Decidable Horn Systems with Difference Constraints Arithmetic

Radu Iosif

VERIMAG/CNRS, Grenoble, France
iosif@imag.fr

Abstract. This paper tackles the problem of the existence of solutions for recursive systems of Horn clauses with second-order variables interpreted as integer relations, and harnessed by a simple first-order theory, such as difference bounds arithmetic. We start by the definition of a simple class of Horn systems with one second-order variable and one non-linear recursive rule, for which we prove the decidability of the problem "does the system has a solution ?". The proof relies on a construction of a tree automaton recognizing all cycles in the weighted graph corresponding to every unfolding tree of the Horn system. We constrain the tree to recognize only cycles of negative weight by adding a Presburger formula that harnesses the number of times each rule is fired, and reduce our problem to the universality of a Presburger-constrained tree automaton. We studied the complexity of this problem and found it to be in 2EXPTIME with a EXPTIME-hard lower bound. In the second part, we drop the univariate restriction and consider multivariate second-order Horn systems with a structural restriction, called flatness. Finally, we show the decidability of the more general class of systems, within the same complexity bounds.

1 Introduction

Systems of Horn clauses, called *Horn systems*, for short, play a central role in constraint logic programming [10], being the reference syntax of declarative programming languages, such as CLP [9] and PROLOG [4]. In particular, Horn systems with linear integer constraints are used to model geometric and, in general, combinatorial problems [16]. Quite recently, (extended) Horn systems have been used to model imperative program semantics together with proof rules for verification of safety, termination and branching temporal logic properties [8, 1]. This paper contributes to the mentioned areas by defining decidable classes of Horn systems and studying the complexity of their associated decision problems.

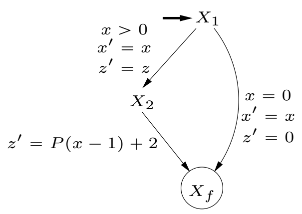
To start with, we consider the recursive procedure in Fig. 1 (a). The small-step semantics of this procedure is captured by the control-flow graph given in Fig. 1 (b), whose nodes represent control locations (in our case X_1, X_2 and X_f, with X_1 and X_f designated *initial* and *final* locations) and edges are labeled with first-order arithmetic formulæ denoting the program semantics (primed variables x' and z' denote the values at the next step). For instance, the edge $X_1 \xrightarrow{x>0 \wedge x'=x \wedge z'=z} X_2$

models the true branch of the if statement, $X_1 \xrightarrow{x=0 \wedge x'=x \wedge z'=0} X_f$ corresponds to the false branch, and $X_2 \xrightarrow{z'=P(x-1)+2} X_f$ is the recursive call.

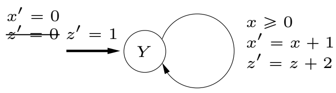
The semantics of each edge in the control-flow graph in Fig. 1 (b) is captured by an equivalent Horn clause, where first-order variables range over integers, and the second-order predicates $X_{1,2,f}(x, z, x', z')$ denote integer relations of arity four. For instance, the last clause in Fig. 1 (c) describes the flow of data values along the call edge $X_2 \xrightarrow{z'=P(x-1)+2} X_f$ from Fig. 1 (b). Intuitively, the constraint $x'' = x' - 1$ models the (transfer of the) parameter values, $z^{\mathrm{iv}} = z''' + 2$ models the return, and $x^{\mathrm{iv}} = x'$ is the frame condition ensuring that the value of the local variable x does not change across the recursive call. With this in mind, Fig. 1 (c) shows the Horn system whose least solution gives, for each control location modeled by a second-order variable $X \in \{X_1, X_2, X_f\}$, the (summary) relation between the values of the first-order program variables $\mathbf{x} = \{x, z\}$ at the initial location X_1, and the values at location X. We denote by $\mathbf{x}'$ the set $\{x' \mid x \in \mathbf{x}\}$, and the same for $\mathbf{x}''$, $\mathbf{x}'''$ and $\mathbf{x}^{\mathrm{iv}}$, respectively.

```
int P(int x)
int z;
assume(x ⩾ 0);
if (x > 0)
  z := P(x-1) + 2;
else
  z := 0
return z;
```

(a)



(b)



$x' = 0 \wedge \cancel{z' = 0}\ z' = 1 \rightarrow Y(\mathbf{x}')$

$Y(\mathbf{x}) \wedge x \geqslant 0 \wedge x' = x + 1 \wedge z' = z + 2 \rightarrow Y(\mathbf{x}')$

(d)

$$
\begin{array}{ll}
x' = x \wedge z' = z & \rightarrow X_1(\mathbf{x}, \mathbf{x}') \\
X_1(\mathbf{x}, \mathbf{x}') \wedge x' > 0 \wedge x'' = x' \wedge z'' = z' & \rightarrow X_2(\mathbf{x}, \mathbf{x}'') \\
X_1(\mathbf{x}, \mathbf{x}') \wedge x' = 0 \wedge z'' = 0 \wedge x'' = x' & \rightarrow X_f(\mathbf{x}, \mathbf{x}'') \\
X_2(\mathbf{x}, \mathbf{x}') \wedge x'' = x' - 1 \wedge X_f(\mathbf{x}'', \mathbf{x}''') \wedge z^{\mathrm{iv}} = z''' + 2 \wedge x^{\mathrm{iv}} = x' & \rightarrow X_f(\mathbf{x}, \mathbf{x}^{\mathrm{iv}})
\end{array}
$$

(c)

Fig. 1. Recursive Programs as Horn Systems

A possible verification condition for the program in Fig. 1 (a) is to show that the relation associated with the final location X_f is the same (modulo projection of the first two entries corresponding to the variables x and z) as the least invariant of the intraprocedural program in Fig. 1 (d) with initial condition $x' = 0 \wedge z' = 0$, namely the relation $\{(n, 2n) \mid n \geqslant 0\}$. In order to verify this property, one would have to compute the least solution of a Horn system. However, even for very simple systems, defining the least solution is beyond the expressive power of classical decidable fragments of first-order arithmetic, such as Presburger arithmetic[1]. To circumvent this problem, we consider a slight modification of the program from Fig. 1 (d), obtained by replacing the initial constraint $z' = 0$ (struck through) with $z' = 1$ (the same for the first Horn clause

[1] See Appendix A for such an example.

in Fig. 1 (d)). The verification condition asks now that the two relations are disjoint, which can be easily encoded using the clause $X_f(\mathbf{x}, \mathbf{x}') \wedge Y(\mathbf{x}') \rightarrow \bot$.

The Horn systems studied in this paper encompass this example. We consider systems of non-linear Horn clauses, of branching degree two or more, with arithmetic formulæ that are conjunctions of *difference constraints* $x - y \leqslant c$, where c is an integer constant. For these systems we ask the question whether a solution (i.e. an assignment of second-order variables to relations) exists. Even though, in general this problem is undecidable[2], we define a non-trivial fragment for which the problem is decidable in 2EXPTIME. Incidentally, the verification condition for the example Fig. 1 fits within our decidable fragment. So far we have not found a matching lower bound, nevertheless, we provide an EXPTIME-hardness result for this class of Horn systems.

The paper is organized as follows. Section 2 gives the preliminary definitions of the syntax and semantics of the Horn systems considered in this paper and Section 3 introduces Presburger-harnessed tree automata, a reasoning tool needed in the main proof (Thm. 1). The core of the method is first explained on a simple class with only one second-order variable and one non-linear recursive rule, called $\mathcal{B}_k$ (Section 4). The decidable class of Horn systems, that constitutes the main contribution of this paper, is defined in Section 5, as a generalization of the $\mathcal{B}_k$ class to more than one second-order variables and more than one non-linear recursive clause. All proofs are given in Appendix B.

Related Work. The complexity of the tuple recognition problem "*given a tuple of values* $\mathbf{v}$*, does* $\mathbf{v}$ *belong to the least fixpoint of the program ?*" for constraint logic programs has been investigated in the seminal work of Revesz [14]. In this work, as in the most literature on logic programming, the complexity is evaluated in terms of the size of the database only (data complexity), instead of the whole program. For *gap-order constraints* i.e. conjunctions of atoms of the form $x - y \leqslant c$, with c a *non-negative* integer (rational) constant, the tuple recognition problem has been found to be in EXPTIME in the branching degree of the system. This result was further generalized to several fragments of linear constraints [13], where the term recognition problem is found to be in PTIME if the coefficients of the linear constraints are either all positive (with upper bounds only) or all negative (with lower bound only), and is EXPTIME-complete for half addition constraints of the form $\pm x \pm y \geqslant c$, with c a non-negative constant. Along this line, the result of Demri *et.al* [7] shows that the coverage problem for branching vector addition systems (BVAS) is complete for 2EXPTIME. An interpretation of their result in terms of constraint logic programs is immediate.

The problem considered in this paper is slightly more general: we want to prove that a given system of Horn clauses (equivalently, a constraint logic program) has a solution that excludes a certain set of tuples. Since we consider difference bounds instead of gap order (half addition, or positive) constraints, i.e. $x - y \leqslant c$, where *c can be negative*, the problem is undecidable, if no restrictions are added. We show that the emptiness problem in which every second-order variable can be the

[2] By reduction from the reachability problem for 2-counter machines [11].

root of at most one unfolding cycle is in 2EXPTIME, and, surprisingly, EXPTIME-hardness occurs even when we limit the number of second-order variables (and, implicitly, of cycles) to one.

Several complexity results were also found for the emptiness problem for *linear* recursive Horn systems, i.e. with branching degree of one: PSPACE-completness for gap-order constraints and unrestricted contol structure [3], NP-completness for *flat control* with octagonal constraints [2] and with vector addition updates and affine guards [6]. The exponential blowup in our case comes from the fact that we consider non-linear recursive systems, of branching degree at least two. For *recursion-free* Horn systems with integer linear constraints, the emptiness problem has been found to be complete for co-NEXPTIME [15].

2 Horn Systems

Syntax. Let $\mathbf{x}$ be a finite set of *first-order variables*, ranging over the set of integers $\mathbb{Z}$. A *difference constraint* is a linear inequality of the form $x \geqslant c$, $x \leqslant c$, or $x - y \leqslant c$, where $x, y \in \mathbf{x}$ and $c \in \mathbb{Z}$ is an integer constant. A finite conjunction of difference constraints is called a *difference bound matrix* (DBM) in the following.

Let $\mathcal{X}$ be a finite set of *second-order variables*, where for each $X \in \mathcal{X}$, we denote by $\#(X) > 0$ the *arity* of X. A *predicate* is a term of the form $X(x_1, \ldots, x_{\#(X)})$, where $X \in \mathcal{X}$ and $x_1, \ldots, x_{\#(X)} \in \mathbf{x}$. A *difference formula* is a possibly quantified boolean combination of difference constraints and predicates. We denote by $FV(\psi)$ the set of first-order variables not occurring under the scope of a quantifier in ψ. We denote by $\perp$ the empty disjunction (false), and by $\top$ the empty conjunction (true). A *Horn clause* is a difference formula of the form:

$$C : \forall \mathbf{x} \forall \mathbf{x}^0 \ldots \forall \mathbf{x}^\ell \,.\, \phi(\mathbf{x}, \mathbf{x}^0, \ldots, \mathbf{x}^\ell) \wedge X_0(\mathbf{x}^0) \wedge \ldots \wedge X_\ell(\mathbf{x}^\ell) \rightarrow h(\mathbf{x}) \ .$$

where ϕ is a DBM, $X_0, \ldots, X_\ell \in \mathcal{X}$ and h is either a predicate or a DBM, denoted as $head(C)$. We say that C has *branching degree* $\ell + 1$ in this case. Throughout this paper we consider sets of variables indexed by natural numbers (and, later on, sequences thereof), and write $\mathbf{x}^i$ for the set $\{x^i \mid x \in \mathbf{x}\}$, where $i \in \mathbb{N}$. In general, we do not write explicitly the universal quantifier prefix for the Horn clauses, and assume that every variable in C is implicitly universally quantified.

A *Horn system* (HS) is a finite set of Horn clauses. A HS $\mathcal{H}$ is said to be *linear* if every right-hand side of a clause in $\mathcal{H}$ contains at most one predicate, and *non-linear* otherwise. A clause C is said to be *rooted* if $head(C)$ is a DBM, and a HS is rooted if it contains at least one rooted clause.

Semantics. Let $\mathcal{H} = \{C_1, \ldots, C_m\}$ be a HS and let $\mathbf{x} = \{x_1, \ldots, x_n\}$ for the rest of this paragraph. A *first-order valuation* (fo-valuation, for short) is a function $\nu : \mathbf{x} \rightarrow \mathbb{Z}$, and by $\mathbb{Z}^{\mathbf{x}}$ we denote the set of fo-valuations with domain $\mathbf{x}$.

A fo-valuation $\nu \in \mathbb{Z}^{\mathbf{x}}$ is a *model* of a DBM ϕ, with $FV(\phi) \subseteq \mathbf{x}$, denoted $\nu \models \phi$, if and only if the formula obtained by replacing each variable $x \in \mathbf{x}$

with $\nu(x)$ is valid according to the semantics of first order arithmetic. Let $[\![\phi]\!] = \{\nu \in \mathbb{Z}^{\mathbf{x}} \mid \nu \models \phi\}$ denote the set of models of ϕ.

A *second-order valuation* (so-valuation, for short) is a function $\sigma : \mathcal{X} \to 2^{\bigcup_{i=1}^{\infty} \mathbb{Z}^i}$ assigning second-order variables to relations, such that $\sigma(X) \subseteq \mathbb{Z}^{\#(X)}$, for all $X \in \mathcal{X}$, i.e. the so-valuation of X is compatible with its arity. For a predicate $P = X(x_1, \ldots, x_{\#(X)})$, we define $[\![P]\!]_\sigma = \{\nu \in \mathbb{Z}^{\mathbf{x}} \mid \langle \nu(x_1), \ldots, \nu(x_{\#(X)})\rangle \in \sigma(X)\}$. In the following we abuse notation and write $[\![\psi]\!]_\sigma$ for $[\![\psi]\!]$, when ψ is a DBM.

A *solution* of the Horn system $\mathcal{H}$ is a so-valuation σ, such that, for each clause $\phi \wedge P_0 \wedge \ldots \wedge P_\ell \to h$ of $\mathcal{H}$ we have: $[\![\phi]\!]_\sigma \cap [\![P_0]\!]_\sigma \cap \ldots \cap [\![P_\ell]\!]_\sigma \subseteq [\![h]\!]_\sigma$. The set of solutions of $\mathcal{H}$ is denoted by $[\![\mathcal{H}]\!]$.

