
Preprint:
Moyasari, A., Beohar, H., Grellois, C. et al. (Submitted: 2026) Tree automata acceptance up to measurable defect. [Preprint - arXiv] (Submitted)

https://doi.org/10.48550/arxiv.2605.27192

Tree Automata Acceptance up to Measurable Defect

Anita Moyasari[1], Harsh Beohar[1], Charles Grellois[1], and Clemens Kupke[2]

[1]School of Computer Science, University of Sheffield, United Kingdom
[2]University of Strathclyde, Glasgow, United Kingdom

Abstract

Automata acceptance can, in several situations of interest, be captured game-theoretically via *acceptance games*. The existence of a winning strategy for Verifier then captures the existence of a winning run-tree of a given automaton over a model. However, such acceptance is rigid, in that it does not allow a measurable defect budget, which can be a challenge in software verification. In this paper, we draw inspiration from how bisimulation distance can be defined as an extension of bisimilarity to define ϵ-acceptance games. Our main theorem shows that a tree $\mathcal{T}$ is ε-accepted iff there is a tree $\mathcal{T}'$ that is accepted in the traditional (rigid) sense and the bisimulation distance of $\mathcal{T}'$ and $\mathcal{T}$ is at most ε. Our work also suggests a strong connection with measure theory, of which we give a preliminary exploration via appropriate examples. Our framework is defined over binary trees with leaves and infinite branches, and strictly contains the case in which binary nodes are seen as probabilistic choice and the defect measures the probability of the set of rejected branches.

2012 ACM Subject Classification: Theory of computation → Automata extensions; Theory of computation → Verification by model checking
Keywords: automata theory, acceptance games, bisimulation distance

1 Introduction

Model checking is a successful approach to verification, in which a program's behaviour is modelled by a mathematical object $\mathcal{M}$ and the specification of the program is expressed as a logical property φ. In many situations of interest (such as modal μ-calculus verification over trees, see for instance [1]), the model-checking problem $\mathcal{M} \vDash \varphi$ can be equivalently formulated by a two-player game played over the states of an automaton $\mathcal{A}_\varphi$ and the model $\mathcal{M}$, where Verifier is trying to assert that the property φ is valid over $\mathcal{M}$, while Falsifier is dually trying to establish that φ is invalid over $\mathcal{M}$.

Such games traditionally model a Boolean, *rigid* notion of acceptance. In this paper, we build on an insight from process algebra [2], in which the notion of *bisimulation* [3] is considered a key equivalence on process terms. It is also Boolean in nature since two pointed labelled transition systems are either bisimilar, or they are not; but this notion has been lifted to the notion of a *bisimulation distance* which is well studied in quantitative settings like probabilistic generalisations of labelled transition systems [4, 5].

In this paper, we extend the traditional, rigid game-theoretic automata acceptance games to a quantitative framework in which a defect budget is allowed. This defect budget provides an upper bound over some measure of the obstruction to acceptance – that we make precise

later in the paper. This turns the game into a quantitative one, in which Verifier plays moves that estimate upper bounds of the defect that would be faced later in a play, subject to thresholds coming from a global constraint. To ease reading, we take advantage of the well-known left-child, right-sibling encoding (cf. [6]) and develop our approach on binary trees with leaves and infinite branches. The approach relies on a specification of how the defect budget is split in between both successors of an inner node; this notably covers cases in which the defect computes the probabilistic measure of the set of accepting branches, but the approach is more general as it is not restricted to linear combinations of errors.

Interestingly enough, our approach builds much more than a mere quantitative extension of automata acceptance games via the introduction of defects. Our main theorem (cf. Theorem 4.1) proves, under the restriction that the automaton is deadlock free, that when the defect is split between the successors of an inner (thus binary) node in a way that, in the coalgebraic world, is called a *distance lifting*, then:

$$\text{acceptance of a tree } \mathcal{T} \text{ with defect at most } \varepsilon$$

is equivalent to the

$$\text{existence of } \mathcal{T}', \text{ accepted in the traditional sense and at distance at most } \varepsilon \text{ of } \mathcal{T}$$

where this distance – *bisimulation distance* – can itself be captured by game-theoretic means.

We also explore the natural connection with measure theory via targeted examples, leaving the systematic treatment of the connection between distance liftings for certain classes of distance functions and measure theory for future work.

Overall, our work introduces a quantitative approach to verification, in which defect is admissible within certain thresholds, with key notions from the coalgebraic community. We present our results in a way that will make them amenable to a future coalgebraic treatment, covering in that way more data structures than trees, yet in a similar manner.

Related Works **Quantitative Languages.** In [7], Chatterjee, Doyen and Henzinger introduce quantitative languages of words. Such languages replace the Boolean value of whether a word w belongs to L by a real number; weighted automata are then used, and their runs output quantitative information. By contrast, our work does not amend trees or automata: by allowing a defect budget in the game, we capture the measure of an obstruction to acceptance; our main theorem that this defect coincides with bisimulation distance.

Probabilistic Model Checking. In probabilistic model-checking (see for instance [8, 9]), a core question is to compute the probability with which a probabilistic model such as a Markov chain satisfies a given logical specification. Our approach enables the verification of Markovian processes – unfolded to a tree form, and for which the distance lifting would be chosen so as to encode the probability distributions. But our approach can also accommodate measure defects that do not come from (the encoding of) probabilistic systems, and is actually structurally rooted in the measure of a defect to traditional acceptance rather than in the extension of automata and models with stochastic information.

Behavioural Metrics. Bisimilarity was extended to bisimulation distance in the context of Markov processes, see [4, 10, 5], and more recently to coalgebras [11]. In the theory of coalgebras, there are two prevalent ways to define distance liftings: one generalises the Kantorovich lifting of distributions known as the codensity lifting [12] of an endofunctor; the other [13] generalises the Wasserstein lifting of distributions known as the coupling-based lifting of an endofunctor in [14]. In general, the two approaches to define lifting are not

equivalent; however, for polynomial endofunctors (like the functor $F = \Sigma_0 \uplus \Sigma_2 \times \_ \times \_$ whose coalgebras are labelled binary trees) over sets it is known to be the same [14].

Our paper is part of our wider program on ϵ-acceptance for data structures presented as coalgebras, extending coalgebraic automata theory [15] and generalising the associated game semantics to a more quantitative setting. In the current paper we limit ourselves to binary trees and do not prove our main theorem (Theorem 4.1) in its full generality (i.e. at the level of coalgebras), which is left for future.

Organization We begin in Section 2 by introducing the necessary preliminaries regarding trees, tree automata, and acceptance games. In Section 3, we generalize these definitions to a quantitative framework and introduce the bisimulation distance game. Within this setting, we extend the classical notion of an acceptance game to that of ϵ-acceptance, incorporating within the traditional acceptance game information about a tolerable defect budget. Section 4 then establishes the formal connection between the quantitative and Boolean versions of these games. Finally, in Section 5, we discuss two applications, specifically focusing on the relationship between these games and measures on binary trees.

2 Preliminaries

The objective of this section is to recall the notion of acceptance games in the context of labelled binary trees from [15]. As mentioned in the introduction, and for the sake of readability, we restrict our attention to binary trees with leaves and infinite branches, taking advantage of the left-child, right-sibling encoding (cf. [6]). Note that all the results of this paper can however be restated for ranked trees over a finite signature containing labels of finite arities other than 0 or 2.

2.1 Labelled Binary Trees

Definition 2.1. A *(binary) tree* is a subset $\mathcal{U}$ of 2^* that is

1. prefix-closed: $\forall w, w' \in 2^* \quad (w \in \mathcal{U} \wedge w' \leq w) \implies w' \in \mathcal{U}$, and
2. sibling-closed: $\forall w \in \mathcal{U} \quad w.0 \in \mathcal{U} \iff w.1 \in \mathcal{U}$.

A *labelled tree*, denoted $\mathcal{T} = (\mathcal{U}, \ell)$, consists of a tree $\mathcal{U}$, a set of labels $\Sigma = \Sigma_0 \uplus \Sigma_2$, and a labelling function $\ell \colon \mathcal{U} \to \Sigma$ satisfying $\forall w \in \mathcal{U} \quad \ell(w) \in \Sigma_2 \iff w.\{0,1\} \in \mathcal{U}$.

Definition 2.2. For a tree $\mathcal{T} = (\mathcal{U}, \ell)$ let $\mathcal{T}_w = (\mathcal{U}_w, \ell_w)$ be a *subtree* of $\mathcal{T}$ rooted at w defined as $\mathcal{U}_w = w^{-1}\mathcal{U} = \{v | wv \in \mathcal{U}\}$ and $\ell_w(v) = \ell(wv)$.

The key idea of our approach is that the behaviour of a tree can be studied using local information at each branching point, rather than requiring global information at once. This is closer in spirit to the coalgebraic presentation of trees (see, for instance, [16]). One advantage of coalgebraic modelling is that it allows a unified treatment of defining conformances (like bisimilarity or bisimulation distance); for instance, we will use one of these methods (cf. Theorem 3.2) to define bisimulation distance over trees in Section 3. This naturally leads to the introduction of a successor function.

