Automata on Directed Graphs: Edge versus Vertex Marking

Dietmar Berwanger, David Janin

Automata on Directed Graphs: Edge versus Vertex Marking*

Dietmar Berwanger and David Janin

LaBRI, Université Bordeaux 1
351, Cours de la Libération, 33405 Talence cedex, France
{dwb|janin}@labri.fr

Abstract. We investigate two models of finite-state automata that operate on rooted directed graphs by marking either vertices (V-automata) or edges (E-automata). Runs correspond to locally consistent markings and acceptance is defined by means of regular conditions on the paths emanating from the root. Comparing the expressive power of these two notions of graph acceptors, we show that E-automata are more expressive than V-automata. Moreover, we prove that E-automata are at least as expressive as the μ-calculus. Our main result implies that every MSO-definable tree language can be recognised by E-automata with uniform runs, that is, runs that do not distinguish between isomorphic subtrees.

Introduction

Extending the formal language theory of words and trees to general classes of graphs is a very challenging endeavour. During the last two decades, this topic has attracted much attention, and several notions of graph-language recognisability have been developed [15, 4, 25, 7, 10, 8, 6].

Over the domain of arbitrary finite graphs, Courcelle [4] proposes a powerful algebraic theory of languages recognisable via interpretations of tree-shaped terms. This characterisation inherits many features from the well-established theory of tree automata [24]. Thus, the notion of recognisability is closed under Boolean operations and projection, and its expressive power reaches beyond Monadic Second-Order Logic (MSO).

For directed graphs of bounded degree, Thomas [25] develops an automata-theoretic approach in terms of *tiling systems*, or more generally *graph acceptors*. These are devices that proceed by marking graph vertices according to local constraints, tailored to match the expressive power of the existential fragment of MSO (monadic Σ_1). The associated notion of recognisability is closed under union, intersection, and projection. In general it is not closed under complement since, on many classes of graphs, monadic Σ_1 is not closed under complement [27].

* This research has been partially supported by the EU RTN GAMES: "Games and Automata for Synthesis and Validation"

The most substantial research is, however, focused on the special case of finite directed acyclic graphs (see [26] for a survey). Robust notions of recognisability are available, in particular, for partially ordered sets which serve as models for concurrent computation [7, 17].

In this paper, we are concerned with arbitrary directed graphs of unbounded degree that may be infinite or contain cycles. Aside from the generic interest, this framework is fundamental for modelling the behaviour of state-transition systems. Our point of departure is the notion of a graph acceptor, introduced by Thomas in [25]. Adapting this notion to graphs of unbounded degree, we define two kinds of finite-state automata that operate on rooted directed graphs by marking either vertices (V-automata) or edges (E-automata), starting from the root and proceeding according to transitions specified by local first-order formulae. Because we are interested in the infinite behaviour of models, we equip these automata with ω-regular acceptance conditions over the marking of paths emanating from the designated root.

The question whether to label edges or vertices is subject to a fundamental choice in the design of automata that may revisit vertices of their input. The most common option is to mark vertices, but the alternative to mark edges also has some tradition, going back to the early 80ies and Kamimura and Slutzki's variant of tree-walking automata over planar graphs [15]. In [22], Potthoff, Seibert, and Thomas discuss the expressive power of graph acceptors that mark edges compared to those that mark vertices and show that, in their specific framework restricted to ranked acyclic graphs of bounded degree, edge and vertex marking lead to the same notion of recognisability.

Taking up an analogue investigation for our extended setting, we find that the situation is radically different over arbitrary graphs, even if global acceptance conditions are not involved. In a comparative study, we separate the expressive power of vertex and edge-marking automata and relate it to Monadic Second-Order Logic and to its bisimulation-invariant fragment, the μ-calculus. Our main result establishes a correspondence between runs of edge-marking automata over arbitrary graphs and vertex-marking automata over trees obtained by unravelling these graphs. Besides showing that edge-marking automata capture the μ-calculus over arbitrary graphs, as vertex-marking automata do on trees, this result opens a perspective on *uniform* recognisability of graph languages.

Outline. The paper is structured as follows. After fixing our notation in Section 1, we introduce V-automata and E-automata in Section 2, and point out some elementary properties. Thus, recognisable classes of graphs are closed under conjunction, disjunction and projection, but not under complement. In terms of MSO, E-automata and V-automata define graph properties in the level Σ_3 of the monadic quantifier-alternation hierarchy, respectively MSO_2, the variant of MSO augmented with edge quantifiers [5].

In Section 3, we show that, over arbitrary directed graphs, E-automata are strictly more expressive than V-automata. Actually, E-automata can describe properties like perfect matching that are not yet definable in MSO. However, we

show that even when we restrict to MSO-definable properties, E-automata are more expressive than V-automata. A separating property is directed reachability.

In Section 4, we investigate the relation between our automata model and the (counting) μ-calculus [16, 2], a logic which captures the MSO-definable properties that are invariant under (counting) bisimulation [14]. These properties are particularly relevant for the specification of state-transition models, because they do not distinguish between a model and its behaviour, understood as the unravelling of its possible computations in a tree-like manner.

Our main technical result is a simulation theorem relating recognisability of graphs and their unravelling. It states that, for every V-automaton $\mathcal{A}$, there exists an equivalent E-automaton that recognises precisely the class of graphs whose unravelling is accepted by $\mathcal{A}$. Intuitively, this means that E-automata can simulate on their input graph the behaviour of a V-automaton on the unravelling of this graph. Notice that every element of the input graph may have infinitely many copies in its unravelling, such that the V-automaton has potentially infinite "space" to apply his marking. However, we argue that all these copies can be encoded into a single marking of the original input element.

We discuss three consequences of this theorem. First, it implies that E-automata subsume the μ-calculus over arbitrary directed graphs, yielding an operative model that differs substantially from previous automata-theoretic characterisations of the μ-calculus (see, e.g., [20, 9, 3, 13]), where automata essentially run on a tree unravelling rather than on the input graph. The model of V-automata is, however, not strong enough to capture the μ-calculus over arbitrary graphs, since it cannot express directed reachability, which is μ-definable. Secondly, it follows that our definition of E-automata, with a universal linear-time condition on infinite computation paths, is fairly robust, as far as expressive power is concerned. In particular, adding branching-time acceptance condition over the computation trees of automata, i.e., the unravelling of runs, would not increase their expressiveness. Finally, when rephrased in terms of automata over infinite trees, our main result shows that every MSO-definable language of infinite trees can be recognised by a non-deterministic tree automata with uniform runs, i.e., runs that do not distinguish between isomorphic subtrees. In other words, shared substructure of the input can also be shared by the run. This uniformisation result is particularly surprising as it does not incur a decrease in expressiveness, as it is usually the case for such normalisations [11, 28].

