a$, which is a model of ψ^a. For this, we extend the Kripke structure $\mathcal{K} = (D, \to, \rho)$ to the Kripke structure $\mathfrak{C}^a = (D, \to, \rho^a)$, where

$$\rho^a(e) = \rho(e) \cup \{[f, a] \mid f^{\mathfrak{C}}(e) = a\}.$$

We clearly have $(\mathfrak{C}^a, d) \models \psi_1$. Moreover, a simple induction over the structure of formulas shows that $(\mathfrak{C}^a, d) \models \psi_0$.

Now, ECTL^* has the finite model property. This follows from the facts that (i) ECTL^*-formulas can be translated into equivalent modal μ-calculus formulas [11], and (ii) that the modal μ-calculus has the finite model property [19]. Therefore, there exists a finite Kripke structure $\mathcal{K}' = (D', \to', \rho')$ and $d' \in D'$ such that $(\mathcal{K}', d') \models \psi^a$. W.l.o.g. we can assume that every node of D' is reachable from the node d'.

We finally construct from $\mathcal{K}'$ a finite model $\mathfrak{C}'$ for our original formula ψ. The underlying Kripke structure is $\mathcal{K}'$, where we can remove the new propositions $[f, a]$. For every $f \in \mathbb{F}_\psi$ we define the mapping $f^{\mathfrak{C}'} : D' \to A$ as follows: Let $e \in D'$ and $f \in \mathbb{F}_\psi$. Since e is reachable from d' and $(\mathcal{K}', d') \models \psi_1$ there must exist a unique $a \in B$ such that $[f, a] \in \rho'(e)$. We set $f^{\mathfrak{C}'}(e) = a$.

We also have $(\mathcal{K}', d') \models \psi_0$. A simple induction finally shows that this implies $(\mathfrak{C}', d') \models \psi$. $\square$

Given this characterization we can prove the following result:

Corollary 58. *Let $\mathcal{Z}$ be the σ-structure defined in* (1) *(or one of its expansions from the previous section). Then* $\mathsf{FINSAT}_{\mathsf{ECTL}^*}(\mathcal{Z})$ *is decidable.*

PROOF. Let $F \subseteq \mathbb{F}$ be the set of function symbols appearing in φ, and choose two fresh function symbols $g, h \in \mathbb{F} \setminus F$. Let ψ be defined as the conjunction of the following two formulas:

$$\psi_1 = \mathsf{A}\,\forall x\, (g(x) = g(S(x)) \wedge h(x) = h(S(x)))$$
$$\psi_2 = \mathsf{A}\,\forall x \bigwedge_{f \in F} g(x) \le f(x) \le h(x)$$

It is not hard to see that φ is finitely $\mathcal{Z}$-satisfiable if and only if $(\varphi \wedge \psi)$ is $\mathcal{Z}$-satisfiable: Suppose that $(\mathfrak{C}, d) \models \varphi \wedge \psi$ for a $\mathcal{Z}$-constraint structure $\mathfrak{C}$, where w.l.o.g. every node is reachable from d. Then ψ_1 enforces that $g^{\mathfrak{C}}$ and $h^{\mathfrak{C}}$ are constant and ψ_2 enforces that every integer $z \in \mathbb{Z}$ that belongs to the image of one of the functions $f^{\mathfrak{C}}$ ($f \in F$) belongs to the interval $[g(v), h(v)]$. By Proposition 57, $\varphi \wedge \psi$ has a finite model, which is also a model of φ.

Vice versa, if φ has a finite model $\mathfrak{C}$, then there are integers $c, d \in \mathbb{Z}$ such that $\mathsf{im}(f^{\mathfrak{C}}) \subseteq [c, d]$ for every $f \in F$. We can extend $\mathfrak{C}$ to a model for $\varphi \wedge \psi$ by defining $g^{\mathfrak{C}}(v) = c$ and $h^{\mathfrak{C}}(v) = d$ for every node v of $\mathfrak{G}$.

