Constructing Infinite Graphs With a Decidable MSO-Theory*

Wolfgang Thomas

RWTH Aachen, Lehrstuhl für Informatik VII
D-52056, Aachen, Germany
E-Mail: thomas@informatik.rwth-aachen.de

Abstract. This introductory paper reports on recent progress in the search for classes of infinite graphs where interesting model-checking problems are decidable. We consider properties expressible in monadic second-order logic (MSO-logic), a formalism which encompasses standard temporal logics and the modal μ-calculus. We discuss a class of infinite graphs proposed by D. Caucal (in MFCS 2002) which can be generated from the infinite binary tree by applying the two processes of MSO-interpretation and of unfolding. The main purpose of the paper is to give a feeling for the rich landscape of infinite structures in this class and to point to some questions which deserve further study.

1 Introduction

A fundamental decidability result which appears in hundreds of applications in theoretical computer science is Rabin's Tree Theorem [23]. The theorem says that the monadic second-order theory (MSO-theory) of the infinite binary tree is decidable. The system of monadic second-order logic arises from first-order logic by adjunction of variables for sets (of tree nodes) and quantifiers ranging over sets. In this language one can express many interesting properties, among them reachability conditions (existence of finite paths between elements) and recurrence conditions (existence of infinite paths with infinitely many points of a given property).

Already in Rabin's paper [23] the main theorem is used to infer a great number of further decidability results. The technique for the transfer of decidability is the method of interpretation: It is based on the idea of describing a structure $\mathcal{A}$, using MSO-formulas, within the structure T_2 of the binary tree. The decidability of the MSO-theory of $\mathcal{A}$ can then be deduced from the fact that the MSO-theory of T_2 is decidable. Rabin considered mainly structures of interest to mathematical logic. For example, he showed that the monadic second-order theory of the rational number ordering $(\mathbb{Q}, <)$ is decidable. In theoretical computer science, the interest shifted to models like transition systems (for example, Kripke structures) and their unfoldings in the form of labelled trees. Also the

* to appear in Proc. MFCS 2003, Springer-Verlag. For the present version, misprints were eliminated and pointers to references corrected (see footnotes).

terminology has changed a little: Rather than speaking of a structure with a decidable MSO-theory one says that the model-checking problem for this structure is decidable with respect to MSO-properties. Thus the search for infinite structures of this kind is tied to one of the fundamental questions in verification, namely to determine the range of structures where the model-checking problem (in our case with respect to MSO-logic) can be solved by automatic procedures.

In this research, the first key result is the Muller-Schupp Theorem [22], stating that the transition graph of a pushdown automaton has a decidable MSO-theory. In [4], Caucal showed that the same holds for the more extended class of prefix-recognizable graphs. In both cases, the proof works by MSO-interpretations in the binary tree T_2. Proof sketches are provided in Section 2 below.

Nearly 20 years ago, in MFCS 1984, A. Semenov [24] presented a decidability result of Muchnik which opened a new track for extending Rabin's Tree Theorem. Muchnik's Theorem says that for a structure whose MSO-theory is decidable also its "tree iteration" has a decidable MSO-theory. This provides another powerful method for the transfer of decidability results. When referring to graphs as structures, a variant of tree iteration is of central importance: the unfolding of a graph as a tree. A short discussion is given in Section 3.

Again in MFCS, one year ago, D. Caucal proposed in [5] to use both transfer techniques (of MSO-interpretation and of unfolding) together, starting with the finite trees and graphs. (Equivalently one can start with the infinite binary tree.) It turns out that by applying MSO-interpretations and unfoldings *in alternation*, a very rich hierarchy of models can be generated, each of them having a decidable MSO-theory. The main purpose of this paper is to provide (in Section 4 below) an intuitive introduction to this Caucal hierarchy. We explain that it provides a comprehensive framework for decidability results on MSO-theories.

In this paper, we pursue a purely model-theoretic view. One should mention that at least two other views are also possible but not taken up here in any depth: First, the tree structures which arise as unfoldings in the hierarchy have been studied already decades ago in the investigation of higher-order recursion schemes (cf. [12]); recent results in the field are due to Knapik, Niwiński, and Urzyczyn [17, 18]. In connection with the evaluation of these recursion schemes, the computational model of "iterated pushdown automaton" was introduced. The (global) transition graphs of iterated pushdown automata coincide with the graphs of the Caucal hierarchy (see [18, 6, 10][1]). Thus, the graphs of the Caucal hierarchy constitute also an interesting chapter of "infinite automata theory" ([25]), where infinite graphs are viewed and used as acceptors of non-regular languages.

