Fundamenta Informaticae 156 (2018) 1–33
DOI 10.3233/FI-2016-0000
IOS Press

Shelah-Stupp's iteration and Muchnik's iteration

Didier Caucal
LIGM-CNRS, Université Paris-Est
77454 Marne-la-Vallée Cedex 2, France
caucal@univ-nc.nc

Teodor Knapik*
ISEA, Université de la Nouvelle Calédonie,
98851 Nouméa Cedex, France
knapik@univ-nc.nc

Abstract. In the early seventies, Shelah proposed a model-theoretic construction, nowadays called "iteration". This construction is an infinite replication in a tree-like manner where every vertex possesses its own copy of the original structure. Stupp proved that the decidability of the monadic second-order (MSO) theory is transferred from the original structure onto the iterated one. In its extended version discovered by Muchnik and introduced by Semenov, the iteration became popular in computer science logic thanks to a paper by Walukiewicz. Compared to the basic iteration, Muchnik's iteration has an additional unary predicate which, in every copy, marks the vertex that is the clone of the possessor of the copy. A widely spread belief that this extension is crucial is formally confirmed in the paper. Two hierarchies of relational structures generated from finite structures by MSO interpretations and either Shelah-Stupp's iteration or Muchnik's iteration are compared. It turns out that the two hierarchies coincide at level 1. Every level of the latter hierarchy is closed under Shelah-Stupp's interation. In particular, the former hierarchy collapses at level 1.

Keywords: infinite-state systems, structure-building operations, Shelah-Stupp's iteration, Muchnik's iteration.

1. Introduction

Monadic second-order (MSO) logic is a restriction of second-order logic which generalises a number of temporal and program logics. Since many relevant properties can be expressed in MSO logic [1], looking for structures with decidable MSO theories has been an active

*Also afiliated as: IMSc, Chennai 600 113, Tamil Nadu, India.

research area. Its foundations have been laid by Büchi [2] (weak MSO) and independently by Trakhtenbrot [3] and Elgot [4] who in 1957-1962 established the decidability of the MSO theory of one successor over $\mathbb{N}$ and then by Rabin who in 1969 [5] established the decidability of the MSO theory of the infinite complete binary tree.

In 1975 Shelah [6] proposed a generalisation of Rabin's result where an infinite tree-like structure is built as output from any input structure and Stupp [7] proved that this construction, which we shall call Shelah-Stupp's iteration or the basic iteration, preserves the decidability of MSO theories. This result has been an important step in the development of MSO-compatible operations, viz., the transformations of structures that preserve decidability of its MSO theories. Other prominent MSO-compatible operations are generalised unions of Shelah [6] and MSO transductions of Courcelle [8] together with their restricted versions that are MSO interpretations. In 1979 Muchnik (Andrei Albertovich) introduced an extension of the basic iteration with a unary predicate and proved its MSO compatibility. Muchnik's unpublished proof is sketched in an invited lecture by Semenov [9]. Finally, the missing full proof has been written by Walukiewicz [10].

On top of above-mentioned Rabin's work, Muller and Shupp obtained several important results. In particular, they proved that transition graphs of pushdown automata have decidable MSO theories [11]. As established in [12], those graphs are precisely the simple graphs of finite degree among HR-equational hypergraphs of Courcelle [13] which belong to an even larger class of VR-equational hypergraphs [14] that also have decidable MSO theories. The latter class is closed under MSO-transductions as it is established by Barthelmann in [15] where he also shows that VR-equational graphs coincide up to isomorphism with prefix-recognisable graphs introduced by the first author [16].

Because isolated examples of graphs with a decidable MSO theory have been known since the end of sixties [17, 18, 19], it has been suggested that combining Shelah-Stupp's or Muchnik's iteration with MSO interpretations would lead to a larger family of structures with decidable MSO theories [20]. In [21] Courcelle and Walukiewicz prove that the unfolding is MSO-compatible. It is therefore suggested to seek for new classes of structures with decidable MSO theories by combining the unfolding and MSO interpretations. The first such class with an independent characterisation appears from [22, 23] as being the class of ranked infinite trees formerly introduced (with no relation to MSO theories) by Englefriet and Schmidt [24] and later studied by Damm [25] who have established that these form an infinite and strict hierarchy. Finally, a more general hierarchy is built [26] through the unfolding which allows one to climb up the hierarchy and through a restricted form of MSO interpretations which yields, from the trees of a given level, a variety of graphs of the same level. However, besides isolated examples, an even larger hierarchy may be obtained via MSO interpretation from the hierarchy of trees considered in [27, 28].

The MSO-compatible operations play an essential role in establishing the decidability of MSO theories. Both the unfolding and Muchnik's iteration allow one to climb up the hierarchy of [26] and have been used in many proofs. It has been observed by several authors that the unfolding is MSO-interpretable within Muchnik's iteration. Shelah-Stupp's iteration is also very useful when one needs an arbitrary number of copies of the original structure. It only differs from Muchnik's iteration by the lack of a unary predicated often called «clone». There is a widely spread belief that this unary predicate plays a crucial role in the expressive power of Muchnik's iteration. In the present paper we closely examine this belief which is further confirmed by the main result.

Inspired by the hierarchy of graphs introduced by the first author [26], we consider an analogous hierarchy of (directed) hypergraphs or relational structures. Within this hierarchy one climbs from level n to level $n+1$ via Muchnik's iteration and expands on a given level using MSO interpretation, starting from level 0 which consists of finite relational structures. We also consider a similarly defined hierarchy where instead of Muchnik's iteration, one uses Shelah-Stupp's iteration. We show that the two hierarchies coincide at level 1 but not beyond. Whereas the first hierarchy is strict, we show that the second hierarchy collapses at level 1.

In order to show the collapse, we need first an insight into level 1. After some reminders on MSO logic in Sect. 2 and definitions of the two hierarchies in Sect. 3, we review suffix-regular expressions and we show that these characterise the structures of level 1. Suffix-regular expressions extend regular expressions and may be considered as standard syntactic description of structures at this level. We then focus in Sect. 4 on the main result which says that, except level 0, every level of the first hierarchy is closed under Shelah-Stupp's iteration. This is established by induction. The induction basis follows from our syntactic characterisation of level 1. For the induction step, we consider a structure of level n, with $n \geq 2$, on which Shelah-Stupp's iteration is applied. As any level n structure is obtained from a structure of level $n-1$ through Muchnik's iteration and MSO interpretation, the essential part of the proof consists in pushing the application of Shelah-Stupp's iteration down to level $n-1$ so that it is applied before Muchnik's iteration. This is possible by performing several additional MSO interpretations. Since the latter do not affect the level of the resulting structure, while Shelah-Stupp's iteration applies on level $n-1$, we may use induction hypothesis to close the proof. The collapse of Shelah-Stupp's iteration entails, inter alia, the impossibility of defining the unfolding by combining that iteration with MSO interpretations.

2. Monadic second-order logic

Iteration in the restricted or general version is an MSO-compatible operation that acts on relational structures. We are therefore interested in MSO logic over such structures. These are (directed) hypergraphs or, if there is no relation of arity greater than 2, (directed) graphs. Two choices are possible for the latter. One may define a 2-sorted structure with a set of vertices, a set of edges and the incidence relation. One may also define (labelled) edges as relation on the set of vertices. The two definitions lead often to different results (see e.g. [8]). For relational structures considered here as directed hypergraphs, we adopt the simpler, namely the second view.

Let Σ be a ranked signature where each relation symbol $a \in \Sigma$ has its arity $\alpha(a) \in \mathbb{N}$. A relational structure $\mathfrak{A}$ over Σ, also called Σ-structure, is a set of hyperedges. Each hyperedge has its label in $a \in \Sigma$. A hyperedge $a(\overline{s}) \in \mathfrak{A}$ with $\overline{s} = (s_1,\dots,s_{\alpha(a)})$ links its vertices $s_1,\dots,s_{\alpha(a)}$ which need not to be all pairwise distinct. A hyperedge of arity 2 is an edge and instead of $a(s,s')$, we write $s \xrightarrow{a} s'$. We write $\mathfrak{A}(a)$ for the set of hyperedges of $\mathfrak{A}$ labelled by $a \in \Sigma$ and $V_{\mathfrak{A}}$ for the set of vertices of $\mathfrak{A}$:

$$V_{\mathfrak{A}} := \bigcup_{a(s_1,\ldots,s_{\alpha(a)})\in\mathfrak{A}} \{s_1,\ldots,s_{\alpha(a)}\} \ .$$

The reader may observe that according to the above definition no relational structure may have isolated vertices. To overcome this drawback, it is enough to add to the signature a distinguished unary predicate symbol, say ρ, for labelling isolated vertices.

A path $s \overset{w}{\dashrightarrow} t$ in $\mathfrak{A}$ from $s \in V_{\mathfrak{A}}$ to $t \in V_{\mathfrak{A}}$, labelled by a word w over a set of symbols from Σ of arity 2 and possibly 1, is defined inductively by

$$\begin{array}{ll} s \overset{\varepsilon}{\dashrightarrow} t, & \text{if } s = t, \\ s \overset{cu}{\dashrightarrow} t, & \text{if } c(s) \text{ and } s \overset{u}{\dashrightarrow} t, \\ s \overset{bu}{\dashrightarrow} t, & \text{if there exists } r \in V_{\mathfrak{A}} \text{ s.t. } s \xrightarrow{b} r \text{ and } r \overset{u}{\dashrightarrow} t \end{array}$$

The syntax of MSO logic over relational structures is defined like for first-order (FO) logic but has, in addition to FO variables written in lower-case $x, y, z, x', x_1, \ldots$, set variables $X, Y, Z, X', X_1, \ldots$ written in capitals. Beyond the usual atomic FO formulae $a(\overline{x})$ for $a \in \Sigma$, there are membership formulae $x \in X$ each of which involve one FO variable and one MSO variable. More general formulae are constructed in the standard way using connectives and quantifiers which may be FO e.g. $\forall x, \exists x$ or MSO e.g. $\forall X, \exists X$. Note that the equality symbol is unnecessary since the identity relation is MSO definable:

$$x = y \quad :\Leftrightarrow \quad \forall X\ (x \in X \Leftrightarrow y \in X) \ .$$

In formulae, we shall sometimes use true, where

$$\mathsf{true} \quad :\Leftrightarrow \quad \forall x\ \ x = x \ .$$

The semantics of MSO logic is defined like for FO, except that set variables range over subsets of the structure. A relational structure $\mathfrak{A}$ satisfies $\exists X \varphi(X)$ if there exists a subset V of $V_{\mathfrak{A}}$ such that $(\mathfrak{A}, V) \vDash \varphi(X)$, viz. $\varphi(X)$ holds in $\mathfrak{A}$ when X is interpreted as subset V.

A well known fact already mentioned in the introduction is that reachability is not FO-definable. More generally, except in particular cases, one cannot define in FO a transitive closure of an FO-definable relation. On the other hand, for any MSO formula $\varphi(x,y)$, the following formula $\mathsf{Tr}_\varphi(x,y)$ defines the reflexive-transitive closure of the binary relation defined by $\varphi(x,y)$:

$$\mathsf{Tr}_\varphi(x,y) \quad :\Leftrightarrow \quad \forall X \Big(\big(x \in X \land \forall x' \forall y' \big((x' \in X \land \varphi(x',y')) \Rightarrow y' \in X \big) \big) \Rightarrow y \in X \Big) \ .$$

Using the latter, for any regular expression $\mathcal{E}$ over a set of symbols of arity 2, one may write an MSO formula $\mathsf{path}_{\mathcal{E}}(x,y)$ suitable for graphs saying that there is a path from one vertex to another labelled by word in the regular language denoted by $\mathcal{E}$. The formula is defined inductively according to the structure of $\mathcal{E}$:

$$
\begin{array}{rcl}
\mathsf{path}_\varnothing(x,y) & :\Leftrightarrow & \neg\mathsf{true}, \\
\mathsf{path}_a(x,y) & :\Leftrightarrow & x \xrightarrow{a} y, \\
\mathsf{path}_r(x,y) & :\Leftrightarrow & r(x) \wedge x = y, \qquad \text{for a unary symbol } r \\
\mathsf{path}_{\mathcal{E}_1+\mathcal{E}_2}(x,y) & :\Leftrightarrow & \mathsf{path}_{\mathcal{E}_1}(x,y) \vee \mathsf{path}_{\mathcal{E}_2}(x,y), \\
\mathsf{path}_{\mathcal{E}_1\mathcal{E}_2}(x,y) & :\Leftrightarrow & \exists\, z \left(\mathsf{path}_{\mathcal{E}_1}(x,z) \wedge \mathsf{path}_{\mathcal{E}_2}(z,y)\right), \\
\mathsf{path}_{\mathcal{E}^*}(x,y) & :\Leftrightarrow & \mathsf{Tr}_{\mathsf{path}_{\mathcal{E}}}(x,y).
\end{array}
$$

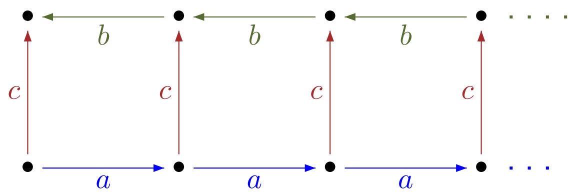
The reader may consult [8] for more examples of MSO-definable relations or properties.

An *MSO interpretation* is a structure-building operation that defines a Ω-structure $\mathfrak{B}$ within a given Σ-structure $\mathfrak{A}$ by means of MSO formulae over Σ. Formally, an MSO interpretation h is given as a *definition scheme* which is a tuple $\langle \delta, (\theta_b)_{b\in\Omega}\rangle$ where δ is an MSO formula with one free FO variable and each θ_b is an MSO formula with $\alpha(b)$ free FO variables. Then $\mathfrak{B} = h(\mathfrak{A})$ is defined as follows:

$$
\begin{array}{rcl}
V_{\mathfrak{B}} & := & \{d \in V_{\mathfrak{A}} \mid (\mathfrak{A}, d) \models \delta(x)\} \\
\mathfrak{B}(b) & := & \{b(\overline{d}) \mid (\mathfrak{A}, \overline{d}) \models \theta_b(\overline{x})\} \qquad \text{for each } b \in \Omega.
\end{array}
$$

We also say that θ_b *defines the relation* b *within* $\mathfrak{A}$. More generally, an n-ary relation ϱ is *MSO-definable within* a structure $\mathfrak{A}$, if there exists an MSO formula $\theta(\overline{x})$ with n free FO variables $\overline{x}$ such that $\varrho = \{\overline{d} \mid (\mathfrak{A}, \overline{d}) \models \theta(\overline{x})\}$.

Example 2.1. Consider an infinite complete binary tree $\mathfrak{T}_{\{0,1\}}$ as a structure over a signature $\Gamma := \{0,1\}$ with $\alpha(0) = \alpha(1) = 2$ where 0 (resp. 1) is the left (resp. right) successor relation. We interpret within $\mathfrak{T}_{\{0,1\}}$ a graph over $\Omega := \{a,b,c\}$ which is a sort of ladder:



An interpretation is given by the definition scheme $\langle \delta(x), \theta_a(x,y), \theta_b(x,y), \theta_c(x,y)\rangle$ where

$$
\begin{array}{rcl}
\delta(x) & :\Leftrightarrow & \exists\, r \left(\neg\exists\, z\ (z \xrightarrow{0} r \vee z \xrightarrow{1} r) \wedge \mathsf{path}_{0^*(\varepsilon+1)}(r,x)\right), \\
\theta_a(x,y) & :\Leftrightarrow & x \xrightarrow{0} y, \\
\theta_b(x,y) & :\Leftrightarrow & \exists\, z_1\ \exists\, z_2\ (z_1 \xrightarrow{0} z_2 \wedge z_1 \xrightarrow{1} y \wedge z_2 \xrightarrow{1} x), \\
\theta_c(x,y) & :\Leftrightarrow & x \xrightarrow{1} y.
\end{array}
$$

Observe that $\theta_a(x,y), \theta_b(x,y), \theta_c(x,y)$ are FO and that $\delta(x)$ which is MSO selects the nodes of the leftmost branch of the tree as well as the immediate right successors of the nodes of this branch.

3. Iteration hierarchy

We introduce in this section a hierarchy of relational structures which is built using iteration and MSO interpretations. We also review level-n pushdown automata since these are closely related to the hierarchy. Finally we study level-1 structures. We show that the two iterations lead to the same level-1 class of structures. We also give an algebraic characterisation of level-1 concrete structures following prefix-recognisable graphs introduced in [16]. By concrete, we mean that the elements of the structure are encoded as words over a finite alphabet.