We first observe that the successor of a node is either in Σ_0, in the case of a leaf, or in $\Sigma_2 \times \mathcal{U} \times \mathcal{U}$ in the case of a branching node. To formalise this distinction, we introduce $F\mathcal{U} = \Sigma_0 \uplus (\Sigma_2 \times \mathcal{U} \times \mathcal{U})$ specifying the possible types of successors.

Definition 2.3. Let $\mathcal{T} = (\mathcal{U}, \ell)$ be a labelled tree and let $\mathrm{Ar} : \mathcal{U} \to \{0, 2\}$ be the *arity* function denoting the arity of a node in $\mathcal{U}$ (i.e., $\mathrm{Ar}(w) = 0$ if $\ell(w) \in \Sigma_0$; $\mathrm{Ar}(w) = 2$ otherwise). For a labelled tree $\mathcal{T} = (\mathcal{U}, \ell)$, its *successor function* is a map $\gamma : \mathcal{U} \to F\mathcal{U}$ defined as follows:

$$\gamma(w) = \begin{cases} \ell(w) & \text{if } \mathrm{Ar}(w) = 0 \\ \big(\ell(w), w.0, w.1\big) & \text{if } \mathrm{Ar}(w) = 2 \end{cases}$$

Henceforth we write $\mathcal{T}$ (and similarly $\mathcal{T}'$) to denote a labelled binary tree $(\mathcal{U}, \ell)$ equipped with a successor function γ (respectively $(\mathcal{U}', \ell')$ with γ').

Remark 2.4. Although we present our definitions for trees, all notions introduced in this paper extend naturally to any similar structure consisting of a carrier set X together with a successor function $\gamma\colon X \to FX$. In several places, we will apply definitions given for trees to other structures of the same type, such as automata with carrier set A and transition function $\delta\colon A \to FA$, without restating them explicitly. This slight abuse of notation should cause no confusion, as the arguments remain unchanged.

Readers familiar with coalgebraic terminology will recognise that all these structures are instances of F-coalgebras for a polynomial endofunctor F over the category of sets. In particular, our labelled binary trees are coalgebras for the functor $F\_ = \Sigma_0 \uplus (\Sigma_2 \times \_ \times \_)$.

We end this subsection with the following well-known result on substituting $\mathcal{T}'$ inside $\mathcal{T}$ at node $w \in \mathcal{U}$.

Proposition 2.5. *Let $\mathcal{T}$ and $\mathcal{T}'$ be binary trees. For any node $w \in \mathcal{U}$, by substituting the subtree $\mathcal{T}_w$ with $\mathcal{T}'$ we get a new tree defined as $\mathcal{T}[w/\mathcal{T}'] = (\overline{\mathcal{U}}, \overline{\ell}, \overline{\gamma})$ where $\overline{\mathcal{U}} = \big(\mathcal{U} \setminus \{v \in \mathcal{U} \mid w \leq v\}\big) \cup \{\, w.v \mid v \in \mathcal{U}' \,\}$, and the successor function $\overline{\gamma}$ is given by*

$$\overline{\gamma}(w) = \begin{cases} \gamma(u) & \text{if } u \in \mathcal{U} \setminus \{v \mid w \leq v\}, \\ \sigma & \text{if } u = w.v \in \{\, w.v \mid v \in \mathcal{U}' \,\},\ \gamma'(v) = \sigma \in \Sigma_0 \\ (\sigma, u.0, u.1) & \text{if } u = w.v \in \{\, w.v \mid v \in \mathcal{U}' \,\},\ \gamma'(v) = (\sigma, v.0, v.1) \text{ for } \sigma \in \Sigma_2 \end{cases}$$

2.2 Automata Acceptance Game

When viewing programs as trees, a common way of analyzing if they satisfy a property is a two-player game characterising some form of bisimulation. In order to define this game, we first need to "lift" a relation between nodes of two tree structures to a relation between their successors, as described in the following definition. We then introduce the acceptance game.

Definition 2.6. Let $\mathcal{T}$, $\mathcal{T}'$ be two trees over the domains $\mathcal{U}, \mathcal{U}'$, respectively. For a relation $R \subseteq \mathcal{U} \times \mathcal{U}'$, we define the *lifted relation* $\bar{R} \subseteq F\mathcal{U} \times F\mathcal{U}'$ as follows:

$$\alpha\ \bar{R}\ \beta \iff \alpha = \beta \in \Sigma_0 \text{ or } \begin{cases} \alpha = (\sigma, w_1, w_2)\ ,\ \beta = (\sigma, w_1', w_2') & \text{for } \sigma \in \Sigma_2, \\ \text{and } \ w_1\ R\ w_1'\ , \text{ and } \ w_2\ R\ w_2'. \end{cases}$$

Definition 2.7. A relation $R \subseteq \mathcal{U} \times \mathcal{U}'$ is called a *bisimulation* between $\mathcal{T}$ and $\mathcal{T}'$ iff $R \subseteq (\gamma \times \gamma')^{-1}(\bar{R})$. In other words, for every $w \in \mathcal{U}$, $w' \in \mathcal{U}'$ we have

$$w\ R\ w' \implies \Big(\ell(w) = \ell'(w') \wedge \big(\mathrm{Ar}(w) = 2 \implies w.0\ R\ w'.0 \wedge w.1\ R\ w'.1\big)\Big).$$

Two subtrees $\mathcal{T}_w$, $\mathcal{T}'_{w'}$ are *bisimilar*, denoted $\mathcal{T}_w \leftrightarrows \mathcal{T}'_{w'}$, iff there is a bisimulation relation R that includes $(w, w') \in R$. In particular, $\mathcal{T} \leftrightarrows \mathcal{T}'$ iff there is a bisimulation relation R that includes $(\mathrm{root}_{\mathcal{T}}, \mathrm{root}_{\mathcal{T}'}) \in R$.

Definition 2.8. A nondeterministic tree automaton $\mathbb{A}$ is a quadruple (A, a_0, Δ, Acc), where A is a finite set of states, $a_0 \in A$ is the initial state, $\Delta : A \to \mathcal{P}FA$ is the transition function, and $Acc \subseteq A^\omega$ is the infinite acceptance condition.

Remark 2.9. A more general acceptance game appears in [15], for an alternating automata with $\Delta\colon A \to \mathcal{P}\mathcal{P}FA$. Our game can be seen as an instance of this general game, in which the choice of Falsifier is always unique, as we do not model alternation. However, as shown in [15, Thm. 5.2], for weak pullback preserving endofunctors F over sets, alternating and non-deterministic automata have the same expressive power. The functor for binary trees satisfies this condition. Hence, in our case alternating F-automata can be simulated by non-deterministic ones over the same functor – this is a coalgebraic generalisation of the well-known Rabin's theorem on the complementation of tree automata [17].

We will now introduce the automata acceptance game, not in its full generality but for the case where all infinite runs/plays are accepted and thus won by the Verifier. As the acceptance condition $Acc = A^\omega$ we represent automata by a triple.

Definition 2.10. Let $\mathbb{A} = (A, a_0, \Delta)$ be a tree automaton and let $\mathcal{T}$ be a labelled tree. The acceptance game associated with $\mathbb{A}$ and $\mathcal{T}_w$ is a two player game defined as follows:
Each round of a game starts with a basic position (a, w) where $a \in A$, $w \in \mathcal{U}$, then

1. At position (a, w) Verifier (often denoted by $\exists$ in the sequel) can choose the next state $\delta(a) \in FA$ of the automaton from the possible choices in $\Delta(a)$.
2. Then $\exists$ picks a relation $R \subseteq A \times \mathcal{U}$ such that $(\delta(a), \gamma(w)) \in \bar{R}$.
3. At position R, Falsifier (denoted $\forall$) can choose any pair (b, v) that belongs to R.

The game then advances to the next round with basic position (b, v).
If a player cannot play then they lose, and over an infinite play $\exists$ wins. We say that $\mathbb{A}$ accepts $\mathcal{T}_w$ when (a_0, w) is a winning position for $\exists$. The set of winning positions for Verifier is denoted as $\mathsf{AWin}_\exists$.
Henceforth we will present games in tabular form, providing all the relevant information.

Position	Player	Admissible moves
$(a, w) \in A \times \mathcal{U}$	$\exists$	$\{(\delta(a), w) \in FA \times \mathcal{U} \mid \delta(a) \in \Delta(a)\}$
$(\delta(a), w) \in FA \times \mathcal{U}$	$\exists$	$R \in \mathcal{P}(A \times \mathcal{U}) \mid (\delta(a), \gamma(w)) \in \bar{R}\}$
$R \in \mathcal{P}(A \times \mathcal{U})$	$\forall$	$\{(b, v) \mid (b, v) \in R\}$

Table 1: Acceptance game for a tree automaton and a binary tree [15].

