ELSEVIER

Annals of Pure and Applied Logic 92 (1998) 35–62

ANNALS OF PURE AND APPLIED LOGIC

Monadic second-order logic, graph coverings and unfoldings of transition systems

Bruno Courcelle[a,*], Igor Walukiewicz[b,1]

[a] *Laboratoire d' Informatique (LaBRI), Université Bordeaux I, 351, Cours de la Libération, F-33405 Talence Cedex, France*

[b] *Institute of Informatics, Warsaw University, Banacha 2, 02-097 Warsaw, Poland*

Received 18 July 1995; accepted 25 September 1997

Abstract

We prove that every monadic second-order property of the unfolding of a transition system is a monadic second-order property of the system itself. An unfolding is an instance of the general notion of graph covering. We consider two more instances of this notion. A similar result is possible for one of them but not for the other. © 1998 Published by Elsevier Science B.V. All rights reserved.

Keywords: Second-order logic; Rabin automaton; Infinite tree; Semantics; Transition Systems; Graph covering

AMS classification: Primary: 68Q55; secondary: 03C80, 03C85

1. Introduction

A transition system is a directed graph (satisfying some conditions); the edges of the graph are called *transitions* and its vertices are called *states*. A transition system can be seen as an abstract form of a program, and the infinite tree obtained by unfolding (or unravelling) of it can be seen as its behaviour. Since transition systems and their behaviours can be represented by logical structures, one can express their properties by logical formulas. We consider here monadic second-order logic (MSOL) as an appropriate logical language because it subsumes many other formalisms, like the μ-calculus or temporal logics (see [6, 9]), and it is decidable on many structures and in particular on infinite binary trees (by Rabin's Theorem, see [14]).

* Corresponding author. E-mail: courcell@labri.u-bordeaux.fr; internet: http://dept-info.u-bordeaux.fr./~courcell/ActSci.html.

[1] Partially supported by Polish KBN grant No. 2 P301 009 06. Part of this work was done at Basic Research in Computer Science, Centre of the Danish National Research Foundation.

PII S0168-0072(97)00048-1

We consider the following conjecture from Courcelle [4]. Suppose $\mathscr{Q}$ is a class of transition systems defined by a MSOL formula. Define the class $\mathscr{P}$ of transition systems R by

$$\mathscr{P} = \{R\colon \mathsf{Un}(R) \in \mathscr{Q}\},$$

where $\mathsf{Un}(R)$ denotes the unfolding of R. The conjecture is that the class $\mathscr{P}$ is definable by a MSOL formula and this formula can be effectively constructed from the one defining $\mathscr{Q}$.

This conjecture was proved in [4] for deterministic transition systems (possibly with infinitely many states) and we prove it here for the class of all systems with at most countable outdegrees.

This new proof is independent of that in [4] and uses a different technique, based on a notion of covering. A *covering* of a transition system (or more generally of a graph) G is a surjective homomorphism $h: G' \to G$ (where G' is another transition system or graph) the restriction of which to the "neighbourhood" of every state or vertex of G' is an isomorphism. We say that h is a k-covering if $h^{-1}(x)$ has a cardinality $\leqslant k$ for each state or vertex x of G. For a transition system if we take as the "neighbourhood" of a state the set of transitions outgoing from it, then there exists a universal covering which is precisely the unfolding. The main lemma (Lemma 14) roughly says that for every MSOL formula $\exists X\,.\,\varphi(X)$ there is an integer k, s.t., for every transition system R: if $\mathsf{Un}(R) \models \exists X\,.\,\varphi(X)$ then there exists a k-covering R' of R and a subset S of R' with $\mathsf{Un}(R') \models \varphi(\mathsf{Un}(S))$. In other words, one can find a sufficiently regular witness S for the existential quantification.

The notion of "neighbourhood" is a "parameter" of the notion of covering. In the case of graphs, we examine two more possibilities for defining coverings. The first possibility is to take the set of edges incident to a vertex as its neighbourhood. Then the results concerning transition systems extend for this notion of covering but only when we allow quantification over edges: every monadic second-order property of the universal covering of a (finite or infinite) graph (relative to this notion of neighbourhood) can be expressed as a monadic second-order property of the graph provided we can quantify over edges of the graph.

A second possibility is to take as neighbourhood of a vertex the subgraph induced by the vertices at a distance at most 1. There exists a corresponding notion of universal covering. However, we exhibit a finite graph G, the universal covering of which is the infinite grid. This shows that the result does not hold here because the monadic theory of the infinite grid is undecidable whereas that of G is decidable (because G is finite).

Finally, we relate unfoldings of transition systems with a construction by Shelah [12] and Stupp [13], extended by Muchnik (reported in [11]), about which we raise some questions that indicate possible developments of the present work.

This paper is organised as follows. *Section* 1 deals with transition systems, their coverings and automata, *Section* 2 deals with monadic second-order logic, *Sections* 3 *and* 4 present some technical lemmas, *Section* 5 gives the main proof, *Section* 6 discusses the Shelah–Stupp–Muchnik construction, *Section* 7 concerns coverings of graphs, and *Section* 8 reviews some open questions.

2. Transition systems

We consider directed graphs G, defined by means of sets: V_G (vertices), E_G (edges) and the source and target mappings, respectively, $\mathsf{src}_G : E_G \to V_G$, $\mathsf{tgt}_G : E_G \to V_G$. We will consider only graphs with finite or countable degrees of vertices. Transition systems are special (labelled) graphs as defined below.

Let n,m be natural numbers and $m \geqslant 1$. A *transition system* of type (n,m) is a tuple $R=(G,x,P_{1R},\dots,P_{nR},Q_{1R},\dots,Q_{mR})$, where G is a directed graph, x is a vertex called the *root of* R from which all other vertices are accessible by a directed path, $P_{1R},\dots,P_{nR}$ are sets of vertices and $Q_{1R},\dots,Q_{mR}$ is a partition of the set of edges. As in the case of graphs we will restrict to transition systems with vertices of at most countable degree. We call such transition systems *countably branching*.

A vertex of G is called a *state* of R and an edge is called a *transition*. A transition in Q_{iR} is said to be *of type* i. In order to have uniform notation, we let:

S_R be the set of states of R, T_R be its set of transitions, $root_R$ be its root, P_{iR} be the ith set of states, Q_{iR} be the set of transitions of type i, $\mathsf{src}_R=\{(t,s)\colon t\in T_R, s\in S_R,\ s$ is the origin (or source) of $t\}$ and $\mathsf{tgt}_R=\{(t,s)\colon t\in T_R, s\in S_R, s$ is the target of $t\}$.

For convenience we shall also write in some cases $s=\mathsf{src}_R(t)$ (or $s=\mathsf{tgt}_R(t)$) if $(t,s)\in\mathsf{src}_R$ (or $(t,s)\in\mathsf{tgt}_R$, respectively).

Let R and R' be two transition systems of type (n,m). We write $R\subseteq R'$ iff

$$S_R \subseteq S_{R'},$$
$$T_R \subseteq T_{R'},$$
$$root_R = root_{R'},$$
$$P_{iR} = P_{iR'} \cap S_R,$$
$$Q_{iR} = Q_{iR'} \cap T_R,$$
$$\mathsf{src}_R = \mathsf{src}_{R'} \cap (T_R \times S_R),$$
$$\mathsf{tgt}_R = \mathsf{tgt}_{R'} \cap (T_R \times S_R).$$

A *homomorphism* $h : R \to R'$ is a mapping $S_R \cup T_R \to S_{R'} \cup T_{R'}$ such that

$$h(S_R) \subseteq S_{R'},$$
$$h(T_R) \subseteq T_{R'},$$
$$h(\mathsf{src}_R(t)) = \mathsf{src}_{R'}(h(t)) \quad \text{for all } t \in T_R,$$
$$h(\mathsf{tgt}_R(t)) = \mathsf{tgt}_{R'}(h(t)) \quad \text{for all } t \in T_R,$$
$$h(root_R) = root_{R'},$$
$$s \in P_{iR} \text{ iff } h(s) \in P_{iR'}, \quad \text{for all } s \in S_R \text{ and } i = 1,\dots,n,$$
$$t \in Q_{iR} \text{ iff } h(t) \in Q_{iR'}, \quad \text{for all } t \in T_R \text{ and } i = 1,\dots,m.$$

A homomorphism $h: R \to R'$ is a *covering* (we shall also say that *R is a covering of R'*) if it is surjective and for every state $s \in S_R$, h is a bijection of $\mathsf{out}_R(s)$ onto $\mathsf{out}_{R'}(h(s))$. (We denote by $\mathsf{out}_R(s)$ the set of transitions t of R such that $\mathsf{src}_R(t) = s$.) We say that h is a *k-covering* if for every $s \in S_{R'}$ the set $h^{-1}(s)$ has at most k elements.

A *path* in R is a finite or infinite sequence of transitions $(t_1, t_2, \ldots)$ such that $root_R = \mathsf{src}_R(t_1)$ and for each i, $\mathsf{tgt}_R(t_i) = \mathsf{src}_R(t_{i+1})$. If this sequence is finite, the target of the last transition is called the *end of the path*.

Fact 1. *If h is a homomorphism $R \to R'$ then the image of every path of R is a path of R'. If furthermore, h is a covering, then every path in R' is the image by h of a unique path in R.*

We now define the *unfolding* $\mathsf{Un}(R)$ of a transition system R; this is a tree, and we shall consider it as the *behaviour* of R.

We let N_R be the set of finite paths in R. We have, in particular, the empty path linking the root to itself. N_R is the set of nodes of $\mathsf{Un}(R)$.

If p and $p' \in N_R$, we define an edge $p \longrightarrow p'$ (equivalently a transition) of type i iff p' extends p by exactly one transition of R of type i. We let Q_i^* denote the set of such transitions.

We let $h_R : N_R \to S_R$ associate with every finite path its end. We let also P_i^* denote the set $h_R^{-1}(P_{iR})$. We obtain a transition system $\mathsf{Un}(R)$ of type (n,m) by defining:

$$S_{\mathsf{Un}(R)} = N_R,$$

$$T_{\mathsf{Un}(R)} = Q_1^* \cup \cdots \cup Q_m^*,$$

$$root_{\mathsf{Un}(R)} = \varepsilon,$$

$$P_{i\mathsf{Un}(R)} = P_i^*,$$

$$Q_{i\mathsf{Un}(R)} = Q_i^*.$$

Fact 2. *The map h_R extends in a unique way to a homomorphism $\mathsf{Un}(R) \to R$ which is a covering.*

Fact 3. *If $m: R \to R'$ is a covering, then there exists a unique isomorphism $\bar{m}$: $\mathsf{Un}(R) \to \mathsf{Un}(R')$ such that $h_{R'} \circ \bar{m} = m \circ h_R$.*

Because of these properties, $\mathsf{Un}(R)$ will be called the *universal covering* of R. The terminology is borrowed from algebraic topology where the notion of universal covering of a topological space is a basic notion.

A transition system of type (n,m) is *deterministic* if no two transitions with the same source belong to the same set Q_i. It is *complete deterministic* if, in addition, each state has exactly m outgoing transitions.

