

Model Checking Linear Coalgebraic Temporal Logics: an Automata-Theoretic Approach

Corina Cîrstea

University of Southampton
cc2@ecs.soton.ac.uk

Abstract. We extend the theory of maximal traces of pointed non-deterministic coalgebras by providing an automata-based characterisation of the set of maximal traces for finite such coalgebras. We then consider *linear* coalgebraic temporal logics interpreted over non-deterministic coalgebras, and show how to reduce the model checking problem for such logics to the problem of deciding the winner in a regular two-player game. Our approach is inspired by the automata-theoretic approach to model checking Linear Temporal Logic over transition systems.

Keywords: coalgebra, trace, temporal logic, automata, model checking

1 Introduction

A coalgebraic version of the well-established notion of trace of a state in a transition system was defined in [4] for coalgebras of type $\mathcal{P} \circ F$ or $\mathcal{P}^+ \circ F$, with $\mathcal{P} : \mathbf{Set} \rightarrow \mathbf{Set}$ ($\mathcal{P}^+ : \mathbf{Set} \rightarrow \mathbf{Set}$) the (non-empty) powerset functor and $F : \mathbf{Set} \rightarrow \mathbf{Set}$ a polynomial functor. This was extended in [2] to coalgebras of type $T \circ F$, with $F : \mathbf{C} \rightarrow \mathbf{C}$ an endofunctor and $T : \mathbf{C} \rightarrow \mathbf{C}$ a monad that distributes suitably over F . Here we return to the setting of [4] and provide an automata-based characterisation of the traces of a pointed $\mathcal{P}^+ \circ F$ -coalgebra.

Coalgebra automata were introduced in [8] as devices that accept pointed coalgebras. For a weak pullback preserving functor F , any pointed F -coalgebra $\mathbb{X} = (X, \gamma, x_0)$ with finite carrier defines a deterministic F -automaton accepting precisely the pointed F -coalgebras that are bisimilar to \mathbb{X} . The key idea underlying this paper is to view a finite, pointed $\mathcal{P}^+ \circ F$ -coalgebra \mathbb{X} not as a deterministic $\mathcal{P}^+ \circ F$ -automaton, but as a *non-deterministic* F -automaton. Our first result provides a characterisation of the set of traces of \mathbb{X} as those elements of the final F -coalgebra that are accepted by this automaton.

The second part of the paper exploits this insight in order to obtain a decision procedure for model-checking the coalgebraic counterparts of linear temporal logic. Such logics arise as certain fragments of the path-based coalgebraic temporal logics defined in [2], and are interpreted over coalgebras of type $\mathcal{P}^+ \circ F$. Specifically, we consider logics whose formulas are of the form $\Box\varphi$, with φ a formula of a coalgebraic fixed point logic determined by a set A of predicate liftings for the functor F . The formula $\Box\varphi$ requires *all* traces of a state in a $\mathcal{P}^+ \circ F$ -coalgebra to satisfy the property expressed by φ . We define a regular two-player

graph game for deciding whether a formula φ as above holds in *some* trace of a finite pointed $\mathcal{P}^+ \circ F$ -coalgebra \mathbb{X} , by combining the Λ -automaton induced by φ (as defined in [3]) with the non-deterministic F -automaton induced by \mathbb{X} .

Our approach is inspired by the automata-theoretic approach to model-checking Linear Temporal Logic [7]. This exploits the observation that both the models and the formulas to be checked can be represented as *Büchi automata*. Then, checking a formula against a model amounts to verifying that the product of the model automaton (whose runs correspond to possible traces of the model) with the automaton induced by the negation of the formula to be verified (which accepts precisely those traces that violate the original formula) does not accept any trace. In our setting, the automata-based characterisation of the traces of pointed $\mathcal{P}^+ \circ F$ -coalgebras is not directly useful, as our model automaton is an F -automaton whereas our formula automaton is a Λ -automaton. However, our approach is similar in nature: we define a regular game $\mathcal{G}_{\neq\emptyset}(\mathbb{X}, \mathbb{A})$ for every finite, pointed $\mathcal{P}^+ \circ F$ -coalgebra \mathbb{X} and every Λ -automaton \mathbb{A} , with the crucial property that $\mathcal{G}_{\neq\emptyset}(\mathbb{X}, \mathbb{A})$ admits a winning strategy for \exists precisely when \mathbb{X} contains a trace that is accepted by \mathbb{A} . This constitutes our second result.

2 Preliminaries

Graph Games A *graph game* played by two players \exists and \forall is a tuple $\mathcal{G} = (B_{\exists}, B_{\forall}, E, Win)$ where $B = B_{\exists} \cup B_{\forall}$ is the disjoint union of *positions* owned by \exists and respectively \forall , $E \subseteq B \times B$ indicates the allowed moves, and $Win \subseteq B^{\omega}$. A *play* in \mathcal{G} is a finite or infinite sequence of positions (b_0, b_1, \dots) such that $(b_i, b_{i+1}) \in E$ for all i . A finite play is lost by the player who cannot move, whereas an infinite play (b_0, b_1, \dots) is won by \exists if and only if $(b_0, b_1, \dots) \in Win$.

A *strategy* in \mathcal{G} for a player $P \in \{\exists, \forall\}$ is a function that maps plays that end in a position $b \in B_P$ to a position $b' \in B$ such that $(b, b') \in E$. Intuitively, a strategy determines a player's next move, depending on the history of the play. A strategy for player P is said to be (i) *history-free* when next moves only depend on the current position, and (ii) *winning from position* $b \in B$ if P wins all plays (b_0, b_1, \dots) with $b_0 = b$ when playing according to that strategy.

A graph game is called *regular* if there exists an ω -regular language L over a finite alphabet C and a map $col : B \rightarrow C$ such that $Win = \{(b_0, b_1, \dots) \in B^{\omega} \mid col(b_0)col(b_1) \dots \in L\}$. A *parity game* is a graph game whose winning condition is defined using a *parity map* $\Omega : B \rightarrow \omega$ with finite range, by $Win = \{(b_0, b_1, \dots) \mid \max\{k \mid k = \Omega(b_i) \text{ for infinitely many } i \in \omega\} \text{ is even}\}$. Any regular game can be transformed into an equivalent parity game [6]. Regular games are determined, that is, either \exists or \forall has a winning strategy from any given position. In addition, parity games admit history-free winning strategies.

Maximal Traces and Executions A definition of infinite traces for coalgebras of type $\mathcal{P} \circ F$ or $\mathcal{P}^+ \circ F$ with F a *polynomial endofunctor* (i.e. constructed from identity and constant functors using products, coproducts and exponents) was given in [4]. This was generalised in [2], where it was shown how to define, for a

coalgebra (X, γ) of type $T \circ F$ with $F : \mathbf{C} \rightarrow \mathbf{C}$ an endofunctor and $T : \mathbf{C} \rightarrow \mathbf{C}$ an affine monad that distributes over F via a distributive law $\lambda : FT \Longrightarrow TF$, a *maximal trace map* $\text{tr}_\gamma : X \rightarrow T(Z)$ as well as a *maximal execution map* $\text{exec}_\gamma : X \rightarrow T(Z_X)$ ¹. Here, Z is the carrier of the final F -coalgebra (Z, ζ) , and Z_X is the carrier of the final $X \times F$ -coalgebra (Z_X, ζ_X) . The states of Z and Z_X represent potential maximal traces, respectively executions. In addition to the information provided by a trace, an execution also records the states visited, including the initial state. The maximal trace (execution) map then assigns to each state in X a suitably-structured collection of traces (respectively executions). For $T = \mathcal{P}^+$, this collection is structured as a set.

