Relating paths in transition systems: the fall of the modal mu-calculus

CATALIN DIMA, Université Paris-Est, France
BASTIEN MAUBERT, Università degli Studi di Napoli "Federico II", Italy
SOPHIE PINCHINAT, Université de Rennes 1, France

We revisit Janin and Walukiewicz's classic result on the expressive completeness of the modal mu-calculus with respect to Monadic Second Order Logic (MSO), which is that the mu-calculus corresponds precisely to the fragment of MSO that is invariant under bisimulation. We show that adding binary relations over finite paths in the picture may alter the situation. We consider a general setting where finite paths of transition systems are linked by means of a fixed binary relation. This setting gives rise to natural extensions of MSO and the mu-calculus, that we call the *MSO with paths relation* and the *jumping mu-calculus*, the expressivities of which we aim at comparing. We first show that "bounded-memory" binary relations bring about no additional expressivity to either of the two logics, and thus preserve expressive completeness. In contrast, we show that for a natural, classic "infinite-memory" binary relation stemming from games with imperfect information, the existence of a winning strategy in such games, though expressible in the bisimulation-invariant fragment of MSO with paths relation, cannot be expressed in the jumping mu-calculus. Expressive completeness thus fails for this relation. These results crucially rely on our observation that the jumping mu-calculus has a tree automata counterpart: the *jumping tree automata*, hence the name of the jumping mu-calculus. We also prove that for *observable* winning conditions, the existence of winning strategies in games with imperfect information is expressible in the jumping mu-calculus. Finally, we derive from our main theorem that jumping automata cannot be projected, and ATL with imperfect information does not admit expansion laws.

CCS Concepts: • **Theory of computation → Logic and verification**; **Modal and temporal logics**; *Automata over infinite objects*; *Automata extensions*;

Additional Key Words and Phrases: Expressiveness, imperfect information, monadic second order logic, mu-calculus, transition systems

ACM Reference Format:
Catalin Dima, Bastien Maubert, and Sophie Pinchinat. 2017. Relating paths in transition systems: the fall of the modal mu-calculus. *ACM Trans. Comput. Logic* 1, 1, Article 1 (January 2017), 33 pages. https://doi.org/0000001.0000001

1 INTRODUCTION

Monadic Second-Order Logic (MSO) is considered as a standard for comparing expressiveness of logics of programs. Ground-breaking results concerning expressiveness and decidability of MSO on infinite graphs were obtained first on "freely-generated" structures (words, trees, tree-like structures, etc.) [49, 53], then on "non-free" structures like grids [33] or infinite graphs generated by regularity-preserving transformations [11, 15]. In all the above settings, the syntax of MSO utilises

Authors' addresses: Catalin Dima, Université Paris-Est, Laboratoire d'Algorithmique, Complexité et Logique (LACL), Créteil, France, dima@u-pec.fr; Bastien Maubert, Università degli Studi di Napoli "Federico II", DIETI, Naples, Italy, bastien.maubert@gmail.com; Sophie Pinchinat, Université de Rennes 1, IRISA, Rennes, France, sophie.pinchinat@irisa.fr.

1529-3785/2017/1-ART1 $15.00
https://doi.org/0000001.0000001

one or more binary relation symbols which are interpreted using the binary edge relations of the graph structure. Additionally, much attention has been brought to the study of enrichments of MSO with unary predicate symbols or with the "equal level" binary predicate (MSO^{eql}) [19, 31, 48].

For many of these settings, MSO has been compared with automata and with modal logics. Standard results on trees are Rabin's expressiveness equivalence between MSO with two successors and automata on binary trees [40], and Janin and Walukiewicz's result [27] showing that the bisimulation-invariant fragment of MSO on transition systems coincides with the μ-calculus. Notable exceptions to the classical trilogy between MSO, modal logics and automata are MSO on infinite partial orders – where only partial results are known [7, 14, 43] – and MSO^{eql}– where, similarly, only partial results are known [48].

More recently, there has been an increased interest in the expressiveness and decidability of logics defined on structures in which two "orthogonal" relations are considered: the so-called temporal epistemic (multi-agent) logics [21], which combine time-passage relations and epistemic relations on the histories of the system. Time-passage relations classically represent the evolution of the system, while each epistemic relation captures some agent's partial observation of the system by relating indistinguishable histories. They allow to reason about what these agents know about the state of the system along its executions. We may identify now an important sub-domain in verification which is concerned with the expressivity, decidability and axiomatisability of logics of knowledge and time [8, 21, 24, 25, 32, 51].

A natural question that arises regarding logics of agents that combine time and knowledge is whether they admit a trilogy similar to the classic one: do there exist natural extensions of MSO, of the μ-calculus, and of tree automata for the temporal epistemic framework, and how would they compare? To the best of our knowledge, these questions remain open. Only partial results exist on relations between some extensions of MSO, μ-calculus, tree automata and other logics of knowledge and time [35, 45, 46, 51].

The first observation is that appropriate extensions of MSO, of the μ-calculus and of tree automata would rely on two sorts of binary relations: those related to the behaviour of the system and those related to epistemic features. While the temporal part of these logics naturally refers to a tree-like structure, the epistemic part requires considering binary relations defined on histories, in order to model powerful agents that remember the whole past. The models of such an extension of MSO neither are tree-like structures, nor grid-like structures, nor graphs within the Caucal hierarchy. The proposals in this direction that we know about are the following. In [51], an encoding of LTL with knowledge into Chain Logic with equal-level predicate, which is a fragment of MSO^{eql}, is mentioned. The epistemic μ-calculus is introduced and its model-checking problem studied in [45, 46]. An extension of this epistemic μ-calculus is studied in [1], and [35] proposed a generalisation of tree automata, called jumping tree automata, which is suited to the study of temporal epistemic logics.

In this work, which is an extended version of [17], we develop a general setting in which models are unfoldings of transition systems together with a binary relation over their finite executions, also called paths or histories. Such relations are called *path relations*, and their definition is general enough to capture all indistinguishability relations considered in temporal epistemic logics, and more. We propose extensions of MSO and of the μ-calculus, respectively called the *Monadic Second Order Logic with path relation* and the *jumping μ-calculus*. MSO with path relation is an extension of MSO interpreted over unfoldings of transition systems equipped with a path relation. The syntax is that of MSO on graphs with an additional binary relation symbol $\rightarrowtail$, interpreted "transversely" on the tree-unfolding according to the binary relation over paths that equips the system. The jumping μ-calculus is a generalisation of the epistemic μ-calculus defined in [45, 46]: it is also evaluated on tree-unfoldings of transition systems, and it features a *jumping modality* $\boxminus$ whose semantics relies on the path relation that equips the system. In case the path relation is seen as modelling histories' indistinguishability for some agent, this modality coincides with the classic knowledge operator K for this agent (see, *e.g.*, [21]). As in the classic setting of [27], definability in the jumping μ-calculus entails definability in MSO with path relation. It is the converse statement that we explore, that is the *expressive completeness* of jumping μ-calculus with regards to (the bisimulation invariant fragment of) MSO with path relation.

We first show that the *jumping tree automata* recently defined in [35] are equivalent to the jumping μ-calculus, and the two-way translation does not depend on the *a priori* fixed path relation. We then address, like in [27], the question whether for bisimulation-closed classes of models, definability in MSO with path relation implies definability in the jumping μ-calculus. As it turns out, the answer to this question depends on the properties of the path relation one considers.

First, we consider the class of *recognisable* relations [3], which typically capture agents with bounded memory. We establish that when the path relation is recognisable, the jumping μ-calculus is expressively complete with regards to the bisimulation-invariant fragment of MSO with path relation. Indeed, recognisable relations being MSO-definable, both our extensions of MSO and the μ-calculus with such relations collapse to the classic MSO and μ-calculus, respectively. Concerning transition systems with bounded branching degree, we obtain that the jumping μ-calculus with recognisable path relation is at most exponentially more succinct than the μ-calculus, while its satisfiability problem is Exptime-complete. These results rely on the effective translation of jumping tree automata equipped with recognisable path relations into two-way tree automata [35].

We then show that considering more powerful path relations can break this expressive completeness, even for *regular relations* (*i.e.*, relations accepted by synchronous finite transducers [3]). As witness of this phenomenon we consider the so-called *synchronous perfect recall* relation over paths, which is central in logics of knowledge and time as well as in games with imperfect information (see, *e.g.*, [21, 41, 45, 46]), and is accepted by a very simple synchronous transducer with only one state. We establish that for synchronous perfect recall, the jumping μ-calculus is not expressively complete with regards to MSO with path relation. To achieve this, we prove that the class of two-player reachability games with imperfect information and synchronous perfect recall (with a fixed number of observations and actions) where the first player wins, cannot be defined in the jumping μ-calculus, but is closed under bisimulation and is definable in our extension of MSO. The proof heavily relies on the equivalence between the jumping μ-calculus and jumping automata: assuming towards a contradiction that there is a jumping automaton that accepts unfoldings of game arenas in which the first player has a winning strategy with imperfect information, we exhibit a family of such unfoldings accepted by the automaton and exploit the "pigeon-hole principle" to show that this automaton also accepts the unfolding of an arena in which the first player does not win. We point out that the proof makes use of unobservable winning conditions: indeed, we also prove that if winning conditions are assumed to be observable, and if the first player remembers her actions, then the class of imperfect-information (either reachability or parity) games where the first player wins is definable in the jumping μ-calculus.

The fact that, when relating paths with some relations such as the one representing synchronous perfect recall, the mu-calculus is no longer as expressive as bisimulation-invariant MSO, motivates the slightly outrageous title of this article. From this expressive incompleteness result, we also derive a number of corollaries concerning jumping tree automata and strategic logics. First, we obtain an argument to show that the class of jumping tree automata is not closed under projection. The second impact concerns logics of coalitions and strategies, and in particular the comparison of Alternating-time Temporal Logic (ATL) with fixed point logics. It is folklore that when agents have perfect information, ATL is subsumed by the bisimulation-invariant fragment of MSO, and thus by the μ-calculus. In addition, and in connection with the latter, ATL with perfect information admits expansion laws, or fixed-point axioms, for combinations of strategic and temporal operators, and can thus be translated into the Alternating-time μ-Calculus (AMC) of [2]. For imperfect information, the situation is very different: it is proved in [9] that the alternation-free fragment of Alternating-time μ-Calculus with epistemic operators and imperfect information ($\mathsf{AEMC_i}$) does not capture ATL with imperfect information ($\mathsf{ATL_i}$) for memoryless semantics, which implies the absence of simple expansion laws for $\mathsf{ATL_i}$ without memory. They also proved that $\mathsf{ATL_i}$ with synchronous perfect recall is not captured by the *alternation-free fragment* of $\mathsf{AEMC_i}$ *without memory*. As a corollary of our main theorem we can strengthen the latter result. We obtain that $\mathsf{ATL_i}$ with synchronous perfect recall is not captured by the *full* $\mathsf{AEMC_i}$ *with synchronous perfect recall*, thus implying the absence even of complex expansion laws/fixed-point axioms for $\mathsf{ATL_i}$ with synchronous perfect recall. Interestingly, the simplicity of the $\mathsf{ATL_i}$ formula and models considered in the proof of Theorem 5.7 makes the latter result very robust to variations in the semantics of the strategic operator.

The paper runs as follows: in Section 2 we develop the framework of our study. In Section 3 we introduce the two logics we consider, MSO with path relation and the jumping μ-calculus, and we state the *expressive completeness problem* as well as our main results on the matter, theorems 3.6 and 3.7. In Section 4 we first prove Theorem 3.6; we then introduce the notion of jumping tree automata, and we show them to be equivalent to the jumping μ-calculus. Thanks to this equivalence we then establish succinctness and complexity results for the jumping μ-calculus with recognisable path relation. The equivalence between jumping tree automata and the jumping μ-calculus is also crucial in the proof of Theorem 3.7, which we present in Section 5. There, we first introduce two-player games with imperfect information and synchronous perfect recall. Then we show that when winning conditions are observable and actions are remembered, the classes of reachability and parity games where the first player wins are definable in the jumping μ-calculus (Theorem 5.2). We finally prove that definability is lost when the assumption that winning conditions are observable is relaxed (Theorem 5.7). In Section 6 we establish several results that follow from the latter theorem: first, that jumping tree automata are not closed under projection, and second, that $\mathsf{ATL_i}$ with synchronous perfect recall does not admit expansion laws for the strategic operators. We conclude and give some perspectives in Section 7.

2 PRELIMINARY NOTIONS

We first fix a few basic notations. Given two words w and w' over some alphabet Σ, we write $w \preceq w'$ if w is a prefix of w'; if $w = a_0a_1\ldots \in \Sigma^\omega$ is an infinite word we let, for each $i \geq 0$, $w[i] := a_i$ and $w[0,i] := a_0a_1\ldots a_i$. For a finite word $w = a_0 \ldots a_{n-1} \in \Sigma^*$, its *length* is $|w| := n$. Also, given a binary relation $R \subseteq A \times B$ between two sets A and B, for every element $a \in A$, we let $R(a) := \{b \mid (a,b) \in R\}$.

In the rest of the paper, we fix $\mathcal{AP} = \{p, p', \ldots\}$ a countable set of *atomic propositions* and $\mathcal{A}ct = \{a, a', \ldots\}$ a countable set of *actions*.

We now introduce transition systems and their unfoldings.

Definition 2.1. A *transition system* is a structure $\mathcal{S} = (S, s_\iota, \{a^\mathcal{S}\}_{a\in\mathcal{A}ct}, \{p^\mathcal{S}\}_{p\in\mathcal{AP}})$, where S is a (possibly infinite) set of *states*, s_ι is an *initial state*, each $a^\mathcal{S}$ is a binary relation over S and each $p^\mathcal{S}$ is a subset of S. We let $s \xrightarrow{a} s'$ stand for $(s,s') \in a^\mathcal{S}$.

Because the logics that we aim at defining are concerned with *paths* in transition systems, *i.e.*, finite sequences of states and actions that start in the initial state and follow the binary relations, these logics cannot be evaluated on the transition systems themselves, but paths must be made accessible first. We do so by considering their tree-unfoldings, that we define now after basic definitions for trees.

Trees. A *tree* is a nonempty, prefix-closed set $\tau \subseteq \mathbb{N}^*$. An element $x \in \tau$ is a *node*, and the empty word ϵ is the *root* of the tree. If $x \cdot i \in \tau$ then $x \cdot i$ is a *child* of x. A node with no child is a *leaf*. We call *branch* a sequence of nodes in τ (either finite or infinite) in which each node (except the first one) is a child of the previous one; a branch is *maximal* if it is infinite, or if it ends up in a leaf. A node y is a *descendant* of a node x if x is a prefix of y (*i.e.*, $x \preceq y$), or equivalently if y can be found along some branch that starts in x. We let $[\tau]_x$ denote the subtree of τ rooted in x: formally, $[\tau]_x := \{y \mid x \preceq y\}$.

A *marked tree* is a pair $t = (\tau, m)$, where τ is a tree and $m : \tau \to (\mathcal{A}ct \times 2^{\mathcal{AP}})$ is a *marking* of the nodes. The intuitive meaning of $m(x) = (a, \ell)$ is that x was reached through action a, and that the set of atomic propositions that hold in x is ℓ. We say that a is the action of x, and ℓ its label. Note that the action of the root is meaningless. We say that y is an *a-child* of a node x if y is a child of x and its action is a, *i.e.*, if $m(y) = (a, \ell)$ for some label ℓ. Given a node x with marking $m(x) = (a, \ell)$, we write $a^x := a$ for its action and $\ell^x := \ell$ for its label.

The *word* of a node x is $w(x) := m(\epsilon)m(x_1)\ldots m(x_n)$, where $\epsilon\, x_1 \ldots x_n (= x)$ is the (unique) branch from the root to x. For a finite subset $AP \subset \mathcal{AP}$, an *AP-tree* is a marked tree $t = (\tau, m)$ such that for every node $x \in \tau$, it holds that $\ell^x \subseteq AP$. For a marked tree $t = (\tau, m)$ and a node $x \in \tau$, we let $[t]_x := ([\tau]_x, m')$, where m' is the restriction of m to $[\tau]_x$. We may write $x \in t$ instead of $x \in \tau$.

For convenience and readability, our trees have (at most) countable branching degree. We point out that, unless otherwise stated, our results still hold for arbitrary branching degree. Specifically, Propositions 4.4 and 4.5 assume bounded branching degree, and Theorem 5.2 considers games with finite branching degree.

We now define unfoldings of transition systems.

Definition 2.2 (Unfoldings). Let $\mathcal{S} = (S, s_\iota, \{a^{\mathcal{S}}\}_{a \in \mathcal{A}ct}, \{p^{\mathcal{S}}\}_{p \in \mathcal{AP}})$ be a transition system. The *unfolding* $t_{\mathcal{S}}$ of $\mathcal{S}$ is the marked tree (τ, m) with least tree τ such that:

(1) ϵ is associated[1] to s_ι, and $\ell^\epsilon = \{p \mid s_\iota \in p^{\mathcal{S}}\}$, and a^ϵ is any action;
(2) for each node $x \in \tau$ associated to state s, letting $\langle s \xrightarrow{a_i} s_i \rangle_{i \in I}$ be an enumeration of the outgoing transitions from s (with $I \subseteq \mathbb{N}$), for each $i \in I$ we have
  - $x \cdot i \in \tau$,
  - $x \cdot i$ is associated to s_i, and
  - $m(x \cdot i) = (a_i, \{p \in \mathcal{AP} \mid s_i \in p^{\mathcal{S}}\})$.

Note that the choice of the enumeration in Point 2 of Definition 2.2 is irrelevant to the logics we consider. Also, every marked tree can be seen as a transition system and is its own unfolding. Therefore in the following we could equivalently forget about transition systems and consider only trees, or keep seeing trees as unfoldings of transition systems. We choose the latter in order to emphasise the particularity of the logics that we introduce, *i.e.*, the necessity to unfold to capture paths. Indeed, paths in a transition system $\mathcal{S}$ are in one-to-one correspondence with the nodes of the marked tree $t_{\mathcal{S}}$.

Let us now define the means by which paths of a transition system are related.

Definition 2.3. A *path relation* is a binary relation over $(\mathcal{A}ct \times 2^{\mathcal{AP}})^*$.

A path relation links finite paths of transition systems over $\mathcal{AP}$ and $\mathcal{A}ct$. It also induces a binary relation between nodes of marked trees (over $\mathcal{AP}$ and $\mathcal{A}ct$) in a natural way by relating nodes x and y whenever their words $w(x)$ and $w(y)$ are related. We use typical element $\rightarrowtail$ for path relations; this symbol emphasises the "transversal" connection of nodes in the trees.

[1]We leave this notion of "associated state" informal, as it is only useful to define the unfolding. We could add a third component in labeled trees that makes formal this association, and forget it after the construction.

Example 2.4. A typical example of path relations are indistinguishability relations used in temporal epistemic logics (see, *e.g.*, [21, 24, 25]). Consider an agent who observes the evolution of a transition system. Now assume that this agent has only a partial, imperfect observation of this system. The agent's observational power is usually defined by two parameters:

- how she observes a given state of the system, and
- how much she remembers of what she observes along executions of the system.

