CATERPILLAR DUALITIES AND REGULAR LANGUAGES

PÉTER L. ERDŐS, CLAUDE TARDIF, AND GÁBOR TARDOS

Abstract. We characterize obstruction sets in caterpillar dualities in terms of regular languages, and give a construction of the dual of a regular family of caterpillars. In particular, we prove that every monadic linear Datalog program with at most one EDB per rule defines the complement of a contraint satisfaction problem.

1. Introduction

A *homomorphism duality* is a couple $(\mathcal{O}, \mathbf{D})$ where $\mathbf{D}$ is a relational structure and $\mathcal{O}$ is a family of relational structures of the same type, such that the following holds.

> For any given relational structure $\mathbf{A}$, there exists a homomorphism from $\mathbf{A}$ to $\mathbf{D}$ if and only if there is no homomorphism from any member $\mathbf{T}$ of $\mathcal{O}$ to $\mathbf{A}$.

The noteworthy homomorphism dualities typically correspond to efficient algorithms for constraint satisfaction problems. These include finite dualities (where the family $\mathcal{O}$ is finite), tree dualities (where $\mathcal{O}$ is a family of trees) and bounded treewidth dualities (where $\mathcal{O}$ is a family of structures with bounded treewidth). More examples are discussed in [2].

"Characterizing dualities" may refer to two distinct types of problems.

- Characterizing targets: Deciding, given a structure $\mathbf{D}$, whether there exists a family $\mathcal{O}_\mathbf{D}$ of structures in a given class (e.g. trees) such that $(\mathcal{O}_\mathbf{D}, \mathbf{D})$ is a duality.
- Characterizing obstruction sets: Deciding, given a family $\mathcal{O}$, whether there exists a structure $\mathbf{D}_\mathcal{O}$ such that $(\mathcal{O}, \mathbf{D}_\mathcal{O})$ is a duality.

The two problems are different. In the case of finite dualities, the characterization of obstruction sets was obtained in 2000 ([9]), and that of targets in 2007 ([8]). The problem of characterizing targets was solved in 1998 ([7]) for tree dualities, and recently in 2009 ([1]) for bounded treewidth dualities. Characterizing obstruction sets remains an open problem both for tree duality and bounded treewidth duality.

The difficulty in characterizing obstruction sets may depend on how the obstructions are represented. In the case of finite dualities, an explicit description of the obstructions is always possible. For infinite families of obstructions, fragments of

Date: April 6, 2013.

2000 *Mathematics Subject Classification.* 68Q19 (05C05,08B70).

Key words and phrases. constraint satisfaction problems, caterpillar duality, regular languages.

The first author's work was supported in part by the Hungarian NSF, under contract NK 78439 and K 68262. The second author's work was supported by grants from NSERC and ARP. The third author's work was supported in part by the NSERC grant 329527 by the Hungarian OTKA grants T-046234, AT048826 and NK-62321.

the Datalog language have proved to be an efficient tool to describe families of obstructions implicitly, through their homomorphic images. The structures with tree duality and bounded treewidth duality all have obstruction sets that can be described in Datalog.

In [3], Carvalho, Dalmau and Krokhin introduced caterpillar dualities as the dualities $(\mathcal{O}, \mathbf{D})$ where $\mathcal{O}$ is describable in the smallest natural recursive fragment of Datalog, namely "monadic linear Datalog with at most one EDB per rule" (see Section 4). They proved that the corresponding targets $\mathbf{D}$ are precisely those which are homomorphically equivalent to a structure with lattice polymorphisms, and that they are recognizable by the existence of a homomorphism of a given superstructure $\mathbf{C}(\mathbf{D})$ to $\mathbf{D}$ (see Section 5).

The purpose of the present paper is to complement the work of Carvalho, Dalmau and Krokhin by solving the characterization of obstructions problem for caterpillar dualities. We will consider a representation of caterpillars by words over a suitable alphabet, and show that caterpillar dualities correspond to regular languages. In particular, this shows that every program in "monadic linear Datalog with at most one EDB per rule" describes the obstruction set of a caterpillar duality. This extends some methods developed in [4] to study antichain dualities for digraphs.

We will provide the necessary background in the next section, and prove our main result in Section 3. The link with Datalog is given in Section 4, and relevant constructions and extensions are discussed in Section 5.