- Example 1.*
1. Let $T = \mathcal{P}^+$ and $F = A \times \text{Id}$. That is, $T \circ F$ -coalgebras are labelled transition systems with the additional requirement that each state has at least one successor². In this case, the potential maximal traces (elements of the final $A \times \text{Id}$ -coalgebra) are given by infinite sequences of elements of A . Also, for a $\mathcal{P}^+ \circ F$ -coalgebra (X, γ) , the potential maximal executions are given by infinite sequences of the form $x_0 a_0 x_1 a_1 \dots$ with $x_i \in X$ and $a_i \in A$ for $i \in \omega$. The maximal execution map $\text{tr}_\gamma : X \rightarrow \mathcal{P}^+(Z)$ assigns to each state $x_0 \in X$, the set of *computation paths* (infinite sequences $x_0 a_0 x_1 a_1 \dots$ with $(a_i, x_{i+1}) \in \gamma(x_i)$ for $i \in \omega$) originating in x_0 . Also, the maximal trace map assigns to each state x_0 , the set of infinite sequences of elements of A that appear in computation paths starting in x_0 .
 2. By changing F to $1 + A \times \text{Id}$ with $1 = \{*\}$, one moves to labelled transition systems with explicit termination (with the same restriction as above). This time, the potential maximal traces and executions also include *finite* ones of the form $a_0 a_1 \dots a_n *$, and respectively $x_0 a_0 x_1 \dots a_n x_{n+1} *$.
 3. Taking $F = A + \text{Id} \times \text{Id}$ results in maximal executions for a $\mathcal{P}^+ \circ F$ -coalgebra (X, γ) being given by (possibly infinite) binary trees whose nodes (including any leaves) are labelled by states of X , and whose leaves (if any) are labelled by elements of A . As before, maximal traces can be obtained from maximal executions by removing the labelings with elements of X from the nodes.

For $T = \mathcal{P}^+$, the trace map of [2] is given by:

$$\text{tr}_\gamma(x) = \{z \in Z \mid \pi_i(z) \in \gamma_i(x) \text{ for all } i \in \omega\} \quad (1)$$

where $(Z, (\pi_i : Z \rightarrow F^i 1)_{i \in \omega})$ defines the limit of the final sequence of F , and where the maps $\gamma_i : X \rightarrow \mathcal{P}^+(F^i 1)$ are defined by induction on i :

- $\gamma_0 = \eta_1 \circ !_X : X \rightarrow \mathcal{P}^+ 1$, where $\eta : \text{Id} \Longrightarrow \mathcal{P}^+$ denotes the unit of the monad \mathcal{P}^+ , and $!_X : X \rightarrow 1$ is the unique map from X to the one-element set 1 ,
- $\gamma_{i+1} = \mu_{F^{i+1} 1} \circ \mathcal{P}^+ \lambda_{F^i 1} \circ \mathcal{P}^+ F \gamma_i \circ \gamma : X \rightarrow \mathcal{P}^+(F^{i+1} 1)$ for $i \in \omega$, where $\mu : \mathcal{P}^+ \circ \mathcal{P}^+ \Longrightarrow \mathcal{P}^+$ denotes the multiplication of the monad \mathcal{P}^+ .

¹ Some additional assumptions on F and T were needed in loc. cit., including preservation of certain ω^{op} -limits by F and $X \times F$.

² This restriction is typical in logics such as CTL*, where it allows the definition of a notion of computation path that only accounts for infinite computations.

Thus, the elements of $\text{tr}_\gamma(x)$ are given by infinite sequences $(z_i)_{i \in \omega}$ with $z_i \in \gamma_i(x)$ for $i \in \omega$. Here, the elements of $F^i 1$ are to be thought of as finite approximations of maximal traces, and the maps γ_i can be regarded as finite approximations of the trace map.

An equivalent definition of the trace map when $T = \mathcal{P}^+$ and when the distributive law $\lambda : F\mathcal{P}^+ \Rightarrow \mathcal{P}^+F$ arises via *relation lifting* is given in [4]. For a relation $\langle r_1, r_2 \rangle : R \rightarrow X \times Y$, the relation $\text{Rel}(F)(R) \subseteq FX \times FY$ is obtained using the epi-mono factorisation of $\langle F(r_1), F(r_2) \rangle$ as follows:

$$\begin{array}{ccc} F(R) & \twoheadrightarrow & \text{Rel}(F)(R) \\ & \searrow \langle F(r_1), F(r_2) \rangle & \downarrow \\ & & F(X) \times F(Y) \end{array}$$

Then, when fixing $\lambda : F\mathcal{P}^+ \Rightarrow \mathcal{P}^+F$ to be given by:

$$a \in \lambda_X(u) \quad \text{iff} \quad (a, u) \in \text{Rel}(F)(\in_X) \quad (2)$$

with $\in_X \subseteq X \times \mathcal{P}^+X$ the membership relation, the trace map $\text{tr}_\gamma : X \rightarrow \mathcal{P}^+Z$ associated to a $\mathcal{P}^+ \circ F$ -coalgebra (X, γ) can be defined by exploiting the observation that the following set carries F -coalgebra structure:

$$U := \{(u_i)_{i \in \omega} \in \prod_{i \in \omega} F^i X \mid (u_{i+1}, u_i) \in B_i \text{ for } i \in \omega\}$$

where the relations $B_i \subseteq F^{i+1}X \times F^i X$ are defined inductively by:

- $B_0 := \{(y, x) \in FX \times X \mid y \in \gamma(x)\}$,
- $B_{i+1} := \text{Rel}(F)(B_i)$ for $i \in \omega$.

Specifically, [4] shows the existence of an isomorphism $\varphi : U_- \rightarrow FU$, with $U_- := \{(u_{i+1})_{i \in \omega} \mid (u_i)_{i \in \omega} \in U\}$. This, in turn, yields an F -coalgebra structure $\sigma : U \rightarrow FU$ on U :

$$U \xrightarrow{\langle \pi_{i+1} \rangle_{i \in \omega}} U_- \xrightarrow[\simeq]{\varphi} FU$$

The trace map $\text{tr}_\gamma : X \rightarrow \mathcal{P}^+Z$ is now given by:

$$\text{tr}_\gamma(x) = \{z \in Z \mid z = !_U(u) \text{ for some } u \in U \text{ with } \pi_0(u) = x\} \quad (3)$$

where $!_U : (U, \sigma) \rightarrow (Z, \zeta)$ is the unique F -coalgebra homomorphism arising from the finality of (Z, ζ) .

Path-Based Coalgebraic Temporal Logics The path-based coalgebraic temporal logics defined in [2] are interpreted over coalgebras of type $T \circ F$, with T and F as before, and are parameterised by sets Λ and Λ_F of monotone predicate liftings for the endofunctors T and F , respectively. In [2], a *(unary)³ predicate*

³ To simplify the presentation, we assume all predicate liftings to be either unary or nullary; however, our results hold for predicate liftings with arbitrary finite arities.

lifting for an endofunctor $F : \mathbf{C} \rightarrow \mathbf{C}$ is taken to be a natural transformation $\lambda : P \Longrightarrow P \circ F$, with $P : \mathbf{C} \rightarrow \mathbf{Set}$ a subfunctor of $\hat{\mathcal{P}} \circ U$, $\hat{\mathcal{P}} : \mathbf{Set} \rightarrow \mathbf{Set}^{\text{op}}$ the contravariant powerset functor, and $U : \mathbf{C} \rightarrow \mathbf{Set}$ mapping a state space to its underlying set. The syntax and semantics of path-based coalgebraic temporal logics are summarised next.

Definition 2 ([2]). *Let Λ and Λ_F denote sets of monotone predicate liftings for T and F , respectively. The language $\mu\mathcal{L} ::= \mu\mathcal{L}_{\Lambda}^{\Lambda_F}(\mathcal{U}, \mathcal{V})$ over a 2-sorted set $(\mathcal{U}, \mathcal{V})$ of propositional variables (with sorts for paths and respectively states) is defined by the following grammar:*

$$\begin{aligned} \mu\mathcal{L}_F \ni \varphi &::= \text{tt} \mid \text{ff} \mid q \mid \Phi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid [\lambda_F]\varphi \mid \eta q. \varphi \\ \mu\mathcal{L} \ni \Phi &::= \text{tt} \mid \text{ff} \mid p \mid [\lambda]\varphi \mid \Phi \wedge \Phi \mid \Phi \vee \Phi \end{aligned}$$

where $q \in \mathcal{U}$, $p \in \mathcal{V}$, $\eta \in \{\mu, \nu\}$, $\lambda_F \in \Lambda_F$ and $\lambda \in \Lambda$.