For the first point, the agent may for instance only have access to the truth value of a subset $Pub \subset \mathcal{AP}$ of "public" atomic propositions, the others being "private". For the second point, classic assumptions are that the agent does not remember anything, in which case we talk about a *memoryless agent*, or that she remembers everything, in which case we talk about an *agent with perfect recall*. These two parameters together induce an *indistinguishability relation*, or *epistemic relation*, that relates two paths of the system if the agent cannot tell the difference between them. In the example of an agent with perfect recall who can only observe a subset *Pub* of public propositions, this relation is the path relation such that two words over $\mathcal{A}ct \times 2^{\mathcal{AP}}$ are related if their projections on $\mathcal{A}ct \times 2^{Pub}$ are equal. If instead we consider a memoryless agent, then two words are related if the projections of *the last letter* of each word are the same. Note that in this example the agent has a perfect observation of the actions that take place in the system.

We end up this section by recalling the classic notion of *bisimulation* [36, 38, 50] between transition systems.

Definition 2.5. Let $\mathcal{S}$ and $\mathcal{S}'$ be two transition systems. A *bisimulation* between $\mathcal{S}$ and $\mathcal{S}'$ is a binary relation $\mathcal{Z} \subseteq S \times S'$ such that, for all $(s, s') \in \mathcal{Z}$:

(1) for all $p \in \mathcal{AP}$, $s \in p^{\mathcal{S}}$ iff $s' \in p^{\mathcal{S}'}$;
(2) for all $a \in \mathcal{A}ct$, and for all $r \in a^{\mathcal{S}}(s)$, there is $r' \in a^{\mathcal{S}'}(s')$ such that $(r, r') \in \mathcal{Z}$;
(3) and vice-versa.

We say that $\mathcal{S}$ and $\mathcal{S}'$ are *bisimilar*, written $\mathcal{S} \leftrightarroweq \mathcal{S}'$, if there is a bisimulation $\mathcal{Z}$ between $\mathcal{S}$ and $\mathcal{S}'$ such that $(s_\iota, s'_\iota) \in \mathcal{Z}$. A class C of transition systems is *bisimulation closed* if $\mathcal{S} \in C$ and $\mathcal{S} \leftrightarroweq \mathcal{S}'$ imply $\mathcal{S}' \in C$, for all $\mathcal{S}$ and $\mathcal{S}'$.

3 EXPRESSIVE COMPLETENESS ISSUES

In the rest of the paper, additionally to the sets $\mathcal{AP}$ and $\mathcal{A}ct$ defined earlier, we fix a countable set of *second order variables* $Var = \{X, Y, \dots\}$. A *valuation* for a marked tree $t = (\tau, m)$ is a mapping $V : Var \to 2^{\tau}$. Classically, given $X \in Var$ and $T \subseteq t$, we let $V[T/X]$ be the valuation that maps X to T, and which coincides with V on all other variables: $V[T/X](Y) = T$ if $Y = X$, and $V(Y)$ otherwise.

3.1 Monadic second order logic with path relation

We define the logic MSO$^{\rightarrowtail}$ as an extension of the Monadic Second Order Logic (MSO) interpreted over transition systems with a path relation.

Formulas of MSO$^{\rightarrowtail}$ are defined inductively by the following grammar:

$$(\text{MSO}^{\rightarrowtail} \ni)\, \psi ::= \quad sr(X) \mid p(X) \mid a(X, Y) \mid X \subseteq Y \mid \neg\psi \mid \psi \vee \psi' \mid \exists X.\psi(X) \mid \; X \rightarrowtail Y$$

where $p \in \mathcal{AP}$ and $X, Y \in Var$. The syntactic fragment of the logic which does not use predicate $\rightarrowtail$ is MSO.

Given a path relation $\rightarrowtail$, an MSO$^{\rightarrowtail}$ formula ψ is interpreted over a marked tree $t = (\tau, m)$ with a valuation $V : Var \to 2^{\tau}$. We write $t, V \models^{\rightarrowtail} \psi$ to mean that ψ holds in marked tree t for valuation V, which is defined inductively as follows:

$$
\begin{array}{lll}
t, V \models^{\looparrowright} sr(X) & \text{if} & V(X) = \{\epsilon\} \\
t, V \models^{\looparrowright} p(X) & \text{if} & \text{for all } x \in V(X), p \in \ell^x \\
t, V \models^{\looparrowright} a(X, Y) & \text{if} & V(X) = \{x\}, V(Y) = \{y\}, \text{ and } y \text{ is an } a\text{-child of } x \\
t, V \models^{\looparrowright} X \subseteq Y & \text{if} & V(X) \subseteq V(Y) \\
t, V \models^{\looparrowright} \neg\psi & \text{if} & t, V \not\models^{\looparrowright} \psi \\
t, V \models^{\looparrowright} \psi \vee \psi' & \text{if} & t, V \models^{\looparrowright} \psi \text{ or } t, V \models^{\looparrowright} \psi' \\
t, V \models^{\looparrowright} \exists X.\psi(X) & \text{if} & \text{there is } T \subseteq t \text{ such that } t, V[T/X] \models^{\looparrowright} \psi(X) \\
t, V \models^{\looparrowright} X \looparrowright Y & \text{if} & V(X) = \{x\}, V(Y) = \{y\}, \text{and } x \looparrowright y
\end{array}
$$

If $\psi \in \mathsf{MSO}^{\looparrowright}$ is a sentence (*i.e.*, it has no free variable), its semantics is independent of the valuation. We then simply write $t \models^{\looparrowright} \psi$. For a sentence $\psi \in \mathsf{MSO}^{\looparrowright}$, a path relation $\looparrowright$ and a transition system $\mathcal{S}$, we write $\mathcal{S} \models^{\looparrowright} \psi$ whenever $t_{\mathcal{S}} \models^{\looparrowright} \psi$.

We let $\mathcal{L}(\psi, \looparrowright) := \{\mathcal{S} \mid \mathcal{S} \models^{\looparrowright} \psi\}$ be the set of models of ψ.

3.2 The jumping μ-calculus

We now define the jumping μ-calculus, whose name comes from their automata counterpart, the jumping tree automata (see Proposition 4.2). The term "jumping" indicates the fact that these automata, in addition to sending copies of themselves to children of the current node in their input tree as classic tree automata do, can also perform transversal "jumps" in the input tree, sending copies to nodes related to the current one by some fixed path relation.

The syntax of the jumping μ-calculus ($\mathsf{L}_\mu^{\looparrowright}$) is defined by the following grammar:

$$(\mathsf{L}_\mu^{\looparrowright} \ni) \, \varphi ::= X \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle a \rangle\varphi \mid \langle\looparrowright\rangle\varphi \mid \mu X.\varphi(X)$$

where $X \in \mathit{Var}$, $p \in \mathcal{AP}$, $a \in \mathcal{Act}$ and in the last rule, X appears only positively (under an even number of negations) in $\varphi(X)$. We classically define the dual operators $\wedge$, $[a]$, $[\looparrowright]$ and ν as follows: $\varphi \wedge \varphi' := \neg(\neg\varphi \vee \neg\varphi')$, $[a]\varphi := \neg\langle a \rangle\neg\varphi$, $[\looparrowright]\varphi := \neg\langle\looparrowright\rangle\neg\varphi$ and $\nu X.\varphi := \neg\mu X.\neg\varphi[X \leftarrow \neg X]$.

Given a path relation $\looparrowright$, a formula of $\mathsf{L}_\mu^{\looparrowright}$ is interpreted over a marked tree $t = (\tau, m)$ with a valuation $V : \mathit{Var} \longrightarrow 2^\tau$. For $\varphi \in \mathsf{L}_\mu^{\looparrowright}$, we inductively define $[\![\varphi]\!]_{\looparrowright}^{t,V} \subseteq \tau$ as follows:

$$
\begin{array}{lll}
[\![X]\!]_{\looparrowright}^{t,V} & := & V(X) \\
[\![p]\!]_{\looparrowright}^{t,V} & := & \{x \in t \mid p \in \ell^x\} \\
[\![\neg\varphi]\!]_{\looparrowright}^{t,V} & := & t \setminus [\![\varphi]\!]_{\looparrowright}^{t,V} \\
[\![\varphi \vee \varphi']\!]_{\looparrowright}^{t,V} & := & [\![\varphi]\!]_{\looparrowright}^{t,V} \cup [\![\varphi']\!]_{\looparrowright}^{t,V} \\
[\![\langle a \rangle\varphi]\!]_{\looparrowright}^{t,V} & := & \{x \in t \mid x \text{ has an } a\text{-child in } [\![\varphi]\!]_{\looparrowright}^{t,V}\} \\
[\![\langle\looparrowright\rangle\varphi]\!]_{\looparrowright}^{t,V} & := & \{x \in t \mid \text{ there exists } y \in [\![\varphi]\!]_{\looparrowright}^{t,V} \text{ such that } x \looparrowright y\} \\
[\![\mu X.\varphi(X)]\!]_{\looparrowright}^{t,V} & := & \bigcap\{T \subseteq t \mid [\![\varphi(X)]\!]_{\looparrowright}^{t,V[T/X]} \subseteq T\}
\end{array}
$$

Note that requiring that X appears only positively in $\varphi(X)$ for each formula $\mu X.\varphi(X)$ in $\mathsf{L}_\mu^{\looparrowright}$ yields a monotone function $T \mapsto [\![\varphi(X)]\!]_{\looparrowright}^{t,V[T/X]}$, which hence has a least fixpoint, namely $[\![\mu X.\varphi(X)]\!]_{\looparrowright}^{t,V}$.

If $\varphi \in \mathsf{L}_\mu^{\looparrowright}$ is a sentence (*i.e.*, it has no free variable), its semantics is independent on the valuation, such that we may omit it from the semantics. For a sentence $\varphi \in \mathsf{L}_\mu^{\looparrowright}$, a path relation $\looparrowright$ and a marked tree t, we write $t \models^{\rightarrowtail} \varphi$ whenever $\epsilon \in [\![\varphi]\!]^{t}_{\rightarrowtail}$, and for a transition system $\mathcal{S}$, we write $\mathcal{S} \models^{\rightarrowtail} \varphi$ if $t_{\mathcal{S}} \models^{\rightarrowtail} \varphi$. We let $\mathcal{L}(\varphi, \rightarrowtail) := \{\mathcal{S} \mid \mathcal{S} \models^{\rightarrowtail} \varphi\}$ be the set of models of φ.

Finally, we let L_μ denote the sublanguage of $\mathsf{L}_\mu^{\rightarrowtail}$ obtained by removing the modality $\Diamond$: as relation $\rightarrowtail$ then becomes superfluous, for $\varphi \in \mathsf{L}_\mu$, we let $\mathcal{L}(\varphi) := \{\mathcal{S} \mid t_{\mathcal{S}} \models \varphi\}$.

3.3 Expressive Completeness

Whichever logic one considers, say $\mathcal{L}$, a class C of transition systems is *$\mathcal{L}$-definable* if there is a formula of $\mathcal{L}$ whose set of models is exactly C.

Proposition 3.1. *For every path relation $\rightarrowtail$, every $L_\mu^{\rightarrowtail}$-definable class is closed under bisimulation.*

To establish this result, it is enough to prove Lemma 3.2 below, which states that whenever two transition systems over $\mathcal{A}ct$ are bisimilar, their unfoldings enriched with a given path relation are also bisimilar as transition systems over $\mathcal{A}ct' := \mathcal{A}ct \cup \{a_{\rightarrowtail}\}$. Indeed, Lemma 3.2 implies that the unfoldings of two bisimilar transition systems satisfy the same $\mathsf{L}_\mu^{\rightarrowtail}$-formulas seen as μ-calculus formulas over $\mathcal{A}ct'$.

Let $\rightarrowtail$ be a path relation. Given a marked tree t, we define the enriched transition system $t^{\rightarrowtail}$ over $\mathcal{AP}$ and $\mathcal{A}ct' := \mathcal{A}ct \cup \{a_{\rightarrowtail}\}$, where $a_{\rightarrowtail}$ is a fresh action symbol, by letting $(x, y) \in a_{\rightarrowtail}^{t^{\rightarrowtail}}$ whenever $w(x) \rightarrowtail w(y)$.

Lemma 3.2. *Let $\mathcal{S}$ and $\mathcal{S}'$ be two transition systems and $\rightarrowtail$ a path relation. If $\mathcal{S} \underline{\leftrightarrow} \mathcal{S}'$, then $t_{\mathcal{S}}^{\rightarrowtail} \underline{\leftrightarrow} t_{\mathcal{S}'}^{\rightarrowtail}$.*

Proof. Since $\mathcal{S}$ and $\mathcal{S}'$ are bisimilar, so are their unfoldings $t_{\mathcal{S}}$ and $t_{\mathcal{S}'}$ (seen as transition systems). More precisely, given a bisimulation relation $\mathcal{Z}$ between $\mathcal{S}$ and $\mathcal{S}'$, we describe how to define a relation $\mathcal{Z}_1$ that is a bisimulation between $t_{\mathcal{S}}$ and $t_{\mathcal{S}'}$, and that only relates nodes that have the same word.

First, let the roots of $t_{\mathcal{S}}$ and $t_{\mathcal{S}'}$ be related by $\mathcal{Z}_1$. Then, for two non-root nodes $x \in t_{\mathcal{S}}$ and $x' \in t_{\mathcal{S}'}$, we let $(x, x') \in \mathcal{Z}_1$ if their parents are related by $\mathcal{Z}_1$, their actions a^x and $a^{x'}$ are the same, and their associated states[2] are related by $\mathcal{Z}$. An easy proof by induction on the nodes' depth shows that nodes related by $\mathcal{Z}_1$ indeed have the same word. It is also easy to check that $\mathcal{Z}_1$ is a bisimulation between $t_{\mathcal{S}}$ and $t_{\mathcal{S}'}$.

Now let us check that $\mathcal{Z}_1$ is also a bisimulation between the enriched structures $t_{\mathcal{S}}^{\rightarrowtail}$ and $t_{\mathcal{S}'}^{\rightarrowtail}$ seen as transition systems over $\mathcal{A}ct' = \mathcal{A}ct \cup \{a_{\rightarrowtail}\}$. Relation $\mathcal{Z}_1$ already satisfies the conditions for atomic propositions and successors by actions in $\mathcal{A}ct$, we just need to check that it also satisfies the conditions for the additional action $a_{\rightarrowtail}$. Let $(x, x') \in \mathcal{Z}_1$, and let $y \in t_{\mathcal{S}}^{\rightarrowtail}$ be an $a_{\rightarrowtail}$-successor of x in $t_{\mathcal{S}}^{\rightarrowtail}$, *i.e.*, y is such that $w(x) \rightarrowtail w(y)$. Because $\mathcal{Z}_1$ is a bisimulation between $t_{\mathcal{S}}$ and $t'_{\mathcal{S}}$ and y is (obviously) reachable from the root of $t_{\mathcal{S}}$, there must be some $y' \in t'_{\mathcal{S}}$ such that $(y, y') \in \mathcal{Z}_1$. Now because $(x, x') \in \mathcal{Z}_1$, we have that $w(x) = w(x')$, and similarly $w(y) = w(y')$. It follows that $w(x') \rightarrowtail w(y')$, so y' is an $a_{\rightarrowtail}$-successor of x' in $t_{\mathcal{S}}^{\rightarrowtail}$. Since $(y, y') \in \mathcal{Z}_1$, Point 2 of Definition 2.5 holds, which concludes. The argument for Point 3 is symmetric. □

The following proposition can easily be established with a straightforward extension of the effective translation of μ-calculus formulas into MSO given, *e.g.*, in [23, Ch14].

Proposition 3.3. *For every path relation $\rightarrowtail$, every $L_\mu^{\rightarrowtail}$-definable class is $MSO^{\rightarrowtail}$-definable.*

We now engage our main concern, the expressive completeness of $\mathsf{L}_\mu^{\rightarrowtail}$ with respect to $\mathsf{MSO}^{\rightarrowtail}$. As in [27], due to Proposition 3.1, this question is meaningful only for bisimulation-closed classes of transition systems. We thus seek properties on the path relation $\rightarrowtail$ so that $\mathsf{L}_\mu^{\rightarrowtail}$ is expressive complete with respect to $\mathsf{MSO}^{\rightarrowtail}$, in the following sense:

[2]See footnote 1 page 5.

Definition 3.4 (Expressive completeness). $\mathsf{L}_\mu^{\leadsto}$ is *expressive complete* with respect to $\mathsf{MSO}^{\leadsto}$ if every bisimulation-closed class of transition systems that is $\mathsf{MSO}^{\leadsto}$-definable is also $\mathsf{L}_\mu^{\leadsto}$-definable.

Classes of relations. We recall the notions of recognisable, regular and rational relations. Let Σ be a finite alphabet: A binary relation over Σ^* is *rational* if there is a finite state automaton with two tapes (a *transducer*) that accepts a pair of words over Σ if, and only if, it is in the relation (see [3] for details). In general, transducers can read their two input words *asynchronously*, *i.e.*, they can progress at different paces on each tape. An example of rational relation is the epistemic relation of an agent with asynchronous perfect recall (*i.e.*, an agent that remembers the whole sequence of observations she makes modulo stuttering; see, *e.g.*, [39]). A binary relation over Σ^* is *regular* if it is accepted by a *synchronous* transducer[3]. An example is the epistemic relation of an agent with synchronous perfect recall [4, 6]. Finally, a relation over Σ^* is *recognisable* if there is a finite family of regular languages $\mathcal{L}_1, \mathcal{L}_1', \dots, \mathcal{L}_n, \mathcal{L}_n' \subseteq \Sigma^*$ such that $\leadsto\ = \bigcup_{i=1}^{n} \mathcal{L}_i \times \mathcal{L}_i'$ (refer to [3, Th.1.5, p.46] for details). For example, if an agent has bounded memory represented by a finite state machine whose states represent the different possible states of the agent's memory (as in, *e.g.*, [18]), then her indistinguishability relation is recognisable (see [34, Lemma 25, p.120]).

Definition 3.5. A path relation $\leadsto$ is *rational* (resp. *regular*, *recognisable*) if there are finite subsets $AP \subset \mathcal{AP}$ and $A \subset \mathcal{A}ct$ such that $\leadsto$ is equal to some rational (resp. regular, recognisable) relation over $(A \times 2^{AP})^*$. In that case we also say that $\leadsto$ is *over signature* (AP, A). Finally, when we fix such a rational, regular or recognisable path relation $\leadsto$ over signature (AP, A), we restrict logics $\mathsf{L}_\mu^{\leadsto}$ and $\mathsf{MSO}^{\leadsto}$ to transition systems over this signature, *i.e.*, transition systems that only involve atomic propositions in AP and actions in A.

We now present the two main results of this work. The first one will be proved in Section 4, the second one in Section 5.

Theorem 3.6. *For any recognisable path relation $\leadsto$, $\mathsf{L}_\mu^{\leadsto}$ is expressive complete w.r.t. $\mathsf{MSO}^{\leadsto}$.*

Theorem 3.7. *There exists a regular (and hence rational) path relation $\leadsto$ such that $\mathsf{L}_\mu^{\leadsto}$ is not expressive complete with regards to $\mathsf{MSO}^{\leadsto}$.*

4 TREE AUTOMATA FOR THE JUMPING μ-CALCULUS

In this section we present the class of *jumping tree automata*, first introduced in [35]. We prove that they are equivalent to the jumping μ-calculus, and from this result we derive corollaries on the expressivity and complexity of the jumping μ-calculus with recognisable path relation, among which Theorem 3.6. Because the semantics of jumping tree automata is given by two-player games, we start with basics on parity games.

4.1 Two-player games

We define two-player turn-based parity games with perfect information. Because the only such games we will consider are acceptance games for tree automata and evaluation games for the μ-calculus, we call the players Verifier and Refuter. A *game arena* is a tuple $G = (V, v_\iota, E, V_1, V_2)$, where V is a set of *positions* partitioned between positions of Verifier (V_1) and those of Refuter (V_2). Binary relation $E \subseteq V \times V$, that we assume left-total[4], is a set of *moves*. Finally, v_ι is the *initial position*. A *play* $\pi \in V^\omega$ is an infinite sequence of positions such that $\pi[0] = v_\iota$, and for all $i \geq 0$, $(\pi[i], \pi[i+1]) \in E$. A *partial play* $\rho = v_0 \dots v_n \in V^*$ is a finite prefix of a play, and we say that it

[3] *i.e.*, it progresses at the same pace on each tape.