A rooted HS can be equivalently rewritten such that, for each rooted clause C, we have $head(C) = \bot$ — each clause $C : \ \phi(\mathbf{x}, \mathbf{x}^0, \ldots, \mathbf{x}^\ell) \wedge X_0(\mathbf{x}^0) \wedge \ldots \wedge X_\ell(\mathbf{x}^\ell) \to \psi(\mathbf{x})$ is equivalent to a set of clauses of this form[3]. From now on, we will consider only HS $\mathcal{H}$ such that $head(C) = \bot$, for each rooted clause C of $\mathcal{H}$.

Unfolding Trees. Let $\mathbb{N}^*$ be the set of sequences of positive integers. We denote by ϵ the empty sequence and by $p.q$ the concatenation of two sequences $p, q \in \mathbb{N}^*$. We say that q is a *prefix* of p, denoted $q \preceq p$ if $p = q.r$, for some sequence $r \in \mathbb{N}^*$. A *prefix-closed* set S has the property that, for all $p \in S$, $q \preceq p$ implies $q \in S$.

A *ranked alphabet* is a countable set of symbols Σ with an associated arity function $\#(\sigma) \geqslant 0$, for all $\sigma \in \Sigma$. A *tree* is a finite partial function $t : \mathbb{N}^* \rightharpoonup_{fin} \Sigma$, whose domain, denoted $dom(t)$, is a finite prefix-closed subset of $\mathbb{N}^*$. For each position $p \in dom(t)$, a position $p.i \in dom(t)$ is called a *child* of p, for some $i \in \mathbb{N}$. The set of children of a position $p \in dom(t)$ in a tree t is always $\{p.0, \ldots, p.(\#(t(p)) - 1)\}$. Let $\mathrm{Fr}(t) = \{p \in dom(t) \mid \forall i \in \mathbb{N} \ . \ p.i \notin dom(t)\}$ be the set of *leaves* (frontier) of the tree t. For a position $p \in dom(t)$, we denote by $t_{|p}$ the subtree of t rooted at p, where $t_{|p}(q) = t(p.q)$, for all $q \in \mathbb{N}^*$.

Let Σ_{HS} be the ranked alphabet of pairs $\langle \psi, X\rangle$, where $\psi(\mathbf{x}, \mathbf{x}^0, \ldots, \mathbf{x}^\ell)$ is a DBM and $X \in \mathcal{X} \cup \{\bot\}$, whose arity is $\#(\langle \psi, X\rangle) = \ell + 1$. Given $\sigma = \langle \psi, X\rangle \in \Sigma_{HS}$, we write $[\sigma]_1$ for ψ, and $[\sigma]_2$ for X.

Definition 1. *An* unfolding tree *of a HS $\mathcal{H}$ is a tree $t : \mathbb{N}^* \rightharpoonup_{fin} \Sigma_{HS}$, where:*
(a) for each $p \in Fr(t)$, we have $t(p) = \langle \phi, X\rangle$ only if $\phi(\mathbf{x}) \to X(\mathbf{x}) \in \mathcal{H}$,
(b) for each $p \in dom(t) \backslash Fr(t)$, we have $t(p) = \langle \phi, X\rangle$ only if

$$\phi(\mathbf{x}, \mathbf{x}^0, \ldots, \mathbf{x}^\ell) \wedge X_0(\mathbf{x}^0) \wedge \ldots \wedge X_\ell(\mathbf{x}^\ell) \to X(\mathbf{x}) \in \mathcal{H}$$

where $[t(p.i)]_2 = X_i$, for all $i = 0, \ldots, \ell$,

[3] The negation of every DBM $\psi = \bigwedge_{i=1}^{p} \delta_i$, where δ_i are difference constraints, is the disjunction $\bigvee_{i=1}^{p} \neg\delta_i$, where each $\neg\delta_i$ is again a difference constraint, e.g. $\neg x - y \leqslant c \leftrightarrow y - x \leqslant -c - 1$. We obtain:

$$[\phi \wedge X_0 \wedge \ldots \wedge X_\ell \to \psi] \Leftrightarrow \bigwedge_{i=1}^{p} [\phi \wedge \neg\delta_i \wedge X_0 \wedge \ldots \wedge X_\ell \to \bot] \ .$$

where φ_i denotes the DBM $\phi \wedge \neg\delta_i$.

(c) if $\mathcal{H}$ is rooted, we have $t(\epsilon) = \langle \phi, \bot \rangle$ and there exists a clause:

$$\phi(\mathbf{x}, \mathbf{x}^0, \ldots, \mathbf{x}^\ell) \wedge X_0(\mathbf{x}^0) \wedge \ldots \wedge X_\ell(\mathbf{x}^\ell) \to \bot \in \mathcal{H}$$

where $[t(p.i)]_2 = X_i$, for all $i = 0, \ldots, \ell$.

We denote by $\mathcal{T}_X(\mathcal{H})$ the set of unfolding trees of the HS $\mathcal{H}$ such that $[t(\epsilon)]_2 = X$. If $\mathcal{H}$ is rooted, we write $\mathcal{T}(\mathcal{H})$ instead of $\mathcal{T}_\bot(\mathcal{H})$.

If $\psi(\mathbf{x}, \mathbf{x}^0, \ldots, \mathbf{x}^\ell)$ is a DBM and $p \in \mathbb{N}^*$ is a tree position, let ψ^p be the formula $\psi[\mathbf{x}^p/\mathbf{x}, \mathbf{x}^{p.0}/\mathbf{x}^0, \ldots, \mathbf{x}^{p.\ell}/\mathbf{x}^\ell]$. The *characteristic formula* $\Phi(t) = \bigwedge_{p \in dom(t)} [t(p)]_1^p$ of an unfolding tree t is the conjunction of all DBMs (whose variables are indexed by the current position) occurring within the labels of t.

It is customary to represent a DBM $\phi(\mathbf{x})$ as a weighted *constraint graph* $\mathcal{G}_\phi = \langle \mathbf{x}, \rightarrow \rangle$, where each vertex corresponds to a first-order variable from $\mathbf{x}$, and there is a weighted edge $x \xrightarrow{c} y$ in $\mathcal{G}_\phi$ if and only if the difference constraint $x - y \leqslant c$ occurs in ϕ[4]. The weight of a path in $\mathcal{G}_\phi$ is the sum of the weights labeling its edges. An *elementary cycle* of $\mathcal{G}_\phi$ is a path with the same variable x at both endpoints, such that x does not occur anywhere else on the path, besides the endpoints. For any DBM ϕ, we have $\phi \to \bot$ if and only if the constraint graph $\mathcal{G}_\phi$ has an elementary cycle of negative weight.

The following lemma gives an equivalent condition for emptiness, that will be used to show decidability of several (non-linear) sub-classes of systems. We recall the previous assumption that, for any HS $\mathcal{H}$ and each rooted clause $C \in \mathcal{H}$, it is the case that $head(C) = \bot$.

Lemma 1. *For any rooted HS $\mathcal{H}$, we have $[\![\mathcal{H}]\!] \neq \emptyset$ if and only if $\Phi(t) \to \bot$, for any unfolding tree $t \in \mathcal{T}(\mathcal{H})$.*

Example 1. Let us consider the following Horn system:

$$\begin{array}{ll}
x - y \leqslant -1 & \to X(x,y) \\
x - x^0 \leqslant 1 \wedge x - x^1 \leqslant 3 \wedge y^0 - x^1 \leqslant 2 \wedge y^1 - y \leqslant -2 \wedge X(x^0, y^0) \wedge X(x^1, y^1) & \to X(x,y) \\
y - x \leqslant 0 \wedge x = x^0 \wedge y = y^0 \wedge X(x^0, y^0) & \to \bot \ .
\end{array}$$

A constraint graph of this system is depicted in Fig. 2. Notice that this constraint graph contains several cycles, and a cycle (depicted with thick lines) of negative weight (-1). To prove that the above system has a solution one must show that every constraint graph of the system has a negative weight cycle (Lemma 1). ■

3 Tree Automata with Presburger Constraints

Presburger Arithmetic. We recall here that *Presburger arithmetic* [12] is the set of formulae defined by the following syntax and interpreted over integers:

$$t ::= c \in \mathbb{Z} \mid x \mid t_1 + t_2 \qquad \phi ::= t_1 \leqslant t_2 \mid \phi_1 \wedge \phi_2 \mid \neg \phi_1 \mid \exists x \ . \ \phi_1$$

[4] The constraints $x \leqslant c$ and $x \geqslant c$ are usually encoded by introducing an extra variable ζ (for zero) and edges $x \xrightarrow{c} \zeta$ for $x \leqslant c$, respectively $\zeta \xrightarrow{-c} x$, for $x \geqslant c$.

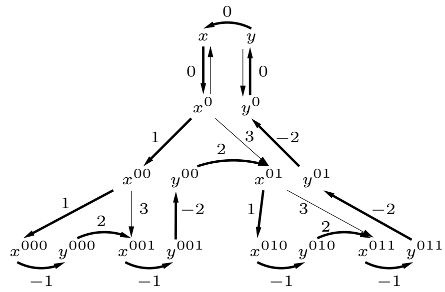


Fig. 2. Constraint Graph with a Cycle of Negative Weight

For a tuple $\mathbf{v} = \langle v_1, \ldots, v_n \rangle \in \mathbb{Z}^n$, we write $\phi(\mathbf{v})$ for the formula $\phi[v_1/x_1, \ldots, v_n/x_n]$ and $\mathbf{v} \models \phi$ for the equivalence $\phi(\mathbf{v}) \leftrightarrow \top$.

The size $|c|$ of an integer constant c is the length of its binary encoding, i.e. $|c| = \mathcal{O}(\log_2 c)$. The *size* of a Presburger formula ϕ, denoted as $|\phi|$, is defined inductively on the structure of ϕ, as: $|x| = 1$, $|t_1 + t_2| = |t_1 \leqslant t_2| = |t_1| + |t_2| + 1$, $|\phi_1 \wedge \phi_2| = |\phi_1| + |\phi_2| + 1$ and $|\neg\phi| = |\exists x \,.\, \phi| = |\phi| + 1$. We extend the size function to (second-order) Horn clauses, by defining $|X(x_1, \ldots, x_\ell)| = \ell + 1$ and, for a Horn system $\mathcal{H} = \{C_1, \ldots, C_m\}$, let $|\mathcal{H}| = \sum_{i=1}^{m} |C_i|$ be the size of $\mathcal{H}$.

Tree Automata. A (finite, non-deterministic, bottom-up) *tree automaton* (TA) is a quadruple $A = \langle Q, \Sigma, \Delta, F \rangle$, where Σ is a finite ranked alphabet, Q is a finite set of *states*, $F \subseteq Q$ is a set of *final states* and Δ is a set of *transition rules* of the form $\sigma(q_1, \ldots, q_n) \to q$, for some $\sigma \in \Sigma$ such that $\#(\sigma) = n$, and $q, q_1, \ldots, q_n \in Q$. The size of a rule is $|\sigma(q_1, \ldots, q_n) \to q| = n + 1$, and the size of the tree automaton is defined as $|A| = \sum_{\rho \in \Delta} |\rho|$.

A *run* of A over a tree $t : \mathbb{N}^* \rightharpoonup_{fin} \Sigma$ is a function $\pi : dom(t) \to Q$ such that, for each node $p \in dom(t)$, we have $q = \pi(p)$ only if $q_i = \pi(p.i)$ for all $i = 0, \ldots, \#(t(p)) - 1$, and there exists a rule $(t(p))(q_0, \ldots, q_{\#(t(p))-1}) \to q \in \Delta$. We write $t \xrightarrow{\pi}_A q$ to denote that π is a run of A over t such that $\pi(\epsilon) = q$. We use $t \Longrightarrow_A q$ to denote that $t \xrightarrow{\pi}_A q$ for some run π, and omit to specify A when it is clear from the context. The *language* of a state q of A is defined as $\mathcal{L}_q(A) = \{t \mid t \Longrightarrow_A q\}$, and the language of A is defined as $\mathcal{L}(A) = \bigcup_{q \in F} \mathcal{L}_q(A)$. The following lemma relates tree automata and Horn systems:

Lemma 2. *For any rooted HS $\mathcal{H}$, there exists a TA $T_{\mathcal{H}}$ such that $\mathcal{L}(T_{\mathcal{H}}) = \mathcal{T}(\mathcal{H})$, and $|T_{\mathcal{H}}| = \mathcal{O}(|\mathcal{H}|)$.*

TA with Presburger Constraints. Given a run π of the tree automaton $A = \langle Q, \Sigma, \Delta, F \rangle$, and an arbitrary indexing $\Delta = \{p_1, \ldots, p_m\}$ of the set of production rules of A, we denote by $\mathcal{P}_\pi(A) \in \mathbb{Z}^m$ the tuple whose i-th entry, denoted as $(\mathcal{P}_\pi(A))_i$, gives the number of times the rule p_i has been used in π, for all $i = 1, \ldots, m$. For a Presburger formula $\phi(x_1, \ldots, x_m)$, let $\mathcal{L}_\phi(A) = \left\{ t \mid \exists q \in F\ \exists \pi \,.\, \mathcal{P}_\pi(A) \models \phi \wedge t \xrightarrow{\pi} q \right\}$ be the set of trees recognized by A using only runs π, such that $\mathcal{P}_\pi(A)$ is a model of ϕ. We denote by $\mathcal{P}(A) = \left\{\mathcal{P}_\pi(A) \in \mathbb{Z}^m \mid \exists q \in F\ \exists t\ .\ t \stackrel{\pi}{\Longrightarrow} q\right\}$ the set of tuples giving the number of times each transition rule of A occurs on a run. A result from [17] proves[5] that it is possible to construct, in time $\mathcal{O}(|A|)$, a Presburger formula $\varphi(y_1, \ldots, y_m)$ such that $\mathbf{v} \models \varphi \Leftrightarrow \mathbf{v} \in \mathcal{P}(A)$, for all $\mathbf{v} \in \mathbb{Z}^m$.

Lemma 3. *Given tree automata $A = \langle Q_A, \Sigma, \Delta_A, F_A\rangle$ and $B = \langle Q_B, \Sigma, \Delta_B, F_B\rangle$, where $\Delta_A = \{p_1, \ldots, p_m\}$, and a Presburger formula $\phi(x_1, \ldots, x_m)$, the following problems are decidable:*
1. *$\mathcal{L}_\phi(A) \cap \mathcal{L}(B) = \varnothing$, and*
2. *$\mathcal{L}_\phi(A) = \mathcal{L}(B)$.*

Moreover, the problem (1) is in NPTIME *and (2) is in* EXPTIME.

4 A Decidable Class of Horn Systems

The question addressed in this paper is, given a HS $\mathcal{H}$, to decide if $[\![\mathcal{H}]\!] = \varnothing$. Observe that we do not aim at computing the solutions in a closed form, but rather answer yes/no to the question whether a solution exists. One reason is that, in general, the least solutions for even the most simple class of non-linear class of HS may not be definable in Presburger arithmetic (Appendix A).