1 Background

1.1 Words.

A *word* over an alphabet A is a partial function $\alpha : \mathbb{N} \to A$ with prefix-closed domain. We say that α is finite, when $\mathrm{dom}(\alpha)$ is so. The set of finite words over the alphabet A is denoted by A^*, whereas the set of infinite words is denoted by A^ω; the union of these two sets is A^∞. The concatenation of a word $\alpha \in A^*$ with a word $\beta \in A^\infty$ is denoted by $\alpha\beta$. This notation naturally extends to sets of words. We sometimes refer to the element $\alpha(i)$ of a word α by α_i.

In the sequel, we will use sets $L \subseteq A^\omega$ as infinitary acceptance conditions. We say that L is a *ω-regular* condition when L a finite union of languages of the form LU^ω where $L \subseteq A^*$ and $U \subseteq A^*$ are nonempty and regular. The set L is called a *parity condition*, if there exist a *priority* mapping $\Omega : A \to \mathbb{N}$ of finite image, such that $L = \{\alpha \in A^\omega \mid \liminf \Omega(\alpha) \equiv 0 \pmod 2\}$, i.e., L is the set of infinite sequences where the least priority that occurs infinitely often is even.

1.2 Graphs.

A *graph* is a structure $\mathcal{G} = (V, E)$ over a domain V of *vertices* with a binary *edge* relation $E \subseteq V \times V$. A *rooted* graph $\mathcal{G}, u$ is a graph with a distinguished *root* vertex u. Given an edge $(v, w) \in E$, we refer to v as its *source* and to w as its *target.* A *(directed) path* in the graph $\mathcal{G}$ is a finite or infinite non-empty sequence $v_1, v_2, \cdots \in V^\infty$ of vertices such that for any two consecutive elements v_i and v_{i+1}, we have $(v_i, v_{i+1}) \in E$. An *undirected path* in $\mathcal{G}$ is a sequence $v_1, v_2 \cdots \in V^\infty$ where, for any two consecutive elements v_i and v_{i+1}, either $(v_i, v_{i+1}) \in E$ or $(v_{i+1}, v_i) \in E$. The *distance* $d(v, v')$ between two vertices v, v' of V, is the least number n such that there exists an undirected path $v_1, \ldots, v_n$ in $\mathcal{G}$ with $v_1 = v$ and $v_n = v'$. If no such path exists, we set $d(v, v') = \infty$.

Fix an alphabet C of colours. For a graph $\mathcal{G} = (V, E)$, a *vertex colouring* over C is a function $\lambda : V \to C$; likewise, an *edge colouring* over C is a function $\gamma : E \to C$. We refer to the expansion of a graph by edge and/or vertex colourings as a *coloured* graph. When, instead of total functions, we consider partial edge or vertex colourings, we refer to them as *markings*. The elements in the domain of such a partial colouring are said to be *marked* by the respective function. In addition to this, we say that a vertex $v \in V$ is *involved* in an edge marking $\gamma : E \to C$, if it is either the source or the target of an edge marked by γ. A vertex colouring $\lambda : V \to C$ is naturally extended to a path π in $\mathcal{G}$ by setting $\lambda(\pi) = \lambda \circ \pi \in C^\infty$. Likewise, for an edge colouring $\gamma : E \to C$, we define the edge colouring of a path $\pi = v_1, v_2, \ldots$ in $\mathcal{G}$ to be the word $\gamma(\pi) \in C^\infty$, such that $\gamma(\pi)(i) = \gamma(\pi(i), \pi(i+1))$ for all indices i with $i+1 \in \mathrm{dom}(\pi)$. For markings, the corresponding definitions are restricted to paths in $\mathcal{G}$ that involve only marked elements.

Besides the functional notation $\mathcal{G} = (V, E, \lambda)$ for vertex-marked graphs, it is sometimes convenient to use a relational notation $\mathcal{G} = (V, E, (P_c)_{c \in C})$, with monadic symbols P_c interpreted by $P_c^{\mathcal{G}} := \{v \in V \mid \lambda(v) = c\}$, for every $c \in C$. Similarly, for edge-marked graphs $\mathcal{G} = (V, E, \gamma)$, we use the notation $\mathcal{G} = (V, E, (R_c)_{c \in C})$ with binary relational symbols R_c interpreted by $R_c^{\mathcal{G}} := \{(v, w) \in E \mid \gamma(v, w) = c\}$, for every colour $c \in C$.

Bisimulation. The main part of this paper is concerned with devices taking as input vertex-coloured graphs. For the sake of clarity, the following definitions are formulated for this setting, the generalisation to edge-coloured graphs being straightforward.

A *counting bisimulation* between two vertex-coloured graphs $\mathcal{G} = (V, E, \lambda)$ and $\mathcal{G}' = (V', E', \lambda')$ is a relation $Z \subseteq V \times V'$ such that, if $(v, v') \in Z$, then

$\lambda(v) = \lambda'(v')$ and Z contains a bijection between the sets $\{w \mid (v,w) \in E\}$ and $\{w' \mid (v',w') \in E'\}$. Two rooted graphs $\mathcal{G}, u$ and $\mathcal{G}', u'$ are counting *bisimilar*, if there exists a counting bisimulation Z between them with $(u,u') \in Z$. Two vertices v, v' of a graph $\mathcal{G}$ are counting bisimilar, if $\mathcal{G}, v$ and $\mathcal{G}, v'$ are so.

The *unravelling* of a graph $\mathcal{G} = (V, E, \lambda)$ from a vertex $u \in V$ is the graph $\mathcal{T}(\mathcal{G}, u)$ with domain $V^{\mathcal{T}}$ consisting of all directed paths π through $\mathcal{G}$ that start from u, edge relation $E^{\mathcal{T}}$ containing all the pairs $(\pi, \pi v) \in V^{\mathcal{T}} \times V^{\mathcal{T}}$, and vertex colouring $\lambda^{\mathcal{T}}$ defined by $\lambda^{\mathcal{T}}(\pi v) = \lambda(v)$. A rooted graph $\mathcal{G}, u$ is a *tree*, if it is isomorphic to its unravelling $\mathcal{T}(\mathcal{G}, u), u$. For trees, and in particular for unravellings, we will generally not specify the root explicitly. Obviously, the natural projection which sends every path $v_1, \ldots, v_\ell \in V^{\mathcal{T}}$ to its last node v_ℓ defines a counting bisimulation between $\mathcal{G}$ and $\mathcal{T}(\mathcal{G}, u)$. It is well-known (see, e.g., [12]), that two graphs $\mathcal{G}, u$ and $\mathcal{G}', u$ are counting bisimilar if, and only if, their unravelling $\mathcal{T}(\mathcal{G}, u)$ and $\mathcal{T}(\mathcal{G}, u)$ are isomorphic.