[4] for all $v \in V$, there is $v' \in V$ such that $(v, v') \in E$

ends in v_n. A *strategy* σ for Verifier is a partial function $\sigma : V^* \to V$ such that for every partial play ρ ending in $v \in V_1$, $\sigma(\rho)$ is defined and $(v, \sigma(\rho)) \in E$. A play π *follows* a strategy σ if for all $i \geq 0$ such that $\pi[i] \in V_1$, $\pi[i+1] = \sigma(\pi[0,i])$, and similarly for partial plays. Given a game arena G and a strategy σ for Verifier in G, we denote by $\text{Out}(G, \sigma)$ the set of *outcomes* of σ, *i.e.* the set of plays in G that follow σ. A strategy is *memoryless* if it only depends on the last position of partial plays.

A *parity game* is a tuple $\mathcal{G} = (G, C)$, where $G = (V, v_\iota, E, V_1, V_2)$ is a game arena and $C : V \to \mathbb{N}$ is a *colouring*: a play is *winning* for Verifier if the least colour occurring infinitely often along the play is even, otherwise it is winning for Refuter. A *winning strategy* for Verifier is a strategy whose outcomes are all winning for Verifier. Finally, as we will focus on winning strategies of Verifier, we say that a position v of a game $\mathcal{G}$ is *winning* if Verifier has a winning strategy in $(\mathcal{G}, v)$, *i.e.* the game $\mathcal{G}$ where the initial position has been changed for v.

We recall that in a parity game with finitely many colours, if a player has a winning strategy then she has a memoryless one [54].

4.2 Jumping tree automata and the jumping μ-calculus

For a set X, $\mathbb{B}^+(X)$ is the set of positive boolean formulas over X, *i.e.* formulas built with elements of X as atomic propositions using only connectives $\vee$ and $\wedge$, where $\wedge$ has precedence over $\vee$. We also allow for formulas $\top$ and $\bot$, and we denote typical elements of $\mathbb{B}^+(X)$ by $\alpha, \beta \ldots$ Let $Dir = \{\langle a \rangle, [a] \mid a \in \mathcal{A}ct\} \cup \{\langle\rightsquigarrow\rangle, [\rightsquigarrow]\}$ be the set of *automaton directions*.

Definition 4.1. A *jumping tree automaton (JTA) over AP* is a tuple $\mathcal{A} = (AP, Q, q_\iota, \delta, C)$ where $AP \subset \mathcal{AP}$ is a finite set of atomic propositions, Q is a finite set of states, $q_\iota \in Q$ is an initial state, $\delta : Q \times 2^{AP} \to \mathbb{B}^+(Dir \times Q)$ is a transition function, and $C : Q \to \mathbb{N}$ is a colouring function.

JTAs resemble alternating tree automata [23, Ch. 9]. *Action directions*, those of the form $\langle a \rangle$ and $[a]$, are meant to go down the input tree, whereas the new *jump directions* $\langle\rightsquigarrow\rangle$ and $[\rightsquigarrow]$ of JTAs rely on an *a priori* given path relation. We shall denote a JTA $\mathcal{A}$ equipped with a path relation $\rightsquigarrow$ by $(\mathcal{A}, \rightsquigarrow)$. Acceptance of an *AP*-tree by $(\mathcal{A}, \rightsquigarrow)$ is defined on a two-player parity game between Verifier and Refuter: let $t = (\tau, m)$ be an *AP*-tree, and let $\mathcal{A} = (AP, Q, q_\iota, \delta, C)$. Consider the following parity game $\mathcal{G}^t_{\mathcal{A},\rightsquigarrow} = (V, v_\iota, E, V_1, V_2, C')$: its set of positions is $V = \tau \times Q \times \mathbb{B}^+(Dir \times Q)$, its initial position is $(\epsilon, q_\iota, \delta(q_\iota, \ell^\epsilon))$, and a position (x, q, α) belongs to Verifier if α is of the form $\alpha_1 \vee \alpha_2$, $[\langle a \rangle, q']$, or $[\langle\rightsquigarrow\rangle, q']$; otherwise it belongs to Refuter. The possible moves in $\mathcal{G}^t_{\mathcal{A},\rightsquigarrow}$ are the following:

$$(x, q, \alpha_1 \dagger \alpha_2) \to (x, q, \alpha_i) \quad \text{where } \dagger \in \{\vee, \wedge\} \text{ and } i \in \{1, 2\}$$

$$(x, q, [@, q']) \to (y, q', \delta(q', \ell^y)) \quad \text{where } @ \in \{\langle a \rangle, [a]\} \text{ and } y \text{ is an } a\text{-child of } x$$

$$(x, q, [\ominus, q']) \to (y, q', \delta(q', \ell^y)) \quad \text{where } \ominus \in \{\langle\rightsquigarrow\rangle, [\rightsquigarrow]\} \text{ and } x \rightsquigarrow y$$

Positions that have no possible move as defined above are made sink positions (to get a game with no dead ends) that are winning for Verifier if they are of the form $(x, q, \top)$, $(x, q, [[a], q'])$ or $(x, q, [[\rightsquigarrow], q'])$, and winning for Refuter otherwise. The colouring function C' of $\mathcal{G}^t_{\mathcal{A},\rightsquigarrow}$ is inherited from the one of $\mathcal{A}$ by letting $C'(x, q, \alpha) = C(q)$. An *AP*-tree t is *accepted* by $(\mathcal{A}, \rightsquigarrow)$ if Verifier has a winning strategy in $\mathcal{G}^t_{\mathcal{A},\rightsquigarrow}$, and a tree is accepted by $(\mathcal{A}, \rightsquigarrow)$ if its underlying *AP*-tree (obtained by intersecting all labels with *AP*) is accepted by $(\mathcal{A}, \rightsquigarrow)$. We let $\mathcal{L}(\mathcal{A}, \rightsquigarrow) = \{\mathcal{S} \mid t_\mathcal{S} \text{ is accepted by } (\mathcal{A}, \rightsquigarrow)\}$. If $\mathcal{A}$ is an alternating tree automaton (*i.e.* it has no jumping action), it needs not be equipped with a relation, and we write $\mathcal{L}(\mathcal{A})$ for the set of systems whose unfoldings it accepts.

In the following, the *size* of a JTA $\mathcal{A}$, written $|\mathcal{A}|$, is the sum of its number of states and its number of colours.

We show that JTA and $\mathsf{L}_\mu^{\rightarrowtail}$ are equally expressive, in a strong sense: we establish a two-way translation that does not depend on the *a priori* chosen path relation.

Proposition 4.2.

(a) *For every formula* $\varphi \in \mathsf{L}_\mu^{\rightarrowtail}$*, there is a JTA* $\mathcal{A}_\varphi$ *such that, for every path relation* $\rightarrowtail$*,* $\mathcal{L}(\varphi, \rightarrowtail) = \mathcal{L}(\mathcal{A}_\varphi, \rightarrowtail)$*,*

(b) *for every JTA* $\mathcal{A}$*, there is an* $\mathsf{L}_\mu^{\rightarrowtail}$*-formula* $\varphi_{\mathcal{A}}$ *such that, for every path relation* $\rightarrowtail$*,* $\mathcal{L}(\mathcal{A}, \rightarrowtail) = \mathcal{L}(\varphi_{\mathcal{A}}, \rightarrowtail)$*.*

Moreover, the translations are effective and linear.

Proof. First, as already explained in the proof of Proposition 3.1, given a path relation $\rightarrowtail$, any marked tree t gives rise to a transition system $t^{\rightarrowtail}$ over actions in $\mathcal{A}ct' := \mathcal{A}ct \cup \{a_{\rightarrowtail}\}$, where $a_{\rightarrowtail}$ is a fresh action symbol. Second, we make the two following observations:

(1) An $\mathsf{L}_\mu^{\rightarrowtail}$-formula over $\mathcal{A}ct$ can be interpreted as an L_μ-formula on transition systems over $\mathcal{A}ct'$, and vice-versa.
(2) A JTA over $\mathcal{A}ct$ can be interpreted as an alternating automaton on transition systems over $\mathcal{A}ct'$, and vice-versa.

We now prove Proposition 4.2.

For Point (a), take a formula $\varphi \in \mathsf{L}_\mu^{\rightarrowtail}$. Let φ' be the same formula as φ, but seen as an L_μ-formula over $\mathcal{A}ct'$. By the classic correspondence between alternating tree automata and the μ-calculus [23, Chap. 9, Chap. 10], one can build in linear time an alternating tree automaton $\mathcal{A}_{\varphi'}$ such that, over transition systems with actions in $\mathcal{A}ct'$, $\mathcal{L}(\varphi') = \mathcal{L}(\mathcal{A}_{\varphi'})$. Now, let $\mathcal{A}_\varphi$ be the same automaton, but seen as a jumping tree automaton over $\mathcal{A}ct$.

First, observe that we have not fixed a path relation before building $\mathcal{A}_\varphi$ from φ. Now, let us take a path relation $\rightarrowtail$. We show that $\mathcal{L}(\varphi, \rightarrowtail) = \mathcal{L}(\mathcal{A}_\varphi, \rightarrowtail)$. It is clear that $\mathcal{L}(\varphi, \rightarrowtail) = \{\mathcal{S} \mid t_{\mathcal{S}}^{\rightarrowtail} \in \mathcal{L}(\varphi')\}$, and also that $\mathcal{L}(\mathcal{A}_\varphi, \rightarrowtail) = \{\mathcal{S} \mid t_{\mathcal{S}}^{\rightarrowtail} \in \mathcal{L}(\mathcal{A}_{\varphi'})\}$; because $\mathcal{L}(\varphi') = \mathcal{L}(\mathcal{A}_{\varphi'})$, we are done.

For Point (b) of Proposition 4.2, we just roll back the above argumentation. □

4.3 The case of recognisable relations

In this section, we restrict our attention to recognisable path relations. The following proposition relies on the fact that recognisable relations are MSO-definable, which is folklore.

Proposition 4.3. *MSO*$^{\rightarrowtail}$ *with recognisable path relation is not more expressive than MSO.*

Proof. If $\rightarrowtail$ is a recognisable path relation over signature (AP, A), by definition there is a finite family of regular languages $\mathcal{L}_1, \mathcal{L}'_1, \dots, \mathcal{L}_n, \mathcal{L}'_n \subseteq (A \times 2^{AP})^*$ such that $\rightarrowtail \; = \bigcup_{i=1}^{n} \mathcal{L}_i \times \mathcal{L}'_i$. By Kleene's theorem, every regular language of finite words is recognisable by a finite state automaton, and by the Büchi-Elgot-Trakhtenbrot theorem every language of finite words accepted by a finite automaton is definable in MSO[S] on words, with the successor binary relation symbol S and one monadic relation symbol $Q_{a,\ell}$ for each $(a, \ell) \in A \times 2^{AP}$ (see [49, Theorem 3.1, p.8]). Therefore, for every $i \in \{1, \dots, n\}$ there are MSO[S] sentences φ_i and φ'_i such that for any word w over $A \times 2^{AP}$, we have that $w \models \varphi_i$ (resp. $w \models \varphi'_i$) iff $w \in \mathcal{L}_i$ (resp. $w \in \mathcal{L}'_i$). We now describe how to transform each such sentence φ into a formula $\widehat{\varphi}(X)$ of MSO on trees such that $t, V \models \widehat{\varphi}(X)$ iff $V(X) = \{x\}$ and $w(x) \models \varphi$. First, one states that X needs be a singleton $\{x\}$ (see, for instance, [23, p. 210]). Then one restricts quantifications to nodes that are prefixes of x (recall that the prefix relation can be expressed in MSO [23, Lemma 12.11, p. 212]), and replaces each formula of the form $S(X, Y)$ with $\bigvee_{a \in A} a(X, Y)$ and each formula of the form $Q_{a,\ell}(X)$ with a formula saying that X is a singleton $\{x\}$ such that x is labelled with ℓ and is either the root or an a-child. It is then easy to see that, on marked trees over signature (AP, A), MSO$^{\rightarrowtail}$ formula $X \rightarrowtail Y$ is equivalent to the MSO formula $\bigvee_{i=1}^{n} \widehat{\varphi_i(X)} \wedge \widehat{\varphi_i'(Y)}$. Therefore every MSO$^{\rightarrowtail}$ formula φ can be turned into an MSO formula φ' such that for every marked tree t over (AP, A) and valuation V, it holds that $t, V \models^{\rightarrowtail} \varphi$ iff $t, V \models \varphi'$. ◻

Remark 1. Observe that in the latter proof, to move from MSO on words to MSO on trees, we use a disjunction over the set of different possible actions. Such a disjunction is also used to express the prefix relation in MSO on trees. This is possible because, following Definition 3.5, we restrict to models that only use the finite number of actions considered in the alphabet of the fixed recognisable path relation. Should it not be so, we would need to enrich MSO on trees, for instance with a relation that relates nodes x and y if y is an a-child of x for *some* action $a \in \mathcal{A}ct$.

Note that Theorem 3.6 is obtained from Proposition 4.3, Proposition 3.1, and the expressive completeness of the μ-calculus with regards to MSO. However, the train of thoughts above that yields the collapse of the jumping μ-calculus down to the μ-calculus uses transformations that cannot be exploited for accurate complexity bounds regarding the jumping μ-calculus.

Below, we first establish that the satisfiability problem for jumping μ-calculus with recognisable path relation is Exptime-complete, and thus no harder than the satisfiability problem for the μ-calculus [20, 44, 47]. We then prove an upper bound on the succinctness of the jumping μ-calculus with regards to the μ-calculus. Our results rely on the automata counterpart of the jumping mu-calculus, and on the relationship between two-way alternating automata and classic tree automata as studied by [52] in the setting of trees with bounded branching degree.

Proposition 4.4. *The satisfiability problem for $\mathsf{L}_\mu^{\rightarrowtail}$ with recognisable path relation over transition systems with bounded branching degree is* Exptime*-complete.*

Proof. The hardness follows from EXPTIME-hardness of the satisfiability problem for standard μ-calculus [20]. Now for the upper bound, the result follows from Proposition 4.2 together with the following two points. First, from a JTA equipped with (an automaton representing a) recognisable relation, one can build in polynomial time an equivalent two-way tree automaton [34, 35]. Second, for trees of bounded arity, the emptiness problem for two-way tree automata is in Exptime [52]. ◻

Proposition 4.5. *The jumping μ-calculus with a recognisable path relation over transition systems with bounded branching degree is at most exponentially more succinct than the μ-calculus.*

Proof. Fix a recognisable relation $\rightarrowtail$. We explain how to build a μ-calculus formula φ' equivalent to a jumping μ-calculus formula φ, whose size is at most exponential in the size of φ. By the equivalence between alternating tree automata and the μ-calculus (with a linear two-way translation) [23, Ch. 10], it suffices to show that for each $\varphi \in \mathsf{L}_\mu^{\rightarrowtail}$ interpreted with any recognisable path relation, there exists an alternating tree automaton of size exponential in the size of φ that accepts precisely the models of φ. Let $\varphi \in \mathsf{L}_\mu^{\rightarrowtail}$. By Proposition 4.2, there exists a jumping tree automaton $\mathcal{A}_\varphi$ of size linear in the size of φ such that $\mathcal{L}(\mathcal{A}_\varphi, \rightarrowtail) = \mathcal{L}(\varphi, \rightarrowtail)$. Then, as proved in [35], there is a two-way tree automaton $\mathcal{A}_\varphi^{\rightarrowtail}$ of size polynomial in the size of $\mathcal{A}_\varphi$ such that $\mathcal{L}(\mathcal{A}_\varphi, \rightarrowtail) = \mathcal{L}(\mathcal{A}_\varphi^{\rightarrowtail})$. Finally, because we consider trees of bounded branching degree, we have by [52] that there is a non-deterministic (hence alternating) tree automaton $\mathcal{B}_\varphi^{\rightarrowtail}$ of size exponential in the size of $\mathcal{A}_\varphi^{\rightarrowtail}$ such that $\mathcal{L}(\mathcal{B}_\varphi^{\rightarrowtail}) = \mathcal{L}(\mathcal{A}_\varphi^{\rightarrowtail})$. Automaton $\mathcal{B}_\varphi^{\rightarrowtail}$ is our candidate, which concludes. ◻

5 GAMES AND THE JUMPING μ-CALCULUS

This section is essentially dedicated to the proof of Theorem 3.7. To this aim, we focus on the property stating the existence of a winning strategy in two-player turned-based reachability games with imperfect information and perfect recall. While this property is bisimilar-invariant and expressible in the monadic second order logic with path relation (MSO$^{\rightarrowtail}$), we demonstrate that it cannot be expressed in the jumping μ-calculus ($\mathsf{L}_\mu^{\rightarrowtail}$). However, we point out that this property can be expressed in $\mathsf{L}_\mu^{\rightarrowtail}$ with the additional assumptions that winning conditions are *observable*, if we consider the notion of perfect recall where also actions are remembered.

5.1 Two-player games with imperfect information and synchronous perfect recall

We recall the classic framework of two-player games with imperfect information [4, 5, 41, 42]. The two players are Eve and Adam. In these games, Eve only partially observes the positions of the game, such that some positions are indistinguishable to her, while Adam has perfect information. Let us fix a countable set of *observations*, $Obs = \{o, o', \dots\}$.

Arenas and plays. An *imperfect-information arena* is a tuple $G^{\mathrm{i}} = (V, v_\iota, \{a^{G^{\mathrm{i}}}\}_{a\in\mathcal{A}ct}, \{o^{G^{\mathrm{i}}}\}_{o\in Obs})$, where V is a set of positions, $v_\iota \in V$ is an initial position, each $a^{G^{\mathrm{i}}}$ is a binary relation over V and each $o^{G^{\mathrm{i}}}$ is a subset of V such that $\{o^{G^{\mathrm{i}}}\}_{o\in Obs}$ forms a partition of V. For every position v, o_v denotes the unique observation such that $v \in o_v$. For $v \in V$ and $a \in \mathcal{A}ct$, we say that a is *available* in v if $a^{G^{\mathrm{i}}}(v) \neq \emptyset$. We assume that there is at least one available action in every position, and that if two positions have the same observation they also share the same set of available actions. Without loss of generality, we additionally require that in any game arena, every position is reachable from the initial position v_ι.

Players take turns, starting with Eve. If the current position is v, Eve chooses an action a available in v, and Adam chooses a new position $v' \in a^{G^{\mathrm{i}}}(v)$. Similarly to perfect information games, a *play* (resp. *partial play*) is an infinite (resp. finite) sequence $\pi = v_0 a_1 v_1 a_2 \dots$ (resp. $\rho = v_0 a_1 v_1 \dots a_n v_n$) such that $v_0 = v_\iota$ and, for all i, $v_{i+1} \in a_{i+1}^{G^{\mathrm{i}}}(v_i)$. For a partial play $\rho = v_0 a_1 v_1 \dots a_n v_n$, we let $|\rho| := n$.