[1] In [18], the equivalence is shown for the unfoldings of the transition graphs, in [5] higher-order pushdown transition graphs are shown to belong to the Caucal Hierarchy, and in [10] the converse (and thus the coincidence result) is established.

2 Interpretations

2.1 General Framework

We consider relational structures $\mathcal{A} = (A, R_1^{\mathcal{A}}, \ldots, R_k^{\mathcal{A}})$, where A is at most countable. The $R_i^{\mathcal{A}}$ are relations of possibly different arities, say $R_i^{\mathcal{A}}$ of arity n_i. The corresponding signature is given by the relation symbols $R_1, \ldots, R_k$. The first-order language over this signature is built up from variables $x, y, \ldots$, atomic formulas $x = y$ and $R_i(x_1, \ldots, x_{n_i})$, where $x, y, x_1, \ldots$ are first-order variables, using the standard propositional connectives $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$ and the quantifiers $\exists, \forall$. The corresponding monadic second-order language (MSO-language) is obtained by adjoining variables $X, Y, \ldots$ for sets of elements (of the universe of a structure) and atomic formulas $X(y)$, meaning that the element y is in the set X.

We use the standard notations; e.g. $\mathcal{A} \models \varphi[a]$ indicates that the structure $\mathcal{A}$ satisfies the formula $\varphi(x)$ with the element a as interpretation of x. Given a formula $\varphi(x_1, \ldots, x_n)$, the relation defined by it in $\mathcal{A}$ is

$$\varphi^{\mathcal{A}} = \{(a_1, \ldots, a_n) \in A^n \mid \mathcal{A} \models \varphi[a_1, \ldots, a_n]\}$$

The structures considered in this paper are edge- and vertex-labelled graphs of the form $G = (V, (E_i)_{i \in I}, (P_j)_{j \in J})$; here V is the set of vertices, I the alphabet of edge labels, $E_i \subseteq V \times V$ is the set of i-labelled edges, and $P_j \subseteq V$ the set of vertices labelled j. We set $E = \bigcup_{i \in I} E_i$.

The *binary tree* is the structure $T_2 = (\{0,1\}^*, S_0, S_1)$ where $S_i = \{(w, wi) \mid w \in \{0,1\}^*\}$. Analogously $T_n = (\{0, \ldots, n-1\}^*, S_0^n, \ldots, S_{n-1}^n)$ is the n-ary infinite tree.

Theorem 1 (Tree Theorem, [23]). *The MSO-theory of T_2 is decidable.*

Let us illustrate the idea of MSO-interpretation by showing that the result holds also for the structures T_n for $n > 2$. As typical example consider $T_3 = (\{0,1,2\}^*, S_0^3, S_1^3, S_2^3)$. We obtain a copy of T_3 in T_2 by considering only the T_2-vertices in the set $T = (10 + 110 + 1110)^*$. A word in this set has the form $1^{i_1}0 \ldots 1^{i_m}0$ with $i_1, \ldots, i_m \in \{1, 2, 3\}$; and we take it as a representation of the element $(i_1 - 1) \ldots (i_m - 1)$ of T_3.

The following MSO-formula $\varphi(x)$ (written in abbreviated suggestive form) defines the set T in T_2:

$$\forall Y[Y(x) \wedge \forall y((Y(y10) \vee Y(y110) \vee Y(y1110)) \rightarrow Y(y)) \ \rightarrow Y(\epsilon)]$$

It says that x is in the closure of ϵ under 10-, 110-, and 1110-successors. The relation $\{(w, w10) | w \in \{0,1\}^*\}$ is defined by the following formula:

$$\psi_0(x, y) := \exists z(S_1(x, z) \wedge S_0(z, y))$$

With the analogous formulas ψ_1, ψ_2 for the other successor relations, we see that the structure with universe φ^{T_2} and the relations $\psi_i^{T_2}$ restricted to φ^{T_2} is isomorphic to T_3.

In general, an MSO-interpretation of a structure $\mathcal{A}$ in a structure $\mathcal{B}$ is given by a "domain formula" $\varphi(x)$ and, for each relation $R^{\mathcal{A}}$ of $\mathcal{A}$, say of arity m, an MSO-formula $\psi(x_1, \ldots, x_m)$ such that $\mathcal{A}$ with the relations $R^{\mathcal{A}}$ is isomorphic to the structure with universe $\varphi^{\mathcal{B}}$ and the relations $\psi^{\mathcal{B}}$ restricted to $\varphi^{\mathcal{B}}$.