That is, *path formulas* φ are built from propositional variables q and *state formulas* Φ using positive boolean operators, modal operators $[\lambda_F]$ and fixpoint operators, while state formulas Φ are built from atomic propositions p and modal formulas $[\lambda]\varphi$ using positive boolean operators.

Such languages are interpreted over pairs consisting of a $T \circ F$ -coalgebra (X, γ) and a 2-sorted valuation $V : (\mathcal{U}, \mathcal{V}) \rightarrow (PZ_X, PX)$ (interpreting path and state variables as sets of maximal executions and respectively of states), by making use of the maximal execution map $\text{exec}_{\gamma} : X \rightarrow TZ_X$.

Definition 3 ([2]). *Given a $T \circ F$ -coalgebra (X, γ) and a 2-sorted valuation $V : (\mathcal{U}, \mathcal{V}) \rightarrow (PZ_X, PX)$, the semantics $\llbracket \varphi \rrbracket_{\gamma, V} \in PZ_X$ of path formulas $\varphi \in \mu\mathcal{L}_F$ and $\llbracket \Phi \rrbracket_{\gamma, V} \in PX$ of state formulas $\Phi \in \mu\mathcal{L}$ is defined by:*

$$\begin{aligned} \llbracket q \rrbracket_{\gamma, V} &= V(q) \\ \llbracket \Phi \rrbracket_{\gamma, V} &= P(\pi_1 \circ \zeta_X)(\llbracket \Phi \rrbracket_{\gamma, V}) \\ \llbracket [\lambda_F]\varphi \rrbracket_{\gamma, V} &= (P(\pi_2 \circ \zeta_X) \circ (\lambda_F)_{Z_X})(\llbracket \varphi \rrbracket_{\gamma, V}) \\ \llbracket \mu q. \varphi \rrbracket_{\gamma, V} &= \text{lfp}((\varphi)_q^{\gamma, V}) \\ \llbracket \nu q. \varphi \rrbracket_{\gamma, V} &= \text{gfp}((\varphi)_q^{\gamma, V}) \\ \llbracket p \rrbracket_{\gamma, V} &= V(p) \\ \llbracket [\lambda]\varphi \rrbracket_{\gamma, V} &= (P\text{exec}_{\gamma} \circ \lambda_{Z_X})(\llbracket \varphi \rrbracket_{\gamma, V}) \end{aligned}$$

and the usual clauses for the boolean operators, where, for $q \in \mathcal{U}$, $(\varphi)_q^{\gamma, V} : PX \rightarrow PX$ denotes the monotone map defined by $(\varphi)_q^{\gamma, V}(Y) = \llbracket \varphi \rrbracket_{\gamma, V'}$ with $V'(p) = V(p)$ for $p \in \mathcal{V}$, $V'(q) = Y$ and $V'(r) = V(r)$ for $r \in \mathcal{U}$, $r \neq q$, whereas $\text{lfp}(-)$ and $\text{gfp}(-)$ construct least and respectively greatest fixpoints.

Thus, a maximal execution satisfies a state formula Φ (regarded as a path formula) precisely when the first state of that execution (obtained by applying $\pi_1 \circ \zeta_X : Z_X \rightarrow X$) satisfies Φ . The F -coalgebra structure $\pi_2 \circ \zeta_X : Z_X \rightarrow FZ_X$ on the set of maximal executions is used to define when an execution satisfies the path

formula $[\lambda_F]\varphi$. Finally, to obtain the set of states satisfying the state formula $[\lambda]\varphi$, with φ a path formula, one uses the map $PZ_X \xrightarrow{(\lambda)_{Z_X}} PTZ_X \xrightarrow{P\text{exec}_\gamma} PX$ to go from a set of maximal executions (those satisfying φ) to a set of states.

In what follows, we restrict attention to coalgebras over \mathbf{Set} and take $U = \mathbf{Id}$ and $P = \hat{\mathcal{P}}$. Moreover, we only consider the case $T = \mathcal{P}^+$ and take $\Lambda = \{\lambda_\square\}$ with $\lambda_\square : \hat{\mathcal{P}} \implies \hat{\mathcal{P}} \circ \mathcal{P}^+$ given by $(\lambda_\square)_X(Y) = \mathcal{P}Y$. In this case, one can consider a fragment of the above language where atomic propositions $p \in \mathcal{V}$ are regarded as path formulas (and hold on a path if they hold in its first state), and where state formulas can only be of the form $\square\varphi$ with φ a path formula, and can not be viewed as path formulas. This yields what we call *linear* coalgebraic temporal logics for $\mathcal{P}^+ \circ F$ -coalgebras, where the linearity pertains to the branching structure arising from the presence of \mathcal{P}^+ : the state formula $\square\varphi$ requires *all* maximal executions from a particular state to satisfy the path formula φ . The syntax of such a fragment is thus parameterised by a choice Λ_F of monotone predicate liftings for the functor F , and can be described as follows:

$$\begin{aligned} \varphi &::= \mathbf{tt} \mid \mathbf{ff} \mid q \mid p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid [\lambda_F]\varphi \mid \eta q.\varphi \\ \Phi &::= \square\varphi \end{aligned}$$

The semantics inherited by this fragment provides, for each pair consisting of a $\mathcal{P}^+ \circ F$ -coalgebra (X, γ) and a 2-sorted valuation $V : (\mathcal{U}, \mathcal{V}) \rightarrow (PZ_X, PX)$, an interpretation of path formulas as subsets of Z_X , and an interpretation of state formulas as subsets of X . In particular, the interpretation of atomic propositions $p \in \mathcal{V}$ is given by $\llbracket p \rrbracket_{\gamma, V} = P(\pi_1 \circ \zeta_X)(V(p))$.

Two observations are now worth making. Firstly, the use of the final $X \times F$ -coalgebra (Z_X, ζ_X) in defining the semantics of path formulas is not needed anymore: since state formulas cannot be viewed as path formulas, the final F -coalgebra would suffice. Secondly, the syntax of path formulas is that of the coalgebraic μ -calculus considered in [3], after regarding propositional variables in \mathcal{V} as nullary modal operators. Based on these observations, the above syntactic fragment can be given an equivalent semantics that makes direct use of the semantics of the coalgebraic μ -calculus (as given in [3]). This is achieved by regarding a pair consisting of a $\mathcal{P}^+ \circ F$ -coalgebra with carrier X and a valuation of type $\mathcal{V} \rightarrow \mathcal{P}X$ as a coalgebra of $\mathcal{P}^+(\mathcal{P}\mathcal{V} \times F)$, and by viewing the elements of $\mathcal{V} \cup \Lambda_F$ as predicate liftings for the functor $\mathcal{P}\mathcal{V} \times F$: an atomic proposition $p \in \mathcal{V}$ yields a nullary predicate lifting $\lambda_p : 1 \implies \hat{\mathcal{P}}(\mathcal{P}\mathcal{V} \times F)$ for $\mathcal{P}\mathcal{V} \times F$ given by $(\lambda_p)_X = \{(P, y) \in \mathcal{P}\mathcal{V} \times FX \mid p \in P\}$, whereas a predicate lifting $\lambda_F : \hat{\mathcal{P}} \implies \hat{\mathcal{P}}F$ for F yields the predicate lifting $\hat{\mathcal{P}}\pi_2 \circ \lambda_F : \hat{\mathcal{P}} \implies \hat{\mathcal{P}}(\mathcal{P}\mathcal{V} \times F)$ for $\mathcal{P}\mathcal{V} \times F$.