In general, however, even the emptiness problem for HS is undecidable, since the reachability problem for the Turing-complete class of 2-counter machines [11] can be encoded using only linear HS. On the other hand, decidability can be recovered by considering only HS that encode *flat* counter machines, i.e. whose control structure is restricted by prohibiting nested loops[6]. The HS we consider in Section 5 extend this flatness condition to non-linear HS, and prove decidability of the emptiness problem in this setting of infinite-state branching transition systems. The first class of HS we study consists of systems of the form:

$$(\mathcal{B}_k)\ \begin{cases} \iota(\mathbf{x}) & \to X(\mathbf{x}) \\ \phi(\mathbf{x}, \mathbf{x}^0, \ldots, \mathbf{x}^{k-1}) \wedge X(\mathbf{x}^0) \wedge \ldots \wedge X(\mathbf{x}^{k-1}) & \to X(\mathbf{x}) \\ e(\mathbf{x}, \mathbf{x}^0) \wedge X(\mathbf{x}^0) & \to \bot \end{cases}$$

where $k \geqslant 2$ is a constant, $\mathbf{x} = \{x_1, \ldots, x_n\}$ and $\iota(\mathbf{x})$, $\phi(\mathbf{x}, \mathbf{x}^0, \ldots, \mathbf{x}^{k-1})$ and $e(\mathbf{x}, \mathbf{x}^0)$ are DBMs. The intuition is that φ describes the transition relation of a (bottom-up) branching counter machine with branching degree $k \geqslant 2$, whereas i and e define its initial and error (forbidden) configurations. For instance, the Horn system in Example 1 belongs to the class $\mathcal{B}_2$.

4.1 A Decision Procedure for $\mathcal{B}_k$ Horn Systems

In order to decide whether a given $\mathcal{B}_k$ system $\mathcal{H}$ has a solution, it is necessary to test if, for every unfolding tree t of $\mathcal{H}$, we have $\Phi(t) \to \bot$ (Lemma 1). Equivalently,

[5] The proof of Theorem 4 , with minimal changes, applies to TA, instead of context-free grammars.

[6] Each control state must be the endpoint of at most one elementary cycle.

this means that every constraint graph from the set $\{\mathcal{G}_{\Phi(t)} \mid t \in \mathcal{T}(\mathcal{H})\}$ has a cycle of negative weight. To check this condition, we first build a TA $A_{\mathcal{H}}$ that recognizes the constraint graphs consisting of exactly one cycle that is, moreover, a subgraph of some graph in the above set. Next, we compute a Presburger formula Ψ that harnesses $A_{\mathcal{H}}$ to recognize only cycles of negative weight.

The decision procedure for $\mathcal{B}_k$ systems works by checking that, *for any unfolding tree* $t \in \mathcal{T}(\mathcal{H})$, there exists an isomorphic tree $u \in \mathcal{L}_{\Psi}(A_{\mathcal{H}})$, such that the constraint graph labeling u is a subgraph of $\mathcal{G}_{\Phi(t)}$. Since $\mathcal{L}_{\Psi}(A_{\mathcal{H}})$ is the set of constraint trees encoding an elementary cycle of negative weight, it follows that $\mathcal{G}_{\Phi(t)}$ has a cycle of negative weight, i.e. $\Phi(t) \rightarrow \bot$. If this is the case for any unfolding tree $t \in \mathcal{T}(\mathcal{H})$, we deduce that $[\![\mathcal{H}]\!] \neq \emptyset$ (Lemma 1). Otherwise, if there exists an unfolding tree $t \in \mathcal{T}(\mathcal{H})$ such that $\Phi(t)$ has a satisyfing assignment, then $[\![\mathcal{H}]\!] = \emptyset$.

To check the latter condition, we build another TA $B_{\mathcal{H}}$ with the same states and transition rules as $A_{\mathcal{H}}$, but working on the alphabet Σ_{HS} of unfolding tree labels (Def. 1), which mimicks the actions of $A_{\mathcal{H}}$. Then we are left with checking whether $\mathcal{L}_{\overline{\Psi}}(B_{\mathcal{H}}) = \mathcal{T}(\mathcal{H})$, where $\overline{\Psi}$ is a Presburger formula that transposes the occurrences of the transition rules of $A_{\mathcal{H}}$ in Ψ to the rules of $B_{\mathcal{H}}$. This equivalence is decidable, by Lemma 3 (2) and the fact that $\mathcal{T}(\mathcal{H})$ is recognizable by a TA (Lemma 2). The 2EXPTIME upper bound follows from the fact that the size of $A_{\mathcal{H}}$ is exponential in the size of $\mathcal{H}$, and the equivalence problem above requires another exponential.

Before we describe in detail the decision procedure for the $\mathcal{B}_k$ class, we introduce further notation. For two graphs $G_1 = \langle N_1, E_1 \rangle$ and $G_2 = \langle N_2, E_2 \rangle$ we define the *subgraph relation* $G_1 \lhd G_2$ if and only if $N_1 = N_2$ and $E_1 \subseteq E_2$. The set of subgraphs of a graph G is denoted by $G^{\triangledown} = \{H \mid H \lhd G\}$.

Let $\mathbf{x}$ be the set of first-order variables of $\mathcal{H}$ in the rest of this section. For any DBM $\varphi(\mathbf{x}, \mathbf{x}^0, \ldots, \mathbf{x}^\ell)$ represented by the constraint graph $\mathcal{G}_\varphi = \langle \mathbf{x} \cup \bigcup_{i=0}^{\ell-1} \mathbf{x}^i, \rightarrow \rangle$, we associate the arity $\#(g) = \ell + 1$ to any subgraph $g \lhd \mathcal{G}_\varphi$. With this definition, we consider that ${\mathcal{G}_\varphi}^{\triangledown}$ is a ranked alphabet, for any DBM $\varphi(\mathbf{x}, \mathbf{x}^0, \ldots, \mathbf{x}^\ell)$.

Let $\mathcal{H}$ be a HS and $\varphi_1, \ldots, \varphi_m$ be the set of DBMs occurring in some clause of $\mathcal{H}$. A tree $t : \mathbb{N}^* \rightharpoonup_{fin} \bigcup_{i=1}^{m} {\mathcal{G}_{\varphi_i}}^{\triangledown}$ encodes a weighted constraint graph $\mathcal{G}_t = \langle \bigcup_{p \in dom(t)} \mathbf{x}^p, \rightarrow_t \rangle$ where, for any $p \in dom(t)$:

- $x^p \xrightarrow{c}_t y^p$ iff $x \xrightarrow{c} y$ is an edge in $t(p)$,
- $x^p \xrightarrow{c}_t y^{p.i}$ iff $x \xrightarrow{c} y^i$ is an edge in $t(p)$, for some $i \in 0, \ldots, \#(t(p)) - 1$,
- $x^{p.i} \xrightarrow{c}_t y^p$ iff $x^i \xrightarrow{c} y$ is an edge in $t(p)$, for some $i \in 0, \ldots, \#(t(p)) - 1$,
- $x^{p.i} \xrightarrow{c}_t y^{p.j}$ iff $x^i \xrightarrow{c} y^j$ is an edge in $t(p)$, for some $i, j \in 0, \ldots, \#(t(p)) - 1$.

In other words, $\mathcal{G}_t$ is the constraint graph obtained by the (disjoint) union of the labels of t, where each variable x (x^i, for some $i \in \mathbb{N}$) occurring at some position $p \in dom(t)$ is replaced by x^p ($x^{p.i}$). An example of a constraint graph $\mathcal{G}_t$ that labels a tree t is given in Fig. 2.

For a tree $t : \mathbb{N}^* \rightharpoonup_{fin} \bigcup_{i=1}^{m} {\mathcal{G}_{\varphi_i}}^{\triangledown}$ and an unfolding tree $u \in \mathcal{T}(\mathcal{H})$, we write $t \blacktriangleleft u$ iff $dom(t) = dom(u)$ and $t(p) \lhd \mathcal{G}_{[u(p)]_1}$, for all $p \in dom(t)$. In other words, $t \blacktriangleleft u$ holds iff t and u are isomorphic, and for each position $p \in dom(t)$, the graph

$t(p)$ is a subgraph of the constraint graph of the DBM labeling $u(p)$. It is rather straightforward to see that $\mathcal{G}_t \lhd \mathcal{G}_{\Phi(u)}$ if $t \blacktriangleleft u$.

Definition of $A_{\mathcal{H}}$. For a constant $k \geqslant 2$ and a $\mathcal{B}_k$ system $\mathcal{H}$ with first-order variables $\mathbf{x}$, second-order variable X, and the DBMs $\iota(\mathbf{x})$, $\phi(\mathbf{x}, \mathbf{x}^0, \ldots, \mathbf{x}^{k-1})$ and $e(\mathbf{x}, \mathbf{x}^0)$, let $A_{\mathcal{H}} = \langle Q_{\mathcal{H}}, \Sigma_{\mathcal{H}}, \Delta_{\mathcal{H}}, F_{\mathcal{H}} \rangle$ be the TA defined as:

- $Q_{\mathcal{H}}$ is the set of tuples $q = \langle \mathbf{x}, E, b, V \rangle$, where:
  - $E \subseteq \mathbf{x} \times \mathbf{x}$ is a transitive binary relation; intuitively, $(x, y) \in E$ and $A_{\mathcal{H}}$ has a run $t \Longrightarrow \langle \mathbf{x}, E, b, V \rangle$ iff there exists a path between x^ϵ and y^ϵ in the constraint graph $\mathcal{G}_t$ associated with t,
  - $b \in \{\bot, \top\}$ is set whenever a complete cycle has been recognized,
  - $V \in \{X, \bot\}$ is the second-order variable labeling the current position in the associated unfolding tree of $\mathcal{H}$, or $\bot$ if we are at the root.
- $\Sigma_{\mathcal{H}} = \mathcal{G}_\iota^{\triangledown} \cup \mathcal{G}_\phi^{\triangledown} \cup \mathcal{G}_e^{\triangledown}$ is the set of subgraphs of the constraint graph of a DBM that occurs in $\mathcal{H}$, namely ι, ϕ or e,
- $F_{\mathcal{H}} = \{\langle \mathbf{x}, E, b, V \rangle \in Q_{\mathcal{H}} \mid b = \top \text{ and } V = \bot\}$, i.e. the TA accepts only when complete cycle has been recognized ($b = \top$), and the current position is also the root of an associated unfolding tree ($V = \bot$).

The definition of the transition rules in $\Delta_{\mathcal{H}}$ is slightly more involved. Let $g = \langle \mathbf{x} \cup \bigcup_{i=0}^{\#(g)-1} \mathbf{x}^i, \rightarrow_g \rangle \in \Sigma_{\mathcal{H}}$ be a constraint graph and let $q_i = \langle \mathbf{x}, E_i, b_i, V_i \rangle \in Q_{\mathcal{H}}$, for $i = 0, \ldots, \#(g) - 1$, be states. Then let $h = \langle \mathcal{N}, \mathcal{E} \rangle$ be the graph:

- $\mathcal{N} = \mathbf{x} \cup \bigcup_{i=0}^{\#(g)-1} \mathbf{x}^i$, and
- $\mathcal{E} = \rightarrow_g \cup \bigcup_{i=0}^{\#(g)-1} \{(x^i, y^i) \mid (x, y) \in E_i\}$.

For any state $q = \langle \mathbf{x}, E, b, V \rangle$, there exists a transition rule $g(q_0, \ldots, q_{\#(g)-1}) \rightarrow q \in \Delta_{\mathcal{H}}$ if only if the following conditions hold:

A. For any two variables $x, y \in \mathbf{x}$, we have $(x, y) \in E$ iff there exists a unique path $x \rightsquigarrow_h y$ in the graph h. Moreover, for every edge $(u, v) \in \mathcal{E}$, there exists $(x, y) \in E$ such that (u, v) is on the (unique) path $x \rightsquigarrow_h y$ in h.

B. The relation between the boolean flags b and $b_0, \ldots, b_{\#(g)-1}$ is defined as:
   1. $b = \mathit{SingleCycle}(h) \vee \bigvee_{i=0}^{\#(g)-1} b_i$, where $\mathit{SingleCycle}(h)$ holds iff h consists of exactly one non-trivial elementary cycle,
   2. $b_i = \top$ for at most one $i = 0, \ldots, \#(g) - 1$,
   3. if $b_i = \top$, for some $i = 0, \ldots, \#(g) - 1$, we have $\rightarrow_g = \emptyset$.

C. $V = X$ if either: (i) $\#(g) = 0$, or (ii) $\#(g) = k \geqslant 2$ and $V_0 = \ldots V_{k-1} = X$, and $V = \bot$ if $\#(g) = 1$ and $V_0 = X$.

Altogether, these conditions ensure that each run of $A_{\mathcal{H}}$ will recognize only those trees whose constraint graph consists of exactly one cycle, and which have an associated (isomorphic) unfolding tree. Observe that the syntax of the $\mathcal{B}_k$ class imposes a tight connection between the number of children and the label of a position $p \in \mathit{dom}(u)$, of an unfolding tree $u \in \mathcal{T}(\mathcal{H})$. As a result, every tree $t : \mathbb{N}^* \rightharpoonup_{fin} \Sigma_{\mathcal{H}}$ has at most one corresponding unfolding tree of $\mathcal{H}$, such that $t \blacktriangleleft u$.

Lemma 4. *For any $\mathcal{B}_k$ HS $\mathcal{H}$, where $k \geqslant 2$, for any tree $t : \mathbb{N}^* \rightharpoonup_{fin} \Sigma_{\mathcal{H}}$, we have $t \in \mathcal{L}(A_{\mathcal{H}})$ if and only if:*

1. $t \blacktriangleleft u$ for some unfolding tree $u \in \mathcal{T}(\mathcal{H})$, and
2. $\mathcal{G}_t$ consists of a single non-trivial elementary cycle.
Moreover, $A_{\mathcal{H}}$ can be built in time $2^{\mathcal{O}(k \cdot |\mathcal{H}|^2)}$.