Then for an MSO-sentence χ (in the signature of $\mathcal{A}$) one can construct a sentence χ' (in the signature of $\mathcal{B}$) such that $\mathcal{A} \models \chi$ iff $\mathcal{B} \models \chi'$. In order to obtain χ' from χ, one has to replace every atomic formula $R(x_1, \ldots, x_m)$ by the corresponding formula $\psi(x_1, \ldots, x_m)$ and to relativize all quantifications to $\varphi(x)$ (for details see e.g. [13]). As a consequence, we note the following:

Proposition 1. *If $\mathcal{A}$ is MSO-interpretable in $\mathcal{B}$ and the MSO-theory of $\mathcal{B}$ is decidable, then so is the MSO-theory of $\mathcal{A}$.*

In the literature a more general type of interpretation is also used, called MSO-transduction (see [8]), where the universe A is represented in a k-fold copy of B rather than in B itself. For the results treated below it suffices to use the simple case mentioned above.

2.2 Pushdown Graphs and Prefix Recognizable Graphs

A graph $G = (V, (E_a)_{a \in A})$ is called *pushdown graph* (over the label alphabet A) if it is the transition graph of the reachable global states of an ϵ-free pushdown automaton. Here a pushdown automaton is of the form $\mathcal{P} = (Q, A, \Gamma, q_0, Z_0, \Delta)$, where Q is the finite set of control states, A the input alphabet, Γ the stack alphabet, q_0 the initial control state, $Z_0 \in \Gamma$ the initial stack symbol, and $\Delta \subseteq Q \times A \times \Gamma \times \Gamma^* \times Q$ the transition relation. A global state (configuration) of the automaton is given by a control state and a stack content, i.e., by a word from $Q\Gamma^*$. The graph $G = (V, (E_a)_{a \in A})$ is now specified as follows:

- V is the set of configurations from $Q\Gamma^*$ which are reachable (via finitely many applications of transitions of Δ) from the initial global state $q_0 Z_0$.
- E_a is the set of all pairs $(p\gamma w, qvw)$ from V^2 for which there is a transition (p, a, γ, v, q) in Δ.

A more general class of graphs, which includes the case of vertices of infinite degree, has been introduced by Caucal [4]. These graphs are introduced in terms of prefix-rewriting systems in which "control states" (as they occur in pushdown automata) are no longer used and where a word on the top of the stack (rather than a single letter) may be rewritten. Thus, a rewriting step can be specified by a triple (u_1, a, u_2), describing a transition from a word $u_1 w$ via letter a to the word $u_2 w$. The feature of infinite degree is introduced by allowing generalized rewriting rules of the form $U_1 \to_a U_2$ with regular sets U_1, U_2 of words. Such a rule leads to the (in general infinite) set of rewrite triples (u_1, a, u_2) with $u_1 \in U_1$ and $u_2 \in U_2$. A graph $G = (V, (E_a)_{a \in A})$ is called *prefix-recognizable* if for some finite system $\mathcal{S}$ of such generalized prefix rewriting rules $U_1 \to_a U_2$ over an alphabet Γ, we have

- $V \subseteq \Gamma^*$ is a regular set,

– E_a consists of the pairs (u_1w, u_2w) where $u_1 \in U_1$, $u_2 \in U_2$ for some rule $U_1 \rightarrow_a U_2$ from $\mathcal{S}$, and $w \in \Gamma^*$.

Theorem 2 (Muller-Schupp [22], Caucal [4]). *The MSO-theory of a pushdown graph is decidable; so is the MSO-theory of a prefix-recognizable graph.*

First we present the proof for pushdown graphs. Let $G = (V, (E_a)_{a \in A})$ be generated by the pushdown automaton $\mathcal{P} = (Q, A, \Gamma, q_0, Z_0, \Delta)$. Each configuration is a word over the alphabet $Q \cup \Gamma$. Taking $m = |Q| + |\Gamma|$ we can represent a configuration by a node of the tree T_m. For technical convenience we write the configurations in reverse order, i.e. as words in $\Gamma^+ Q$. We give an MSO-interpretation of G in T_m. The formula $\psi_a(x, y)$ which defines E_a in T_m has to say the following:

> "there is a stack content w such that $x = (p\gamma w)^R$ and $y = (qvw)^R$ for a rule (p, a, γ, v, q) of Δ."

This is easily formalized (even with a first-order formula), using the successor relations in T_m to capture the prolongation of w by γ, p, q and by the letters of v. Now it is easy to write down also the desired domain formula $\varphi(x)$ which defines the configurations reachable from q_0Z_0. We refer to $(q_0Z_0)^R$ as definable element of the tree T_m and to the union E of the relations E_a, defined by $\bigvee_{a \in A} \psi_a(x, y)$. The formula $\varphi(x)$ says that

> "each set X which contains $(q_0Z_0)^R$ and is closed under taking E-successors also contains x."