2. Preliminaries

Relational structures. A *type* is a finite set $\sigma = \{R_1, \dots, R_m\}$ of *relation symbols*, each with an *arity* r_i assigned to it. A σ-structure is a relational structure $\mathbf{A} = \langle A; R_1(\mathbf{A}), \dots, R_m(\mathbf{A})\rangle$ where A is a non-empty set called the *universe* of $\mathbf{A}$, and $R_i(\mathbf{A})$ is an r_i-ary relation on A for each i. The elements of $R_i(\mathbf{A})$, $1 \le i \le m$ will be called *hyperedges* of $\mathbf{A}$. By analogy with the graph theoretic setting, the universe of $\mathbf{A}$ will also be called its vertex-set, denoted $V(\mathbf{A})$.

A σ-structure $\mathbf{A}$ may be described by its bipartite *incidence multigraph* $\mathrm{Inc}(\mathbf{A})$ defined as follows. The two parts of $\mathrm{Inc}(\mathbf{A})$ are $V(\mathbf{A})$ and $\mathrm{Block}(\mathbf{A})$, where

$$\mathrm{Block}(\mathbf{A}) = \{(R, (x_1, \dots, x_r)) : R \in \sigma \text{ has arity } r \text{ and } (x_1, \dots, x_r) \in R(\mathbf{A})\},$$

and with edges $e_{a,i,B}$ joining $a \in V(\mathbf{A})$ to $B = (R, (x_1, \dots, x_r)) \in \mathrm{Block}(\mathbf{A})$ when $x_i = a$. Thus, the degree of $B = (R, (x_1, \dots, x_r))$ in $\mathrm{Inc}(\mathbf{A})$ is precisely r. Here "degree" means number of incident edges rather than number of neighbors because parallel edges are possible: If $x_i = x_j = a \in V(\mathbf{A})$, then $e_{a,i,B}$ and $e_{a,j,B}$ both join a and B. An element $a \in V(\mathbf{A})$ is called a *leaf* if it has degree one in $\mathrm{Inc}(\mathbf{A})$, and a *non-leaf* otherwise. Similarly, a block of $\mathbf{A}$ is called *pendant* if it is incident to at most one non-leaf, and *non-pendant* otherwise. A σ-structure $\mathbf{T}$ is called a *σ-tree* (or *tree* for short) if $\mathrm{Inc}(\mathbf{T})$ is a (graph-theoretic) tree, that is, it is connected and has no cycles or parallel edges. A σ-tree is called a *σ-path* if it has at most two pendant blocks. A σ-tree is called a *σ-caterpillar* if it is either a σ-path or it can be turned into a σ-path by removing all its pendant blocks (and the leaves attached to them). Figures 1 and 2 of Section 3 depict a caterpillar and its incidence multigraph.

Homomorphisms. For σ-structures $\mathbf{A}$ and $\mathbf{B}$, a *homomorphism* from $\mathbf{A}$ to $\mathbf{B}$ is a map $f : V(\mathbf{A}) \mapsto V(\mathbf{B})$ such that $f(R_i(\mathbf{A})) \subseteq R_i(\mathbf{B})$ for all $i = 1, \dots, m$, where for

any relation $R \in \sigma$ of arity r we have

$$f(R) = \{(f(x_1), \ldots, f(x_r)) : (x_1, \ldots, x_r) \in R\}.$$

We write $\mathbf{A} \to \mathbf{B}$ if there exists a homomorphism from $\mathbf{A}$ to $\mathbf{B}$, and $\mathbf{A} \not\to \mathbf{B}$ otherwise. We write $\mathbf{A} \leftrightarrow \mathbf{B}$ when $\mathbf{A} \to \mathbf{B}$ and $\mathbf{B} \to \mathbf{A}$; $\mathbf{A}$ and $\mathbf{B}$ are then called *homomorphically equivalent*. For a finite structure $\mathbf{A}$, we can always find a structure $\mathbf{B}$ such that $\mathbf{A} \leftrightarrow \mathbf{B}$ and the cardinality of $V(\mathbf{B})$ is minimal with respect to this property. It is well known (see [9]) that any two such structures are isomorphic. We then call $\mathbf{B}$ the *core* of $\mathbf{A}$.

Automata.