Indistinguishability relation and uniformity. Eve's imperfect observation of the game yields an equivalence relation over partial plays, gathering those plays that are indistinguishable to her. This relation depends on the observation function o as well as on Eve's memory abilities. We focus on the classic case of *synchronous perfect recall* [39, 41], where Eve remembers the whole sequence of observations that she receives as well as her actions. We therefore define the *indistinguishability equivalence* over partial plays as follows: for $\rho = v_0 a_1 \dots a_n v_n$ and $\rho' = v'_0 a'_1 \dots a'_n v'_n$, we let

$$\rho \sim \rho' \text{ if for all } 1 \leq i \leq n, o_{v_i} = o_{v'_i} \text{ and } a_i = a'_i.$$

Eve's choices can only be based on the observations she receives, so that her strategy must be defined *uniformly* over partial plays that are indistinguishable to her, as captured by the following definition. Formally, a *strategy* for Eve is a partial function $\sigma : \{v_\iota\}(\mathcal{A}ct \cdot V)^* \to \mathcal{A}ct$ such that for two partial plays ρ and ρ', if $\rho \sim \rho'$, then $\sigma(\rho) = \sigma(\rho')$. We say that a play $\pi = v_0 a_1 v_1 \dots$ *follows* a strategy σ if for all $i \geq 0$, $a_{i+1} = \sigma(v_0 a_1 v_1 \dots a_i v_i)$.

Winning conditions and classes of games. A *parity (resp. reachability) game with imperfect information* $\mathcal{G}^{\mathrm{i}}$ is an imperfect-information arena $G^{\mathrm{i}} = (V, v_\iota, \{a^{G^{\mathrm{i}}}\}_{a\in\mathcal{A}ct}, \{o^{G^{\mathrm{i}}}\}_{o\in Obs})$ together with a parity winning condition $C : V \to \mathbb{N}$ (resp. reachability winning condition $W \subseteq V$).

Observe that a parity (resp. reachability) game with imperfect information is a transition system over $\mathcal{AP} = Obs \cup \mathbb{N}$ (resp. $\mathcal{AP} = Obs \cup \{W\}$) and $\mathcal{A}ct$. To address logical definability of the existence of winning strategies, we consider games that only use a finite number of actions, observations and parities.

For every finite subsets of actions $A \subset \mathcal{A}ct$ and observations $O \subset Obs$, and every $k \in \mathbb{N}$, we define the classes $\mathcal{P}(A, O, k)$ (resp. $\mathcal{R}(A, O)$) of parity (resp. reachability) games with imperfect information where actions range over A, observations range over O, and parities are not greater than k. Note that each of these classes is definable in the propositional μ-calculus.

Observe also that the relation $\sim$ on partial plays induces a path relation, that we shall also write $\sim$ in the rest of this section. Also all notations where $\rightsquigarrow$ occurs will be specialised using $\sim$, *e.g.*, $\mathsf{L}_\mu^\sim$, $\mathsf{MSO}^\sim$, $\models^\sim$, etc.

The question that we address in this section is whether the subclass of $\mathcal{P}(A, O, k)$ composed of games where Eve has a winning strategy is $\mathsf{L}_\mu^\sim$-definable, and similarly for the class $\mathcal{R}(A, O)$.

As we show, the answers depend on whether the winning condition is *observable* [12] or not. More precisely, a parity (resp. reachability) winning condition is *observable* if $o_v = o_{v'}$ implies $c(v) = c(v')$ (resp. $v \in W$ iff $v' \in W$). Note that the class of (parity or reachability) games with observable winning condition, with finitely many actions, observations and parities, is also definable in the μ-calculus. We define the classes $\mathcal{P}_o(A, O, k)$ and $\mathcal{R}_o(A, O)$ as the subclasses of $\mathcal{P}(A, O, k)$ and $\mathcal{R}(A, O)$ (respectively) where games have observable winning conditions.

Now, we recall the following result:

Proposition 5.1 ([6]). *Let $\mathcal{G}^{\mathrm{i}}$ and $\mathcal{G}^{\mathrm{i}\prime}$ be imperfect-information parity games. If $\mathcal{G}^{\mathrm{i}} \leftrightarroweq \mathcal{G}^{\mathrm{i}\prime}$, then Eve has a winning strategy in $\mathcal{G}^{\mathrm{i}}$ if, and only if, she has a winning strategy in $\mathcal{G}^{\mathrm{i}\prime}$.*

We point out that this result also holds for reachability games, and therefore the classes $\mathcal{P}(A, O, k)$, $\mathcal{R}(A, O)$, $\mathcal{P}_o(A, O, k)$ and $\mathcal{R}_o(A, O)$ are closed by bisimulation.

5.2 Observable winning conditions: definability in the jumping μ-calculus

This section is dedicated to the proof of the following result:

Theorem 5.2. *The subclasses of $\mathcal{R}_o(A, O)$ and $\mathcal{P}_o(A, O, k)$ with finite branching where Eve has a winning strategy are both $\mathsf{L}_\mu^\sim$-definable.*

For a finite subset of actions $A \subset \mathcal{A}ct$ and observations $O \subset \mathcal{O}bs$, for $k \in \mathbb{N}$, define:

$$\mathrm{WinReach}^A := \mu X.(W \vee \bigvee_{a \in A} \boxed{\sim}\boxed{a} X), \text{ and}$$

$$\mathrm{WinParity}_k^A := \nu X_0.\mu X_1 \ldots \eta X_k. \bigvee_{0 \le i \le k} (i \wedge \bigvee_{a \in A} \boxed{\sim}\boxed{a} X_i),$$

where $\eta = \mu$ if k is odd, ν otherwise.

We establish the following result, from which Theorem 5.2 follows.

Proposition 5.3. *For all $\mathcal{G}^{\mathrm{i}}{}_r \in \mathcal{R}_o(A, O)$ and $\mathcal{G}^{\mathrm{i}}{}_p \in \mathcal{P}_o(A, O, k)$, it holds that*

$$\mathcal{G}^{\mathrm{i}}{}_r \models^\sim \mathit{WinReach}^A \quad \text{if, and only if,} \quad \text{Eve has a winning strategy in } \mathcal{G}^{\mathrm{i}}{}_r, \quad \text{and}$$

$$\mathcal{G}^{\mathrm{i}}{}_p \models^\sim \mathit{WinParity}_k^A \quad \text{if, and only if,} \quad \text{Eve has a winning strategy in } \mathcal{G}^{\mathrm{i}}{}_p.$$

Proof of Proposition 5.3. We only treat the case of parity games, the case of reachability games being similar and simpler.

We assume that the reader is familiar with evaluation games for the μ-calculus (see [16]). Let $\mathcal{G}^{\mathrm{i}} = (V, v_\iota, \{a^{\mathcal{G}^{\mathrm{i}}}\}_{a \in A}, \{o^{\mathcal{G}^{\mathrm{i}}}\}_{o \in O}, C) \in \mathcal{P}_o(A, O, k)$ be a parity game with imperfect information and finite branching. Seeing formula $\mathrm{WinParity}_k^A$ as a formula from basic μ-calculus over $A' = A \cup \{a_\sim\}$, we also let $\mathcal{G}^*$ be the evaluation game of $\mathrm{WinParity}_k^A$ on the transition system $t_{\mathcal{G}^{\mathrm{i}}}^\sim$. This game is a perfect-information game with parity condition. Its positions are of the form (φ, x), where φ is a subformula of $\mathrm{WinParity}_k^A$, and x is a node in $t_{\mathcal{G}^{\mathrm{i}}}$. Because there is a one-to-one correspondence between nodes in $t_{\mathcal{G}^{\mathrm{i}}}$ and partial plays in $\mathcal{G}^{\mathrm{i}}$, we can use the latter view and consider that positions in $\mathcal{G}^*$ are of the form (φ, ρ).

Game's functioning. In this game, the initial position is $(\mathrm{WinParity}_k^A, v_\iota)$. If for each $0 \leq i \leq k$ we let φ_i be the unique subformula of $\mathrm{WinParity}_k^A$ of the form $\eta_i X_i.\psi$, then the only possible $(k+1)$ first moves (say, for Refuter) are to go through $(\varphi_1, v_\iota), \ldots, (\varphi_k, v_\iota)$ and reach position $(\bigvee_{0\leq i\leq k}(i \wedge \bigvee_{a\in A} \boxed{\sim}\boxed{a} X_i), v_\iota)$, which belongs to Verifier. In this position, Verifier chooses some $i \in \{1, \ldots, k\}$ and moves to $(i \wedge \bigvee_{a\in A} \boxed{\sim}\boxed{a} X_i, v_\iota)$. Next, Refuter can either move to (i, v_ι) or $(\bigvee_{a\in A} \boxed{\sim}\boxed{a} X_i, v_\iota)$. In the former case, the game moves to a position that is winning for Verifier if i is the colour of v_ι, and winning for Refuter otherwise. In the latter case, Verifier chooses an action $a \in A$ and moves to $(\boxed{\sim}\boxed{a} X_i, v_\iota)$; then Refuter chooses some partial play ρ with $v_\iota \sim \rho$ and moves to $(\boxed{a} X_i, \rho)$. Observe that, since all plays start in v_ι, the only partial play equivalent to v_ι is v_ι, so that in this first round Refuter can only choose $\rho = v_\iota$. Then, from $(\boxed{a} X_i, v_\iota)$ Refuter chooses some a-child of v_ι in $t_{\mathcal{G}^{\mathrm{i}}}$, or equivalently some position $v' \in a^{G^{\mathrm{i}}}(v_\iota)$, and moves to $(X_i, v_\iota a v')$, from where the only move is to go to $(\varphi_i, v_\iota a v')$, from where a new round begins.

Verifier's real choices. We have seen that the only positions where Verifier makes a choice are either of the form $(\bigvee_{0\leq i\leq k}(i \wedge \bigvee_{a\in A} \boxed{\sim}\boxed{a} X_i), \rho)$, or $(\bigvee_{a\in A} \boxed{\sim}\boxed{a} X_i, \rho)$. But positions of the first type do not offer Verifier a real choice as the only move that does not make her lose immediately is to pick the colour of the current position in $\mathcal{G}^{\mathrm{i}}$, *i.e.*, to choose $i = C(\mathrm{last}(\rho))$, where $\mathrm{last}(\rho)$ is the position in which ρ ends. In the following we thus assume, without loss of generality, that Verifier always makes the right choice in such positions.

Relevant colours. Also, the colouring C^* of positions in $\mathcal{G}^*$ is such that only colours of positions of the form (X_i, ρ) are relevant (other positions all have colours higher than k), and it is defined as $C^*(X_i, \rho) = i$. For this reason, for a play π^* in $\mathcal{G}^*$, we shall write $C^*(\pi^*)$ for the sequence of relevant colours in π^*, *i.e.*, the sequence of colours of positions of the form (X_i, ρ) in π^*, and similarly for partial plays. Note that if a partial play ρ^* in $\mathcal{G}^*$ ends in a position of the form $(X_i, \rho a v)$, then $|C^*(\rho^*)| = |\rho|$. Given a strategy σ^* for Verifier in $\mathcal{G}^*$, we let $C^*(\sigma^*) := \{C^*(\pi) \mid \pi \in \mathrm{Out}(\mathcal{G}^*, \sigma^*)\}$. Finally, for a memoryless strategy σ^* of Verifier and a position of the form $(\bigvee_{a\in A} \boxed{\sim}\boxed{a} X_i, \rho)$, if σ^* is such that $\sigma^*(\bigvee_{a\in A} \boxed{\sim}\boxed{a} X_i, \rho) = (\boxed{\sim}\boxed{a} X_i, \rho)$, we shall abuse notation and vocabulary and write $\sigma^*(\bigvee_{a\in A} \boxed{\sim}\boxed{a} X_i, \rho) = a$, as well as say that σ^* chooses a in this position.

We have that $\mathcal{G}^{\mathrm{i}} \models^{\sim} \mathrm{WinParity}_k^A$ if, and only if, Verifier has a winning strategy in $\mathcal{G}^*$ [16]. It thus remains to prove the following proposition, and we are done:

Proposition 5.4. *Verifier has a winning strategy in $\mathcal{G}^*$ iff Eve has a winning strategy in $\mathcal{G}^{\mathrm{i}}$.*

Proof. We start with the right to left implication. Assume that Eve has a winning strategy σ in $\mathcal{G}^{\mathrm{i}}$. We define a memoryless strategy σ^* for Verifier in $\mathcal{G}^*$ as follows: For every position of the form

$$v^* = (\bigvee_{0\leq i\leq k} (i \wedge \bigvee_{a\in A} \boxed{\sim}\boxed{a} X_i), \rho),$$

we let

$$\sigma^*(v^*) := (i \wedge \bigvee_{a\in A} \boxed{\sim}\boxed{a} X_i, \rho), \text{where } i = C(\mathrm{last}(\rho)),$$

and for every position of the form

$$v^* = (\bigvee_{a\in A} \boxed{\sim}\boxed{a} X_i, \rho),$$

we let

$$\sigma^*(v^*) := (\boxed{\sim}\boxed{a} X_i, \rho), \text{where } a = \sigma(\rho).$$

To prove that σ^* is winning we prove the following lemma (the proof of which can be found in Appendix A).

Lemma 5.5. *Let $\pi^* \in Out(\mathcal{G}^*, \sigma^*)$. For every finite prefix ρ^* of π^* such that $last(\rho^*) = (X_i, \rho av)$ for some $0 \leq i \leq k$, some partial play ρ, action a and position v, it holds that $C^*(\rho^*) = C(\rho)$ and ρav follows σ.*

We now prove that all outcomes of σ^* are winning for Verifier. Let $\pi^* \in \text{Out}(\mathcal{G}^*, \sigma^*)$. By Lemma 5.5, there are infinitely many prefixes ρ^* of π^* such that $C^*(\rho^*) = C(\rho)$ for some ρ that follows σ. Consider the tree made of the set of all such ρ. Because $\mathcal{G}^{\text{i}}$ has finite branching, so has this tree, and by König's lemma we can extract a play π that follows σ and such that $C(\pi) = C^*(\pi^*)$, from which we can conclude that $C^*(\pi^*)$ verifies the parity condition. This finishes the first implication.

Before proving the left to right implication, we establish the following lemma (the proof can be found in Appendix B):

Lemma 5.6. *Let ρ and ρ' be two partial plays in $\mathcal{G}^{\text{i}}$ such that $\rho \sim \rho'$. For every two partial plays ρ^* and $\rho^{*\prime}$ in $\mathcal{G}^*$ that end respectively in (X_i, ρ) and (X_j, ρ'), it holds that $C^*(\rho^*) = C^*(\rho^{*\prime})$.*

We can now finish the proof of Proposition 5.4.

For the left to right implication, assume that Verifier has a winning strategy in $\mathcal{G}^*$. Because $\mathcal{G}^*$ is a parity perfect-information game with finitely many colours, Verifier has a memoryless winning strategy σ^* [54]. We aim at defining a uniform winning strategy for Eve in $\mathcal{G}^{\text{i}}$. To do so, we first show that σ^* in $\mathcal{G}^*$ can be made uniform in the following sense: We say that a strategy for Verifier in $\mathcal{G}^*$ is *uniform on a partial play ρ* of $\mathcal{G}^{\text{i}}$ if it chooses the same action $a \in A$ in all positions of the form $(\bigvee_{a \in A} \boxed{\sim}\boxed{a} X_i, \rho')$, where $\rho' \sim \rho$. We first show that there exists a winning strategy for Verifier that is uniform on all partial plays. To do so we inductively define, for every $n \geq 0$, a strategy σ^*_n such that:

(1) σ^*_n is memoryless,
(2) σ^*_n is uniform on partial plays of length $|\rho| \leq n$, and
(3) $C^*(\sigma^*_n) \subseteq C^*(\sigma^*)$.

Observe that because σ^* is winning, Point 3 implies that σ^*_n is also winning. First, let $\sigma^*_0 := \sigma^*$, which is memoryless, clearly verifies Point 3, and is uniform on partial plays of length no greater than 0, as there are none.

Now, take $n \geq 0$ and assume that σ^*_n has been defined. For a partial play ρ such that $|\rho| = n+1$, we say that the position $(\bigvee_{a \in A} \boxed{\sim}\boxed{a} X_i, \rho)$ is *reachable by σ^*_n* if there is a play in $\text{Out}(\mathcal{G}^*, \sigma^*_n)$ that contains position $(\bigvee_{a \in A} \boxed{\sim}\boxed{a} X_i, \rho)$. For every $\sim$-equivalence class $\rho_\sim$ of partial plays in $\mathcal{G}^{\text{i}}$ of length $n+1$, define $a_{\rho_\sim} \in A$ as follows: if there exists $\rho \in \rho_\sim$ such that $(\bigvee_{a \in A} \boxed{\sim}\boxed{a} X_i, \rho)$ is reachable by σ^*_n, then choose one such ρ and let $a_{\rho_\sim} := \sigma^*_n(\bigvee_{a \in A} \boxed{\sim}\boxed{a} X_i, \rho)$. Otherwise, define $a_{\rho_\sim}$ arbitrarily.

Now, the memoryless strategy σ^*_{n+1} is defined as follows, where (φ, ρ) is a position of Verifier:

$$\sigma^*_{n+1}(\varphi, \rho) = \begin{cases} a_{[\rho]_\sim} & \text{if } \varphi = \bigvee_{a \in A} \boxed{\sim}\boxed{a} X_i \text{ and } |\rho| = n+1, \\ \sigma^*_n(\varphi, \rho) & \text{otherwise.} \end{cases}$$

It is clear that σ^*_{n+1} is uniform over partial plays of length no greater than $n+1$. We show that, in addition, $C^*(\sigma^*_{n+1}) \subseteq C^*(\sigma^*_n)$. By induction hypothesis, $C^*(\sigma^*_n) \subseteq C^*(\sigma^*)$, so that Point 3 will follow. Take a play $\pi^* \in \text{Out}(\mathcal{G}^*, \sigma^*_{n+1})$. Either π^* is an outcome of σ^*_n, in which case $C^*(\pi^*) \in C^*(\sigma^*_n)$, or there are ρ^* and $\pi^{*\prime}$ such that

$$\pi^* = \rho^* \cdot (\bigvee_{a \in A} \boxed{\sim}\boxed{a} X_i, \rho)(\boxed{\sim}\boxed{a} X_i, \rho)(\boxed{a} X_i, \rho')(X_i, \rho' av) \cdot \pi^{*\prime},$$

where $|\rho| = n+1$, $a = \sigma^*_{n+1}(\bigvee_{a\in A} \boxed{\sim}\boxed{a}X_i, \rho)$, $\rho' \sim \rho$ and $v \in a^{\mathcal{G}^{\mathrm{i}}}(\mathrm{last}(\rho'))$.

We show that $C^*(\pi^*) \in C^*(\sigma^*_n)$. First, observe that position $(\bigvee_{a\in A} \boxed{\sim}\boxed{a}X_i, \rho)$ is reachable by σ^*_{n+1}, and because σ^*_{n+1} coincides with σ^*_n on all positions in ρ^*, $(\bigvee_{a\in A} \boxed{\sim}\boxed{a}X_i, \rho)$ is also reachable by σ^*_n. Therefore, by definition of $a_{[\rho]_\sim}$, there exists $\rho'' \in [\rho]_\sim$ such that $(\bigvee_{a\in A} \boxed{\sim}\boxed{a}X_i, \rho'')$ is also reachable by σ^*_n and $a_{[\rho]_\sim} = \sigma^*_n(\bigvee_{a\in A} \boxed{\sim}\boxed{a}X_i, \rho'')$. By definition of σ^*_{n+1} we have $\sigma^*_{n+1}(\bigvee_{a\in A} \boxed{\sim}\boxed{a}X_i, \rho) = a_{[\rho]_\sim}$, so that $\sigma^*_{n+1}(\bigvee_{a\in A} \boxed{\sim}\boxed{a}X_i, \rho) = \sigma^*_n(\bigvee_{a\in A} \boxed{\sim}\boxed{a}X_i, \rho'') = a$.