3.1. Shelah-Stupp's iteration and Muchnik's iteration

We now recall the definition of the iteration introduced by Shelah and the extension by Muchnik.

Definition 3.1. (Iteration)
Given a relational structure $\mathfrak{A}$ over Σ and a new binary relation symbol $\sharp \notin \Sigma$, the *basic (or Shelah-Stupp's) iteration of* $\mathfrak{A}$, written $\mathfrak{A}^\sharp$, is the following relational structure over $\Sigma \cup \{\sharp\}$:

$$\begin{aligned}\mathfrak{A}^\sharp \quad := \quad & \{a(ws_1,\ldots,ws_{\alpha(a)}) \mid a \in \Sigma \wedge w \in V_{\mathfrak{A}}^* \wedge a(s_1,\ldots,s_{\alpha(a)}) \in \mathfrak{A}\} \\ & \cup \{w \xrightarrow{\sharp} ws \mid w \in V_{\mathfrak{A}}^+ \wedge s \in V_{\mathfrak{A}}\} \ .\end{aligned}$$

Muchnik's iteration of $\mathfrak{A}$ is defined like the basic iteration $\mathfrak{A}^\sharp$ extended with a unary predicate $\& \notin \Sigma \cup \{\sharp\}$:

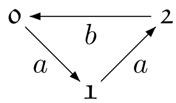
$$\mathfrak{A}^{\sharp,\&} \quad := \quad \mathfrak{A}^\sharp \cup \{\&(wss) \mid w \in V_{\mathfrak{A}}^* \wedge s \in V_{\mathfrak{A}}\} \ .$$

The iteration may be understood as an operation which builds a structure made of a countable number of copies of the original structure. The vertices of each copy are encoded by words over the alphabet that is precisely the set of the vertices of the original structure. Within a given copy all encoding words have the same length and differ only by the last letter that indicates the original vertex. Every vertex possesses its own private copy of the original structure. Within each copy, the common prefix of words encoding vertices is precisely the encoding of the vertex in another copy (or in the original) that «owns» the copy. Walukiewicz [10] made popular the name of «son» for the relation

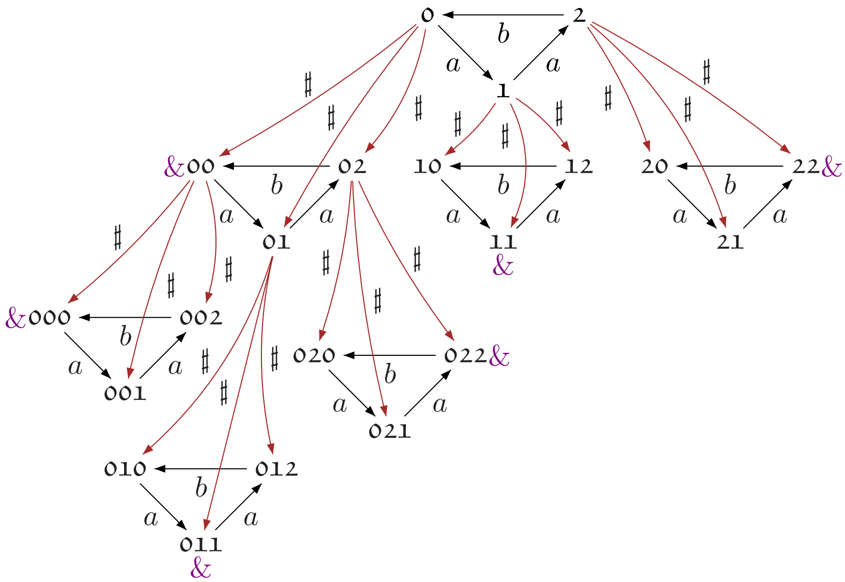
$$\{(w,ws) \mid w \in V_{\mathfrak{A}}^+ \wedge s \in V_{\mathfrak{A}}\}$$

which may be represented by edges from a vertex to all vertices of the private copy of the original structure owned by the vertex. Instead of «son» we use an arbitrary symbol to label this relation, mostly $\sharp$ or \$. Similarly we use an arbitrary label, mostly &, for the vertices of the form wss with $w \in V_{\mathfrak{A}}^+$ instead of the popular name «clone». Note that there is only one so marked vertex in each copy which is precisely the image of the «owner» ws under an isomorphism $f(wt) = wst$ where $t \in V_{\mathfrak{A}}$. The marked vertex wss is a sort of exact copy of the owner ws.

Example 3.2. We consider a graph $\mathfrak{G}$ with 3 vertices $\{0, 1, 2\}$ and 3 edges with labels in $\{a, b\}$ depicted as follows:



Muchnik's iteration $\mathfrak{G}^{\sharp,\&}$ of the above graph is an infinite structure, a portion of each is depicted below.



The basic iteration $\mathfrak{G}^{\sharp}$ of $\mathfrak{G}$ is depicted similarly. It only lacks &-labels.

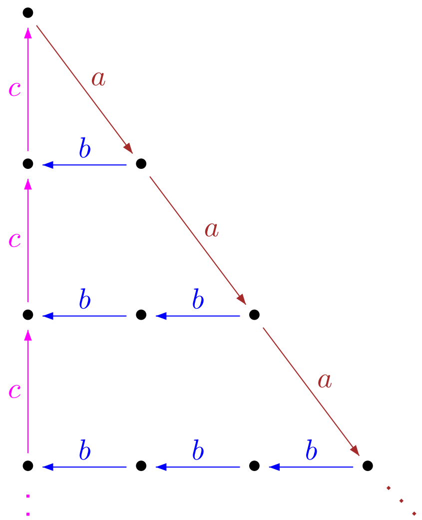
3.2. The hierarchy

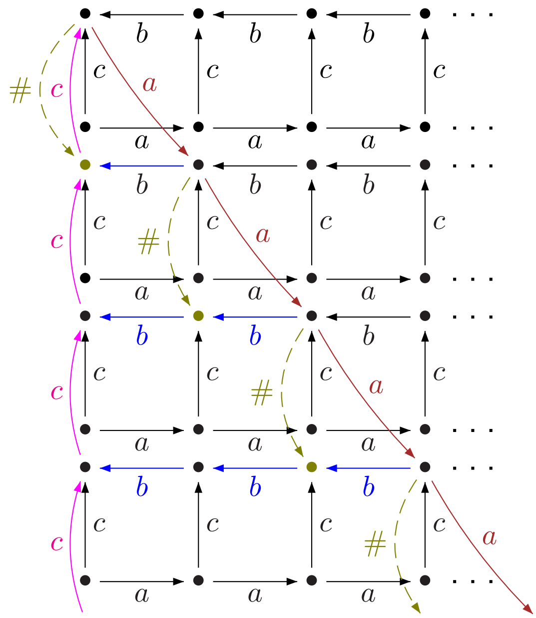
In [26] a hierarchy of infinite graphs is defined in terms of two graph operations: the unfolding and the inverse regular mapping. In [29], Carayol and Wöhrle show that this hierarchy can be alternatively defined in terms of Muchnik's iteration and MSO-interpretation. By allowing symbols of arity higher than 2, the latter definition is extended to arbitrary relational structures.

Definition 3.3. (Iteration hierarchy)
For every $n \in \mathbb{N}$ we define a family $\mathfrak{Hgr}_n$ of relational structures (or hypergraphs) of level n as follows

$$\mathfrak{Hgr}_0 \text{ is the family of finite relational structures,}$$
$$\mathfrak{Hgr}_{n+1} := \{f(\mathfrak{A}^{\sharp,\&}) \mid \mathfrak{A} \in \mathfrak{Hgr}_n \wedge f \text{ is an MSO interpretation}\}.$$

Example 3.4. The ladder from Example 2.1, say $\mathfrak{L}$, is in $\mathfrak{Hgr}_1$. The next construction shows that the following triangle on the left





belongs to $\mathfrak{Hgr}_2$. Indeed, this triangle may be interpreted within $\mathfrak{L}^{\#,\&}$ (depicted on the right with the triangle superimposed on it) via the scheme

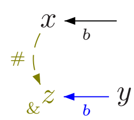
$$\langle \delta(x), \theta_a(x,y), \theta_b(x,y), \theta_c(x,y) \rangle$$

where

$$\begin{aligned}
\delta(x) \quad &:\Leftrightarrow \quad \exists y \big(\neg\exists z\,(z \xrightarrow{\#} y \vee y \xrightarrow{b} z \vee y \xrightarrow{c} z) \wedge \exists z\,(\mathsf{Tr}_{\theta_a}(y,z) \wedge \mathsf{path}_{b^*}(z,x))\big),\\
\theta_a(x,y) \quad &:\Leftrightarrow \quad \exists z \big(\&(z) \wedge x \xrightarrow{\#} z \wedge y \xrightarrow{b} z\big),\\
\theta_b(x,y) \quad &:\Leftrightarrow \quad x \xrightarrow{b} y,\\
\theta_c(x,y) \quad &:\Leftrightarrow \quad \neg\exists z\,(x \xrightarrow{b} z \vee x \xrightarrow{c} z) \wedge \neg\exists z\,(y \xrightarrow{b} z \vee y \xrightarrow{c} z)\\
&\qquad\quad \wedge \exists z \big(\mathsf{path}_{\#\&b^*}(z,x) \wedge \mathsf{path}_{b^*}(z,y)\big)\,.
\end{aligned}$$

On a simplified picture of $\mathfrak{L}^{\#,\&}$ we only represent copies possessing vertices selected by $\delta(x)$ and relevant $\#$-labelled edges, namely those with &-labelled target (label & omitted). In the original ladder $\mathfrak{L}$, $\delta(x)$ selects the unique vertex with no outgoing edges as "input" for the reflexive-transitive closure collecting vertices of an a^*-labelled path (cf. $\mathsf{Tr}_{\theta_a}(y,z)$) of the resulting graph. From this collection, vertices accessible by b^*-labelled paths are selected.

New edges $x \xrightarrow{a} y$ correspond to the pattern depicted on the right. Although these edges involve both relevant and irrelevant copies, the latter are eliminated as only the vertices reachable from the unique sink of $\mathfrak{L}$ by new a^*b^*-labelled paths are selected for the resulting structure by δ.



Every new edge $x \xrightarrow{c} y$ goes from the sink of a copy to the sink of its owner but similarly to new a-labelled edges, only relevant copies will be finally concerned due to selection by δ.

3.3. Level-n pushdown automata

In [29], Carayol and Wöhrle show that, up to an ε-closure,[1] the hierarchy of graphs of [26], may be characterised via transition graphs of level-n pushdown automata (n-pda for short). A 1-pda, is a standard pushdown automaton. Instead of a usual pushdown store (pds for short), a 2-pda has a level-2 pds, each element of which is a usual (level-1) pds. Within a level-2 pds, the topmost level-1 pds may be accessed by means of standard pushdown operations but in addition it may be removed or duplicated. An n-pda is obtained by generalising this idea to any level n.

Formally, level-n (resp. non empty level-n) pds, written Γ_*^n (resp. Γ_+^n) over Γ is defined by

$$\begin{array}{rclcrcl} \Gamma_*^0 & := & \Gamma, & \qquad & \Gamma_+^0 & := & \Gamma, \\ \Gamma_*^{k+1} & := & (\Gamma_*^k)^*, & & \Gamma_+^{k+1} & := & (\Gamma_*^k)^+. \end{array}$$

Note that if $V \subseteq \Gamma_*^n$ then $V_*^m \subseteq \Gamma_*^{m+n}$. If $u \in \Gamma_*^k$ and $s \in \Gamma_*^{k-1}$ then $u{\lrcorner}s \in \Gamma_+^k$ stands for the pds u with s added on top of it. The set of level-n pds operations, written Ops_n, consists of

$$\begin{array}{rrcll}
\mathsf{top}_0\colon \Gamma_+^n \to \Gamma_*^{n-1}, & \mathsf{top}_0(u{\lrcorner}s) & = & s, & \\
\mathsf{top}_{-k}\colon \Gamma_+^n \to \Gamma_*^{n-k-1}, & \mathsf{top}_{-k}(u{\lrcorner}s) & = & \mathsf{top}_{1-k}(s), & \text{for } 1 \le k < n, \\
\mathsf{pop}_0\colon \Gamma_+^n \to \Gamma_*^n, & \mathsf{pop}_0(u{\lrcorner}s) & = & u, & \\
\mathsf{pop}_{-k}\colon \Gamma_+^n \to \Gamma_+^n, & \mathsf{pop}_{-k}(u{\lrcorner}s) & = & u{\lrcorner}\mathsf{pop}_{1-k}(s), & \text{for } 1 \le k < n, \\
\mathsf{push}_0\colon \Gamma_+^n \to \Gamma_+^n, & \mathsf{push}_0(u{\lrcorner}s) & = & (u{\lrcorner}s){\lrcorner}s & \text{for } n > 1, \\
\mathsf{push}_{-k}\colon \Gamma_+^n \to \Gamma_+^n, & \mathsf{push}_{-k}(u{\lrcorner}s) & = & u{\lrcorner}\mathsf{push}_{1-k}(s), & \text{for } 2 \le k < n, \\
\mathsf{push}_c\colon \Gamma^* \to \Gamma^+, & \mathsf{push}_c(u) & = & u{\lrcorner}c, & \text{for } c \in \Gamma, \\
\mathsf{push}_c\colon \Gamma_+^n \to \Gamma_+^n, & \mathsf{push}_c(u{\lrcorner}s) & = & u{\lrcorner}\mathsf{push}_c(s), & \text{for } 2 \le n \text{ and } c \in \Gamma.^2
\end{array}$$

An n-pda $\mathcal{P}$ is a tuple $(Q, \Sigma, \Gamma, q_0, \iota, \Delta, f)$, where Q is a finite set of states, Σ is the input alphabet, Γ is the pds alphabet, $q_0 \in Q$ is the initial state, $\iota \in \Gamma_+^n$ is the initial pds, $f \in Q$ is the final state and $\Delta \subseteq \Gamma \times Q \times (\Sigma \cup \varepsilon) \times \mathrm{Ops}_n \times Q$ is the set of transition rules of the form $(c, p) \xrightarrow{b} (\mathrm{op}, q)$ with $c \in \Gamma$, $p \in Q$, $b \in \Sigma \cup \{\varepsilon\}$, $\mathrm{op} \in \mathrm{Ops}_n$ and $q \in Q$.

The transition graph $\mathfrak{G}_{\mathcal{P}}$ of $\mathcal{P}$ is a subset of $(\Gamma_+^n \times Q) \times (\Sigma \cup \{\varepsilon\}) \times (\Gamma_*^n \times Q)$ defined by

$$\mathfrak{G}_{\mathcal{P}} \quad := \quad \{(s, p) \xrightarrow{b} (\mathrm{op}(s), q) \mid (\mathsf{top}_{1-n}(s), p) \xrightarrow{b} (\mathrm{op}, q) \in \Delta\} \ .$$

A word $w \in \Sigma^*$ is accepted by $\mathcal{P}$ and belongs to its language, written $\mathscr{L}(\mathcal{P})$, if it labels a path in $\mathfrak{G}_{\mathcal{P}}$ from (ι, q_0) to (s, f) for some $s \in \Gamma_*^n$:

$$\mathscr{L}(\mathcal{P}) := \left\{ w \in \Sigma^* \mid \exists\, s \in \Gamma_*^n \ (\iota, q_0) \underset{\mathfrak{G}_{\mathcal{P}}}{\overset{w}{\dashrightarrow}} (s, f) \right\} \ .$$

[1]In a transition graph of an automaton or machine, ε-labelled edges correspond to silent moves. An ε-closure consist in replacing, for each letter a of the input alphabet, every path $x \overset{\varepsilon^* a \varepsilon^*}{\dashrightarrow} y$ by an edge $x \xrightarrow{a} y$.

[2]Note that the name push_c is overloaded.

Note that in an unconventional way pds operations are indexed here by negative integers saying how deeply in the nesting level, starting from level n, the operation applies. In such a way we do not need to care about the exact level of pds operations. It just has to be strictly greater than the absolute value of the indexing integer.