1.3 Logic.

We consider standard predicate logics, in particular First-Order Logic (FO) and Monadic Second-Order Logic (MSO) interpreted over coloured graphs. Given an alphabet C of vertex colours, we write P_C for the collection of monadic symbols $(P_c)_{c \in C}$. When using edge colours from an alphabet D, we write R_D for the collection of binary relational symbols $(R_d)_{d \in D}$. Thus, the vocabulary of formulae is typically (a subset of) $E \cup P_C \cup R_D$. We refer to any formula $\varphi(x)$ with precisely one free first-order variable as a *predicate*, and to a formula without free first-order variables as a *sentence*. For any integer k, the *k-sphere* around a vertex v of a graph is the set of vertices w such that $d(v,w) \leq k$. An FO-predicate $\varphi(x)$ is called *k-local* around x, if it is equivalent to the predicate $\varphi'(x)$ obtained by relativising every quantifier in φ to elements of the k-sphere around x. We say that a predicate $\varphi(x)$ is local around x if it is k-local for some k.

We also consider MSO_2, the extension of MSO where quantification over sets of edges is provided [5]. The syntax of MSO_2 allows binary second-order variables and quantification over them. Given a graph $\mathcal{G} = (V, E)$, the semantics of this quantification is however relativised to subsets of E. It is well known that, on arbitrary graphs, MSO_2 is strictly more expressive than MSO. Though, on trees and on graphs of bounded degree, hence in particular on grids, MSO and MSO_2 are equally expressive [5].

The monadic quantifier-alternation hierarchy is defined as follows. The first level, monadic Π_0, also called monadic Σ_0, is the set of FO-formulae. Then, for every n, the level monadic Σ_{n+1} (resp. monadic Π_{n+1}) is the closure of the set of monadic Π_n-formulae (respectively monadic Σ_n-formulae) under existential quantification (respectively universal quantification). This hierarchy is known to be strict over arbitrary graphs, i.e., for every $n \in \mathbb{N}$, there exists a property φ_n definable in MSO that is not definable in the level Σ_n of the monadic hierarchy. More recently [18], it has also been shown that the monadic hierarchy is strict already over finite grids. Since MSO and MSO_2 are equally expressive over

grids, and because the translation between the two preserves the quantification structure, this strictness result carries over to MSO_2.

A class $\mathcal{K}$ of rooted graphs is counting-bisimulation *closed* if, for any graph $\mathcal{G}, u$, we have $\mathcal{G}, u \in \mathcal{K}$ if, and only if, there exists a counting-bisimilar graph $\mathcal{G}', u' \in \mathcal{K}$. A sentence φ of FO or MSO is counting-bisimulation *invariant*, if its model class is counting-bisimulation closed, that is, for any two counting-bisimilar graphs $\mathcal{G}, u$ and $\mathcal{G}, u'$ we have $\mathcal{G}, u \models \varphi$ if, and only if, $\mathcal{G}', u' \models \varphi$. The (counting) μ-calculus [16, 2, 12] is an extension of (counting) modal logic with fixed-point operators that provides an effective syntax for the (counting) bisimulation-invariant fragment of MSO [14, 12].

2 Vertex and edge-marking automata

Traditionally, automata are finite-state devices that produce a marking of their input objects with states. The process of marking starts from designated input elements and propagates locally, depending on the local properties of the input structure and of the previously produced marking. When the input structures are homogeneous, these propagation transitions can often be described pictorially. However, as we are concerned with coloured directed graphs of unbounded branching which are not homogeneous, we choose a more abstract way to describe transitions using local FO-formulae that refer to both the input structure and the produced marking.

We introduce automata that take as input graphs with vertex colourings and produce either edge or vertex markings. Whenever we speak of a Σ-coloured graph, we mean a graph with a vertex colouring over a finite alphabet Σ.

Definition 1 (V-automaton). *Let Σ be a finite alphabet of vertex colours. A* vertex-marking automaton (V-automaton) *for Σ-coloured graphs is a tuple*

$$\mathcal{A} = (Q, \Sigma, \delta_0, \delta, Acc)$$

with a finite set Q of states*, two local formulae $\delta_0(x), \delta(x) \in$ FO, called* root constraint *respectively* transition specification*, over the vocabulary $E \cup P_\Sigma \cup P_Q$ of Σ-coloured graphs augmented with unary symbols associated to the states of Q, and an ω-regular* acceptance condition *$Acc \subseteq Q^\omega$.*

Given a Σ-coloured graph $\mathcal{G} = (V, E, \lambda)$ with a designated root u, a *run* of the V-automaton $\mathcal{A}$ on $\mathcal{G}, u$ is a vertex marking $\rho : V \to Q$ with the following properties:

(i) *initial condition:* $\mathcal{G}, \rho \models \delta_0(u)$, and
(ii) *local consistency:* for every vertex $v \in V$ marked by ρ, we have $\mathcal{G}, \rho \models \delta(v)$.

A run ρ is *accepting* if, for every infinite path π in $\mathcal{G}$ that starts from u and consists of vertices marked by ρ, we have $\rho(\pi) \in Acc$. A graph $\mathcal{G}, u$ is *accepted* by $\mathcal{A}$, if there exists an accepting run of $\mathcal{A}$ on $\mathcal{G}, u$. We define $\mathcal{L}_V(\mathcal{A})$ to be the

class of all rooted graphs accepted by the V-automaton $\mathcal{A}$. A class $\mathcal{K}$ of rooted graphs is V-*recognisable*, if there exists a V-automaton $\mathcal{A}$ with $\mathcal{L}_V(\mathcal{A}) = \mathcal{K}$.

We observe that V-automata generalise most of the classical nondeterministic automata models, such as top-down or bottom-up automata over finite trees, but also, e.g., Muller-automata over infinite trees.