For prefix-recognizable graphs, a slight generalization of the previous proof is needed. Let G be a prefix-recognizable graph with a regular set $V \subseteq \Gamma^*$ of vertices. We describe an MSO-interpretation of G in the tree T_m where m is the size of Γ. We start with a formula $\psi(x, y)$ which defines the edge relation induced by a single rule $U_1 \rightarrow_a U_2$ with regular U_1, U_2. The formula expresses for x, y that there is a word (= tree node) w such that $x = u_1w$, $y = u_2w$ with $u_1 \in U_1$, $u_2 \in U_2$. If $\mathcal{A}_1, \mathcal{A}_2$ are finite automata recognizing U_1, U_2 respectively, this can be phrased as follows:

> "there is a node w such that $\mathcal{A}_1$ accepts the path segment from x to w and $\mathcal{A}_2$ the path segment from y to w."

Acceptance of a path segment is expressed by requiring a corresponding automaton run. Its existence can be coded by a tuple of subsets over the considered path segment (for an automaton with 2^k states a k-tuple of sets suffices). The disjunction of such formulas taken for all a-rules gives the desired formula defining the edge relation E_a. The domain formula $\varphi(x)$ is provided in the same way, now referring to the path segment from node x back to the root.

Using the interpretation of T_m in T_2, the decidability claims follow from Rabin's Tree Theorem. It is interesting to note that the prefix-recognizable graphs in fact coincide with the graphs which are MSO-interpretable in T_2 ([2]).

3 Unfoldings

Let $G = (V, (E_i)_{i \in I}, (P_j)_{j \in J})$ be a graph and v_0 a designated vertex of V. The *unfolding* of G from v_0 is a structure of the form $t(G, v_0) = (V', (E'_i)_{i \in I}, (P'_j)_{j \in J})$. Its domain V' is the set of all paths from v_0; here a path from v_0 is a sequence $v_0 i_1 v_1 \ldots i_k v_k$ where for $h \leq k$ we have $(v_{h-1}, v_h) \in E_{i_h}$. A pair (p, q) of paths is in E'_i iff q is an extension of p by an edge from E_i, and we have $p \in P'_j$ iff the last element of p is in P_j.

As an example consider the singleton graph G_0 with vertex v_0 and two edge relations E_0, E_1, both of which contain the edge (v_0, v_0). The unfolding of G_0 is (isomorphic to) the binary tree T_2. This example illustrates the power of the unfolding operation: Starting from the trivial singleton graph (which of course has a decidable MSO-theory), we obtain the binary tree T_2 where decidability of the MSO-theory is a deep result.

The unfolding operation takes sequences of edges (as elements of the unfolded structure). A related construction, called *tree iteration*, refers to sequences of elements instead. It has the advantage that it covers arbitrary relational structures without extra conventions. To spare notation we define it only over graphs, as considered above.

The tree iteration of a graph $G = (V, (E_i)_{i \in I}, (P_j)_{j \in J})$ is the structure $G^* = (V^*, S, C, (E^*_i)_{i \in I}, (P^*_j)_{j \in J})$ where $S = \{(w, wv) \mid w \in V^*, v \in V\}$ ("successor"), $C = \{(wv, wvv) \mid w \in V^*, v \in V\}$ ("clone relation"), $E^*_i = \{(wu, wv) \mid w \in V^*, (u, v) \in E_i\}$, and $P^*_j = \{wv \mid w \in V^*, v \in P_j\}$.

From the singleton graph mentioned above one obtains by tree iteration a copy of the natural number ordering rather than of the binary tree. However, the structure T_2 can be generated by tree iteration from the two element structure $(\{0, 1\}, P_0, P_1)$ using the two predicates $P_0 = \{0\}$ and $P_1 = \{1\}$. The unfolding $t(G, v_0)$ can be obtained by a monadic transduction from G^*, more precisely by an MSO-interpretation in a twofold copy of G.

Both operations preserve the decidability of the MSO-theory. Again we state this only for graphs:

Theorem 3 (Muchnik, Walukiewicz, Courcelle (cf. [24]), [26], [11])). *If a graph has a decidable MSO-theory, then its unfolding from a definable vertex and its tree iteration also have decidable MSO-theories.*

Extending earlier work of Shelah and Stupp, the theorem was shown for tree iterations by A. Muchnik (see [24]). A full proof is given by Walukiewicz in [26]; for a very readable account we recommend [1]. For the unfolding operation see the papers [9, 11] by Courcelle and Walukiewicz.