Remark 2.11. The above game is the general coalgebraic form of the acceptance game. This can be simplified to its standard form by observing that if $\exists$ has a winning move R, then its restriction to immediate successor pairs is also winning, since the validity of R depends only on successors. Concretely, any such R is either $\varnothing$ (when $\ell(a) = \ell(w) \in \Sigma_0$) or $\{(b_0, w.0), (b_1, w.1)\}$ (when $\ell(a) = \ell(w) \in \Sigma_2$, $\delta(a) = (\sigma, b_0, b_1)$, and $\gamma(w) = (\sigma, w.0, w.1)$). Consequently, Falsifier's choices may be restricted to these successors without loss of generality. Formally, a position is winning in the original game iff it is winning in this restricted version.

3 Quantitative Games for Bisimulation Distance and ϵ-acceptance

3.1 Distance Lifting

In order to obtain quantitative versions of (bisimulation/acceptance) games and their associated results, we need a 'general' notion of distance lifting (instead of relation lifting). A relation can be seen as a function taking values in $\{0,1\}$; this perspective naturally extends to functions to the unit interval $[0,1]$. In this setting, we call such functions distances and introduce a suitable notion of distance lifting.

Definition 3.1. Let $\mathcal{T}$ and $\mathcal{T}'$ be two trees of domains $\mathcal{U}$ and $\mathcal{U}'$, respectively. Then every order preserving function $\mathsf{f}\colon [0,1]\times[0,1]\to[0,1]$ gives rise to a distance lifting $\bar{d}\colon F\mathcal{U}\times F\mathcal{U}'\to[0,1]$ for a given $d\colon \mathcal{U}\times\mathcal{U}'\to[0,1]$ as follows:

$$\bar{d}(\alpha,\beta)=\begin{cases}0 & \text{if } \alpha=\beta\in\Sigma_0\\ \mathsf{f}\big(d(w_1,w_1')\,,\,d(w_2,w_2')\big) & \text{if } \alpha=(\sigma,w_1,w_2)\ ,\ \beta=(\sigma,w_1',w_2') \text{ for } \sigma\in\Sigma_2\ .\\ 1 & \text{otherwise}\end{cases}$$

Remark 3.2. Perhaps, for category theory minded readers, it is useful to note that the above distance lifting is an instance of coupling based lifting [13] of a set endofunctor $F\colon \mathbf{Set}\to\mathbf{Set}$. The following construction is similar to [13, Equation 5], but our $\mathcal{V}$-valued relations (where $\mathcal{V}$ is the Lawvere quantale $([0,1],\geq)$) are over two different sets (unlike in op. cit. where they are restricted to binary relations over the same set).

Now the function f gives rise to an evaluation function $\mathsf{ev}_\mathsf{f}\colon F[0,1]\to[0,1]$ as follows: $\sigma\in\Sigma_0\mapsto 0$ (for each $\sigma\in\Sigma_0$) and $(\sigma,r,r')\mapsto\mathsf{f}(r,r')$ (for each $(\sigma,r,r')\in\Sigma_2\times[0,1]\times[0,1]$). Notice that the evaluation function ev_f is also monotone whenever f is, soon as we enriched $F[0,1]$ by lifting the order $\geq$ on $[0,1]$ and letting Σ_0,Σ_2 to be discretely ordered. As a result, we have a following composition of monotone maps:

$$([0,1]^{\mathcal{U}\times\mathcal{U}'},\geq)\xrightarrow{\mathsf{ev}_\mathsf{f}\circ F(\_)}([0,1]^{F(\mathcal{U}\times\mathcal{U}')},\geq)\xrightarrow{(\lambda_{U,U'})_!}([0,1]^{F(\mathcal{U})\times F(\mathcal{U}')},\geq),\tag{1}$$

where $\lambda_{U,U'}$ is the tupling of projections $\langle F\mathrm{proj}_\mathcal{U},F\mathrm{proj}'_\mathcal{U}\rangle\colon F(\mathcal{U}\times\mathcal{U}')\to F(\mathcal{U})\times F(\mathcal{U}')$ and $f_!\colon[0,1]^X\to[0,1]^Y$ (for a function $f\colon X\to Y$) is the left adjoint to reindexing map f^* (see [13] for more details) given by $f_!(p)(y)=\inf_{f(x)=y}p(x)$. In this categorical setting, we obtain the following result (which is not required to understand the rest of this paper, but aimed at the reader with a coalgebraic appetite):

Lemma 3.3. *Let $d\colon\mathcal{U}\times\mathcal{U}'\to[0,1]$. Then, distance lifting of d is equivalent to the application of the composition given in* (1) *on d; i.e., $\bar{d}=(\lambda_{\mathcal{U},\mathcal{U}'})_!(\mathsf{ev}_\mathsf{f}\circ F(d))$.*

Proof. Let $\alpha\in F(\mathcal{U}),\beta\in F(\mathcal{U}')$. Then using (1) we have

$$(\lambda_{\mathcal{U},\mathcal{U}'})_!(\mathsf{ev}_\mathsf{f}\circ F(d))(\alpha,\beta)=\inf_{r\in\Gamma(\alpha,\beta)}\mathsf{ev}_\mathsf{f}\circ F(d)(r),$$

where $\Gamma(\alpha,\beta)=\{r\in F(\mathcal{U}\times\mathcal{U}')\mid F(\mathrm{proj}_\mathcal{U})(r)=\alpha\wedge F(\mathrm{proj}_{\mathcal{U}'})(r)=\beta\}$. Then we identify the following cases based on the type of $\alpha\in F(\mathcal{U}),\beta\in F(\mathcal{U}')$:

1. Let $\alpha,\beta\in\Sigma_0$. Then $r\in\Gamma(\alpha,\beta)\iff(\alpha=r\wedge r=\beta)$. Then we distinguish two cases:

   (a) Let $\alpha=\beta$. Then, using the definition of ev_f we find

   $$(\lambda_{\mathcal{U},\mathcal{U}'})_!(\mathsf{ev}_\mathsf{f}\circ F(d))(\alpha,\beta)=\mathsf{ev}_\mathsf{f}\circ F(d)(\alpha)=0=\bar{d}(\alpha,\beta)$$

(b) Let $\alpha \neq \beta$. Then, $\Gamma(\alpha,\beta) = \emptyset$ and using empty infima corresponds to 1 we find that

$$(\lambda_{\mathcal{U},\mathcal{U}'})_!(\mathsf{ev}_\mathsf{f} \circ F(d))(\alpha,\beta) = 1 = \bar{d}(\alpha,\beta).$$

2. Let $\alpha \in \Sigma_0$ and $\beta \in \Sigma_2 \times \mathcal{U}' \times \mathcal{U}'$. Then $\Gamma(\alpha,\beta) = \emptyset$ and the remainder is similar to Item 1b.

3. Let $\alpha \in \Sigma_2 \times \mathcal{U} \times \mathcal{U}$ and $\beta \in \Sigma_0$. Similar to the above case.

4. Let $\alpha \in \Sigma_2 \times \mathcal{U} \times \mathcal{U}$ and $\beta \in \Sigma_2 \times \mathcal{U}' \times \mathcal{U}'$. Then we further distinguish two cases (below proj_{Σ_2} is the mapping that project elements from Σ_2):

   (a) Let $\mathrm{proj}_{\Sigma_2}(\alpha) = \mathrm{proj}_{\Sigma_2}(\beta)$. I.e., there are $\sigma \in \Sigma_2$, $w_1, w_2 \in \mathcal{U}$, and $w_1', w_2' \in \mathcal{U}'$ such that $\alpha = (\sigma, w_1, w_2)$ and $\beta = (\sigma, w_1', w_2')$. Then, there is a unique $r \in \Gamma(\alpha,\beta)$ and is given by $r = (\sigma, (w_1, w_1'), (w_2, w_2')) \in F(\mathcal{U} \times \mathcal{U}')$. Using the definition of ev_f we get:

$$(\lambda_{\mathcal{U},\mathcal{U}'})_!(\mathsf{ev}_\mathsf{f} \circ F(d))(\alpha,\beta) = \mathsf{ev}_\mathsf{f} \circ F(d)(r) = \mathsf{f}(d(w_1,w_1'), d(w_2,w_2')) = \bar{d}(\alpha,\beta).$$

   (b) Let $\mathrm{proj}_{\Sigma_2}(\alpha) \neq \mathrm{proj}_{\Sigma_2}(\beta)$. Then $\gamma(\alpha,\beta) = \emptyset$ and the remainder is similar to Item 1b.