Since $\mathsf{SAT}_{\mathsf{ECTL}^*}(\mathcal{Z})$ is decidable (Theorem 32) so is $\mathsf{FINSAT}_{\mathsf{ECTL}^*}(\mathcal{Z})$. $\square$

We can use Corollary 58 to show that for every linear order $\mathcal{L}$ (extended with the equality relation), $\mathsf{FINSAT}_{\mathsf{ECTL}^*}(\mathcal{L})$ is decidable:

Corollary 59. *Let $(L, <)$ be a linear order and define $\mathcal{L} = (L, <, \equiv)$ where $\equiv$ is the equality relation on L. Then $\mathsf{FINSAT}_{\mathsf{ECTL}^*}(\mathcal{L})$ can be reduced to $\mathsf{FINSAT}_{\mathsf{ECTL}^*}(\mathcal{Z})$, and is therefore decidable.*

Proof. First assume that L is infinite. Let $\varphi \in \mathsf{ECTL}^*(\{<, \equiv\})$ and let $\mathfrak{C}$ be a finite $\mathcal{Z}$-constraint graph in which φ holds. Choose $a, b \in \mathbb{Z}$ such that $\mathsf{im}(f^{\mathfrak{C}}) \subseteq [a, b]$ for every function symbol f that appears in φ. Let $n = b - a$. Since L is infinite, there exists elements $l_0, \ldots, l_n \in L$ such that $l_0 < l_1 < \cdots < l_n$ in $(L, <)$. Let $\mathfrak{C}'$ be the $\mathcal{L}$-constraint graph with the same underlying Kripke structure as $\mathfrak{C}$ and $f^{\mathfrak{C}'}(d) = l_i$ if $f^{\mathfrak{C}}(d) = a + i$. This is clearly a finite model of φ over the domain $\mathcal{L}$. By reversing the role of $\mathcal{L}$ and $\mathcal{Z}$, we can show that φ is finitely $\mathcal{Z}$-satisfiable if φ is finitely $\mathcal{L}$-satisfiable.

If L is a finite set with $c = |L|$, then we can reduce $\mathsf{FINSAT}_{\mathsf{ECTL}^*}(\mathcal{L})$ again to $\mathsf{FINSAT}_{\mathsf{ECTL}^*}(\mathcal{Z})$ by mapping a formula $\varphi \in \mathsf{ECTL}^*(\{<, \equiv\})$ to $\varphi \wedge \psi$, where ψ is a variant of the formula from the proof of Corollary 58. Using the relations $\equiv_1$ and $\equiv_c$ we have to bound the image of every function $f \in \mathbb{F}$ that appears in φ to the interval $[1, c]$. $\square$

It is open whether there is a linear order for which $\mathsf{SAT}_{\mathsf{ECTL}^*}(\mathcal{L})$ is undecidable.

Remark 60. Instead of using reductions to the satisfiability problem, one can proof all decidability results of this section with the following approach.

Analogously to the definition of $\mathsf{EHD}(\mathcal{L})$ (for a logic $\mathcal{L}$), say that a σ-structure $\mathcal{A}$ has the property $\mathsf{EHD}_{\mathsf{fin}}(\mathcal{L})$ if there is a computable function that maps every finite subsignature $\tau \subseteq \sigma$ to an $\mathcal{L}$-sentence φ_τ such that for every countable τ-structure $\mathcal{B}$ we have the following: There exists a homomorphism $h : \mathcal{B} \to \mathcal{A}$ with finite image if and only if $\mathcal{B} \models \varphi_\tau$.

Then we can follow exactly all the steps relating property $\mathsf{EHD}(\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B}))$ of a structure $\mathcal{A}$ with decidability of $\mathsf{SAT}_{\mathsf{ECTL}^*}(\mathcal{A})$ and obtain a proof that $\mathsf{FINSAT}_{\mathsf{ECTL}^*}(\mathcal{A})$ is decidable for every negation-closed structure $\mathcal{A}$ with property $\mathsf{EHD}_{\mathsf{fin}}(\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B}))$. The results stated above then follow from the fact that every infinite linear order has property $\mathsf{EHD}_{\mathsf{fin}}(\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B}))$: A constraint graph allows a homomorphism with finite image to an infinite linear order if and only if there is a bound on the length of the longest $<$-chain (after contraction of $\equiv$-edges as usual).