As a small application of the theorem we show a result (of which we do not know a reference) on structures $(\mathbb{N}, \mathrm{Succ}, P)$, the successor structure of the natural numbers with an extra unary predicate P. Consider the binary tree T_2 expanded by the predicate $P' = \{w \in \{0, 1\}^* \mid \ |w| \in P\}$, the "level predicate" for P. Now the MSO-theory of $(\mathbb{N}, \mathrm{Succ}, P)$ is decidable iff the MSO-theory of $(\mathbb{N}, \mathrm{Succ}_0, \mathrm{Succ}_1, P)$ is decidable where $\mathrm{Succ}_0 = \mathrm{Succ}_1 = \mathrm{Succ}$. The unfolding of the latter structure is the binary tree expanded by the level predicate for P. Hence we obtain:

Proposition 2. *If the MSO-theory of* $(\mathbb{N}, \mathrm{Succ}, P)$ *is decidable, then so is the MSO-theory of the binary tree expanded by the level predicate for* P.

4 Caucal's Hierarchy

In [5], Caucal introduced the following hierarchy $(\mathcal{G}_n)$ of graphs, together with a hierarchy $(\mathcal{T}_n)$ of trees:

- $\mathcal{T}_0$ = the class of finite trees
- $\mathcal{G}_n$ = the class of graphs which are MSO-interpretable in a tree of $\mathcal{T}_n$
- $\mathcal{T}_{n+1}$ = the class of unfoldings of graphs in G_n

By the results of the preceding sections (and the fact that a finite structure has a decidable MSO-theory), each structure in the Caucal hierarchy has a decidable MSO-theory. By a hierarchy result of Damm [12] on higher-order recursion schemes, the hierarchy is strictly increasing.

In Caucal's paper [5], a different formalism of interpretation (via "inverse rational substitutions") is used instead of MSO-interpretations. We work with the latter to keep the presentation more uniform; the equivalence between the two approaches has been established by Carayol and Wöhrle [10].

Let us take a look at some structures which occur in this hierarchy. It is clear that $\mathcal{G}_0$ is the class of finite graphs, while $\mathcal{T}_1$ contains the so-called regular trees (alternatively defined as the infinite trees which have only finitely many non-isomorphic subtrees). Figure 1 (upper half) shows a finite graph and its unfolding as a regular tree:

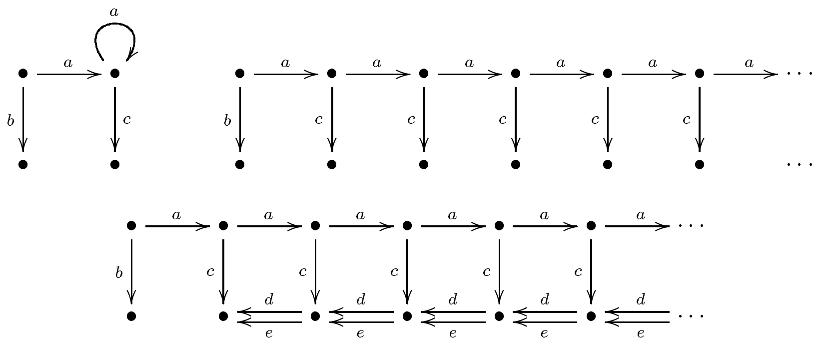

Fig. 1. A graph, its unfolding, and a pushdown graph

By an MSO-interpretation we can obtain the pushdown graph of Figure 1 in the class $\mathcal{G}_1$; the domain formula and the formulas defining E_a, E_b, E_c are trivial, while

$$\psi_d(x, y) = \psi_e(x, y) = \quad \exists z \exists z'(E_a(z, z') \wedge E_c(z, y) \wedge E_c(z', x))$$

Let us apply the unfolding operation again, from the only vertex without incoming edges. We obtain the "algebraic tree" of Figure 2, belonging to $\mathcal{T}_2$ (where for the moment one should ignore the dashed line).

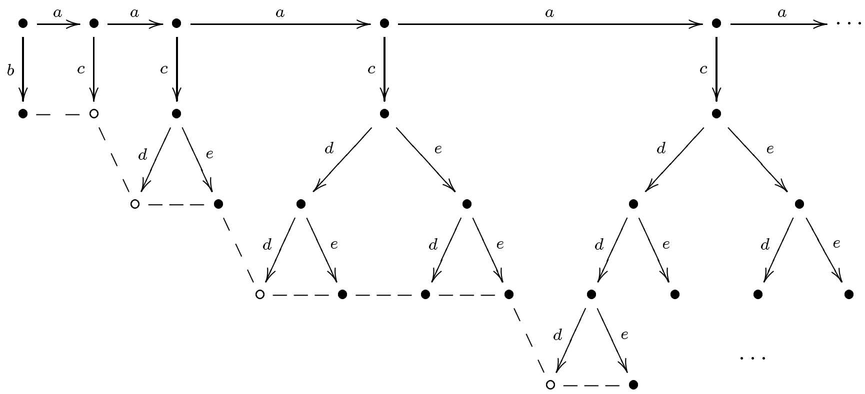

Fig. 2. Unfolding of the pushdown graph of Figure 1

As a next step, let us apply an MSO-interpretation to this tree which will produce a graph (V, E, P) in the class $\mathcal{G}_2$ (where E is the edge relation and P a unary predicate). Referring to Figure 2, V is the set of vertices which are located along the dashed line, E contains the pairs which are successive vertices along the dashed line, and P contains the special vertices drawn as non-filled circles. This structure is isomorphic to the structure $(\mathbb{N}, \mathrm{Succ}, P_2)$ with the successor relation Succ and predicate P_2 containing the powers of 2.