Because $(\bigvee_{a\in A} \boxed{\sim}\boxed{a}X_i, \rho'')$ is reachable by σ^*_n, there exists $\rho^{*\prime}$ such that $\rho^{*\prime} \cdot (\bigvee_{a\in A} \boxed{\sim}\boxed{a}X_i, \rho'')$ is a partial play that follows σ^*_n. And because $a = \sigma^*_n(\bigvee_{a\in A} \boxed{\sim}\boxed{a}X_i, \rho'')$, we get that

$$\rho^{*\prime} \cdot (\bigvee_{a\in A} \boxed{\sim}\boxed{a}X_i, \rho'')(\boxed{\sim}\boxed{a}X_i, \rho'')$$

also follows σ^*_n. Now, as $\rho'' \sim \rho \sim \rho'$, we have that

$$\rho^{*\prime} \cdot (\bigvee_{a\in A} \boxed{\sim}\boxed{a}X_i, \rho'')(\boxed{\sim}\boxed{a}X_i, \rho'')(\boxed{a}X_i, \rho')$$

is also a valid partial play that follows σ^*_n; and because $v \in a^{\mathcal{G}^{\mathrm{i}}}(\mathrm{last}(\rho'))$, so is

$$\rho^{*\prime\prime} := \rho^{*\prime} \cdot (\bigvee_{a\in A} \boxed{\sim}\boxed{a}X_i, \rho'')(\boxed{\sim}\boxed{a}X_i, \rho'')(\boxed{a}X_i, \rho')(X_i, \rho' av).$$

Finally, let us define:

$$\pi^{*\prime\prime} := \rho^{*\prime\prime} \cdot \pi^{*\prime}.$$

Because π^* follows the memoryless strategy σ^*_{n+1} and $\pi^{*\prime}$ is a suffix of π^*, we have that $\pi^{*\prime}$ also follows the strategy σ^*_{n+1}. Observing that σ^*_n and σ^*_{n+1} coincide on all positions of $\pi^{*\prime}$, we get that $\pi^{*\prime}$ follows σ^*_n. And because $\rho^{*\prime\prime}$ also follows σ^*_n, we obtain that $\pi^{*\prime\prime} \in \mathrm{Out}(\mathcal{G}^*, \sigma^*_n)$. Now, clearly, $\rho' av \sim \rho' av$, so that by Lemma 5.6 we obtain that

$$C^*(\rho^* \cdot (\bigvee_{a\in A} \boxed{\sim}\boxed{a}X_i, \rho)(\boxed{\sim}\boxed{a}X_i, \rho)(\boxed{a}X_i, \rho')(X_i, \rho' av)) = C^*(\rho^{*\prime\prime}),$$

and thus $C^*(\pi^*) = C^*(\pi^{*\prime\prime})$, which concludes.

We now consider σ^*_U, the limit of the sequence $\{\sigma^*_n\}_{n\geq 0}$. More precisely, for every partial play ρ^* that ends in a position of Verifier, let $\sigma^*_U(\rho^*) := \sigma^*_{|\rho^*|}(\rho^*)$. Clearly, σ^*_U is uniform on all partial plays. Also, because no σ^*_n adds new sequences of colours to the outcome, neither does σ^*_U, and therefore it is winning for Verifier in $\mathcal{G}^*$.

Now, let us define a winning (uniform) strategy for Eve in $\mathcal{G}_i$. For every partial play ρ that ends in a position of color i, let $\sigma(\rho) := \sigma^*_U(\bigvee_{a\in A} \boxed{\sim}\boxed{a}X_i, \rho)$. Clearly, because σ^*_U is uniform on all partial plays, σ is uniform. It is then not hard to check that every sequence of colours induced by σ in $\mathcal{G}^{\mathrm{i}}$ is also induced by σ^*_U in $\mathcal{G}^*$, and therefore σ is winning for Eve in $\mathcal{G}_i$. This finishes the proof of Proposition 5.4, and thus also of Proposition 5.3 and Theorem 5.2. ◻

Remark 2. Some works consider notions of perfect recall where the player does not necessarily remember her own actions [6, 13]. Here, to establish Theorem 5.2 we need the assumption that Eve sees and remember her actions. If she did not, formula $\mathrm{WinParity}^A_k$ would be too strong a requirement: allowing the $\boxed{\sim}$ operator in the formula to range over partial plays with the same sequence of observations but different actions amounts to require the existence of a winning strategy even from equivalent partial plays that *do not* follow the strategy being built by the evaluation of the formula. This is clearly more demanding than what the existence of a winning strategy challenges.

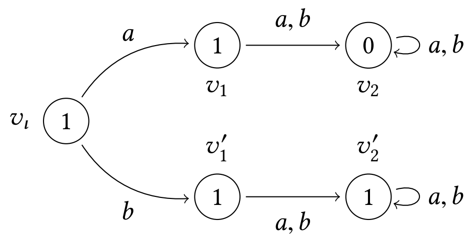

Fig. 1. Example of a parity game with imperfection information. Parities are indicated inside the positions.

For instance, consider the game in Figure 1, where a position's observation is its parity, and assume that Eve has perfect recall but does not remember her actions; in particular the two plays $v_\iota a v_1$ and $v_\iota b v_1'$ are $\sim$-equivalent. Eve clearly has a winning strategy by playing a in the initial position v_ι. However $\text{WinParity}_1^A = \nu X_0.\mu X_1.(0 \wedge \bigvee_{a\in A} \boxed{\sim}\boxed{a} X_0) \vee (1 \wedge \bigvee_{a \in A} \boxed{\sim}\boxed{a} X_1)$ does not hold in v_ι: indeed, in the evaluation game for WinParity_1^A, even if Verifier chooses the winning action a in position $(\bigvee_{a\in A} \boxed{\sim}\boxed{a} X_1, v_\iota)$, the game reaches position $(X_1, v_\iota a v_1)$ and then, assuming as usual that Verifier chooses the right colour for v_1, Refuter can choose to reach position $(\bigvee_{a\in A} \boxed{\sim}\boxed{a} X_1, v_\iota a v_1)$; from there, no matter what action Verifier chooses, Refuter can win. Assume that Verifier chooses $c \in \{a, b\}$. The game goes to position $(\boxed{\sim}\boxed{c} X_1, v_\iota a v_1)$. Because we have assumed that Eve does not remember her actions, we have that $v_\iota a v_1$ is equivalent to $v_\iota b v_1'$, and thus Refuter can move to $(\boxed{c} X_1, v_\iota b v_1')$. It is then easy to see that Refuter can force the game to stabilise among positions whose second component ends in v_2', the colour of which is 1, so that Refuter wins.

The assumption that Eve remembers her actions is used in the proof of Lemma 5.5 (Appendix A).

On the other hand, Theorem 5.7 also holds in the case where Eve does not remember her actions, as its proof adapts very easily. In fact the only part to change in the proof is the third case of the **Zig** part in the proof of Lemma 5.10, which is simpler if Eve does not remember her actions.

We now turn to the general case, where the winning condition is not necessarily consistent with the observations.

5.3 Non-observable winning conditions: undefinability in the jumping μ-calculus

In contrast with the result presented in the previous section, for non-observable reachability winning conditions we establish the following result:

Theorem 5.7. *For finite subsets $A \subset \mathcal{A}ct$ and $O \subset \mathcal{O}bs$, if $|A| \geq 2$ then the subclass of $\mathcal{R}(A, O)$ where Eve has a winning strategy is not $L_\mu^\sim$-definable.*

Before establishing Theorem 5.7, we show how it entails Theorem 3.7: First, observe that the synchronous perfect-recall relation $\sim$ is regular (a one-state transducer that accepts it can easily be exhibited). Second, by Proposition 5.1, the class $\mathcal{R}(A, O)$ considered in Theorem 5.7 is closed under bisimulation. Also, observe that the existence of a winning strategy is expressible in MSO, and that with the path relation one can express that a strategy is uniform (the reader may refer to the next section for details). The sub-class of $\mathcal{R}(A, O)$ where Eve wins is thus definable in MSO$^\sim$ and invariant under bisimulation, but by Theorem 5.7 it is not $\mathsf{L}_\mu^\sim$-definable, hence Theorem 3.7.

Proof of Theorem 5.7. Assume that A contains at least two actions, a_0 and a_1. The proof is dealt with by contradiction: Assume that there is a formula $\Phi_{\text{Win}} \in \mathsf{L}_\mu^\sim$ such that for every $\mathcal{G}^{\text{i}} \in \mathcal{R}(A, O)$,

$\mathcal{G}^i \models^{\sim} \Phi_{\text{Win}}$ if, and only if, Eve has a winning strategy in $\mathcal{G}^i$. By Proposition 4.2, there is a JTA $\mathcal{A} = (AP, Q, q_\iota, \delta, C)$ such that $\mathcal{L}(\Phi_{\text{Win}}, \sim) = \mathcal{L}(\mathcal{A}, \sim)$. Let $N := |Q| + 1$.

Outline. We describe 2^N (tree unfoldings of) reachability games in $\mathcal{R}(A, O)$, written t_i. In each of them, Eve has a winning strategy, thus each t_i is accepted by $\mathcal{A}$ and we can choose a winning strategy σ_i for Verifier in $\mathcal{G}^{t_i}_{\mathcal{A},\sim}$, the acceptance game of $\mathcal{A}$ on t_i (see Lemma 5.8). We then employ the "pigeon hole" principle to show that there exist two trees t_i and t_j that we can combine into a new tree t_0 such that:

- Eve has no winning strategy in t_0, and
- σ_i and σ_j can be combined into a winning strategy σ_0 for Verifier in $\mathcal{G}^{t_0}_{\mathcal{A},\sim}$ (Proposition 5.9).

These two points provide the desired contradiction. We now present the details of the proof.

Family of reachability games. The family of game unfoldings that we consider is depicted in Figure 2. Formally, we only describe finite trees, but the full unfoldings of games that we aim at defining are easily obtained by adding a_0-loops on leaves and by unfolding them. In these games, Eve is *blind* as all positions at a given depth have the same observation, either o_1 in the first two levels or o_2 below.

For each $i \in \{1, \dots, 2^N\}$, the tree $t_i = (\tau_i, m_i)$ is given by:

(1) $m_i(\epsilon) = (a_0, \{o_1\})$ (Recall that the action in the marking of the root is meaningless).
(2) In node ϵ, there are $2^N + 2$ a_0-children, and no a_1-child. The 2^N leftmost ones are in W, but not the two rightmost ones. Formally, $\tau_i \cap \mathbb{N} = \{0, \dots, 2^N + 1\}$. For readability, we call x_{m+1} the node m for each $m \in \{0, \dots, 2^N + 1\}$ (see Figure 2). Regarding markings, for $1 \le k \le 2^N$, $m_i(x_k) = (a_0, \{o_1, W\})$, and for $k \in \{2^N + 1, 2^N + 2\}$, $m_i(x_k) = (a_0, \{o_1\})$.
(3) For $1 \le k \le 2^N + 2$, node x_k has one a_0-child $y_k = x_k \cdot 0$, outside W: for $1 \le k \le 2^N + 2$, $m_i(y_k) = (a_0, \{o_1\})$.
(4) For each $1 \le k \le 2^N + 2$, the subtree $[t_i]_{y_k}$ is a full binary tree of height N in which each non-leaf node $y \ge y_k$ has an a_0-child (in direction 0) and an a_1-child (in direction 1). The markings are as follows. Regarding actions, for $1 \le k \le 2^N + 2$ and $w \in \{0,1\}^{\le N}$, the action in $y_k \cdot w$ is $a_i^{y_k \cdot w} := a_c$, where c is the last letter of w. Now, concerning labellings, for each $1 \le k \le 2^N$, we let $w_k \in \{0,1\}^N$ be the binary representation of $k - 1$. Then, for $w \in \{0,1\}^{\le N}$ and $1 \le k \le 2^N$, if $w = w_k$ we let $\ell_i^{y_k \cdot w} = \{o_2, W\}$, and $\ell_i^{y_k \cdot w} = \{o_2\}$ otherwise. Regarding subtrees at node y_k with $k \in \{2^N + 1, 2^N + 2\}$, if $w = w_i$ we let $\ell_i^{y_k \cdot w} = \{o_2, W\}$, and $\ell_i^{y_k \cdot w} = \{o_2\}$ otherwise.

Observe that for all $i, j \in \{1, \dots, 2^N\}$, t_i and t_j share the same underlying tree, that we shall write τ: $\tau_i = \tau_j = \tau$. Observe also that for all $1 \le k \le 2^N$, the subtrees $[t_i]_{y_k}$ are identical for all i: in these subtrees, the only node in W is $y_k \cdot w_k$. Between the different trees t_i, the markings only differ on the leaves of $[\tau]_{y_{2^N+1}}$ and $[\tau]_{y_{2^N+2}}$: for $1 \le i \le 2^N$, in $[t_i]_{y_{2^N+1}}$ and $[t_i]_{y_{2^N+2}}$, the only nodes in W are $y_{2^N+1} \cdot w_i$ and $y_{2^N+2} \cdot w_i$. Finally, remark that, since Eve is blind, her strategies are simply described by (infinite) sequences of actions.

For each $1 \le i \le 2^N$, write $\mathcal{G}_i = (V^i, v^i_\iota, E^i, V^i_1, V^i_2, C^i)$ for $\mathcal{G}^{t_i}_{\mathcal{A},\sim}$, *i.e.*, the (perfect information) acceptance game of $\mathcal{A}$ on t_i with relation $\sim$.

Lemma 5.8. *For all $1 \le i \le 2^N$, Verifier wins the game $\mathcal{G}_i$.*

Proof. Let $i \in \{1, \dots, 2^N\}$. The sequence of actions $a_0 a_0 w_i a_0^\omega$ is a winning strategy of Eve in t_i. Thus $t_i \models^{\sim} \Phi_{\text{Win}}$, *i.e.* $t_i \in \mathcal{L}(\mathcal{A}, \sim)$, and therefore Verifier has a winning strategy in $\mathcal{G}_i$. □

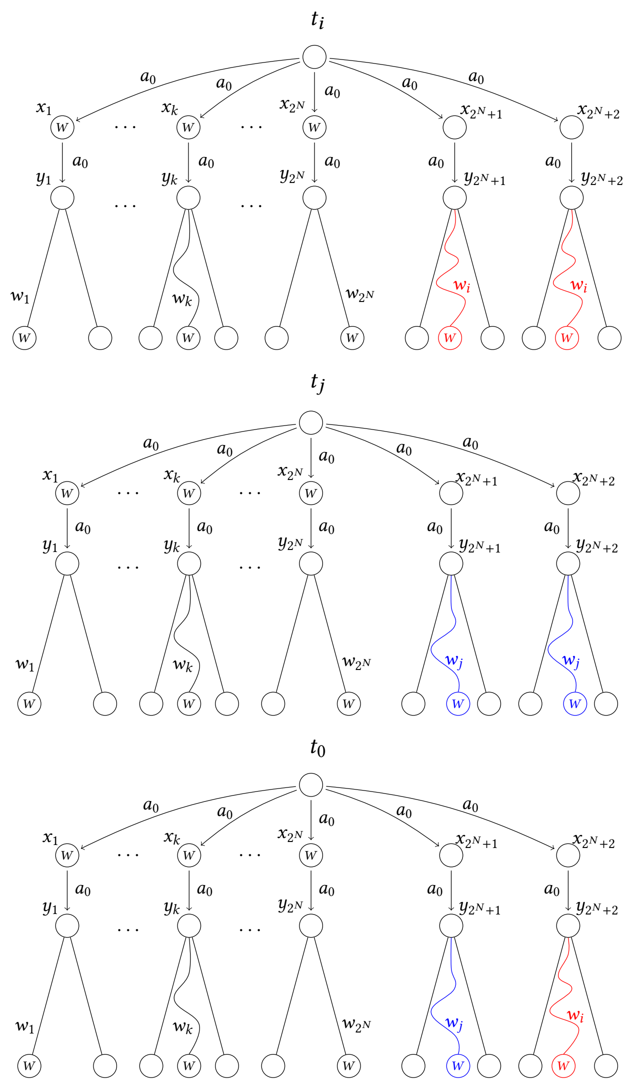

Fig. 2. The tree t_i, the tree t_j, and the hybrid tree t_0.

For each $1 \leq i \leq 2^N$, choose a winning strategy σ_i for Verifier in $\mathcal{G}_i$; this is possible as, by Lemma 5.8, there exists one for each i. For every $1 \leq i \leq 2^N$, we also define the function $\text{visit}_{\sigma_i} : \tau \to 2^Q$ that associates to each node x of τ the set of states in which $\mathcal{A}$ can visit x when Verifier follows σ_i in the acceptance game $\mathcal{G}_i$; formally:

$$\text{visit}_{\sigma_i}(x) := \{q \mid \exists \pi \in \text{Out}(\mathcal{G}_i, \sigma_i), \exists n \geq 0, \exists \alpha \in \mathcal{B}^+(Dir \times Q) \text{ s.t. } \pi[n] = (x, q, \alpha)\}.$$

Pigeon hole and combined tree t_0. Since there are at most $2^{|Q|}$ different such sets of states, and we have 2^N strategies with $N = |Q| + 1$, there must exist $i \neq j$ such that $\text{visit}_{\sigma_i}(y_{2^N+1}) = \text{visit}_{\sigma_j}(y_{2^N+1})$. For the rest of the proof we fix such a pair (i, j). We define the game unfolding t_0, obtained from t_i by replacing the subtree $[t_i]_{y_{2^N+1}}$ with $[t_j]_{y_{2^N+1}}$ (see Figure 2). Note that t_0 is the unfolding of a game in $\mathcal{R}(A, O)$ and that by design of t_0, Eve has no winning strategy in t_0. Indeed, the same sequence of actions cannot lead to a node marked by W in both subtrees $[t_0]_{y_{2^N+1}}$ and $[t_0]_{y_{2^N+2}}$. Therefore, $t_0 \not\models^{\sim} \Phi_{\text{Win}}$, and thence $t_0 \notin \mathcal{L}(\mathcal{A}, \sim)$, *i.e.*, Verifier does not have a winning strategy in the acceptance game $\mathcal{G}^{t_0}_{\mathcal{A},\sim}$ of JTA $\mathcal{A}$ on t_0 with path relation $\sim$.

We now establish Proposition 5.9 below, which provides a contradiction with the latter and terminates the proof of Theorem 5.7.

Proposition 5.9. *Verifier has a winning strategy in $\mathcal{G}^{t_0}_{\mathcal{A},\sim}$.*

Proof. First, let us write $\mathcal{G}_0 = (V^0, v^0_\iota, E^0, V^0_1, V^0_2, C^0)$ for $\mathcal{G}^{t_0}_{\mathcal{A},\sim}$, and observe that the three games $\mathcal{G}_i$, $\mathcal{G}_j$ and $\mathcal{G}_0$ share the same set of positions: $V^0 = V^i = V^j = \tau \times Q \times \mathbb{B}^+(Dir \times Q)$ that we now write V, partitioned between V_1 and V_2. Also, for all $1 \leq k \leq 2^N + 2$ we have $\ell^{y_k}_0 = \ell^{y_k}_i = \ell^{y_k}_j$ $(= \{o\})$, that we now write ℓ. Because positions of the form $(y_k, q, \delta(q, \ell))$ play an important role in the following, we succinctly write them v^q_k. We start with the following lemma, which will allow us to transfer the existence of winning strategies in positions of the form v^q_k from $\mathcal{G}_i$ and $\mathcal{G}_j$ to $\mathcal{G}_0$:

Lemma 5.10.