□

Definition 3.4. A distance function $d : \mathcal{U} \times \mathcal{U}' \to [0,1]$ is a *bisimulation distance function* iff $\forall (w,w') \in \mathcal{U} \times \mathcal{U}' \quad d(w,w') \geq \bar{d}(\gamma(w), \gamma'(w'))$, where

$$\bar{d}(\gamma(w),\gamma'(w')) = \begin{cases} 1 & \text{if } \ell(w) \neq \ell'(w') \\ 0 & \text{if } \ell(w) = \ell'(w') \wedge \mathrm{Ar}(w) = 0 \\ \mathsf{f}\big(d(w.0,w'.0)\,,\, d(w.1,w'.1)\big) & \text{otherwise} \end{cases}$$

Just like traditional bisimilarity $\leftrightarrows$ is the largest bisimulation relation, we can now define the bisimulation distance bd as the greatest post fixpoint of the following functional:

$$([0,1]^{\mathcal{U}\times\mathcal{U}'}, \geq) \;\bar{\to}\; ([0,1]^{F(\mathcal{U})\times F(\mathcal{U}')}, \geq) \xrightarrow{(\gamma\times\gamma')\circ} ([0,1]^{\mathcal{U}\times\mathcal{U}'}, \geq).$$

Note that the well known Knaster-Tarski fixpoint theorem guarantees the existence of bd. We say that the *bisimulation distance* of $\mathcal{T}_w$ and $\mathcal{T}'_{w'}$ is ε iff $\mathsf{bd}(\mathcal{T}_w, \mathcal{T}'_{w'}) = \varepsilon$. In particular, the bisimulation distance of $\mathcal{T}$ and $\mathcal{T}'$ is ε iff $\mathsf{bd}(\mathrm{root},\mathrm{root}) = \varepsilon$.

3.2 Bisimulation Distance, Game-Theoretically

Bisimulation distance enables one to analyse how far apart the behaviours of two trees sit. This can be formalised by introducing a two-player game (cf. Theorem 3.5) in which Verifier aims to establish that the distance between the trees is at most ε, while Falsifier attempts to decrease ε to a point where Verifier can no longer maintain a winning strategy.

Definition 3.5. Let $\mathcal{T}$ and $\mathcal{T}'$ be two trees over the domains $\mathcal{U}$ and $\mathcal{U}'$, respectively. The *bisimulation distance game* between $\mathcal{T}$ and $\mathcal{T}'$ is a two player game defined as in Table 2.

If a player cannot play then they lose, and over an infinite play $\exists$ wins the game. If starting from a position (w,w',ε) Verifier wins the game we say that (w,w',ε) is a winning position for $\exists$ and we denote the set of such positions as $\mathsf{Win}^{\mathsf{bd}}_\exists$. Furthermore, if a triple (w,w',ε) belongs to $\mathsf{Win}^{\mathsf{bd}}_\exists$, we say that ε is a winning distance for w,w'.

Position	Player	Admissible moves
$(w, w', \varepsilon_1) \in \mathcal{U} \times \mathcal{U}' \times [0,1]$	$\exists$	$\{d : \mathcal{U} \times \mathcal{U}' \to [0,1] \mid \bar{d}(\gamma(w), \gamma'(w')) \leq \varepsilon_1 \}$
$d \in \mathcal{U} \times \mathcal{U}' \to [0,1]$	$\forall$	$\{(v, v', \varepsilon_2) \in \mathcal{U} \times \mathcal{U}' \times [0,1] \mid d(v, v') \leq \varepsilon_2\}$

Table 2: The bisimulation distance game for two binary trees $\mathcal{T}$ and $\mathcal{T}'$

Proposition 3.6. *The bisimulation distance game has the following properties:*

1. *For all $w \in \mathcal{U}$ and $w' \in \mathcal{U}'$, the triple $(w, w', 1) \in \mathsf{Win}_{\exists}^{\mathsf{bd}}$.*
2. *If $(w, w', \varepsilon_1) \in \mathsf{Win}_{\exists}^{\mathsf{bd}}$ and $\varepsilon_2 \geq \varepsilon_1$, then $(w, w', \varepsilon_2) \in \mathsf{Win}_{\exists}^{\mathsf{bd}}$.*

Proof. For each part we present a winning strategy for Verifier i.e a valid move for each of their turns:

1. At each round, $\exists$ can choose the constant distance function $d = 1$. Since the starting position is $(w, w', 1)$ this d satisfies the property $\bar{d}(\gamma(w), \gamma'(w')) \leq 1$ and the game moves, by the choice of $\forall$ to the next position $(v, v', 1)$ and the same argument applies.

2. Assume $(w, w', \varepsilon_1) \in \mathsf{Win}_{\exists}^{\mathsf{bd}}$. Then $\exists$ has a winning strategy from this position, in particular a first move given by some distance d such that

$$\bar{d}(\gamma(w), \gamma'(w')) \leq \varepsilon_1 \leq \varepsilon_2$$

   So, d is also a valid move from (w, w', ε_2). Moreover, since d is part of a winning strategy, every response of $\forall$ leads to a position in $\mathsf{Win}_{\exists}^{\mathsf{bd}}$. This means, the same strategy is winning from (w, w', ε_2), and therefore $(w, w', \varepsilon_2) \in \mathsf{Win}_{\exists}^{\mathsf{bd}}$.

□

Theorem 3.7. *For a bisimulation distance game associated with $\mathcal{T}$ and $\mathcal{T}'$ we have*

$$\forall_{w \in \mathcal{U}, w' \in \mathcal{U}', \varepsilon \in [0,1]} \quad (w, w', \varepsilon) \in \mathsf{Win}_{\exists}^{\mathsf{bd}} \iff \mathsf{bd}(\mathcal{T}_w, \mathcal{T}'_{w'}) \leq \varepsilon.$$

Proof. $\boxed{\Leftarrow}$ Assume that $\mathsf{bd}(\mathcal{T}_w, \mathcal{T}_{w'}{}') \leq \varepsilon$, hence $d^*(w, w') \leq \varepsilon$. We describe a winning strategy for Verifier: at every round, $\exists$ choose the distance d^*.

At the initial position (w, w', ε), this is a valid move since $\varepsilon \geq d^*(w, w') \geq \bar{d}(\gamma(w), \gamma'(w'))$. After any move of Falsifier, the game moves to a basic position $(w_1, w_1', \varepsilon_1)$ with $d^*(w_1, w_1') \leq \varepsilon_1$, and the same argument applies. At every position (v, v', κ) reached during the play, the invariant $d^*(v, v') \leq \kappa$ is preserved. Hence, $\exists$ can continue playing d^* indefinitely which means $\exists$ has a winning strategy starting from (w, w', ε). So $(w, w', \varepsilon) \in \mathsf{Win}_{\exists}^{\mathsf{bd}}$.

$\boxed{\Rightarrow}$ Assume $(w, w', \varepsilon) \in \mathsf{Win}_{\exists}^{\mathsf{bd}}$. We first construct a bisimulation distance d such that $d(w, w') \leq \varepsilon$. Define

$$d(v, v') = \inf\{\kappa \mid (v, v', \kappa) \in \mathsf{Win}_{\exists}^{\mathsf{bd}}\} \quad \text{for all } v \in \mathcal{U},\ v' \in \mathcal{U}'.$$

We show that d is a bisimulation distance, i.e.

$$d(v, v') \geq \bar{d}(\gamma(v), \gamma'(v')) \quad \text{for all } v \in \mathcal{U},\ v' \in \mathcal{U}'.$$

Fix v, v' and let $d(v, v') = \mu$. If $\mu = 1$, the inequality is trivial. Suppose $\mu < 1$. By definition of d and Theorem 3.6(Item 2), for every $\kappa > \mu$ we have $(v, v', \kappa) \in \mathsf{Win}_{\exists}^{\mathsf{bd}}$. Hence $\exists$ has a

winning strategy from (v, v', κ), by playing a distance d_κ such that $\bar{d_\kappa}(\gamma(v), \gamma'(v')) \leq \kappa$, and any valid choice of $\forall$, in particular all triples $(u, u', d_\kappa(u, u'))$, belong to $\mathsf{Win}^{\mathsf{bd}}_{\exists}$.
By definition of d, it follows that $d(u, u') \leq d_\kappa(u, u')$ for all u, u', hence $d \succeq d_\kappa$. So by monotonicity of the lifting,

$$\bar{d}(\gamma(v), \gamma'(v')) \leq \bar{d_\kappa}(\gamma(v), \gamma'(v')) \leq \kappa.$$

Since this holds for all $\kappa > \mu$, we obtain

$$\bar{d}(\gamma(v), \gamma'(v')) \leq \mu = d(v, v').$$

Therefore, d is a bisimulation distance.

Finally, since d^* is the best bisimulation distance, we have $d^*(w, w') \leq d(w, w')$. Moreover, $d(w, w') \leq \varepsilon$ by definition of d and the assumption $(w, w', \varepsilon) \in \mathsf{Win}^{\mathsf{bd}}_{\exists}$. Hence

$$\mathsf{bd}(\mathcal{T}_w, \mathcal{T}_{w'}) = d^*(w, w') \leq d(w, w') \leq \varepsilon,$$

which concludes the proof. □

Corollary 3.8. *The set of winning distances for any $w \in \mathcal{U}$ and $w' \in \mathcal{U}'$ is of the form of a closed interval $[\varepsilon, 1]$ where $\varepsilon = \mathsf{bd}(\mathcal{T}_w, \mathcal{T}_{w'})$.*

Consequently, the set of winning distances is closed, and the infimum is attained. This establishes the existence of a minimal winning distance ε which coincides with the bisimulation distance between the corresponding subtrees.

Furthermore, the bisimulation distance can be characterised in terms of winning values of ε: either the optimal value is obtained, showing the exact distance from a property, or we have an upper bound given by a winning ε. Examples of such optimal winning values and their interpretation are discussed in Section 5.