8. An Undecidable Extension

$\mathsf{ECTL}^*(\sigma)$ extends ECTL^* with constraints which allow to reason about concrete numerical values. One characteristic of these constraints is that they have a fixed depth: we can compare the values assigned to the register variables at fixed positions, *e.g.*, we can express equality between the value of f_1 at the current position and the value of f_2 at the i^{th} next position along a path using the formula $f_1 x = f_2 S^i(x)$.

Different logics like metric temporal logic (MTL) and timed propositional temporal logic (TPTL), both extensions of linear temporal logic (LTL), can specify properties of data words, which are basically $\mathcal{A}$-constraint paths with only one register variable, where $\mathcal{A}$ is typically the set of natural numbers or real numbers, see [2]. In these logics, one can compare the current data value with future values at arbitrary distance from the current position. For instance, we can express the property that there is a future data value which is equal to the current one with the TPTL-formula $x.\mathsf{F}\,(x = 0)$. It is interesting whether we can add this feature to $\mathsf{ECTL}^*(\sigma)$ and preserve decidability. To this end, we want to extend the atomic constraints from (5) to new ones of the form: $(f_1 O_1(x), \ldots, f_k O_k(x)) \in R$ where $O_i = S^j$ for some $j \in \mathbb{N}$ or $O_i = \mathsf{F}$. Intuitively, $O_i = \mathsf{F}$ would refer to the f_i-value at some (existentially quantified) future position of the path.

On the concrete domain $(\mathbb{Z}, <, =)$, this would allow to express, for instance, the property that there is a future position in which the f_1-value is greater than the f_2-value in the next position: $f_1\mathsf{F}(x) > f_2S(x)$.

Unfortunately we can show that this leads to undecidability of the satisfiability problem, even in very restricted settings: Even if we consider as the starting point logic LTL instead of ECTL^*, adding these new constraints causes undecidability of the satisfiability problem on very simple concrete domains.

Definition 61. $\mathsf{LTL}[\mathsf{F},\mathsf{X}](\sigma)$ is the extension of LTL defined by the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathsf{X}\varphi \mid \varphi \,\mathsf{U}\, \varphi \mid R(f_1O_1, \ldots, f_kO_k)$$

where $p \in \mathbb{P}$, $R \in \sigma$, $k = \mathsf{ar}(R)$, and for all $1 \leq j \leq k$, $f_j \in \mathbb{F}$ and $O_j = \mathsf{X}^{i_j}$ for some $i_j \in \mathbb{N}$ or $O_j = \mathsf{F}$.

$\mathsf{LTL}[\mathsf{F},\mathsf{X}](\sigma)$ is standard LTL extended by atomic constraints as those from $\mathsf{ECTL}^*(\sigma)$ where also future operator may occur inside the constraints. Since we now add our constraints to a temporal logic (instead of MSO on paths) we slightly change the syntax: As usual in temporal logics, we use no variables to point to nodes of the Kripke structure and we use X as the symbol for the successor function, i.e., in our new constraints the term $f_i\mathsf{X}^j$ replaces the $\mathsf{ECTL}^*(\sigma)$-term $f_iS^i(x)$. We use the usual abbreviations, in particular $\mathsf{G}\,\varphi$ (φ holds globally in the future) and $\mathsf{F}\,\varphi$ (φ holds finally in the future).