Example 2. (contd. from Ex. 1) Fig. 3 (a,b) shows two runs of $A_{\mathcal{H}}$ recognizing constraint trees from the unfolding tree represented in Fig. 2. In each state $\langle \mathbf{x}, E, b, V \rangle$ on a run, the relation E is represented by dashed lines. The b flags of all states within each run are $\perp$, with the exception of the state labeling the root, which is set to $\top$, due to the fact that, in this particular case, each cycle closes at the root. The variable V is X everywhere. Notice that the cycle from Fig. 3 (a) is of positive weight, whereas the one from (b) is of negative weight. ■

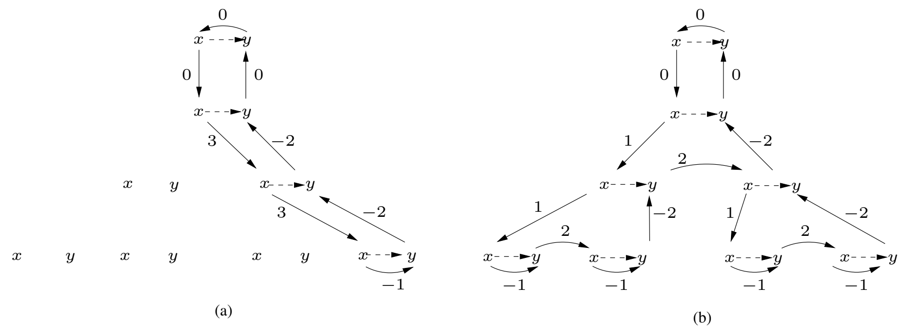


Fig. 3. Runs of $A_{\mathcal{H}}$ that Recognize Constraint Cycles

Definition of Ψ. Given $A_{\mathcal{H}} = \langle Q_{\mathcal{H}}, \Sigma_{\mathcal{H}}, \Delta_{\mathcal{H}}, F_{\mathcal{H}} \rangle$, where $\Delta_{\mathcal{H}} = \{p_1, \ldots, p_m\}$, for some $m > 0$, let $\psi(x_1, \ldots, x_m)$ be the formula defining the set $\mathcal{P}(A_{\mathcal{H}}) \subseteq \mathbb{N}^m$. Such a formula can be computed in linear time as described in [17, Theorem 4]. For each constraint graph $\sigma \in \Sigma_{\mathcal{H}}$, let $w(\sigma)$ be the sum of the weights labeling the edges in σ. For a transition rule $p = \sigma(q_0, \ldots, q_{k-1}) \rightarrow q \in \Delta_{\mathcal{H}}$, let $w(p) = w(\sigma) \in \mathbb{Z}$. With these notations, we define:

$$\Psi(x_1, \ldots, x_m) \; : \; \Psi_{\mathcal{H}}(x_1, \ldots, x_m) \wedge \sum_{i=1}^{m} x_i \cdot w(p_i) < 0$$

The above formula harnesses $A_{\mathcal{H}}$ to recognize only trees $t \in \mathcal{L}_{\Psi}(A_{\mathcal{H}})$ of negative weight $w(\mathcal{G}_t) < 0$. Thus, $t \in \mathcal{L}_{\Psi}(A_{\mathcal{H}})$ if and only if $\mathcal{G}_t$ consists of a single non-trivial elementary cycle of negative weight, and moreover, this cycle is a subgraph of $\mathcal{G}_{\Phi(u)}$, for some unfolding tree $u \in \mathcal{T}(\mathcal{H})$ (Lemma 4).

Definition of $B_{\mathcal{H}}$. The purpose of the $B_{\mathcal{H}}$ TA is to recognize only those unfolding trees $u \in \mathcal{T}(\mathcal{H})$ for which there exists a tree $t \in \mathcal{L}(A_{\mathcal{H}})$ such that $t \blacktriangleleft u$, or equivalently, the constraint graph $\mathcal{G}_t$ is a subgraph of $\mathcal{G}_{\Phi(u)}$. The idea is that $B_{\mathcal{H}}$ mimicks the actions of $A_{\mathcal{H}}$ on the alphabet Σ_{HS} of unfolding tree labels.

The formal definition is $B_{\mathcal{H}} = \langle Q_{\mathcal{H}} \times \Sigma_{\mathcal{H}}, \Sigma_{HS}, \overline{\Delta}_{\mathcal{H}}, F_{\mathcal{H}} \times \Sigma_{\mathcal{H}} \rangle$, where $Q_{\mathcal{H}}$, $\Sigma_{\mathcal{H}}$ and $F_{\mathcal{H}}$ are the ones from the definition of $A_{\mathcal{H}}$ and, for all constraint graphs $g, g_0, \dots, g_\ell \in \Sigma_{\mathcal{H}}$, all unfolding tree labels $\langle \varphi, V \rangle \in \Sigma_{HS}$ and $q, q_0, \dots, q_\ell \in Q_{\mathcal{H}}$:

$$\langle \varphi, V \rangle(\langle q_0, g_0 \rangle, \dots, \langle q_\ell, g_\ell \rangle) \to \langle q, g \rangle \in \overline{\Delta}_{\mathcal{H}}$$

if and only if (1) $g \in \mathcal{G}_{\varphi}{}^{\nabla}$, (2) $g(q_0, \dots, q_\ell) \to q \in \Delta_{\mathcal{H}}$ and (3) $q = \langle \mathbf{x}, E, b, V \rangle$. A key observation is that every transition rule in $\overline{\Delta}_{\mathcal{H}}$ has a unique associated transition rule in $\Delta_{\mathcal{H}}$. We assume in the following that $\Delta_{\mathcal{H}} = \{p_1, \dots, p_m\}$ is a fixed indexing of $\Delta_{\mathcal{H}}$ and $\overline{\Delta}_{\mathcal{H}} = \{r_1, \dots, r_s\}$ is a fixed indexing of $\overline{\Delta}_{\mathcal{H}}$. Then there exists a unique mapping of the rules of $B_{\mathcal{H}}$ into the rules of $A_{\mathcal{H}}$. Formally, we define $\delta : \{1, \dots, s\} \to \{1, \dots, m\}$, where $\delta(i) = j$, for all $i = 1, \dots, s$ iff:

$$\begin{array}{ll} r_i = \langle \varphi, V \rangle(\langle q_0, g_0 \rangle, \dots, \langle q_\ell, g_\ell \rangle) \to \langle q, g \rangle & \in \overline{\Delta}_{\mathcal{H}} \\ p_j = \qquad\qquad\qquad g(q_0, \dots, q_\ell) \to q & \in \Delta_{\mathcal{H}} \ . \end{array}$$

In the rest of this section, we consider that the mapping δ is fixed as well.

Definition of $\overline{\Psi}$. Since the rules of $A_{\mathcal{H}}$ are not in one-to-one correspondence with the rules of $B_{\mathcal{H}}$, we need to instrument the Presburger constraint Ψ to work with the rules of $B_{\mathcal{H}}$ instead. The result is another Presburger formula $\overline{\Psi}$, that is used to harness the runs of $B_{\mathcal{H}}$.

Given two tuples $\mathbf{u} \in \mathbb{Z}^m$ and $\mathbf{v} \in \mathbb{Z}^s$, we define the relation $\mathbf{u} \prec_\delta \mathbf{v}$ as $\mathbf{u}_i = \sum_{\delta(j)=i} \mathbf{v}_j$, for all $i = 1, \dots, m$. Let $\mathbf{e}_{i,j} \in \mathbb{Z}^j$ be the tuple whose i-th element is 1 and all other elements besides i are 0. It is easy to see that $\mathbf{e}_{\delta(i),m} \prec_\delta \mathbf{e}_{i,s}$, for all $i \in \{1, \dots, s\}$. Also, for all $\mathbf{u}_1, \mathbf{u}_2 \in \mathbb{Z}^m$ and $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{Z}^s$, we have $\mathbf{u}_1 + \mathbf{u}_2 \prec_\delta \mathbf{v}_1 + \mathbf{v}_2$ if $\mathbf{u}_1 \prec_\delta \mathbf{v}_1$ and $\mathbf{u}_2 \prec_\delta \mathbf{v}_2$, where the addition of tuples is defined pointwise. Given a Presburger formula $\Phi(x_1, \dots, x_m)$, we define:

$$\overline{\Phi}(x_1, \dots, x_s) = \exists y_1 \dots \exists y_m \ . \ \Phi(y_1, \dots, y_m) \wedge \bigwedge_{i=1}^{m} y_i = \sum_{\delta(j)=i} x_i \ .$$

The next lemma gives the relation between $A_{\mathcal{H}}$ and $B_{\mathcal{H}}$. As with Lemma 4, the correspondence between the trees $t \in \mathcal{L}(A_{\mathcal{H}})$, labeled by constraint cycles, and trees $u \in \mathcal{L}(B_{\mathcal{H}})$, that are also unfolding trees of $\mathcal{H}$, is defined solely by the arities of the labels of t.

Lemma 5. *Given a $\mathcal{B}_k$ HS $\mathcal{H}$, where $k \geqslant 2$, for every state $q \in Q_{\mathcal{H}}$:*
1. *for every run $u \xRightarrow{\rho} \langle q, g \rangle$ of $B_{\mathcal{H}}$, there exists a tree t, such that $t(\epsilon) = g$, and a run $t \xRightarrow{\pi} q$ of $A_{\mathcal{H}}$, such that $t \blacktriangleleft u$ and $\mathcal{P}_\pi(A_{\mathcal{H}}) \prec_\delta \mathcal{P}_\rho(B_{\mathcal{H}})$,*
2. *for every run $t \xRightarrow{\pi} q$ of $A_{\mathcal{H}}$, there exist a tree u and a run $u \xRightarrow{\rho} \langle q, t(\epsilon) \rangle$ of $B_{\mathcal{H}}$, such that $t \blacktriangleleft u$ and $\mathcal{P}_\pi(A_{\mathcal{H}}) \prec_\delta \mathcal{P}_\rho(B_{\mathcal{H}})$.*

The following theorem summarizes the result of this section.

Theorem 1. *Given a $\mathcal{B}_k$ Horn system $\mathcal{H}$, for any $k \geqslant 2$, the problem $[\![\mathcal{H}]\!] = \varnothing$ can be decided in* 2EXPTIME.

The next lemma proves a lower bound for the problem of existence of solutions in the $\mathcal{B}_k$ class. Although it does not match the 2EXPTIME upper bound from Thm. 1, this lower bound shows that the problem requires at least one exponential.

Lemma 6. *The class of problems* $\{[\![\mathcal{H}]\!] = \varnothing\}_{\mathcal{H}\in\mathcal{B}_k}$ *is* EXPTIME-*hard.*

5 Flat Horn Systems with Difference Constraints

In this section, we generalize the $\mathcal{B}_k$ class by removing the univariate restriction concerning second-order variables, and considering non-linear recursive systems with any number of second-order variables. In order to preserve decidability of the emptiness problem, we require the syntax of HS to meet the *flatness* condition, defined next.

Definition 2. *A* cycle *of a HS* $\mathcal{H}$ *is a tree* $\gamma : \mathbb{N}^* \rightharpoonup_{fin} \Sigma_{HS}$ *such that:*
(a) for all positions $p \in dom(\gamma)$, *we have* $\gamma(p) = \langle \phi, X\rangle$ *only if*

$$\phi(\mathbf{x}, \mathbf{x}^0, \ldots, \mathbf{x}^\ell) \wedge X_0(\mathbf{x}^0) \wedge \ldots \wedge X_\ell(\mathbf{x}^\ell) \rightarrow X(\mathbf{x}) \in \mathcal{H}$$

where $[\gamma(p.i)]_2 = X_i$, *for all* $i = 0, \ldots, \ell$, *and*
(b) there exists a position $p \in Fr(\gamma)$, *such that* $[\gamma(p)]_2 = X$.
The variable $X \in \mathcal{X}$ *is called the* endpoint *of* γ. *The cycle* γ *is said to be* elementary *if* $[\gamma(p)]_2 \neq X$, *for all* $p \in dom(\gamma)\setminus(\{\epsilon\} \cup Fr(\gamma))$.

An elementary cycle of a HS $\mathcal{H}$ is said to be *maximal* if it is not strictly included in another elementary cycle of $\mathcal{H}$. A rooted Horn system is said to be *flat* if every second-order variable is the endpoint of at most one maximal elementary cycle. We denote by $\mathcal{F}$ the class of flat Horn systems. For example, the union of the HS in Fig 1 (c) and (d), to which the clause $X_f(\mathbf{x}, \mathbf{x}') \wedge Y(\mathbf{x}') \rightarrow \bot$ is added, is flat: it is clearly rooted, X_f, Y are both endpoints of exactly one maximal elementary cycle each, whereas X_1, X_2 are not endpoints.

We show that the emptiness problem for flat HS is decidable, by generalizing the method from Section 4 to handle multivariate systems. Given a flat system $\mathcal{H}$, the first step is to reduce each elementary cycle to a single (recursive) clause. We consider w.l.o.g. that each second-order variable that occurs within the head of a clause of $\mathcal{H}$ also occurs in a subgoal on the left-hand side of a clause, in other words, there are no useless clauses in $\mathcal{H}$.

Formally, let $\gamma : \mathbb{N}^* \rightharpoonup_{fin} \Sigma_{HS}$ be an elementary cycle, where $\gamma(\epsilon) = \langle \phi, X\rangle$ and $\mathrm{Fr}(\gamma) = \{p_0, \ldots, p_\ell\}$ are the leaves of γ (enumerated in some total order, e.g. the lexicographical order) such that $[\gamma(p_0)]_2 = X_0, \ldots, [\gamma(p_\ell)]_2 = X_\ell$. Since γ is a cycle with endpoint X, we must have $X \in \{X_0, \ldots, X_\ell\}$. We define the clause $C_\gamma \ : \ \varPhi_\gamma(\mathbf{x}, \mathbf{x}^0, \ldots, \mathbf{x}^\ell) \wedge X_0(\mathbf{x}^0) \wedge \ldots \wedge X_\ell(\mathbf{x}^\ell) \rightarrow X(\mathbf{x})$, where $\varPhi(\gamma)$ is the characteristic formula of γ and the formula:

$$\varPhi_\gamma \ : \ \left(\exists_{p\in dom(\gamma)\setminus\{\epsilon,p_0,\ldots,p_\ell\}}\mathbf{x}^p \ . \ \varPhi(\gamma)\right)[\mathbf{x}/\mathbf{x}^\epsilon, \mathbf{x}^0/\mathbf{x}^{p_0}, \ldots, \mathbf{x}^\ell/\mathbf{x}^{p_\ell}]$$

is obtained from the composition of the DBMs labeling γ. Since DBMs have quantifier elimination[7] the formula Φ_γ is a DBM, and we can replace, for every cycle γ, the clauses of $\mathcal{H}$ that constitute the cycle γ by the clause C_γ.

This transformation yields an equisatisfiable flat HS, denoted $\mathcal{H}^\circ$. Since, moreover, quantifier elimination in DBMs takes cubic time[8], $\mathcal{H}^\circ$ can be obtained from $\mathcal{H}$ in time $\mathcal{O}(|\mathcal{H}|^3)$. A flat HS $\mathcal{H}$ is said to be *reduced* if $\mathcal{H} = \mathcal{H}^\circ$. It is not difficult to see that $\mathcal{H}$ and $\mathcal{H}^\circ$ are equisatisfiable, that is $[\![\mathcal{H}]\!] \neq \emptyset$ if and only if $[\![\mathcal{H}^\circ]\!] \neq \emptyset$. We assume w.l.o.g. that the HS under consideration is reduced.