Example 3.5. Consider a 2-pda $\mathcal{P}_{\text{sqr}} = (\{0,1,2,3,f\},\{a,b\},\{\bot,a,b\},0,[\bot][\bot],\Delta,f)$ accepting the square language on $\{a,b\}$, $L_{\text{sqr}} := \{ww \mid w \in \{a,b\}^*\}$. The set of transition rules Δ is

$$\begin{array}{lll}
(\bot,0) \xrightarrow{a} (\mathsf{push}_a,0), & (a,0) \xrightarrow{a} (\mathsf{push}_a,0), & (b,0) \xrightarrow{a} (\mathsf{push}_a,0), \\
(\bot,0) \xrightarrow{b} (\mathsf{push}_b,0), & (a,0) \xrightarrow{b} (\mathsf{push}_b,0), & (b,0) \xrightarrow{b} (\mathsf{push}_b,0), \\
(\bot,0) \xrightarrow{\varepsilon} (\mathsf{pop}_{-1},f), & (a,0) \xrightarrow{\varepsilon} (\mathsf{push}_{-1},1), & (b,0) \xrightarrow{\varepsilon} (\mathsf{push}_{-1},1), \\
(\bot,2) \xrightarrow{\varepsilon} (\mathsf{pop}_{-1},3), & (a,1) \xrightarrow{\varepsilon} (\mathsf{pop}_0,2), & (b,1) \xrightarrow{\varepsilon} (\mathsf{pop}_0,2), \\
(\bot,3) \xrightarrow{\varepsilon} (\mathsf{pop}_0,f), & (a,2) \xrightarrow{\varepsilon} (\mathsf{push}_{-1},1), & (b,2) \xrightarrow{\varepsilon} (\mathsf{push}_{-1},1), \\
 & (a,3) \xrightarrow{a} (\mathsf{pop}_{-1},3), & (b,3) \xrightarrow{b} (\mathsf{pop}_{-1},3).
\end{array}$$

The automaton pushes the letters as these are read on top of the topmost pds. Then it guesses the middle of the word and performs a sequence of ε-transitions which alternatively copy the topmost pds and pop its topmost letter. When $\bot$ is detected, $\mathcal{P}_{\text{sqr}}$ pops the whole topmost pds and starts reading the input again. The letter read has to agree with the letter on the top of the topmost pds whereas, the whole topmost pds is popped. When $\bot$ is detected, $\mathcal{P}_{\text{sqr}}$ switches to its final state.
Here is a path in $\mathfrak{G}_{\mathcal{P}_{\text{sqr}}}$ accepting $abbabb$ where each configuration is written as a sequence of level-1 pds enclosed in brackets and followed by the current state.

$$\begin{array}{l}
[\bot][\bot]0 \xrightarrow{a} [\bot][\bot a]0 \xrightarrow{b} [\bot][\bot ab]0 \xrightarrow{b} [\bot][\bot abb]0 \xrightarrow{\varepsilon} [\bot][\bot abb][\bot abb]1 \\
\xrightarrow{\varepsilon} [\bot][\bot abb][\bot ab]2 \xrightarrow{\varepsilon} [\bot][\bot abb][\bot ab][\bot ab]1 \xrightarrow{\varepsilon} [\bot][\bot abb][\bot ab][\bot a]2 \\
\xrightarrow{\varepsilon} [\bot][\bot abb][\bot ab][\bot a][\bot a]1 \xrightarrow{\varepsilon} [\bot][\bot abb][\bot ab][\bot a][\bot]2 \xrightarrow{\varepsilon} [\bot][\bot abb][\bot ab][\bot a]3 \\
\xrightarrow{a} [\bot][\bot abb][\bot ab]3 \xrightarrow{b} [\bot][\bot abb]3 \xrightarrow{b} [\bot]3 \xrightarrow{\varepsilon} [\bot]f
\end{array}$$

We shall use pds operations for proving several essential lemmas in the sequel.

3.4. Suffix-recognisable structures and level 1

By definition, level 0 of the iteration hierarchy corresponds to all finite structures. In the case of graphs, the level-1 of the iteration hierarchy consist, up to graph isomorphism, of prefix-recognisable (resp. suffix-recognisable) graphs defined as follows in [16]:

$$\begin{array}{rcl}
\mathfrak{A} & = & \bigcup_{i=1}^{m} (U_i \xrightarrow{a_i} V_i) W_i \\
\big(\text{resp. } \mathfrak{A} & = & \bigcup_{i=1}^{m} W_i (U_i \xrightarrow{a_i} V_i)\ \big)
\end{array}$$

for some $m \in \mathbb{N}$ and some $U_i, V_i, W_i \in \mathcal{R}\mathrm{eg}(\Gamma^*)$ that are non empty regular sets, and $a_1,\ldots,a_m \in \Sigma$. In the above, the operation $(U_i \xrightarrow{a_i} V_i)W_i$ (resp. $W_i(U_i \xrightarrow{a_i} V_i)$) defines the edge relation for label a_i as follows:

$$\begin{array}{rcl} (U_i \xrightarrow{a_i} V_i)W_i & := & \{(uw, vw) \mid u \in U_i, v \in V_i, w \in W_i\} \\ \big(\text{resp.}\;\; W_i(U_i \xrightarrow{a_i} V_i) & := & \{(wu, wv) \mid u \in U_i, v \in V_i, w \in W_i\}\;\big) \end{array}$$

In fact, these consist of two operations, namely Cartesian product and one sided multiplication. Because prefixes of pairs of words related by an edge form a recognisable relation, Damian Niwiński suggested to name those graphs prefix-recognisable.

Prefix-recognisable graphs may be considered as a syntactic characterisation of the family of graphs that are MSO-interpretable within the complete infinite binary tree [15, 30]. Although the term prefix-recognisable became common, it turns out that suffix-recognisable graphs are more consistent for this characterisation. Otherwise, a prefix-recognisable graph has to be transformed into its isomorphic suffix-recognisable twin by mirroring its vertices. The choice between prefix or suffix depends on the side of the multiplication: left for suffix and right for prefix. If both are combined we get bifix graphs [31, 32] which do not enjoy, in general, the decidability of their MSO theories.

The first prefix-recognisable-like characterisation of relations which are MSO-interpretable within the complete infinite binary tree $\mathfrak{T}_{\{0,1\}}$ is done by Angluin and Hoover [33][3]. Other such characterisations are given by Läuchli and Savioz [34], Carayol and Colcombet [35], and Blumensath [30]. The two latter papers consider, more generally, relational structures.

In the next definition, we review suffix-regular expressions and suffix-recognisable relations of arbitrary arity as defined in [35]. The latter use the generalisation of left multiplication of a relation $\mathcal{R} \subseteq (\Gamma^*)^m$ by a set $W \subseteq \Gamma^*$:

$$W\mathcal{R} \quad := \quad \{(wu_1, \ldots, wu_m) \mid w \in W \wedge (u_1, \ldots u_m) \in \mathcal{R}\} \ .$$

Since a relational structure is assimilated to a set of labelled hyperedges, the above operation is extended in the usual way to relational structures:

$$W\mathfrak{A} \quad := \quad \{a(wu_1, \ldots, wu_m) \mid a \in \Sigma \wedge w \in W \wedge a(u_1, \ldots u_m) \in \mathfrak{A}\} \ .$$

For introducing suffix-recognisable relations, we also need permutations. An n-permutation σ is a bijection of $[n]$ into itself extended to $\prod_{i=1}^{n} E_i$ in the usual way:

$$\text{for } (e_1, \ldots, e_n) \in \prod_{i=1}^{n} E_i \qquad \sigma(e_1, \ldots, e_n) = (e_{\sigma(1)}, \ldots, e_{\sigma(n)}) \ .$$

We often denote a permutation $\sigma\colon [n] \to [n]$, by the tuple $[\sigma^{-1}(1), \ldots, \sigma^{-1}(n)]$. For instance, given $\sigma\colon [3] \to [3]$ such that $\sigma(1) = 3, \sigma(2) = 1, \sigma(3) = 2$, we have $\sigma(a, b, c) = (c, a, b)$ and we may also write $[3, 1, 2](a, b, c) = (c, a, b)$.

[3]More generally, the complete infinite k-ary tree is considered there.

Definition 3.6. (Suffix-regular expressions and suffix-recognisable relations)
The set of *suffix-regular expressions* on Γ of arity $m \in \mathbb{N} \setminus \{0\}$, written $S\mathcal{R}\mathrm{eg}_m(\Gamma^*)$, is the smallest set of expressions such that

$$
\begin{array}{lll}
\mathcal{R}\mathrm{eg}(\Gamma^*) \subseteq S\mathcal{R}\mathrm{eg}_1(\Gamma^*), & & \\
\text{if } \mathcal{R}, \mathcal{S} \in S\mathcal{R}\mathrm{eg}_k(\Gamma^*) & \text{then} & \mathcal{R} \cup \mathcal{S} \in S\mathcal{R}\mathrm{eg}_k(\Gamma^*). \\
\text{if } \mathcal{R} \in S\mathcal{R}\mathrm{eg}_k(\Gamma^*) \text{ and } \mathcal{S} \in S\mathcal{R}\mathrm{eg}_l(\Gamma^*) & \text{then} & \mathcal{R} \times \mathcal{S} \in S\mathcal{R}\mathrm{eg}_{k+l}(\Gamma^*), \\
\text{if } W \in \mathcal{R}\mathrm{eg}(\Gamma^*) \text{ and } \mathcal{R} \in S\mathcal{R}\mathrm{eg}_k(\Gamma^*) & \text{then} & W\mathcal{R} \in S\mathcal{R}\mathrm{eg}_k(\Gamma^*), \\
\text{if } \mathcal{R} \in S\mathcal{R}\mathrm{eg}_k(\Gamma^*) \text{ and } \sigma \text{ is a } k\text{-permutation} & \text{then} & \sigma(\mathcal{R}) \in S\mathcal{R}\mathrm{eg}_k(\Gamma^*).
\end{array}
$$

The set of *prefix-regular expressions* is defined likewise, except that, in the fourth case, left multiplication $W\mathcal{R}$ is replaced by right multiplication $\mathcal{R}W$.

Any subset of $(\Gamma^*)^m$ denoted by a suffix-regular (resp. prefix-regular) expression of arity m on Γ is called a *suffix-recognisable* (resp. *prefix-recognisable*) *relation* on Γ^* of arity m, and is written $S\mathcal{R}\mathrm{ec}_m(\Gamma^*)$.

In case the encoding of elements does not matter, we simply write $S\mathcal{R}\mathrm{ec}_m$ to denote the class of m-ary suffix-recognisable relations up to isomorphism.

Here is an example of a suffix-regular expression on $\{0,1\}^*$ of arity 3:

$$[1,3,2]\big(\Gamma^*(\varepsilon \times \varepsilon) \times \Gamma^*\big) \ .$$

It is easy to see that this relation is also obtained by interpreting in $\mathfrak{T}_{\{0,1\}}$ the definition scheme $\langle \delta(x), \theta_a(x,y,z)\rangle$ where

$$
\begin{array}{rcl}
\delta(x) & = & \mathsf{true}, \\
\theta_a(x,y,z) & = & (x = z).
\end{array}
$$

For the example of the ladder (see Example 2.1), we write expressions for each labelled relation: $0^*(\varepsilon \times 0)$ for a, $0^*(01 \times 1)$ for b and $0^*(\varepsilon \times 1)$ for c. In a compact syntax of [16] which is suitable for graphs, we have $0^*(\varepsilon \xrightarrow{a} 0 + \varepsilon \xrightarrow{b} 1 + \varepsilon \xrightarrow{c} 1)$.

It is stated in [35] that the family of suffix-recognisable relations on $\{0,1\}^*$ is precisely the family of relations that are MSO-definable in the complete infinite binary tree $\mathfrak{T}_{\{0,1\}}$.

Theorem 3.7. An n-ary relation $\mathcal{R} \subseteq (\{0,1\}^*)^n$ is suffix-recognisable, if, and only if, $\mathcal{R}$ is MSO-definable in the complete infinite binary tree $\mathfrak{T}_{\{0,1\}}$.

The proof of this statement is sketched [35]. We give it a complete proof following a different idea. For that, we need to quickly review a few points about the decidability the MSO theory of $\mathfrak{T}_{\{0,1\}}$ also known as the decidability of the theory of two successors or S2S (see [36] for more details). The variables occurring in an S2S formula ψ form the set $\mathrm{Var}(\psi)$ and ψ uses atomic formulae of the form $x \xrightarrow{0} y$ and $x \xrightarrow{1} y$ for the two successors. Under some interpretation $\nu\colon \{0,1\}^* \to \wp(\mathrm{Var}(\psi))$, ψ is satisfied by a complete infinite binary tree $\mathfrak{T}^{\nu}_{\{0,1\}}$ with nodes labelled by variables of $\mathrm{Var}(\psi)$, each node $w \in \{0,1\}^*$ having possibly several labels forming the set $\nu(w)$. The standard decision procedure for S2S relies on the construction of a Muller tree automaton $\mathscr{A}_\psi$ such that $\mathfrak{T}^\nu_{\{0,1\}} \models \psi$ iff $\mathscr{A}_\psi$ accepts $\mathfrak{T}^\nu_{\{0,1\}}$. We call it *the automaton modelling* ψ. More precisely $\mathscr{A}_\psi = \big(Q, \wp(\mathrm{Var}(\psi)), \Delta, \iota, \mathcal{F}\big)$ where Q is a finite set of states, $\Delta \subseteq Q \times \wp(\mathrm{Var}(\psi)) \times Q \times Q$ is a transition relation, $\iota \in Q$ is an initial state and $\mathcal{F} \subseteq \wp(Q)$ is a set of accepting sets of states. A run of $\mathscr{A}_\psi$ on $\mathfrak{T}^\nu_{\{0,1\}}$ produces a labelling $\varrho: \{0,1\}^* \to Q$ of the nodes of $\mathfrak{T}_{\{0,1\}}$ such that $\varrho(\varepsilon) = \iota$ and $\big(\varrho(w), \nu(w), \varrho(w0), \varrho(w1)\big) \in \Delta$ for all $w \in \{0,1\}^*$. Such a run is accepting if, for every infinite branch of $\mathfrak{T}^\varrho_{\{0,1\}}$, the set of states occurring infinitely often in the branch belongs to $\mathcal{F}$. The latter condition is called *Muller acceptance*.

An S2S formula $\varphi(\overline{x})$ defining an n-ary relation on $\mathfrak{T}_{\{0,1\}}$ has exactly n free pairwise distinct variables $\overline{x} = (x_1, \ldots, x_n)$ which are FO variables. For $\mathfrak{T}^\nu_{\{0,1\}}$ to be a model of $\varphi(\overline{x})$, every FO variable labels exactly one node of $\mathfrak{T}^\nu_{\{0,1\}}$. The Muller tree automaton $\mathscr{A}_\varphi$ modelling $\varphi(\overline{x})$ is such that the relation defined by $\varphi(\overline{x})$ in $\mathfrak{T}_{\{0,1\}}$ is precisely

$$\Big\{(w_1, \ldots, w_n) \in (\{0,1\}^*)^n \mid \mathfrak{T}^\nu_{\{0,1\}} \in \mathscr{L}(\mathscr{A}_\varphi) \wedge \bigwedge_{i \in [n]} \nu(w_i) \cap \{x_1, \ldots, x_n\} = x_i\Big\} \ .$$

Proof of Theorem 3.7:

$\Rightarrow$

Assume that $\mathcal{R}$ is suffix-recognisable. The claim that $\mathcal{R}$ is MSO-definable in $\mathfrak{T}_{\{0,1\}}$ is established by induction on the structure of the expression denoting $\mathcal{R}$.

- case $\mathcal{R} = W$ with $W \in \mathcal{R}\mathrm{eg}(\Gamma^*)$

  Then W is defined by the formula $\exists\, y\ \big(\mathsf{root}(y) \wedge \mathsf{path}_W(y,x)\big)$.

- case $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2$ with $\mathcal{R}_1, \mathcal{R}_2 \in S\mathcal{R}\mathrm{ec}_k(\Gamma^*)$

  By induction hypothesis $\mathcal{R}_1$ and $\mathcal{R}_2$ are defined by some MSO formulae $\varphi_1(\overline{x})$ and $\varphi_2(\overline{x})$. Then $\mathcal{R}$ is defined by $\varphi_1(\overline{x}) \vee \varphi_2(\overline{x})$.

- case $\mathcal{R} = \mathcal{R}_1 \times \mathcal{R}_2$ with $\mathcal{R}_1 \in S\mathcal{R}\mathrm{ec}_k(\Gamma^*)$ and $\mathcal{R}_2 \in S\mathcal{R}\mathrm{ec}_l(\Gamma^*)$

  By induction hypothesis $\mathcal{R}_1$ and $\mathcal{R}_2$ are defined by some MSO formulae $\varphi_1(\overline{x})$ and $\varphi_2(\overline{y})$ where $\overline{x}$ and $\overline{y}$ are tuples of pairwise distinct variables. Then $\mathcal{R}$ is defined by $\varphi_1(\overline{x}) \wedge \varphi_2(\overline{y})$.