The semantics of $\mathsf{LTL}[\mathsf{F},\mathsf{X}](\sigma)$ is mostly inherited from that of LTL, but we evaluate a formula on an $\mathcal{A}$-constraint path $\mathfrak{P}$ with underlying Kripke path $\mathcal{P}$ (while LTL is evaluated over Kripke paths). Note that the mappings $f^{\mathfrak{P}}$ and $\mathcal{A}$ only play a role for evaluating constraints: $\mathfrak{P} \models R(f_1O_1, \ldots, f_kO_k)$ if and only if there are $i_1, \ldots, i_k$ such that $(f_1^{\mathfrak{P}}(\mathfrak{P}(i_1)), \ldots, f_k^{\mathfrak{P}}(\mathfrak{P}(i_k))) \in I(R)$ and $i_l = j$ if $O_l = \mathsf{X}^j$ for all $1 \leq l \leq k$ and $j \in \mathbb{N}$.

Theorem 62. *Satisfiability for* $\mathsf{LTL}[\mathsf{F},\mathsf{X}](\{<,\equiv\})$ *over the concrete domains* $(\mathbb{Z}, <, \equiv)$ *and* $(\mathbb{N}, <, \equiv)$ *is undecidable.*

To obtain this result we use *incrementing counter automata*, in short ICAs, first introduced in [14]. In contrast to their definition in [14], we use input-free ICAs, but this does not change things, since we are only interested in the emptiness problem.

Definition 63. An *incrementing counter automaton* (ICA) C with ε-transitions and zero testing is a tuple $C = (Q, q_I, n, \delta, F)$, where:

- Q is a finite set of states,
- $q_I \in Q$ is the initial state,
- $n \in \mathbb{N}$ is the number of counters,
- $\delta \subseteq Q \times L \times Q$ is the transition relation over the instruction set $L = \{\mathsf{inc}_i, \mathsf{dec}_i, \mathsf{ifz}_i \mid 1 \leq i \leq n\}$, and
- $F \subseteq Q$ is the set of accepting states.

A configuration of C is a pair (q, v) where $q \in Q$ and $v : \{1, \ldots, n\} \to \mathbb{N}$ is a *counter valuation*. For configurations (q, v), (q', v'), and an instruction $l \in L$ there is an *exact transition* $(q, v) \xrightarrow{l}_{\dagger} (q', v')$ of C if and only if $(q, l, q') \in \delta$ and one of the following cases holds:

- $l = \mathsf{inc}_i$ for some i, $v(j) = v'(j)$ for $j \neq i$, and $v'(i) = v(i) + 1$
- $l = \mathsf{dec}_i$ for some i, $v(j) = v'(j)$ for $j \neq i$, $v(i) > 0$, and $v'(i) = v(i) - 1$
- $l = \mathsf{ifz}_i$ for some i, $v(i) = 0$, and $v'(j) = v(j)$ for all j.

We define a partial order $\leq$ on counter valuations as follows: $v \leq w$ if and only if $v(i) \leq w(i)$ for all $1 \leq i \leq n$. The transitions of C are of the form $(q,w) \xrightarrow{l} (q',w')$ such that there are v, v' with an exact transition $(q,v) \xrightarrow{l}_{\dagger} (q',v')$, $w \leq v$, and $v' \leq w'$.

An infinite run of C is an infinite sequence of transitions $(q_0, v_0) \xrightarrow{l_0} (q_1, v_1) \xrightarrow{l_1} \cdots$ such that $q_0 = q_I$. An infinite run is accepting if and only if some accepting state occurs infinitely often.

Essentially, ICAs relax the conditions on transitions, by letting faulty increments occur at any time. The problem whether an ICA admits an accepting run is deeply connected to that of the halting problem (for finite runs) and of the recurring state problem (for infinite runs) of *insertion channel machines with emptiness testing*, see [24]. Their computational power is strictly weaker than that of perfect channel machines, but emptiness is still undecidable on infinite words, which makes them a useful tool for undecidability proofs.

Theorem 64 (see Theorem 2.9b of [14]). *The existence of an infinite accepting run for ICAs is undecidable and Π^0_1-complete.*

To prove undecidability of the satisfiability problem for $\mathsf{LTL}[\mathsf{F},\mathsf{X}](\{<,\equiv\})$ over $(\mathbb{Z},<,\equiv)$, we use a reduction (for the method we drew inspiration from [14]) from the infinite accepting run problem for ICAs.