The next result gives an equivalent semantics for the linear fragments of path-based coalgebraic temporal logics, solely in terms of maximal trace maps:

Proposition 4. *Let $(Z_{\mathcal{P}\mathcal{V}}, \zeta_{\mathcal{P}\mathcal{V}})$ denote the final $\mathcal{P}\mathcal{V} \times F$ -coalgebra, let (X, γ) be an arbitrary $\mathcal{P}\mathcal{V} \times (\mathcal{P}^+ \circ F)$ -coalgebra (incorporating a $\mathcal{P}^+ \circ F$ -coalgebra and a valuation of type $\mathcal{V} \rightarrow \mathcal{P}(X)$), and let $(X, \tilde{\gamma})$ be the $\mathcal{P}^+ \circ (\mathcal{P}\mathcal{V} \times F)$ -coalgebra given by $\mathbf{st}_{\mathcal{P}\mathcal{V}, F} \circ \gamma$, where $\mathbf{st}_{\mathcal{P}\mathcal{V}, F} : \mathcal{P}\mathcal{V} \times (\mathcal{P}^+ \circ F) \implies \mathcal{P}^+ \circ (\mathcal{P}\mathcal{V} \times F)$ is the*

strength of the monad \mathcal{P}^+ :

$$X \xrightarrow{\gamma} \mathcal{P}\mathcal{V} \times \mathcal{P}^+(FX) \xrightarrow{\text{st}_{\mathcal{P}\mathcal{V}, FX}} \mathcal{P}^+(\mathcal{P}\mathcal{V} \times FX)$$

Then $x \models_{\gamma} \Box \varphi$ iff $z \models_{\zeta_{\mathcal{P}\mathcal{V}}} \varphi$ for all $z \in \text{tr}_{\tilde{\gamma}}(x)$, where $\models_{\zeta_{\mathcal{P}\mathcal{V}}}$ is the satisfaction relation between states of $(Z_{\mathcal{P}\mathcal{V}}, \zeta_{\mathcal{P}\mathcal{V}})$ and variable-free formulas of the coalgebraic μ -calculus induced by $\mathcal{V} \cup \Lambda_F$, and $\text{tr}_{\tilde{\gamma}} : X \rightarrow \mathcal{P}^+Z_{\mathcal{P}\mathcal{V}}$ is the trace map of $(X, \tilde{\gamma})$.

- Example 5.* 1. Let $F = A \times \text{Id}$ and $\Lambda_F = \{a \mid a \in A\} \cup \{\text{O}\}$, with the predicate liftings $\lambda_a : 1 \implies \hat{\mathcal{P}} \circ (A \times \text{Id})$ for $a \in A$ and $\lambda_{\text{O}} : \hat{\mathcal{P}} \implies \hat{\mathcal{P}} \circ (A \times \text{Id})$ being given by $(\lambda_a)_X(*) = \{a\} \times X$ and $(\lambda_{\text{O}})_X(Y) = A \times Y$. This yields a linear temporal logic for $\mathcal{P}^+ \circ (A \times \text{Id})$ -coalgebras, with, for example, the formula $\Box \mu X.(a \vee \text{O}X)$ stating that the label a occurs along every trace of a pointed $\mathcal{P}^+ \circ (A \times \text{Id})$ -coalgebra.
2. For $F = 1 + A \times \text{Id}$, one can obtain a logic that also talks about termination by extending a variant of the previous set of predicate liftings with the nullary predicate lifting $\lambda_{\perp} : 1 \implies \hat{\mathcal{P}} \circ (1 + A \times \text{Id})$ given by $(\lambda_{\perp})_X(*) = \{\iota_1(*)\}$. In the resulting logic, the formula $\Box \mu X.(\perp \vee \text{O}X)$ states that all maximal executions of a pointed $\mathcal{P}^+ \circ (1 + A \times \text{Id})$ -coalgebra are finite ones.
3. For $F = A + \text{Id} \times \text{Id}$, a natural choice for Λ_F is the set $\{a \mid a \in A\} \cup \{[\pi_1], [\pi_2]\}$, with the predicate liftings $\lambda_a : 1 \implies \hat{\mathcal{P}} \circ (A + \text{Id} \times \text{Id})$ for $a \in A$ and $\lambda_{\pi_1}, \lambda_{\pi_2} : \hat{\mathcal{P}} \implies \hat{\mathcal{P}} \circ (A + \text{Id} \times \text{Id})$ being given by $(\lambda_a)_X(*) = \{\iota_1(a)\}$, $(\lambda_{\pi_1})_X(Y) = \iota_2(Y \times X)$ and $(\lambda_{\pi_2})_X(Y) = \iota_2(X \times Y)$. In the resulting temporal logic, the formula $\Box \mu X.(a \vee [\pi_1]X \vee [\pi_2]X)$ requires all maximal traces of a pointed $\mathcal{P}^+ \circ (A + \text{Id} \times \text{Id})$ -coalgebra to have a leaf labelled by a .

Linear coalgebraic temporal logics can thus be viewed as coalgebraic generalisations of Linear Temporal Logic [7]. In what follows, we consider $\mathcal{P}^+ \circ F$ -coalgebras instead of $\mathcal{P}^+ \circ (\mathcal{P}\mathcal{V} \times F)$ -coalgebras, and assume that the interpretation for any atomic propositions has already been incorporated into the functor F .

Coalgebra Automata We now recall the various coalgebraic notions of automaton that are required for the subsequent development. The definitions that follow assume that the functor $F : \text{Set} \rightarrow \text{Set}$ preserves weak pullbacks.

Definition 6 ([5]). A (parity) alternating F -automaton is a tuple (A, a_0, δ, Ω) with A a finite set of states, a_0 the initial state, $\delta : A \rightarrow \mathcal{P}\mathcal{P}FA$ the transition function and $\Omega : A \rightarrow \omega$ a parity map. If for all $a \in A$, all elements of $\delta(a)$ are singletons, the automaton is called non-deterministic. Also, if for all $a \in A$, $\delta(a)$ is of the form $\{\{\varphi_a\}\}$ for some $\varphi_a \in FA$, the automaton is called deterministic.

Definition 7 ([5]). Given an F -automaton $\mathbb{A} = (A, a_0, \delta, \Omega)$ and an F -coalgebra $\mathbb{S} = (S, \gamma)$, the acceptance game $\mathcal{G}(\mathbb{A}, \mathbb{S})$ is the parity game defined by:

Position: b	Player: $P(b)$	Admissible moves: $E[b]$	Priority
$(s, a) \in S \times A$	\exists	$\{(s, \Phi) \in S \times \mathcal{P}(FA) \mid \Phi \in \delta(a)\}$	$\Omega(a)$
$(s, \Phi) \in S \times \mathcal{P}(FA)$	\forall	$\{(s, \varphi) \in S \times FA \mid \varphi \in \Phi\}$	0
$(s, \varphi) \in S \times FA$	\exists	$\{(Z \subseteq S \times A \mid (\gamma(s), \varphi) \in \text{Rel}(F)(Z))\}$	0
$Z \in \mathcal{P}(S \times A)$	\forall	Z	0

The automaton \mathbb{A} is said to accept a pointed coalgebra (\mathbb{S}, s_0) if (s_0, a_0) is a winning position for \exists in $\mathcal{G}(\mathbb{A}, \mathbb{S})$.

Positions $(s, a) \in S \times A$ of the acceptance game are called *basic* positions. Here, the aim of \exists is to show that the state s of the coalgebra fits the description provided by the state a of the automaton. The game also contains intermediary positions of three other types. Before arriving at a basic position again, \exists must provide a *witnessing relation* $Z \subseteq S \times A$ in a position of type $S \times FA$.

The game $\mathcal{G}(\mathbb{A}, \mathbb{S})$ is closely related to the *bisimulation game* of [1].

Definition 8 ([1]). Let $\mathbb{S} = (S, \gamma)$ and $\mathbb{S}' = (S', \gamma')$ be two F -coalgebras. The bisimulation game $\mathcal{B}(\mathbb{S}, \mathbb{S}')$ is the graph game defined by:

Position: b	Player: $P(b)$	Admissible moves: $E[b]$
$(s, s') \in S \times S'$	\exists	$\{Z \subseteq S \times S' \mid (\gamma(s), \gamma(s')) \in \text{Rel}(F)(Z)\}$
$Z \subseteq S \times S'$	\forall	Z

Finite plays of $\mathcal{B}(\mathbb{S}, \mathbb{S}')$ are lost by the player who can not move, whereas infinite plays are won by \exists .