3.3 The ϵ-acceptance Game

Definition 3.9. Let $\mathbb{A} = (A, a_0, \Delta)$ be a tree automaton and $\mathcal{T}$ a binary tree over a domain $\mathcal{U}$ with successor function $\gamma \colon \mathcal{U} \to F(\mathcal{U})$. The acceptance game associated with $\mathbb{A}$ and $\mathcal{T}_w$ is a two player game where the moves of each player are shown in Table 3.

Position	Player	Admissible moves
(a, w, ε_1)	$\exists$	$\{(\delta(a), w, \varepsilon_1) \in F(A) \times \mathcal{U} \times [0,1] \mid \delta(a) \in \Delta(a)\}$
$(\delta(a), w, \varepsilon_1) \in F(A) \times \mathcal{U} \times [0,1]$	$\exists$	$\{d : A \times \mathcal{U} \to [0,1] \mid \bar{d}(\delta(a), \gamma(w)) \leq \varepsilon_1\}$
$d \in A \times \mathcal{U} \to [0,1]$	$\forall$	$\{(b, v, \varepsilon_2) \in A \times \mathcal{U} \times [0,1] \mid d(b, v) \leq \varepsilon_2\}$

Table 3: ϵ-acceptance game for a tree automaton $\mathbb{A}$ and a binary tree $\mathcal{T}$

If a player cannot play they lose, and for any infinite play $\exists$ wins. We denote by $\mathsf{AWin}^{\epsilon}_{\exists}$ the set of winning positions for Verifier and $\mathcal{T}_w$ is said to be ε-accepted by $\mathbb{A}$ when $(a_0, w, \varepsilon) \in \mathsf{AWin}^{\epsilon}_{\exists}$.

Remark 3.10. By a similar argument to the bisimulation distance game, if $(a, w, \varepsilon_1) \in \mathsf{AWin}^{\epsilon}_{\exists}$ and $\varepsilon_1 \leq \varepsilon_2$, then $(a, w, \varepsilon_2) \in \mathsf{AWin}^{\epsilon}_{\exists}$. Hence, without loss of generality, we may assume that choices of $\forall$ are of the form $(b, v, d(b, v))$, where d is the function previously chosen by $\exists$.

4 ϵ-Acceptance Is Acceptance Plus Bisimulation Distance

We are now ready to state and prove the main theorem of the paper, establishing the connection between acceptance and quantitative acceptance.

Throughout this section, we work with a tree automaton $\mathbb{A} = (A, a_0, \Delta)$ where every infinite word is accepted and $\Delta(a) \neq \emptyset$ for every $a \in A$ (i.e. the automaton is deadlock free).

Theorem 4.1. *Let $\mathbb{A} = (A, a_0, \Delta)$ be a deadlock free tree automaton and let $\mathcal{T}$ be a labelled tree. Then the following statements are equivalent:*

1. *$\mathcal{T}$ is ε_0-accepted by $\mathbb{A}$.*
2. *There is a binary tree $\mathcal{T}'$ such that $\mathsf{bd}(\mathcal{T}', \mathcal{T}) \leq \varepsilon_0$, and $\mathcal{T}'$ is accepted by $\mathbb{A}$.*

We prove the two directions of the theorem separately; the proof of the direction $2 \implies 1$ is immediate once we have the following lemma which states that the positions of the form $(a, w, 1)$ are always winning for Verifier in the ϵ-acceptance game.

Lemma 4.2. *For a deadlock free automaton $\mathbb{A}$ and a tree $\mathcal{T}$, we have $(a, w, 1) \in \mathsf{AWin}^{\epsilon}_{\exists}$ (for every $a \in A, w \in \mathcal{U}$).*

Proof. At position $(a, w, 1)$, $\exists$ can choose any $\delta(a) \in \Delta(a)$ since $\Delta(a) \neq \emptyset$ and distance function d is the constant 1 function. Then any choice of $\forall$ will be of the form $(b, v, 1)$ that will be the new position of the game and the same argument applies. Therefore $\exists$ always has a move, and the play will be infinite. So $(a, w, 1) \in \mathsf{AWin}^{\epsilon}_{\exists}$. □

Proof of Theorem 4.1 ($2 \implies 1$). Assume the existence of such $\mathcal{T}'$ as given in Item 2 of Theorem 4.1. We prove a stronger property in the sense:

$$\forall a \in A, w \in \mathcal{U}, w' \in \mathcal{U}' \ \Big((a, w') \in \mathsf{AWin}_{\exists} \wedge (w', w, \varepsilon) \in \mathsf{Win}^{\mathsf{bd}}_{\exists} \Big) \implies (a, w, \varepsilon) \in \mathsf{AWin}^{\epsilon}_{\exists}.$$

So let a, w, w' be such that $(a, w') \in \mathsf{AWin}_{\exists}$ and $(w', w, \varepsilon) \in \mathsf{Win}^{\mathsf{bd}}_{\exists}$ hold. Without loss of generality, let $\varepsilon < 1$; otherwise, the result follows from Theorem 4.2. Since $(a, w') \in \mathsf{AWin}_{\exists}$ Verifier plays some $\delta(a) \in \Delta(a)$ and relation $R \subseteq A \times \mathcal{U}'$ such that $\delta(a) \ \bar{R} \ \gamma'(w')$. Moreover, since $(w', w, \varepsilon) \in \mathsf{Win}^{\mathsf{bd}}_{\exists}$ we know that Verifier in the bisimulation distance game plays some $d \colon \mathcal{U}' \times \mathcal{U} \to [0, 1]$ such that $\bar{d}(\gamma'(w'), \gamma(w)) \leq \varepsilon$. So consider the function

$$d^R(b, v) = \inf_{v' \in \mathcal{U}', bRv'} d(v', v) \quad \text{(for each } b \in A, v \in \mathcal{U}\text{)}.$$

We show that d^R is a valid move in the ϵ-acceptance game. Note that $\ell(a) = \ell(w')$ (resp. $\ell(w') = \ell(w)$) because $\delta(a) \ \bar{R} \ \gamma'(w')$ (resp. $\bar{d}(\gamma'(w'), \gamma(w)) \leq \varepsilon < 1$). Thus, $\ell(a) = \ell(w)$ and performing the case distinction on the type of $\ell(w)$ we get the following two cases:

1. Let $\ell(w) \in \Sigma_0$. Then by Theorem 2.11 we know that $R = \emptyset$; therefore, d^R corresponds to constant 1 function. Moreover, from Theorem 3.1, we know that $\overline{d^R}(\delta(a), \gamma(w)) = 0 \leq \varepsilon$; thus d^R is a valid move. Furthermore, any response of Falsifier will be of the form $(b, v, 1)$ which belongs to $\mathsf{AWin}^{\epsilon}_{\exists}$ by Theorem 4.2. So $(a, w, \varepsilon) \in \mathsf{AWin}^{\epsilon}_{\exists}$.

2. Let $\ell(w) = \sigma \in \Sigma_2$. Then, $\delta(a) = (\sigma, b_0, b_1)$ (for some $b_0, b_1 \in A$), $\gamma(w) = (\sigma, w.0, w.1)$, and $\gamma'(w') = (\sigma, w'.0, w'.1)$. By Theorem 2.11 we know that $R = \{(b_0, w'.0), (b_1, w'.1)\}$. Then $d^R(b_i, w.i) = d(w'.i, w.i)$ (for $i \in \{0, 1\}$). Moreover, using Theorem 3.1 we get:

$$\begin{aligned} \overline{d^R}(\delta(a), \gamma(w)) &= \mathsf{f}(d^R(b_0, w.0), d^R(b_1, w.1) \\ &= \mathsf{f}(d(w'_0, w_0), d(w'_1, w_1))) \\ &= \bar{d}(\gamma'(w'), \gamma(w)) \leq \varepsilon. \end{aligned}$$

Thus, d^R is a valid move in this case. Furthermore, any response of Falsifier in the ϵ-acceptance game will be either of the form $(b, v, 1)$ (which is winning for $\exists$ by Theorem 4.2), or is of the form $(b_i, w.i, \varepsilon')$ (for $\varepsilon' \geq d^R(b_i, w.i) = d(w'.i, w.i)$ and $i \in \{0, 1\}$).