Proof (Theorem 62). Let $C = (Q, q_I, n, \delta, F)$ be an ICA. We define an $\mathsf{LTL}[\mathsf{F},\mathsf{X}](\{<,\equiv\})$-formula φ_C on the atomic proposition set $\mathbb{P} = Q \cup L$ where $L = \{\mathsf{inc}_i, \mathsf{dec}_i, \mathsf{ifz}_i \mid 1 \leq i \leq n\}$, which is satisfiable over the concrete domain $\mathcal{A} = (\mathbb{Z},<,\equiv)$ (or $\mathcal{A} = (\mathbb{N},<,\equiv)$) if and only if C has an infinite accepting run.

To encode a successful run of C, we require that an $\mathcal{A}$-constraint path $\mathfrak{P}$ satisfies the properties below:

- In each position of the path $\mathfrak{P}$, one and only one state from Q occurs, and one and only one operation from L occurs:
$$\varphi_{\text{struct}} := \mathsf{G}\Big(\bigvee_{q\in Q} q \wedge \bigvee_{l\in L} l \wedge \bigwedge_{\substack{q,q'\in Q\\ q\neq q'}} (q \to \neg q') \wedge \bigwedge_{\substack{l,l'\in L\\ l\neq l'}} (l \to \neg l')\Big).$$
- The computation starts with the initial state and reaches a final state infinitely often:
$$\varphi_{\text{Büchi}} := q_I \wedge \bigvee_{q\in F} \mathsf{G}\,\mathsf{F}\, q.$$
- The transition relations of C are encoded in the following way:
$$\varphi_{\text{trans}} := \mathsf{G} \bigwedge_{q\in Q} \Big(q \to \bigvee_{(q,l,q')\in\delta} (l \wedge \mathsf{X}q')\Big).$$
- We fix $2n$ pairwise different function symbols $f_i, g_i \in \mathbb{F}$ for $1 \leq i \leq n$. We use their interpretations to identify each inc_i-operation and dec_i-operation, respectively. While the identifiers are assigned univocally for the increment instructions, more than one decrement can have the same identifier value. To make sure that each inc-operation on counter i is assigned a unique value, we require that at every position of the path $\mathfrak{P}$, which corresponds to an inc_i-operation, $f_i^{\mathfrak{P}}$ is assigned a strictly greater value than in the previous position, and otherwise remains constant.

  For the sequence of values of $g_i^{\mathfrak{P}}$ we only require that it stays constant whenever the instruction dec_i does not occur, and it is otherwise non-decreasing:
$$\varphi_{\mathsf{inc}} := \mathsf{G} \bigwedge_{i=1}^{n} \Big((\mathsf{inc}_i \to f_i < f_i\mathsf{X}) \wedge (\neg\mathsf{inc}_i \to f_i = f_i\mathsf{X})\Big)$$
$$\varphi_{\mathsf{dec}} := \mathsf{G} \bigwedge_{i=1}^{n} \Big((g_i \leq g_i\mathsf{X}) \wedge (\neg\mathsf{dec}_i \to g_i = g_i\mathsf{X})\Big).$$

- Whenever a zero test on counter i occurs, the counter should be empty. To make sure that a run respects this property, we should check that, for each increase on counter i, we can find *at least* a corresponding decrease. It is not necessary that this correspondence is exact, since a faulty increase can occur at any time, making additional decreases possible. We use the identifier functions f_i and g_i to match each inc_i, which is eventually followed by a ifz_i, to a dec_i with the same identifier:

$$\varphi_{\mathsf{ifz1}} := \mathsf{G} \bigwedge_{i=1}^{n} \Big((\mathsf{inc}_i \wedge \mathsf{F}\, \mathsf{ifz}_i) \rightarrow (f_i \mathsf{X} = g_i \mathsf{F}\,) \Big).$$