Fact 4. *Let R and R' be complete deterministic transition systems of the same type. There is at most one homomorphism $R \to R'$ and such a homomorphism is a covering. It exists iff there exists a mapping $h : S_R \to S_{R'}$ such that: (a) $h(root_R) = root_{R'}$, (b) for every transition $x \to x'$ of R there is in R' a transition $h(x) \to h(x')$ of the same type, (c) for every $x \in S_R$ and every i, we have $x \in P_{iR}$ iff $h(x) \in P_{iR'}$.*

2.1. Parity automata and transition systems

In this section we introduce parity automata and prove a lemma about the runs of such automata. This lemma will be used to prove the regularisation lemma (Lemma 14).

We denote by $\mathscr{T}$ the infinite complete binary tree. Its nodes are (as usual) identified with words from $\{1,2\}^*$. It is a complete deterministic transition system of type $(0,2)$. We denote by $\mathscr{T}_n$ the set of tuples of the form $(\mathscr{T}, P_1, \ldots, P_n)$, where $P_1, \ldots, P_n$ are sets of nodes of $\mathscr{T}$. These tuples can be considered as infinite complete binary trees the nodes of which are labelled by subsets of $\{1, \ldots, n\}$; they are complete deterministic transition systems of type $(n, 2)$.

A *parity-automaton* is a tuple $\mathscr{A} = \langle S, \Sigma, I, \delta, \Omega \rangle$, where

- S is a finite nonempty set of *states*;
- Σ is a finite set called the *alphabet*; we will assume that it is the set of subsets of $\{1, \ldots, n\}$ for some natural number n;
- $I \subseteq S$ is the set of *initial states*;
- $\delta \subseteq S \times \Sigma \times S \times S$ is the *transition relation*;
- $\Omega : S \to \mathscr{N}$ is a function defining the acceptance condition. (We use $\mathscr{N}$ to denote the set of natural numbers.)

A *run* of $\mathscr{A}$ on a tree $\mathscr{B} \in \mathscr{T}_n$ is a function $r : \mathscr{T} \to S$, such that, $r(root_{\mathscr{B}}) \in I$ and for every node x of $\mathscr{T}$ (i.e. $x \in \{1,2\}^*$):

$$(r(x), \{i \colon P_{i\mathscr{B}}(x)\}, r(x1), r(x2)) \in \delta,$$

here $x1$ and $x2$ denote nodes obtained from x by appending 1 and 2, respectively, at the end of x, i.e., are the left and right successors of the node x.

To define when a run is *accepting* let us introduce a notation. For an infinite sequence of natural numbers $m_1, m_2, \ldots$ let $\mathsf{Inf}(m_1, m_2, \ldots)$ be the set of numbers appearing infinitely often in the sequence. We say that a run r is *accepting* if for every sequence of nodes $n_0, n_1, \ldots$ forming a path in $\mathscr{T}$, the smallest number in $\mathsf{Inf}(\Omega(r(n_0)), \Omega(r(n_1)), \ldots)$ is even. We say that $\mathscr{A}$ *accepts* a tree $\mathscr{B}$ if there is an accepting run of $\mathscr{A}$ on $\mathscr{B}$. The language *recognized* by $\mathscr{A}$ is the set of trees accepted by $\mathscr{A}$.

We are interested in parity automata because they capture the power of monadic second-order logic on binary trees while having a useful "regularity" property (see Lemma 6). (Monadic second-order logic is formally introduced in the next section.)

Theorem 5 (Mostowski [7]). *A subset of $\mathscr{T}_n$ is the language recognised by a Rabin automaton iff it is the language recognised by a parity automaton. Hence for every* *formula* $\alpha(X_1,\ldots,X_n)$ *of monadic second-order logic there is a parity automaton* $\mathscr{A}$ *such that for every* $\mathscr{B}\in\mathscr{T}_n$

$$\mathscr{B}\models\alpha(P_1,\ldots,P_n) \quad \textit{iff} \quad \mathscr{B}\in L(\mathscr{A}).$$

Parity automata are easier to work with than Muller or Rabin tree automata [10] because they admit *regular runs*, a notion we will define now. For a tree $\mathscr{B}\in\mathscr{T}_n$ and a node $x\in\mathscr{B}$ let $\mathscr{B}/x$ denote a subtree issued from $\mathscr{B}$. We will say that r is a *regular run on* $\mathscr{B}$ if for every two nodes x,y of $\mathscr{B}$:

if $r(x)=r(y)$ and $\mathscr{B}/x$ is isomorphic to $\mathscr{B}/y$ then $r(h(u))=r(u)$ for every node u of $\mathscr{B}/x$, where h is the isomorphism: $\mathscr{B}/x\rightarrow\mathscr{B}/y$.

Intuitively, a run is regular if it behaves identically on isomorphic subtrees provided the states assigned to the roots of these trees are the same.

Lemma 6. *For every parity automaton* $\mathscr{A}$ *and every tree* $\mathscr{B}$: *if* $\mathscr{A}$ *accepts* $\mathscr{B}$ *then there is a regular accepting run of* $\mathscr{A}$ *on* $\mathscr{B}$.

Proof. The lemma follows from the results about games with parity conditions considered in [8, 6]. It was shown there that such games have memoryless strategies. We will briefly recall this result here and show how it applies in our case.

A *parity game* is a bipartite graph $G=\langle V=V_{\mathrm{I}}\cup V_{\mathrm{II}},\ E\subseteq V\times V,\ \Omega:V\rightarrow\{1,\ldots,n\}\rangle$ with vertices labelled by numbers from $\{1,\ldots,n\}$.

A *play* from some vertex $v_1\in V_{\mathrm{I}}$ is played as follows: first player I chooses a vertex $v_2\in V_{\mathrm{II}}$ with $E(v_1,v_2)$, then player II chooses a vertex $v_3\in V_{\mathrm{I}}$ with $E(v_2,v_3)$, and so on ad infinitum unless one of the players cannot make a move. If a player cannot make a move he looses. The result of an infinite play is an infinite path $v_1,v_2,v_3,\ldots$. This path is *winning* for player I if in the sequence $\Omega(v_1),\Omega(v_2),\Omega(v_3),\ldots$ the smallest number appearing infinitely often is even. The play from a vertex of V_{II} is defined similarly but this time player II starts.

A *strategy* σ for player I is a function assigning to every sequence of vertices $\boldsymbol{v}$ ending in a vertex from V_{I} a vertex $\sigma(\boldsymbol{v})\in V_{\mathrm{II}}$, such that, $E(v,\sigma(\boldsymbol{v}))$. A strategy is *memoryless* iff $\sigma(\boldsymbol{v})=\sigma(\boldsymbol{w})$ whenever $\boldsymbol{v}$ and $\boldsymbol{w}$ end in the same vertex. A strategy is *winning* iff it guarantees a win for player I whenever he follows the strategy. Similarly, we define a strategy for player II.

Theorem 7 (Emerson and Jutla [6] and Mostowski [8]). *In every parity game one of the players has a winning strategy. If a player has a winning strategy then he has a memoryless strategy.*

Now, we will show how to use this theorem in our case. We first construct a game that is a "product" of the automaton $\mathscr{A}$ and the graph obtained from $\mathscr{B}$ by identifying isomorphic subtrees. Define the relation $\approx$ on nodes of $\mathscr{B}$ by: $m\approx n$ if the subtrees issued from m and n are isomorphic. Let $\mathscr{C}$ be a transition system obtained by quotienting $\mathscr{B}$ by the $\approx$ relation. Let $V_{\mathrm{I}} = S_{\mathscr{A}} \times S_{\mathscr{C}}$, i.e., the set of pairs consisting of a state of the automaton and a node of $\mathscr{C}$. Let $V_{\mathrm{II}} = \delta_{\mathscr{A}} \times S_{\mathscr{C}}$, i.e., the set of pairs consisting of a transition of the automaton (an element of $\delta_{\mathscr{A}}$) and a node of $\mathscr{C}$. There is an edge from a vertex $(s,[n]) \in V_{\mathrm{I}}$ to a vertex $((s,a,s_1,s_2),[n]) \in V_{\mathrm{II}}$ if $a = \{i: P_{i,\mathscr{B}}(n)\}$ (we use $[n]$ to denote the equivalence class of n with respect to the $\approx$ relation). There are edges from a vertex $((s,a,s_1,s_2),[n]) \in V_{\mathrm{II}}$ to vertices $(s_1,[n1])$ and $(s_2,[n2])$; as before $n1$ denotes the node obtained by concatenating 1 at the end of n. Observe that from vertices in V_{I} there may be many edges or there may be no edges at all. On the other hand, every vertex in V_{II} has exactly two edges going from it. Finally, we define the function Ω by letting $\Omega((s,[n])) = \Omega(s)$, i.e., we use the function Ω of the automaton $\mathscr{A}$.

Theorem 7 applies to the game just defined. From the assumption that there is an accepting run of $\mathscr{A}$ on $\mathscr{B}$ it follows that there is a winning strategy for player I from the vertex $(s_0,[n_0])$, i.e., the pair consisting of the initial state of $\mathscr{A}$ and the equivalence class of the root of $\mathscr{B}$. This strategy is to take a transition suggested by the run. Hence, by Theorem 7 there exists a memoryless strategy in the game. This memoryless strategy induces a regular run of $\mathscr{A}$ on $\mathscr{B}$. $\square$

3. Monadic second-order logic

Let U be a finite set of relational symbols. We denote by $STR(U)$ structures of type U. Any two isomorphic structures are considered as equal. Typically, U will contain a unary symbol rt and binary symbols $\mathsf{src}, \mathsf{tgt}, Q_1, \ldots, Q_m$.

We let $\mathscr{L}_2(n,m)$ be the set of MS formulas written with the relation symbols $\mathsf{rt}, \mathsf{src}, \mathsf{tgt}, Q_1, \ldots, Q_m$ (and of course $\subseteq$ and $\in$) and with free variables in $\{X_1, \ldots, X_n\}$.

In order to express properties of transition systems by monadic second-order (MS in short) formulas, we represent a transition system R of type (n,m) by the relational structure:

$$|R|_2 = \langle S_R \cup T_R, \mathsf{rt}_R, \mathsf{src}_R, \mathsf{tgt}_R, P_{1R}, \ldots, P_{nR}, Q_{1R}, \ldots, Q_{mR} \rangle,$$

where $\mathsf{rt}_R = \{root_R\}$. We say that such a transition system has the type (n,m).

We define $|R|_2 \models \alpha$, where $\alpha \in \mathscr{L}_2(n,m)$, by taking $P_{1R}, \ldots, P_{nR}$ as respective values of $X_1, \ldots, X_n$. It will be convenient to restrict to the fragment of the logic without first-order variables. First-order variables can be represented by set variables together with a formula restricting them to range over singletons. For this to work we extend the meanings of the relations $\mathsf{rt}, \mathsf{src}, \mathsf{tgt}, Q_1, \ldots, Q_n$ to hold for appropriate singleton sets. We omit the standard details (see [5]).

The properties of the behaviour $\mathsf{Un}(R)$ of a system R as above can be expressed in a similar way by formulas of $\mathscr{L}_2(n,m)$ (since $\mathsf{Un}(R)$ is a transition system of type (n,m)). However, we shall use the following simpler representation: For a transition system R of type (n,m) we let

$$|R|_1 = \langle S_R, \mathsf{rt}_R, \mathsf{suc}_{1R}, \dots, \mathsf{suc}_{mR}, P_{1R}, \dots, P_{nR} \rangle,$$

where $(x,y) \in \mathsf{suc}_{iR}$ iff there is in Q_{iR} a transition from x to y.