So it remains to show that $(b_i, w.i, \varepsilon') \in \mathsf{AWin}^{\epsilon}_{\exists}$. Suppose Falsifier in the ϵ-acceptance game plays with $(b_i, w.i, \varepsilon')$ (for some $i \in \{0, 1\}$ and $\varepsilon' \geq d(w'.i, w.i)$). Then $(b_i, w'.i) \in \mathsf{AWin}_{\exists}$ as it can be the response of Falsifier in the acceptance game to the winning move R (cf. Theorem 2.11). In addition, $(w'.i, w.i, \varepsilon) \in \mathsf{Win}^{\mathsf{bd}}_{\exists}$ as it is a valid response for Falsifier in the bisimulation distance game. Thus, we are back to the same condition as the start of the round and so $\exists$ can repeat the same argument. □

The converse direction (i.e."$1 \implies 2$") requires more work since we need to construct a labelled tree $\mathcal{T}'$ which itself is based on the idea of a run tree. Suppose $(a, w, \varepsilon) \in \mathsf{AWin}^{\epsilon}_{\exists}$, then a run tree $\mathcal{T}_{(a,w,\varepsilon)}$ is simply the tree generated by following a winning strategy of Verifier (see Theorem 4.4 for the formal definition of domain of a run tree). Now if we encountered a position $(b, v, 1)$ while exploring the run tree $\mathcal{T}_{(a,w,\varepsilon)}$, the successor function functions associated to b and v can be of different type due to the definition of distance lifting (Theorem 3.1). Moreover, since these positions $(b, v, 1)$ are winning for Verifier in the ϵ-acceptance game, thus we cannot use the successor function of v to construct our $\mathcal{T}'$.

To this end, we replace the first occurrence of such positions $(b, v, 1)$ in the run tree by the tree $\mathcal{T}_{(b)}$ accepted by the automaton state b (cf. Theorem 4.3). In other words, the tree $\mathcal{T}'$ (used in the proof of "$1 \implies 2$") is the tree obtained by removing nodes of the form $(b, v, 1)$ from a run tree $\mathcal{T}_{(a,w,\varepsilon)}$ and then attaching the tree $\mathcal{T}_{(b)}$ as defined in the next lemma. $(a, v, 1)$ which is winning for $\exists$ by Lemma 1, or is of the form $(b_i, w.i, \varepsilon)$ for $\varepsilon \geq d(b_i, w.i), i \in \{0, 1\}$.

Lemma 4.3. *For any $a \in A$, there exists a binary tree $\mathcal{T}_{(a)}$ such that $(a, root) \in \mathsf{AWin}_{\exists}$.*

Proof. The idea of the proof is to choose a fix a family of successor functions $\delta(a) \in \Delta(a)$ since $\Delta(a)$ (for each $a \in A$) is nonempty. Now inductively define S_n as follows:

$$S_0 = \{(a, \text{root})\} \qquad S_{j+1} = \{(c, w.0), (c', w.1) \mid \exists b \in A \quad (b, w) \in S_k \wedge \delta(b) = (\sigma, c, c')\}.$$

It is easy to show by induction that these sets has the property that any w can appear at most once in a pair $(b, w) \in S_{|w|}$ (the notation $|w|$ denotes the length of the word w). Let $S = \bigcup_{j=0}^{\infty} S_j$. Now we define $\mathcal{T}_{(a)}$ as follows:

$$\gamma_a \colon \mathcal{U}_{(a)} \to F\mathcal{U}_{(a)} \quad \text{with } \gamma_a(w) = \begin{cases} \delta(b) & \text{if } (b, w) \in S_{|w|} \wedge \delta(b) \in \Sigma_0 \\ (\sigma, w.0, w.1) & \text{if } (b, w) \in S_{|w|} \wedge \delta(b) = (\sigma, c, c') \end{cases},$$

where $\mathcal{U}_{(a)} = \pi_2(S)$ where $\pi_2(S)$ projects the second component from the elements in S. Notice that this γ is well defined as in the second case $(c, w.0), (c', w.1)$ belong to S and so $w.0, w.1$ belong to $\mathcal{U}_{(a)}$. We next show that $(a, \text{root}) \in \mathsf{AWin}_{\exists}$. At position $(a, \text{root}) \in S$, let $\exists$ choose the same $\delta(a) \in \Delta(a)$ and to play the correct $R \subseteq A \times \mathcal{U}_{(a)}$ we identify two cases:

1. Let $\delta(a) = \sigma \in \Sigma_0$. Then since $(a, \text{root}) \in S \implies \text{root} \in \mathcal{U}_{(a)}$ and so $\gamma_a(\text{root}) = \sigma$. Therefore $\exists$ can play $R = \varnothing$ that satisfies $\delta(a)\ \bar{R}\ \gamma(\text{root})$ and $\forall$ gets stuck and loses.

2. Let $\delta(a) = (\sigma, b, b')$. Then $\gamma_a(\text{root}) = (\sigma, 0, 1)$ (recall root is the empty string, so root.i = i for $i \in \{0, 1\}$). Now $\exists$ can play $R = \{(b, 0), (b', 1)\}$ which satisfies $\delta(a)\ \bar{R}\ \gamma_a(\text{root})$ and $\forall$ can choose either $(b, 0)$ or $(b', 1)$. However, by the definition of S, both of these pairs belong to S and so we are back to the same condition as the begining of the round. Hence $\exists$ can repeat the same argument.

So either $\forall$ looses in some round, or the play is infinite, and therefore won by $\exists$. $\square$

Let $\mathcal{T}$ be a labelled tree such that $(a_0, \text{root}, \varepsilon_0) \in \mathsf{AWin}^{\epsilon}_{\exists}$. For all of the triples $(a, w, \varepsilon) \in \mathsf{AWin}^{\epsilon}_{\exists}$, Verifier has a non empty set of winning moves to play which consists of a choice of successor functions in $\Delta(a)$ and a distance function. For each $(a, w, \varepsilon) \in \mathsf{AWin}^{\epsilon}_{\exists}$, let $\delta_{(a,w,\varepsilon)}(a) \in \Delta(a)$ and $d_{(a,w,\varepsilon)} \colon A \times \mathcal{U} \to [0,1]$ denote the successor and distance functions played by Verifier to win at the position (a, w, ε). Then we let $M = \bigcup_{j=0}^{\infty} M_j$ and define M_n inductively as follows: $M_0 = \{(a_0, \text{root}, \varepsilon_0)\}$ and

$$M_{n+1} = \Big\{ \Big(b_i, w.i, d_{(a,w,\varepsilon)}(b_i, w.i) \Big) \mid i \in \{0,1\} \wedge (a, w, \varepsilon) \in M_n \wedge \delta_{(a,w,\varepsilon)}(a) = (\sigma, b_0, b_1) \wedge \gamma(w) = (\sigma, w.0, w.1) \Big\}.$$

The next proposition states that projecting the second component from the elements in M results in the domain of a run tree, denoted $\mathcal{T}_{(a_0,\text{root},\varepsilon)}$, whenever $(a_0, \text{root}, \varepsilon) \in \mathsf{AWin}^{\epsilon}_{\exists}$.

Proposition 4.4. *The set $\pi_2(M)$ (i.e. projecting the second component from the triples in M) is both prefix closed and sibling closed.*

Proof of Theorem 4.1($1 \implies 2$). Assume $\mathcal{T}$ is ε_0-accepted by $\mathcal{A}$, so $(a_0, \text{root}, \varepsilon_0) \in \mathsf{AWin}^{\epsilon}_{\exists}$. Define the following sets using the domain M of a run tree constructed above:

- $\mathcal{U}_1 = \{w \mid \exists a \in A, \varepsilon \in [0,1) \quad (a, w, \varepsilon) \in M\}$;
- $\mathcal{U}_2 = \Big\{ v \mid \exists a \in A \Big((a, v, 1) \in M \wedge \forall u \; (u \leq v \implies \nexists b \; (b, u, 1) \in M) \Big) \Big\}$;
- $\mathcal{U}_3 = \{v.t \mid v \in \mathcal{U}_2 \text{ with } (a, v, 1) \in M \wedge t \in \mathcal{U}_{(a)}\}$.

Notice that $\mathcal{U}_2 \subset \mathcal{U}_3$ as $t \in \mathcal{U}_{(a)}$ can be the empty string.

Now we will construct a tree $\mathcal{T}'$ with the desired property by first defining its domain as $\mathcal{U}' = \mathcal{U}_1 \cup \mathcal{U}_3$. It is prefix-closed and sibling-closed by tree substitution (Theorem 2.5), since both M and all $\mathcal{T}_{(a)}$ are binary trees. Moreover, define the successor function $\gamma' \colon \mathcal{U}' \to F(\mathcal{U}')$:

$$\gamma'(w) = \begin{cases} \gamma(w) & \text{if } w \in \mathcal{U}_1 \\ \gamma_a(t) & \text{if } w \in \mathcal{U}_3 \wedge w = v.t \text{ (for some } t) \end{cases},$$

where γ_a is the successor function as defined in the proof of Theorem 4.3. Next we claim the following two properties hold from which the final result follows directly since the starting position $(a_0, \text{root}, \varepsilon_0)$ has the said property and therefore $(\text{root}, \text{root}, \varepsilon_0) \in \mathsf{Win}^{\mathsf{bd}}_{\exists}$ and $(a_0, \text{root}) \in \mathsf{AWin}_{\exists}$.

Claim 4.5. 1. $\forall w \in \mathcal{U}_1 \cup \mathcal{U}_2 \quad (a, w, \varepsilon) \in \mathsf{AWin}^{\epsilon}_{\exists} \implies (w, w, \varepsilon) \in \mathsf{Win}^{\mathsf{bd}}_{\exists}$.