- case $\mathcal{R} = W\mathcal{S}$ with $W \in \mathcal{R}\mathrm{eg}(\Gamma^*)$ and $\mathcal{S} \in S\mathcal{R}\mathrm{ec}_k(\Gamma^*)$

  By induction hypothesis $\mathcal{S}$ is defined by some MSO formula $\psi(\overline{x})$. Let then

  $$\mathscr{A}_\psi = \big(Q, \wp(\mathrm{Var}(\psi)), \Delta, q_0, \Omega\big)$$

  be a Muller tree automaton modelling ψ. Let $\mathscr{B} = (Q', \Theta, q'_0, F)$ be a finite deterministic and complete (word) automaton accepting W. Assuming that $Q \cap Q' = \varnothing$, we construct a Muller tree automaton $\mathscr{A}_{W\mathcal{S}} := \big(Q \cup Q', \wp(\mathrm{Var}(\psi)), \Delta \cup \Delta', q'_0, \Omega\big)$, where

  $$\Delta' \ := \ \{(p, \varnothing, q, q') \mid (p,0,q), (p,1,q') \in \Theta\} \cup \{(f, P, p, q) \mid f \in F \wedge (q_0, P, p, q) \in \Delta\} \ .$$

  Now, the MSO formula associated to $\mathscr{A}_{W\mathcal{S}}$, say $\varphi(\overline{x})$, defines $W\mathcal{S}$ within $\mathfrak{T}_{\{0,1\}}$.

- case $\mathcal{R} = \sigma(\mathcal{S})$ with a k-permutation σ and $\mathcal{S} \in S\mathcal{R}\mathrm{ec}_k(\Gamma^*)$

  By induction hypothesis $\mathcal{S}$ is defined by some MSO formulae $\varphi(x_1, \ldots, x_k)$. Then $\mathcal{R}$ is defined by $\varphi(x_{\sigma(1)}, \ldots, x_{\sigma(k)})$.

$\Leftarrow$

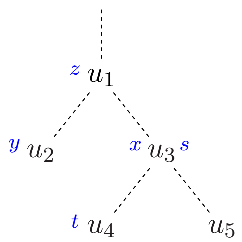
Assume that $\mathcal{R}$ is defined within $\mathfrak{T}_{\{0,1\}}$ by an S2S formula $\varphi(\overline{x})$ where $\overline{x} = (x_1, \ldots, x_n)$ is the tuple of (pairwise distinct) free variables of φ. For a tuple, $\overline{t} = (t_1, \ldots, t_m)$, $\mathsf{set}(\overline{t})$ stands for $\{t_i \mid i \in [m]\}$. For $k \leq m$, $\mathbf{A}_k(\overline{t})$ denotes the set of k-arrangements of $\overline{t}$:

$$\mathbf{A}_k(\overline{t}) \coloneqq \{(t_{f(1)}, \ldots, t_{f(k)}) \mid f\colon [k] \to [m] \text{ injective}\} \ .$$

Let $\mathscr{A}_\varphi = \big(Q, \wp(\mathrm{Var}(\varphi)), \Delta, \iota, \mathcal{F}\big)$ be a Muller tree automaton modelling $\varphi(\overline{x})$. We denote by $\mathscr{A}_{\varphi,q}$ the automaton resulting from replacing in $\mathscr{A}_\varphi$ its initial state ι by some state $q \in Q$. The first projection is written π_1. An *interpreting run* of $\mathscr{A}_{\varphi,q}$ on $\mathfrak{T}_{\{0,1\}}$ is a labelling $\mu\colon \{0,1\}^* \to Q \times \wp(\mathrm{Var}(\varphi))$ such that $\pi_1 \circ \mu$ is an accepting run of $\mathscr{A}_{\varphi,q}$ on $\mathfrak{T}^{\pi_2 \circ \mu}_{\{0,1\}}$. The set of all interpreting runs of $\mathscr{A}_{\varphi,q}$ is written $\mathsf{ir}(\mathscr{A}_{\varphi,q})$. We set $\mathsf{free}_\mu(u) \coloneqq \pi_2(\mu(u)) \cap \mathsf{set}(\overline{x})$ for any $u \in \{0,1\}^*$. We say that an interpreting run μ of $\mathscr{A}_{\varphi,q}$ is *consistent at with a k-arrangement* $\overline{t} = (t_1, \ldots, t_k) \in \mathbf{A}_k(\overline{x})$, if the depth-first ordered tuple of all nodes $\overline{w_{\overline{t}}} = (w_1, \ldots, w_k)$ of $\mathfrak{T}^\mu_{\{0,1\}}$, repetitions allowed, with labels in $\{x_1, \ldots, x_n\}$ (i.e. $\mathsf{free}_\mu(w_i) \neq \varnothing$ for every $i \in [k]$) is such that,

- $\mathsf{free}_\mu(w_i) \subseteq \mathsf{set}(\overline{t})$, for all $i \in [k]$,
- $\mathsf{free}_\mu(w) = \varnothing$ for all $w \in \{0,1\}^* \setminus \mathsf{set}(\overline{w_{\overline{t}}})$,
- $t_i \in \mathsf{free}_\mu(w_i)$, for all $i \in [k]$.

Since every variable of $\{x_1, \ldots, x_n\}$ occurs exactly once in an interpreting run μ of $\mathscr{A}_\varphi$ and at most once in an interpreting run μ of $\mathscr{A}_{\varphi,q}$, every node u of $\mathfrak{T}^\mu_{\{0,1\}}$ appears $|\mathsf{free}_\mu(u)|$ times in $\overline{w_{\overline{t}}}$. Up to such repetitions, the order of $\overline{w_{\overline{t}}}$ is the order in which nodes of $\mathsf{set}(\overline{w_{\overline{t}}})$ would be visited in a depth first search of $\mathfrak{T}^\mu_{\{0,1\}}$ (left branch first, nodes appearing in infix order) limited to the depth of the deepest node carrying a label in $\{x_1, \ldots, x_n\}$. For example, an interpreting run putting variables $\{x, y, z, s, t\}$ as follows



is consistent with (y, z, t, x, s) (and with (y, z, t, s, x)). The corresponding depth-first ordered tuple of nodes is $(u_2, u_1, u_4, u_3, u_3)$.

Claim $\mathscr{T}_{\overline{t}} \coloneqq \{\mathfrak{T}^\mu_{\{0,1\}} \mid \mu \in \mathsf{ir}(\mathscr{A}_{\varphi,q}) \text{ is consistent with } \overline{t}\,\}$ is regular for every arrangement $\overline{t}$ of $\overline{x}$.

Proof of the claim It is easy to design a Muller tree automaton $\mathscr{A}_{\bar{t}}$ which checks for consistency. It keeps a list of labels to encounter starting at the root with $\bar{t}$ and, while visiting nodes without label in $\{x_1, \ldots, x_n\}$, it nondeterministically breaks the current list into two sublists. It remembers the first sublist for the left subtree and second sublist for the right subtree. When $\mathscr{A}_{\bar{t}}$ visits a node with labels in $\{x_1, \ldots, x_n\}$, these must form a factor, say $(x_{l_j}, x_{l_{j+1}}, \ldots, x_{l_{j+k-1}})$, of the current list, say $(x_{l_1}, \ldots, x_{l_m})$ for some $k \le m \le n$. It then remembers $(x_{l_1}, \ldots, x_{l_{j-1}})$ for the left subtree and $(x_{l_{j+k}}, \ldots, x_{l_m})$ for the right subtree. When the current list, say $\bar{s}$, matches $\mathsf{free}_\mu(u)$ of the current node, say u (i.e. $\mathsf{free}_\mu(u) = \mathsf{set}(\bar{s})$), the whole subtree rooted at u is labelled with a designated state f. The set of accepting sets of states realising Muller condition for $\mathscr{A}_{\bar{t}}$ is $\{\{f\}\}$.
Now, the automaton accepting $\mathscr{T}_{\bar{t}}$ is obtained as the product of automata $\mathscr{A}_\varphi \times \mathscr{A}_{\bar{t}}$. $\lhd$

Let $m \ge 1$. We call a *3-split* of a tuple $(t_1, \ldots, t_m)$ a triple

$$\big((t_1, \ldots, t_{j-1}), (t_j, t_{j+1}, \ldots, t_{j+k-1}), (t_{j+k}, \ldots, t_m)\big) \ .$$

such that, if $k = 0$, then $j \ge 2$ and $m \ge j + k$ (if the middle tuple has length 0, then both remaining tuples have non-zero length). The set of all 3-splits of a tuple $\bar{t}$ is written $3\mathrm{sp}(\bar{t})$. We say that an interpreting run $\mu \in \mathsf{ir}(\mathscr{A}_{\varphi,q})$ is *consistent with a 3-split* $(\bar{y}, \bar{z}, \bar{s})$ of an arrangement of $\bar{x}$, if there is a node $u \in \{0,1\}^*$ such that

- $\mathsf{free}_\mu(u) = \mathsf{set}(\bar{z})$,
- μ restricted to the left subtree under u is consistent with $\bar{y}$,
- μ restricted to the right subtree under u is consistent with $\bar{s}$.

Similarly to the above claim about $\mathscr{T}_{\bar{t}}$, it is easy to show that

$$\mathscr{T}_{(\bar{y},\bar{z},\bar{s}),q} := \{\mathfrak{T}^\mu_{\{0,1\}} \mid \mu \in \mathsf{ir}(\mathscr{A}_{\varphi,q}) \text{ is consistent with } (\bar{y}, \bar{z}, \bar{s})\}$$

is regular for every 3-split $(\bar{y}, \bar{z}, \bar{s})$ of an arrangement of $\bar{x}$.

Claim For every regular set of complete binary trees $\mathscr{T}$ with node labels in some finite set, any set of nodes $L_{\mathscr{T},\psi}$ of trees of $\mathscr{T}$ defined by an MSO formula $\psi(x)$, namely

$$L_{\mathscr{T},\psi} := \big\{w \in \{0,1\}^* \mid \mathfrak{T}^\nu_{\{0,1\}} \in \mathscr{T} \wedge (T^\nu_{\{0,1\}}, w) \models \psi(x)\big\},$$

is a regular set of words.

Proof of the claim Let $\mathscr{A} = \big(Q, \mathcal{P}, \Delta, \iota, \mathcal{F}\big)$ be the product $\mathscr{A}_{\mathscr{T}} \times \mathscr{A}_\psi$ of a Muller tree automaton $\mathscr{A}_{\mathscr{T}}$ such that $\mathscr{L}(\mathscr{A}_{\mathscr{T}}) = \mathscr{T}$ with a Muller tree automaton $\mathscr{A}_\psi$ modelling an MSO formula $\psi(x)$ defining some set of nodes of trees of $\mathscr{T}$. We consider the finite (word) automaton $\mathscr{B} = (Q \times \mathcal{P}, \Theta, I, F)$ on $\{0,1\}$ with set of states $Q \times \mathcal{P}$, transition relation

$$\begin{aligned}\Theta := \{((q,P),0,(q_1,P_1)) \mid (q,P,q_1,q_2) \in \Delta \wedge (q_1,P_1) \in \pi_{1,2}(\Delta)\} \cup \\ \{((q,P),1,(q_2,P_2)) \mid (q,P,q_1,q_2) \in \Delta \wedge (q_2,P_2) \in \pi_{1,2}(\Delta)\},\end{aligned}$$

set of initial states $I := \{(\iota, P) \mid (\iota, P, q_1, q_2) \in \Delta\}$ and set of final states F defined as follows

$$\begin{aligned}(q,P) \in F \quad :\Leftrightarrow \quad & x \text{ occurs in } P \quad \text{and} \\ & \text{there exists a path } (q,P) \underset{\mathscr{B}}{\dashrightarrow} (p_1,P_1) \underset{\mathscr{B}}{\longrightarrow} (p_2,P_2) \underset{\mathscr{B}}{\longrightarrow} \cdots \underset{\mathscr{B}}{\longrightarrow} (p_k,P_k) \underset{\mathscr{B}}{\longrightarrow} (p_1,P_1) \\ & \qquad \text{such that } \bigcup_{i\in[k]} \{p_i\} \in \mathcal{F} \ .\end{aligned}$$

It follows that $\mathscr{L}(\mathscr{B}) = L_{\mathscr{T},\psi}$. ◁

We are ready to give an inductive construction of a suffix-regular expression $\mathcal{E}_\varphi$ for the relation defined by $\varphi(\overline{x})$. For $\overline{y} \in \mathbf{A}_n(\overline{x})$, let $\sigma_{\overline{y},\overline{x}}: [n] \to [n]$ be a permutation such that $\sigma(\overline{y}) = \overline{x}$. For $\overline{z} \in \mathbf{A}_k(\overline{x})$, we define $\Delta_{\overline{z}} := \{(q, \mathcal{Y}, q_1, q_2) \in \Delta \mid \mathcal{Y} \cap \mathsf{set}(\overline{x}) = \mathsf{set}(\overline{z})\}$. The expression is given by

$$\mathcal{E}_\varphi \quad := \quad \bigcup_{\overline{y}\in\mathbf{A}_n(\overline{x})} \sigma_{\overline{y},\overline{x}}\, \mathcal{E}(\iota, \overline{y})$$

where, for $p \in Q$ and $\overline{s} \in \mathbf{A}_k(\overline{x})$ with $k \geq 1$,

$$\mathcal{E}(p,\overline{s}) \quad := \quad \bigcup_{(\overline{y},\overline{z},\overline{t})\,\in\,3\mathrm{sp}(\overline{s})} \ \bigcup_{(q,\mathcal{Y},q_1,q_2)\,\in\,\Delta_{\overline{z}}} L_{\mathscr{T}_{(\overline{y},\overline{z},\overline{t}),p},\psi[q,\mathcal{Y},q_1,q_2]}\big(0\mathcal{E}(q_1,\overline{y}) \times \prod_{|\overline{z}|} \varepsilon \times 1\mathcal{E}(q_2,\overline{t})\big)$$

and where $\psi[q, \mathcal{Y}, q_1, q_2](x)$ is a formula satisfied at every node u of a tree of $\mathscr{T}_{(\overline{y},\overline{z},\overline{t}),p}$ such that u is labelled $(q, \mathcal{Y})$ and q_1 (resp. q_2) occurs in the label of $u0$ (resp. $u1$). The expression for $\mathcal{E}(p,\overline{s})$ is completed with the case of zero-length tuple of variables: $\mathcal{E}(p,()) := \mathbb{1}$ where $\mathbb{1}$ denotes the neutral element for the Cartesian product. r □

The following corollary is immediate as the set of MSO-definable k-ary relations within a given structure forms a Boolean algebra.

Corollary 3.8. For every $k \in \mathbb{N}$, $S\mathcal{R}\mathrm{ec}_k(\Gamma^*)$ is a Boolean algebra.

As expected, suffix-recognisable relations are components of suffix-recognisable structures.

Definition 3.9. (Suffix-recognisable structures)
A Σ-structure $\mathfrak{A}$ is suffix-recognisable if $\mathfrak{A}(a) \in S\mathcal{R}\mathrm{ec}_m$ for each $a \in \Sigma$ with $\alpha(a) = m$. The class of suffix-recognisable Σ-structures is written $\mathfrak{S}\mathfrak{R}\mathfrak{ec}(\Sigma)$.

Theorem 3.7 adapted to the latter definition is stated as the following corollary.

Corollary 3.10. A Σ-structure is suffix-recognisable, if, and only if, it is MSO-interpretable in the complete infinite binary tree $\mathfrak{T}_{\{0,1\}}$.

Since the composition of two MSO interpretations is again an MSO interpretation [8], from Theorem 3.7 we get the following corollary.

Corollary 3.11. The family of suffix-recognisable structures is closed under MSO-interpretations.

In order to show that level 1 of the iteration hierarchy consists, up to isomorphism, of suffix-recognisable structures we start with the following lemma.

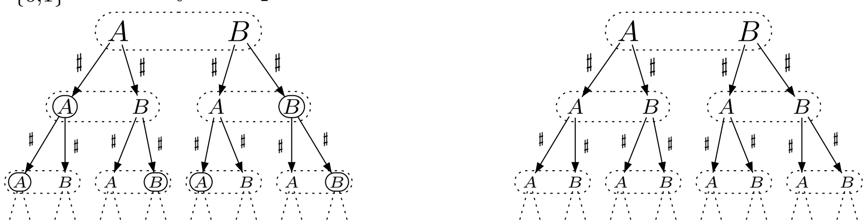
Lemma 3.12. Muchnik's and the basic iteration of every finite structure is, up to isomorphism, suffix-recognisable.