When the type σ consists only of binary relations, a σ-structure $\mathbf{A}$ is an edge-labeled directed multigraph. If we specify sets $I, T \subseteq V(\mathbf{A})$ of initial and terminal states respectively, we get a nondeterministic automaton $(\mathbf{A}, I, T)$. The type σ is then viewed as an alphabet. A word $w \in \sigma^*$ naturally corresponds to a directed σ-path P_w with $|w|$ edges with labels successively specified by the letters of w. A walk is a homomorphism $\phi : P_w \to \mathbf{A}$. If ϕ maps the first and last vertices of P_w to vertices in I and T respectively, then the word w is *accepted* by $(\mathbf{A}, I, T)$. The set of such words is called the *language accepted by* $(\mathbf{A}, I, T)$.

We recall a few basic facts from automata theory. The reader is referred to standard references (e.g. [11]) for a thorough treatment. A language $\mathcal{L} \subseteq \sigma^*$ is called *regular* if it is the language accepted by some nondeterministic automaton. It is well known that a language is regular if and only if it can be described by a "regular expression", that is, an expression constructed from letters in σ using unions, concatenation and the star operation. Regular languages are also preserved by other basic operations such as intersection and complementation.

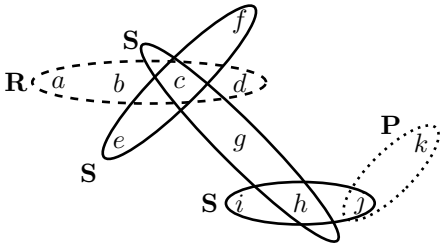
An automaton $(\mathbf{A}, I, T)$ is called *deterministic* if I is a singleton and for every $a \in V(\mathbf{A})$ and $R \in \sigma$, there is a unique $b \in V(\mathbf{A})$ such that $(a, b) \in R(\mathbf{A})$. It is well known that for every non-deterministic automaton $(\mathbf{A}, I, T)$, there exists a deterministic automaton $\Delta(\mathbf{A}, I, T)$ which accepts the same language.

3. Caterpillars

Graph-theoretic caterpillars consist of a path "body" to which are connected a number of pendant "leg" edges. The incidence multigraph of a σ-path $\mathbf{T}$ is a graph-theoretic caterpillar in which the body alternates between elements of $V(\mathbf{T})$ and of Block($\mathbf{T}$), and the legs connect elements of Block($\mathbf{T}$) to leaves. The incidence multigraph Inc($\mathbf{T}$) of a σ-caterpillar $\mathbf{T}$ can be transformed into a graph-theoretic caterpillar by removing the leaves of Inc($\mathbf{T}$) that are in $V(\mathbf{T})$ (corresponding to leaves *of* $\mathbf{T}$). The non-leaves of $\mathbf{T}$ are on the body of this graph-theoretic caterpillar. They can be linearly ordered $x_1, \ldots, x_n$ such that x_i, x_{i+1} are incident to one common non-pendant block B_i for $i = 1, \ldots, n-1$. The remaining blocks of $\mathbf{T}$ are pendant, and each of them is incident to one of $x_1, \ldots, x_n$. In this section we present a way to represent σ-caterpillars by words over a suitable alphabet.

Given a type σ, we define σ_2 as follows: For every $R \in \sigma$ of arity k and for every $(i, j) \in \{1, \ldots, k\}^2$, σ_2 contains the symbol $R^{(i,j)}$. Thus σ_2 can be viewed as an alphabet or as a type consisting of binary relations.

As an alphabet, σ_2 allows us to represent σ-caterpillars in a natural way: If $\mathbf{T}$ is a σ-caterpillar and $x_1, \ldots, x_n$ are its non-leaves with their natural ordering, then



$V(\mathbf{T}) = \{a, \ldots, k\}$
$\sigma = \{R, S, P\}$ with arities $4, 3, 2$.
$R = \{(a, b, c, d)\}$;
$S = \{(e, c, f), (c, g, h), (i, h, j)\}$;
$P = \{(j, k)\}$.