The restricted form of a reduced HS $\mathcal{H}$ enables us now to define the tree automata $A_\mathcal{H}$ and $B_\mathcal{H}$ along the same lines as in the previous (Section 4). Formally, $A_\mathcal{H} = \langle Q_\mathcal{H}, \Sigma_\mathcal{H}, \Delta_\mathcal{H}, F_\mathcal{H}\rangle$ and $B_\mathcal{H} = \langle Q_\mathcal{H} \times \Sigma_\mathcal{H}, \Sigma_{HS}, \overline{\Delta_\mathcal{H}}, F_\mathcal{H} \times \Sigma_\mathcal{H}\rangle$, where:

- $\Sigma_\mathcal{H} = \bigcup \{{\mathcal{G}_\varphi}^\triangledown \mid \varphi \text{ occurs in } \mathcal{H}\}$,
- $Q_\mathcal{H}$ is the set of tuples $q = \langle \mathbf{x}, E, b, V\rangle$ where $E \subseteq \mathbf{x} \times \mathbf{x}$ and $b \in \{\top, \bot\}$ are as in the previous (Section 4) and $V \in \mathcal{X} \cup \{\bot\}$ can be any second-order variable of $\mathcal{H}$ or $\bot$,
- $F_\mathcal{H} = \{\langle \mathbf{x}, E, b, V\rangle \in Q_\mathcal{H} \mid b = \top \text{ and } V = \bot\}$,
- for each clause $C \ : \ \phi(\mathbf{x}, \mathbf{x}^0, \ldots, \mathbf{x}^\ell) \wedge X_0(\mathbf{x}^0) \wedge \ldots \wedge X_\ell(\mathbf{x}^\ell) \rightarrow X(\mathbf{x}) \in \mathcal{H}$, there exists transition rules:

$$\begin{array}{ll} p_i : & g(q_0, \ldots, q_\ell) \rightarrow q \quad\quad \in \Delta_\mathcal{H} \\ p_j : & \langle \phi, X\rangle(\langle q_0, g_0\rangle, \ldots, \langle q_\ell, g_\ell\rangle) \rightarrow \langle q, g\rangle \in \overline{\Delta_\mathcal{H}} \end{array}$$

  where $g \in {\mathcal{G}_\phi}^\triangledown$, $q_k = \langle \mathbf{x}, E_k, b_k, X_k\rangle$, $g_k \in \Sigma_\mathcal{H}$ and E_k, b_k are defined in the same way as for $A_\mathcal{H}$ (Section 4), for all $k = 0, \ldots, \ell$, respectively. If $\Delta_\mathcal{H} = \{p_1, \ldots, p_m\}$ and $\overline{\Delta_\mathcal{H}} = \{\overline{p}_1, \ldots, \overline{p}_s\}$, we define the mapping $\delta : \{1, \ldots, s\} \rightarrow \{1, \ldots, m\}$ as in the previous.

The following lemma formalizes the correspondence between $A_\mathcal{H}$ and $B_\mathcal{H}$, being a generalization of Lemmas 4 and 5 to the class of flat Horn systems. Observe that, because $\mathcal{H}$ is restricted, any cyclic run of $A_\mathcal{H}$ is mapped to exactly one (cyclic) run of $B_\mathcal{H}$, which corresponds to a unique cycle of an unfolding tree of $\mathcal{H}$.

Lemma 7. *Given a flat reduced Horn system $\mathcal{H}$, for every state $q \in Q_\mathcal{H}$ we have:*

1. *for every run $u \stackrel{\rho}{\Longrightarrow} \langle q, g\rangle$ of $B_\mathcal{H}$, there exists a tree t, such that $t(\epsilon) = g$, and a run $t \stackrel{\pi}{\Longrightarrow} q$ of $A_\mathcal{H}$, such that $t \blacktriangleleft u$ and $\mathcal{P}_\pi(A_\mathcal{H}) \prec_\delta \mathcal{P}_\rho(B_\mathcal{H})$,*
2. *for every run $t \stackrel{\pi}{\Longrightarrow} q$ of $A_\mathcal{H}$, there exist a tree u and a run $u \stackrel{\rho}{\Longrightarrow} \langle q, t(\epsilon)\rangle$ of $B_\mathcal{H}$, such that $t \blacktriangleleft u$ and $\mathcal{P}_\pi(A_\mathcal{H}) \prec_\delta \mathcal{P}_\rho(B_\mathcal{H})$.*

Moreover, for every tree $t \in \mathcal{L}(A_\mathcal{H})$, the constraint graph $\mathcal{G}_t$ consists of a single elementary cycle, and $A_\mathcal{H}$, $B_\mathcal{H}$ can be both built in $2^{\mathcal{O}(k \cdot |\mathcal{H}|^2)}$ time.

The following theorem gives the main result of this section.

Theorem 2. *The class of problems $\{[\![\mathcal{H}]\!] = \emptyset\}_{\mathcal{H} \in \mathcal{F}}$ is decidable in* 2EXPTIME *and is* EXPTIME*-hard.*

[7] If $\phi(\mathbf{x})$ be a DBM, and $\mathcal{G}^*_\phi$ is the constraint graph obtained from $\mathcal{G}_\phi$ by adding all edges $x_i \xrightarrow{c} x_j$, $c = \min\left\{c_{ij} + c_{jk} \mid x_i \xrightarrow{c_{ij}} x_j,\ x_j \xrightarrow{c_{jk}} x_k\right\}$ to $\mathcal{G}_\phi$, then $\exists x \ . \ \phi$ is the DBM corresponding to the elimination of x, and all edges incident to x, from $\mathcal{G}^*_\phi$.

[8] The constraint graph $\mathcal{G}_\varphi$ is strenghtened in $\mathcal{O}(|\varphi|^3)$ by the Floyd-Warshall algorithm.

6 Conclusions

We address the problem of existence of solutions for Horn systems harnessed by difference bounds constraints. Even though the problem is, in general, undecidable, we identify a simple classes of non-linear recursive systems, called $\mathcal{B}_k$, for constants $k \geqslant 2$, for which the problem has an algorithmic solution. We further generalize from $\mathcal{B}_k$ to flat systems, and show that the emptiness problem is both in 2EXPTIME and EXPTIME-hard.

References

1. T. A. Beyene, M. Brockschmidt, and A. Rybalchenko. CTL+FO verification as constraint solving. In *SPIN '14*, pages 101–104, 2014.
2. M. Bozga, R. Iosif, and F. Konečný. Safety problems are np-complete for flat integer programs with octagonal loops. In *VMCAI '14*, volume 8318 of *LNCS*, pages 242–261, 2014.
3. L. Bozzelli and S. Pinchinat. Verification of gap-order constraint abstractions of counter systems. *Theoretical Computer Science*, 523:136, 2014.
4. A. Colmerauer and P. Roussel. The birth of prolog. *SIGPLAN Not.*, 28(3):37–52, 1993.
5. H. Comon, M. Dauchet, R. Gilleron, C. Löding, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications. `http://www.grappa.univ-lille3.fr/tata`, 2007.
6. S. Demri, A. K. Dhar, and A. Sangnier. On the complexity of verifying regular properties on flat counter systems. In *ICALP '13*, volume 7966 of *LNCS*, pages 162–173, 2013.
7. S. Demri, M. Jurdziński, O. Lachish, and R. Lazić. The covering and boundedness problems for branching vector addition systems. *Journal of Computer and System Sciences*, 79(1):23–38, 2012.
8. S. Grebenshchikov, N. P. Lopes, C. Popeea, and A. Rybalchenko. Synthesizing software verifiers from proof rules. In *PLDI '12*, pages 405–416, 2012.
9. N. Heintze, J. Jaffar, S. Michaylov, P. Stuckey, and R. Yap. The clp($\mathcal{R}$) programmer's manual. Technical report, Computer Science Dept., Monash University, 1992.
10. J. Jaffar and J.-L. Lassez. Constraint logic programming. In *POPL '87*, pages 111–119, 1987.
11. M. Minsky. *Finite and Infinite Machines*. Prentice-Hall, 1967.
12. M. Presburger. Über die vollständigkeit einer gewissen systems der arithmetik ganzer zahlen, in welchem die addition als einzige operation hervortritt. In *Comptes Rendus du I congrès de Mathématiciens des Pays Slaves*, pages 92–101. 1929.
13. P. Revesz. Safe datalog queries with linear constraints. In *CP '98*, volume 1520 of *LNCS*, pages 355–369. 1998.
14. P. Z. Revesz. A closed-form evaluation for datalog queries with integer (gap)-order constraints. *Theoretical Computer Science*, 116(1):117 – 149, 1993.
15. P. Rümmer, H. Hojjat, and V. Kuncak. Disjunctive interpolants for horn-clause verification. In *CAV '13*, volume 8044 of *LNCS*, pages 347–363. 2013.
16. P. Van Hentenryck. *Constraint Satisfaction in Logic Programming*. MIT Press, Cambridge, MA, USA, 1989.
17. K. N. Verma, H. Seidl, and T. Schwentick. On the complexity of equational horn clauses. In *CADE-20*, pages 337–352, 2005.

A Horn systems with Exponential Least Solutions

The following example shows that the closed form of the least solutions of even very simple Horn systems with difference constraints is not definable in Presburger arithmetic. Given two solutions σ and σ' of a HS $\mathcal{H}$, we say that σ is *smaller* than σ' if $\sigma(X) \subseteq \sigma'(X)$, and define $(\sigma \cap \sigma')(X) = \sigma(X) \cap \sigma'(X)$, for all $X \in \mathcal{X}$. It is not difficult to see that, when a HS has a solution, it also has a unique *least solution* $\mu\mathcal{H} = \bigcap\{\sigma \mid \sigma \in [\![\mathcal{H}]\!]\}$. Let $\mathcal{H}$ be the HS below:

$$\begin{array}{ll} \varphi(x,y,z,x^0,y^0,z^0,x^1,y^1,z^1) \wedge X(x^0,y^0,z^0) \wedge X(x^1,y^1,z^1) & \to X(x,y,z) \\ x+1 = z \wedge y = 1 & \to X(x,y,z) \end{array}$$

where $\varphi \equiv (x = x^0 \wedge z = z^1 \wedge x^1 = z^0 + 1 \wedge y = y^0 + 1 \wedge y^0 = y^1)$. Let t be the unfolding tree from Fig. 4 (a). The constraint graph $\mathcal{G}_{\Phi(t)}$ is depicted in Fig. 4 (b). To avoid cluttering in Fig. 4 (a), we represent equality constraints by thin bi-directional edges, and the thick directed edges (e.g. $y^0 \to y$) stand for increment by one[9] (e.g. $y = y^0 + 1$). For the same reason, we chose not to represent the constraints of the form $y = 1$ in Fig. 4. Such constraints would require an extra (zero) variable.

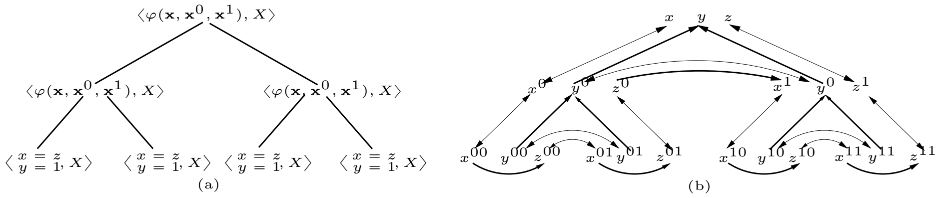


Fig. 4. Non-linear Horn System with Exponential Least Solution

The intuition is that the y variable records the height of the tree ($y = y^0 + 1$) and also ensures that the binary constraint tree is balanced ($y^0 = y^1$). The variables x and z are used to count the leaves of the tree, i.e. the value $z - x$ at the root gives the number of leaves of the balanced binary tree, which is $2^y - 1$. The least solution of $\mathcal{H}$ is thus $\mu\mathcal{H}(X) = \{\langle a,b,c\rangle \in \mathbb{Z}^3 \mid c - a = 2^b - 1\}$.

B Missing proofs

B.1 Proof of Lemma 1

Proof. "$\Rightarrow$" Suppose that $\mathcal{H}$ has a solution $\sigma : \mathcal{X} \to 2^{\bigcup_{i=1}^{\infty}\mathbb{Z}^i}$ and let t be an unfolding tree of $\mathcal{H}$. By contradiction, suppose that $\Phi(t) = \bigwedge_{p \in dom(t)} [t(p)]_1^p$ has

[9] Observe that an equality constraint $y^0 = y^1$ is encoded by the edges $y^0 \xrightarrow{0} y^1 \xrightarrow{0} y^0$, whereas $y = y^1 + 1$ is encoded by $y \xrightarrow{1} y^1 \xrightarrow{-1} y$.

a satisfiable valuation $\nu : \bigcup_{p \in dom(t)} \mathbf{x}^p \to \mathbb{Z}$. For each $p \in dom(t)$, we denote by $\mathbf{v}_p$ the tuple $\langle \nu(x_1^p), \ldots, \nu(x_n^p) \rangle \in \mathbb{Z}^n$.

We will show, by induction on the structure of $t_{|p}$, that for all $p \in dom(t) \setminus \{\epsilon\}$, we have $\mathbf{v}_p \in \sigma([t(p)]_2)$. For the base case $p \in \mathrm{Fr}(t)$, we have $t(p) = \langle \phi, X \rangle$ only if there exists a clause $\phi(\mathbf{x}) \to X(\mathbf{x}) \in \mathcal{H}$. Since σ is a solution of $\mathcal{H}$, we have $[\![\phi]\!]_\sigma \subseteq [\![X(\mathbf{x})]\!]_\sigma$ and since $\mathbf{v}_p \models \phi$, we obtain that $\mathbf{v}_p \in \sigma(X)$. For the induction step, let $\{p.0, \ldots, p.\ell\} \subseteq dom(t)$ be the set of children of $p \in dom(t) \setminus \{\epsilon\}$. We have $t(p) = \langle \phi, X \rangle$ only if there exists a clause:

$$\phi(\mathbf{x}, \mathbf{x}^0, \ldots, \mathbf{x}^\ell) \wedge X_0(\mathbf{x}^0) \wedge \ldots \wedge X_\ell(\mathbf{x}^\ell) \to X(\mathbf{x}) \in \mathcal{H}$$

and $[t(p.i)]_2 = X_i$, for all $i = 0, \ldots, \ell$. By the induction hypothesis, we have $\mathbf{v}_{p.i} \in \sigma(X_i)$, for all $i = 0, \ldots, \ell$. Since $\phi(\mathbf{v}_p, \mathbf{v}_{p.0}, \ldots, \mathbf{v}_{p.\ell}) \leftrightarrow \top$ and

$$[\![\phi]\!]_\sigma \cap [\![X_0(\mathbf{x}^0)]\!]_\sigma \cap \ldots \cap [\![X_\ell(\mathbf{x}^\ell)]\!] \subseteq [\![X(\mathbf{x})]\!]_\sigma$$

we obtain that $\mathbf{v}_p \in \sigma(X)$.