(1) *For all $q \in Q$ and $k \neq 2^N + 1$, $(\mathcal{G}_0, v^q_k) \Leftrightarrow (\mathcal{G}_i, v^q_k)$, and*
(2) *for all $q \in Q$ and $k \neq 2^N + 2$, $(\mathcal{G}_0, v^q_k) \Leftrightarrow (\mathcal{G}_j, v^q_k)$.*

The proof of this lemma can be found in Appendix C.

Winning strategy for Verifier in $\mathcal{G}_0$: intuition. Let us define $\text{Start}_\tau := \{\epsilon, x_1, \ldots, x_{2^N+2}\}$, the two first levels of τ, and $\text{Start}_\mathcal{G} := \{(x, q, \alpha) \in V \mid x \in \text{Start}_\tau\}$. Observe that every play in $\mathcal{G}_0$ starts in $v_\iota = (\epsilon, q_0, \delta(q_0, \ell^\epsilon_0))$, hence in $\text{Start}_\mathcal{G}$. Note that a play may remain in $\text{Start}_\mathcal{G}$ for ever if it keeps jumping without going down. Otherwise, it exits $\text{Start}_\mathcal{G}$ by reaching some node y_k, in position v^q_k for some q. Observe also that from any position of $\text{Start}_\mathcal{G}$, the set of moves available in $\mathcal{G}_0$ and in $\mathcal{G}_i$ (and in $\mathcal{G}_j$) are the same. In $\mathcal{G}_0$, we let Verifier follow σ_i as long as the game is in $\text{Start}_\mathcal{G}$. If the game remains in $\text{Start}_\mathcal{G}$ for ever, the obtained play is an outcome of σ_i, which is winning for Verifier in $\mathcal{G}_i$. Because positions have the same colour in all acceptance games, this play is also winning for Verifier in $\mathcal{G}_0$. Otherwise, the play exits $\text{Start}_\mathcal{G}$ by going down the tree, hence it reaches some position v^q_k. Because v^q_k has been reached by the winning strategy σ_i, it is a winning position for Verifier in $\mathcal{G}_i$. If $k \neq 2^N + 1$, by Point 1 of Lemma 5.10, $(\mathcal{G}_0, v^q_k) \Leftrightarrow (\mathcal{G}_i, v^q_k)$, and by Proposition 5.1, Verifier also has a winning strategy from v^q_k in $\mathcal{G}_0$. If $k = 2^N + 1$, because $\text{visit}_{\sigma_i}(y_{2^N+1}) = \text{visit}_{\sigma_j}(y_{2^N+1})$, σ_j also visits position $v^q_{2^N+1}$, and therefore $v^q_{2^N+1}$ is a winning position for Verifier in $\mathcal{G}_j$. By Point 2 of Lemma 5.10, $\mathcal{G}_0, v^q_k \Leftrightarrow \mathcal{G}_j, v^q_k$, and by Proposition 5.1, Verifier also has a winning strategy from v^q_k in $\mathcal{G}_0$.

Winning strategy for Verifier in $\mathcal{G}_0$: formal definition. First, for each position of the form v^q_k, if v^q_k is a winning position for Verifier in $\mathcal{G}_0$, we pick a winning strategy for Verifier in $(\mathcal{G}_0, v^q_k)$ that we call $\sigma_{v^q_k}$. Recall that $\text{Start}_\tau = \{\epsilon, x_1, \ldots, x_{2^N+2}\}$ consists in the two first levels of τ, and $\text{Start}_\mathcal{G} = \{(x, q, \alpha) \in V \mid x \in \text{Start}_\tau\}$. Take a partial play ρ in $\mathcal{G}_0$ ending in a position of Verifier.

- If $\rho \in \text{Start}^*_\mathcal{G}$, let $\sigma_0(\rho) := \sigma_i(\rho)$.
- Otherwise, there exist ρ', k, q and ρ'' such that $\rho = \rho' \cdot v^q_k \cdot \rho''$, and $\rho' \in \text{Start}^*_\mathcal{G}$. Then:
  - If v^q_k is a winning position for Verifier in $\mathcal{G}_0$, then $\sigma_{v^q_k}$ is defined, and we let
  $$\sigma_0(\rho) := \sigma_{v^q_k}(v^q_k \cdot \rho'').$$
  - Otherwise, define $\sigma_0(\rho)$ arbitrarily.

Proof that σ_0 is winning for Verifier in $\mathcal{G}_0$. Let $\pi \in \text{Out}(\mathcal{G}_0, \sigma_0)$. If $\pi \in \text{Start}_{\mathcal{G}}^{\omega}$, because the labelings of t_0 and t_i are the same on Start_{τ}, π is also a play in $\mathcal{G}_i$. Moreover, because σ_0 is defined as σ_i on $\text{Start}_{\mathcal{G}}$ and π follows σ_0, we have that π follows σ_i, which is winning for Verifier in $\mathcal{G}_i$, so π is winning for Verifier in $\mathcal{G}_0$ (recall that positions have the same colours in the different acceptance games). Otherwise, there exist ρ, k, q and π' such that $\pi = \rho \cdot v_k^q \cdot \pi'$ and $\rho \in \text{Start}_{\mathcal{G}}^*$. Again, $\rho \cdot v_k^q$ is also a partial play in $\mathcal{G}_i$, and it follows σ_i. Because σ_i is winning for Verifier in $\mathcal{G}_i$, we have that v_k^q is a winning position in $\mathcal{G}_i$. We distinguish two cases.

- $k \neq 2^N+1$: since v_k^q is a winning position for Verifier in $\mathcal{G}_i$, by Lemma 5.10 and Proposition 5.1, v_k^q is also a winning position for Verifier in $\mathcal{G}_0$.
- $k = 2^N + 1$: By definition, we have $q \in \text{visit}_{\sigma_i}(y_{2^N+1})$. Because $\text{visit}_{\sigma_i}(y_{2^N+1}) = \text{visit}_{\sigma_j}(y_{2^N+1})$, there exists an outcome of σ_j in $\mathcal{G}_j$ that visits $v_{2^N+1}^q$. Because σ_j is winning for Verifier in $\mathcal{G}_j$, we get that $v_{2^N+1}^q$ is a winning position in $\mathcal{G}_j$. Again, by Lemma 5.10 and Proposition 5.1, we obtain that $v_{2^N+1}^q = v_k^q$ is also a winning position for Verifier in $\mathcal{G}_0$.

In both cases, $\sigma_{v_k^q}$ is defined, and by definition of σ_0, we have that $v_k^q \cdot \pi' \in \text{Out}((\mathcal{G}_0, v_k^q), \sigma_{v_k^q})$. Because $\sigma_{v_k^q}$ is winning for Verifier in $(\mathcal{G}_0, v_k^q)$, $v_k^q \cdot \pi'$ verifies the parity condition, and therefore also does $\pi = \rho \cdot v_k^q \cdot \pi'$. So π is winning for Verifier, which concludes the proof of Proposition 5.9. ◻

Remark 3. The whole proof would also work with only one observation shared by all positions. To comply with the requirement that positions with the same observation must have the same set of available actions, we have to either use two different observations, or add dummy moves, but we find the latter solution to be more cumbersome.

Also, observe that when there is only one action, say $A = \{a\}$, then in every game in $\mathcal{R}(A, O)$ with reachability winning condition W there is only one possible (uniform) strategy for Eve, which is to always play a. This strategy is winning if, and only if, every path from the initial position visits W, which is expressible by the μ-calculus formula $\mu X.(W \vee \boxed{a} X)$.

6 IMPACT

We discuss the impact of our results, first on the (im)possibility to project jumping automata, and second on the (im)possibility to characterise strategic operators from ATL with imperfect information via fixed points.

6.1 Projection of jumping tree automata

We first recall the projection operation for tree automata, introduced by [40]: Given a nondeterministic tree automaton $\mathcal{A}$ over $AP \subset \mathcal{AP}$ and an atomic proposition $p \in AP$, one can build a non-deterministic tree automaton $\mathcal{A}'$ over $AP \setminus \{p\}$ that accepts a marked tree if, and only if, it is the projection on $AP \setminus \{p\}$ of some tree in $\mathcal{L}(\mathcal{A})$. This operation captures the existential second-order quantification of MSO.

This projection operation does not apply to alternating tree automata. However, the Simulation Theorem of [37], allows one to first nondeterminise alternating tree automata, and then project the obtained nondeterministic automata.

Since jumping tree automata generalise alternating tree automata, the projection operation does not apply to them. Relying on Theorem 3.7 we prove that, unlike the case of alternating automata, there is no hope of exploiting some sort of Simulation theorem to project jumping automata.

Corollary 6.1. *The class of jumping automata equipped with the synchronous perfect recall relation is not closed under projection.*

Proof. We exhibit a jumping automaton $\mathcal{A}_X$ whose language has a projection that cannot be described by any jumping automaton. Let us fix a finite set of actions $A \subset \mathcal{A}ct$ such that $|A| > 1$, as well as a finite set of observations $O \subset \mathcal{O}bs$. The automaton $\mathcal{A}_X$ arises from our construction from Theorem 5.7: it is meant to read unfoldings of two-player reachability games with imperfect information from $\mathcal{R}(A, O)$, marked with an extra proposition X, and it is equipped with the synchronous perfect recall relation $\sim$ (see Section 4.1). Automaton $\mathcal{A}_X$ thus runs on alphabet $AP = O \cup \{W, X\}$, and it accepts a game unfolding whenever proposition X marks a subtree t_X that describes a winning strategy for Eve. Actually, instead of explicitly describing $\mathcal{A}_X$, we define a jumping μ-calculus formula φ_X, and by Proposition 4.2 we obtain $\mathcal{A}_X$ such that $\mathcal{L}(\varphi_X, \sim) = \mathcal{L}(\mathcal{A}_X, \sim)$. Formula φ_X is the conjunction of three properties: the first one, Strat_X, states that t_X characterises a strategy for Eve[5]; the second one, Unif_X, states that the strategy described by t_X is uniform; lastly, Win_X states that the strategy described by t_X is winning for Eve, *i.e.*, that every path of t_X eventually meets W.

Formally, we let $\varphi_X := \mathrm{Strat}_X \wedge \mathrm{Unif}_X \wedge \mathrm{Win}_X$ where:

$$\begin{cases} \mathrm{Strat}_X & := X \wedge \nu Z.[X \longrightarrow \bigvee_{a\in A}(\boxed{a}X \wedge \bigwedge_{b\neq a} \boxed{b}\neg X \wedge \boxed{a}Z)] \\ \mathrm{Unif}_X & := \nu Z.[X \longrightarrow (\bigvee_{a\in A} \boxed{\sim}\boxed{a}X) \wedge \bigwedge_{a\in A} \boxed{a}Z] \\ \mathrm{Win}_X & := \mu Z.[W \vee \bigwedge_{a\in A} \boxed{a}(X \longrightarrow Z)] \end{cases}$$

Because the jumping μ-calculus translates into $\mathsf{MSO}^{\sim}$ there exists a formula $\psi(X) \in \mathsf{MSO}^{\sim}$ which is equivalent to φ_X, thus characterised by $\mathcal{A}_X$. Now, it is clear that the $\mathsf{MSO}^{\sim}$ formula $\exists X \psi(X)$ captures the subclass of $\mathcal{R}(A, O)$ where there exists a winning strategy for Eve. If the projection of automaton $\mathcal{A}_X$ onto the propositional alphabet $O \cup \{W\}$ (abstracting from X) were a jumping automaton, this subclass would be $\mathsf{L}_{\mu}^{\sim}$-definable, which would contradict Theorem 5.7. □

Notice that Corollary 6.1 is not that surprising a result. When considering classic non-deterministic automata, projection can be achieved by guessing the missing label when a node of the input tree is visited; this process is sound since any run of the automaton visits each node at most once. On the contrary, for jumping tree automata, the classic notion of non-determinism would not make this process sound because jumps may still allow to visit several times the same node.

6.2 ATL with imperfect information

The second impact concerns the place of logics of coalitions and strategies among logics of programs, and in particular their relationship with the epistemic μ-calculus. One of the most influential logics for strategic abilities is Alternating-time Temporal Logic (ATL), introduced by [2]. The models of this logic are concurrent game structures (CGSs), which are transition systems where actions are tuples of moves, one for each agent in a fixed finite set Ag. The interpretation is that in each round, each agent (or player) chooses a move, and Nature chooses a state attainable from the current state through this tuple of moves. The syntax is essentially that of CTL where the existential path quantifier is replaced with the strategy quantifier $\langle\!\langle A \rangle\!\rangle$, where $A \subseteq Ag$ is a coalition of agents. The intuitive meaning of formula $\langle\!\langle A \rangle\!\rangle\varphi$ is "agents in A each have a strategy so that together they enforce that φ will hold whatever the other agents do". It is well-known that, in the case of perfect information, ATL can be encoded in the μ-calculus (assuming a fixed finite set of possible moves for a fixed finite set of agents), and the fact that the μ-calculus captures many different logics such

[5] In fact it only describes the definition of the strategy on its set of outcomes, which is enough here.

as ATL or the dynamic logic of [22] (see [28]) contributes to its predominant position among logics of programs.

$\mathsf{ATL_i}$ **and epistemic μ-calculus.** We explain how our expressivity incompleteness result of Theorem 3.7 establishes that this remarkable situation is shaken in the case of imperfect information. ATL with imperfect information ($\mathsf{ATL_i}$), as defined in [2], is the same as ATL except that models are concurrent game structures with imperfect information ($\mathsf{CGS_i}$), in which each agent has a partial observation of the game structure modelled by a mapping from states to a set of observations. As a consequence, strategies for an agent are required to be uniform with regards to her observation. The existence of a winning strategy for Eve in a two-player reachability game with imperfect information and synchronous perfect recall can thus be expressed in $\mathsf{ATL_i}$ by the formula $\langle\langle \text{Eve} \rangle\rangle \mathbf{F} W$ (where W is the reachability winning condition). But by Theorem 5.7, it cannot be expressed in the epistemic μ-calculus. It follows that in the imperfect-information setting with synchronous perfect recall, $\mathsf{ATL_i}$ is not subsumed by the epistemic μ-calculus, which we argue is the most natural extension of the μ-calculus to imperfect information. Also, observe that the proof of Theorem 5.7 considers games over a fixed set of two actions.

Corollary 6.2. $\mathsf{ATL_i}$ *with synchronous perfect recall is not subsumed by the epistemic μ-calculus with synchronous perfect recall, even when restricted to CGS_i over a fixed finite set of actions of size at least two.*

$\mathsf{ATL_i}$ **and expansion laws.** Theorem 5.7 also impacts the possibility to characterise ATL strategic operators as smallest or greatest fixed points. It is well known that in the perfect-information case, combinations of strategic and temporal operators admit expansion laws: for instance, $\langle\langle A \rangle\rangle \mathbf{F} p$ is equivalent to $p \vee \langle\langle A \rangle\rangle \mathbf{X} \langle\langle A \rangle\rangle \mathbf{F} p$. These expansion laws can be turned into equivalent fixed point formulas in the Alternating-time μ-Calculus (AMC), introduced in [2]. AMC is essentially an adaptation of the μ-calculus to concurrent game structures, where standard modalities are replaced with "one-step" strategic modalities: modalities are of the form $\langle A \rangle$, where $A \subseteq Ag$ is a coalition of agents, and the meaning of $\langle A \rangle p$ is "agents in A have a move to enforce that in the next step, p holds". It is thus equivalent to the ATL formula $\langle\langle A \rangle\rangle \mathbf{X} p$. According to the above expansion law, the ATL formula $\langle\langle A \rangle\rangle \mathbf{F} p$ is thus equivalent to the AMC formula $\mu X.p \vee \langle A \rangle X$, and in fact ATL translates into AMC [2], which shows that in a sense strategies in ATL can be computed one step at a time. Also, when axiomatising ATL, such expansion laws provide useful fixed-point axioms.

Theorem 5.7 allows us to establish that this sort of fixed-point characterisation does not exist in the context of imperfect information: if there were, they could be translated into AMC with imperfect information ($\mathsf{AMC_i}$), which we show cannot be done. To the best of our knowledge, the only existing semantics for $\mathsf{AMC_i}$ is defined in [9], where $\mathsf{ATL_i}$ is compared to $\mathsf{AMC_i}$ in a setting where, unlike our perfect recall setting, agents have no memory at all and thus can only base their actions on their observation of the current system's state. In their semantics, $\langle A \rangle \varphi$ holds in a state s if agents in A have a joint move that, from any state observationally equivalent to s for any of the agents in A, ensures that φ holds in the next step whatever the other agents do. This is meant to help capture the existence of uniform strategies used in $\mathsf{ATL_i}$ without memory. However they show that this is not enough: $\mathsf{ATL_i}$ without memory cannot be expressed in $\mathsf{AMC_i}$ without memory. Actually they show the stronger result that this is also the case when $\mathsf{AMC_i}$ is enriched with knowledge operators for each agent, obtaining the Alternating-time Epistemic μ-calculus with imperfect information ($\mathsf{AEMC_i}$): for memoryless semantics, $\mathsf{AEMC_i}$ does not subsume $\mathsf{ATL_i}$. This implies that in the memoryless setting, the $\mathsf{ATL_i}$ formula $\langle\langle A \rangle\rangle \mathbf{F} p$ does not admit any "expansion law" like it does in the perfect information case, even if knowledge operators can be used: if it did, it could be turned into an equivalent $\mathsf{AEMC_i}$ formula.

What about agents with perfect recall? In [9] a comparison between $\mathsf{ATL_i}$ with perfect recall and $\mathsf{AEMC_i}$ without memory show that the latter does not capture the former. We argue that this is not very surprising, since there is no operator in $\mathsf{AEMC_i}$ without memory that could possibly capture the notion of uniformity for strategies of agents with perfect recall. We propose here to show that even when $\mathsf{AEMC_i}$ is given perfect recall semantics, it still does not capture $\mathsf{ATL_i}$ with perfect recall. We recall that $\sim_a$ is the indistinguishability equivalence over finite plays for agent a with synchronous perfect recall, as defined in Section 5.1, and we propose the following natural semantics for $\mathsf{AEMC_i}$ with synchronous perfect recall. We define the semantics of the one-step strategy quantifier as follows: formula $\langle A\rangle\varphi$ holds in a *partial play* ρ if there is a joint move for agents in A that ensures to have φ in the next step from *any partial play* ρ' such that $\rho \sim_a \rho'$ for some $a \in A$. In a similar way, the semantics of the epistemic operator K_a for agent $a \in Ag$ is defined as follows: $K_a\varphi$ holds in a partial play ρ if φ holds in all partial plays $\rho' \sim_a \rho$.

We first establish the following lemma.

Lemma 6.3. *If we fix a finite set of moves Mov, the epistemic μ-calculus subsumes* $\mathsf{AEMC_i}$ *over the class of concurrent game structures over Mov.*

Proof. Assuming a fixed finite set of moves Mov for all agents and all concurrent game structures, the one-step strategic operator of $\mathsf{AEMC_i}$ can easily be expressed in epistemic μ-calculus (recall that we assumed a fixed finite set of agents Ag):

$$\langle A\rangle p \equiv \bigvee_{\vec{m}\in Mov^A} \bigwedge_{a\in A} K_a \bigwedge_{\vec{m}'\in Mov^{Ag\setminus A}} \boxed{b}p, \qquad \text{where } b = (\vec{m},\vec{m}').$$

All other operators of $\mathsf{AEMC_i}$, including knowledge operators, can be trivially translated into epistemic μ-calculus. □

We now show how Theorem 5.7 entails the following result:

Corollary 6.4. $\mathsf{ATL_i}$ *with synchronous perfect recall is not subsumed by* $\mathsf{AEMC_i}$ *with synchronous perfect recall.*

Proof. As already mentioned, the property $\mathcal{P}$ saying that there exists a winning strategy for Eve in a two-player reachability game with imperfect information and synchronous perfect recall can be expressed in $\mathsf{ATL_i}$ with synchronous perfect recall with the formula $\Phi = \langle\!\langle Eve\rangle\!\rangle \mathbf{F}W$, where W marks the winning positions.