FIGURE 1. The σ-caterpillar $\mathbf{T}$

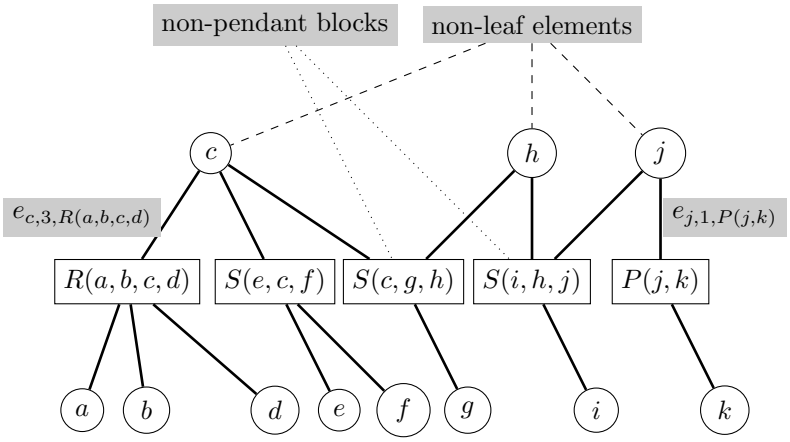


FIGURE 2. The incidence multigraph Inc($\mathbf{T}$)

$\sigma_2 = \{R^{11}, \ldots, R^{14}, R^{21}, \ldots, R^{44}, S^{11}, S^{12}, S^{13}, S^{21}, \ldots, S^{33}, P^{11}, P^{12}, P^{21}, P^{22}\}$

(A)

$R^{33}S^{22}S^{13}S^{23}P^{11}$

$S^{12}R^{33}S^{13}S^{23}P^{12}$

(B)

FIGURE 3. (a) The alphabet σ_2 (b) two of the many words representing $\mathbf{T}$

$\mathbf{T}$ corresponds to the σ_2-word

$$X_1 L_1 X_2 L_2 X_3 \cdots X_{n-1} L_{n-1} X_n,$$

where X_i is the concatenation of all $R^{(j,j)}$s such that $\mathbf{T}$ has a pendant block $(R, (a_1, \ldots, a_k))$ with $a_j = x_i$, and L_i is $R^{(j,k)}$ such that $\mathbf{T}$ has a non-pendant block $(R, (a_1, \ldots, a_\ell))$ with $a_j = x_i$, $a_k = x_{i+1}$. A σ-caterpillar consisting of a single block $(R, (a_1, \ldots, a_k))$ can be represented by any letter of the letters $R^{(i,j)}$, and the σ-caterpillar consisting of one vertex and no blocks is represented by the empty word. In general, different words may represent the same σ-caterpillar. However a σ-caterpillar may be retrieved from any word representing it, as detailed below.

Let β be the natural construction which takes a σ-structure $\mathbf{A}$ and produces a corresponding σ_2-structure $\beta(\mathbf{A})$. That is, $V(\beta(\mathbf{A})) = V(\mathbf{A})$, and for $R \in \sigma$ and $(x_1, \ldots, x_k) \in R(\mathbf{A})$, we put $(x_i, x_j) \in R^{(i,j)}(\beta(\mathbf{A}))$ for all $(i, j) \in \{1, \ldots, k\}^2$. Then β is a functor from the category of σ-structures to that of σ_2-structures, that is, if $\mathbf{A} \to \mathbf{B}$, then $\beta(\mathbf{A}) \to \beta(\mathbf{B})$. The converse does not hold in general. However, the definition of the functor β fits the general mold of the right adjoint functors considered in [6, 10]. Therefore there exists a corresponding left adjoint β^* from the category of σ_2-structures to that of σ-structures, with the following property:

Theorem 3.1 ([10])**.** *For a σ_2-structure $\mathbf{A}$ and a σ-structure $\mathbf{B}$, we have*

$$\mathbf{A} \to \beta(\mathbf{B}) \Leftrightarrow \beta^*(\mathbf{A}) \to \mathbf{B}.$$

We now explain the construction of $\beta^*(\mathbf{A})$ (given in [6, 10]). We first construct an auxiliary structure $\beta^*(\mathbf{A})^+$. For each element $x \in V(\mathbf{A})$, $V(\beta^*(\mathbf{A})^+)$ contains a corresponding (isolated) element x', and for each $(x, y) \in R^{(i,j)}(\mathbf{A})$, $V(\beta^*(\mathbf{A})^+)$ contains additional elements $x_1, \ldots, x_k$ (where $R \in \sigma$ has arity k) and the hyperedge $(x_1, \ldots, x_k) \in R(\beta^*(\mathbf{A})^+)$. $\beta^*(\mathbf{A})$ is then the quotient $(\beta^*(\mathbf{A})^+)/\sim$ obtained through natural identifications. That is, for $(x, y) \in R^{(i,j)}(\mathbf{A})$ and the corresponding $(x_1, \ldots, x_k) \in R(\beta^*(\mathbf{A})^+)$, $\sim$ identifies x_i with x' and x_j with y'.