[1] shows that the bisimilarity of two states $s \in S$ and $s' \in S'$ is equivalent to the existence of a winning strategy for \exists from position (s, s') in $\mathcal{B}(\mathbb{S}, \mathbb{S}')$.

For a *deterministic* automaton \mathbb{A} , the acceptance game $\mathcal{G}(\mathbb{A}, \mathbb{S})$ is essentially the bisimulation game $\mathcal{B}(\mathbb{S}, \mathbb{A}')$, where $\mathbb{A}' = (A, \alpha, a_0)$ is the F -coalgebra given by $\alpha(a) = \varphi_a$ for $a \in A$. However, for arbitrary alternating automata, \exists and \forall must play a small sub-game in each round starting in a basic position, in order to arrive at a position where \exists must provide a witnessing relation Z . The first occurrence of the powerset functor in the codomain of the transition function of an alternating automaton indicates a choice for \exists in this sub-game, whereas the second occurrence of \mathcal{P} indicates a choice for \forall . In the case of non-deterministic automata, \forall has no real choice while playing the previously-mentioned sub-game.

Remark 9. Any finite, pointed F -coalgebra $\mathbb{S} = (S, \gamma, s_0)$ can be turned into an F -automaton $\mathbb{A}_{\mathbb{S}, s_0} = (S, s_0, \Delta_\gamma, \Omega_0)$, where $\Delta_\gamma(s) = \{\{\gamma(s)\}\}$ and $\Omega_0(s) = 0$ for $s \in S$. Moreover, this automaton accepts a pointed F -coalgebra (D, δ, d_0) if and only if the states s_0 and d_0 are bisimilar (see e.g. [8] for details).

Automata that account for the satisfaction of coalgebraic μ -calculus formulas by states of F -coalgebras were considered in [3]. For a set A of monotone predicate liftings for F , A -automata differ from F -automata in that their transition functions have type $A \rightarrow \mathcal{L}_0 A(A)$, with $\mathcal{L}_0(X)$ the set of lattice terms over X . In the acceptance game for a A -automaton, the aim of \exists is to show that a pointed coalgebra satisfies the formula encoded by the automaton. Thus, occurrences of conjunction in the lattice terms used to define the transition function of a A -automaton correspond to choices of \forall in the associated acceptance game, whereas occurrences of disjunction correspond to choices of \exists . These choices are, however, made implicit by the presentation of the acceptance game given below.

Definition 10 ([3]). Let Λ be a set of monotone predicate liftings for F . A Λ -automaton is a tuple (A, a_0, δ, Ω) with A a finite set of states, a_0 the initial state, $\delta : A \rightarrow \mathcal{L}_0\Lambda(A)$ the transition function and $\Omega : A \rightarrow \omega$ a parity map.

The following is a slight reformulation of the acceptance game for a Λ -automaton:

Definition 11 ([3]). Given a Λ -automaton $\mathbb{A} = (A, a_0, \delta, \Omega)$ and an F -coalgebra $\mathbb{S} = (S, \gamma)$, the acceptance game $\mathcal{G}(\mathbb{A}, \mathbb{S})$ is the parity game defined by:

Position: b	Player: $P(b)$	Admissible moves: $E[b]$	Priority
$(s, a) \in S \times A$	\exists	$\{(Z \subseteq S \times A \mid \gamma(s) \in \llbracket \delta(a) \rrbracket_Z^1\}$	$\Omega(a)$
$Z \in \mathcal{P}(S \times A)$	\forall	Z	0

where for $Z \subseteq S \times A$ and $\varphi \in \mathcal{L}_0\Lambda(A)$, $\llbracket \varphi \rrbracket_Z^1 \in \mathcal{P}(FS)$ is defined inductively by

$$\llbracket [\lambda](a_1, \dots, a_n) \rrbracket_Z^1 = \lambda_S(\llbracket a_1 \rrbracket_Z, \dots, \llbracket a_n \rrbracket_Z)$$

and the usual clauses for conjunction and disjunction. Here, $\llbracket a \rrbracket_Z \in \mathcal{P}S$ is given by $\llbracket a \rrbracket_Z := \{s \in S \mid (s, a) \in Z\}$, for $a \in A$. The automaton \mathbb{A} is said to accept a pointed coalgebra (\mathbb{S}, s_0) if (s_0, a_0) is a winning position for \exists in $\mathcal{G}(\mathbb{A}, \mathbb{S})$.

As shown in [3], any coalgebraic μ -calculus formula φ can be mapped to a Λ -automaton accepting precisely the pointed F -coalgebras that satisfy φ .

3 Automata-Based Characterisation of Maximal Traces

We now provide an automata-theoretic characterisation of the set of maximal traces of a pointed $\mathcal{P}^+ \circ F$ -coalgebra (X, γ, x_0) with finite carrier, by defining a non-deterministic F -automaton that accepts exactly those traces (elements of the final F -coalgebra) which belong to $\text{tr}_\gamma(x_0)$. The construction of this automaton is straightforward: we simply regard the coalgebra map γ as the transition function of a *non-deterministic* F -automaton with a trivial parity map. As a result, for each trace of x_0 , \exists 's moves in basic positions (z, x) of the acceptance game for this F -automaton can be chosen so as to select precisely the values $y \in \gamma(x)$ that generate the given trace.

In what follows, F is assumed to be a polynomial and standard functor (hence pullback and non-empty intersection preserving), and $\lambda : F\mathcal{P}^+ \Longrightarrow \mathcal{P}^+F$ is assumed to be as in (2). Both restrictions are required in order to make use of the alternative definition of the trace map of a $\mathcal{P}^+ \circ F$ -coalgebra given in (3).

Definition 12. The trace automaton induced by a pointed $\mathcal{P}^+ \circ F$ -coalgebra (X, γ, x_0) is the non-deterministic F -automaton $\mathbb{X}_\gamma = (X, \delta_\gamma, x_0, \Omega_0)$ where:

- $\delta_\gamma : X \rightarrow \mathcal{P}\mathcal{P}FX$ is given by $\delta_\gamma(x) = \{\{y\} \mid y \in \gamma(x)\}$ for $x \in X$,
- $\Omega_0 : X \rightarrow \omega$ is given by $\Omega_0(x) = 0$ for $x \in X$.

This definition should be compared with that of the automaton in Remark 9 – the difference is that the non-determinism provided by the coalgebra γ is now used to provide a *non-deterministic* F -automaton, as opposed to a *deterministic* $\mathcal{P}^+ \circ F$ -automaton. Our first main result can now be stated as follows.

Theorem 13. *Let (X, γ, x_0) be a pointed $\mathcal{P}^+ \circ F$ -coalgebra, and let (Z, ζ) denote the final F -coalgebra. Then, for $z_0 \in Z$, $z_0 \in \text{tr}_\gamma(x_0)$ iff \mathbb{X}_γ accepts the pointed coalgebra (Z, ζ, z_0) .*

To prove this result we need the following definition.