Remark 1. Notice that the run of an automaton is independent of the part of the input graph that is unreachable from the designated root, since local consistency is enforced just at marked vertices and extends only to a neighbourhood of the current vertex, so that no marking of an unreachable vertex can be required. However we may assume, without loss of generality, that accepting runs of a V-automaton $\mathcal{A}$ are total functions. To achieve this, we can add an extra dummy state $\bot$ and modify the transition specification to be $P_\bot \vee \delta$ and the acceptance condition to include the set $Q^*\{\bot\}(Q \cup \{\bot\})^\omega$.

Lemma 1. *Every V-recognisable class of graphs is definable in the level Σ_3 of* MSO*.*

Proof. Let $\mathcal{A} = (Q, \Sigma, \delta_0, \delta, Acc)$ be a V-automaton with state set $\{1, \dots, n\}$. We construct an MSO-formula $\varphi_\mathcal{A}$ of the form $\exists P_1 \cdots \exists P_n \big(\forall x \psi(x) \wedge \varphi_{Acc}\big)$ with $\varphi_{Acc} \in \Pi_2$ and $\psi \in \Sigma_0$ such that, for every graph $\mathcal{G}$ with a designated root u, we have $\mathcal{G}, u \models \varphi_\mathcal{A}$ if, and only if, $\mathcal{G}, u \in \mathcal{L}_V(\mathcal{A})$. In this formula, the block of existential quantifiers $\exists P_1 \cdots \exists P_n$ guesses a vertex marking, the subformula $\forall x \psi(x)$ expresses the local constraints,

$$\psi(x) := \big(x = u \to \delta_0(x)\big) \wedge \big(\bigvee_{q \in Q} P_q \to \delta(x)\big),$$

and φ_{Acc} checks the infinitary path condition. To see that φ_{Acc} can be described in Σ_2, notice that its negation $\neg\varphi_{Acc}$ expresses the property that there exists a marked path starting at u that does not satisfy the ω-regular acceptance condition Acc. Using the representation of $Q^\omega \setminus Acc$ as a non-deterministic Büchi word automaton, this property can be defined by a monadic Σ_2-formula (more precisely, a $\mu\nu$-formula of the μ-calculus). □

Our second automata model differs from V-automata only by its way of applying state labels to edges rather than to vertices.

Definition 2 (E-automaton). *Let Σ be a finite alphabet of vertex colours. An* edge-marking automaton (E-automaton) *for Σ-coloured graphs is a tuple*

$$\mathcal{A} = (Q, \Sigma, \delta_0, \delta, Acc),$$

with a finite set Q of states*, two local predicates $\delta_0(x), \delta(x) \in$ FO, called root constraint and transition specification, over the vocabulary $E \cup P_\Sigma \cup R_Q$ of Σ-coloured graphs augmented with binary relational symbols associated to the states of Q, and an ω-regular* acceptance condition *$Acc \subseteq Q^\omega$.*

Given a Σ-coloured graph $\mathcal{G} = (V, E, \lambda)$ with a designated root u, an accepting run of the E-automaton $\mathcal{A}$ on $\mathcal{G}, u$ is now an edge marking $\rho : E \to Q$ with the following properties:

(i) *initial condition:* $\mathcal{G}, \rho \models \delta_0(u)$, and
(ii) *local consistency:* for every vertex $v \in V$ involved in ρ, we have $\mathcal{G}, \rho \models \delta(v)$.

A run ρ is *accepting* if, for every infinite path $\pi = v_0, v_1, \ldots$ in $\mathcal{G}$ that starts from the root $u = v_0$ and proceeds along edges (v_i, v_{i+1}) marked by ρ, we have $\rho(\pi) \in Acc$. As in the case of V-automata, we say that the graph $\mathcal{G}, u$ is *accepted* by $\mathcal{A}$ if there exists an accepting run of $\mathcal{A}$ on $\mathcal{G}, u$, and we define $\mathcal{L}_E(\mathcal{A})$ to be the class of rooted graphs accepted by the E-automaton $\mathcal{A}$. A class $\mathcal{K}$ of rooted graphs is E-*recognisable*, if there exists an E-automaton $\mathcal{A}$ with $\mathcal{L}_E(\mathcal{A}) = \mathcal{K}$.

We remark that there are E-recognisable classes of graphs that cannot be described in MSO. An example is the class of graphs that allow a perfect matching between the vertices reachable from the root. Essentially, this is because edge marking corresponds to a quantification over sets of edges which is not available in MSO. Nevertheless, for every E-automaton, the class $\mathcal{L}_E(\mathcal{A})$ is definable in MSO_2. The proof is a straightforward adaptation of the proof of Lemma 1.

Lemma 2. *Every E-recognisable class of graphs is definable in the level Σ_3 of MSO_2.*

2.1 Elementary properties

We survey some elementary properties of our graph automata. An essential feature is that we can specify grid properties, even without marking edges, by simulating a grid vocabulary consisting of two functional edge symbols, say R_N and R_E, standing for North and East. Towards this, we use two extra monadic symbols P_N and P_E, and we require that the root is in both P_N and P_E, and every vertex v has exactly two outgoing edges (v, v_E) and (v, v_N) such that either both or none of v and v_E belong to P_E whereas they never belong together to P_N and, similarly, either both or none of v and v_N belong to P_N whereas they never belong together to P_E. The intended grid relations can now be defined by $R_N := \{(x, y) \in E \mid P_N(x) \to P_N(y)\}$ and $R_E := \{(x, y) \in E \mid P_E(x) \to P_E(y)\}$.

Lemma 3. *E-automata and V-automata with $k{+}1$-local transition specifications are strictly more expressive than E-automata respectively V-automata with k-local transition specification.*

Proof. A corresponding statement for tiling systems is proved in [26]. The argument carries over to our automata. □

Lemma 4. *Both V-recognisable and E-recognisable classes of graphs are closed under union, intersection, and projection.*

Proof. These properties follow directly from the definition of our automata model. To show, for instance, closure under union for V-automata, consider two V-automata $\mathcal{A} = (Q, \Sigma, \delta_0, \delta, Acc)$ and $\mathcal{A}' = (Q', \Sigma, \delta_0', \delta', Acc')$. Then the automaton over the state set $Q \cup Q'$, with root constraint $\delta_0 \vee \delta_0'$, transition specification $\delta \vee \delta'$, and infinitary condition $Acc \cup Acc'$ recognises $\mathcal{L}_V(\mathcal{A}) \cup \mathcal{L}_V(\mathcal{A}')$. □

Lemma 5. *Neither V-recognisable nor E-recognisable classes of graphs are closed under complement. The statement also holds for classes of finite graphs.*

Proof. We have already seen that FO-definable classes of finite grids can be recognised by V-automata and also by E-automata. If, in addition to being closed under projection, recognisable classes were closed under complement, any MSO-definable class of finite grids would be recognisable, and hence Σ_3-definable, by Lemma 1 respectively Lemma 2. This contradicts the infiniteness of the monadic hierarchy over grids [18]. □

3 E-automata versus V-automata

In this section, we compare the expressive power of the two notions of automata. It turns out that E-automata are, even on finite graphs, strictly more expressive than V-automata.