Given a word $w \in \sigma_2^*$, the corresponding σ_2-path $\mathbf{P}_w$ has $|w| + 1$ elements successively joined by the relations indicated by the letters of w. The *caterpillar represented by w* is then $\beta^*(\mathbf{P}_w)$. (Note that the construction of $\beta^*(\mathbf{P}_w) = (\beta^*(\mathbf{P}_w)^+)/\sim$ identifies elements corresponding to distinct elements of $\mathbf{P}_w$ whenever some $R^{(i,i)} \in \sigma_2$ occurs in w.)

Lemma 3.2. *Let σ be a type and $\mathbf{A}$ a σ-structure. Then the family of σ_2-words representing the σ-caterpillars that admit homomorphisms to $\mathbf{A}$ is a regular language.*

Proof. Let $\mathbf{T}$ be a σ-caterpillar, w a word representing it and $\mathbf{P}_w$ the σ_2-path corresponding to w. Then the adjunction property yields

$$\mathbf{P}_w \to \beta(\mathbf{A}) \Leftrightarrow \beta^*(\mathbf{P}_w) \to \mathbf{A}.$$

with $\beta^*(\mathbf{P}_w) \simeq \mathbf{T}$. Since $\beta(\mathbf{A})$ can be viewed as a nondeterministic automaton with all states being initial and terminal, this shows that the corresponding words w indeed constitute a regular language. □

Since the complement of a regular language is again regular, the family of σ-caterpillar obstructions of any σ-structure $\mathbf{A}$ is again represented by a regular language.

Theorem 3.3. *Let σ be a type, $\mathcal{L}$ a regular language over σ_2 and $\mathcal{O}$ the family of σ-caterpillars represented by $\mathcal{L}$. Then there exists a σ-structure $\mathbf{A}$ such that $(\mathcal{O}, \mathbf{A})$ is a homomorphism duality.*

Proof. Let $(\mathbf{D}, I, T)$ be a deterministic automaton which recognizes $\mathcal{L}$. We define the structure $\mathbf{A} = \Gamma(\mathbf{D}, I, T)$ as follows. $V(\mathbf{A})$ is the set of subsets of $V(\mathbf{D})$ containing the initial state but none of the terminal states. For a relation $R \in \sigma$ of arity k, $R(\mathbf{A})$ is defined as follows: We put $(X_1, \ldots, X_k) \in R(\mathbf{A})$ if for all $(i, j) \in \{1, \ldots, k\}^2$ and for all $a \in X_i$, the unique b such that $(a, b) \in R^{(i,j)}(\mathbf{D})$ is in X_j.

Let $\mathbf{B}$ be a structure such that no σ-caterpillar represented by $\mathcal{L}$ admits a homomorphism to $\mathbf{B}$. Let w be a word over σ_2 such that there exists a homomorphism $\phi : \beta^*(\mathbf{P}_w) \to \mathbf{B}$. Then, ϕ induces a homomorphism $\phi_2 : \mathbf{P}_w \to \beta(\mathbf{B})$, and we denote $b_{w,\phi}$ the image of the last vertex of $\mathbf{P}_w$ under ϕ_2. Also, there is a unique homomorphism of $\mathbf{P}_w$ to $\mathbf{D}$ mapping the first element to the initial state, and we denote d_w the image of the last vertex of $\mathbf{P}_w$. Using every possible w and $\phi : \mathbf{T} \to \mathbf{B}$ we define a map $\psi : V(\mathbf{B}) \to \mathcal{P}(D)$ as follows. For an element b of $\mathbf{B}$, $\psi(b)$ is the set of all elements d_w such that $b = b_{w,\phi}$. Then $\psi(b)$ always contains the initial state (because the empty word represents the one-element σ-caterpillar with no hyperedges, which can be mapped to b) and never a terminal state (because otherwise $\mathbf{P}_w$ is an accepting path in $\mathbf{D}$, thus $w \in \mathcal{L}$ and hence $\mathbf{P}_w \not\to \mathbf{B}$). Thus ψ is a map from $V(\mathbf{B})$ to $V(\mathbf{A})$. We prove that it is a homomorphism of $\mathbf{B}$ to $\mathbf{A}$. Let R be a relation in σ of arity k, and $(b_1, \dots, b_k) \in R(\mathbf{B})$. For $(i, j) \in \{1, \dots, k\}^2$ and $d \in \psi(b_i)$, there exists a word w such that $d_w = d$ and there exists a homomorphism $\phi : \beta^*(\mathbf{P}_w) \to \mathbf{B}$ such that $b_{w,\phi} = b_i$. By appending $R^{(i,j)}$ to w, we get a new word w' such that $\phi : \beta^*(\mathbf{P}_w) \to \mathbf{B}$ naturally extends to $\phi' : \beta^*(\mathbf{P}_{w'}) \to \mathbf{B}$, with $b_{w,\phi'} = b_j$. Therefore the unique element $d_{w'}$ such that $(d_w, d_{w'}) \in R^{(i,j)}$ is in $\psi(b_j)$. This shows that ψ is a homomorphism.