  We should also enforce the fact that, for each inc_i, the correspondent dec_i occurs after inc_i and before ifz_i. For this we require that g_i is never assigned a higher value than f_i, and that they coincide in the occurrence of a zero test instruction on counter i. Since g_i cannot decrease, this means that any dec_i with the same value of an inc_i-instruction should happen before the zero test:

$$\varphi_{\mathsf{ifz2}} := \mathsf{G} \bigwedge_{i=1}^{n} (f_i \geq g_i \wedge (\mathsf{ifz}_i \rightarrow f_i = g_i)).$$

Let φ_C be the conjunction of all the above formulas. We prove the following equivalence:

$$C \text{ has an accepting run} \quad \Longleftrightarrow \quad \varphi_C \text{ is satisfiable}$$

Proof of $\Longrightarrow$. Let $r = (q_0, v_0) \xrightarrow{l_0} (q_1, v_1) \xrightarrow{l_1} \cdots$ be a successful run of C. Starting from this we define an $\mathcal{A}$-constraint path $\mathfrak{P} = (\mathcal{A}, \mathcal{P}, \gamma)$ which satisfies φ_C, where $\mathcal{A}$ can be $(\mathbb{N}, <, \equiv)$ or $(\mathbb{Z}, <, \equiv)$.

First of all we define $\mathcal{P} = (\mathbb{N}, \rightarrow, \rho)$, where $\rightarrow$ is the successor relation on the natural numbers, and $\rho(i) = \{q_i, l_i\}$ for all $i \in \mathbb{N}$. Since the run is successful, this ensures that $\varphi_{\text{struct}} \wedge \varphi_{\text{Büchi}} \wedge \varphi_{\mathsf{trans}}$ is satisfied.

Now we define the interpretations of f_i and g_i. For all $1 \leq i \leq n$, we define $f_i(0) = g_i(0) = 0$. For all other nodes $j \geq 1$ we define

$$f_i(j) = \begin{cases} f_i(j-1) + 1 & \text{iff } l_{j-1} = \mathsf{inc}_i \\ f_i(j-1) & \text{otherwise} \end{cases}$$

$$g_i(j) = \begin{cases} g_i(j-1) + 1 & \text{iff } l_{j-1} = \mathsf{dec}_i \text{ and } g_i(j-1) < f_i(j-1) \\ g_i(j-1) & \text{otherwise} \end{cases}$$

Clearly these choices of f_i and g_i make φ_{inc} and φ_{dec} true. To prove that also φ_{ifz1} and φ_{ifz2} hold, we note that, since r is a successful run of C, a transition with operation ifz_i can only occur if counter i is empty. Therefore, the number of increase instructions on counter i, between any two ifz_i, should be matched by an equal or greater number of decrease instructions. By definition of the functions, for each increase on the value of f_i which is eventually followed by a zero test on counter i, there is a corresponding increase on the value of g_i. Furthermore, whenever g_i reaches the value of f_i, the value of g_i is no longer increased until f_i grows again, thus ensuring that $\varphi_{\mathsf{ifz1}} \wedge \varphi_{\mathsf{ifz2}}$ holds.

Proof of $\Longleftarrow$. Let $\mathfrak{P} = (\mathcal{A}, \mathcal{P}, \gamma)$ be a constraint path such that $\mathfrak{P} \models \varphi_C$. $\mathcal{A}$ can be $(\mathbb{N}, <, \equiv)$ or $(\mathbb{Z}, <, \equiv)$, this does not change the proof. We define a run

$$r = (q_0, v_0) \xrightarrow{l_0} (q_1, v_1) \xrightarrow{l_1} \cdots$$

of C and prove that it is accepting. By $\varphi_{\text{struct}} \wedge \varphi_{\text{Büchi}}$ the label $\rho(\mathfrak{P}(i))$ of every node of the path $\mathfrak{P}$ contains one and only one symbol q from Q and l from L. We set $q_i = q$ and $l_i = l$. Since $\varphi_{\text{Büchi}}$ holds, q_0 is the initial state q_I, and ab accepting state is visited infinitely often. Since φ_{trans} holds, for every $i \in \mathbb{N}$ we have that $(q_i, l_i, q_{i+1}) \in \delta$. We set to zero the initial value of every counter $1 \leq j \leq n$: $v_0(j) = 0$. For all later positions $i \geq 1$ we define:

$$v_i(j) = \begin{cases} v_{i-1}(j) + 1 & \text{iff } l_{i-1} = \mathsf{inc}_j \\ v_{i-1}(j) - 1 & \text{iff } l_{i-1} = \mathsf{dec}_j \text{ and } v_{i-1}(j) > 0 \\ v_{i-1}(j) & \text{otherwise.} \end{cases}$$