2. $\forall w \in \mathcal{U}_1 \cup \mathcal{U}_2 \quad (a, w, \varepsilon) \in M \implies (a, w) \in \mathsf{AWin}_{\exists}$. $\square$

Proof of Claim. For Item 1, without loss of generality, let $w \in \mathcal{U}_1$ (otherwise the result follows from Theorem 3.6(Item 1). Then $(a, w, \varepsilon) \in M$ (for some a and $\varepsilon < 1$). Therefore $\gamma'(w) = \gamma(w) = \sigma$. If $\sigma \in \Sigma_0$, $\exists$ can choose $d = 1$ and the position is winning due to Theorem 3.6(Item 1). If $\sigma \in \Sigma_2$, then $\gamma'(w) = (\sigma, 0, 1) = \gamma(w)$ and $\exists$ can choose $d : \mathcal{U} \times \mathcal{U}' \to [0,1]$ such that $d(w.0, w.0) = d_{(a,w,\varepsilon)}(w.0, w.0)$ and $d(w.1, w.1) = d_{(a,w,\varepsilon)}(w.1, w.1)$ and $d = 1$ otherwise. Then the choice of $\forall$ is either $(v, v', 1)$ that is winning for $\exists$ by Theorem 3.6(Item 1), or it is $(w.i, w.i, d_{(a,w,\varepsilon)}(w.i, w.i))$ by Theorem 3.10. But this position has the property that it

is in $\mathsf{AWin}^{\epsilon}_{\exists}$ and $w.i \in \mathcal{U}_1 \cup \mathcal{U}_2$ by definition. So we are back at the same condition and $\exists$ can repeat the same strategy. So the play is infinite and therefore winning by $\exists$.

For Item 2, if $w \in \mathcal{U}_2$, then by definition $\mathcal{T}'_w$ is exactly the tree that is accepted by $\mathcal{A}$ starting from w. So let $w \in \mathcal{U}_1$. Then $(a, w, \varepsilon) \in M$ for $\varepsilon < 1$ and $\gamma'(w) = \gamma(w) = \sigma$. If $\sigma \in \Sigma_0$, $\exists$ can choose $R = \varnothing$ and wins. If $\sigma \in \Sigma_2$, then $\gamma'(w) = \gamma(w) = (\sigma, w.0, w.1)$ and $\exists$ can choose $\delta_{a,w,\varepsilon}(a)$. Also as $\varepsilon < 1$ we have the property $\bar{d}_{a,w,\varepsilon}\big(\delta_{a,w,\varepsilon}(a), \gamma'(w)\big) \leq \varepsilon < 1$ so $\delta_{a,w,\varepsilon}(a)$ has to be of the form (σ, b_0, b_1). Now for the relation $\exists$ can choose $R = \{(b_0, w.0), (b_1, w.1)\}$ and the next move $(b_i, w.i)$ of $\forall$ is such that $w.i \in \mathcal{U}_1 \cup \mathcal{U}_2$ as $(b_i, w.i, d_{(a,w,\varepsilon)}(b_i, w.i)) \in M$. So back to the same condition and $\exists$ can repeat the same strategy. So the play is infinite and therefore winning by $\exists$. ⊣

5 Relating Defect and Measure

In this section, we apply our main result to two case studies: measuring termination and failed executions of programs, modelled as trees. To this end, we define a measure on subtrees and relate the quantity of interest to a distance ε accepted in the ϵ-acceptance game. The corresponding characterisations are stated in Theorem 5.1 and Theorem 5.3.

Let $\mathcal{T}$ be a binary tree. For every subtree $\mathcal{T}_w$, the *measure* of $\mathcal{T}_w$ is defined as

$$\mu(\mathcal{T}_w) = 2^{-|w|}$$

The existence and unicity of μ are guaranteed by Carathéodory's extension theorem [18]. Then $(\mathcal{T}, \mu)$ is a measure space. Note that μ is also a probability measure because $\mu(\mathcal{T}) = 1$. We also introduce a notation reminiscent of conditional probabilities. Consider two binary strings $w, v \in \mathcal{U}$, the *relative measure* $\mu(v \mid w) = 2^{|w|-|v|}$ if w is a prefix of v; 0 otherwise.

Measuring termination Let $\Sigma_0 = \{\star\}, \Sigma_2 = \{\sigma\}$. Let $\mathbb{A}$ be a tree automaton with a single state $A = \{a_0\}$ such that $\mathrm{Ar}(a_0) = 2$, and transition function $\Delta(a_0) = \{(\sigma, a_0, a_0)\}$. Moreover, let $Acc = A^{\omega}$. Consider the distance lifting induced by $\mathsf{f}\colon [0,1] \times [0,1] \to [0,1]$ given by $\mathsf{f}(x, y) = \frac{1}{2}x + \frac{1}{2}y$. Then the following proposition holds.

Proposition 5.1. *For a tree $\mathcal{T}$ and the set $L = \{l \in \mathcal{U} \mid Ar(v) = 0\}$ of leaves we have*

$$(a_0, \mathit{root}, \varepsilon) \in \mathsf{AWin}^{\epsilon}_{\exists} \iff \varepsilon \geq \sum_{l \in L} \mu(l \mid \mathit{root}).$$

Proof. $\boxed{\Leftarrow}$ Assume $\varepsilon \geq \sum_{l \in L} \mu(l \mid \mathrm{root})$; starting from the position $(a_0, \mathrm{root}, \varepsilon)$, we show that $\exists$ has a winning strategy. The choice of $\delta(a_0)$ is unique, and for the distance function $\exists$ can choose d as follows. If $\ell(w) = \star$, let $d = 1$, and if $\ell(w) = \sigma$, define

$$d(a_0, \mathrm{root}.i) = \sum_{l \in L} \mu(l \mid \mathrm{root}.i)\ (i \in \{0,1\}), \qquad d(a_0, v) = 1 \text{ for } v \notin \{\mathrm{root}.0, \mathrm{root}.1\}.$$

In the first case, $\bar{d}(\delta(a_0), \gamma(\mathrm{root})) = 1$ because $\mathrm{Ar}(\mathrm{root}) \neq \mathrm{Ar}(a_0)$. However, since w is a leaf, $\sum_{l \in L} \mu(l \mid \mathrm{root}) = 1$, hence $\varepsilon \geq 1$, and d is valid. Any response $(a_0, v, 1)$ of $\forall$ belongs to $\mathsf{AWin}^{\epsilon}_{\exists}$ by Theorem 4.2.
In the second case, when $\ell(\mathrm{root}) = \ell(a_0) = \sigma \in \Sigma_2$, we simplify $\bar{d}(\delta(a_0), \gamma(\mathrm{root}))$ as follows:

$$\mathsf{f}\big(d(a_0, \mathrm{root}.0), d(a_0, \mathrm{root}.1)\big) = \tfrac{1}{2}\big(d(a_0, \mathrm{root}.0) + d(a_0, \mathrm{root}.1)\big) = \sum_{l \in L} \mu(l \mid \mathrm{root}) \leq \varepsilon.$$

The last equality follows from the definition of μ. Hence d is valid. Any response of $\forall$ is then either $(a_0, v, 1)$ which belongs to $\mathsf{AWin}^{\epsilon}_{\exists}$, or $(a_0, w.i, \varepsilon_i)$ with $\varepsilon_i \geq d(a_0, .i) = \sum_{l \in L} \mu(l \mid \text{root}.i)$ for $i \in \{0, 1\}$. However, in the second case, the invariant is preserved, and we return to the same condition as before (with another w instead of root), so $\exists$ can repeat this strategy. Therefore, the play is infinite and belongs to Acc, and hence $(a_0, \text{root}, \varepsilon) \in \mathsf{AWin}^{\epsilon}_{\exists}$.

$\boxed{\Rightarrow}$ To prove this direction, we need the following claim.

Claim 5.2. The only tree accepted by $\mathbb{A}$ is the full binary tree $\mathcal{T}_A$ in which every node is labelled by σ. I.e., $\mathcal{T}_A = (\mathcal{U}_A, \ell)$ where $\mathcal{U}_A = \{0, 1\}^*$ and $\gamma_A(w) = (\sigma, w.0, w.1)$ for $w \in \mathcal{U}_A$.

Proof of Claim. The accepted tree exists since $\mathbb{A}$ satisfies the conditions of Theorem 4.3. Let $\mathcal{T}_A$ be such a tree. We show that every node of $\mathcal{T}_A$ must have label σ and exactly two children.

Assume $(a_0, w) \in \mathsf{AWin}_{\exists}$. Then $\exists$ has a move R such that $\delta(a_0)\, \bar{R}\, \gamma_A(w)$, which implies $\gamma_A(w) = (\sigma, w.0, w.1)$. Moreover, any possible response of $\forall$, namely $(a_0, w.0)$ or $(a_0, w.1)$, again belongs to $\mathsf{AWin}_{\exists}$, so the argument can be repeated.