Therefore, if no σ-caterpillar represented by $\mathcal{L}$ admits a homomorphism to $\mathbf{B}$, then $\mathbf{B}$ admits a homomorphism to $\mathbf{A}$. It remains to prove that no σ-caterpillar represented by $\mathcal{L}$ admits a homomorphism to $\mathbf{A}$. For $w \in \mathcal{L}$, suppose that there exists a homomorphism $\phi : \beta^*(\mathbf{P}_w) \to \mathbf{A}$. This corresponds to a homomorphism $\phi_2 : \mathbf{P}_w \to \beta(\mathbf{A})$. Since the initial state is in the image of the first element of $\mathbf{P}_w$, a terminal state is in the image of its last element, which is impossible. □

According to Theorem 3.3, for every regular σ_2-language $\mathcal{L}$, there exists a duality $(\mathcal{O}, \mathbf{A})$ such that $\mathcal{O}$ is the family of σ-caterpillars represented by $\mathcal{L}$. However $\mathcal{L}$ may be smaller than the set $\mathcal{L}^+$ of all words representing σ-caterpillar obstructions to $\mathbf{A}_\mathcal{L}$; however by Lemma 3.2, $\mathcal{L}^+$ is also regular (since its complement is regular). Between $\mathcal{L}$ and $\mathcal{L}^+$ there are usually non-regular languages which also represent a complete set of obstructions to $\mathbf{A}$. There may even be such non-regular languages that do not contain $\mathcal{L}$. Therefore, the complete characterization of obstruction sets for caterpillar dualities may be stated as follows:

Theorem 3.4. *Let $\mathcal{L}$ be a σ_2-language, $\mathcal{O}$ the family of σ-caterpillars represented by $\mathcal{L}$, $\mathcal{O}^+$ the family of σ-caterpillars which contain homomorphic images of members of $\mathcal{O}$ and $\mathcal{L}^+$ the collection of words representing these σ-caterpillars. Then there exists a duality $(\mathcal{O}, \mathbf{A})$ if and only if $\mathcal{L}^+$ is regular.*

4. Caterpillar Datalog programs

A *caterpillar Datalog program* is a "monadic linear Datalog program with at most one EDB per rule", that is, a set of rules of the form

$$a \in \rho_i \leftarrow b \in \rho_j \text{ and } (x_1, \dots, x_k) \in R \text{ with } x_m = a, x_n = b. \tag{1}$$

Here R is a relation in a type σ of arity k (called an extensional database or EDB), and ρ_i, ρ_j are unary auxiliary relations that are not in σ and that will be defined recursively (they are called intensional databases or IDBs). The auxiliary relations are monadic, that is, unary, and the program is "linear" since at most one auxiliary relation is used in the condition on the right side of the arrow. (See [7] for a

description of general Datalog programs.) In addition, the first rule is a formal initialization:

$$a \in \rho_1 \leftarrow a = a, \tag{2}$$

and there are terminal rules of the form

$$\text{goal} \leftarrow a \in \rho_i. \tag{3}$$

A Datalog program is usually seen as a way to construct relations $\rho_1, \rho_2, \ldots$ in a σ-structure $\mathbf{B}$ recursively, by a repeated application of the rules that apply, until a certain "goal" is achieved. Note that in a caterpillar Datalog program, all the rules can be rewritten in terms of the type σ_2: The rule 1 can be written

$$a \in \rho_i \leftarrow b \in \rho_j \text{ and } (b, a) \in R^{(n,m)}. \tag{4}$$

In this modified form, the program can be executed in $\beta(\mathbf{B})$. We see that the "goal" is achieved when a certain σ_2-walk is found in $\beta(\mathbf{B})$, which corresponds to finding a homomorphic image of the corresponding σ-caterpillar in $\mathbf{B}$.