3.1 Encoding V-automata into E-automata

Proposition 1. *Every V-recognisable class of graphs is also E-recognisable.*

Proof. For simplicity, we assume here that automata are normalised so that accepting runs are total functions. Given a V-automaton $\mathcal{A} = (Q, \Sigma, \delta_0, \delta, Acc)$, we construct an E-automaton $\mathcal{B} = (Q \times Q, \Sigma, \delta_0', \delta', Acc')$ that marks edges of its input graph with pairs of (vertex) states from Q, in such a way that the marking of a vertex v with a state q in a run of $\mathcal{A}$ corresponds to the marking of all incoming and outgoing edges from v by pairs of the form (q', q) respectively (q, q') in a run of $\mathcal{B}$.

The following one-local formula expresses that the edge marking around a vertex z encodes a vertex marking of z by q:

$$\varphi_q(z) = \forall y \Big[\Big(E(y,z) \to \bigvee_{q' \in Q} R_{(q',q)}(y,z) \Big) \wedge \Big(E(z,y) \to \bigvee_{q' \in Q} R_{(q,q')}(z,y) \Big) \Big].$$

Now, we define the root constraint $\delta_0'(x)$ and the transition specification $\delta'(x)$ for $\mathcal{B}$ to be the conjunction of $\bigvee_{q \in Q} \varphi_q(x)$ with the formula obtained from $\delta_0(x)$ respectively $\delta(x)$ by replacing every atom $P_q z$ with the subformula $\varphi_q(z)$. The acceptance condition Acc' consists of all infinite words $\beta \in (Q \times Q)^\omega$ for which there exists a word $\alpha \in Acc$, such that $\beta_i = (\alpha_i, \alpha_{i+1})$, for all indices i.

To verify that the construction is correct in the case of graphs with no isolated vertices (these need to be treated separately, but pose no great difficulty), let

us consider a rooted graph $\mathcal{G}, u \in \mathcal{L}_V(\mathcal{A})$ and let $\rho : V \to Q$ be an accepting run of the V-automaton $\mathcal{A}$. Then, the marking $\rho' : E \to Q \times Q$ defined by $\rho'(v, w) = (\rho(v), \rho(w))$, for every edge $(v, w) \in E$, is an accepting run of the E-automaton $\mathcal{B}$ on $\mathcal{G}, u$. Conversely, for an accepting run $\rho' : E \to Q \times Q$ of $\mathcal{B}$ on a graph $\mathcal{G}, u$, the conditions δ'_0 and δ' ensure that, for every vertex $v \in V$, there exists a unique state $q \in Q$ such that $\mathcal{G}, \rho' \models \varphi_q(v)$; the vertex-marking $\rho : V \to Q$ defined by associating to every vertex $v \in V$ this unique state, is an accepting run of the V-automaton $\mathcal{A}$ on $\mathcal{G}$. □

3.2 E-automata are more expressive than V-automata

Next, we prove that edge-marking yields a strict increase in expressiveness over vertex-marking. Instead of relying on properties from $\mathrm{MSO}_2 \setminus \mathrm{MSO}$ we show, moreover, that separating properties exist already in MSO.

Theorem 1. *There exists an E-recognisable class of directed graphs that is* MSO*-definable, but not V-recognisable.*

Proof. We show that directed reachability is definable by an E-automaton, but not by a V-automaton. Let $\mathcal{K}$ be the class of graphs $\mathcal{G}, u$ over the alphabet $\Sigma = \{a, b\}$ in which there exists a finite directed path from u to a vertex v with $\lambda(v) = a$. Clearly, this class is definable in MSO, in fact, already in the μ-calculus by the formula $\mu X.(P_a \vee \Diamond X)$.

To see that $\mathcal{K}$ is E-recognisable, consider the automaton $\mathcal{B} = (\{q\}, \Sigma, \delta_0, \delta, \emptyset)$ with only one state q. The root constraint δ_0 states that, if the root is not coloured with a, precisely one outgoing edge is marked,

$$\delta_0(x) := \neg P_a \to \exists y \big(Exy \wedge Rxy \wedge \forall z(Exy \wedge Rxz \to z = y) \big).$$

The transition specification requires that, if a vertex is not coloured with a and has an incoming marked edge, then precisely one outgoing edge is marked,

$$\delta(x) := \big(\neg P_a \wedge \exists y(Eyx \wedge Ryx) \big) \to \exists y \big(Exy \wedge Rxy \wedge \forall z(Exz \wedge Rxz \to z = y) \big).$$

In this way, accepting runs of $\mathcal{B}$ correspond to markings of a directed path from the root to some vertex of colour a. Therefore, $\mathcal{L}_V(\mathcal{B}) = \mathcal{K}$.

To show that $\mathcal{K}$ is not V-recognisable, we use an idea of Ajtai and Fagin [1]. Towards a contradiction, let $\mathcal{A} = (Q, \Sigma, \delta_0, \delta, Acc)$ be a V-automaton with $\mathcal{L}_V(\mathcal{A}) = \mathcal{K}$. We fix real number p between 0 and 1. For every integer n, we construct a random Σ-coloured graph $\mathcal{G}_p^n = (V_n, E_n, \lambda_n)$ over the domain $V_n = \{0, \ldots, n\}$ with the following edge relation:

- for all $i < n$, the *forward-edge* $(i, i + 1)$ is contained in E_n, and
- for every (i, j) with $1 \leq i < j \leq n$, the *back-edge*(j, i) is contained in E_n with probability p.

The vertices of $\mathcal{G}_p^n$ are coloured by $\lambda_n(i) = b$, for all $i < n$, and $\lambda_n(n) = a$.