We let $\mathscr{L}_1(n,m)$ denote the set of MS formulas written with the symbols $\mathsf{rt}, \mathsf{suc}_1, \dots, \mathsf{suc}_m$ (in addition to $\subseteq$ and $\in$) and having their free variables in $\{X_1, \dots, X_n\}$. Again, we define $|R|_1 \models \alpha$ for $\alpha \in \mathscr{L}_1(n,m)$ by taking $P_{1R}, \dots, P_{nR}$ as values of $X_1, \dots, X_n$, respectively. By the results of Courcelle [3], the same properties of trees can be represented by formulas of $\mathscr{L}_2$ and $\mathscr{L}_1$.

Our objective is to prove the following theorem.

Theorem 8. *Let $n, m \in \mathscr{N}$, $m \geqslant 1$. For every formula $\varphi \in \mathscr{L}_1(n,m)$ one can construct a formula $\psi \in \mathscr{L}_2(n,m)$ such that, for every countably branching transition system R of type (n,m):*

$$|R|_2 \models \psi \Leftrightarrow |\mathsf{Un}(R)|_1 \models \varphi.$$

We shall need the notion of an MS-definable transduction of relational structures that we now recall from [2]. This is nothing more than the notion of first-order interpretation, modified so as to work for MS-logic.

Let U and U' be two finite ranked sets of relation symbols. Let $\mathscr{W}$ be a finite set of set variables, called here the set of *parameters*. (It is not a loss of generality to assume that all parameters are set variables.) A *(U,U')-definition scheme* is a tuple of formulas of the form

$$\Delta = (\varphi, \psi_1, \dots, \psi_k, (\theta_w)_{w \in (U')^* k}),$$

where $k > 0$, $[k]$ denotes the set $\{1, \dots, k\}$, $(U')^* k = \{(q, \boldsymbol{j}) \mid q \in U',\ \boldsymbol{j} \in [k]^{\rho(q)}, \rho(q)$ is the arity of $q\}$; $\varphi \in \mathrm{MS}(U, \mathscr{W})$; $\psi_i \in \mathrm{MS}(U, \mathscr{W} \cup \{x_1\})$ for $i = 1, \dots, k$; $\theta_w \in \mathrm{MS}(U, \mathscr{W} \cup \{x_1, \dots, x_{\rho(q)}\})$ for $w = (q, \boldsymbol{j}) \in (U')^* k$. These formulas are intended to define a structure R' in $STR(U')$ from a structure R in $STR(U)$ and will be used in the following way. The formula φ defines the domain of the corresponding transduction; namely, R' is defined only if φ is true in R. Assuming this condition fulfilled, the formulas $\psi_1, \dots, \psi_k$ define the domain of R' as the disjoint union of the sets $D_1, \dots, D_k$, where D_i is the set of elements in the domain of R that satisfy ψ_i. Finally, the relation $q_{R'}$ is defined by the formulas θ_w for $w = (q, \boldsymbol{j}) \in (U')^* k$. Here are the formal definitions.

Let $R \in STR(U)$, let μ be a $\mathscr{W}$-assignment in R. If $(R, \mu) \models \varphi$ then *Δ defines in (R, μ) a U'-structure R'* as follows:

(i) $S_{R'} = \{(d,i) \mid d \in S_R,\ i \in [k],\ (R, \mu, d) \models \psi_i\} \subseteq S_R \times [k]$,

(ii) for each q in U'
$q_{R'} = \{((d_1, i_1), \dots, (d_t, i_t)) \in S_{R'}^t \mid (S, \mu, d_1, \dots, d_t) \models \theta_{(q, \boldsymbol{j})}\}$,
where $\boldsymbol{j} = (i_1, \dots, i_t)$ and $t = \rho(q)$.

(By $(R,\mu,d_1,\ldots,d_t)\models\theta_{(q,j)}$, we mean $(R,\mu')\models\theta_{(q,j)}$, where μ' is the assignment extending μ, such that $\mu'(x_i)=d_i$ for all $i=1,\ldots,t$; a similar convention is used for $(R,\mu,d)\models\psi_i$.)

Since R' is associated in a unique way with R,μ and Δ whenever it is defined, i.e., whenever $(R,\mu)\models\varphi$, we can use the functional notation $def_\Delta(R,\mu)$ for R'.

The *transduction defined by* Δ is the relation $def_\Delta := \{(R,R') \mid R'=def_\Delta(R,\mu)$ for some $\mathscr{W}$-assignment μ in $R\}\subseteq STR(U)\times STR(U')$. A transduction $f\subseteq STR(U)\times STR(U')$ is *MS-definable* if it is equal to def_Δ for some (U,U')-definition scheme Δ. In the case when $\mathscr{W}=\emptyset$, we say that f is *MS-definable without parameters* (note that it is functional). We shall refer to the integer k by saying that def_Δ is k-copying; if $k=1$ we say that it is *non-copying* and we can write more simply Δ as $(\varphi,\psi,(\theta_q)_{q\in U'})$. In this case:

$$S_{R'}=\{d\in S_R:\ (R,\mu,d)\models\psi\}$$

and for each q in U'

$$q_{R'}=\{(d_1,\ldots,d_t)\in D^t_{R'}:\ (R,\mu,d_1,\ldots,d_t)\models\theta_q\},\ \text{where } t=\rho(q).$$

We give an example concerning automata on words: the product of a finite-state automaton $\mathscr{A}$ by a *fixed* finite-state automaton $\mathscr{F}$. A finite-state automaton is defined as a 5-tuple $\mathscr{A}=\langle S,\Sigma,I,\delta,F\rangle$ where: S is a finite set of states; Σ is a finite input alphabet (here we shall take $\Sigma=\{a,b\}$); $I\subseteq S$ is a set of initial states; δ is a transition relation which is here a subset of $S\times\Sigma\times S$ (we consider nondeterministic automata without ε-transitions); $F\subseteq S$ is the set of of final states. The language recognized by $\mathscr{A}$ is denoted by $L(\mathscr{A})$. The automaton $\mathscr{A}$ is represented by the relational structure: $|\mathscr{A}|=\langle S,I,F,trans_a,trans_b\rangle$ where $trans_a$ and $trans_b$ are binary relations and:

$trans_a(p,q)$ holds if and only if $(p,a,q)\in\delta$,

$trans_b(p,q)$ holds if and only if $(p,b,q)\in\delta$.

Let $\mathscr{F}=\langle S',\Sigma,I',\delta',F'\rangle$ be a similar automaton, and $\mathscr{A}\times\mathscr{F}=\langle S\times S',\Sigma,I\times I',\delta'',F\times F'\rangle$ be the product automaton intended to recognise the language $L(\mathscr{A})\cap L(\mathscr{F})$. We assume that S' is $\{1,\ldots,k\}$ (let us recall that $\mathscr{F}$ is fixed). We let Δ be the k-copying definition scheme $(\varphi,\psi_1,\ldots,\psi_k,(\theta_w)_{w\in(U')*k})$, where $U'=\{trans_a,trans_b,I,F\}$ and:

φ is the constant *true* (because every structure in $STR(U')$ represents an automaton which may have inaccessible states and useless transitions);

$\psi_1,\ldots,\psi_k$ are the constant *true*;

$\theta_{(trans_a,i,j)}(x_1,x_2)$ is the formula $trans_a(x_1,x_2)$ if (i,a,j) is a transition of $\mathscr{F}$, and is the constant *false* otherwise;

$\theta_{(trans_b,i,j)}$ is defined similarly;

$\theta_{(I,i)}(x_1)$ is the formula $I(x_1)$ if i is an initial state of $\mathscr{F}$, and is *false* otherwise;

$\theta_{(F,i)}(x_1)$ is defined similarly.

It is not hard to check that $|\mathscr{A}\times\mathscr{F}|=def_\Delta(|\mathscr{A}|)$. Note that the language recognised by an automaton is nonempty if and only if there is a path in its graph from some initial state to some final state. This later property is expressible in monadic second-order logic. Hence, it follows from Proposition 10 below that, for a fixed rational language K, the set of structures representing automata $\mathscr{A}$ such that $L(\mathscr{A}) \cap K$ is nonempty is definable. This construction is used systematically in Courcelle [4].

Fact 9. *The domain of an* MS-*definable transduction is* MS-*definable.*

Proof. Let Δ be a definition scheme as in the general definition with $\mathscr{W} = \{X_1, \dots, X_n\}$. We recall that $\mathscr{W}$ is the set of parameters. The image of a structure R under def_Δ is defined for the values of parameters that satisfy φ. Hence, the domain of def_Δ is $\{R\colon R \models \exists X_1, \dots, X_n . \varphi\}$. □

The following proposition says that if $R' = def_\Delta(R, \mu)$, i.e., if R' is defined in (R, μ) by Δ, then the monadic second-order properties of R' can be expressed as monadic second-order properties of (R, μ). The usefulness of MS-definable transductions is based on this proposition.

Let $\Delta = (\varphi, \psi_1, \dots, \psi_k, (\theta_w)_{w \in (U')^* k})$ be a (U, U')-definition scheme, written with a set of parameters $\mathscr{W}$. Let $\mathscr{V}$ be a set of set variables disjoint from $\mathscr{W}$. For every variable X in $\mathscr{V}$, for every $i = 1, \dots, k$, we let X_i be a new variable. We let $\mathscr{V}' := \{X_i\colon X \in \mathscr{V}, i = 1, \dots, k\}$. For every mapping $\eta : \mathscr{V}' \to \mathscr{P}(S_R)$, we let $\eta \uparrow k : \mathscr{V} \to \mathscr{P}(S_R \times [k])$ be defined by $(\eta \uparrow k)(X) = \eta(X_1) \times \{1\} \cup \cdots \cup \eta(X_k) \times \{k\}$. Note that, even if R' is well defined, the mapping $\eta \uparrow k$ is not necessarily a $\mathscr{V}$-assignment in R', because $(\eta \uparrow k)(X)$ is not necessarily a subset of the domain of R' which is a possibly proper subset of $S_R \times [k]$. With these notations we can state:

Proposition 10 (Courcelle [2]). *Let Δ be a (U, U')-definition scheme with the set of parameters $\mathscr{W}$. For every formula β in* $\mathrm{MS}(U', \mathscr{V})$ *one can construct a formula γ in* $\mathrm{MS}(U, \mathscr{V}' \cup \mathscr{W})$ *such that, for every R in $STR(U)$, for every assignment $\mu : \mathscr{W} \to R$ and for every assignment $\eta : \mathscr{V}' \to R$, we have:*

$def_\Delta(R, \mu)$ is defined (if it is, we denote it by R'), $\eta \uparrow k$ is a $\mathscr{V}$-assignment in R', and $(R', \eta \uparrow k) \models \beta$ if and only if $(R, \eta \cup \mu) \models \gamma$.

From this proposition, we get easily [2]:

Proposition 11. (1) *The inverse image of an* MS-*definable class of structures under an* MS-*definable transduction is* MS-*definable.*

(2) *The composition of two* MS-*definable transductions is* MS-*definable.*

Definition 12. Let $\mathscr{K}$ and $\mathscr{K}'$ be two classes of structures with $\mathscr{K} \subseteq STR(U)$ and $\mathscr{K}' \subseteq STR(U')$, and let f be a transduction from $\mathscr{K}$ to $\mathscr{K}'$. We say that f is *MS-compatible* if there exists an algorithm that associates with every MS-formula φ over U' an MS-formula $\widehat{\varphi}$ over U such that, for every structure $R \in \mathscr{K}$

$R \models \widehat{\varphi}$ iff $R' \models \varphi$ for some $R' \in f(R)$.