Let $t(\epsilon) = \langle \phi, \bot \rangle$ be the label of the root of t. Then there exists a clause:

$$\phi(\mathbf{x}, \mathbf{x}^0, \ldots, \mathbf{x}^\ell) \wedge X_0(\mathbf{x}^0) \wedge \ldots \wedge X_\ell(\mathbf{x}^\ell) \to \bot \in \mathcal{H}$$

where $\{0, \ldots, \ell\}$ are the children of the root in $dom(t)$. Since σ is a solution of $\mathcal{H}$, we have that:

$$[\![\phi]\!]_\sigma \cap [\![X_0(\mathbf{x}^0)]\!] \cap \ldots \cap [\![X_\ell(\mathbf{x}^\ell)]\!] = \varnothing$$

hence $\phi(\mathbf{x}, \mathbf{u}_0, \ldots, \mathbf{u}_\ell) \to \bot$, for all $\mathbf{u}_i \in \sigma(X_i)$ and all $i = 0, \ldots, \ell$. Since we proved that $\mathbf{v}_p \in \sigma([t(p)]_2)$, for all $p \in dom(t) \setminus \{\epsilon\}$, it follows that $\phi(\mathbf{x}, \mathbf{v}_0, \ldots, \mathbf{v}_\ell) \to \bot$, and, consequently, $\phi(\mathbf{v}_\epsilon, \mathbf{v}_0, \ldots, \mathbf{v}_\ell) \to \bot$, contradicting with the fact that ν is a satisfying valuation of $\Phi(t)$ and, implicitly, of $\phi = [t(\epsilon)]_1$.

"$\Leftarrow$" We define the following sequence of valuations $\{\sigma_i\}_{i=0}^\infty$:

$$\begin{aligned}
\sigma_0(X) &= \{\mathbf{v} \mid \phi \to X \in \mathcal{H},\ \phi(\mathbf{v}) \leftrightarrow \top\} \\
\sigma_{i+1}(X) &= \left\{\mathbf{v} \mid \phi \wedge X_0 \wedge \ldots \wedge X_\ell \to X \in \mathcal{H},\ \exists_{j=0}^{\ell} \mathbf{v}_j \in \textstyle\bigcup_{k=0}^{i} \sigma_k(X_j)\ .\ \phi(\mathbf{v}, \mathbf{v}_0, \ldots, \mathbf{v}_\ell) \leftrightarrow \top\right\}
\end{aligned}$$

for all $i \geqslant 0$. We show that, if $\Phi(t) \to \bot$ for each unfolding tree t of $\mathcal{H}$, then the so-valuation defined as $\sigma(X) = \bigcup_{i=0}^\infty \sigma_i(X)$ is a solution for $\mathcal{H}$. Clearly σ satisfies all non-rooted clauses of $\mathcal{H}$. Let

$$\phi(\mathbf{x}, \mathbf{x}^0, \ldots, \mathbf{x}^\ell) \wedge X_0(\mathbf{x}^0) \wedge \ldots \wedge X_\ell(\mathbf{x}^\ell) \to \bot$$

be any rooted clause of $\mathcal{H}$. To show that σ satisfies this clause, suppose, by contradiction that there exists a tuple:

$$\mathbf{v} \in [\![\phi]\!]_\sigma \cap [\![X_0(\mathbf{x}^0)]\!] \cap \ldots \cap [\![X_\ell(\mathbf{x}^\ell)]\!]\ .$$

Hence there exists tuples $\mathbf{v}_j \in \sigma(X_j)$, for all $j = 0, \ldots, \ell$, such that $\phi(\mathbf{v}, \mathbf{v}_0, \ldots, \mathbf{v}_\ell) \leftrightarrow \top$. We build top-down an unfolding tree t of $\mathcal{H}$ and a satisfying valuation for $\Phi(t)$. Let $t(\epsilon) = \langle \phi, \bot \rangle$. By the definition of σ, there exist $i_0, \ldots, i_\ell$ such that $\mathbf{v}_j \in \sigma_{i_j}(X_j)$, for all $j = 0, \ldots, \ell$. For each such i_j, two cases are possible:

- if $i_j = 0$ then there exists a clause $\varphi \to X_j$ such that $\varphi(\mathbf{v}_j) \leftrightarrow \top$. In this case, let $t(j) = \langle \varphi, X_j \rangle$ be a leaf.
- else, $i_j > 0$ and there exists a clause $\varphi \wedge Y_0 \wedge \ldots \wedge Y_s \to X_j$ and tuples $\mathbf{v}_{j0}, \ldots, \mathbf{v}_{js}$ such that $\varphi(\mathbf{v}_j, \mathbf{v}_{j0}, \ldots, \mathbf{v}_{js}) \leftrightarrow \top$. In this case, let $t(j) = \langle \varphi, X_j \rangle$ and continue inductively building the subtrees $t_{j0}, \ldots, t_{js}$.

We have thus build an unfolding tree t of $\mathcal{H}$ and a satisfying valuation $\nu(\mathbf{x}^p) = \mathbf{v}_p$, for all $p \in dom(t)$, contradiction. □

B.2 Proof of Lemma 2

Proof. We define $T_{\mathcal{H}} = \langle \mathcal{X} \cup \{\bot\}, \Sigma_{HS}, \Delta, \{\bot\} \rangle$, where:
- for each clause $\varphi(\mathbf{x}, \mathbf{x}^0, \ldots, \mathbf{x}^\ell) \wedge X_0(\mathbf{x}^0) \wedge \ldots \wedge X_\ell(\mathbf{x}^\ell) \to X(\mathbf{x})$ we have a transition rule: $\langle \varphi, X \rangle(X_0, \ldots, X_\ell) \to X \in \Delta$, and
- for each clause $\varphi(\mathbf{x}, \mathbf{x}^0, \ldots, \mathbf{x}^\ell) \wedge X_0(\mathbf{x}^0) \wedge \ldots \wedge X_\ell(\mathbf{x}^\ell) \to \bot$ we have a transition rule: $\langle \varphi, X \rangle(X_0, \ldots, X_\ell) \to \bot \in \Delta$.

Clearly $|T_{\mathcal{H}}| = \mathcal{O}(|\mathcal{H}|)$. The proof that $\mathcal{L}(T_{\mathcal{H}}) = \mathcal{T}(\mathcal{H})$ is an easy exercise. □

B.3 Proof of Lemma 3

Proof. (1) Let $A \times B$ be the TA recognizing the language $\mathcal{L}(A \times B) = \mathcal{L}(A) \cap \mathcal{L}(B)$. By the hypothesis, we have $\Delta_A = \{p_1, \ldots, p_m\}$ and let $\Delta_B = \{p'_1, \ldots, p'_\ell\}$ in the following. The states of $A \times B$ are $Q_{A \times B} = Q_A \times Q_B$, its final states are $F_{A \times B} = F_A \times F_B$ and there is a transition rule $p^\times_{ij} = \sigma((q_1, q'_1), \ldots, (q_n, q'_n)) \to (q, q') \in \Delta_{A \times B}$ if and only if $p_i = \sigma(q_1, \ldots, q_n) \to q \in \Delta_A$ and $p'_j = \sigma(q'_1, \ldots, q'_n) \to q' \in \Delta_B$, for some $i = 1, \ldots, m$ and $j = 1, \ldots, \ell$.

Among lines of the proof of Thm. 4 in [17], we build an existential Presburger formula $\varphi(y_{11}, \ldots, y_{m\ell})$ that defines the set:

$$\mathcal{P}(A \times B) = \left\{ \mathcal{P}_\theta(A \times B) \mid \exists (q, q') \in F_{A \times B} \; \exists t \; . \; t \stackrel{\theta}{\Longrightarrow}_{A \times B} (q, q') \right\} \; .$$

We prove next that (1) holds if and only if the formula:

$$\Phi(x_1, \ldots, x_m, y_{11}, \ldots, y_{m\ell}) \; : \; \varphi(y_{11}, \ldots, y_{m\ell}) \wedge \bigwedge_{i=1}^{m} x_i = \sum_{j=1}^{\ell} y_{ij} \wedge \phi(x_1, \ldots, x_m)$$

has a satisfying valuation. Clearly, the second condition is decidable, since Φ is an existential Presburger formula.

"$\Rightarrow$" Let $t \in \mathcal{L}_\phi(A) \cap \mathcal{L}(B)$ be a tree. If $t \in \mathcal{L}_\phi(A)$, there exists $q \in F_A$ and a run π of A such that $t \stackrel{\pi}{\Longrightarrow}_A q$ and $\mathcal{P}_\pi(A) \models \phi$. Also, since $t \in \mathcal{L}(B)$, there exists $q' \in F_B$ and a run ρ of B such that $t \stackrel{\rho}{\Longrightarrow}_B q'$. Let $\theta : dom(t) \to Q_{A \times B}$ be the run of $A \times B$ defined as $\theta(p) = (\pi(p), \rho(p))$, for all $p \in dom(t)$. Clearly, we have $t \stackrel{\theta}{\Longrightarrow}_{A \times B} (q, q')$. It is not difficult to see that $\mathcal{P}_\pi(A) \cdot \mathcal{P}_\theta(A \times B) \models \Phi(x_1, \ldots, x_m, y_{11}, \ldots, y_{m\ell})$.

"$\Leftarrow$" Let $\mathbf{v} \in \mathbb{Z}^m$, $\mathbf{u} \in \mathbb{Z}^{m \cdot \ell}$ be two tuples such that $\mathbf{v} \cdot \mathbf{u} \models \Phi(x_1, \ldots, x_m, y_{11}, \ldots, y_{m\ell})$. Since $\mathbf{u} \models \varphi(y_{11}, \ldots, y_{k\ell})$, there exists a tree t, a state $(q, q') \in F_{A \times B}$ and a run

$\theta : dom(t) \to Q_{A\times B}$, such that $t \xRightarrow{\theta}_{A\times B} (q, q')$ and $\mathcal{P}_\theta(A \times B) = \mathbf{u}$. By the definition of $A \times B$, we have that $t \in \mathcal{L}(B)$. To show that $t \in \mathcal{L}_\phi(A)$, we define the run $\pi : dom(t) \to Q_A$ as $\pi(p) = (\theta(p))_1$, for all $p \in dom(t)$. First, notice that $t \xRightarrow{\pi}_A q$. Second, by the definition of π, as the projection on the first entry from each label of θ, we have $(\mathcal{P}_\pi(A))_i = \sum_{j=1}^{\ell} \mathbf{u}_{ij}$. But since $\mathbf{v} \cdot \mathbf{u} \models \Phi$, we have $\mathbf{v}_i = \sum_{j=1}^{\ell} \mathbf{u}_{ij}$, hence $\mathcal{P}_\pi(A) = \mathbf{v}$ and $\mathcal{P}_\pi(A) \models \phi$, thus $t \in \mathcal{L}_\phi(A)$.

(2) The inclusion $\mathcal{L}_\phi(A) \subseteq \mathcal{L}(B)$ is equivalent to the emptiness problem $\mathcal{L}_\phi(A) \cap \mathcal{L}(\overline{B}) = \emptyset$, where $\overline{B}$ is the complement of B, i.e. $t \in \mathcal{L}(B) \Leftrightarrow t \notin \mathcal{L}(\overline{B})$ for any tree $t : \mathbb{N}^* \rightharpoonup_{fin} \Sigma$. By point (1), this problem is decidable. To decide the inclusion $\mathcal{L}(B) \subseteq \mathcal{L}_\phi(A)$, we notice that the complement of the tree language $\mathcal{L}_\phi(A)$ is $\overline{\mathcal{L}_\phi(A)} = \mathcal{L}(\overline{A}) \cup \mathcal{L}_{\neg\phi}(A)$, where $\overline{A}$ is the complement of A, defined as before. Thus $\mathcal{L}(B) \subseteq \mathcal{L}_\phi(A)$ if and only if the following hold:

- $\mathcal{L}(\overline{A}) \cap \mathcal{L}(B) = \emptyset$, which is clearly decidable, and
- $\mathcal{L}_{\neg\phi}(A) \cap \mathcal{L}(B) = \emptyset$, which is decidable using point (1).

Regarding the upper bound complexity of (1), notice that $|A \times B| = |A| \cdot |B|$ and $|\Phi| = |\varphi| + |A \times B| + |\phi|$. By Thm. 4 in [17], we have $|\varphi| = \mathcal{O}(|A \times B|) = \mathcal{O}(|A| \cdot |B|)$, hence $|\Phi| = \mathcal{O}(|A| \cdot |B| + |\phi|)$. Since deciding the satisfiability of a quantifier-free Presburger formula is in NPTIME in the size of that formula, we obtain that (1) is in NPTIME. Concerning (2), we have reduced the equivalence between $\mathcal{L}_\phi(A)$ and $\mathcal{L}(B)$ to an instance of (1) and a language equivalence between A and B. The first problem is in NPTIME and the second in EXPTIME, hence we obtain that (2) is in EXPTIME. $\square$

B.4 Proof of Lemma 4

Proof. "$\Rightarrow$" (1) We prove a more general statement: for all $t \in \mathcal{L}_{\langle \mathbf{x},E,b,V\rangle}(A_\mathcal{H})$ there exists an unfolding tree $u \in \mathcal{T}_V(\mathcal{H})$ such that $t \blacktriangleleft u$. We prove this fact by induction on the structure of t.

For the base case $dom(t) = \{\epsilon\}$, the only possible run of $A_\mathcal{H}$ over t applies a transition rule of the form $g \to \langle \mathbf{x}, E, b, X\rangle$, where $\#(g) = 0$. Since the only symbols of zero arity are the ones in $\mathcal{G}_\iota^{\triangledown}$, we obtain that $g \lhd \mathcal{G}_\iota$. By defining u as $dom(u) = \{\epsilon\}$ and $u(\epsilon) = \langle i, X\rangle$, we have $t \blacktriangleleft u$.