Clearly, for every n, the graph $\mathcal{G}_p^n$ with root 0 belongs to $\mathcal{K}$. For every element $i < n$, let now $\mathcal{G}_{p,i}^n$ be the graph obtained from $\mathcal{G}_p^n$ by removing the forward-edge $(i, i+1)$. Obviously, $\mathcal{G}_{p,i}^n, 0 \notin \mathcal{K}$. A technical theorem of [1] implies that, for every size of a vertex-alphabet C and every quantifier rank r, there exists an integer n and a value for p, such that the following property holds with positive probability: for every marking $\rho : V_n \to Q$ of $\mathcal{G}_p^n$, there exists an index $i < n$ such that, for every FO-formula of quantifier rank at most r, if $\mathcal{G}_p^n, \rho \models \varphi$ then $\mathcal{G}_{p,i}^n, \rho \models \varphi$.

Applying this statement to an accepting run ρ of $\mathcal{A}$ on $\mathcal{G}_p^n, 0$, it implies that there exist values $n \in \mathbb{N}$, $p \in (0,1)$, and $i < n$ such that, with positive probability, ρ is also an accepting run of $\mathcal{A}$ on $\mathcal{G}_{p,i}^n$ since the initial condition and the local consistency of $\mathcal{A}$ cannot distinguish between the graphs $\mathcal{G}_{p,i}^n, 0$ and $\mathcal{G}_p^n, 0$ when they are marked in the same way. Notice that the path condition cannot discriminate between these graphs either, since any infinite path in the former is also an infinite path in the latter. Hence, we have $\mathcal{G}_{p,i}^n, 0 \in \mathcal{L}_V(\mathcal{A})$, in contradiction to our assumption that $\mathcal{L}_V(\mathcal{A}) = \mathcal{K}$. □

4 Simulation and uniform recognisability

In this section, we establish a relation between runs of automata on graphs with runs on their unravelling. This allows us, on the one hand, to conclude that over arbitrary graphs E-automata capture the counting μ-calculus and, on the other hand, that V-automata over trees can be normalised to mark isomorphic subtrees identically.

4.1 E-automata and the μ-calculus

Observe that, on trees, the marking of edges can be simply moved to their targets, and hence the notions of E-recognisability and V-recognisability coincide. Furthermore, V-automata –with one-local root constraint and transition specification– generalise MSO tree-automata [29, 12]. Accordingly, for classes of trees, recognisability equals MSO-definability.

Theorem 2 (Rabin [23], Muchnik and Walukiewicz [19, 29]). *For every* MSO *formula φ there exists a one-local V-automaton $\mathcal{A}_\varphi$ such that, for every tree $\mathcal{T}$, we have $\mathcal{T} \models \varphi$ if, and only if, $\mathcal{T} \in \mathcal{L}_V(\mathcal{A}_\varphi)$.*

Proof. This follows from Walukiewicz's automata-theoretic characterisation of MSO on trees. As already observed in [12], Walukiewicz's automata are, in this case, V-automata with transition specification definable by means of counting one-local formulae. □

The following theorem is our main result. Informally, it states that every V-automaton on trees can be simulated by an E-automaton on graphs that is

equivalent in the sense that a graph is accepted by the E-automata if, and only, if its unravelling is accepted by the V-automaton.

Theorem 3. *For every V-automaton $\mathcal{A}$ we can construct an E-automaton $\mathcal{B}$ such that a rooted graph $\mathcal{G}, u$ is accepted by $\mathcal{B}$ if, and only if, its tree unravelling $\mathcal{T}(\mathcal{G}, u)$ is accepted by $\mathcal{A}$.*

Proof. Let $\mathcal{A} = (Q, \Sigma, \delta_0, \delta, Acc)$ be a V-automaton. Without loss of generality [29], we may assume that Acc is a parity condition, and that the root constraint and the transition specification are one-local formulae.

As in the proof of Proposition 1, we construct an E-automaton that operates by encoding a vertex-marking run into an edge-marking one. However, in that setting, the two markings were defined on the same graph. Here, we need to overlay all the (counting bisimilar) copies of a graph vertex that occur in the unravelling. To handle this, we proceed by a power-set construction. Essentially, we intend to mark every edge (v, w) in $\mathcal{G}$ by the set of all pairs of states (q, q'') that label copies of v and w connected in $\mathcal{T}(\mathcal{G}, u)$.

Formally, let $\mathcal{B} = (Q', \Sigma, \delta_0', \delta', Acc')$ be an E-automaton with set of states $Q' = \mathscr{P}(Q \times Q)$. The following formula expresses that the edge marking around a vertex z encodes that a copy of z in the unravelling tree is marked by q:

$$\varphi_q(z) := \forall y \Big[\Big(E(z,y) \to \bigvee_{\substack{q' \in Q' \\ (q,q'') \in q'}} R_{q'}(z,y) \Big) \wedge \Big(E(y,z) \to \bigvee_{\substack{q' \in Q' \\ (q'',q) \in q'}} R_{q'}(y,z) \Big) \Big].$$

Observe that the corresponding formulae in the proof of Proposition 1 are mutually exclusive for different states $q \in Q$. Here, several formulae φ_q may hold at one vertex, as it corresponds to the overlay of several vertices in the unravelling.

The root constraint $\delta_0'(x)$ and the transition specification $\delta'(x)$ for $\mathcal{B}$ is obtained from $\delta_0(x)$ respectively $\delta(x)$ by replacing every atom $P_q z$ with the subformula $\varphi_q(z)$. The infinitary path condition is defined in terms of traces. Given an infinite word $\beta \in (\mathscr{P}(Q \times Q))^\omega$, we say that a word $\alpha \in Q^\omega$ is a *trace* of β, if it is the case that $(\alpha_i, \alpha_{i+1}) \in \beta_i$ for every index i. The acceptance condition Acc' consists of all infinite words $\beta \in Q'^\omega$ for which every trace α belongs to Acc. Clearly, if Acc is regular, then Acc' is regular as well. Notice however, that even when Acc is a parity condition, Acc' is not a prefix-invariant property.