Definition 14. *For $y \in FY$, let*

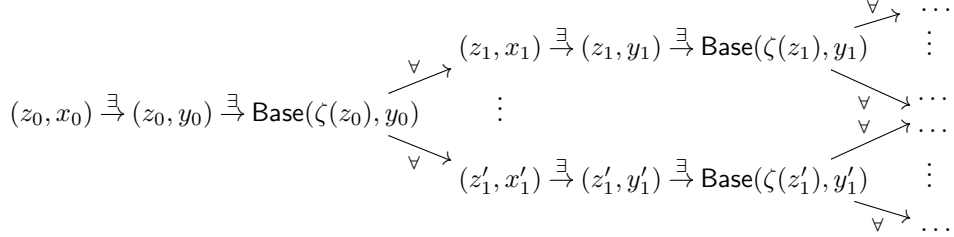
$$\text{Base}(y) := \bigcap \{Y' \subseteq Y \mid y \in FY'\}$$

For $(z, y) \in FZ \times FY$ such that $(z, y) \in \text{Rel}(F)(R)$ for some $R \subseteq Z \times Y$, let

$$\text{Base}(z, y) := \bigcap \{R \subseteq Z \times Y \mid (z, y) \in \text{Rel}(F)(R)\}$$

The preservation of non-empty intersections by F gives $y \in F\text{Base}(y)$. Thus, $\text{Base}(y)$ is the smallest set Y' with the property that $y \in FY'$. Similarly, one can show by induction on the structure of the polynomial functor F that $\text{Rel}(F)$ preserves non-empty intersections. As a result, whenever $\text{Base}(z, y)$ exists, it is the smallest relation R with the property that $(z, y) \in \text{Rel}(F)(R)$. We now return to the proof of Theorem 13 which, due to space limitations, is only outlined here.

Proof (sketch). For the *if* direction, let \mathcal{S} denote a winning strategy for \exists in $\mathcal{G}(\mathbb{X}_\gamma, (Z, \zeta, z_0))$, and assume w.l.o.g. that \mathcal{S} prescribes the smallest possible witnessing relations, namely $\text{Base}(\zeta(z), y)$, in positions $(z, y) \in Z \times FX$. The \mathcal{S} -conform $\mathcal{G}(\mathbb{X}_\gamma, (Z, \zeta, z_0))$ -plays can now be visualised as branches of the following tree (where positions in which \forall has no choice have been omitted):



\exists 's moves provided by \mathcal{S} can then be used to define an F -coalgebra structure $\xi : D \rightarrow FD$ on the set D of basic positions reachable from (z_0, x_0) through an \mathcal{S} -conform $\mathcal{G}(\mathbb{X}_\gamma, (Z, \zeta, z_0))$ -play. Moreover, the winning strategy \mathcal{S} can be mirrored in the bisimulation game $\mathcal{B}((Z, \zeta, z_0), (D, \xi, (z_0, x_0)))$, by letting \exists move in a position $(z, (z, x))$ to the position $\{(z', (z', x')) \mid (z', x') \in \text{Base}(\zeta(z), y)\}$, where (z, y) is the move prescribed by \mathcal{S} in position (z, x) . This yields a winning strategy for \exists in the bisimulation game, and hence $(Z, \zeta, z_0) \sim (D, \xi, (z_0, x_0))$. Using this and the definition of λ given in (2), one can show that $\pi_i(z_0) \in \gamma_i(x_0)$ for $i \in \omega$, which finally yields $z_0 \in \text{tr}_\gamma(x_0)$ by the definition of tr_γ in (1).

For the *only if* direction one can prove more generally that, for $x \in X$ and $z \in \text{tr}_\gamma(x)$, \exists has a (history-free) winning strategy from position (z, x) in $\mathcal{G}(\mathbb{X}_\gamma, (Z, \zeta, z_0))$. The key idea is to make use of the following property of the trace map, as proved in [4]:

$$\{\zeta(z) \mid z \in \text{tr}_\gamma(x)\} = \bigcup \{\lambda_Z((F\text{tr}_\gamma)(y)) \mid y \in \gamma(x)\}$$

for each $x \in X$. Using the above, $z \in \text{tr}_\gamma(x)$ yields some $y \in \gamma(x)$ such that $\zeta(z) \in \lambda_Z((F\text{tr}_\gamma)(y))$. This, in turn, yields a suitable choice of a trace $z_{x'} \in \text{tr}_\gamma(x')$ for each $x' \in \text{Base}(y)$. Our winning strategy now requires \exists to move in position (z, x) to (z, y) and then immediately to the relation $\{(z_{x'}, x') \mid x' \in \text{Base}(y)\}$. Clearly this guarantees that \exists will never be stuck in a play, and since all infinite $\mathcal{G}(\mathbb{X}_\gamma, (Z, \zeta, z_0))$ -plays are won by \exists , it follows that the proposed strategy is a winning strategy for \exists in $\mathcal{G}(\mathbb{X}_\gamma, (Z, \zeta, z_0))$. \square

Remark 15. Under the assumption that the coalgebra (X, γ) contains no duplicates (see Definition 16), an alternative proof of the previous result can be given by showing that the coalgebra (D, ξ) constructed in the above proof is a sub-coalgebra of the coalgebra (U, σ) defined in Section 2.

4 Model Checking Linear Coalgebraic Temporal Logics

This section defines a regular graph game, called the *non-emptiness game*, for deciding whether a formula $\Box\varphi$ of a linear coalgebraic temporal logic (induced by a set Λ of predicate liftings for F) holds in a finite, pointed $\mathcal{P}^+ \circ F$ -coalgebra \mathbb{X} . We represent the *negation* of the formula φ as a Λ -automaton⁴, and use the non-deterministic F -automaton induced by \mathbb{X} as a description of its traces. The game we define has the property that winning strategies of \exists correspond to traces of \mathbb{X} that are accepted by the Λ -automaton. Thus, the existence of a winning strategy for \exists is equivalent to the formula $\Box\varphi$ not holding in \mathbb{X} , while the winning strategy itself provides a counter-example for the statement $\mathbb{X} \models \Box\varphi$.

Although such a game can be defined for any pointed $\mathcal{P}^+ \circ F$ -coalgebra, the game has a much simpler presentation for coalgebras *with no duplicates*. We will therefore only define the game for such coalgebras, and show that any $\mathcal{P}^+ \circ F$ -coalgebra with finite carrier can be transformed into a $\mathcal{P}^+ \circ F$ -coalgebra with no duplicates, and whose carrier is still finite. The next definition formalises the idea of a state x occurring more than once in some $y \in FX$.

Definition 16. A $\mathcal{P}^+ \circ F$ -coalgebra (X, γ) contains duplicates if there exist $u \in X$, $y \in \gamma(u)$, $x \in \text{Base}(y)$ and $y' \in F(X + 1)$ such that:

- $\iota_1(x), \iota_2(*) \in \text{Base}(y')$,
- $y = F[1_X, x](y')$ (where $x : 1 \rightarrow X$ maps $*$ to x).

We then call $y \in FX$ a duplicate type of (X, γ) , and x a duplicate state of y .

Thus, an F -coalgebra (X, γ) contains duplicates if there exist $x, u \in X$ and $y \in \gamma(u)$ such that x occurs at least twice in y .

Example 17. Let $F = \{0, 1\} + \text{Id} \times \text{Id}$. The elements of the final F -coalgebra are (possibly infinite) binary trees with leaves labelled by either 0 or 1. Now let (X, γ) be the $\mathcal{P}^+ \circ F$ -coalgebra given by $X = \{x_0, x\}$, $\gamma(x_0) = \{(x, x)\}$, $\gamma(x) = \{0, 1\}$. In this case, (x, x) is a duplicate type of (X, γ) , and x is a duplicate state of (x, x) . We also note in passing that in this case, $\text{tr}_\gamma(x) = \{0, 1\}$, whereas $\text{tr}_\gamma(x_0)$ consists of all four binary trees of depth 1 with leaves labelled by either 0 or 1.

⁴ We assume that Λ contains enough predicate liftings to encode negations of formulas.

Lemma 18. *Assume that F preserves finite sets⁵. There exists an effective procedure translating any $\mathcal{P}^+ \circ F$ -coalgebra (X, γ) with finite carrier into a $\mathcal{P}^+ \circ F$ -coalgebra (X', γ') with finite carrier and no duplicates, and with a surjective $\mathcal{P}^+ \circ F$ -coalgebra homomorphism $\pi : (X', \gamma') \rightarrow (X, \gamma)$.*

The next definition will allow the formulation of the winning condition for the non-emptiness game.