To prepare a corresponding MSO-interpretation, we use formulas such as $E_{d^*}(x, y)$ which expresses

> "all sets which contain x and are closed under taking E_d-successors contain y, and y has no E_d-successor"

As domain formula we use

$$\varphi(x) = \exists z(E_b(z, x) \vee \exists y(E_c(z, y) \wedge E_{(d+e)^*}(y, x))).$$

The required edge relation E is defined by $\psi(x, y) = \exists z \exists z'(\psi_1(x, y) \vee \psi_2(x, y) \vee \psi_3(x, y))$ where

- $\psi_1(x, y) \;=\; E_a(z, z') \wedge E_b(z, x) \wedge E_c(z', y)$
- $\psi_2(x, y) \;=\; E_a(z, z') \wedge E_{ce^*}(z, x) \wedge E_{cd^*}(z', y)$
- $\psi_3(x, y) \;=\; E_{de^*}(z, x) \wedge E_{ed^*}(z, y)$

Finally we define P by the formula $\chi(x) = \exists z \exists z'(E_c(z, z') \wedge E_{d^*}(z', x))$.

We infer that the MSO-theory of $(\mathbb{N}, \mathrm{Succ}, P_2)$ is decidable, a result first proved by Elgot and Rabin [14] with a different approach. The idea of [14], later applied to many other expansions of the successor structure by unary predicates, is to transform first a given MSO-sentence φ to an equivalent Büchi automaton $\mathcal{B}_\varphi$, so that $(\mathbb{N}, \mathrm{Succ}, P_2) \models \varphi$ iff $\mathcal{B}_\varphi$ accepts the characteristic 0-1-sequence α_{P_2} (with $\alpha_{P_2}(i) = 1$ iff $i \in P_2$). By contracting the 0-segments between the letters 1, one can modify α_{P_2} to an ultimately periodic sequence β such that $\mathcal{B}_\varphi$ accepts α_P iff $\mathcal{B}_\varphi$ accepts β. Whether $\mathcal{B}_\varphi$ accepts such a "regular model" β is decidable. Note that this reduction to a regular model depends on the sentence φ under consideration. The generation of $(\mathbb{N}, \mathrm{Succ}, P_2)$ as a model in $\mathcal{G}_2$ provides a uniform decidability proof.

In [7], the contraction method was adapted to cover all morphic predicates P (coded by morphic 0-1-words). Caucal [5] and Fratani and Sénizergues [15] have shown that such models $(\mathbb{N}, \mathrm{Succ}, P)$ also occur in the Caucal hierarchy[2]. In the present paper we discuss another structure treated already in [14]: the structure $(\mathbb{N}, \mathrm{Succ}, \mathrm{Fac})$ where Fac is the set of factorial numbers. We start from a simpler pushdown graph than the one used above and consider its unfolding, which is the comb structure indicated by the thick arrows of the lower part of the figure.

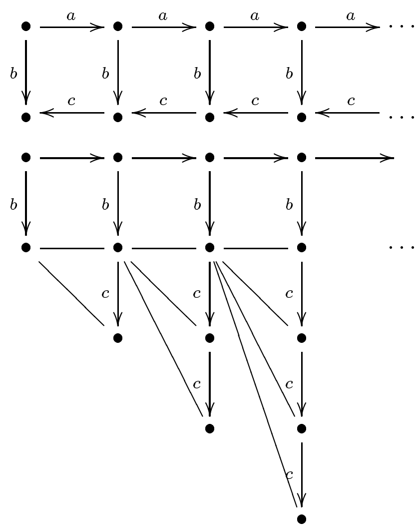

Fig. 3. Preparing for the factorial predicate

We number the vertices of the horizontal line by $0, 1, 2 \ldots$ and call the vertices below them to be of "level 0", "level 1", "level 2" etc. Now we use the simple MSO-interpretation which takes all tree nodes as domain and introduces for $n \geq 0$ a new edge from any vertex of level $n+1$ to the first vertex of level n. This introduces the thin lines in Figure 3 as new edges (assumed to point

[2] In [5] this is proved for morphic P; a more general class is obtained in [15].

backwards). The reader will be able to write down a defining MSO-formula. Note that the top vertex of each level plays a special role since it is the target of an edge labelled b, while the remaining ones are targets of edges labelled c.