Proof:
Let $\mathfrak{A}$ be a finite structure. We set $\Gamma := V_{\mathfrak{A}}$. Let $\sharp, \& \notin \Sigma$. The relation corresponding to $a \in \Sigma$ in both $\mathfrak{A}^{\sharp,\&}$ and $\mathfrak{A}^{\sharp}$ is a finite union

$$\bigcup_{a(s_1,\ldots,s_{\alpha(a)})\in\mathfrak{A}} \Gamma^* a(s_1,\ldots,s_{\alpha(a)}) \ .$$

For $\sharp$ we have $\bigcup_{s\in\Gamma} \Gamma^+(\varepsilon \xrightarrow{\sharp} s)$ and for & we have $\bigcup_{s\in\Gamma} \Gamma^* \&(ss)$. □

Starting from a 2-element structure $A \quad B$ over 2 unary labels $\{A, B\}$ by Muchnik's iteration and even by basic iteration, we get a structure (see below) where a complete infinite binary tree $\mathfrak{T}_{\{0,1\}}$ is readily interpreted.



This leads to the following corollary.

Corollary 3.13.

1. $\mathfrak{Itr}_1 \ \equiv \ \mathfrak{SRec}$,
2. $\mathfrak{Itr}_1 \ \equiv \ \{f(\mathfrak{A}^{\sharp}) \mid \mathfrak{A} \in \mathfrak{Itr}_0 \wedge f$ is an MSO interpretation$\}$.

Proof:
Set $\mathfrak{Itr}_1^{\text{basic}} := \{f(\mathfrak{A}^{\sharp}) \mid \mathfrak{A} \in \mathfrak{Itr}_0 \wedge f$ is an MSO interpretation$\}$. From Lemma 3.12 and Corollary 3.11 it follows that (up to isomorphism) $\mathfrak{Itr}_1 \subseteq \mathfrak{SRec}$ and $\mathfrak{Itr}_1^{\text{basic}} \subseteq \mathfrak{SRec}$. Both $\mathfrak{Itr}_1 \supseteq \mathfrak{SRec}$ and $\mathfrak{Itr}_1^{\text{basic}} \supseteq \mathfrak{SRec}$ (up to isomorphism) follow from the fact that $\mathfrak{T}_{\{0,1\}}$ is MSO-interpretable within a structure that is obtained as basic or Muchnik's iteration of a two-element structure and the fact that every suffix-recognisable structure is MSO-interpretable within $\mathfrak{T}_{\{0,1\}}$ (Theorem 3.7). Since the composition of two MSO interpretations is again an MSO interpretation, we are done. □

The above corollary suggests that similarly to the iteration hierarchy $\mathfrak{Itr}_n$, one might define the basic iteration hierarchy, say $\mathfrak{Itr}_n^{\text{basic}}$. However, as we shall see in the next section, the latter hierarchy collapses at level 1.

4. Closure under basic iteration

This section starts by the statement of the main theorem which is established by induction. After showing the induction basis, the proof of the induction step is split into two lemmas. While the proof of Lemma 4.2 is concise, Lemma 4.3 is based on a more subtle construction. We explain this construction step by step following a simple example. Each step uses an MSO interpretation.

Theorem 4.1. For any $n \geq 1$, the family of structures $\mathcal{Hgr}_n$ is closed under basic iteration.

Proof:
Let $n \in \mathbb{N} \setminus \{0\}$ and $\mathfrak{A}$ be a structure in $\mathcal{Hgr}_n$. The claim that $\mathfrak{A}^\sharp \in \mathcal{Hgr}_n$ is established by induction on n.

- $n = 1$

  According to Corollary 3.13 w.l.o.g. we may assume that $\mathfrak{A}$ is a suffix-recognisable structure. By definition

$$\begin{aligned}\mathfrak{A}^\sharp = \{a(s_1 \dots s_k u_1, \dots, s_1 \dots s_k u_{\alpha(a)}) \mid k \in \mathbb{N} \wedge s_1, \dots, s_k \in V_{\mathfrak{A}} \\ \wedge\, a(u_1 \dots, u_{\alpha(a)}) \in \mathfrak{A}\} \\ \cup \{s_1 \dots s_k \xrightarrow{\sharp} s_1 \dots s_k t \mid k \in \mathbb{N} \wedge s_1, \dots, s_k, t \in V_{\mathfrak{A}}\} \ .\end{aligned}$$

  In order to keep track of original vertices (words) that would be lost under concatenation, we introduce a separator $\dagger \notin \Gamma$:

$$\begin{aligned}\mathfrak{A}^\sharp &= \{a(s_1\dagger \dots \dagger s_k \dagger u_1, \dots, s_1\dagger \dots \dagger s_k \dagger u_{\alpha(a)}) \mid k \in \mathbb{N} \wedge s_1, \dots, s_k \in V_{\mathfrak{A}} \\ &\qquad \wedge\, a(u_1 \dots, u_{\alpha(a)}) \in \mathfrak{A}\} \\ &\quad \cup \{s_1\dagger \dots \dagger s_k \xrightarrow{\sharp} s_1\dagger \dots \dagger s_k \dagger t \mid k \in \mathbb{N} \wedge s_1, \dots, s_k, t \in V_{\mathfrak{A}}\} \\ &= \{a(wu_1, \dots, wu_{\alpha(a)}) \mid w \in (V_{\mathfrak{A}}\dagger)^* \wedge a(u_1 \dots, u_{\alpha(a)}) \in \mathfrak{A}\} \\ &\quad \cup \{w \xrightarrow{\sharp} w\dagger t \mid w \in (V_{\mathfrak{A}}\dagger)^* V_{\mathfrak{A}} \wedge t \in V_{\mathfrak{A}}\} \\ &= \{w.a(u_1 \dots u_{\alpha(a)}) \mid w \in (V_{\mathfrak{A}}\dagger)^* \wedge a(u_1 \dots, u_{\alpha(a)}) \in \mathfrak{A}\} \\ &\quad \cup \{w.(\varepsilon \xrightarrow{\sharp} t) \mid w \in (V_{\mathfrak{A}}\dagger)^* \wedge t \in V_{\mathfrak{A}}\} \\ &= (V_{\mathfrak{A}}\dagger)^*.\mathfrak{A} \ \cup \ (V_{\mathfrak{A}}\dagger)^* V_{\mathfrak{A}}.(\varepsilon \xrightarrow{\sharp} V_{\mathfrak{A}}) \ .\end{aligned}$$

  Thus $\mathfrak{A}^\sharp$ is suffix-recognisable. Consequently $\mathfrak{A}^\sharp \in \mathcal{Hgr}_1$.

- $n > 1$ (induction step)

  According to the definition of the hierarchy, $\mathfrak{A} \in \mathcal{Hgr}_n$ is obtained from a structure in $\mathcal{Hgr}_{n-1}$, say $\mathfrak{B}$, through Muchnik's iteration followed by an MSO interpretation, say f:

$$\mathfrak{A} = f(\mathfrak{B}^{\sharp,\&})$$

Consider a basic iteration $\mathfrak{A}^{\$}$ of $\mathfrak{A}$ for a new binary symbol $\$ \notin \Sigma \cup \{\sharp, \&\}$. Then $\mathfrak{A}^{\$} = f(\mathfrak{B}^{\sharp,\&})^{\$}$.

We establish in subsequent lemmas that there exist MSO interpretations $f_{\$}$, g and h such that

$$\big(f(\mathfrak{B}^{\sharp,\&})\big)^{\$} \equiv f_{\$}(g((h(\mathfrak{B})^{\$})^{\sharp,\&})) \ . \tag{i}$$

Since $h(\mathfrak{B}) \in \mathfrak{Hgr}_{n-1}$, by induction hypothesis we have $h(\mathfrak{B})^{\$} \in \mathfrak{Hgr}_{n-1}$. Then

$$(h(\mathfrak{B})^{\$})^{\sharp,\&} \in \mathfrak{Hgr}_n$$

and also $g((h(\mathfrak{B})^{\$})^{\sharp,\&}) \in \mathfrak{Hgr}_n$. Finally

$$f_{\$}(g((h(\mathfrak{B})^{\$})^{\sharp,\&})) \in \mathfrak{Hgr}_n \ .$$

Equality (i) is established in two steps. First, using Lemma 4.2, the existence of an MSO interpretation $f_{\$}$ such that

$$f(\mathfrak{B}^{\sharp,\&})^{\$} = f_{\$}((\mathfrak{B}^{\sharp,\&})^{\$})$$

is obtained. Second, using Lemma 4.3, the existence of MSO interpretations g and h such that

$$(\mathfrak{B}^{\sharp,\&})^{\$} \equiv g((h(\mathfrak{B})^{\$})^{\sharp,\&})$$

is ascertained.

□

The above proof relies upon two lemmas. The first one states that basic iteration and MSO interpretations commute, provided a slight adaptation of the latter.

Lemma 4.2. For every MSO interpretation f and $\$ \notin \Sigma$, there exists an MSO interpretation $f_{\$}$ such that, for every relational structure $\mathfrak{C}$ over Σ, one has

$$f(\mathfrak{C})^{\$} = f_{\$}(\mathfrak{C}^{\$}) \ .$$

Proof:
Let f be an MSO interpretation. Observe first that equality

$$f(\mathfrak{C})^{\$} = f(\mathfrak{C}^{\$})$$

does not hold because f may add hyperedges across distinct copies of $\mathfrak{C}$. Thus definition scheme $\langle \delta, (\theta_a)_{a\in\Sigma}\rangle$ of f has to be adapted as follows. Each formula $\theta_a(x_1, \ldots, x_{\alpha(a)})$ needs to be relativised w.r.t. vertices of the same copy (viz., sharing the same \$-ancestor) or of the original (viz., no \$-ancestor) $\mathfrak{C}$ within $\mathfrak{C}^{\$}$. Therefore $\theta_a^{\$}(x_1, \ldots, x_{\alpha(a)})$ is defined as being the following formula:

$$\big(\exists\, y\, (y \xrightarrow{\$} x_1 \wedge \cdots \wedge y \xrightarrow{\$} x_{\alpha(a)}) \vee \forall\, y\, (\neg y \xrightarrow{\$} x_1 \wedge \cdots \wedge \neg y \xrightarrow{\$} x_{\alpha(a)})\big) \wedge \theta_a(x_1, \ldots, x_{\alpha(a)}) \ .$$

Thus, the definition scheme of $f_{\$}$ is $\langle \delta, (\theta_a^{\$})_{a\in\Sigma}\rangle$. □

The second lemma involved in the induction step of the proof of the main theorem states that basic iteration and Muchnik's iteration commute up to two MSO interpretations. This is the crux and the remainder of this section aims at providing a clear presentation of a proof this lemma.

Lemma 4.3. There exists MSO interpretations g and h, such that every relational structure $\mathfrak{B}$ satisfies

$$(\mathfrak{B}^{\sharp,\&})^{\$} \equiv g((h(\mathfrak{B})^{\$})^{\sharp,\&}) \ .$$

The above lemma is the key lemma for the induction step of Theorem 4.1. As the induction step deals with structures of level $n > 1$, obtained via n iterations (and MSO interpretations), we shall adopt the following convention. We consider that the vertices of level-n structure belong to Γ_*^n where Γ is the set of vertices of the finite structure from which $\mathfrak{A}$ has been obtained through n steps. This is consistent with the definition of iteration since a level-n pds is a word over Γ_*^{n-1}. Moreover, in a basic iteration $\mathfrak{C}^\sharp$ or Muchnik's iteration $\mathfrak{C}^{\sharp,\&}$ of some $\mathfrak{C} \in \mathfrak{Hor}_{n-1}$, the inverse of $\xrightarrow{\sharp}$ corresponds to pop_0 whereas push_0 may be identified in $\mathfrak{C}^{\sharp,\&}$ with those edges $\xrightarrow{\sharp}$ that point to &-labelled vertices. In fact, as observed in [29], the reader may notice that all level-n pds operations are first-order definable within an n-fold Muchnik-iterated structure provided that iterations symbols $\sharp_1, \ldots, \sharp_n$ are pairwise distinct. With this idea in mind, we begin a discussion that will lead to the proof the above key lemma. In this lemma, we consider a structure $\mathfrak{B}$ and its iterations possibly combined with MSO interpretations: $(\mathfrak{B}^{\sharp,\&})^{\$}$ and $g((h(\mathfrak{B})^{\$})^{\sharp,\&})$. At some stage, we shall also deal with a Muchnik's iteration $\mathfrak{B}^{\sharp,\&}$ and a basic iteration $h(\mathfrak{B})^{\$}$. According to our convention, the vertices of the latter structures are pds over $V_{\mathfrak{B}}$. The vertices of $(\mathfrak{B}^{\sharp,\&})^{\$}$ and $g((h(\mathfrak{B})^{\$})^{\sharp,\&})$ are level-2 pds over $V_{\mathfrak{B}}$. Such a level-2 pds, say v, is written $[t_{1,1} \ldots t_{1,l_1}][t_{2,1} \ldots t_{2,l_2}] \ldots [t_{k,1} \ldots t_{k,l_k}]$ where all $t_{i,j}$ are in $V_{\mathfrak{B}}$ and $t_{k,1} \ldots t_{k,l_k}$ is the topmost level-1 pds, viz. $\mathsf{top}_0(v) = t_{k,1} \ldots t_{k,l_k}$ and $\mathsf{top}_{-1}(v) = t_{k,l_k}$.

In order to define a mapping between the vertices of $(\mathfrak{B}^{\sharp,\&})^{\$}$ on one hand and, provided MSO interpretations h and g, those of $g((h(\mathfrak{B})^{\$})^{\sharp,\&})$ on the other hand, and establish that this is an isomorphism, we need a way to point at a vertex. For this reason, we shall decorate iteration labels $\sharp$ and $\$$ with top_{-1} of the target vertex according to the following definition.

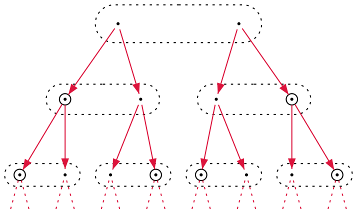
Definition 4.4. (Decorated label)
Given an edge $u \xrightarrow{£} v$ with $£ \in \{\sharp, \$\}$, of a doubly iterated structure, *its decorated label*, written $£(s)$, is the corresponding iteration label $£$ decorated with $s = \mathsf{top}_{-1}(v)$ of the target v of the edge.

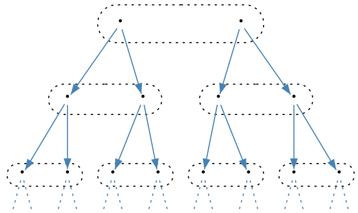
For structures considered in the sequel, we assume that iteration edges have its labels implicitly decorated. We write $u \xrightarrow{\sharp(s)} v$ or $u \xrightarrow{\$(s)} v$ in order to emphasise that $s = \mathsf{top}_{-1}(v)$ although we may still write $u \xrightarrow{\sharp} v$ or $u \xrightarrow{\$} v$ in the case the decoration does not matter.

We denote by Λ, the set of decorated labels of $(\mathfrak{B}^{\sharp,\&})^{\$}$, $(\mathfrak{B}^{\$})^{\sharp,\&}$ and its interpreted variants:

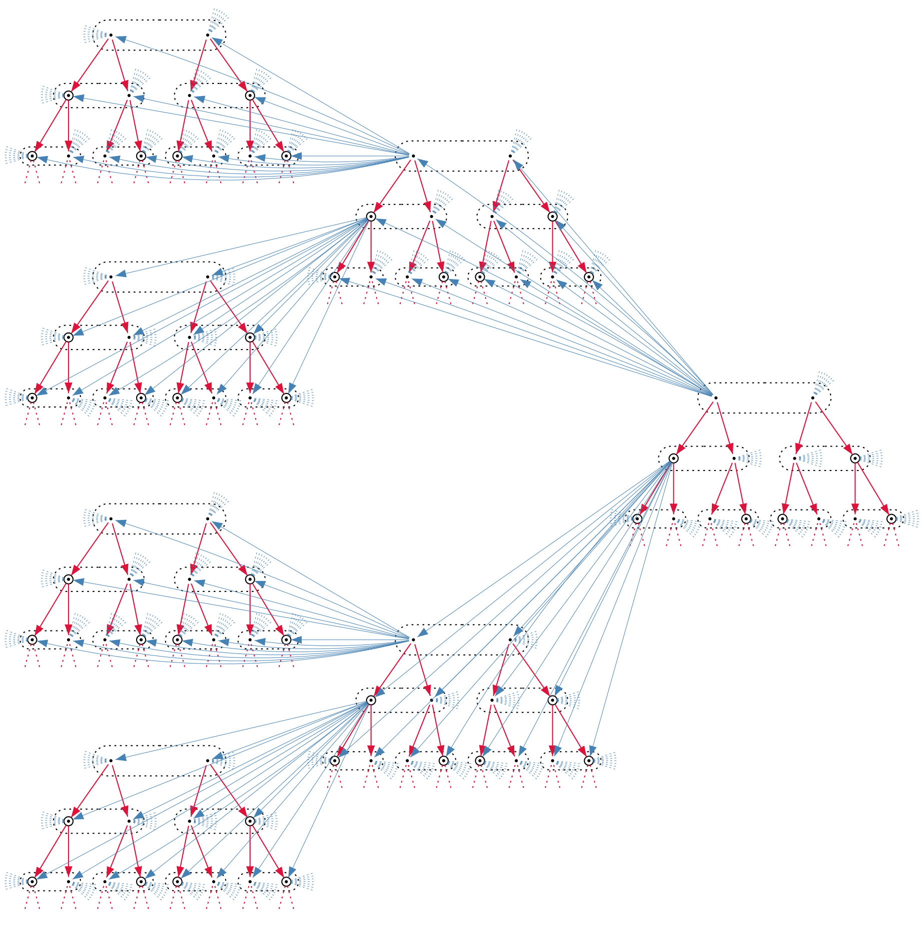
$$\Lambda \quad := \quad \{\sharp(s) \mid s \in V_{\mathfrak{B}}\} \cup \{\$(s) \mid s \in V_{\mathfrak{B}}\} \ .$$

Before addressing the proof of the above lemma, let us consider an example of a structure $\mathfrak{D}$ with two vertices and no relation as well as its iterations $\mathfrak{D}^{\sharp,\&}$, $\mathfrak{D}^{\$}$, $(\mathfrak{D}^{\sharp,\&})^{\$}$, $(\mathfrak{D}^{\$})^{\sharp,\&}$. In order to trim the picture, we omit labels over edges. We use colours instead: for $\sharp$-labelled edges and for \$-labelled edges. Moreover, &-labelled vertices are circled and its labels are omitted.