Definition 19. *Let $\Omega : A \rightarrow \omega$ be a parity map. A trace through a sequence of relations $(R_i)_{i \in \omega}$ with $R_i \subseteq A \times A$ for $i \in \omega$ is an infinite sequence $(a_i)_{i \in \omega} \in A^\omega$ such that $(a_i, a_{i+1}) \in R_i$ for each $i \in \omega$. A trace $(a_i)_{i \in \omega}$ through $(R_i)_{i \in \omega}$ is called bad w.r.t. Ω if $\max\{k \mid k = \Omega(a_i) \text{ for infinitely many } i \in \omega\}$ is odd.*

Definition 20. *Let $\mathbb{X} = (X, \gamma, x_0)$ be a pointed $\mathcal{P}^+ \circ F$ -coalgebra with no duplicates, and let $\mathbb{A} = (A, a_0, \delta, \Omega)$ be a Λ -automaton. The non-emptiness game $\mathcal{G}_{\neq \emptyset}(\mathbb{X}, \mathbb{A})$ is the graph game defined by:*

Position	Player	Admissible moves
$(x, R) \in X \times \mathcal{P}(A \times A)$	\exists	$\{(y, \text{ran}(R)) \in FX \times \mathcal{P}A \mid y \in \gamma(x)\}$
$(y, A') \in FX \times \mathcal{P}A$	\exists	$\{(\text{Base}(y), Z) \mid Z : A' \rightarrow \mathcal{P}(\text{Base}(y) \times A)$ s.t. $y \in \llbracket \delta(a) \rrbracket_{Z_a}$ for $a \in A'\}$
$(B, Z) \in \mathcal{P}(X) \times \mathcal{P}(X \times A)^{A'}$	\forall	$\{(x, \bigcup_{a \in A'} \{(a, a') \mid (x, a') \in Z_a\}) \mid x \in B\}$

with initial position $(x_0, \{(a_0, a_0)\})$, where $\text{ran}(R)$ denotes the range of the relation R . An infinite match with basic positions $(x_0, R_0), (x_1, R_1), \dots$ is won by \exists if and only if no trace through the sequence of relations $R_0 R_1 \dots$ is bad w.r.t. Ω .

The basic positions of $\mathcal{G}_{\neq \emptyset}(\mathbb{X}, \mathbb{A})$ are given by pairs consisting of a state $x \in X$ and a relation $R \subseteq A \times A$, whereas the witnessing relations for positions of type (y, A') can be regarded as families of functions of type $Z_a : X \rightarrow \mathcal{P}(A)$ (one for each $a \in A'$). To explain the reasons behind this definition and the assumption that \mathbb{X} has no duplicates, let us imagine that the initial position of $\mathcal{G}_{\neq \emptyset}(\mathbb{X}, \mathbb{A})$ was (x_0, a_0) (the obvious choice, given that \exists 's goal in this game is to prove that x_0 admits a trace that "satisfies the formula a_0 "). In this position, \exists would have to provide a choice of $y_0 \in \gamma(x_0)$ that "makes the formula a_0 true". This move would be followed by \exists providing a witnessing relation $Z \subseteq X \times A$ for (y_0, a_0) . However, the play would not be able to continue with \forall choosing an element of this relation, which would then result in a new basic position (x_1, a_1) , since in situations where *several* pairs in Z have the *same* first component $x_1 \in X$, the second components of those pairs may or may not need to be satisfied by the same trace of x_1 . If x_1 is not a duplicate state of y_0 , it is clear that a *single* trace of x_1 should make all formulas a_1 with $(x_1, a_1) \in Z$ true. If, on the other hand, x_1 is a duplicate state of y_0 , different occurrences of x_1 could (and might need to) be unfolded in different ways in order to satisfy a_0 in y_0 . For instance,

⁵ This further restricts polynomial functors by only allowing finite exponents and finite constant functors. However, the restriction to finite constant functors is superfluous.

given the pointed coalgebra (X, γ, x_0) of Example 17, a Λ -automaton could be devised that accepts only traces with both 0 and 1 as leaves, and to obtain a trace of x_0 with this property, different traces of x would need to be considered for the two occurrences of x in $\gamma(x_0)$. In the presence of duplicate states, the witnessing relation Z does not provide sufficient information to decide which formulas should hold for which unfoldings. On the other hand, by assuming that (X, γ) has no duplicates, it becomes much easier to define \exists 's possible moves in positions given by witnessing relations Z : *all* pairs in Z with the same first component x_1 must be witnessed by the *same* choice of $y_1 \in \gamma(x_1)$. Now to accommodate this, \exists 's moves in basic positions would have to be of type $Z : X \rightarrow \mathcal{P}(A)$. However, to define a winning condition for infinite games, we would have to assign priorities to basic positions of type (x, A') with $A' \in \mathcal{P}(A)$, which is not possible in a meaningful way. We therefore take the fairly standard approach of using traces through sequences of relations for defining the winning condition. This leads to basic positions of type $X \times \mathcal{P}(A \times A)$, where in a position (x, R) , \exists 's goal is to prove true all formulas in $\text{ran}(R)$. Finally, the presence of $\text{Base}(y)$ in the definition of admissible moves for \exists in positions (y, A') is justified by the fact that only pairs (x, a') with $x \in \text{Base}(y)$ are relevant to the satisfaction by $y \in FX$ of the formula represented by a .

Remark 21. A version of the non-emptiness game that also applies to coalgebras with duplicates could be defined by moving from witnessing relations of type $Z_a \in \mathcal{P}(X \times A)$ (or $Z_a : X \rightarrow \mathcal{P}(A)$) to witnessing relations of type $Z : X \rightarrow \mathcal{P}(\mathcal{P}(A))$, where elements of A that must be simultaneously satisfied on some trace of x are grouped appropriately in $Z_a(x)$. However, the conditions specifying that Z is a witnessing relation would become much more complex.

The non-emptiness game proceeds as follows:

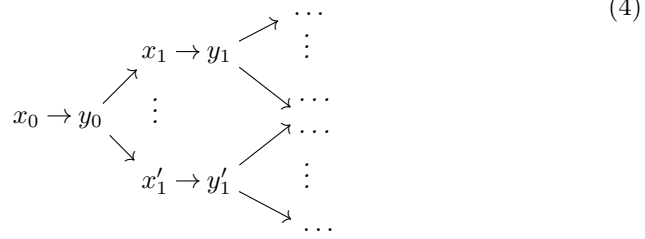
- in a basic position (x, R) (in which \exists must show that some trace of x makes all formulas in $\text{ran}(R)$ true), \exists chooses $y \in \gamma(x)$ and moves to $(y, \text{ran}(R))$;
- in a position (y, A') (in which \exists must show that a suitable choice of trace for each $x \in \text{Base}(y)$ makes all formulas in A' true), \exists provides a suitable witnessing relation $Z_a \subseteq X \times A$ for each formula $a \in A'$;
- in a position (B, Z) with $Z : A' \rightarrow \mathcal{P}(B \times A)$, \forall chooses some $x \in B$ and collects all second components of pairs (x, a') in one of the Z_a s – these formulas must all be satisfied by the *same* trace of x . The resulting position records the corresponding a for a pair (x, a') , to be used in the formulation of the winning condition. The game is now again in a basic position.

The game $\mathcal{G}_{\neq\emptyset}(\mathbb{X}, \mathbb{A})$ is ω -regular: there exists a parity $\mathcal{P}(A \times A)$ -word automaton accepting exactly those sequences of relations which do not contain a bad trace.

The second main result of the paper now states that winning strategies for \exists in $\mathcal{G}_{\neq\emptyset}(\mathbb{X}, \mathbb{A})$ correspond to traces of \mathbb{X} that are accepted by \mathbb{A} . This is proved with the help of the following lemma.