Note that $v_i(j)$ is always positive. It remains to show that

$$(q_i, v_i) \xrightarrow{l_i} (q_{i+1}, v_{i+1}) \tag{12}$$

according to Definition 63. We only discuss the non trivial cases.

- If $l_i = \mathsf{dec}_j$ and $v_i(j) = 0$, then also $v_{i+1}(j) = 0$. Let v' be the counter valuation that assigns $v'(j) = 1$ and coincides with v_i on all other counters. Then, $(q_i, v') \xrightarrow{\mathsf{dec}_j}_{\dagger} (q_{i+1}, v_{i+1})$ is an exact transition. Since $v_i \leq v'$, we get (12).
- If $l_i = \mathsf{ifz}_j$, then we need to show $v_i(j) = 0$ in order to get (12). For this to hold, it is enough to notice that $\varphi_{\mathsf{ifz}1}$ and $\varphi_{\mathsf{ifz}2}$ ensure that for every inc_j followed by a ifz_j there is a dec_j, and this occurs before ifz_j. Hence, every time we increase $v_k(j)$ by one for some $k < i$, we also decrease it by one before the zero test. All other decreases do not alter the value of the counter.

We conclude that, since the infinite accepting run problem for ICAs is undecidable and Π^0_1-complete, satisfiability for $\mathsf{LTL}[\mathsf{F},\mathsf{X}](\{\equiv,<\})$ over $(\mathbb{N},<,\equiv)$ and $(\mathbb{Z},<,\equiv)$ is also undecidable and Π^0_1-hard. □

Obviously the undecidability result also applies to CTL^* and ECTL^* extended with this new kind of constraints. We currently do not know the decidability status for LTL extended by constraints where we only allow the F operator and not the X operator.

9. Open problems

As already mentioned in the introduction, it remains open to determine the complexity of CTL^*-satisfiability with constraints over $\mathcal{Z}$ (for ECTL^* with constraints over $\mathcal{Z}$ satisfiability is non-elementary). Clearly, this problem is $2\mathsf{EXPTIME}$-hard due to the known lower bound for CTL^*-satisfiability. To get an upper complexity bound, one should investigate the complexity of the emptiness problem for puzzles from [3] (see Lemma 5).

An interesting structure for which the decidability status for satisfiability of ECTL^* and even LTL with constraints is open, is $(\{0,1\}^*, \leq_p, \not\leq_p)$, where $\leq_p$ is the prefix order on words, and $\not\leq_p$ is its complement. We recently proved that this structure does not have the property $\mathsf{EHD}(\mathsf{Bool}(\mathsf{MSO}, \mathsf{WMSO{+}B}))$. Hence, a new strategy is needed.

Acknowledgments. We are grateful to Szymon Toruńczyk for fruitful discussions.

[1] R. Alur and T. A. Henzinger. A really temporal logic. In *Proc. FOCS 1989*, pages 164–169. IEEE Computer Society, 1989.

[2] R. Alur and T. Henzinger. Real-time logics: complexity and expressiveness. In *Information and Computation*, vol. 104, 390–401, 1993.

[3] M. Bojańczyk and S. Toruńczyk. Weak MSO+U over infinite trees. In *Proc. STACS 2012*, vol. 14 of *LIPIcs*, 648–660. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2012.

[4] M. Bojańczyk and S. Toruńczyk. Weak MSO+U over infinite trees (long version). available at `http://www.mimuw.edu.pl/~bojan/papers/wmsou-trees.pdf`

[5] L. Bozzelli and R. Gascon. Branching-time temporal logic extended with qualitative Presburger constraints. In *Proc. LPAR 2006*, LNCS 4246, 197–211. Springer, 2006.