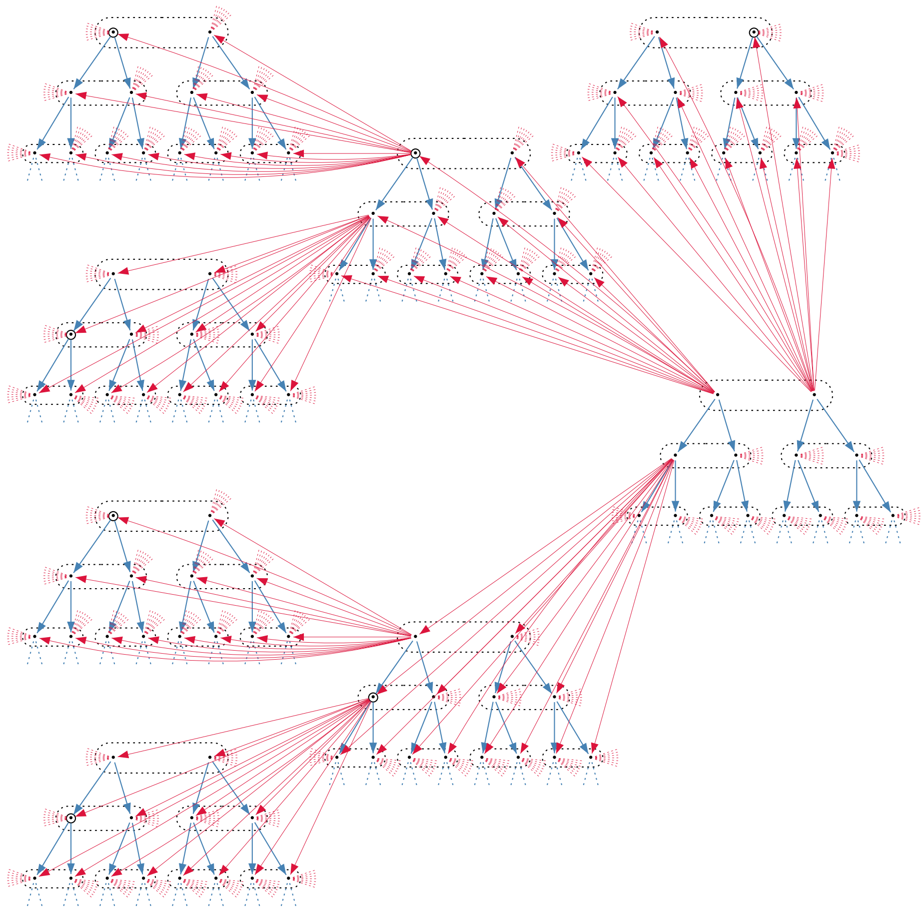




As the example is developed, we explain the construction, the proof of Lemma 4.3 is based upon.



This construction will end, up to isomorphism, with iteration $(\mathfrak{D}^{\sharp,\&})^{\$}$ depicted above, starting from iteration $(\mathfrak{D}^{\$})^{\sharp,\&}$ depicted as follows.



More precisely a structure isomorphic to $(\mathfrak{D}^{\sharp,\&})^{\$}$ shall be interpreted within $(\mathfrak{D}^{\$})^{\sharp,\&}$ through three steps. While addressing the construction of this interpretation, we wish to point out two noticeable substructures of $(\mathfrak{D}^{\sharp,\&})^{\$}$ and of $(\mathfrak{D}^{\$})^{\sharp,\&}$ that are isomorphic. The first one, depicted on Fig. 1, is a substructure $\mathfrak{D}_1$ of $(\mathfrak{D}^{\sharp,\&})^{\$}$ obtained by forgetting every $\$$-labelled edge, when the target of which is also the target of a $\sharp$-labelled edge. Thus, every vertex v of substructure $\mathfrak{D}_1$ has outgoing $\$$-labelled edges exactly to every vertex of the «topmost» copy of $\mathfrak{D}$ within v's private copy of $\mathfrak{D}^{\sharp,\&}$. Analogously, we may associate to every structure $\mathfrak{B}$ and its double iteration $(\mathfrak{B}^{\sharp,\&})^{\$}$, a substructure $\mathfrak{B}_1$. A useful fact that may be generalised from the example of $\mathfrak{D}_1$ is that for each word $\kappa \in \Lambda^*$ over the set of decorated labels Λ, a path in a structure like $\mathfrak{D}_1$ labelled by κ is unique from a given vertex u. Moreover, every vertex of $\mathfrak{D}_1$ is accessible from $\mathfrak{D}$ via a path with label in Λ^*.

Lemma 4.5. Let $\mathfrak{B}_1$ be the substructure obtained from $(\mathfrak{B}^{\sharp,\&})^{\$}$ by forgetting every $\$$-labelled edge, the target of which is also the target of a $\sharp$-labelled edge.

1. For every word $\kappa \in \Lambda^*$ and each vertex $u \in V_{\mathfrak{B}_1}$, there is a unique path in $\mathfrak{B}_1$ starting at u and labelled by κ.

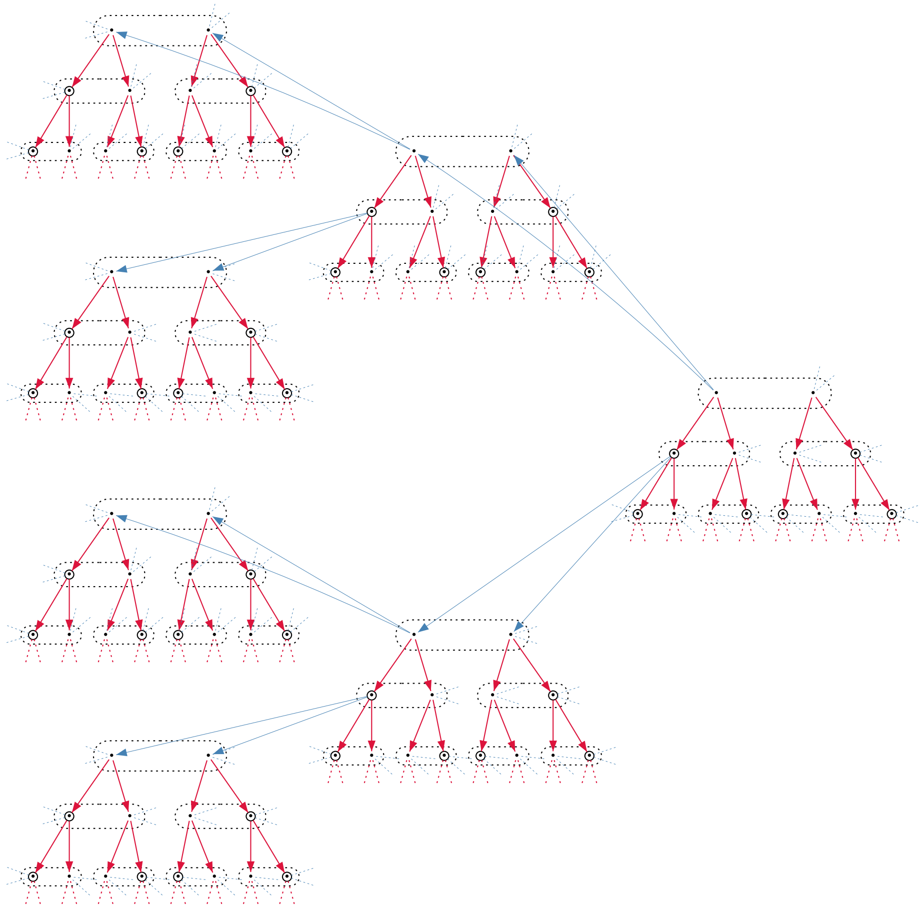


Figure 1. Substructre $\mathfrak{D}_1$

2. Each vertex of $\mathfrak{B}_1$ is accessible from $\mathfrak{B}$ via a path with label in Λ^*.

Proof:

1. We show that there is exactly one edge in $\mathfrak{B}_1$ with a given source and decorated label. Consider an edge of $\mathfrak{B}_1$ with source $u \in V_{\mathfrak{B}_1}$ and decorated label $\lambda \in \Lambda$. We know that

$$u = [t_{1,1} \dots t_{1,l_1}][t_{2,1} \dots t_{2,l_2}] \dots [t_{k,1} \dots t_{k,l_k}], \quad \text{for some } t_{i,j} \in V_{\mathfrak{B}} \ .$$

- Case $\lambda = \sharp(s)$ for some $s \in V_{\mathfrak{B}}$.

  This case occurs if, and only if, $u \xrightarrow{\sharp(s)} v$, where

$$v = [t_{1,1} \dots t_{1,l_1}][t_{2,1} \dots t_{2,l_2}] \dots [t_{k,1} \dots t_{k,l_k} s] \ .$$

- Case $\lambda = \$(s)$ for some $s \in V_{\mathfrak{B}}$.

  This case occurs if, and only if, $u \xrightarrow{\$(s)} v$, where
  $$v = [t_{1,1} \dots t_{1,l_1}][t_{2,1} \dots t_{2,l_2}] \dots [t_{k,1} \dots t_{k,l_k}][s] \ .$$
  This is because every $\$$-labelled edge from u to a vertex of the form
  $$[t_{1,1} \dots t_{1,l_1}][t_{2,1} \dots t_{2,l_2}] \dots [t_{k,1} \dots t_{k,l_k}][s_0 \dots s_n s],$$
  with $n \in \mathbb{N}$, that would also have $\$(s)$ as decorated does not exist anymore in $\mathfrak{B}_1$ since it has been removed in the construction of $\mathfrak{B}_1$ from $(\mathfrak{B}^{\sharp,\&})^{\$}$.

2. Let $v = [t_{1,1} \dots t_{1,l_1}][t_{2,1} \dots t_{2,l_2}] \dots [t_{k,1} \dots t_{k,l_k}]$ be a vertex of $\mathfrak{B}_1$. By induction on the length of v, $|v| := \sum_{i=1}^{k} l_i$, we show that there exists a vertex $u \in V_{\mathfrak{B}}$ and a word $\sigma \in \{\sharp, \$\}^*$ such that $u \overset{\sigma}{\dashrightarrow} v$, viz., there exists a σ-labelled path from u to v.

   - Case $|v| = 1$.

     Then $v \in V_{\mathfrak{B}}$ and $v \overset{\varepsilon}{\dashrightarrow} v$.

   - <u>case $|v| > 1$</u>
     - Subcase $l_k = 1$.

       Then $v = [t_{1,1} \dots t_{1,l_1}][t_{2,1} \dots t_{2,l_2}] \dots [t_{k-1,1} \dots t_{k-1,l_{k-1}}][t_{k,1}]$ and there is a vertex $v' = [t_{1,1} \dots t_{1,l_1}][t_{2,1} \dots t_{2,l_2}] \dots [t_{k-1,1} \dots t_{k-1,l_{k-1}}]$ such that $v' \xrightarrow{\$} v$. Indeed, v has no ingoing $\sharp$-labelled edges. Consequently its ingoing $\$$-labelled edge has not been removed. Now $|v'| < |v|$ and, by induction hypothesis, there is a vertex $u \in V_{\mathfrak{B}}$ and a word $\sigma' \in \{\sharp, \$\}^*$ such that $u \overset{\sigma'}{\dashrightarrow} v'$. Hence $u \overset{\sigma'\$}{\dashrightarrow} v$.
     - Subcase $l_k > 1$.

       Then $v = [t_{1,1} \dots t_{1,l_1}][t_{2,1} \dots t_{2,l_2}] \dots [t_{k,1} \dots t_{k,l_k}]$ and there is a vertex
       $$v' = [t_{1,1} \dots t_{1,l_1}][t_{2,1} \dots t_{2,l_2}] \dots [t_{k,1} \dots t_{k,l_k - 1}]$$
       such that $v' \xrightarrow{\sharp} v$. Now $|v'| < |v|$ and, by induction hypothesis, there is a vertex $u \in V_{\mathfrak{B}}$ and a word $\sigma' \in \{\sharp, \$\}^*$ such that $u \overset{\sigma'}{\dashrightarrow} v'$. Hence $u \overset{\sigma'\sharp}{\dashrightarrow} v$.

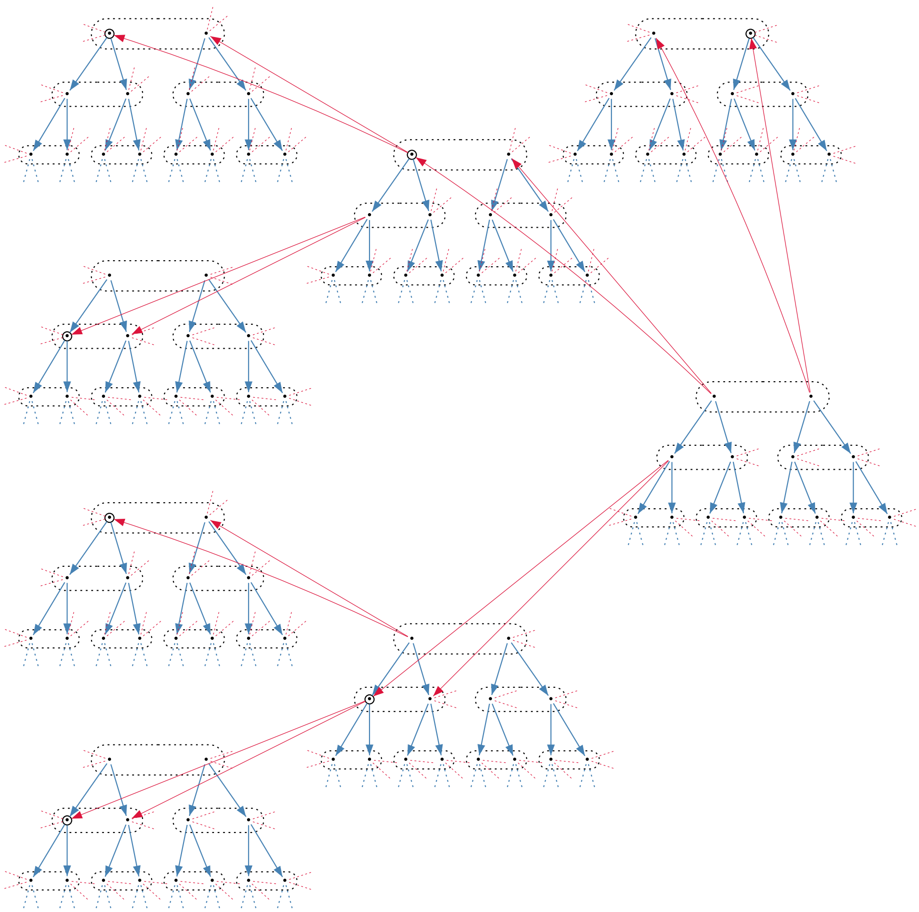
□

After defining the substructure $\mathfrak{D}_1$ of $(\mathfrak{D}^{\sharp,\&})^{\$}$, we shall define a structure $\mathfrak{D}_2$ which is a substructure of $(\mathfrak{D}^{\$})^{\sharp,\&}$. This is the second among the two noticeable substructures mentioned earlier between which the isomorphism is straightforward to establish. This isomorphism is helpful in building an isomorphism between $(\mathfrak{D}^{\sharp,\&})^{\$}$ and the structure that we are going to interpret within $(\mathfrak{D}^{\$})^{\sharp,\&}$ through three steps. After the first two steps, we will obtain $\mathfrak{D}_2$.