Consider the tree obtained from this graph by unfolding. It has subtrees consisting of a single branch off level 0, 2 branches off level 1, $2 \cdot 3$ branches off level 2, and generally $(n+1)!$ branches off level n. Referring to the c-labelled edges these branches are arranged in a natural (and MSO-definable) order. To capture the structure $(\mathbb{N}, \mathrm{Succ}, \mathrm{Fac})$, we apply an interpretation which (for $n \geq 1$) cancels the branches starting at the b-edge target of level n (and leaves only the branches off the targets of c-edges). As a result, $(n+1)! - n!$ branches off level n remain for $n \geq 1$, while there is one branch off level 0. Numbering these remaining branches, the $n!$-th branch appears as first branch off level n. Note that we traverse this first branch off a given level by disallowing c-edges after the first c-edge. So a global picture like Figure 2 emerges, now representing the factorial predicate. Summing up, we have generated the structure $(\mathbb{N}, \mathrm{Succ}, \mathrm{Fac})$ as a graph in $\mathcal{G}_3$.

So far we have considered expansions of the successor structure of the natural numbers by unary predicates. We now discuss the expansion by an interesting unary function (here identified with its graph, a binary relation). It is the *flip function*, introduced in [21] in the study of a hierarchical time structure (involving different time granularities). The function flip associates 0 to 0 and for each nonzero n that number which arises from the binary expansion of n by modifying the least significant 1-bit to 0. An illustration of the graph Flip of this function is given in Figure 4. It is easy to see that the structure $(\mathbb{N}, \mathrm{Succ}, \mathrm{Flip})$

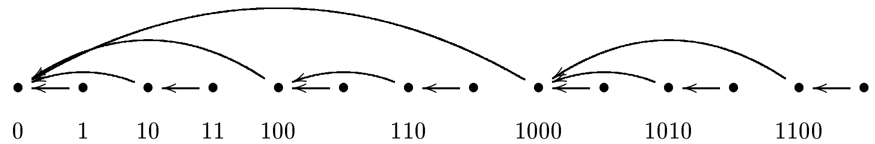

Fig. 4. Graph of flip function

can be obtained from the algebraic tree of Figure 2 by an MSO-interpretation. A Flip-edge will connect vertex u to the last leaf vertex v which is reachable by a d^*-path from an ancestor of u; if such a path does not exist, an edge to the target of the b-edge (representing number 0) is taken.

Other parts of arithmetic can also be captured by suitable structures of the Caucal hierarchy. For example, it can be shown that a semilinear relation (a relation definable in Presburger arithmetic) can be represented by a suitable graph. As the simplest example consider the relation $x + y = z$. It can be represented in a comb structure like Figure 3 where each vertical branch is infinite and for each edge a corresponding back-edge (with dual label) is introduced. In the unfolding of this infinite comb structure, a vertex on column x and row y

allows a path precisely of length $x + y$ via the back-edges to the origin. In this way, graphs can be generated which (as acceptors of languages) are equivalent to the Parikh automata of [19].

5 Outlook

The examples treated above should convince the reader that the Caucal Hierarchy supplies a large reservoir of interesting models where the MSO-theory is decidable. Many problems are open in this field. We mention some of them.
1. Studying and extending the range of the Caucal Hierarchy: We do not know much about the graphs on levels ≥ 3 of the Caucal hierarchy. Which structures of arithmetic (with domain $\mathbb{N}$ and some relations over $\mathbb{N}$) occur there? How to decide on which level a given structure occurs? Is it possible to obtain a still richer landscape of models by invoking the operation of tree iteration (possibly for structures with relations of arity > 2, as in [3])?
2. Comparison with other approaches to generate infinite graphs: There are representation results which allow to generate, for $n > 0$, the graphs of level n from a *single* tree of level n, respectively as the transition graphs of higher-level pushdown automata (see [5, 6] and the references mentioned there). There are as yet only partial results which settle the relation between the graphs of Caucal's hierarchy and the synchronized rational (or "automatic") graphs, the rational graphs, and the graphs generated by ground term rewriting systems (cf. e.g. [25, 20] and the references mentioned there).
3. Complexity of Model-Checking: The reduction of the MSO-model-checking problem for an unfolded graph to the corresponding problem for the original graph involves a non-elementary blow-up in complexity. When using restricted logics one can avoid this. For example, Cachat [6] has shown that μ-calculus model-checking over graphs of level n is possible in n-fold exponential time.