It follows from Proposition 10 that every MS-definable transduction is MS-compatible.

Our main result (Theorem 8) says that the transduction $|R|_2 \mapsto |\mathsf{Un}(R)|_1$ is MS-compatible for R ranging over countably branching transition systems of type (n,m).

We will use MS-definable transductions for constructing k-coverings of graphs. The following proposition will be used in Section 5 in the proof of Theorem 8.

Proposition 13. *Let $k,m \geqslant 1$, let $n \geqslant 0$. There exists an* MS-*definable transduction associating with every transition system R of type (n,m) the set of its k-coverings (where a system R is represented by a structure $|R|_2$).*

Proof. Let R be a transition system of type (n,m) and $h: R' \to R$ be a k-covering.

By choosing an arbitrary linear ordering of each set $h^{-1}(x)$, $x \in S_R$, we can assume that $S_{R'} \subseteq S_R \times [k]$ and $h(x,i) = x$ for every i such that $(x,i) \in S_{R'}$. We can assume that $root_{R'} = (root_R, 1)$.

For each $i \in [k]$, we let $Y_i = \{x \in S_R \colon (x,i) \in S_{R'}\}$. For $i,j \in [k]$, we let

$$Z_{i,j} = \{t \in T_R \colon h(t') = t \text{ for some } t' \in T_{R'} \text{ with source } (\mathsf{src}_R(t), i)$$
$$\text{and target } (\mathsf{tgt}_R(t), j)\}$$

Since h is a bijection of $out_{R'}(x)$ onto $out_R(h(x))$ for every $x \in S_{R'}$ it follows that for every $t \in Z_{i,j}$, there is a unique $t' \in T_{R'}$, with source $(\mathsf{src}_R(t), i)$ and target $(\mathsf{tgt}_R(t), j)$ such that $h(t') = t$. We shall identify t' with the triple (t,i,j).

Hence,

$$S_{R'} = \bigcup\{Y_i \times \{i\} \colon 1 \leqslant i \leqslant k\}, \tag{1}$$

$$T_{R'} = \bigcup\{Z_{i,j} \times \{(i,j)\} \colon i,j \in [k]\}. \tag{2}$$

This gives a description of $|R'|_2$ as the output of a definable transduction taking as input $|R|_2$ and the parameters $Y_1, \ldots, Y_k, Z_{1,1}, \ldots, Z_{k,k}$.

Specifically, we have:

$$\mathsf{rt}_{R'} = \{(x,1)\} \text{ where } x \text{ is the unique state in } \mathsf{rt}_R, \tag{3}$$

$$\mathsf{src}_{R'} = \{((t,i,j),(x,i)) \colon i,j \in [k],\ t \in Z_{i,j},\ (t,x) \in \mathsf{src}_R\}, \tag{4}$$

$$\mathsf{tgt}_{R'} = \{((t,i,j),(x,j)) \colon i,j \in [k],\ t \in Z_{i,j},\ (t,x) \in \mathsf{tgt}_R\}, \tag{5}$$

$$P_{iR'} = \{(x,j) \colon x \in P_{iR} \cap Y_j,\ j \in [k]\},\ i = 1, \ldots, n, \tag{6}$$

$$Q_{iR'} = \{(t,j,j') \colon t \in Q_{iR} \cap Z_{j,j'}, j,j' \in [k]\},\ i = 1, \ldots, m. \tag{7}$$

In this construction, we have assumed that the parameters $Y_1, \ldots, Y_k, Z_{1,1}, \ldots, Z_{k,k}$ are defined from a k-covering R' of R. In order to ensure that the constructed transduction *only defines* k-coverings of the input transduction systems we must find a formula $\varphi(Y_1, \ldots, Y_k, Z_{1,1}, \ldots, Z_{k,k})$ which verifies that the structure defined by (1)–(7) is actually of the form $|R'|_2$ for some k-covering R' of R.

We consider the following conditions:

$$S_R = \bigcup\{Y_i \colon 1 \leqslant i \leqslant k\}, \tag{8}$$

$$T_R = \bigcup\{Z_{i,j} \colon i,j \in [k]\}. \tag{9}$$

For every $i \in [k]$, every $x \in Y_i$, every transition $t \in \mathsf{out}_R(x)$ there is one and only one $j \in [k]$ such that $t \in Z_{i,j}$, (10)

Every state of R' is accessible by a path from $root_{R'}$. (11)

Conditions (8)–(11) can be written as an MS-formula φ in parameters $Y_1, \dots, Y_k, Z_{1,1}, \dots, Z_{k,k}$ to be evaluated in $|R|_2$. Let us review them: (8)–(9) state that the mapping $h : S_{R'} \cup T_{R'} \to S_R \cup T_R$ defined by

$$h((x,i)) = x \quad \text{if } (x,i) \in S_{R'}$$

and

$$h((t,(i,j))) = t \quad \text{if } (t,(i,j)) \in T_{R'}$$

is surjective. From its definition it is a homomorphism. Condition 11 states that it is a covering. Condition 11 states that R' is indeed a transition system.

Hence, $\varphi(Y_1, \dots, Y_k, Z_{1,1}, \dots, Z_{k,k})$ is the desired formula which completes the proof. □

4. A regularisation lemma

If R is a transition system of type (n,m) and $Y \subseteq S_R$, we denote by $R * Y$ the system of type $(n+1,m)$ consisting of R augmented with Y as the $(n+1)$th set of states.

The following lemma is a crucial step for the main theorem.

Lemma 14. *Let $n \geqslant 0$ and $\alpha \in \mathscr{L}_1(n+1,2)$. One can find an integer k such that, for every (possibly infinite) complete deterministic transition system R of type $(n,2)$, if $|\mathsf{Un}(R)|_1 \models \exists X_{n+1} . \alpha$, then there exists a k-covering R' of R and a subset Y of $S_{R'}$ such that $|\mathsf{Un}(R' * Y)|_1 \models \alpha$.*

Proof. The idea of the proof is the following. If $T = |\mathsf{Un}(R)|_1 \models \exists X_{n+1} . \alpha$ then an appropriate meaning of X_{n+1} can be represented as a run of an automaton on T. Then one can also find a suitable meaning for X_{n+1} that can be represented as a regular run on T. For every subtree, a regular run on this subtree is determined by the state assigned to the root of the subtree and the isomorphism class of the subtree. As there are finitely many, say k, states, the corresponding regular run can be defined in the unfolding of a k-covering R' of R. Hence, the set $X_{n+1} \subseteq \mathsf{Un}(R)$ satisfying α can be replaced by the set resulting from the unfolding of a subset of R'.

Let $R \in \mathscr{T}_n$ be as in the assumption of the lemma. Denote by $h_R : \mathsf{Un}(R) \to R$ the canonical homomorphism sending a path to its endpoint.

By Theorem 5 there exists a parity automaton $\mathscr{A} = \langle S, \mathscr{P}(\{1,\dots,n+1\}), I, \delta, \Omega \rangle$ recognising the set of trees: $L(\mathscr{A}) = \{U \in \mathscr{T}_{n+1}\colon\ |U|_1 \models \alpha\}$. Define the automaton $\mathscr{A}' = \langle S', \mathscr{P}(\{1,\dots,n\}), I', \delta', \Omega' \rangle$, where:

$$S' = \{(s,i)\colon\ s \in S,\ i = 0,1\},$$
$$I' = \{(s,i)\colon\ s \in I,\ i = 0,1\},$$
$$((s,0), a, (s_1,i_1), (s_2,i_2)) \in \delta' \quad \text{if } (s,a,s_1,s_2) \in \delta \text{ and } i_1, i_2 \in \{0,1\},$$
$$((s,1), a, (s_1,i_1), (s_2,i_2)) \in \delta' \quad \text{if } (s,a \cup \{n+1\},s_1,s_2) \in \delta \text{ and } i_1, i_2 \in \{0,1\},$$
$$\Omega'(s,i) = \Omega(s) \quad \text{for } i \in \{0,1\}.$$

It is easy to see that $L(\mathscr{A}') = \{U \in \mathscr{T}_n\colon\ |U|_1 \models \exists X_{n+1} \,.\, \alpha\}$. Hence, $\mathsf{Un}(R) \in L(\mathscr{A}')$. So, by Lemma 6, $\mathscr{A}'$ has a regular run $r : \mathsf{Un}(R) \to S'$ on R.

We are going to define the system R' required in the lemma. Intuitively, R' is a folding of $\mathsf{Un}(R)$ respecting the run r, i.e., if two nodes of $\mathsf{Un}(R)$ are assigned different states then they are not identified in the folding.

Let $R' = \langle S_{R'}, T_{R'}, \mathsf{rt}_{R'}, \mathsf{src}_{R'}, \mathsf{tgt}_{R'}, P_{1R'}, \dots, P_{nR'}, Q_{1R'}, Q_{2R'} \rangle$, where

- $S_{R'}$ is the set of elements $(n,(s,i)) \in S_R \times S'$ such that there exists $x \in \mathsf{Un}(R)$ with $h_R(x) = n$ and $r(x) = (s,i)$.
- We have a transition from $(n,(s,i))$ to $(n',(s',i'))$ if there exists $x \in \mathsf{Un}(R)$ such that $h_R(x) = n$, $r(x) = (s,i)$, and, in $\mathsf{Un}(R)$, there is a transition from x to some x' with $h_R(x') = n'$ and $r(x') = (s',i')$. The type of the transition is the same as the type of the transition from x to x'.
- $\mathsf{rt}_{R'}$ is $(\mathsf{rt}_R, r(\mathsf{rt}_R))$.
- $P_{jR'}(n,(s,i))$ iff $P_{jR}(n)$

Claim 15. *R' is a complete deterministic transition system of type $(n,2)$.*

Proof. It is easy to see that all states of R' are accessible. We are left to show that from every state there is exactly one transition of each type.

Let $(n,(s,i)) \in S_{R'}$. We will show that it has exactly one transition of type 1. By definition of $S_{R'}$ there is $x \in \mathsf{Un}(R)$ such that $h_R(x) = n$ and $r(x) = (s,i)$. Because R is complete deterministic there exists exactly one node $x' \in \mathsf{Un}(R)$ to which there is a type 1 transition from x. We have that $(h_R(x'), r(x')) \in S_{R'}$ and there is a transition of type 1 from $(n,(s,i))$ to $(h_R(x), r(x'))$.