To verify that the construction is correct, let $\mathcal{G} = (V, E, \lambda)$ be a coloured graph with a distinguished root u, and let $\mathcal{T}(\mathcal{G}, u) = (V^{\mathcal{T}}, E^{\mathcal{T}}, \lambda^{\mathcal{T}})$ be its unravelling from u. Recall that $\mathcal{T}(\mathcal{G}, u)$ is built from $\mathcal{G}$ by taking as vertices the finite paths in $\mathcal{G}$ starting from u. Let $f : V^{\mathcal{T}} \to V$ be the projection that maps every finite path in $V^{\mathcal{T}}$ to its last vertex. Assuming that $\mathcal{T}(\mathcal{G}, u) \in \mathcal{L}_V(\mathcal{A})$, let $\rho : V^{\mathcal{T}} \to Q$ be an accepting run of the automaton $\mathcal{A}$ on this unravelling. We define the edge marking $\rho' : E \to Q'$ by setting, for every $(v, w) \in E$,

$$\rho'(v,w) = \big\{ \big(\rho(v'), \rho(w')\big) \in Q \times Q \mid (v', w') \in E^{\mathcal{T}} \text{ with } f(v') = v \text{ and } f(w') = w \big\}.$$

It is not difficult to check that ρ' defined in such a way is an accepting run of the automaton $\mathcal{B}$ on $\mathcal{G}, u$.

Conversely, assume $\mathcal{G}, u \in \mathcal{L}_E(\mathcal{B})$ and let $\rho' : E \to Q'$ be an accepting run of the E-automaton $\mathcal{B}$ on the graph $\mathcal{G}, u$. By induction on the length of paths in $V^{\mathcal{T}}$, and exploiting the memoryless determinacy of parity games [9], one can verify that there exists a vertex marking $\rho : V^{\mathcal{T}} \to Q$ such that $\mathcal{T}(\mathcal{G}, u), \rho \models \delta_0(u)$ and, for every path of the form πvw in $V^{\mathcal{T}}$, we have $(\rho(\pi v), \rho(\pi vw)) \in \rho'(v, w)$ and $\mathcal{T}(\mathcal{G}, u), \rho \models \delta(\pi vw)$. By definition of $\mathcal{B}$, it follows that ρ is an accepting run of automaton $\mathcal{A}$ on the unravelling $\mathcal{T}(\mathcal{G}, u)$. □

Corollary 1. *Every class of graphs definable in the counting μ-calculus is recognisable by an E-automaton.*

Proof. Let φ be a formula in the counting μ-calculus. By Theorem 2 there exists a V-automaton $\mathcal{A}_\varphi$ such that for every tree, $\mathcal{T} \in \mathcal{L}_V(\mathcal{A})$ if, and only if $\mathcal{T} \models \varphi$. Since φ is counting-bisimulation invariant, a graph $\mathcal{G}, u$ satisfies φ if, and only if, its unravelling $\mathcal{T}(\mathcal{G}, u)$ also satisfies φ, that is, if $\mathcal{T}(\mathcal{G}, u) \in \mathcal{L}_V(\mathcal{A})$. Now, the E-automaton $\mathcal{B}$ that simulates $\mathcal{A}$ according to Theorem 3 recognises the model class of φ, as $\mathcal{G}, u \in \mathcal{L}_E(\mathcal{B})$ if, and only if, $\mathcal{T}(\mathcal{G}, u) \in \mathcal{L}_V(\mathcal{A})$. □

4.2 Application to expressiveness

As a consequence of the above results, it follows that our ω-regular path conditions for E-automata are optimal in the sense that a more general model, where global acceptance conditions are given by MSO-formulae interpreted over the unravelling of locally consistent markings, would not be more expressive.

Proposition 2. *Consider an E-automaton $\mathcal{A} = (Q, \delta_0, \delta, Acc)$, and an* MSO*-formula φ over infinite trees with edges marked by Q. Then, there exists an E-automaton $\mathcal{A}_\varphi$ such that $\mathcal{G}, u \in \mathcal{A}_\varphi$ if, and only if, there is a run ρ of $\mathcal{A}$ on $\mathcal{G}, u$ such that $\mathcal{T}\big((\mathcal{G}, \rho), u\big) \models \varphi$.*

Proof. We refer to a generalisation of our automata model over input graphs where both edges and vertices are coloured. According to (straighforward generalisations of) Theorem 2 and 3, there exists an E-automaton $\mathcal{B}$ running on graphs $\mathcal{H}, u$ with Q-coloured edges such that, for every such input graph $\mathcal{T}(\mathcal{H}, u) \models \varphi$ if, and only if, $\mathcal{H}, u \in L_E(\mathcal{B})$. The desired automaton $\mathcal{A}_\varphi$ is then obtained by combining the automata $\mathcal{A}$ and $\mathcal{B}$ as a wreath product in which $\mathcal{B}$ reads runs of $\mathcal{A}$. □

4.3 Weakly uniform tree automata

A consequence of Gurevich and Shelah's Non-Uniformisation Theorem [11] for Monadic Second-Order Logic on the binary tree is that there exist MSO-definable languages that are not recognised by unambiguous tree-automata [28]. This negative result suggests that developing a notion of uniform recognisability [27] for MSO-definable languages of infinite trees could be very difficult. On the other hand, a success in achieving a notion of uniform recognisability may bear with itself many decision and classification results as those obtained, e.g., for languages of infinite words [21]. Our result on E-automata shows that at least a weak notion of uniformity is available without sacrificing expressiveness.

Definition 3. *Given a V-automaton $\mathcal{A}$ and a tree $\mathcal{T}$, we say that a run ρ of $\mathcal{A}$ on $\mathcal{T}$ is* uniform*, if any two isomorphic subtrees of $\mathcal{T}$ are marked by ρ in the same way. We say that a V-automaton $\mathcal{A}$ is* weakly uniform *if, for every tree $\mathcal{T} \in L_V(\mathcal{A})$, there exists an accepting run of $\mathcal{A}$ on $\mathcal{T}$ that is uniform.*

Observe that, in contrast to Thomas' notion of uniform recognisability, our assertion of weak uniformity is constrained to the particular input tree; runs over isomorphic subtrees of different trees may be different.

Theorem 4. *For every* MSO*-formula φ on trees, there exists a weakly uniform V-automaton that recognises the models of φ.*

Proof. Let $\mathcal{A}$ be a V-automaton equivalent to φ according to Theorem 2. Further, let $\mathcal{B} = (Q, \delta_0, \delta, Acc)$ be the E-automaton that simulates $\mathcal{A}$ according to Theorem 3, such that, for every graph, $\mathcal{G}, u \in \mathcal{L}_E(\mathcal{B})$ if, and only if, $\mathcal{T}(\mathcal{G}, u) \models \varphi$. For every graph $\mathcal{G}, u$ and every accepting run ρ of $\mathcal{B}$ on $\mathcal{G}, u$, let $\rho_{\mathcal{T}}$ be the marking induced by ρ on the unravelling $\mathcal{T}(\mathcal{G}, u)$. Since $\mathcal{L}_E(\mathcal{B})$ is closed under counting bisimulation, we can modify the transition specification δ, without modifying $\mathcal{L}_E(\mathcal{B})$, in such a way that, whenever ρ is an accepting run of $\mathcal{B}$ on $\mathcal{G}, u$, the unravelling $\rho_{\mathcal{T}}$ of the marking $(\mathcal{G}, \rho), u$ is also an accepting run of $\mathcal{B}$ on $\mathcal{T}(\mathcal{G}, u)$.