We show that this formula with synchronous perfect recall semantics cannot be expressed in $\mathsf{AEMC_i}$ with synchronous perfect recall semantics, which concludes the proof. Assume towards a contradiction that there is such an $\mathsf{AEMC_i}$ formula $\Psi \equiv \Phi$. Let us fix a finite set of moves Mov such that $|Mov| > 1$, and let C_{Mov} be the class of $\mathsf{CGS_i}$ over Mov. By Lemma 6.3, there is an epistemic μ-calculus formula Ψ' that is equivalent to Ψ over C_{Mov}, which we write $\Psi \equiv_{C_{Mov}} \Psi'$. Since formulas Φ and Ψ are equivalent over the class of all concurrent game structures, in particular they are equivalent over C_{Mov} and thus Ψ' is equivalent to Φ over C_{Mov}:

$$\Phi \equiv_{C_{Mov}} \Psi'. \tag{1}$$

On the other hand, since $|Mov| > 1$ we have by Theorem 5.7 that property $\mathcal{P}$ over C_{Mov} cannot be expressed in the epistemic μ-calculus over C_{Mov}. Since the $\mathsf{ATL_i}$ formula Φ captures property $\mathcal{P}$ over all concurrent game structures, and in particular over C_{Mov}, we obtain that Φ does not have a translation in epistemic μ-calculus over C_{Mov}, which contradicts Equation (1). □

We remark that in [9], the proof that $\mathsf{ATL_i}$ with perfect recall is not subsumed by $\mathsf{AEMC_i}$ without memory relies on the fact that the one-step strategic operator and the knowledge operators are memoryless, and it does not adapt to $AEMC_i$ with the perfect recall semantics we propose here. It thus seems that a different proof technique was required to establish Corollary 6.4, and we provided one using epistemic μ-calculus as intermediary logic and jumping automata as a useful tool to establish the impossibility to express a simple ATL_i formula in this logic.

Robustness to semantic variants. The simplicity of the formula we consider for the inexpressivity proof of Theorem 5.7, as well as the shape of the family of game arenas we consider, also make our result quite robust to variants in the semantics of ATL_i. Indeed, there exist in the literature various semantics for the strategic operator of ATL_i. In the basic one that we considered so far, $\langle\langle A\rangle\rangle\varphi$ holds if there exists a strategy profile for A such that φ holds in all outcomes from the current position [2]. In the *de dicto* semantics, there must be one such strategy profile from each position indistinguishable to the current one for one of the agents in A, and in the *de re* semantics the same strategy profile must work from each such indistinguishable position (see [26] for a detailed discussion on the matter). In all these semantics, it is usually considered that when a new strategy quantifier is met, all agents forget the past, as well as their previous strategies. Instead, the recent *no forgetting* semantics considers that agents never forget the past [10]. ATL with *strategy context* goes further by allowing agents to keep their previously assigned strategies when evaluating a strategy quantifier in which they are not involved [29, 30].

Observe that in the proof of Corollary 6.4, the ATL_i formula that we prove not to be expressible in $AEMC_i$, $\Phi = \langle\langle Eve\rangle\rangle \mathbf{F} W$, does not have nested strategy quantifiers, hence its meaning is the same in the classic semantics, no forgetting or strategy context. In addition, in all game arenas considered in the proof of Theorem 5.7 the initial state does not have any other indistinguishable state for Eve. The meaning of Φ on these CGS_i is thus also independent from the choice of the basic, de dicto or de re semantics. It follows that for neither of these semantics does Φ have a translation in the epistemic μ-calculus with synchronous perfect recall, as such a translation would be equivalent to Φ on the restricted class of games considered in the proof of Theorem 5.7.

Corollary 6.5. *Corollary 6.4 holds for the basic,* de dicto *and* de re *semantics of* ATL_i *with synchronous perfect recall, with or without* no forgetting *or* strategy context.

Corollaries 6.4 and 6.5 for the case of perfect recall, together with the result from [9] for the case of memoryless agents, settle in the negative the question of whether ATL with imperfect information admits expansion laws as in the perfect information setting.

7 CONCLUSION AND PERSPECTIVES

We have developed a general setting based on transition systems equipped with path relations, which, for particular relations, captures models of agency with time and knowledge.

Inspired by the seminal expressive completeness result of Janin and Walukiewicz [27], which states an equal expressivity of the bisimulation-invariant fragment of MSO and of the classic μ-calculus, we have proposed adapted extensions of MSO and of the μ-calculus to the setting of transition systems with path relations: MSO with path relation and the jumping μ-calculus, respectively. The path relation is a parameter which gives rise to as many different logics, and, consequently, to as many different expressive completeness problems; Our results reveal that the answer to the expressive completeness problem is unsurprisingly sensitive to the choice of the path relation: whereas the answer to the expressive completeness problem is identically positive for the whole class of recognisable path relations (Theorem 3.6), it does not hold on the larger class of regular path relations (Theorem 3.7).

Noticeably, the latter Theorem 3.7 arises from the use of the synchronous perfect recall path relation – a typical knowledge semantics in models of agency extensively studied in the literature.

We believe that it is worth studying the class of regular path relations to identify which features of the synchronous perfect recall relation causes the expressive completeness result to fail. For example, one may seek for a general argument that would extend Theorem 3.7 to some subclass of regular path relations.

Regarding our methodology to attain Theorem 3.7, it is worth noticing the role played by jumping tree automata introduced in [35], and which we proved here to be a counterpart of the jumping μ-calculus. They are essential in most of our secondary results, such as Proposition 4.4 on the complexity of the satisfiability problem for the jumping μ-calculus, and Proposition 4.5 on the succinctness of the jumping μ-calculus with regards to the μ-calculus. More importantly, they are central in the non-trivial proof of Theorem 5.7, which entails our main result, Theorem 3.7 and other consequences discussed in the previous section.

Noticeable are also our answers to the definability in the jumping μ-calculus of two-player reachability and parity games with imperfect information and synchronous perfect recall where Eve wins, in the spirit of [16] for the μ-calculus and perfect information games. We have established that whereas games of imperfect information with reachability or parity condition are definable as soon as winning conditions are observable and actions are remembered (Theorem 5.2), this is not the case anymore when observability of winning conditions is relaxed (Theorem 5.7).

We have also shown in Section 6.2 how Theorem 5.7 implies that, unlike ATL with perfect information, ATL with imperfect information does not admit expansion laws. This closes a natural question related to the axiomatisability of ATL with imperfect information, that received attention for the last few years.

It is important to remark that, as shown in the proof of Corollary 6.1, the jumping μ-calculus with synchronous perfect recall semantics is expressive enough to capture non-regular properties of strategies, such as being uniform. The crucial flaw of this logic, that makes it unfit for strategic reasoning in the imperfect information setting, is its inability to capture second-order quantification over strategies, as opposed to MSO with path relation or logics for strategic reasoning such as ATL_i. The evidence that the (jumping) μ-calculus falls short in the framework of transition systems with path relations calls for the quest for an appropriate candidate that would play the role classic μ-calculus plays in the perfect-information setting, *i.e.*, capture the bisimulation-invariant fragment of MSO with path relation. In particular, would this candidate be attained by an enrichment of the jumping μ-calculus with some kind of quantification over uniform strategies?

ACKNOWLEDGMENTS

This work has received funding from the ANR for the project EQINOCS ANR-11-BS02-004 and from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement EPS 313360).

REFERENCES

[1] R. Alur, P. Černý, and S. Chaudhuri. 2007. Model checking on trees with path equivalences. In *TACAS'07*. Springer, 664–678.

[2] R. Alur, Th. A. Henzinger, and O. Kupferman. 2002. Alternating-time temporal logic. *J. ACM* 49, 5 (2002), 672–713.

[3] J. Berstel. 1979. *Transductions and context-free languages*. Vol. 4. Teubner Stuttgart. https://doi.org/10.1007/978-3-663-09367-1

[4] D. Berwanger, K. Chatterjee, M. De Wulf, L. Doyen, and T. A. Henzinger. 2010. Strategy construction for parity games with imperfect information. *Inf. Comput.* 208, 10 (2010), 1206–1220.

[5] D. Berwanger and L. Doyen. 2008. On the Power of Imperfect Information. In *FSTTCS*. 73–82.

[6] D. Berwanger and L. Kaiser. 2010. Information Tracking in Games on Graphs. *Journal of Logic, Language and Information* 19, 4 (2010), 395–412.

[7] B. Bollig and M. Leucker. 2006. Message-passing automata are expressively equivalent to EMSO logic. *Theor. Comput. Sci.* 358, 2-3 (2006), 150–172. https://doi.org/10.1016/j.tcs.2006.01.014

[8] N. Bulling, J. Dix, and W. Jamroga. 2010. Model checking logics of strategic ability: Complexity. In *Specification and Verification of Multi-Agent Systems*, M. Dastani, K. V. Hindriks, and J.-J. C. Meyer (Eds.). Springer, 125–160.

[9] N. Bulling and W. Jamroga. 2011. Alternating Epistemic Mu-Calculus. In *Proceedings of IJCAI'2011*. IJCAI/AAAI, 109–114.

[10] Nils Bulling, Wojciech Jamroga, and Matei Popovici. 2014. ATL* With Truly Perfect Recall: Expressivity and Validities. In *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*. 177–182. https://doi.org/10.3233/978-1-61499-419-0-177

[11] D. Caucal. 2003. On infinite transition graphs having a decidable monadic theory. *Theor. Comput. Sci.* 290, 1 (2003), 79–115. https://doi.org/10.1016/S0304-3975(01)00089-5

[12] K. Chatterjee and L. Doyen. 2010. The complexity of partial-observation parity games. In *LPAR 17*. Springer, 1–14.

[13] Krishnendu Chatterjee and Laurent Doyen. 2014. Partial-Observation Stochastic Games: How to Win when Belief Fails. *ACM Trans. Comput. Log.* 15, 2 (2014), 16. https://doi.org/10.1145/2579821

[14] B. Courcelle and I. Durand. 2012. Automata for the verification of monadic second-order graph properties. *J. Applied Logic* 10, 4 (2012), 368–409. https://doi.org/10.1016/j.jal.2011.07.001

[15] B. Courcelle and J. Engelfriet. 2012. *Graph Structure and Monadic Second-Order Logic - A Language-Theoretic Approach*. Cambridge University Press. http://www.cambridge.org/fr/knowledge/isbn/item5758776/?site_locale=fr_FR

[16] A. Dawar and E. Grädel. 2008. The Descriptive Complexity of Parity Games. In *Computer Science Logic, 22nd International Workshop, CSL 2008, 17th Annual Conference of the EACSL, Bertinoro, Italy, September 16-19, 2008. Proceedings (Lecture Notes in Computer Science)*, M. Kaminski and S. Martini (Eds.), Vol. 5213. Springer, 354–368. https://doi.org/10.1007/978-3-540-87531-4_26

[17] Catalin Dima, Bastien Maubert, and Sophie Pinchinat. 2015. Relating Paths in Transition Systems: The Fall of the Modal Mu-Calculus. In *Mathematical Foundations of Computer Science 2015 - 40th International Symposium, MFCS 2015, Milan, Italy, August 24-28, 2015, Proceedings, Part I*. 179–191. https://doi.org/10.1007/978-3-662-48057-1_14

[18] S. Dziembowski, M. Jurdzinski, and I. Walukiewicz. 1997. How much memory is needed to win infinite games?. In *LICS*. IEEE, 99–110. https://doi.org/10.1109/LICS.1997.614939

[19] C. C. Elgot and M. O. Rabin. 1966. Decidability and Undecidability of Extensions of Second (First) Order Theory of (Generalized) Successor. *J. Symb. Log.* 31, 2 (1966), 169–181. https://doi.org/10.2307/2269808

[20] E. A. Emerson and C. S. Jutla. 1999. The Complexity of Tree Automata and Logics of Programs. *SIAM J. Comput.* 29, 1 (1999), 132–158. https://doi.org/10.1137/S0097539793304741

[21] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. 2004. *Reasoning about knowledge*. The MIT Press.

[22] M. J. Fischer and R. E. Ladner. 1979. Propositional dynamic logic of regular programs. *Journal of computer and system sciences* 18, 2 (1979), 194–211.

[23] E. Grädel, W. Thomas, and Th. Wilke. 2002. *Automata, Logics, and Infinite Games, volume 2500 of LNCS*. Springer Verlag.

[24] J. Y. Halpern, R. van der Meyden, and M. Y. Vardi. 2004. Complete Axiomatizations for Reasoning about Knowledge and Time. *SIAM J. Comput.* 33, 3 (2004), 674–703.

[25] J. Y. Halpern and M. Y. Vardi. 1989. The complexity of reasoning about knowledge and time. 1. Lower bounds. *J. Comp. Sys. Sci.* 38, 1 (1989), 195–237. https://doi.org/10.1145/12130.12161

[26] W. Jamroga and Th. Ågotnes. 2007. Constructive knowledge: what agents can achieve under imperfect information. *Journal of Applied Non-Classical Logics* 17, 4 (2007), 423–475.

[27] D. Janin and I. Walukiewicz. 1996. On the expressive completeness of the propositional mu-calculus with respect to monadic second order logic. In *Proceedings of CONCUR'96*. Springer, 263–277.

[28] D. Kozen. 1983. Results on the propositional μ-calculus. *Theoretical computer science* 27, 3 (1983), 333–354.

[29] François Laroussinie and Nicolas Markey. 2015. Augmenting ATL with strategy contexts. *Inf. Comput.* 245 (2015), 98–123. https://doi.org/10.1016/j.ic.2014.12.020

[30] François Laroussinie, Nicolas Markey, and Arnaud Sangnier. 2015. ATLsc with partial observation. In *Proceedings Sixth International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2015, Genoa, Italy, 21-22nd September 2015*. 43–57. https://doi.org/10.4204/EPTCS.193.4

[31] Hans Läuchli and Christian Savioz. 1987. Monadic second order definable relations on the binary tree. *The Journal of Symbolic Logic* 52, 01 (1987), 219–226.

[32] A. Lomuscio and Fr. Raimondi. 2006. MCMAS: A Model Checker for Multi-agent Systems. In *Proceedings of TACAS'2006 (LNCS)*, Vol. 3920. 450–454.

[33] O. Matz, N. Schweikardt, and W. Thomas. 2002. The Monadic Quantifier Alternation Hierarchy over Grids and Graphs. *Inf. Comput.* 179, 2 (2002), 356–383. https://doi.org/10.1006/inco.2002.2955

[34] B. Maubert. 2014. *Logical foundations of games with imperfect information: uniform strategies*. Ph.D. Dissertation. Université de Rennes 1. http://www.theses.fr/en/2014REN1S001

[35] B. Maubert and S. Pinchinat. 2013. Jumping Automata for Uniform Strategies. In *Proceedings of FSTTCS'13*. 287–298. https://doi.org/10.4230/LIPIcs.FSTTCS.2013.287

[36] R. Milner. 1983. Calculi for synchrony and asynchrony. *Theor. Comp. Sci.* 25, 3 (1983), 267–310.

[37] D. E. Muller and P. E. Schupp. 1995. Simulating alternating tree automata by nondeterministic automata: New results and new proofs of the theorems of Rabin, McNaughton and Safra. *Theoretical Computer Science* 141, 1–2 (17 April 1995), 69–107.

[38] D. Park. 1981. Concurrency and Automata on Infinite Sequences. In *Proc. 5th GI Conf. on Th. Comp. Sci., LNCS 104*. Springer-Verlag, 167–183.

[39] B. Puchala. 2010. Asynchronous Omega-Regular Games with Partial Information. In *Proceedings of MFCS'2010*. 592–603.

[40] M. O. Rabin. 1969. Decidability of second-order theories and automata on infinite trees. *Trans. Amer. Math. Soc.* 141 (1969), 1–35.

[41] Jean-François Raskin, Krishnendu Chatterjee, Laurent Doyen, and Thomas A. Henzinger. 2007. Algorithms for Omega-Regular Games with Imperfect Information. *LMCS* 3, 3 (2007), 287–302. https://doi.org/10.2168/LMCS-3(3:4)2007

[42] John H. Reif. 1984. The complexity of two-player games of incomplete information. *Journal of computer and system sciences* 29, 2 (1984), 274–301. https://doi.org/10.1016/0022-0000(84)90034-5

[43] F. Reiter. 2014. Distributed Graph Automata. *CoRR* abs/1404.6503 (2014). http://arxiv.org/abs/1404.6503

[44] Nikolay V. Shilov. 1997. Program Schemata vs. Automata for Decidability of Program Logics. *Theor. Comput. Sci.* 175, 1 (1997), 15–27. https://doi.org/10.1016/S0304-3975(96)00168-5

[45] N. V. Shilov and N. O. Garanina. 2002. *Combining Knowledge and Fixpoints*. Technical Report Preprint n.98, `http://www.iis.nsk.su/files/preprints/098.pdf`. A.P. Ershov Institute of Informatics Systems, Novosibirsk.

[46] Nikolay V Shilov, Natalya Olegovna Garanina, and K-M Choe. 2006. Update and abstraction in model checking of knowledge and branching time. *Fundamenta Informaticae* 72, 1-3 (2006), 347–361.

[47] Nikolay V. Shilov, S. O. Shilova, and A. Yu. Bernshtein. 2016. Program schemata technique for propositional program logics: A 30-year history. *Programming and Computer Software* 42, 4 (2016), 239–256. https://doi.org/10.1134/S036176881604006X

[48] W. Thomas. 1992. Infinite Trees and Automaton-Definable Relations over omega-Words. *Theor. Comput. Sci.* 103, 1 (1992), 143–159.

[49] W. Thomas. 1997. Languages, Automata, and Logic. In *Handbook of Formal Languages*, G. Rozenberg and A. Salomaa (Eds.). Vol. 3, Beyond Words. Springer Verlag, 389–455.

[50] Johan Van Benthem. 1984. Correspondence theory. In *Handbook of philosophical logic*. Springer, 167–247.

[51] R. van der Meyden and N. Shilov. 1999. Model Checking Knowledge and Time in Systems with Perfect Recall (Extended Abstract). In *Proceedings of FSTTCS'99 (LNCS)*, Vol. 1738. 432–445.

[52] M. Y. Vardi. 1998. Reasoning about The Past with Two-Way Automata. In *Proceedings of ICALP'98*, Vol. 1443. 628–641.

[53] I. Walukiewicz. 2002. Monadic second-order logic on tree-like structures. *Theor. Comput. Sci.* 275, 1-2 (2002), 311–346. https://doi.org/10.1016/S0304-3975(01)00185-2

[54] Wieslaw Zielonka. 1998. Infinite Games on Finitely Coloured Graphs with Applications to Automata on Infinite Trees. *Theor. Comput. Sci.* 200, 1-2 (1998), 135–183.