To see that there is only one transition of type 1 from $(n,(s,i))$ consider some node $y \in \mathsf{Un}(R)$ such that $h_R(y) = h_R(x) = n$. In particular, subtrees $\mathsf{Un}(R)/x$ and $\mathsf{Un}(R)/y$ are isomorphic. Let y' be the target of the type 1 transition from y. We have $h_R(y') = h_R(x')$. By the definition of the regular run we have $r(y') = r(x')$. $\square$

Claim 16. *R' is a k-covering of R, where k is the number of states of the automaton $\mathscr{A}'$.*

Proof. Define the mapping $h' : R' \to R$ by $h'(n,(s,i)) = n$. It is easy to check that it extends to a homomorphism of transition systems. By Fact 4 it is a covering. By the definition of h', the inverse image of a state of R can have at most k elements. $\square$

To finish the proof of Lemma 14 we must find a set Y such that $|\mathsf{Un}(R' * Y)|_1 \models \alpha$. Define $Y = \{(n,(s,i)) \in S_{R'}: i = 1\}$. We define a run r' of $\mathscr{A}$ on $\mathsf{Un}(R' * Y)$ by: $r'(u) = s$ if u is a path ending in a node $(n,(s,i)) \in S_{R'}$, for some n and i. It is easy to check that this is an accepting run. Hence, $\mathsf{Un}(R' * Y) \in L(\mathscr{A})$ and we get $|\mathsf{Un}(R' * Y)|_1 \models \alpha$. $\square$

We consider Lemma 14 as a regularisation lemma because it says that if $|\mathsf{Un}(R)|_1$ contains a set Z satisfying α then it contains another one having a special "regular" form, defined from the unfolding of a k-covering of R.

5. Edge contractions and the proof of the main result

Our next aim is to extend Lemma 14 to transition systems that are not deterministic. We first consider systems of type $(n,1)$. If R is a transition system of type $(n,1)$, then each node of the tree $\mathsf{Un}(R)$ has some unordered set of successors. In case R is countably branching, $\mathsf{Un}(R)$ can be represented in the binary tree in way that we now describe.

We define a transformation that makes a tree $T \in \mathscr{T}_{n+1}$ (which is a system of type $(n+1,2)$) into a tree $c(T)$ of type $(n,1)$.

Let $T \in \mathscr{T}_{n+1}$ be defined by an $(n+1)$-tuple of subsets of $\{1,2\}^*$, namely by $(P_{1T},\dots,P_{n+1T})$. We let $c(T)$ be the tree such that:

- $S_{c(T)} = (\{1,2\}^* \setminus P_{1T}) \cup \{\varepsilon\}$;
- $x \to y$ in $c(T)$ iff there is in T a path of the form $x \to z_1 \to z_2 \to \cdots \to z_p \to y$ with $p \geqslant 0$ and $z_1, z_2, \dots, z_p \in P_{1T}$ ($x \to y$ is a shorthand for "there is a transition from x to y");
- $P_{i-1c(T)} = P_{iT} \cap S_{c(T)}$ for $i = 2,\dots,n+1$.

Our next aim is to define a similar operation on transition systems so that

$$\mathsf{Un}(c(R)) = c(\mathsf{Un}(R)).$$

A *special transition system* is a system R of type $(n+1,2)$, for some n, such that

(1) R is complete deterministic;
(2) $root_R \notin P_{1R}$;
(3) $P_{1R} \cap (P_{2R} \cup \cdots \cup P_{n+1R}) = \emptyset$.

We now define a transformation c that transforms any special transition system R of type $(n+1,2)$ into one of type $(n,1)$. We let $c(R)$ be such that

- $S_{c(R)} = S_R \setminus P_{1R}$;
- $P_{ic(R)} = P_{i+1R} \cap S_{c(R)}$ for $i = 2,\dots,n$;
- $root_{c(R)} = root_R$;

- $x \to y$ is a transition of $c(R)$ iff $x, y \in S_{c(T)}$ and we have a path in R of the form $x \to z_1 \to z_2 \to \cdots \to z_p \to y$ with $x, y \notin P_{1R}$, $z_1, z_2, \ldots, z_p \in P_{1R}$, $p \geqslant 0$.

Lemma 17. *If R is special then we have* $c(\mathsf{Un}(R)) = \mathsf{Un}(c(R))$.

Proof. Easy verification. □

Lemma 18. *For every countably branching transition system R of type $(n, 1)$ one can construct a special transition system, $Bin(R)$ of type $(n+1, 2)$ such that $c(Bin(R)) = R$.*

Proof. We let R' be the transition system of type $(n+1, 2)$ defined as follows:
(1) we add a new "sink" state $\perp$ and two transitions $\perp \to \perp$: one of type 1 and one of type 2,
(2) for each state $s \in S_R$ we do the following:
 (a) if $\mathsf{out}_R(s) = \emptyset$, we add two transitions $s \to \perp$ of types 1 and 2;
 (b) if $\mathsf{out}_R(s) = \{t\}$, we add a transition $s \to \perp$ of type 2 (note that the transition t is necessarily of type 1);
 (c) if $\mathsf{out}_R(s)$ consists of two transitions, we make one of them a type 2 transition, the other transition continues to be of type 1.
 (d) if $\mathsf{out}_R(s)$ consists of at least three transitions $t_1, \ldots, t_k$ then we add new states $u_2, \ldots, u_{k-1}$. For $i = 2, \ldots, k-1$ we change the source of t_i to u_i. We also change the source of t_k to u_{k-2}. We add new transitions $s \to u_2$, $u_i \to u_{i+1}$ for $i = 2, \ldots, k-1$. All added transitions as well as the transition t_k become transitions of type 2. Transitions $t_1, \ldots, t_{k-1}$ continue to be of type 1;
 (e) if $\mathsf{out}_R(s)$ is infinite but countable then we enumerate the transitions $t_1, t_2, \ldots$ and proceed similarly to the previous case.
(3) We let $P_{1Bin(R)}$ consist of all "new states" (the state $\perp$ and the states introduced in the steps 2c and 2d above) and we let $P_{i+1Bin(R)} = P_{iR}$ for every $i = 1, \ldots, n$. □

Lemma 19. *If R is a special transition system and K is a k-covering of $Bin(R)$ then K is also special and $c(K)$ is a k-covering of R.*

Proof. We let $h : K \to Bin(R)$ be a k-covering. We first check that K is a special system. Condition 1 of the definition of a special system (saying that K is complete deterministic) holds because every covering of a complete deterministic system is complete deterministic. Conditions 2 and 3 hold easily.

It remains to prove that $c(K)$ is a k-covering of R. Let us consider $h : S_{c(K)} \to S_R$. It is the desired covering. This follows from the observations establishing that K is a special system. □

Proposition 20. *Let $n \geqslant 0$ and $\alpha \in \mathscr{L}_1(n+1, 1)$. One can find an integer k, such that for every countably branching transition system R of type $(n, 1)$ if $|\mathsf{Un}(R)|_1 \models \exists X_{n+1} . \alpha$*

then there exists a k-covering R' of R and a subset Y of $S_{R'}$ such that $|\mathsf{Un}(R' * Y)|_1 \models \alpha$.

Proof. We first construct a formula $\beta \in \mathscr{L}_1(n+2,2)$ such that for every tree T in $\mathscr{T}_{n+2}$ we have

$$|T|_1 \models \beta \text{ iff } P_{1T} \cap (P_{2T} \cup \cdots \cup P_{n+1T}) = \emptyset \text{ and } |c(T)|_1 \models \alpha.$$

This is possible because the mapping from $|T|_1$ to $|c(T)|_1$ is a definable transduction of structures. We let k be the integer associated with β by Lemma 14.

Let R be a transition system of type $(n,1)$ such that $|\mathsf{Un}(R)|_1 \models \exists X_{n+1} . \alpha$. For some set $Z \subseteq S_{\mathsf{Un}(R)}$ we have thus

$$|\mathsf{Un}(R) * Z|_1 \models \alpha.$$

Because $Bin(R)$ is a special transition system, from Lemmas 17 and 18 we have: $\mathsf{Un}(R) = c(\mathsf{Un}(Bin(R)))$. It also follows that $Z \subseteq S_{\mathsf{Un}(Bin(R))}$ and $Z \cap P_{1\mathsf{Un}(Bin(R))} = \emptyset$. Hence,

$$|\mathsf{Un}(Bin(R)) * Z|_1 \models \beta.$$

By Lemma 14 we have some k-covering K of $Bin(R)$ and some $Y \subseteq S_K$ such that

$$|\mathsf{Un}(K * Y)|_1 \models \beta.$$

It holds in particular that $P_{1K} \cap Y = \emptyset$. By Lemma 19, $c(K)$ is a k-covering of R and $Y \subseteq S_{c(K)}$.

Hence, $c(K)$ is the desired system R' since

$$|c(\mathsf{Un}(K * Y))|_1 \models \alpha$$

and

$$c(\mathsf{Un}(K * Y)) = \mathsf{Un}(c(K * Y)) = \mathsf{Un}(c(K) * Y). \qquad \square$$

Proof of Theorem 8. Let us first consider the case of the systems of type $(n,1)$. We want to show that for every formula $\varphi \in \mathscr{L}_1(n,1)$, one can construct a formula $\widehat{\varphi} \in \mathscr{L}_2(n,1)$ such that, for every transition system R of type $(n,1)$:

$$|R|_2 \models \widehat{\varphi} \quad \text{iff} \quad |\mathsf{Un}(R)|_1 \models \varphi.$$

The proof proceeds by induction on the structure of φ. We assume that φ is a closed formula. This is not a restriction as two formulas are equivalent iff the closed formulas obtained by substituting unary relational symbols for free variables are equivalent.

If φ is a closed atomic formula then $\widehat{\varphi} = \varphi$. The cases for conjunction and negation are obvious.

Assume $\varphi = \exists X . \alpha(X)$. By Proposition 20 there is an integer k such that for every transition system of type $(n,1)$:

> $|\mathsf{Un}(R)|_1 \models \exists X . \alpha(X)$ iff there exists a k-covering R' of R and a subset Y of $S_{R'}$, such that, $|\mathsf{Un}(R' * Y)|_1 \models \alpha[P_{n+1}/X]$.

By induction assumption we have a formula $\widehat{\alpha}[P_{n+1}/X]$, such that, for every transition system K of type $(n+1,1)$:

$$|K|_2 \models \widehat{\alpha}[P_{n+1}/X] \quad \text{iff} \quad |\mathsf{Un}(K)|_1 \models \alpha[P_{n+1}/X].$$

It remains to show that the property:

there exist a k-covering R' of R such that $R' \models \exists X . \widehat{\alpha}(X)$

is MS-definable.

By Proposition 13 we know that the transduction associating with R the set of its k coverings is MS-definable. (This transduction has parameters $Y_1,\dots,Y_k,Z_{1,1},\dots,Z_{k,k}$; each admissible choice of parameters gives us a k-covering). Proposition 10 gives us the desired formula $\widehat{\varphi}$.

We now prove the theorem for systems of the general type (n,m) with $m \geqslant 1$.

We define a transformation α making a transition system R of type (n,m) into a transition system $\alpha(R)$ of type $(n+m,1)$ such that the transduction $|R|_2 \mapsto |\alpha(R)|_2$ is MS-definable, and a transformation β from transition systems of type $(n+m,1)$ to transition systems of type (n,m) such that the transduction $|R|_1 \mapsto |\beta(R)|_1$ is MS-definable and

$$\mathsf{Un}(R) = \beta(\mathsf{Un}(\alpha(R))) \tag{12}$$

for every transition system R of type (n,m). Clearly, such transformations reduce the general case of Theorem 8 to the case of systems of type $(n,1)$ which we have just proved.

Definition of α. Let R be a transition system of type (n,m) with $m \geqslant 2$.