Therefore, a caterpillar Datalog program will achieve its goal on the structures which contain homomorphic images of σ-caterpillars belonging to a certain family. To see that this family is regular, we consider the nondeterministic automaton $(\mathbf{C}, I, T)$ of type σ_2 described by the rules of the programs: $V(\mathbf{C})$ is the set of IDBs of the program, and for each rule

$$a \in \rho_i \leftarrow b \in \rho_j \text{ and } (b, a) \in R^{(n,m)}$$

we put $(\rho_j, \rho_i) \in R^{(n,m)}(\mathbf{C})$. We put $I = \{\rho_1\}$, and the terminal states are the states ρ_i appearing in terminal rules. Thus a goal-achieving derivation in a structure $\mathbf{B}$ must correspond to a word accepted by $(\mathbf{C}, I, T)$, and the family of such words is regular. Combining this with Theorem 3.3 we get the following.

Theorem 4.1. *For every caterpillar Datalog program, there exists a structure* $\mathbf{A}$ *such that an input structure* $\mathbf{B}$ *admits a homomorphism to* $\mathbf{A}$ *if and only if the program does not achieve its goal on* $\mathbf{B}$.

5. Characterization of targets with caterpillar duality

For a type σ, a regular σ_2-language $\mathcal{L}$ may be described by a regular expression, an automaton (deterministic or nondeterministic) which recognizes it or a caterpillar Datalog program. The previous section explains how to convert a caterpillar Datalog program into a nondeterministic automaton which recognizes the same language. We refer to [11] for the conversion from regular expression to automaton, and for the construction Δ which takes a nondeterministic automaton $(\mathbf{B}, I, T)$ and constructs a deterministic automaton $(\mathbf{D}, I', T') = \Delta(\mathbf{B}, I, T)$ which accepts the same language. Thus, if the regular σ_2-language $\mathcal{L}$ is recognized by the automaton $(\mathbf{B}, I, T)$, then the corresponding caterpillar duality is $(\mathcal{O}, \mathbf{A})$, where $\mathcal{O}$ is the family of σ-caterpillars represented by $\mathcal{L}$ and $\mathbf{A} = \Gamma \circ \Delta(\mathbf{B}, I, T)$, Γ being the construction described in the proof of Theorem 3.3.

Now for any σ-structure $\mathbf{A}$, $(\beta(\mathbf{A}), V(\mathbf{A}), V(\mathbf{A}))$ is a nondeterministic automaton which recognizes the σ_2-language of words representing σ-caterpillars which admit a homomorphism to $\mathbf{A}$, and $\Delta(\beta(\mathbf{A}), V(\mathbf{A}), V(\mathbf{A}))$ is a deterministic automaton which serves the same purpose. Let $\Delta^*(\beta(\mathbf{A}), V(\mathbf{A}), V(\mathbf{A}))$ be the deterministic automaton obtained from $\Delta(\beta(\mathbf{A}), V(\mathbf{A}), V(\mathbf{A}))$ by interchanging the

set of terminal states with its complement. Then $\Delta^*(\beta(\mathbf{A}), V(\mathbf{A}), V(\mathbf{A}))$ is a deterministic automaton which recognizes the σ_2-language of words representing the set $\mathcal{O}$ of σ-caterpillars which do not admit a homomorphism to $\mathbf{A}$, and $(\mathcal{O}, \Gamma \circ \Delta^*(\beta(\mathbf{A}), V(\mathbf{A}), V(\mathbf{A})))$ is the corresponding caterpillar duality, which has the following properties.

Theorem 5.1. $\mathbf{C}(\mathbf{A}) = \Gamma \circ \Delta^*(\beta(\mathbf{A}), V(\mathbf{A}), V(\mathbf{A}))$ *has caterpillar duality, and for any* σ*-structure* $\mathbf{B}$ *with caterpillar duality, there exists a homomorphism of* $\mathbf{A}$ *to* $\mathbf{B}$ *if and only if there exists a homomorphism of* $\mathbf{C}(\mathbf{A})$ *to* $\mathbf{B}$. *In particular,* $\mathbf{A}$ *itself has caterpillar duality if and only if there exists a homomorphism of* $\mathbf{C}(\mathbf{A})$ *to* $\mathbf{A}$.