The first step towards $\mathfrak{D}_2$ consists in eliminating useless $\sharp$-labelled edges. We only keep such edges from every vertex to a copy of the original structure, here $\mathfrak{D}$, possessing an &-marked vertex. The result is similar to $\mathfrak{D}_1$ where every vertex has outgoing $\sharp$-labelled edges to its own copy of $\mathfrak{D}$ with &-marked clone of the vertex (by definition of $\mathfrak{D}^{\sharp,\&}$). However, such useless $\sharp$-labelled edges requires an extra care because the equivalence relation saying that two vertices belong to the same copy is not MSO-definable, except in a particular case where the Gaifman graph of the original structure is connected. In order to deal with the general case, we need to add auxiliary $\diamond$-labelled edges (not depicted in subsequent figures), where $\diamond \notin \Sigma \cup \{\sharp, \$\}$ is a new label, between every ordered pair of vertices of the original structure $\mathfrak{D}$ via an interpretation h defined in a usual way:

$$\begin{aligned} \delta(x) &:\Leftrightarrow \mathsf{true} \\ \theta_\diamond(x,y) &:\Leftrightarrow \mathsf{true} \\ \theta_a(\overline{x}) &:\Leftrightarrow a(\overline{x}) \qquad \text{for } a \in \Sigma \ . \end{aligned}$$

Thus in the example, instead of $(\mathfrak{D}^{\$})^{\sharp,\&}$, the construction really starts from $(h(\mathfrak{D})^{\$})^{\sharp,\&}$. After selecting appropriate $\sharp$-labelled edges, $\diamond$-labelled edges may be forgotten. In the example, this leads to the following structure $g_1((h(\mathfrak{D})^{\$})^{\sharp,\&})$:



The interpretation for g_1 is obvious:

$$\begin{aligned} \delta(x) \quad &:\Leftrightarrow \quad \mathsf{true} \\ \theta_\sharp(x,y) \quad &:\Leftrightarrow \quad x \xrightarrow{\sharp} y \wedge \exists z\,(\&(z) \wedge y \xrightarrow{\diamond} z) \\ \theta_a(\overline{x}) \quad &:\Leftrightarrow \quad a(\overline{x}) \qquad \text{for } a \in \Sigma \cup \{\&, \$\} \ . \end{aligned} \tag{4.1}$$

At the second step, we restrict the structure to the substructure induced by vertices that are accessible from the original structure via a $\{\sharp, \$\}^*$-labelled path by means of interpretation g_2:

$$\begin{aligned} \delta(x) \quad &:\Leftrightarrow \quad \exists\, y\,\big(\mathsf{root}(y) \wedge \mathsf{path}_{\{\sharp,\$\}^*}(y,x)\big) \\ \theta_a(\overline{x}) \quad &:\Leftrightarrow \quad a(\overline{x}) \qquad \text{for } a \in \Sigma \cup \{\sharp, \&, \$\} \end{aligned}$$

In the example, this leads to the following structure $\mathfrak{D}_2 := g_2(g_1((h(\mathfrak{D})^{\$})^{\sharp,\&}))$, depicted in Fig. 2, which is the second substructure we are looking for.

Like for $\mathfrak{D}_1$, it may be generalised from the example of $\mathfrak{D}_2$ that for each word $\kappa \in \Lambda^*$ over the set of decorated labels Λ, a path in a structure like $\mathfrak{D}_2$ labelled by κ is unique from a given vertex u.

Lemma 4.6. Let $\mathfrak{B}_2 := g_2(g_1((h(\mathfrak{B})^{\$})^{\sharp,\&}))$ where g_1 and g_2 are the MSO interpretations defined so far. Then the following holds.

1. For every word $\kappa \in \Lambda^*$ and each vertex $u \in V_{\mathfrak{B}_2}$, there is a unique path in $\mathfrak{B}_2$ starting at u and labelled by κ.

2. Each vertex of $\mathfrak{B}_2$ is accessible from $\mathfrak{B}$ via a path with label in Λ^*.

Proof:

1. We show that there is exactly one edge in $\mathfrak{B}_2$ with a given source and decorated label.

   Consider an edge of $\mathfrak{B}_2$ with source $u \in V_{\mathfrak{B}_2}$ and decorated label $\lambda \in \Lambda$. We know that

   $$u = [t_{1,1} \ldots t_{1,l_1}][t_{2,1} \ldots t_{2,l_2}] \ldots [t_{k,1} \ldots t_{k,l_k}], \quad \text{for some } t_{i,j} \in V_{\mathfrak{B}} \ .$$

   - case $\lambda = \sharp(s)$ for some $s \in V_{\mathfrak{B}}$

     We claim that this case occurs if, and only if, $u \xrightarrow{\sharp(s)} v$, where

     $$v = [t_{1,1} \ldots t_{1,l_1}][t_{2,1} \ldots t_{2,l_2}] \ldots [t_{k,1} \ldots t_{k,l_k}][t_{k,1} \ldots t_{k,l_k-1} s] \ .$$

     Indeed, $\sharp$-labelled edges of $\mathfrak{B}_2$ form a subset of the second-level iteration edges of $(\mathfrak{B}^{\$})^{\sharp,\&}$ and are precisely the $\sharp$-labelled edges of $g_1((h(\mathfrak{B})^{\$})^{\sharp,\&})$. According to the definition scheme of g_1 (see Equation (4.1)), all $\sharp$-labelled edges sharing the same source u target the vertices of the same copy of $\mathfrak{B}$, say $\mathfrak{B}'$, within a given copy

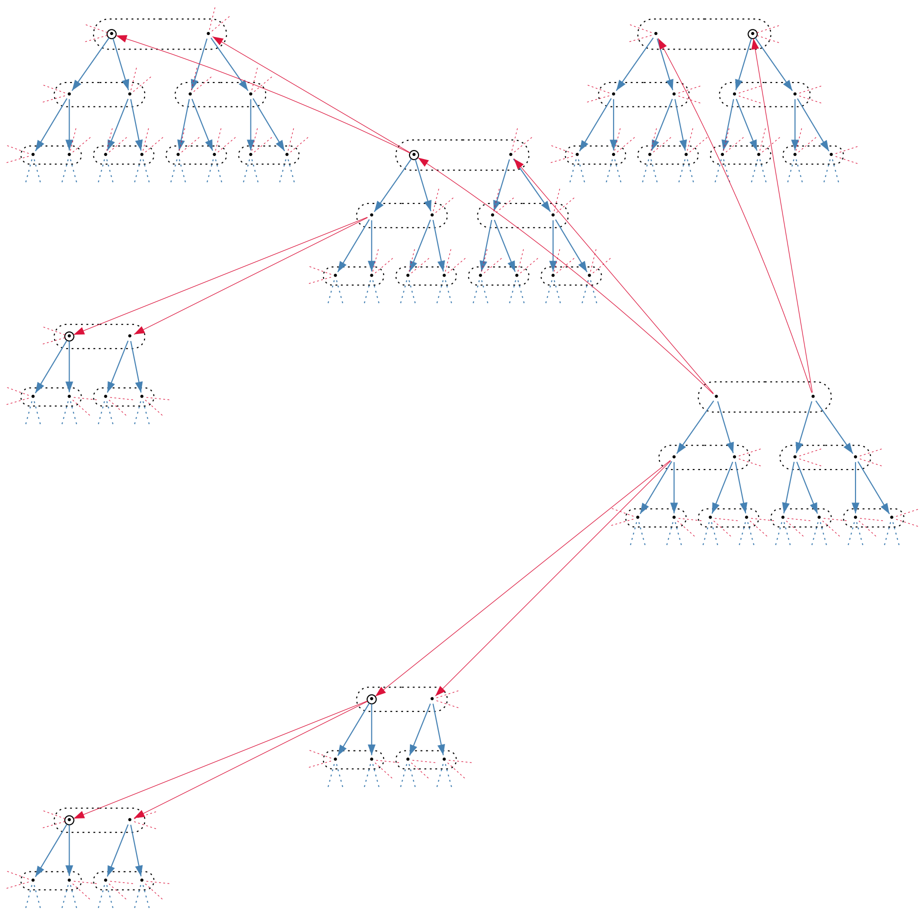


Figure 2. Substructre $\mathfrak{D}_2$

of $\mathfrak{B}^\$$. The latter copy of $\mathfrak{B}^\$$ has exactly one &-labelled vertex, say v', and this vertex is in $\mathfrak{B}'$. According to the definition of Muchnik's iteration, we have

$$v' = [t_{1,1} \ldots t_{1,l_1}][t_{2,1} \ldots t_{2,l_2}] \ldots [t_{k,1} \ldots t_{k,l_k}][t_{k,1} \ldots t_{k,l_k}]$$

i.e. $\mathsf{top}_0(v') = \mathsf{top}_0(\mathsf{pop}_0(v'))$. Now, within the same copy of $\mathfrak{B}$, vertices only differ by its top_{-1}. Thus

$$v = [t_{1,1} \ldots t_{1,l_1}][t_{2,1} \ldots t_{2,l_2}] \ldots [t_{k,1} \ldots t_{k,l_k}][t_{k,1} \ldots t_{k,l_k-1} s]$$

because v and v' are both in $\mathfrak{B}'$ whereas $\mathsf{top}_{-1}(v)$ corresponds to the decorated label of $u \xrightarrow{\sharp(s)} v$.

- case $\lambda = \$(s)$ for some $s \in V_{\mathfrak{B}}$

  This case occurs if, and only if, $u \xrightarrow{\$(s)} v$, where

$$v = [t_{1,1} \dots t_{1,l_1}][t_{2,1} \dots t_{2,l_2}] \dots [t_{k,1} \dots t_{k,l_k} s] \ .$$

2. Immediate from the definition scheme of g_2.

□

As the example has been developed, two structures have been distinguished:

- $\mathfrak{D}_1$ which is a substructure of $(\mathfrak{D}^{\sharp,\&})^{\$}$ (see Fig 1),
- $\mathfrak{D}_2$ which is a substructure of $(\mathfrak{D}^{\$})^{\sharp,\&}$ (see Fig 2).

The reader may notice that $\mathfrak{D}_1$ and $\mathfrak{D}_2$ are isomorphic. This fact is generalised in the following lemma.

Lemma 4.7. Let $\mathfrak{B}_1$ be a substructure obtained from $(\mathfrak{B}^{\sharp,\&})^{\$}$ by forgetting every $\$$-labelled edge, the target of which is also the target of a $\sharp$-labelled edge. Let $\mathfrak{B}_2 := g_2(g_1((h(\mathfrak{B})^{\$})^{\sharp,\&}))$ where g_1 and g_2 are the MSO interpretations defined so far. Then $\mathfrak{B}_1 \equiv \mathfrak{B}_2$.

Proof:
Putting together the two statements of Lemma 4.5, we conclude that there is a bijection between $V_{\mathfrak{B}_1}$ and $V_{\mathfrak{B}} \times \Lambda^*$. Similarly, from Lemma 4.6, we get a bijection between $V_{\mathfrak{B}} \times \Lambda^*$ and $V_{\mathfrak{B}_2}$. Putting together these bijections, we obtain a bijection $\mu: V_{\mathfrak{B}_1} \to V_{\mathfrak{B}_2}$ defined by

$$\mu(v_1) = v_2 \quad :\Leftrightarrow \quad \exists\, u \in V_{\mathfrak{B}}\ \exists\, \kappa \in \Lambda^*\ (u \overset{\kappa}{\underset{\mathfrak{B}_1}{\dashrightarrow}} v_1 \wedge u \overset{\kappa}{\underset{\mathfrak{B}_2}{\dashrightarrow}} v_2) \tag{i}$$

for all $v_1 \in V_{\mathfrak{B}_1}$ and $v_2 \in V_{\mathfrak{B}_2}$. We establish that this bijection is in fact an isomorphism of relational structures $\mu: \mathfrak{B}_1 \to \mathfrak{B}_2$.

- We have $u \xrightarrow[\mathfrak{B}_1]{\sharp} v \Leftrightarrow \mu(u) \xrightarrow[\mathfrak{B}_2]{\sharp} \mu(v)$ and $u \xrightarrow[\mathfrak{B}_1]{\$} v \Leftrightarrow \mu(u) \xrightarrow[\mathfrak{B}_2]{\$} \mu(v)$, for all $u, v \in V_{\mathfrak{B}_1}$. This follows directly from the definition (i) of μ via Λ^*-labelled paths.
- We claim $a(\overline{v}) \in \mathfrak{B}_1 \Leftrightarrow a(\mu(\overline{v})) \in \mathfrak{B}_2$ for each $a \in \Sigma$ and every $\overline{v} \in V_{\mathfrak{B}_1}^{\alpha(a)}$.
  Indeed, $a(\overline{v}) \in \mathfrak{B}_1$,

  iff all vertices of $\overline{v} = (v_1, \dots, v_{\alpha(a)})$ are in the same copy of $\mathfrak{B}$ and $a(\overline{s}) \in \mathfrak{B}$, where $\overline{s} = (s_1, \dots, s_{\alpha(a)})$ with $s_i = \mathsf{top}_{-1}(v_i)$, for $i \in [\alpha(a)]$, viz., there is an original hyperedge $a(\overline{s})$ corresponding to the copy $a(\overline{v})$,

  iff all vertices of $\overline{v}$ have the same ancestor for $\sharp$ (resp. $\$$), say u, and $a(\overline{s}) \in \mathfrak{B}$

  iff $u \xrightarrow[\mathfrak{B}_1]{\sharp(s_i)} v_i$ (resp. $u \xrightarrow[\mathfrak{B}_1]{\$(s_i)} v_i$) for $i \in [\alpha(a)]$ and $a(\overline{s}) \in \mathfrak{B}$

iff $\mu(u) \xrightarrow[\mathfrak{B}_2]{\sharp(s_i)} \mu(v_i)$ (resp. $\mu(u) \xrightarrow[\mathfrak{B}_2]{\$(s_i)} \mu(v_i)$) for $i \in [\alpha(a)]$ and $a(\overline{s}) \in \mathfrak{B}$

iff all vertices of $\mu(\overline{v})$ have the same ancestor for $\sharp$ (resp. $\$$), say $\mu(u)$, and $a(\overline{s}) \in \mathfrak{B}$

iff all vertices of $\mu(\overline{v}) = (\mu(v_1), \ldots, \mu(v_{\alpha(a)}))$ are in the same copy of $\mathfrak{B}$ and $a(\overline{s}) \in \mathfrak{B}$, where $\overline{s} = (s_1, \ldots, s_{\alpha(a)})$ with $s_i = \mathsf{top}_{-1}(\mu(v_i))$, for $i \in [\alpha(a)]$,

iff $a(\mu(\overline{v})) \in \mathfrak{B}_2$.

- We claim $\&(v) \in \mathfrak{B}_1 \Leftrightarrow \&(\mu(v)) \in \mathfrak{B}_2$ for every $v \in V_{\mathfrak{B}_1}$.

$\Rightarrow$ Assume $\&(v) \in \mathfrak{B}_1$. Then

$$v = [t_{1,1} \ldots t_{1,l_1}][t_{2,1} \ldots t_{2,l_2}] \ldots [t_{k,1} \ldots t_{k,l_k} ss]$$

for some $t_{i,j} \in V_{\mathfrak{B}}$ with $i \in [k], j \in [l_i]$ and some $s \in V_{\mathfrak{B}}$, and there exists

$$u = [t_{1,1} \ldots t_{1,l_1}][t_{2,1} \ldots t_{2,l_2}] \ldots [t_{k,1} \ldots t_{k,l_k} s]$$

such that $u \xrightarrow[\mathfrak{B}_1]{\sharp(s)} v$. There are 3 disjoint cases.