The idea of the construction of $\alpha(R)$ is to replace a state x of R by m states $(x,1),\dots,(x,m)$ in R' and to replace a transition $y \to x$ of type i by m transitions from $(y,1),\dots,(y,m)$ to (x,i) all of type 1. (If there is no transition of type i from y to x then we need not put in $\alpha(R)$ the state (x,i).)

Here is the formal definition of $\alpha(R)$. Suppose

$$R = \langle S_R, T_R, \mathsf{src}_R, \mathsf{tgt}_R, root_R, P_{1R},\dots,P_{nR}, Q_{1R},\dots,Q_{mR}\rangle.$$

Recall that $[m]$ denotes the set $\{1,\dots,m\}$. First we define the system R' which is the 5-tuple

$$\langle S_{R'}, T_{R'}, \mathsf{src}_{R'}, \mathsf{tgt}_{R'}, root_{R'}, P_{1R'},\dots,P_{nR'}, P'_{1R'},\dots,P'_{mR'}\rangle,$$

where

$$S_{R'} = S_R \times [m],$$

$$T_{R'} = T_R \times [m],$$

$$(s,i) = \mathsf{src}_{R'}(t,j) \text{ iff } s = \mathsf{src}_R(t) \text{ and } i = j,$$

$$(s,i) = \mathsf{tgt}_{R'}(t,j) \text{ iff } s = \mathsf{tgt}_R(t) \text{ and } t \in Q_{iR},$$

$$root_{R'} = (root_R, 1),$$

$$P_{iR'}(s,j) \Leftrightarrow s \in S_R \quad \text{and} \quad P_{iR}(s) \text{ for } i = 1,\ldots,n,$$

$$P'_{iR'}(s,j) \Leftrightarrow s \in S_R \quad \text{and} \quad i = j \text{ for } i = 1,\ldots,n.$$

Then R' is "almost" a transition system of type $(n+m,1)$: "almost" because some states may be unreachable. One obtains $\alpha(R)$ by restricting R' to the reachable states and transitions. It is clear from this definition that $|\alpha(R)|_2$ is definable from $|R|_2$ by a definable transduction. We omit the details.

Definition of β. Let R' be a transition system of the form

$$\langle S_{R'}, T_{R'}, \mathsf{src}_{R'}, \mathsf{tgt}_{R'}, root_{R'}, P_{1R'},\ldots,P_{nR'}, P'_{1R'},\ldots,P'_{mR'}\rangle,$$

where $P_{1R'},\ldots,P_{nR'},P'_{1R'},\ldots,P'_{mR'}$ are properties of states. Then we define a transition system $\beta(R)$ iff $(P'_{1R'},\ldots,P'_{mR'})$ forms a partition of $S_{R'}$. If this is the case we let $\beta(R') = R$ where $S_R = S_{R'}$, $T_R = T_{R'}$, $\mathsf{src}_R = \mathsf{src}_{R'}$, $\mathsf{tgt}_R = \mathsf{tgt}_{R'}$, $root_R = root_{R'}$, $P_{iR} = P_{iR'}$ for $i = 1,\ldots,n$ and $Q_{iR} = \{t \in T_{R'} \mid \mathsf{tgt}_R(t) \in P'_{iR'}\}$ for $i = 1,\ldots,n$. It is clear that $|\beta(R)|_1$ is definable from $|R|_1$ by a definable transduction.

It is also clear from the construction that $\beta(\mathsf{Un}(\alpha(R)))$ is well defined for every transition system of type (n,m) and that

$$\beta(\mathsf{Un}(\alpha(R))) = \mathsf{Un}(R).$$

This completes the proof of Theorem 8. $\square$

6. The Shelah–Stupp–Muchnik construction

We recall a construction and a result from Shelah and Stupp [12, 14] extended by Muchnik. The result by Muchnik is stated without a proof in Semenov [11] and a new proof is sketched in [15]. We establish that it yields an improvement of our main result.

We let U be a finite set of relational symbols where each symbol r has a finite arity $\rho(r)$. We recall that we denote by $STR(U)$ the class of all U-structures, i.e., of tuples of the form $M = \langle D_M, (r_M)_{r\in U}\rangle$ where D_M is a nonempty set (the domain of M) and $r_M \subseteq D_M^{\rho(r)}$ for every $r \in U$.

We let *son* and *cl* be two relation symbols, binary and unary, respectively, which are not in U. We let $U^+ = U \cup \{son, cl\}$. We let D_M^* and $(D_M)^+$ stand for the set of finite sequences over D_M and the set of finite nonempty sequences, respectively.

With $M \in STR(U)$ we associate the U^+-structure:

$$M^+ = \langle (D_M)^+, (r_{M^+})_{r\in U}, son_{M^+}, cl_{M^+}\rangle,$$

where

$$r_{M^+} = \{(wd_1,\dots,wd_{\rho(r)}):\ w\in D_M^*, (d_1,\dots,d_{\rho(r)})\in r_M\},$$

$$son_{M^+} = \{(w,wd):\ w\in D_M^*, d\in D_M\},$$

$$cl_{M^+} = \{wdd:\ w\in D_M^*, d\in D_M\}.$$

Intuitively, M^+ is a "tree built over M"; *son* is the corresponding successor relation and *cl* is the set of *clones*, i.e., of elements of M^+ that are "like their fathers" (if $son(x,y)$ we also say that x is the father of y; it is unique).

Theorem 21 (Muchnik [11] and Walukiewicz [15]). *The mapping $M\mapsto M^+$ is* MS-*compatible. In words, for every formula φ in* MS(U^+) *one can construct a formula ψ in* MS(U), *such that, for every $M\in STR(U)$:*

$$M^+\models\varphi \quad \textit{iff} \quad M\models\psi.$$

It is stated (without a proof) in Shelah [12] and Thomas [14] that, if a structure M has a decidable monadic theory then so has the structure M^+ with respect to the language $MS(U^+ - \{cl\})$. This statement weakens Theorem 21 in two respects: the "clone" relation is omitted and the statement only concerns decidability of theories and not translations of formulas. From Theorem 21, one gets the following improvement of Theorem 8:

Theorem 22. *For every $n,m\in\mathcal{N}$ with $m\geqslant 1$, the transduction*:

$$|R|_1\mapsto|\mathsf{Un}(R)|_1$$

is MS-*compatible, where R ranges over simple transition systems of type (n,m).*

A transition system is *simple* if no two distinct transitions have the same source, target and type.

Since some properties of simple graphs are MS-expressible with edge set quantifications but not without them, the result of Theorem 22 is an improvement of Theorem 8. (The property that a simple directed graph has a directed spanning tree of out-degree no bigger than some constant is an example of such a property; the existence of a Hamiltonian circuit is another example; see [3, p. 125].)

Theorem 22 follows from Theorem 21 because the unfolding of R is MS-definable in $|R|_1^+$ (see Proposition 24). Before showing this we will introduce a useful definition.

If Q is a binary relation on D_M, then we let Q^{tr} and Q^{rot} (respectively, the *translation* and the *rotation* of Q) be defined as follows:

$$Q^{\mathrm{tr}} = \{(wd,wd'):\ w\in D_M^*, (d,d')\in Q\},$$

$$Q^{\mathrm{rot}} = \{(wd,wdd'):\ w\in D_M^*, (d,d')\in Q\}.$$

(Note that Q^{tr} is defined from Q like r_{M^+} is from r_M.)

Claim 23. *If Q is* MS-*definable in M then so are Q^{tr} and Q^{rot} in M^+.*

Proof. To prove this for Q^{tr} it is enough to observe that:

$$Q^{\mathrm{tr}}(x,y) \quad \text{iff} \quad \exists z(son_{M^+}(z,x) \wedge son_{M^+}(z,y) \wedge \varphi'(z,x,y)),$$

where $\varphi'(z,x,y)$ is the relativization to the set of sons of z (sons in the sense of M^+) of the formula $\varphi(x,y)$ defining Q in M. For Q^{rot}, we have

$$Q^{\mathrm{rot}}(x,y) \quad \text{iff} \quad \exists z(son_{M^+}(x,z) \wedge son_{M^+}(x,y) \wedge cl_{M^+}(z) \wedge Q^{\mathrm{tr}}(z,y))$$

which proves the claim. □

Theorem 22 is an immediate consequence of

Proposition 24. *For every $n,m \in \mathcal{N}$, $m \geqslant 1$, the transduction $(|R|_1)^+ \mapsto |\mathsf{Un}(R)|_1$, where R is a simple transition system of type (n,m), is* MS-*definable.*

Proof. Assume $M = |R|_1 = \langle S_R, root_R, \mathsf{suc}_{1R}, \dots, \mathsf{suc}_{mR}, P_{1R}, \dots, P_{nR}\rangle$. We define a binary relation W on $(S_R)^+$ as follows:

$$W = W_1 \cup \dots \cup W_m, \quad \text{where for each } i,\ W_i = (\mathsf{suc}_{iR})^{\mathrm{rot}}.$$

We let $N \subseteq (S_R)^+$ be defined as follows:

$$y \in N \ \text{iff there exists } x \in (S_R)^+ \text{ such that } root_{M^+}(x) \wedge (\forall z\ \neg son_{M^+}(z,x)) \wedge (x,y) \in W^*.$$

Note that the first two conjuncts of the above condition define x uniquely since $root_R$ consists of a unique state (x is r where $root_R = \{r\}$). We use W^* to denote the transitive closure of W. Hence, N is the set of elements of S_R^+ that are accessible from this x by a directed path with edges in W.

Claim 25. $|\mathsf{Un}(R)|_1 = \langle N, W_1', \dots, W_m', P_1', \dots, P_n'\rangle$, *where $W_i' = W_i \cap (N \times N)$ and $P_j' = P_{jR} \cap N$ for every $i = 1, \dots, m$ and $j = 1, \dots, n$.*

Proof. We define a bijection h of $Paths(R)$ (the set of nodes of $\mathsf{Un}(R)$) onto N. Let p be a path in $Paths(R)$, say $p = (t_1, \dots, t_k)$, $t_1, \dots, t_k \in T_R$. We let $h(p) = (s_0, \dots, s_k) \in (S_R)^+$ where s_0 is the initial state of R and for each $i = 1, \dots, k$, s_{i-1} is the source of t_i and s_i is the target of t_i.

Since R is simple, h is one-to-one. If $s_i \to s_{i+1}$ is a transition of type j then $W_j((s_0, \dots, s_i), (s_0, \dots, s_{i+1}))$ holds. Hence, h maps $Paths(R)$ onto N.

It is then easy to verify that every $y \in N$ is the image by h of some path p (the proof is by induction on the least integer k such that $(x,y) \in W^k$ where x is the element of $(S_R)^+$ used in the definition of N). Finally, h is an isomorphism. We omit the details. □

It is clear from the definition that N is a definable subset of $(S_R)^+$ (by an MS-formula on M^+) and that the relations $W_1',\ldots,W_m',P_1',\ldots,P_n'$ are MS-definable. Hence $|\mathsf{Un}(R)|_1$ can be obtained from $(|R|_1)^+$ by a definable transduction. □

The proof of this proposition is due to W. Thomas (private communication).