[6] C. Carapelle, A. Kartzow, and M. Lohrey. Satisfiability of CTL* with constraints. In *Proc. CONCUR 2013*, LNCS 8052, pages 455–469. Springer, 2013.

[7] K. Čerāns. Deciding properties of integral relational automata. In *Proc. ICALP 1994*, LNCS 820, 820:35–46. Springer, 1994.

[8] T. Colcombet and C. Löding. Regular cost functions over finite trees. In *Proc. LICS 2010*, 70–79. IEEE Computer Society, 2010.

[9] B. Courcelle. Monadic second-order definable graph transductions: a survey *Theor. Comput. Sci.*, 126:53–75, 1994.

[10] B. Courcelle. The monadic second-order logic of graphs V: On closing the gap between definability and recognizability. *Theor. Comput. Sci.*, 80(2):153–202, 1991.

[11] M. Dam. CTL* and ECTL* as fragments of the modal mu-calculus. *Theor. Comput. Sci.*, 126(1):77–96, 1994.

[12] S. Demri and D. D'Souza. An automata-theoretic approach to constraint LTL. *Inf. Comput.*, 205(3):380–415, 2007.

[13] S. Demri and R. Gascon. Verification of qualitative $\mathbb{Z}$ constraints. *Theor. Comput. Sci.*, 409(1):24–40, 2008.

[14] S. Demri and R. Lazić. LTL with the freeze quantifier and register automata. *ACM Trans. Comput. Logic*, 10(3), 16:1–16:30, 2009.
[15] E. A. Emerson and C. S. Jutla. The complexity of tree automata and logics of programs. *SIAM Journal on Computing*, 29(1):132–158, 1999.
[16] R. Gascon. An automata-based approach for CTL* with constraints. *Electr. Notes Theor. Comput. Sci.*, 239:193–211, 2009.
[17] D. Janin and I. Walukiewicz On the Expressive Completeness of the Propositional mu-Calculus with Respect to Monadic Second Order Logic. In *Proc. CONCUR 1996*, LNCS 1119, 263–277. Springer, 1996.
[18] R. Koymans. Specifying real-time properties with metric temporal logic. *Real-Time Systems*, 2(4):255–299, 1990.
[19] D. Kozen A finite model theorem for the propositional μ-calculus. *Studia Logica* 47(3):233–241, 1988.
[20] C. Lutz. Description logics with concrete domains-a survey. In *Advances in Modal Logic 4*, pages 265–296. King's College Publications, 2003.
[21] C. Lutz. Combining interval-based temporal reasoning with general TBoxes. *Artificial Intelligence*, 152(2):235 – 274, 2004.
[22] C. Lutz. NEXPTIME-complete description logics with concrete domains. *ACM Trans. Comput. Log.*, 5(4):669–705, 2004.
[23] C. Lutz and M. Milicic. A tableau algorithm for description logics with concrete domains and general TBoxes. *J. Autom. Reasoning*, 38(1-3):227–259, 2007.
[24] J. Ouaknine and J. Worrell. On metric temporal logic and faulty Turing machines. *Proc. FOSSACS 2006*, LNCS 3921, pages, 217–230. Springer, 2006.
[25] M. O. Rabin. Decidability of second-order theories and automata on infinite trees. *Trans. Amer. Math. Soc.*, 141:1–35, 1969.
[26] W. Thomas. Computation tree logic and regular omega-languages. In *Proc. REX Workshop 1988*, LNCS 354, 690–713. Springer, 1988.
[27] M. Y. Vardi and P. Wolper. Yet another process logic (preliminary version). In *Proc. Logic of Programs 1983*, LNCS 164, 501–512. Springer, 1983.
[28] I. Walukiewicz. Monadic second-order logic on tree-like structures *Theor. Comput. Sci.*, 275(1-2):311–346, 2002.
[29] P. Wolper. Temporal logic can be more expressive. In *Information and Control*, 56, 72–99, 1983.