6 Acknowledgment

Many thanks are due to Didier Caucal for numerous fruitful discussions and to my collaborators and students Jan Altenbernd, Thierry Cachat, Christof Löding, and Stefan Wöhrle for their help.

References

1. D. Berwanger, A. Blumensath, The monadic theory of tree-like structures, in [16], pp. 285-302.
2. A. Blumensath, Prefix-recognizable graphs and monadic second-order logic, Rep. AIB-06-2001, RWTH Aachen, 2001.
3. A. Blumensath, Axiomatising tree-interpretable structures, in: *Proc. 19th STACS*, Springer LNCS 2285 (2002), 596-607.

4. D. Caucal, On infinite transition graphs having a decidable monadic theory, in: *Proc. 23rd ICALP* (F. Meyer auf der Heide, B. Monien, Eds.), Springer LNCS 1099 (1996), 194-205 [Full version in: *Theor. Comput. Sci.* 290 (2003), 79-115].
5. D. Caucal, On infinite graphs having a decidable monadic theory, in: *Proc. 27th MFCS* (K. Diks, W. Rytter, Eds.), Springer LNCS 2420 (2002), 165-176.
6. Th. Cachat, Higher order pushdown automata, the Caucal Hierarchy of graphs and parity games, in *Proc. 30th ICALP, 2003*, Springer LNCS (to appear).
7. O. Carton, W. Thomas, The monadic theory of morphic infinite words and generalizations, in: *Proc. 25th MFCS* (M. Nielsen, B. Rovan, Eds.), Springer LNCS 1893 (2000), 275-284.
8. B. Courcelle, Monadic second-order graph transductions: a survey, *Theor. Comput. Sci.* 126 (1994), 53-75.
9. B. Courcelle, The monadic second-order logic of graphs IX: machines and their behaviours, *Theor. Comput. Sci.* 151 (1995), 125-162.
10. A. Carayol, S. Wöhrle, The Caucal Hierarchy of infinite graphs in terms of logic and higher-order pushdown automata, submitted.
11. B. Courcelle, I. Walukiewicz, Monadic second-order logic, graph coverings and unfoldings of transition systems, *Ann. Pure Appl. Logic* 92 (1998), 35-62.
12. W. Damm, The IO and OI hierarchies, *Theor. Comput. Sci.* 20 (1982), 95-208.
13. H.D. Ebbinghaus, J. Flum, W. Thomas, *Mathematical Logic*, Springer, Berlin-Heidelberg-New York 1984.
14. C.C. Elgot, M.O. Rabin, Decidability and undecidability of extensions of second (first) order theory of (generalized) successor, *J. Symb. Logic* 31 (1966), 169-181.
15. S. Fratani, G. Sénizergues, personal communication.
16. *Automata, Logics, and Infinite Games* (E. Grädel, W. Thomas, Th. Wilke, Eds.), Springer LNCS 2500 (2002), Springer-Verlag, Berlin-Heidelberg-New York 2002.
17. T. Knapik, D. Niwinski, P. Urzyczyn, Deciding monadic theories of hyperalgebraic trees, in: *TCLA 2001* (S. Abramsky, Ed.), Springer LNCS 2044 (2001), 253-267.
18. T. Knapik, D. Niwinski, P. Urzyczyn, Higher-order pushdown trees are easy, in: *Proc. 5th FoSSaCS* (M. Nielsen, U. Engberg, Eds.), Springer LNCS 2303 (2002), 205-222.
19. F. Klaedtke, H. Rueß, Monadic second-order logic with cardinalities, in: Proc. 30th ICALP 2003, Springer LNCS (to appear).
20. C. Löding, Ground tree rewriting graphs of bounded tree width, in: *Proc. 19th STACS*, Springer LNCS 2285 (2002), 559-570.
21. A. Montanari, A. Peron, A. Policriti, Extending Kamp's Theorem to model time granularity, *J. Logic Computat.* 12 (2002), 641-678.
22. D. Muller, P. Schupp, The theory of ends, pushdown automata, and second-order logic, *Theor. Comput. Sci.* 37 (1985), 51-75.
23. M.O. Rabin, Decidability of second-order theories and automata on infinite trees, *Trans. Amer. Math. Soc.* 141 (1969), 1-35.
24. A. Semenov, Decidability of monadic theories, in: *Proc. MFCS 1984* (M.P. Chytil, V. Koubek, Eds.), Springer LNCS 176 (1984), 162-175.
25. W. Thomas, A short introduction to infinite automata. In: *Proc. 5th International Conference "Developments in Language Theory"*, Springer LNCS 2295 (2002), 130-144.
26. I. Walukiewicz, Monadic second-order logic on tree-like structures, *Theor. Comput. Sci.* 275 (2002), 311-346.