Example. Let $U=\emptyset$, $M=\langle\{0,1\}\rangle$. Consider $M^+=\langle\{0,1\}^+, son_{M^+}, cl_{M^+}\rangle$. One can define the complete binary tree $B=\langle N,\mathsf{suc}_1,\mathsf{suc}_2\rangle$ in M^+ as follows: one lets x be an arbitrary element of M^+ having no father; one lets N be the set of elements y of $\{0,1\}^+$ such that $(x,y)\in(son_{M^+})^*$, one lets then

$$\mathsf{suc}_1(u,v) \;\Leftrightarrow\; son_{M^+}(u,v)\wedge cl_{M^+}(v),$$

$$\mathsf{suc}_2(u,v) \;\Leftrightarrow\; son_{M^+}(u,v)\wedge \neg cl_{M^+}(v).$$

There are only two choices for x and the corresponding structures are both isomorphic to B.

It follows that the monadic theory of B reduces to that of M^+. The later is decidable since the monadic theory of M is decidable (as M is finite).

7. Graph coverings

We have seen that the mapping from a transition system to its universal covering is MS-compatible (where a system R is represented by $|R|_2$ or $|R|_1$). We ask the same question for graphs. We consider actually two different notions of covering for which the answers are completely different.

7.1. Bidirectional coverings

We consider directed graphs G, defined by means of sets: V_G (vertices), E_G (edges) and the source and target mappings respectively $\mathsf{src}_G : E_G \to V_G$, $\mathsf{tgt}_G : E_G \to V_G$. For convenience we restrict here to connected graphs. The extension of the results to disconnected graphs is easy.

For $x\in V_G$ we denote by $in_G(x)$ the set of edges of G with target x; we denote by $out_G(x)$ the set of edges with source x.

Definition 26 (*Bidirectional covering*). Let G,G' be connected graphs. A homomorphism $h:G'\to G$ is a bidirectional covering iff it is surjective and for every $x\in V_{G'}$, h is a bijection of $in_{G'}(x)$ onto $in_G(h(x))$ and of $out_{G'}(x)$ onto $out_G(h(x))$.

For short, we shall write *b-covering* for bidirectional covering. Unlike coverings, *b*-coverings treat incoming edges exactly as outgoing edges.

Definition 27 (*Signed edges, walks*). A signed edge of G is a pair (e,η), where $e \in E_G$ and $\eta \in \{+,-\}$. We define src_G and tgt_G for signed edges as follows:

$$\begin{aligned} &\mathsf{src}_G(e,+)=\mathsf{src}_G(e), \quad \mathsf{src}_G(e,-)=\mathsf{tgt}_G(e),\\ &\mathsf{tgt}_G(e,+)=\mathsf{tgt}_G(e), \quad \mathsf{tgt}_G(e,-)=\mathsf{src}_G(e). \end{aligned}$$

We let suc_G be the binary relation on signed edges:

$$\mathsf{suc}_G((e,\eta),(e',\eta')) \quad \text{iff} \quad \mathsf{tgt}_G(e,\eta)=\mathsf{src}_G(e',\eta') \wedge (e=e' \Rightarrow \eta=\eta').$$

A walk in G is a finite sequence of signed edges $w=((e_1,\eta_1),\dots,(e_k,\eta_k))$ such that $\mathsf{suc}((e_i,\eta_i),(e_{i+1},\eta_{i+1}))$ holds for all $i=1,\dots,k-1$. We say that w is a walk from $\mathsf{src}_G(e_1,\eta_1)$ to $\mathsf{tgt}_G(e_k,\eta_k)$.

Intuitively, a walk is a path in G traversing edges in either direction. A signed edge (e_i,η_i) represents a traversal of e_i in the standard direction if $\eta_i=+$ and in the reverse direction if $\eta_i=-$. A walk is not allowed to take the same edge twice consecutively in opposite directions.

Fact 28. *If $h:G'\to G$ is a homomorphism and $w=((e_1,\eta_1),\dots,(e_k,\eta_k))$ is a walk from x to y in G' then the image of w defined as the sequence $((h(e_1),\eta_1),\dots,(h(e_k),\eta_k))$ is a walk in G from $h(x)$ to $h(y)$.*

Fact 29. *If $h:G'\to G$ is a b-covering, $x'\in V_{G'}$, $h(x')=x$ and w is a walk from x to y in G, then there is a unique walk w' in G' from x' to some y' such that $h(w')=w$. Vertex y' satisfies $h(y')=y$.*

We now construct a b-covering of a graph G in terms of walks. Let G be connected, let $s\in V_G$. Denote by $W(s)$ the set of all the walks from s to arbitrary vertices. We put in $W(s)$ the empty walk ε and assume that it goes from s to s.

We let H be the graph such that

$$V_H=W(s), \quad E_H=\text{a disjoint copy of } W(s)-\{\varepsilon\}.$$

If $w.(e,\eta)\in E_H$ for some $e\in E_G$ and $\eta\in\{+,-\}$, we let $\mathsf{src}_H(w.(e,\eta))=w$ and $\mathsf{tgt}_H(w.(e,\eta))=w.(e,\eta)$ if $\eta=+$ and $\mathsf{src}_H(w.(e,\eta))=w.(e,\eta)$ and $\mathsf{tgt}_H(w.(e,\eta))=w$ otherwise.

We now let $h:H\to G$ be the homomorphism such that

$$\begin{aligned} &h(\varepsilon)=s,\\ &h(w)=x \quad \text{such that } w \text{ goes from } s \text{ to } x,\ w\in V_H-\{\varepsilon\},\\ &h(w)=e \quad \text{where } w\in E_H \text{ is of the form } w'.(e,\eta). \end{aligned}$$

Fact 30. *$h:H\to G$ is a b-covering.*

Proposition 31. *For every b-covering $k : K \to G$ there is a surjective homomorphism $m : H \to K$ such that $k \circ m = h$ which is a b-covering. For every two such homomorphisms $m, m' : H \to K$, there is an automorphism i of H, such that, $m' = m \circ i$.*

Proof. Easy consequence of Facts 28 and 29. □

We shall call H the *universal* b-covering of G and denote it by $UBC(G)$.

Theorem 32. *The transduction mapping $|G|_2$ to $|UBC(G)|_1$ for connected graphs G is* MS*-compatible.*

Proof. We first recall that the structure $|G|_2$ defining G is $\langle V_G \cup E_G, \text{r-src}_{|G|_2}, \text{r-tgt}_{|G|_2}\rangle$ where:

$$\text{r-src}_{|G|_2} = \{(e, \text{src}_G(e)) \colon e \in E_G\},$$
$$\text{r-tgt}_{|G|_2} = \{(e, \text{tgt}_G(e)) \colon e \in E_G\}.$$

(In order to avoid confusions between functions and relations we use $\text{r-src}_{|G|_2}$ to denote the binary relation associated with the unary function $\text{src}_G : E_G \to V_G$, and similarly for $\text{r-tgt}_{|G|_2}$.)

In order to handle signed edges by logical formulas, we shall consider the structure

$$|G|_3 = \langle V_G \cup E_G \times \{+,-\}, \text{r-src}_{|G|_3}, \text{r-tgt}_{|G|_3}, \text{dir}^+_{|G|_3}, \text{dir}^-_{|G|_3}\rangle,$$

where

$$\text{dir}^+_{|G|_3} = \{(e,+) \colon e \in E_G\}, \qquad \text{dir}^-_{|G|_3} = \{(e,-) \colon e \in E_G\},$$
$$\text{r-src}_{|G|_3} = \{(f, \text{src}_G(f)) \colon f \in E_G \times \{+,-\}\},$$
$$\text{r-tgt}_{|G|_3} = \{(f, \text{tgt}_G(f)) \colon f \in E_G \times \{+,-\}\}.$$

It is easy to construct an MS-transduction transforming $|G|_2$ into $|G|_3$.

Next, we show that $UBC(G)$ is MS-definable in $|G|_3^+$, hence is definable from $|G|_3^+$ by a MS-transduction. First, observe that suc_G is MS-definable in $|G|_3$, hence $(\text{suc}_G)^{\text{rot}}$ is MS-definable in $|G|_3^+$ by Claim 23.

The elements of the domain of $|G|_3^+$ are nonempty sequences of elements of $|G|_3$. We shall select a subset N of them corresponding to the walks from some vertex s to all the vertices of G. Such a set can be characterised by the following conditions:

(1) N is closed under $(\text{suc}_G)^{\text{rot}}$ (i.e., if $x \in N$ and $(\text{suc}_G)^{\text{rot}}(x,y)$ holds then $y \in N$);

(2) if $x \in N$ and $y \in D_{|G|_3^+}$ and $son_{|G|_3^+}(y,x)$ holds then $y \in N$ and $(\text{suc}_G)^{\text{rot}}(y,x)$ holds;

(3) there is a unique element $s_N \in D_{|G|_3^+}$, such that, $\text{r-src}_{|G|_3}(x, s_N)$ holds for every $x \in N$ for which there is no y with $son_{|G|_3^+}(y,x)$.

A set $N \cup \{s_N\}$ will be the set of nodes of $UBC(G)$ we are constructing. Different choices of N correspond to different choices of the root vertex s_N in the condition (3) and will yield the same covering up to isomorphism.

We define the edge relation $Q \subseteq N \times N$ as follows:

(1) if $x, y \in N$ and $son_{|G|_3^+}(x,y)$, we put an edge $(x,y) \in Q$ if $\mathsf{dir}^+_{|G|_3^+}(y)$ and put an edge $(y,x) \in Q$ if $\mathsf{dir}^-_{|G|_3^+}(y)$;

(2) if $y \in N$ and $son_{|G|_3^+}(x,y)$ for no $x \in N$, then we put an edge $(s_N, y) \in Q$ if $\mathsf{dir}^+_{|G|_3^+}(y)$ and an edge $(y, s_n) \in Q$ if $\mathsf{dir}^-_{|G|_3^+}(y)$.

It is easy to check that $\langle N, Q \rangle$ is isomorphic to $UBC(G)$.

We obtain thus that the transduction $|G|_2 \mapsto |UBC(G)|_1$ is MS-compatible because it can be written as the following composition:

$$|G|_2 \mapsto |G|_3 \mapsto |G|_3^+ \mapsto |UBC(G)|_1,$$

where the first and the third transformations are MS-definable, whereas the second is MS-compatible by Theorem 21. This completes the proof. □

Open problem: Can one change $|G|_2$ to $|G|_1$ in the statement of Theorem 32 for simple graphs G? (It is false for nonsimple graphs as multiple edges are identified in $|G|_1$.)

7.2. Distance-1-coverings

For every graph G and every $x \in V_G$, we denote by $B_G(x)$ the subgraph of G induced by $\{x\} \cup V$, where V is the set of vertices adjacent to x.

A *distance-1-covering* (a d1-covering for short) is a b-covering $h : G' \to G$ such that for every $y \in V_{G'}$, h is an isomorphism: $B_{G'}(y) \to B_G(h(y))$.

Example. G' is d1-covering of G where G and G' are presented in Fig. 1 and h maps x' and x'' to x for $x \in \{a,b,c,d\}$.

The graph G_2 is a b-covering of the graph G_1 presented in Fig. 1. But G_2 is not a d1-covering. Clearly, G_1 is isomorphic to all its d1-coverings since $G_1 = B_{G_1}(x)$ for some x.