A PROOF OF LEMMA 5.5

Lemma 5.5. *Let $\pi^* \in Out(\mathcal{G}^*, \sigma^*)$. For every finite prefix ρ^* of π^* such that last$(\rho^*) = (X_i, \rho a v)$ for some $0 \leq i \leq k$, some partial play ρ, action a and position v, it holds that $C^*(\rho^*) = C(\rho)$ and $\rho a v$ follows σ.*

Proof. Assume that Lemma 5.5 does not hold. There exists a longest finite prefix ρ^* of π^* such that last$(\rho^*) = (X_i, \rho a v)$ for some $0 \leq i \leq k$, some partial play ρ, action a and position v, and such that $C^*(\rho^*) = C(\rho)$ and $\rho a v$ follows σ. We build a strictly longer prefix of π^* that also verifies the above property, thus obtaining a contradiction.

From $(X_i, \rho a v)$ there is no choice for either player for the next $k - i + 1$ moves, so that

$$\rho^*{}_1 := \rho^* \cdot (\varphi_i, \rho a v) \ldots (\varphi_k, \rho a v)(\bigvee_{0 \leq i \leq k} (i \wedge \bigvee_{a \in A} \boxminus\boxed{a} X_i), \rho a v)$$

is also necessarily a prefix of π^*. Then, by definition of σ^*, Verifier chooses $i = C(v)$, so that

$$\rho^*{}_2 := \rho^*{}_1 \cdot (\bigvee_{a \in A} \boxminus\boxed{a} X_i, \rho a v), \text{ where } i = C(v),$$

is also a prefix of π^*. Again by definition of σ^*,

$$\rho^*{}_3 := \rho^*{}_2 \cdot (\boxminus\boxed{b} X_i, \rho a v), \text{ where } b = \sigma(\rho a v),$$

is a prefix of π^*. Then, there exists ρ' such that $\rho' \sim \rho a v$ and

$$\rho^*{}_4 := \rho^*{}_3 \cdot (\boxed{b} X_i, \rho')$$

is a prefix of π^*, and there also exists $v' \in b^{\mathcal{G}^{\mathrm{i}}}(\text{last}(\rho'))$ such that

$$\rho^*{}_5 := \rho^*{}_4 \cdot (X_i, \rho' b v')$$

is a prefix of π^*.

We show that $\rho^*{}_5$ is a good candidate, and our proof by contradiction is done. First, it is clear that $\rho^*{}_5$ is strictly longer than ρ^*. Then, by definition of C^* (recall that C^* only considers colours of positions of the form (X_i, ρ)), we have that $C^*(\rho^*{}_5) = C^*(\rho^*) \cdot i$. By assumption, $C^*(\rho^*) = C(\rho)$, and since $i = C(v)$, we have $C^*(\rho^*{}_5) = C(\rho a v)$. Then, because $\rho a v \sim \rho'$, and because colours are observable, we obtain that $C^*(\rho^*{}_5) = C(\rho')$. It remains to prove that $\rho' b v'$ follows σ.

From the fact that $\sim$ is synchronous perfect recall and that actions are visible, one can easily show that if a partial play follows a uniform strategy, then every equivalent play also does, as all their strict prefixes of same length are equivalent and followed by the same action. Therefore, since $\rho' \sim \rho a v$ and $\rho a v$ follows σ, also does ρ'. Next, we have that $b = \sigma(\rho a v)$, and again, because $\rho' \sim \rho a v$ and σ is uniform, $\sigma(\rho') = \sigma(\rho a v) = b$, so that $\rho' b v'$ also follows σ, which concludes. □

B PROOF OF LEMMA 5.6

Lemma 5.6. *Let ρ and ρ' be two partial plays in $\mathcal{G}^{\mathrm{i}}$ such that $\rho \sim \rho'$. For every two partial plays ρ^* and $\rho^{*\prime}$ that end respectively in (X_i, ρ) and (X_j, ρ'), it holds that $C^*(\rho^*) = C^*(\rho^{*\prime})$.*

Proof. By induction on the common length n of ρ and ρ'.

Base case: There is only one partial play of length 1: $\rho = \rho' = v_\iota$. Also, in $\mathcal{G}^*$, the initial position is $(\text{WinParity}_k^A, v_\iota)$, and every play π^* starts as follows:

$$\begin{aligned}\pi^* = \; & (\text{WinParity}_k^A, v_\iota)(\varphi_1, v_\iota) \ldots (\varphi_k, v_\iota)(\textstyle\bigvee_{0 \leq i \leq k}(i \wedge \bigvee_{a \in A} \boxminus\boxed{a} X_i), v_\iota)(\bigvee_{a \in A} \boxminus\boxed{a} X_i, v_\iota) \cdot \\ & (\boxminus\boxed{a} X_i, v_\iota)(\boxed{a} X_i, v_\iota)(X_i, v_\iota a v) \cdot \pi^{*\prime}\end{aligned}$$

where $0 \leq i \leq k$, $a \in A$, $v \in a^{\mathcal{G}^{\mathrm{i}}}(v_\iota)$ and $\pi^{*\prime}$ is a play in $(\mathcal{G}^*, (\varphi_i, v_\iota a v))$. In particular, recall that the only partial play equivalent to v_ι is itself.

Now one can observe that there are no ρ^* and $\rho^{*\prime}$ that end in (X_i, v_ι) and (X_j, v_ι) such that Lemma 5.6 holds for $|\rho| = 1$.

Inductive case: Let $\rho = \rho_1 a v$ and $\rho' = \rho_1' a' v'$. Because $\rho \sim \rho'$, we have that $\rho_1 \sim \rho_1'$, $a = a'$ and $o_v = o_{v'}$. Suppose that there are ρ^* and $\rho^{*\prime}$ that end respectively in (X_i, ρ) and (X_j, ρ'). They are of the form

$$\begin{aligned} \rho^* = \quad & {\rho^*}_1 \cdot (\bigvee_{a\in A} \boxminus\boxed{a}X_i, \rho_2)(\boxminus\boxed{a}X_i, \rho_2)(\boxed{a}X_i, \rho_1)(X_i, \rho_1 \cdot av), \text{and} \\ \rho^{*\prime} = \quad & {\rho^*}'_1 \cdot (\bigvee_{a\in A} \boxminus\boxed{a}X_j, \rho_2')(\boxminus\boxed{a}X_j, \rho_2')(\boxed{a}X_j, \rho_1')(X_j, \rho_1' \cdot av'), \end{aligned}$$

where $\rho_2 \sim \rho_1$ and $\rho_2' \sim \rho_1'$. Also, recall that we have assumed Verifier to always pick the right parity (otherwise she loses immediately), so that $i = C(\text{last}(\rho_2))$ and $j = C(\text{last}(\rho_2'))$. Because colours are observable and $\rho_2 \sim \rho_1 \sim \rho_1' \sim \rho_2'$, we have that $i = j$. Now, either $C^*({\rho^*}_1) = \epsilon = C^*({\rho^*}'_1)$, and therefore $C^*(\rho^*) = i = C^*(\rho^{*\prime})$, which concludes, or ${\rho^*}_1$ and ${\rho^*}'_1$ have prefixes that end in (X_k, ρ_2) and (X_l, ρ_2'), respectively. By induction hypothesis, as $\rho_2 \sim \rho_2'$, we have that $C^*({\rho^*}_1) = C^*({\rho^*}'_1)$ and therefore $C^*(\rho^*) = C^*(\rho^{*\prime})$. □

C PROOF OF LEMMA 5.10

Lemma 5.10.

(1) *For all $q \in Q$ and $k \neq 2^N + 1$, $(\mathcal{G}_0, v_k^q) \Leftrightarrow (\mathcal{G}_i, v_k^q)$, and*
(2) *for all $q \in Q$ and $k \neq 2^N + 2$, $(\mathcal{G}_0, v_k^q) \Leftrightarrow (\mathcal{G}_j, v_k^q)$.*

Proof. For convenience, for $v, v' \in V$ and $k \in \{0, i, j\}$, we shall write $v \to_k v'$ if $(v, v') \in E_k$.

We start with Point 1 of Lemma 5.10. The rough idea is that the trees $t_0 = (\tau, m_0)$ and $t_i = (\tau, m_i)$ are the same, except for the labelings of the subtree $[\tau]_{y_{2^N+1}}$. So as long as the games remain out of this subtree, a move from a position (x, q, α) in $\mathcal{G}_0$ can be simulated by the same move in $\mathcal{G}_i$, and vice versa. Problems arise when, in one of the acceptance games, a move jumps to a node in $[\tau]_{y_{2^N+1}}$. This is dealt with by noticing that $[t_0]_{y_{2^N+1}} = [t_i]_{y_j}$, and $[t_i]_{y_{2^N+1}} = [t_0]_{y_i}$. Therefore, if a move in $\mathcal{G}_0$ jumps to a node in $[\tau]_{y_{2^N+1}}$, this is simulated in $\mathcal{G}_i$ by jumping to the corresponding node in $[\tau]_{y_j}$. More precisely, if $\mathcal{G}_0$ jumps to some node $y_{2^N+1} \cdot w$, where $w \in \{0,1\}^*$, this is simulated in $\mathcal{G}_i$ by jumping to node $y_j \cdot w$. This is possible, because in these games with only one observation, all plays (or nodes) that contain the same sequence of actions are related. For a node x, let us define $a(x)$ as the sequence of actions taken by Eve from the root to x, and observe that for all $1 \leq i \leq 2^N + 2$, for all $w \in \{0,1\}^N \cdot \{0\}^*$, $a(y_i \cdot w) = a_0 a_0 a_{w[1]} a_{w[2]} \ldots a_{\text{last}(w)}$. Therefore, if a node is related to $y_{2^N+1} \cdot w$, it is also related to $y_j \cdot w$. Similarly, if a move in $\mathcal{G}_i$ jumps to a node in $[\tau]_{y_{2^N+1}}$, this is simulated in $\mathcal{G}_0$ by jumping to the corresponding node in $[\tau]_{y_i}$.

Let us define the binary relation $Z \subseteq V_0 \times V_i$ as the smallest relation such that, for all $q \in Q$ and all $\alpha \in \mathbb{B}^+(Dir \times Q)$:

- $\forall k \neq 2^N + 1, \forall x \in [\tau]_{y_k}, (x, q, \alpha) Z (x, q, \alpha)$,
- $\forall w \in \{0,1\}^*, (y_{2^N+1} \cdot w, q, \alpha) Z (y_j \cdot w, q, \alpha)$, and
- $\forall w \in \{0,1\}^*, (y_i \cdot w, q, \alpha) Z (y_{2^N+1} \cdot w, q, \alpha)$.

First, for all $q \in Q$ and $k \neq 2^N+1$, we have by definition of Z that $(y_k, q, \delta(q, \ell_0^{y_k})) Z (y_k, q, \delta(q, \ell_i^{y_k}))$, *i.e.*, $v_k^q Z v_k^q$ (recall that $\ell_0^{y_k} = \ell_i^{y_k}$). To prove that $(\mathcal{G}_0, v_k^q) \Leftrightarrow (\mathcal{G}_i, v_k^q)$, it remains to prove that Z is a bisimulation between $\mathcal{G}_0$ and $\mathcal{G}_i$, which we do now.

Take $(v, v') \in Z$. By definition of Z, there are x, x', q and α such that $v = (x, q, \alpha)$ and $v' = (x', q, \alpha)$. Also, x and x' are on the same level in the tree τ, either on the level of the nodes $\{y_k \mid 1 \leq k \leq 2^N + 2\}$ or below.

Harmony on atomic propositions (Point 1 of Definition 2.5): by definition of the colours in acceptance games, it holds that $C^0(v) = C(q) = C^i(v')$. Also, because v and v' contain the same formula α, we have that $v \in V_1$ iff $v' \in V_1$, and *idem* for V_2. Therefore v and v' agree on all atomic propositions in $\mathcal{AP} = \{V_1, V_2\} \cup \mathbb{N}$.

Zig (Point 2 of Definition 2.5): Take $u \in V$ such that $v \to_0 u$. We need to find some $u' \in V$ such that $v' \to_i u'$ and uZu'. According to the possible moves in the semantic games (see Section 4), the move $v \to_0 u$ is of one of the three following kinds:

(1) split α without moving in the tree nor changing state,
(2) go down to a child of x in a state q', or
(3) jump to a node y such that $x \rightsquigarrow y$ in a state q'.

Case 1: We have $u = (x, q, \beta)$, where β is some subformula of α. By definition of acceptance games, this splitting is also possible in $\mathcal{G}_i$: $(x', q, \alpha) \to_i (x', q, \beta)$. Writing $u' := (x', q, \beta)$, this becomes $v' \to_i u'$. Because we have $(x, q, \alpha)Z(x', q, \alpha)$, by definition of Z we have $(x, q, \alpha)Z(x', q, \alpha)$ for all α', and in particular $(x, q, \beta)Z(x', q, \beta)$, *i.e.*, uZu'.

Case 2: In this case, we have $\alpha = \Diamond q'$ or $\alpha = \Box q'$, and $u = (y, q', \delta(q', \ell_0^y))$ for some child y of x; write $\beta := \delta(q', \ell_0^y)$ and $y := x \cdot c$ (where $c \in \{0, 1\}$).

First, observe that by definition of Z, x and x' are on the same level in the tree ($|x| = |x'|$), and therefore if $x \cdot c$ exists in τ, so does $x' \cdot c$. It follows, by definition of acceptance games, that $v' \to_i (x' \cdot c, q', \delta(q', \ell_i^{x' \cdot c}))$ exists in $\mathcal{G}_i$; write $y' := x' \cdot c$, $\beta' := \delta(q', \ell_i^{y'})$ and $u' := (x' \cdot c, q', \beta')$. We prove that uZu'.

We distinguish three possibilities again, according to the definition of Z and the fact that $(x, q, \alpha)Z(x', q, \alpha)$.

- $x = x'$ (and therefore, also $y = y'$). By definition of Z, we have that $x \notin [\tau]_{y_{2^N+1}}$, and thus $y \notin [\tau]_{y_{2^N+1}}$. This implies that $\ell_0^y = \ell_i^y$. Therefore $\beta = \beta'$, and $u = u'$, which, by definition of Z, entails that uZu'.
- $x = y_{2^N+1} \cdot w$ for some w. Because vZv', by definition of Z we have that $x' = y_j \cdot w$ (and thus $y' = y_j \cdot w \cdot c$). It holds that $\ell_0^{y_{2^N+1} \cdot w \cdot c} = \ell_i^{y_j \cdot w \cdot c}$, *i.e.*, $\ell_0^y = \ell_i^{y'}$, hence $\beta = \beta'$. We thus have $u = (y_{2^N+1} \cdot w \cdot c, q', \beta)$ and $u' = (y_j \cdot w \cdot c, q', \beta)$, and by definition of Z we obtain that uZu'.
- $x = y_i \cdot w$ for some w. Because vZv', by definition of Z we have that $x' = y_{2^N+1} \cdot w$ (and thus $y' = y_{2^N+1} \cdot w \cdot c$). Again, it holds that $\ell_0^{y_i \cdot w \cdot c} = \ell_i^{y_{2^N+1} \cdot w \cdot c}$, *i.e.*, $\ell_0^y = \ell_i^{y'}$, and therefore $\beta = \beta'$. We thus have $u = (y_i \cdot w \cdot c, q', \beta)$ and $u' = (y_{2^N+1} \cdot w \cdot c, q', \beta)$, and by definition of Z we obtain that uZu'.

Case 3: We have that $\alpha = \diamondsuit q'$ or $\alpha = \boxminus q'$ for some q', $u = (y, q', \beta)$ for some y such that $x \rightsquigarrow y$ and $\beta = \delta(q', \ell_0^y)$. By definition of Z we have that x and x' contain the same sequence of actions ($a(x) = a(x')$), and because Eve does not observe anything, the nodes related by $\rightsquigarrow$ to x are the same as those related to x' (they are all the nodes y that contain the same action sequence as x and x'). We therefore have $a(y) = a(x) = a(x')$. We distinguish two cases: the case where the jump $v \to_0 u$ is not towards a node in $[\tau]_{y_{2^N+1}}$, and the case where it is. In the latter case, it is simulated in $\mathcal{G}_i$ by a jump to the corresponding node in $[\tau]_{y_j}$.

- $y \in [\tau]_{y_k}$ for some $k \neq 2^N + 1$: since $a(x') = a(y)$, we have that $x' \rightsquigarrow y$, and thus the move $v' \to_i (y, q', \delta(q', \ell_i^y)) =: u'$ exists in $\mathcal{G}_i$. Now, because $\ell_0^y = \ell_i^y$, we have that $u = u'$, and finally uZu'.
- $y \in [\tau]_{y_{2^N+1}}$: take $w \in \{0, 1\}^*$ such that $y = y_{2^N+1} \cdot w$. The move $v \to_0 u = (y_{2^N+1} \cdot w, q', \delta(q', \ell_0^{y_{2^N+1} \cdot w}))$ is simulated in $\mathcal{G}_i$ by the move $v' \to_i (y_j \cdot w, q', \delta(q', \ell_i^{y_j \cdot w}))$. To show that this move exists in $\mathcal{G}_i$, it is enough to show that $x' \rightsquigarrow y_j \cdot w$, *i.e.*, $a(x') = a(y_j \cdot w)$. By definition of Z, we already know that $a(x) = a(x')$, and because $x \rightsquigarrow y$ and $y = y_{2^N+1} \cdot w$, we know that $a(x) = a(y_{2^N+1} \cdot w)$. So $a(x') = a(y_{2^N+1} \cdot w)$, and clearly $a(y_{2^N+1} \cdot w) = a(y_j \cdot w)$, so we indeed have that $x' \rightsquigarrow y_j \cdot w$.

Let $y' := y_j \cdot w$ and $u' := (y', q', \delta(q', \ell_i^{y'}))$. Because $\ell_0^{y_{2^N+1} \cdot w} = \ell_i^{y_j \cdot w}$, we have $\delta(q', \ell_i^{y'}) = \delta(q', \ell_0^{y}) = \beta$. We thus have $u = (y_{2^N+1} \cdot w, q', \beta)$, $u' = (y_j \cdot w, q', \beta)$ and therefore, by definition of Z, we get that uZu'.

Zag (Point 3 of Definition 2.5): the proof is almost symmetrical to the one for Zig. The only difference is in the second subcase of Case 3: a move in $\mathcal{G}_i$ that jumps to a node of the form $y_{2^N+1} \cdot w$, where $w \in \{0,1\}^*$, is simulated in $\mathcal{G}_0$ by a jump to the node $y_i \cdot w$. We make use of the third point in the definition of Z to show that the two resulting positions are in the relation.

So Z is a bisimulation between $\mathcal{G}_0$ and $\mathcal{G}_i$, which concludes the proof of Point 1 in Lemma 5.10.

We turn to the proof of Point 2 in Lemma 5.10. The only difference with Point 1 is that, while t_0 and t_i differ on the labeling of $[\tau]_{y_{2^N+1}}$, t_0 and t_j differ on the labeling of $[\tau]_{y_{2^N+2}}$. So here, the only moves that cannot be simulated by the same one in the other game are those that jump in $[\tau]_{y_{2^N+2}}$.

We define the following binary relation $Z' \subseteq V^0 \times V^j$, very similar to Z, as the smallest relation such that, for all $q \in Q$ and all $\alpha \in \mathbb{B}^+(Dir \times Q)$:

- $\forall k \neq 2^N + 2$, $\forall x \in [\tau]_{y_k}$, $(x, q, \alpha)Z'(x, q, \alpha)$,
- $\forall w \in \{0,1\}^*$, $(y_{2^N+2} \cdot w, q, \alpha)Z'(y_i \cdot w, q, \alpha)$, and
- $\forall w \in \{0,1\}^*$, $(y_j \cdot w, q, \alpha)Z'(y_{2^N+2} \cdot w, q, \alpha)$.

The rest of the proof is just the same as for Point 1. □

Received April 2017; revised November 2017; accepted June 2018