This is essentially the characterization obtained in [3]. Note that Δ^* and Γ are both exponential constructions, so that $\mathbf{C}$ is a doubly exponential construction.

With a slight modification, the same type of characterization also holds for caterpillar dualities with additional properties. The most distinctive case is that of path dualities, where the obstructions are σ-paths. The σ-paths correspond to σ_2-words not containing any of the symbols $R^{(i,i)}$ except possibly as first or last letter. For a type σ, let $\sigma_2 = \sigma_{2,0} \cup \sigma_{2,1}$, where $\sigma_{2,0}$ contains all the symbols $R^{(i,i)}$ and $\sigma_{2,1}$ all the symbols $R^{(i,j)}$, $i \neq j$. Define $\mathcal{L}_P \subseteq \sigma_2^*$ by $\mathcal{L}_P = (\{\epsilon\} \cup \sigma_{2,0}) \circ \sigma_{2,1}^* \circ (\{\epsilon\} \cup \sigma_{2,0})$. For a σ-structure $\mathbf{A}$, let $\mathcal{L}_\mathbf{A}$ be the language representing the σ-caterpillar obstructions to $\mathbf{A}$. Then $\mathcal{L}_\mathbf{A}$ and $\mathcal{L}_P$ are regular languages, hence so is $\mathcal{L}_P \cap \mathcal{L}_\mathbf{A}$. Therefore with the construction Γ we can build a structure $\mathbf{C}_P(\mathbf{A})$ such that $\mathbf{A}$ has path duality if and only if there exists a homomorphism of $\mathbf{C}_P(\mathbf{A})$ to $\mathbf{A}$. A similar statement holds for any intersection $\mathcal{L} \cap \mathcal{L}_\mathbf{A}$, where $\mathcal{L} \subseteq \sigma_2^*$ is a regular language.

References

[1] L. Barto, M. Kozik, Constraint satisfaction problems of bounded width, sl Proc. 50th IEEE Symp. Foundations of Computer Science, FOCS'09 (2009), 595–603.

[2] A. Bulatov, A. Krokhin, B. Larose, Dualities for Constraint Satisfaction Problems, *Complexity of Constraints* **LNCS 5250** (2008), 93–124.

[3] C. Carvalho, V. Dalmau, A. Krokhin, Caterpillar Duality for Constraint Satisfaction Problems, *Proc. 23rd IEEE Symp. on Logic in Computer Science LICS'08* (2008), 307–316.

[4] P. L. Erdős, C. Tardif, G. Tardos, On infinite-finite duality pairs of directed graphs, manuscript (2012).

[5] P.L. Erdős, D. Pálvölgyi, C. Tardif, G. Tardos, On infinite-finite tree-duality pairs of relational structures, manuscript (2012).

[6] J. Foniok, C. Tardif, Adjoint functors and tree duality, *Discrete Mathematics and Theoretical Computer Science* **11** (2) (2009), 97–110.

[7] T. Feder, M. Vardi, The computational structure of monotone monadic SNP and constraint satisfaction: A study through Datalog and group theory, *SIAM J. of Computing* **28** (1998), 57–104.

[8] B. Larose, C. Loten, C. Tardif, A Characterisation of first order definable constraint satisfaction problems, *Log. Methods Comput. Sci.* **3** (4) (2007), paper 4:6, (22 pp.)

[9] J. Nešetřil, C. Tardif, Duality theorems for finite structures (Characterising gaps and good characterisations), *J. Combin. Theory* (B) **80** (2000), 80–97.

[10] A. Pultr, The right adjoints into the categories of relational systems. In *Reports of the Midwest Category Seminar, IV* **LNM 137** (1970), 100–113.

[11] M. Sipser, *Introduction to the Theory of Computation*, PWS Publishing Company, Boston, 1997.

Alfréd Rényi Institute of Mathematics, Hungarian Academy of Sciences, Budapest, P.O. Box 127, H-1364 Hungary

E-mail address: erdos.peter@renyi.mta.hu

Royal Military College of Canada, PO Box 17000 Station "Forces", Kingston, Ontario, Canada, K7K 7B4
E-mail address: Claude.Tardif@rmc.ca

Alfréd Rényi Institute of Mathematics, Hungarian Academy of Sciences, Budapest, P.O. Box 127, H-1364 Hungary
E-mail address: tardos.gabor@renyi.mta.hu