We conclude by proving that $\mathcal{B}$ modified in such a way is weakly uniform. Given any tree $\mathcal{T} \in L_E(\mathcal{B})$, consider the quotient $\mathcal{G}_{\mathcal{T}}, u_\varepsilon$ of $\mathcal{T}$ under counting bisimulation, with u_ε corresponding to the class of the root. Then there exists an accepting run ρ of $\mathcal{B}$ on $\mathcal{G}_{\mathcal{T}}, u_\varepsilon$ (since $\mathcal{T}$ and $\mathcal{G}_{\mathcal{T}}, u_\varepsilon$ are counting bisimilar). Now, one can observe that $\rho_{\mathcal{T}}$ is a uniform accepting run of $\mathcal{B}$ on $\mathcal{T}$. We obtain the desired V-automaton over trees, by pushing the state-marking of $\mathcal{B}$ from edges towards their target. □

References

[1] M. Ajtai and R. Fagin, *Reachability is harder for directed rather than undirected finite graphs*, Journal of Symbolic Logic, 55 (1990), pp. 113–150.

[2] A. Arnold and D. Niwiński, *Rudiments of mu-calculus*, vol. 146 of Studies in Logic and the Foundations of Mathematics, North-Holland, 2001.

[3] O. Bernholtz and O. Grumberg, *Branching time temporal logic and amorphous tree automata*, in Conf. on Concurrency Theory (CONCUR), vol. 715 of LNCS, Springer-Verlag, 1993, pp. 262–277.

[4] B. Courcelle, *The monadic second order logic of graph I: Recognizable sets of finite graphs*, Inf. and Comp., 85 (1990), pp. 12–75.

[5] ———, *The monadic second-order logic of graphs VI: On several representations of graphs by logical structures*, Discrete Applied Mathematics, 54 (1994), pp. 117–149.

[6] B. Courcelle and P. Weil, *The recognizability of sets of graphs is a robust property*, Theoretical Computer Science, 342 (2005), pp. 173–228.

[7] V. Diekert, *The Book of Traces*, World Scientific Publishing Co., Inc., 1995.

[8] M. Droste, P. Gastin, and D. Kuske, *Asynchronous cellular automata for pomsets*, Theoretical Computer Science, 247 (2000), pp. 1–38.

[9] E. A. Emerson and C. S. Jutla, *Tree automata, mu-calculus and determinacy*, in IEEE Symp. on Logic in Computer Science (LICS), 1991, pp. 368–377.
[10] D. Giammaresi and A. Restivo, *Two-dimensional languages*, in Handbook of Formal Languages, G. Rozenberg and A. Salomaa, eds., vol. III, Springer-Verlag, 1997, pp. 215–268.
[11] Y. Gurevich and S. Shelah, *Rabin's uniformization problem*, J. Symb. Log., 48 (1983), pp. 1105–1119.
[12] D. Janin and G. Lenzi, *Relating levels of the mu-calculus hierarchy and levels of the monadic hierarchy*, in IEEE Symp. on Logic in Computer Science (LICS), IEEE Computer Society, 2001, pp. 347–356.
[13] D. Janin and I. Walukiewicz, *Automata for the modal mu-calculus and related results*, in Mathematical Found. of Comp. Science (MFCS), vol. 969 of LNCS, Springer-Verlag, 1995.
[14] ———, *On the expressive completeness of the modal mu-calculus with respect to monadic second order logic*, in Conf. on Concurrency Theory (CONCUR), vol. 1119 of LNCS, Springer-Verlag, 1996, pp. 263–277.
[15] T. Kamimura and G. Slutzki, *Parallel and two-way automata on directed ordered acyclic graphs*, Information and Control, 49 (1981), pp. 10–51.
[16] D. Kozen, *Results on the propositional μ-calculus*, Theoretical Comp. Science, 27 (1983), pp. 333–354.
[17] K. Lodaya and P. Weil, *Series-parallel languages and the bounded-width property*, Theoretical Computer Science, 237 (2000), pp. 347–380.
[18] O. Matz and W. Thomas, *The monadic quantifier alternation hierarchy over finite graphs is infinite*, in IEEE Symp. on Logic in Computer Science (LICS), 1997, pp. 236–244.
[19] A. A. Muchnik, *Games on infinite trees and automata with dead-ends: a new proof for the decidability of the monadic second order theory of two successors*, *Bull.* EATCS, 42 (1992), pp. 220–267. (traduction d'un article en Russe de 1984).
[20] D. Niwiński, *Fixed points vs. infinite generation*, in IEEE Symp. on Logic in Computer Science (LICS), 1988, pp. 402–409.
[21] D. Perrin and J.-E. Pin, *Infinite Words*, Academic Press, 2002.
[22] A. Potthoff, S. Seibert, and W. Thomas, *Nondeterminism versus determinism of finite automata over directed acyclic graphs*, Bull. Belg. Math. Soc. Simon Stevin, 1 (1994), pp. 285–298.
[23] M. O. Rabin, *Decidability of second order theories and automata on infinite trees*, Trans. Amer. Math. Soc., 141 (1969), pp. 1–35.
[24] J. W. Thatcher and J. B. Wright, *Generalized finite automata theory with an application to a decision problem of second–order logic*, Math. Systems Theory, 2 (1968), pp. 57–81.
[25] W. Thomas, *On logics, tilings, and automata*, in Proceedings of the 18th international colloquium on Automata, languages and programming, New York, NY, USA, 1991, Springer-Verlag New York, Inc., pp. 441–454.
[26] ———, *Automata theory on trees and partial orders*, in TAPSOFT'97, M. D. M. Bidoit, ed., no. 1214 in LNCS, Springer-Verlag, 1997, pp. 20–38.
[27] ———, *Uniform and nonuniform recognizability.*, Theor. Comput. Sci., 292 (2003), pp. 299–316.
[28] I. Walukiewicz, *On the ambiguity problem*, 1994. Notes non publiées.
[29] ———, *Monadic second order logic on tree-like structures*, Theoretical Comp. Science, 275 (2002), pp. 311–346.