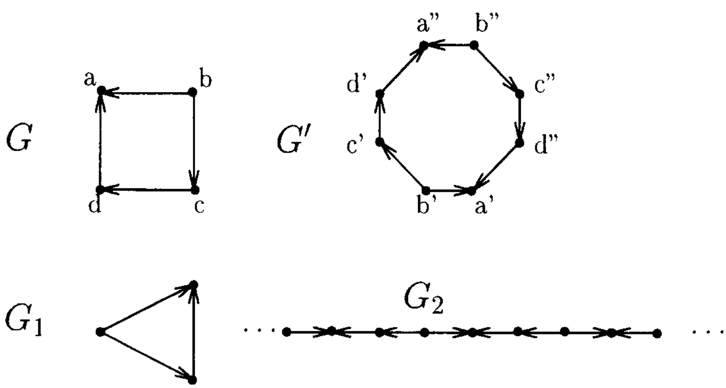

Fig. 1. Example of d1-covering and b-covering.

We shall now construct a universal d1-covering of a graph G as a quotient of its universal b-covering $UBC(G)$.

We let $H = UBC(G)$ (see Fact 30 above) and $h: H \to G$ be the canonical b-covering. We let $E \subseteq (V_H \times V_H) \cup (E_H \times E_H)$ be the equivalence relation defined as

$$\{(u,v)\colon\ h(u) = h(v) \text{ and } u, v \text{ belong to a connected component of } h^{-1}(B_G(x)) \text{ for some } x\}.$$

We let H' be the quotient graph $H|E$, we let $k: H \to H'$ be the canonical surjective homomorphism such that $h = h' \circ k$. It is not hard to see that h' is a d1-covering of G and that every d1-covering $m: G' \to G$ factors into $h' \circ m'$, where $m': G' \to H'$ is a surjective homomorphism and furthermore a d1-covering. We shall call H' the *universal-d1-covering* of G and denote it by $UDC(G)$.

Proposition 33. *The mapping $|G|_2 \mapsto |UDC(G)|_1$ is not* MS-*compatible even if G is restricted to finite connected graphs of degree at most* 6.

Proof. We construct a finite connected graph G of degree 6, such that, $UDC(G)$ is the infinite grid (augmented with diagonals on each square). Since the monadic theory of $UDC(G)$ is undecidable (even if MS-formulas do not use quantification over sets of edges), and since the monadic theory of $|G|_2$ is decidable (since G is finite) it follows that MS-formulas expressing properties of $UDC(H)$ cannot be translated into equivalent MS-formulas on $|H|_2$ in a uniform way, for all finite connected graphs H, even of bounded degree at most 6.

The infinite grid with diagonals is the graph H such that

$$V_H = Int \times Int,$$
$$E_H = \{((x,y),(x',y'))\colon\ x, y, x', y' \in Int,\ x \leqslant x' \leqslant x+1, y \leqslant y' \leqslant y+1, (x,y) \neq (x',y')\}.$$

Int denotes the set of integers. Fig. 2 shows a portion of H.

For $x, x' \in Int$ we let $x \sim x'$ iff $x - x'$ is a multiple of 4. For $(x,y),(x',y') \in V_H$ we let $(x,y) \sim (x',y')$ iff $x \sim x'$ and $y \sim y'$. For $e, e' \in E_H$ linking, respectively, z_1 to z_2 and z_1' to z_2', we let $e \sim e'$ iff $z_1 \sim z_1'$ and $z_2 \sim z_2'$.

We let G be the quotient graph $H| \sim$. Then G is the graph partially shown on Fig. 3. We let h be the canonical surjective homomorphism $h: H \to G$.

Furthermore, h is a d1-covering of G. In order to prove that $H = UDC(G)$ it is enough to prove that if $k: K \to H$ is a d1-covering then k is an isomorphism.

So let $k: K \to H$ be a d1-covering of H. If k is not an isomorphism, there exist $x, y \in V_K$ such that $x \neq y$ and $k(x) = k(y)$. Let us select such a pair where x and y are at minimal distance, say n. Hence, in K there exists a walk from x to y of the form $w = ((e_1, \eta_1), \ldots, (e_n, \eta_n))$. Its image under k is a walk $k(w) = ((k(e_1), \eta_1), \ldots, (k(e_n), \eta_n))$ from $z = k(x)$ to itself.

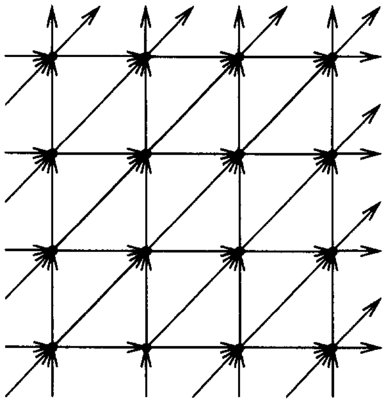

Fig. 2. A portion of H.

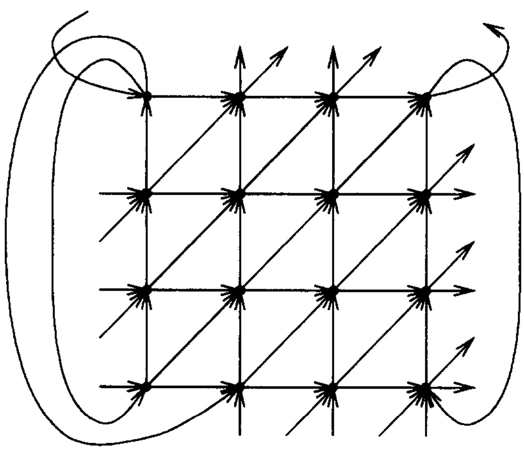

Fig. 3. Graph G.

The intermediate vertices on this walk are pairwise distinct and distinct with z because otherwise, n would not be the distance between x and y or one could find a pair $x', y' \in V_K$ such that $k(x')=k(y')$, $x' \neq y'$ and the distance between x' and y' is less than n.

Consider now $k(w)$. It defines a cycle on the planar graph H (where edges can be traversed in either direction). This cycle is simple (it does not cross itself) and has a certain *area* namely, the number of triangles forming its interior part. We shall prove that we can replace w by a walk w' from x to y of the same length and such that the area of $k(w')$ is strictly smaller than that of $k(w)$. This will give us a contradiction and prove that k is an isomorphism.

Let u be the unique vertex of $k(w)$ having a maximal first component among those that have a maximal second component. We first assume that $u \neq k(x) = k(y)$. Let $u = (u_0, u_1)$. Let v and v' be the two neighbours of u on the circular walk $k(w)$. Up to exchanges of v and v' we have the following possible cases (by the maximality conditions on u_0 and u_1):

Case 1: $v=(u_0-1,u_1)$, $v'=(u_0-1,u_1-1)$.

Case 2: $v=(u_0,u_1-1)$, $v'=(u_0-1,u_1-1)$.

Case 3: $v=(u_0-1,u_1)$, $v'=(u_0,u_1-1)$.

However, case 1 cannot happen because w is minimal. Let us check this. Let $\bar{u}$ be the vertex of w with $k(\bar{u})=u$. Since k is an isomorphism between $B_K(\bar{u})$ and $B_H(u)$ since $v,v'\in B_H(u)$ and are adjacent, so are $\bar{v}=k^{-1}(v)$ and $\bar{v}'=k^{-1}(v')$ in $B_K(\bar{u})$. It follows that w can be replaced by a shorter walk, which connects directly $\bar{v}$ and $\bar{v}'$ and skips $\bar{u}$. This contradicts the hypothesis that w has a minimal length.

Case 2 cannot happen for a similar reason.

In case 3 we cannot connect directly $\bar{v}$ and $\bar{v}'$ but we can link them via the unique vertex $k^{-1}(u_0-1,u_1-1)$ in $B_K(\bar{u})$ (note that v,v' and (u_0-1,u_1-1) belong all to $B_H(u)$). The resulting walk w' is such that $k(w')$ has a smaller area than $k(w)$ (smaller by 2).

If $u=k(x)=k(y)$ we use a similar argument by replacing u by the unique vertex of $k(w)$ having a minimal first component among those that have a minimal second component. The argument goes through with $+1$ instead of -1 everywhere. □

8. Conclusions

We have shown the main conjecture of [4] (see Theorem 8) saying that the unfolding operation is MS-compatible provided graphs (or transition systems) are represented in a way making it possible to quantify over sets of edges (or of transitions). It follows in particular that the unfolding of a graph or a transition system having a decidable MS-theory, still has a decidable MS-theory.

A stronger form of this result follows from Theorem 21.

We have also considered "bidirectional unfolding" of graphs. Although it is very close to unfolding, we could extend the main theorem only for the logic with the power to quantify over edges of a graph. Whether one can strengthen the theorem and get the result for the logic with quantification limited to vertices is an open question.

These unfoldings have been defined as instances of the very general topological notion of covering (for appropriate notions of neighbourhood). The two notions correspond to neighbourhoods of increasing strengths. For the next step (distance 1-coverings), we loose the MS-compatibility we have for the unfolding. In particular, the transformation of a graph into its universal-d1-covering does not preserve decidability of the MS-theory.

Acknowledgements

The authors would like to thank Wolfgang Thomas and Suzanne Zeitman for many significant comments yielding simplifications in the proofs.

References

[1] B. Bollobas, Extremal Graph Theory, Academic Press, New York, 1978.
[2] B. Courcelle, Monadic second-order graph transductions: a survey, Theoret. Comput. Sci. 126 (1994) 53–75.
[3] B. Courcelle, The monadic second-order logic on graphs VI: on several representations of graphs by relational structures, Disc. Appl. Math. 54 (1994) 117–149. (Erratum in Disc. App. Math. 63 (1995) 199–200).
[4] B. Courcelle, The monadic second-order logic on graphs IX: machines and behaviours, Theoret. Comput. Sci. 151 (1995) 125–162.
[5] B. Courcelle, The expression of graph properties and graph transformations in monadic second-order logic, in: G. Rozenberg (Ed.), Hand-book of Graph Transformations: Foundations, vol. 1, World Scientific, Singapore, 1997, pp. 313–400.
[6] E.A. Emerson, C.S. Jutla, Tree automata, mu-calculus and determinacy, Proc. FOCS 91, 1991, pp. 368–377.
[7] A.W. Mostowski, Regular expressions for infinite trees and a standard form of automata, in: A. Skowron (Ed.), 5th Symp on Computation Theory, Lecture Notes in Computer Science, vol. 208, 1984, pp. 157–168.
[8] A.W. Mostowski, Games with forbidden positions, Technical Report 78, University of Gdansk, 1991.
[9] D. Niwiński, Fixed points vs. infinite generation, in LICS '88, 1988, pp. 402–409.
[10] M. Rabin, Decidability of second-order theories and automata on infinite trees, Trans. Amer. Math. Soc. 141 (1969) 1–35.
[11] A. Semenov, Decidability of monadic theories, in MFCS '84, Lecture Notes in Computer Science, vol. 176, Springer, Berlin, 1984, pp. 162–175.
[12] S. Shelah, The monadic second-order theory of order, Ann. Math. 102 (1975) 379–419.
[13] J. Stupp, The lattice-model is recursive in the original model, Institute of Mathematics, The Hebrew University, Jerusalem, January 1975.
[14] W. Thomas, Language, automata and logic, in: Handbook of Formal Language Theory, Vol. 3, Springer, Berlin, Heidelberg, 1997, pp. 389–455.
[15] I. Walukiewicz, Monadic second order logic on tree-like structures, STACS '96, Lecture Notes in Computer Science, vol. 1046, 1996, pp. 401–414.