For the inductive step, let $t_{|0}, \ldots, t_{|\ell-1}$ be the subtrees of the root in t, for some $\ell \geqslant 1$. Let $t \xRightarrow{\pi} \langle \mathbf{x}, E, b, V\rangle$ be a run of $A_\mathcal{H}$ over t, and $\pi(i) = \langle \mathbf{x}, E_i, b_i, V_i\rangle$, for all $i = 0, \ldots, \ell - 1$. We distinguish two cases:

- if $\ell = 1$ then $V_0 = X$, $V = \bot$ and $g(\pi(0)) \to \langle \mathbf{x}, E, b, V\rangle$ is the final transition rule on π, where $\#(g) = 1$ (point C of the definition of $\Delta_\mathcal{H}$). Since the only symbols of arity 1 are the ones in $\mathcal{G}_e^{\triangledown}$, we have $g \lhd \mathcal{G}_e$. By the induction hypothesis, we obtain an unfolding tree $u_0 \in \mathcal{T}_X(\mathcal{H})$ such that $t_{|0} \blacktriangleleft u_0$. We define u as $dom(u) = dom(t)$, $u(\epsilon) = \langle e, \bot\rangle$ and $u_{|0} = u_0$. The checks that $u \in \mathcal{T}_\bot(\mathcal{H})$ and $t \blacktriangleleft u$ are immediate.
- else ($\ell > 1$) we have $\ell = k \geqslant 2$, $V_0 = \ldots V_{k-1} = V = X$ and $g(\pi(0), \ldots, \pi(k-1)) \to \langle \mathbf{x}, E, b, V\rangle$ is the final transition rule on π, where $\#(g) = k$ (point C of the definition of $\Delta_\mathcal{H}$). Since the only symbols of arity $k \geqslant 2$ are the ones in Σ_ϕ, we have $g \lhd \mathcal{G}_\phi$. By the induction hypothesis, we obtain unfolding trees

$u_0, \ldots, u_{k-1} \in \mathcal{T}_X(\mathcal{H})$ such that $t_{|i} \blacktriangleleft u_i$, for all $i = 0, \ldots, k-1$. We define u as $dom(u) = dom(t)$, $u(\epsilon) = \langle \phi, X \rangle$ and $u_{|i} = u_i$, for all $i = 0, \ldots, k-1$. The checks $u \in \mathcal{T}_X(\mathcal{H})$ and $t \blacktriangleleft u$ are immediate.

(2) If $t \in \mathcal{L}(A_\mathcal{H})$ then $t \stackrel{\pi}{\Longrightarrow} q$, for some final state $q \in F_\mathcal{H}$. Let $\pi(p) = \langle \mathbf{x}, E_p, V_p, b_p \rangle$ be the states on the run π, for all $p \in dom(t)$. We also define the graphs $h_p = \langle \mathcal{N}_p, \mathcal{E}_p \rangle$, where:

- $\mathcal{N}_p = \mathbf{x}^p \cup \bigcup_{i=0}^{\#(t(p))-1} \mathbf{x}^{p.i}$,
- $\mathcal{E}_p = \rightarrow_{t(p)} \cup \bigcup_{i=0}^{\#(t(p))-1} \{(x^{p.i}, y^{p.i}) \mid (x, y) \in E_{p.i}\}$.

The following fact can be shown by induction on the structure of $t_{|p}$, applying point A of the definition of $\Delta_\mathcal{H}$ inductively:

Fact 3 *For all $p \in dom(t)$, there exists a path $x^p \rightsquigarrow y^p$ in h_p if and only if there exists a unique path $x^p \rightsquigarrow y^p$ in $\mathcal{G}_{t_{|p}}$, and moreover, every edge in $\mathcal{G}_{t_{|p}}$ is part of such a path.*

Since $q = \pi(\epsilon) \in F_\mathcal{H}$, we have $b_\epsilon = \top$. By the points (B1) and (B2) of the definition of $\Delta_\mathcal{H}$, there exists a unique maximal sequence $\epsilon = p_0, \ldots, p_s \in dom(t)$, such that $b_{p_j} = \top$ and p_j is a child of p_{j-1}, we have $b_p = \bot$, for all $j = 1, \ldots, s$. Moreover, since the sequence is maximal, for all $p \in dom(t) \setminus \{p_0, \ldots, p_s\}$. By (B1) h_{p_s} consists of a unique non-trivial elementary cycle, hence by Fact 3 above, $\mathcal{G}_{t_{|p_s}}$ also consists of a unique non-trivial elementary cycle.

To complete the proof of (2), we must show that there are no other edges in $\mathcal{G}_t$, except for the ones on this cycle. Since $b_{p_j} = \top$, we have $\mathcal{E}_{p_j} = \emptyset$, for all $j = 0, \ldots, s-1$ (B3). Then, every constraint graph $\mathcal{G}_{t_{|r}}$ has an empty edge set, for every child $r \neq p_{j+1}$ of some p_j, $j = 0, \ldots, s-1$ – assuming the opposite would contradict with Fact 3 above.

"$\Leftarrow$ Let $t : \mathbb{N}^* \rightharpoonup_{fin} \Sigma_\mathcal{H}$ be a tree, such that $\mathcal{G}_t$ consists of a single non-trivial elementary cycle, and $t \blacktriangleleft u$, for some $u \in \mathcal{T}(\mathcal{H})$. We build a run $\pi : dom(t) \to Q_\mathcal{H}$ of $A_\mathcal{H}$, such that $t \stackrel{\pi}{\Longrightarrow} q$, for some $q \in \mathcal{F}_\mathcal{H}$. We define $\pi(p)$ and prove that $t_{|p} \stackrel{\pi_{|p}}{\Longrightarrow} \pi(p)$ is a run of $A_\mathcal{H}$, by induction on the structure of $t_{|p}$.

For the base case $p \in \mathrm{Fr}(t)$, we have $u(p) = \langle i, X \rangle$ and $t(p) \lhd \mathcal{G}_\iota$, hence $\#(t(p)) = 0$. Let $\pi(p) = \langle \mathbf{x}, E_p, b_p, V_p \rangle$, where:

- $E_p = \{(x, y) \mid x \rightsquigarrow y \text{ is a path in } t(p)\}$,
- $b_p = \mathit{SingleCycle}(t(p))$, and
- $V_p = X$.

To show that $t(p) \to \pi(p)$ is a transition rule of $A_\mathcal{H}$, it is sufficient to prove point (A) from the definition of $\Delta_\mathcal{H}$ (points B and C hold trivially because $\#(t(p)) = 0$). But (A) holds by the assumption that $\mathcal{G}_t$ consists of a single non-trivial elementary cycle.

For the inductive step, let $p.0, \ldots, p.(\#(t(p))-1)$ be the children of p in $dom(t)$. Also, let $u(p) = \langle \varphi, V \rangle$, for $\varphi \in \{\phi, e\}$ and $V \in \{X, \bot\}$. By the induction hypothesis, we have $t_{|p.i} \stackrel{\pi}{\Longrightarrow} \pi(p.i) = \langle \mathbf{x}, E_{p.i}, b_{p.i}, V_{p.i} \rangle$, for all $i = 0, \ldots, \#(t(p)) - 1$. We define the graph $h_p = \langle \mathcal{N}_p, \mathcal{E}_p \rangle$ as in the previous, and $\pi(p) = \langle \mathbf{x}, E_p, b_p, V_p \rangle$ as follows:

- $E_p = \{(x, y) \mid x^p \rightsquigarrow y^p \text{ in } h_p\}$,

– $b_p = \mathit{SingleCycle}(h_p) \vee \bigvee_{i=0}^{\#(t(p))-1} b_i$, and
– $V_p = V$.

We can verify points (A), (B) and (C) from the definition of $\Delta_{\mathcal{H}}$ using the fact that $\mathcal{G}_t$ consists of a single non-trivial elementary cycle, and deduce that $t(p)(\pi(p.0), \ldots, \pi(p.(\#(t(p)) - 1))) \to \pi(p)$ is a transition rule of $A_{\mathcal{H}}$. The proof involves a case split on the arity of $t(p)$ and distinguishes the cases $\#(t(p)) = 1$, i.e. $t(p) \in \mathcal{G}_e{}^{\triangledown}$, and $\#(t(p)) = k \geqslant 2$, i.e. $t(p) \in \mathcal{G}_\phi{}^{\triangledown}$.

Concerning the time complexity of the construction of $A_{\mathcal{H}}$, observe first that this is $\mathcal{O}(|A_{\mathcal{H}}|)$, since building $A_{\mathcal{H}}$ amounts to enumerating its transition rules. There are at most $\|\Sigma_{\mathcal{H}}\| \cdot \|Q_{\mathcal{H}}\|^{k+1}$ transition rules in $\Delta_{\mathcal{H}}$, each of which being of size at most $k+1$. Hence $|A_{\mathcal{H}}| \leqslant (k+1) \cdot \|\Sigma_{\mathcal{H}}\| \cdot \|Q_{\mathcal{H}}\|^{k+1}$. Since, for any DBM φ, the number of edges in a constraint graph $\mathcal{G}_\varphi$ is at most $|\varphi|$, we have $\|\mathcal{G}_\varphi{}^{\triangledown}\| \leqslant 2^{|\varphi|}$. We compute:

$$\begin{aligned}\|\Sigma_{\mathcal{H}}\| &\leqslant \|\mathcal{G}_\iota{}^{\triangledown}\| + \|\mathcal{G}_e{}^{\triangledown}\| + \|\mathcal{G}_\phi{}^{\triangledown}\| \\ &\leqslant 2^{|i|} + 2^{|e|} + 2^{|\phi|} \leqslant 2^{|i|+|e|+|\phi|} \leqslant 2^{|\mathcal{H}|} \ .\end{aligned}$$

Since each state $q = \langle \mathbf{x}, E, b, V \rangle \in Q_{\mathcal{H}}$ is defined by a relation $E \subseteq \mathbf{x} \times \mathbf{x}$, a boolean flag b and a variable $V \in \{X, \bot\}$, we have:

$$\|Q_{\mathcal{H}}\| \leqslant 4 \cdot 2^{\|\mathbf{x}\|^2} \leqslant 2^{|\mathcal{H}|^2}$$

We have $|A_{\mathcal{H}}| \leqslant (k+1) \cdot 2^{|\mathcal{H}|+(k+1)|\mathcal{H}|^2}$, thus we obtain $|A_{\mathcal{H}}| = 2^{\mathcal{O}(k \cdot |\mathcal{H}|^2)}$. □

B.5 Proof of Lemma 5

Fact 4 *For any Presburger formula $\Phi(x_1, \ldots, x_m)$ and two tuples $\mathbf{u} \in \mathbb{Z}^m$ and $\mathbf{v} \in \mathbb{Z}^s$, such that $\mathbf{u} \prec_\delta \mathbf{v}$, we have $\mathbf{u} \models \Phi \Leftrightarrow \mathbf{v} \models \overline{\Phi}$.*

Proof. "$\Rightarrow$" If $\mathbf{u} \models \Phi$, we choose $y_i = \mathbf{u}_i$, for all $i = 1, \ldots, m$, in $\overline{\Phi}$. Since $\mathbf{u}_i = \sum_{\delta(j)=i} \mathbf{v}_i$, we obtain:

$$\begin{aligned}&\mathbf{v} \models \Phi(\mathbf{u}) \wedge \textstyle\bigwedge_{i=1}^m \mathbf{u}_i = \sum_{\delta(j)=i} x_i \\ &\mathbf{v} \models \exists y_1 \ldots \exists y_m \ . \ \Phi(y_1, \ldots, y_m) \wedge \textstyle\bigwedge_{i=1}^m y_i = \sum_{\delta(j)=i} x_i \\ &\mathbf{v} \models \overline{\Phi} \ .\end{aligned}$$

"$\Leftarrow$" If $\mathbf{v} \models \overline{\Phi}$, there exists $\mathbf{u}' \in \mathbb{Z}^m$ such that $\mathbf{u}' \models \Phi$ and $\mathbf{u}'_i = \sum_{\delta(j)=i} \mathbf{v}_i = \mathbf{u}_i$, for all $i = 1, \ldots, m$. Hence $\mathbf{u} = \mathbf{u}'$, hence $\mathbf{u} \models \Phi$. □

Proof. Let $\iota(\mathbf{x})$, $\phi(\mathbf{x}, \mathbf{x}^0, \ldots, \mathbf{x}^k)$ and $e(\mathbf{x}, \mathbf{x}^0)$ be the DBMs occurring in $\mathcal{H}$.

(1) By induction on the structure of u. For the base case $dom(u) = \{\epsilon\}$, we have $\rho(\epsilon) = \langle q, g \rangle$ if $u(\epsilon) = \langle i, X \rangle$ and there exists a transition rule $r_i = \langle i, X \rangle \to q \in \overline{\Delta}_{\mathcal{H}}$, for some $i \in \{1, \ldots, s\}$. By the definition of $\overline{\Delta}_{\mathcal{H}}$, we have $g \lhd \mathcal{G}_\iota$, and there exists a transition rule $p_{\delta(i)} = g \to q \in \Delta_{\mathcal{H}}$. We define $dom(t) = dom(\pi) = \{\epsilon\}$, $t(\epsilon) = g$ and $\pi(\epsilon) = q$. We clearly have $t \xrightarrow{\pi} q$ and $t \blacktriangleleft u$. Moreover, we have $\mathcal{P}_\pi(A_{\mathcal{H}}) = \mathbf{e}_{\delta(i),m} \prec_\delta \mathbf{e}_{i,s} = \mathcal{P}_\rho(B_{\mathcal{H}})$.

For the inductive step, let $0, \ldots, \ell$ be the children of the root of u, for some $\ell \geqslant 0$, and let $r_i = \langle \varphi, V \rangle (\langle q_0, g_0 \rangle, \ldots, \langle q_\ell, g_\ell \rangle) \to \langle q, \underline{g} \rangle$ be the last transition rule fired in ρ, for some $i \in \{1, \ldots, s\}$. By the definition of $\overline{\Delta}_{\mathcal{H}}$, there exists a transition rule $p_{\delta(i)} = g(q_0, \ldots, q_\ell) \to q$ in $A_{\mathcal{H}}$, such that $g \in \mathcal{G}_\varphi{}^\nabla$. Since $u_{|j} \xrightarrow{\rho_{|j}} \langle q_j, g_j \rangle$ are runs of $B_{\mathcal{H}}$, by the induction hypothesis, we obtain trees t_j and runs $t_j \xrightarrow{\pi_j} q_j$ of $A_{\mathcal{H}}$, such that $t_j \blacktriangleleft u_j$ and $\mathcal{P}_{\pi_j}(A_{\mathcal{H}}) \prec_\delta \mathcal{P}_{\rho_{|j}}(B_{\mathcal{H}})$, for all $j = 0, \ldots, \ell$. We define $t(\epsilon) = g$, $t_{|j} = t_j$ and $\pi(\epsilon) = q$, $\pi_{|j} = \pi_j$, for all $j = 0, \ldots, \ell$, respectively. It is easy to check that $t \xrightarrow{\pi} q$ is a run of $A_{\mathcal{H}}$ and that $t \blacktriangleleft u$. Moreover, we have $\mathcal{P}_\pi(A_{\mathcal{H}}) = \mathbf{e}_{\delta(i),m} + \sum_{j=0}^{\ell} \mathcal{P}_{\pi_j}(A_{\mathcal{H}}) \prec_\delta \mathbf{e}_{i,s} + \sum_{j=0}^{\ell} \mathcal{P}_{\rho_{|j}}(B_{\mathcal{H}})$.