1. u has neither $\sharp$- nor $\$$-ancestor, viz., $u = [s]$, whereas $k = 1$ and $l_k = 0$. Then $\mu(u) = u$. In $\mathfrak{B}_2$, u has an outgoing $\sharp(s)$-labelled edge with target $[s][s]$. Thus $\mu(v) = [s][s]$ which is, for the second level iteration, the clone of $[s]$. Hence $\&(\mu(v)) \in \mathfrak{B}_2$.
2. u has a $\sharp$-ancestor, say w. Then $w \xrightarrow[\mathfrak{B}_1]{\sharp(s)} u \xrightarrow[\mathfrak{B}_1]{\sharp(s)} v$ and $\mu(w) \xrightarrow[\mathfrak{B}_2]{\sharp(s)} \mu(u) \xrightarrow[\mathfrak{B}_2]{\sharp(s)} \mu(v)$ as already established. Consequently $\mathsf{top}_{-1}(\mu(u)) = \mathsf{top}_{-1}(\mu(v)) = s$ and $\mathsf{top}_0(\mu(u)) = s_1 \ldots s_p s$, viz., $\mu(u) = \xi \lrcorner [s_1 \ldots s_p s]$ for some $\xi \in (V_{\mathfrak{B}})^2_*$ and $s_1, \ldots, s_p \in V_{\mathfrak{B}}$. According to the definition scheme of g_1 (see Equation (4.1)), all $\sharp$-labelled edges in $\mathfrak{B}_2$ sharing the same source u target the vertices of the same copy of $\mathfrak{B}$, say $\mathfrak{B}'$, within a given copy of $\mathfrak{B}^\$$. The latter copy of $\mathfrak{B}^\$$ has exactly one &-labelled vertex. This vertex is in $\mathfrak{B}'$ and, according to the definition of Muchnik's iteration, it is precisely $\xi \lrcorner [s_1 \ldots s_p s][s_1 \ldots s_p s]$. Obviously, it is the only vertex in $\mathfrak{B}'$ such that its top_{-1} is s. Thus
$$\mu(v) = \xi \lrcorner [s_1 \ldots s_p s][s_1 \ldots s_p s]$$
and $\&(\mu(v)) \in \mathfrak{B}_2$.
3. u has a $\$$-ancestor, say w. This means that $k > 1$ but $l_k = 0$. We have therefore
$$\begin{aligned} w &= [t_{1,1} \ldots t_{1,l_1}][t_{2,1} \ldots t_{2,l_2}] \ldots [t_{k-1,1} \ldots t_{k-1,l_{k-1}}], \\ u &= [t_{1,1} \ldots t_{1,l_1}][t_{2,1} \ldots t_{2,l_2}] \ldots [t_{k-1,1} \ldots t_{k-1,l_{k-1}}][s], \\ v &= [t_{1,1} \ldots t_{1,l_1}][t_{2,1} \ldots t_{2,l_2}] \ldots [t_{k-1,1} \ldots t_{k-1,l_{k-1}}][ss] \ . \end{aligned}$$
Then $w \xrightarrow[\mathfrak{B}_1]{\$(s)} u \xrightarrow[\mathfrak{B}_1]{\sharp(s)} v$ and $\mu(w) \xrightarrow[\mathfrak{B}_2]{\$(s)} \mu(u) \xrightarrow[\mathfrak{B}_2]{\sharp(s)} \mu(v)$ as already established. Consequently $\mathsf{top}_{-1}(\mu(u)) = \mathsf{top}_{-1}(\mu(v)) = s$ and we conclude similarly to Case 2.

$\Leftarrow$ Assume $\&(\mu(v)) \in \mathfrak{B}_2$. Then

$$\mu(v) = [t_{1,1}\dots t_{1,l_1}][t_{2,1}\dots t_{2,l_2}]\dots[t_{k,1}\dots t_{k,l_k}s][t_{k,1}\dots t_{k,l_k}s]$$

for some $t_{i,j} \in V_{\mathfrak{B}}$ with $i \in [k], j \in [l_i]$ and some $s \in V_{\mathfrak{B}}$, and there exists

$$\mu(u) = [t_{1,1}\dots t_{1,l_1}][t_{2,1}\dots t_{2,l_2}]\dots[t_{k,1}\dots t_{k,l_k}s]$$

such that $\mu(u) \xrightarrow[\mathfrak{B}_1]{\sharp(s)} \mu(v)$. There are three disjoint cases.

1. $\mu(u)$ has neither $\sharp$- nor \$-ancestor, viz., $\mu(u) = [s]$, whereas $k = 0$. Then $\mu(u) = u$. In $\mathfrak{B}_1$, u has an outgoing $\sharp(s)$-labelled edge. Its target is therefore $[ss]$. Thus $v = [ss]$ which is the clone of $[s]$. Hence $\&(v) \in \mathfrak{B}_1$.
2. $\mu(u)$ has a $\sharp$-ancestor, say $\mu(w)$. Then $\mu(w) \xrightarrow[\mathfrak{B}_2]{\sharp(s)} \mu(u) \xrightarrow[\mathfrak{B}_2]{\sharp(s)} \mu(v)$ and $w \xrightarrow[\mathfrak{B}_1]{\sharp(s)} u \xrightarrow[\mathfrak{B}_1]{\sharp(s)} v$ as already established. Consequently $\mathsf{top}_{-1}(u) = \mathsf{top}_{-1}(v) = s$ and $\mathsf{top}_0(u) = [s_1\dots s_n s]$, viz.,

$$u = \xi\lrcorner[s_1\dots s_p s]$$

for some $\xi \in (V_{\mathfrak{B}})^2_*$ and $s_1,\dots,s_p \in V_{\mathfrak{B}}$. Now $v = \xi\lrcorner[s_1\dots s_p ss]$, because

$$u \xrightarrow[\mathfrak{B}_1]{\sharp(s)} v \ .$$

Hence $\&(v) \in \mathfrak{B}_1$.
3. $\mu(u)$ has a \$-ancestor, say $\mu(w)$. Then $\mu(w) \xrightarrow[\mathfrak{B}_2]{\$(s)} \mu(u) \xrightarrow[\mathfrak{B}_2]{\sharp(s)} \mu(v)$ and $w \xrightarrow[\mathfrak{B}_1]{\$(s)} u \xrightarrow[\mathfrak{B}_1]{\sharp(s)} v$ as already established. Consequently $\mathsf{top}_{-1}(u) = \mathsf{top}_{-1}(v) = s$ and we conclude like in the latter case.

$\square$

According to the above lemma $\mathfrak{D}_1$ and $\mathfrak{D}_2$ are isomorphic. Remember that $\mathfrak{D}_1$ is a substructure obtained from $(\mathfrak{D}^{\sharp,\&})^\$$ by forgetting every \$-labelled edge, the target of which is also the target of a $\sharp$-labelled edge. Thus, from $\mathfrak{D}_1$ we obtain $(\mathfrak{D}^{\sharp,\&})^\$$, by putting back the forgotten edges. We put a \$-labelled edge from a vertex x to a vertex y, whenever there is a $\$\sharp^+$-labelled path from x to y or we keep an existing edge $x \xrightarrow{\$} y$. This is defined by the following interpretation g_3:

$$\begin{aligned} \delta(x) &:\Leftrightarrow \mathsf{true} \\ \theta_\$(x,y) &:\Leftrightarrow \mathsf{path}_{\$\sharp^*}(x,y) \\ \theta_a(\overline{x}) &:\Leftrightarrow a(\overline{x}) \qquad \text{for } a \in \Sigma \cup \{\sharp, \&\} \end{aligned}$$

But since $\mathfrak{D}_1 \equiv \mathfrak{D}_2$, we have

$$(\mathfrak{D}^{\sharp,\&})^{\$} = g_3(\mathfrak{D}_1) \equiv g_3(\mathfrak{D}_2) = g_3(g_2(g_1((h(\mathfrak{D})^{\$})^{\sharp,\&}))) \ .$$

Thus interpretation g_3 is the third and the final step in building from $(\mathfrak{D}^{\$})^{\sharp,\&}$ a structure isomorphic to $(\mathfrak{D}^{\sharp,\&})^{\$}$. The constructions developed around this example are generalised in a straightforward way. These generalisations let us complete the pending proof of Lemma 4.3.

Proof of Lemma 4.3:
Let $\mathfrak{B}_1$ be a substructure obtained from $(\mathfrak{B}^{\sharp,\&})^{\$}$ by forgetting every $\$$-labelled edge, the target of which is also the target of a $\sharp$-labelled edge. We have

$$(\mathfrak{B}^{\sharp,\&})^{\$} = g_3(\mathfrak{B}_1)$$

since interpretation g_3 adds formerly forgotten $\$$-labelled edges.
Let $\mathfrak{B}_2 := g_2(g_1((h(\mathfrak{B})^{\$})^{\sharp,\&}))$. According to Lemma 4.7

$$\mathfrak{B}_1 \ \equiv \ \mathfrak{B}_2 \ .$$

Hence

$$(\mathfrak{B}^{\sharp,\&})^{\$} = g_3(\mathfrak{B}_1) \equiv g_3(\mathfrak{B}_2) = g_3(g_2(g_1((h(\mathfrak{B})^{\$})^{\sharp,\&}))) \ .$$

Therefore, there exist MSO interpretations $g := g_3 \circ g_2 \circ g_1$ and h, such that

$$(\mathfrak{B}^{\sharp,\&})^{\$} \equiv g((h(\mathfrak{B})^{\$})^{\sharp,\&}) \ .$$

□

By closing the proof of Lemma 4.3 we have just established the main result of this paper, namely Theorem 4.1. This theorem states that every level of the iteration hierarchy of relational structures is closed under basic iteration.

5. Final remarks

From the proof of the main result we may wish to extract a «normal form» for building a relational structure of level n from a finite structure by combining MSO-interpretations with n or $n-1$ Muchnik's iterations and 0 or more of basic iterations. In this normal form all basic iterations are pushed onto level 1 except, possibly one basic iteration applied at level 0. The induction basis shows how the former may be eliminated in a representation of level 1 structures as suffix-recognisable structures. However, instead of this concrete representation, one may wish a more abstract representation where a level 1 structure is obtained by MSO-interpretation from the infinite complete binary tree. This raises the question how to eliminate basic iterations using an MSO-interpretation.

Although the hierarchy based on Shelah-Stupp's iteration collapses at level 1, one may wonder if within level 1, one may define a finer strict hierarchy where one climbs up from one layer to the next layer via basic iteration. If so one may expect that such a layer structuring is transferred to every level of Muchnik's iteration.

Acknowledgements

We wish to thank anonymous referees for their help in improving the paper via numerous remarks they provided us.

References

[1] Courcelle B. The Expression of Graph Properties and Graph Transformations in Monadic Second–Order Logic. In: Rozenberg G (ed.), Handbook of Graph Grammars and Computing by Graph Transformation, volume 1, pp. 313–400. World Scientific, 1997.

[2] Büchi JR. On a decision method in restricted second order arithmetic. In: Logic, Methodology and Philosophy of Science (Proc. 1960 Internat. Congr). Stanford University Press, 1962 pp. 1–11.

[3] Trakhtenbrot B. Finite automata and the logic of monadic predicates. *Doklady Akademii Nauk SSSR*, 1961. **140**:326–329.

[4] Elgot CC. Decision Problems for Finite Automata Design and Related Arithmetics. *Transactions of the American Mathematical Society*, 1961. **98**:21–52.

[5] Rabin MO. Decidability of Second–order Theories and Automata on Infinite Trees. *Transactions of the American Mathematical Society*, 1969. **141**:1–35.

[6] Shelah S. The Monadic Theory of Order. *Annals of Mathematics*, 1975. **102**:379–419.

[7] Stupp J. The Lattice Model is Recursive in the Original Model. The Hebrew University, 1975.

[8] Courcelle B, Engelfriet J. Graph structure and Monadic Second-Order Logic, a Language Theoretic Approach. Cambridge University Press, 2012.

[9] Semenov AL. Decidability of Monadic Theories. In: Chytil M, Koubek V (eds.), Mathematical Foundations of Computer Science, LNCS 176. Springer, Praha, 1984 pp. 162–175.

[10] Walukiewicz I. Monadic second–order logic on tree–like structures. *Theoretical Computer Science*, 2002. **275**(1–2):311–346. doi:10.1016/S0304-3975(01)00185-2.

[11] Muller DE, Schupp PE. The Theory of Ends, Pushdown Automata and Second–order Logic. *Theoretical Computer Science*, 1985. **37**:51–75.

[12] Caucal D. On the Regular Structure of Prefix Rewriting. *Theoretical Computer Science*, 1992. **106**:61–86.

[13] Courcelle B. The Monadic Second–order Logic of Graphs, II: Infinite Graphs of Bounded Width. *Mathematical System Theory*, 1989. **21**:187–221.

[14] Courcelle B. The Monadic Second–order Logic of Graphs, VII: Graphs as Relational Structures. *Theoretical Computer Science*, 1992. **101**:3–33.

[15] Barthelmann K. On Equational Simple Graphs. Technical Report 9/97, Johannes Gutenberg Universität, Mainz, 1998.

[16] Caucal D. On Infinite Transition Graphs Having a Decidable Monadic Second–order Theory. *Theoretical Computer Science*, 2003. **290**(1):79–115. doi:10.1016/S0304-3975(01)00089-5.

[17] Siefkes D. Decidable Extensions of Monadic Second-order Successor Arithmetics. Mannheim: Bibilogr. Inst., 1969 pp. 341–472.

[18] Elgot CC, Rabin MO. Decidability and Undecidability of Extensions of the Second–order Theory of Successor. *Journal of Symbolic Logic*, 1966. **31**:169–181.

[19] Thomas W. The Theory of Successor with an Extra Predicate. *Mathematische Annalen*, 1978. **237**:121–132.

[20] Caucal D. Sur des graphes infinis réguliers. Habilitation, Université de Rennes I, 1998.

[21] Courcelle B, Walukiewicz I. Monadic Second–Order Logic, Graph Coverings and Unfoldings of Transition Systems. *Annals of Pure and Applied Logic*, 1998. **92**:35–62.

[22] Knapik T, Niwiński D, Urzyczyn P. Deciding Monadic Theories of Hyperalgebraic Trees. In: Abramsky S (ed.), 5 th International Conference on Typed Lambda Calculi and Applications, LNCS 2044. Kraków, 2001 pp. 253–267. doi:10.1007/3-540-45413-6_21.

[23] Knapik T, Niwiński D, Urzyczyn P. Higher–Order Pushdown Trees are Easy. In: Nielsen M, Engberg U (eds.), FoSSaCS, LNCS 2303. Grenoble, 2002 pp. 205–222. doi:10.1007/3-540-45931-6_15.

[24] Engelfriet J, Schmidt EM. IO and OI. I. *Journal of Computer and System Sciences*, 1977. **15**:328–353.

[25] Damm W. The IO– and OI–hierarchies. *Theoretical Computer Science*, 1982. **20**(2):95–208.

[26] Caucal D. On Infinite Terms Having a Decidable Monadic Theory. In: Diks K, Rytter W (eds.), MFCS '2002, LNCS 2420. Springer, Warsaw, 2002 pp. 165–176. doi:10.1007/3-540-45687-2_13.

[27] Knapik T, Niwiński D, Urzyczyn P, Walukiewicz I. Unsafe Grammars and Panic Automata. In: Caires L, Italiano GF, Monteiro L, Palamidessi C, Yung M (eds.), ICALP, LNCS 3580. Lisbon, 2005 pp. 1450–1461.

[28] Ong CL. On Model-Checking Trees Generated by Higher-Order Recursion Schemes. In: LICS. 2006 pp. 81–90. doi:10.1109/LICS.2006.38.

[29] Carayol A, Wöhrle S. The Caucal Hierarchy of Infinite Graphs in Terms of Logic and Higher-Order Pushdown Automata. In: FSTTCS, volume 2914 of *Lecture Notes in Computer Science*. 2003 pp. 112–123. doi:10.1007/978-3-540-24597-1_10.

[30] Blumensath A. Axiomatising Tree-Interpretable Structures. *Theory of Computing Systems*, 2004. **37**(1):3–27. doi:10.1007/s00224-003-1104-8.

[31] Karhumäki J, Kunc M, Okhotin A. Computing by commuting. *Theor. Comput. Sci.*, 2006. **356**(1-2):200–211. doi:10.1016/j.tcs.2006.01.051.

[32] Altenbernd J. Reachability over word rewriting systems. Ph.D. thesis, RWTH Aachen University, 2009.

[33] Angluin D, Hoover DN. Regular Prefix Relations. *Mathematical Systems Theory*, 1984. **17**(3):167–191. doi:10.1007/BF01744439.

[34] Läuchli H, Savioz C. Monadic Second Order Definable Relations on the Binary Tree. *J. Symb. Log.*, 1987. **52**(1):219–226. doi:10.2307/2273878.

[35] Carayol A, Colcombet T. On Equivalent Representations of Infinite Structures. In: ICALP, volume 2719 of *Lecture Notes in Computer Science*. 2003 pp. 599–610.

[36] Weyer M. Decidability of S1S and S2S. In: Grädel E, Thomas W, Wilke T (eds.), Automata, Logics, and Infinite Games: A Guide to Current Research, volume LNCS 2500. 2001 pp. 207–230. doi:10.1007/3-540-36387-4_12.