Lemma 22. *Let (X, γ) be a $\mathcal{P}^+ \circ F$ -coalgebra with no duplicates, and let (U, σ) be the F -coalgebra defined in Section 2. Then, states $u \in U$ are in one-to-one*

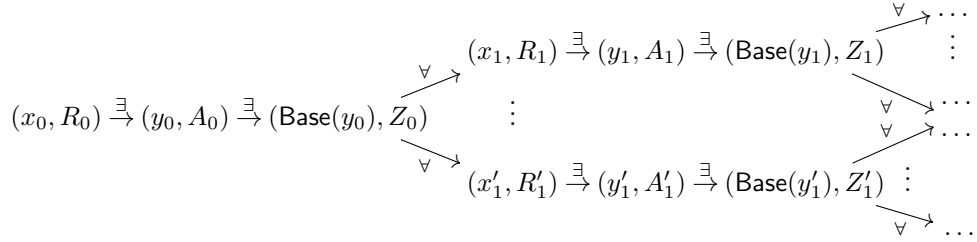
correspondence with infinite trees of the following shape:



with $y_i \in \gamma_i(x_i)$ and with one child of y_i for each $x_{i+1} \in \mathbf{Base}(y_i)$.

Theorem 23. Let $\mathbb{X} = (X, \gamma, x_0)$ be a pointed $\mathcal{P}^+ \circ F$ -coalgebra and let $\mathbb{A} = (A, a_0, \delta, \Omega)$ be a Λ -automaton. Then \exists has a winning strategy in $\mathcal{G}_{\neq \emptyset}(\mathbb{X}, \mathbb{A})$ iff there exists a trace $z \in \text{tr}_\gamma(x_0)$ such that (Z, ζ, z) is accepted by \mathbb{A} .

Proof (sketch). For the *only if* direction, let \mathcal{S} denote a winning strategy for \exists in $\mathcal{G}_{\neq \emptyset}(\mathbb{X}, \mathbb{A})$, and observe that \mathcal{S} -conform $\mathcal{G}_{\neq \emptyset}(\mathbb{X}, \mathbb{A})$ -plays can be visualised as branches of the following tree:



where each node of type $(\mathbf{Base}(y_i), Z_i)$ has exactly one child for each $x_{i+1} \in \mathbf{Base}(y_i)$. A subtree of this tree rooted in some (x_i, R_i) determines an infinite tree of the type required by Lemma 22, and this, in turn, yields a state $u_i \in U$. Moreover, u_0 describes the desired behaviour of a trace of x_0 that is accepted by the automaton \mathbb{A} , since the above tree provides winning strategies for \exists in each of the acceptance games $\mathcal{G}(\mathbb{X}, (U, \sigma, u_0))$ and $\mathcal{G}(\mathbb{A}, (U, \sigma, u_0))$. On the one hand, \mathcal{S} -conform $\mathcal{G}_{\neq \emptyset}(\mathbb{X}, \mathbb{A})$ -plays can be mirrored in the acceptance game $\mathcal{G}(\mathbb{X}, (U, \sigma, u_0))$ by letting \exists move in a position (x_i, u_i) to the witnessing relation $\{(x_{i+1}, u_{i+1}) \mid x_{i+1} \in \mathbf{Base}(y_i)\}$, and this yields a winning strategy for \exists in $\mathcal{G}(\mathbb{X}, (U, \sigma, u_0))$. On the other hand, \mathcal{S} -conform $\mathcal{G}_{\neq \emptyset}(\mathbb{X}, \mathbb{A})$ -plays can be mirrored in the acceptance game $\mathcal{G}(\mathbb{A}, (U, \sigma, u_0))$: pairs consisting of a path through the above tree and a trace through the sequence of relations determined by that path correspond to $\mathcal{G}(\mathbb{A}, (U, \sigma, u_0))$ -plays in which \exists plays essentially the witnessing relations Z_{a_i} (but with x_i substituted by u_i) in positions of type (u_i, a_i) with $a_i \in A_i$. Moreover, the property that any such path is winning for \exists in $\mathcal{G}_{\neq \emptyset}(\mathbb{X}, \mathbb{A})$ translates to any $\mathcal{G}(\mathbb{A}, (U, \sigma, u_0))$ -play that is played according to the proposed strategy being winning for \exists in $\mathcal{G}(\mathbb{A}, (U, \sigma, u_0))$. Thus, u_0 defines a trace of $\mathbb{X} = (X, \gamma, x_0)$ that satisfies the property described by the Λ -automaton \mathbb{A} .

For the *if* direction, note that a trace $z \in \text{tr}_\gamma(x_0)$ such that (Z, ζ, z) is accepted by \mathbb{A} yields an element $u_0 \in U$ that is accepted by \mathbb{A} (as acceptance by \mathbb{A} is invariant under bisimulation), as well as a winning strategy \mathcal{S} for \exists in the acceptance game $\mathcal{G}(\mathbb{A}, (U, \sigma, u_0))$. Lemma 22 can now be used to obtain an infinite tree similar to that in (4), which, in turn, yields a winning strategy \mathcal{S}' for \exists in $\mathcal{G}_{\neq\emptyset}(\mathbb{X}, \mathbb{A})$: the choices of $y_i \in \gamma(x_i)$ made in the definition of u_0 provide the choices required in basic positions (x_i, R_i) reached from $(x_0, \{(a_0, a_0)\})$ through \mathcal{S}' -conform plays, whereas in positions of the form $(y_i, A_i) \in FX \times PA$ with $A_i = \text{ran}(R_i)$, \mathcal{S}' prescribes that \exists moves to $(\text{Base}(y_i), (Z_a)_{a \in A_i})$ with $Z_a \in \mathcal{P}(\text{Base}(y_i) \times A)$ being obtained from \exists 's \mathcal{S} -conform move in the position (u_i, a) of $\mathcal{G}(\mathbb{A}, (U, \sigma, u_0))$, by replacing any $u_{i+1} \in \text{Base}(u_i)$ with the corresponding x_{i+1} . The fact that \mathcal{S} is winning for \exists in $\mathcal{G}(\mathbb{A}, (U, \sigma, u_0))$ then results in \mathcal{S}' being winning for \exists in $\mathcal{G}_{\neq\emptyset}(\mathbb{X}, \mathbb{A})$. \square

5 Concluding Remarks

We provided an automata-theoretic characterisation of the set of traces of a finite, pointed $\mathcal{P}^+ \circ F$ -coalgebra, with F a polynomial endofunctor. Next, we defined a regular graph game that can be used to decide whether a formula of a linear coalgebraic temporal logic (also introduced in this paper) holds in a finite, pointed $\mathcal{P}^+ \circ F$ -coalgebra that contains no duplicates.

Future work includes generalising these results to non-polynomial functors F , studying model-checking algorithms based on such regular games, and extending the techniques proposed here to more general coalgebraic types and path-based temporal logics, as considered in [2].

References

1. A. Baltag. A logic for coalgebraic simulation. In *Proc. CMCS 2000*, volume 33 of *Electr. Notes Theor. Comput. Sci.*, pages 42–60, 2000.
2. C. Cirstea. Generic infinite traces and path-based coalgebraic temporal logics. In *Proc. CMCS 2010*, volume 264 of *Electr. Notes Theor. Comput. Sci.*, pages 83–103, 2010.
3. G. Fontaine, R. Leal, and Y. Venema. Automata for coalgebras: an approach using predicate liftings. In *Proc. ICALP 2010*, volume 6199 of *Lecture Notes in Computer Science*, pages 381–392, 2010.
4. B. Jacobs. Trace semantics for coalgebras. In *Proc. CMCS 2004*, volume 106 of *Electr. Notes Theor. Comput. Sci.*, pages 167–184, 2004.
5. C. Kupke and Y. Venema. Coalgebraic automata theory: basic results. *Logical Methods in Computer Science*, 4(4), 2008.
6. R. Mazala. Infinite games. In *Automata, Logics, and Infinite Games*, Lecture Notes in Computer Science, pages 23–42. Springer, 2002.
7. M. Vardi. Automata-theoretic techniques for temporal reasoning. In *Handbook of Modal Logic*, pages 971–990. Addison Wesley, 2005.
8. Y. Venema. Automata and fixed point logics: a coalgebraic perspective. *Information and Computation*, 204:637–678, 2006.