(2) By induction on the structure of t. For the base case $dom(t) = \{\epsilon\}$, the run π consists of a single transition rule $p_j = t(\epsilon) \to q \in \Delta_{\mathcal{H}}$, where $\#(t(\epsilon)) = 0$ and $q = \langle \mathbf{x}, E, b, V \rangle$. But then $t(\epsilon) \in \mathcal{G}_\iota{}^\nabla$, $V = X$ and there exists a transition rule $r_i = \langle \iota, X \rangle \to \langle q, t(\epsilon) \rangle \in \overline{\Delta}_{\mathcal{H}}$, where $\delta(i) = j$. Then we define u and ρ as $dom(u) = dom(\rho) = \{\epsilon\}$, $u(\epsilon) = \langle \iota, X \rangle$ and $\rho(\epsilon) = \langle q, t(\epsilon) \rangle$. Clearly, $u \xrightarrow{\rho} \langle q, t(\epsilon) \rangle$ is a run of $B_{\mathcal{H}}$, $t \blacktriangleleft u$ and $\mathcal{P}_\pi(A_{\mathcal{H}}) = \mathbf{e}_{\delta(i),m} \prec_\delta \mathbf{e}_{i,s} = \mathcal{P}_\rho(B_{\mathcal{H}})$.

For the inductive step, let $0, \ldots, \ell$, for some $\ell \geqslant 0$, be the children of the root of t, and $p_j = t(\epsilon)(q_0, \ldots, q_\ell) \to q \in \Delta_{\mathcal{H}}$ be the last transition rule of π, where $q = \langle \mathbf{x}, E, b, V \rangle$. Since $t_{|h} \xrightarrow{\pi_{|h}} q_h$, for all $h = 0, \ldots, \ell$, by the induction hypothesis we obtain trees u_h and runs $u_h \xrightarrow{\rho_h} \langle q_h, t(h) \rangle$ of $B_{\mathcal{H}}$, such that $t_h \blacktriangleleft u_h$ and $\mathcal{P}_{\pi_{|h}}(A_{\mathcal{H}}) \prec_\delta \mathcal{P}_{\rho_h}(B_{\mathcal{H}})$, for all $h = 0, \ldots, \ell$. We distinguish two cases:

1. if $\#(t(\epsilon)) = 1$, then $t(\epsilon) \in \mathcal{G}_e{}^\nabla$ and $V = \bot$. Then there exists a transition rule $r_i = \langle e, \bot \rangle (\langle q_0, t(0) \rangle, \ldots, \langle q_\ell, t(\ell) \rangle) \to \langle q, t(\epsilon) \rangle \in \overline{\Delta}_{\mathcal{H}}$, where $\delta(i) = j$. We define u and ρ as $u(\epsilon) = \langle e, \bot \rangle$, $\rho(\epsilon) = \langle q, t(\epsilon) \rangle$ and $u_{|h} = u_h$, $\rho_{|h} = \rho_h$, for all $h = 0, \ldots, \ell$. Clearly, $u \xrightarrow{\rho} \langle q, t(\epsilon) \rangle$ is a run of $B_{\mathcal{H}}$, $t \blacktriangleleft u$ and $\mathcal{P}_\pi(A_{\mathcal{H}}) = \mathbf{e}_{\delta(i),m} + \sum_{h=0}^{\ell} \mathcal{P}_{\pi_{|h}}(A_{\mathcal{H}}) \prec_\delta \mathbf{e}_{i,s} + \sum_{h=0}^{\ell} \mathcal{P}_{\rho_h}(B_{\mathcal{H}}) = \mathcal{P}_\rho(B_{\mathcal{H}})$.
2. else $\#(t(\epsilon)) = k \geqslant 2$, $t(\epsilon) \in \mathcal{G}_\phi{}^\nabla$ and $V = X$. The rest of the proof is identical to the one from the previous point.

□

B.6 Proof of Theorem 1

Proof. Let $\iota(\mathbf{x})$, $\phi(\mathbf{x}, \mathbf{x}^0, \ldots, \mathbf{x}^k)$ and $e(\mathbf{x}, \mathbf{x}^0)$ be the DBMs occurring in $\mathcal{H}$. Let $A_{\mathcal{H}}$, $B_{\mathcal{H}}$, Ψ and $\overline{\Psi}$ be the TA and Presburger formulae defined above, respectively. It is sufficient to prove that:

$$[\![\mathcal{H}]\!] \neq \emptyset \Leftrightarrow \mathcal{L}_{\overline{\Psi}}(B_{\mathcal{H}}) = \mathcal{T}(\mathcal{H}) \ .$$

By Lemma 2 there exists a TA $T_{\mathcal{H}}$ such that $|T_{\mathcal{H}}| = \mathcal{O}(|\mathcal{H}|)$ and $\mathcal{L}(T_{\mathcal{H}}) = \mathcal{T}(\mathcal{H})$. Since $A_{\mathcal{H}}$ can be constructed in time $2^{\mathcal{O}(k \cdot |\mathcal{H}|^2)}$ (Lemma 4), the same holds for $B_{\mathcal{H}}$ and thus, $|B_{\mathcal{H}}| = 2^{\mathcal{O}(k \cdot |\mathcal{H}|^2)}$. Also we have $|\Psi| = \mathcal{O}(|A_{\mathcal{H}}|)$ and $|\overline{\Psi}| = |\Psi| + \mathcal{O}(|B_{\mathcal{H}}|) = 2^{\mathcal{O}(k \cdot |\mathcal{H}|^2)}$. By Lemma 3 (2) we obtain the 2EXPTIME bound.

By Lemma 1, we have $[\![\mathcal{H}]\!] \neq \emptyset$ iff for any unfolding tree $u \in \mathcal{T}(\mathcal{H})$ we have $\Phi(u) \to \bot$ iff for any $u \in \mathcal{T}(\mathcal{H})$, the constraint graph $\mathcal{G}_{\Phi(u)}$ has a non-trivial elementary cycle γ of negative weight. But then there exists a tree $t : \mathbb{N}^* \rightharpoonup_{fin} \Sigma_{\mathcal{H}}$, such that $t \blacktriangleleft u$ and $\mathcal{G}_t$ consists of exactly one elementary cycle. By Lemma 4, we obtain, equivalently, that $t \in \mathcal{L}(A_{\mathcal{H}})$. We have thus:

$$[\![\mathcal{H}]\!] \neq \emptyset \Leftrightarrow \mathcal{T}(\mathcal{H}) = \{u \in \mathcal{T}(\mathcal{H}) \mid \exists t \in \mathcal{L}_{\Psi}(A_{\mathcal{H}}) \text{ and } t \blacktriangleleft u\} \enspace .$$

We are left with proving that $\mathcal{L}_{\overline{\Psi}}(B_{\mathcal{H}}) = \{u \in \mathcal{T}(\mathcal{H}) \mid \exists t \in \mathcal{L}_{\Psi}(A_{\mathcal{H}}) \text{ and } t \blacktriangleleft u\}$.

"$\subseteq$" Let $u \in \mathcal{L}_{\overline{\Psi}}(B_{\mathcal{H}})$ and $u \stackrel{\rho}{\Longrightarrow} \langle q, g\rangle$ be a run of $B_{\mathcal{H}}$, for some $q \in F_{\mathcal{H}}$ and $g \in \Sigma_{\mathcal{H}}$. By Lemma 5 (1), there exists a tree $t \in \mathcal{L}(A_{\mathcal{H}})$ such that $t \blacktriangleleft u$ and $A_{\mathcal{H}}$ has a run $t \stackrel{\pi}{\Longrightarrow} q$, such that $\mathcal{P}_\pi(A_{\mathcal{H}}) \prec_\delta \mathcal{P}_\rho(B_{\mathcal{H}})$. Since $\mathcal{P}_\rho(B_{\mathcal{H}}) \models \overline{\Psi}$, we obtain $\mathcal{P}_\rho(A_{\mathcal{H}}) \models \Psi$, by Fact 4, hence $t \in \mathcal{L}_{\Psi}(A_{\mathcal{H}})$.

"$\supseteq$" Let $u \in \mathcal{T}(\mathcal{H})$ be a tree such that $t \blacktriangleleft u$ for some $t \in \mathcal{L}_{\Psi}(A_{\mathcal{H}})$. Then $A_{\mathcal{H}}$ has a run $t \stackrel{\pi}{\Longrightarrow} q$ for some state $q \in Q_{\mathcal{H}}$ and $\mathcal{P}_\pi(A_{\mathcal{H}}) \models \Psi$. By Lemma 5 (2) $B_{\mathcal{H}}$ has a run $u' \stackrel{\rho}{\Longrightarrow} \langle q, t(\epsilon)\rangle$, for some $u' \in \mathcal{L}(B_{\mathcal{H}})$, such that $t \blacktriangleleft u'$ and $\mathcal{P}_\pi(A_{\mathcal{H}}) \prec_\delta \mathcal{P}_\rho(B_{\mathcal{H}})$. Since $t \blacktriangleleft u$ and $t \blacktriangleleft u'$ we get $dom(u) = dom(u')$. Since $u' \in \mathcal{L}(B_{\mathcal{H}})$, it is easy to check that $u' \in \mathcal{T}(\mathcal{H})$. Since $dom(u) = dom(u')$, we obtain that $u = u'$, because the following hold, as a consequence of the fact that $u, u' \in \mathcal{T}(\mathcal{H})$ are both unfolding trees of $\mathcal{H}$:

- $u(\epsilon) = u'(\epsilon) = \langle e, \bot\rangle$,
- $u(p) = u'(p) = \langle \iota, X\rangle$, for all $p \in \mathrm{Fr}(u)$, and
- $u(p) = u'(p) = \langle \phi, X\rangle$, for all $p \in dom(u) \backslash (\mathrm{Fr}(u) \cup \{\epsilon\})$.

Hence $u \in \mathcal{L}(B_{\mathcal{H}})$. We get further that $u \in \mathcal{L}_{\overline{\Psi}}(B_{\mathcal{H}})$, because $\mathcal{P}_\pi(A_{\mathcal{H}}) \prec_\delta \mathcal{P}_\rho(B_{\mathcal{H}})$ and $\mathcal{P}_\pi(A_{\mathcal{H}}) \models \Psi$, by Fact 4. □

B.7 Proof of Lemma 6

Proof. By reduction from the universality problem for tree automata, which is a known EXPTIME-complete problem [5, Theorem 1.7.7]. Let $A = \langle Q, \Sigma, \Delta, F\rangle$ be a TA. For simplicity, we assume that $\Sigma = \{f, g\}$, $\#(f) = 2$ and $\#(g) = 0$ – the proof below can be adapted to ranked alphabets with symbols of arbitrary arities, at the expense of introducing further technicalities.

We build a $\mathcal{B}_k$ system $\mathcal{H}_A$ such that $[\![\mathcal{H}_A]\!] = \emptyset$ if and only if $\mathcal{L}(A)$ contains all trees $t : \{0,1\}^* \rightharpoonup_{fin} \Sigma$. Let $Q = \{q_1, \ldots, q_\ell\}$ be the states of A. We build DBMs $\iota_A(\mathbf{x})$, $\phi_A(\mathbf{x}, \mathbf{x}^0, \mathbf{x}^1)$ and $e_A(\mathbf{x}, \mathbf{x}^0)$, where $\mathbf{x} = \{x_1, \ldots, x_{2\ell}\}$. The idea is to use the variables such that x_{2i} and x_{2i+1} denote q_i, for all $i = 1, \ldots, \ell$. We define $\mathcal{H}_A$ as follows:

$$\begin{aligned} \iota_A(\mathbf{x}) &: \textstyle\bigwedge_{g \to q_i} x_{2i} - x_{2i+1} \leqslant 0 \\ \phi_A(\mathbf{x}, \mathbf{x}^0, \mathbf{x}^1) &: \textstyle\bigwedge_{f(q_u, q_v) \to q_i} x_{2i} - x^0_{2u} \leqslant 0 \wedge x^0_{2u+1} - x^1_{2v} \leqslant 0 \wedge x^1_{2v+1} - x_{2u+1} \leqslant 0 \\ e_A(\mathbf{x}, \mathbf{x}^0) &: \textstyle\bigwedge_{q_i \in F} x_{2i+1} - x_{2i} \leqslant -1 \wedge \bigwedge_{j=1}^{2\ell} x_j = x^0_j \end{aligned}$$

Observe that the reduction takes polynomial time in the size of A. It is not difficult to verify that every unfolding tree of $\mathcal{H}_A$ has a cycle if and only if $\mathcal{L}(A)$ contains all trees $t : \{0,1\}^* \rightharpoonup_{fin} \{f, g\}$. Since every cycle in some unfolding tree of $\mathcal{H}_A$ is necessarily of negative weight, we have $[\![\mathcal{H}_A]\!] = \emptyset$ if and only if A is universal. □

B.8 Proof of Lemma 7

Proof. Along the same lines as the proof of Lemma 5. □

B.9 Proof of Theorem 2

Proof. Along the lines of the proof of Thm. 1, using Lemma 7 instead of Lemma 5. The definitions of the harness formulae Ψ and $\overline{\Psi}$ are identical to the ones used in the proof of Thm. 1. To show the inclusion $\mathcal{L}_{\overline{\Psi}}(B_{\mathcal{H}}) \supseteq \{u \in \mathcal{T}(\mathcal{H}) \mid \exists t \in \mathcal{L}_{\Psi}(A_{\mathcal{H}}) \wedge t \blacktriangleleft u\}$, one must show that two unfolding trees $u, u' \in \mathcal{T}(\mathcal{H})$ such that $dom(u) = dom(u')$ and $[u(p)]_2 = [u'(p)]_2$ for all $p \in dom(u)$, must be identical. The fact that $\mathcal{H}$ is reduced is used here.

The 2EXPTIME upper bound follows from the sizes of $A_{\mathcal{H}}$, $B_{\mathcal{H}}$, Ψ and $\overline{\Psi}$, that are $2^{\mathcal{O}(k|\mathcal{H}|^2)} = 2^{\mathcal{O}(|\mathcal{H}|^3)}$, where $k \leqslant |H|$ is the maximal branching degree of all clauses in $\mathcal{H}$. The EXPTIME-hard lower bound comes from Lemma 6. □