Since $(a_0, \text{root}) \in \mathsf{AWin}_{\exists}$, starting from the root we inductively obtain that every node is labelled by σ and has two successors. Hence $\mathcal{T}_A$ is the full binary tree. $\dashv$

Assume $(a_0, \text{root}, \varepsilon) \in \mathsf{AWin}^{\epsilon}_{\exists}$. By Theorem 3.7, there exists a tree $\mathcal{T}'$ accepted by $\mathbb{A}$ such that $\mathsf{bd}(\mathcal{T}', \mathcal{T}) \leq \varepsilon$. By the above claim, $\mathcal{T}'$ is the full binary tree $\mathcal{T}_A$. Since $\gamma_A(v) = (\sigma, v.0, v.1)$ for all v and bd is a bisimulation distance, we can write (below $w \in \mathcal{U}$):

$$\mathsf{bd}(w, w) \geq \overline{\mathsf{bd}}(\gamma_A(w), \gamma(w)) = \begin{cases} 1 & \text{if } \gamma(w) = \star, \\ \frac{1}{2}\big(\mathsf{bd}(w.0, w.0) + \mathsf{bd}(w.1, w.1)\big) & \text{otherwise.} \end{cases}$$

Hence, $\mathsf{bd}(w, w) = 1$ if $\gamma(w) = \star$ and $\mathsf{bd}(w, w) \geq \frac{1}{2}\big(\mathsf{bd}(w.0, w.0) + \mathsf{bd}(w.1, w.1)\big)$ otherwise. On the other hand, by definition of the measure μ, for all $w \in \mathcal{U}$ we have

$$\sum_{l \in L} \mu(l \mid w) = \begin{cases} 1 & \text{if } \gamma(w) = \star, \\ \frac{1}{2}\Big(\sum_{l \in L} \mu(l \mid w.0) + \sum_{l \in L} \mu(l \mid w.1)\Big) & \text{otherwise.} \end{cases}$$

Thus, $\sum_{l \in L} \mu(l \mid w) \leq \mathsf{bd}(w, w)$ (for $w \in \mathcal{U}$); so $\sum_{l \in L} \mu(l \mid \text{root}) \leq \mathsf{bd}(\text{root}, \text{root}) \leq \varepsilon$. $\square$

Measuring failed executions We now illustrate the approach on a very simple yet crucial property, checked by a non-deterministic automaton: the absence of error labels in a binary tree with leaves. The quantitative aspect game will *tolerate* error labels, but up to a certain measure that can not be greater than ε. So let $\Sigma_0 = \{\star\}, \Sigma_2 = \{\mathsf{ok}, \mathsf{error}\}$.

First, we define $\mathcal{T}_{err}$ as the set of subtrees of $\mathcal{T}$ whose root is labeled with error, and $\widehat{\mathcal{T}_{err}}$ as the subset of those trees whose ancestors are all labeled with ok. Since any element of $\mathcal{T}_{err} \setminus \widehat{\mathcal{T}_{err}}$ must be a subtree of an element of $\widehat{\mathcal{T}_{err}}$, they have the same measure. Similarly, we define the conditional sets $\mathcal{T}_{err} \mid w$ and $\widehat{\mathcal{T}_{err}} \mid w$ where the measure originates from the node w and we have $\mu(\mathcal{T}_{err} \mid w) = \frac{\mu(\mathcal{T}_{err})}{2^{|w|}}$ (and similarly for $\widehat{\mathcal{T}_{err}} \mid w$), effectively normalizing the measure relative to the depth of the prefix w.

Let $\mathbb{A}$ be a tree automaton with a single state $A = \{a_0\}$ and the transition function $\Delta(a_0) = \{\star, (\mathsf{ok}, a_0, a_0)\}$. Moreover, let $Acc = A^{\omega}$. Consider the same distance lifting induced by $\mathsf{f}\colon [0, 1] \times [0, 1] \to [0, 1]$ given by $\mathsf{f}(x, y) = \frac{1}{2}x + \frac{1}{2}y$. Then

Proposition 5.3. $(a_0, \mathit{root}, \varepsilon) \in \mathsf{AWin}^{\epsilon}_{\exists} \iff \varepsilon \geq \mu\left(\widehat{\mathcal{T}_{err}}\right)$

Proof. $\boxed{\Leftarrow}$ Assume $\varepsilon \geq \mu\left(\widehat{\mathcal{T}_{err}}\right) = \mu\left(\widehat{\mathcal{T}_{err}} \mid \text{root}\right)$. We now define a winning strategy for $\exists$ starting from $(a_0, \text{root}, \varepsilon)$. We proceed by case distinction:

1. If $\gamma(\text{root}) = \star$, then $\exists$ can play $\delta(a_0) = \star$ and $d = 1$, which is a valid move since $\bar{d}(\delta(a_0), \gamma(\text{root})) = 0$. And the next move of $\forall$ will be $(a_0, v, 1) \in \mathsf{AWin}^{\epsilon}_{\exists}$ by Theorem 4.2.

2. If $\gamma(\text{root}) = (\mathsf{error}, \text{root}.0, \text{root}.1)$, then $\exists$ can choose any $\delta(a_0) \in \Delta(a_0)$ and $d = 1$. In this case, $\mathcal{T}_{\text{root}}$ is itself a subtree starting with an error root, so that $\mu\left(\widehat{\mathcal{T}_{err}}\right) = 1$, and therefore $\varepsilon \geq 1$ (implying $\varepsilon = 1$). Thus d is a valid choice as $\bar{d}(\delta(a_0), \gamma(\text{root})) \leq \varepsilon$. Then any choice of $\forall$ will be of the form $(a_0, v, 1)$, that again belongs to $\mathsf{AWin}^{\epsilon}_{\exists}$ by Theorem 4.2.

3. If $\gamma(\text{root}) = (\mathsf{ok}, \text{root}.0, \text{root}.1)$, then $\exists$ can choose $\delta(a_0) = (ok, a_0, a_0)$ and d as follows:

   $$d(a_0, \text{root}.i) = \mu\left(\widehat{\mathcal{T}_{err}} \mid \text{root}.i\right) \ (i \in \{0,1\}) \qquad d(a_0, v) = 1 \quad \text{for } v \notin \{\text{root}.0, \text{root}.1\}$$

   It is a valid choice because of the following derivation.

   $$\bar{d}(\delta(a_0), \gamma(\text{root})) = \frac{1}{2}\left(\mu\left(\widehat{\mathcal{T}_{err}} \mid \text{root}.0\right) + \mu\left(\widehat{\mathcal{T}_{err}} \mid \text{root}.1\right)\right) = \mu\left(\widehat{\mathcal{T}_{err}}\right) \leq \varepsilon$$

   Notice that the last equality holds because the label of root itself is ok. So, when measuring the set $\widehat{\mathcal{T}_{err}} \mid \text{root}$, every error-rooted subtree in that set must start from root.0 or root.1, and cannot be from root itself. The next move of $\forall$ must then either be $(a_0, v, 1)$ (which again belongs to $\mathsf{AWin}^{\epsilon}_{\exists}$), or $(a_0, \text{root}.i, \varepsilon_i)$ with $\varepsilon_i \geq d(a_0, \text{root}.i) = \mu\left(\widehat{\mathcal{T}_{err}} \mid \text{root}.i\right)$ for $i \in \{0,1\}$. In the latter case, the invariant is preserved (with w instead of root), and we return to the same condition as the beginning of the round, so that $\exists$ can repeat this strategy. This induces an infinite play, thus belonging to Acc.

$\boxed{\Rightarrow}$ Assume that $(a_0, \text{root}, \varepsilon) \in \mathsf{AWin}^{\epsilon}_{\exists}$. Then, by Theorem 4.1, there is a tree $\mathcal{T}'$ that is accepted by $\mathcal{A}$ with $\mathsf{bd}(\mathcal{T}', \mathcal{T}) \leq \varepsilon$. The construction of $\mathcal{T}'$ in the proof of Theorem 4.1 has the following form: starting from the root, $\mathcal{T}'$ coincides with $\mathcal{T}$ until an error node is met. Once we reach an error node, we replace it with a subtree accepted by $\mathcal{A}$. That is because the run tree following a winning strategy of $\exists$ will be equal to $\mathcal{T}$ on a path that is free of error, and when we see an error node v, for any choice of d and δ of $\exists$, $\bar{d}(\delta(a_0), \gamma(v)) = 1$, so $(a_0, v, 1)$ will appear for the first time in the run tree and that is when we substitute $\mathcal{T}_{(a_0)}$. Moreover, $\mathcal{T}_{(a_0)}$ can be a single leaf $\star$, as it is always accepted by $\mathcal{A}$ ($\exists$ can choose $\delta(a_0) = \star$ and $d = 1$). So $\mathcal{T}'$ would be equal to $\mathcal{T}$ until it reaches an error node, at which point it becomes a leaf $\star$.

As bd is a bisimulation distance, for each $v \in \mathcal{U}'$:

$$\mathsf{bd}(v,v) \geq \bar{\mathsf{bd}}(\gamma'(v), \gamma(v)) = \begin{cases} 0 & \text{if } \gamma'(v) = \gamma(v) = \star \\ \frac{1}{2}(\mathsf{bd}(v.0, v.0) + \mathsf{bd}(v.1, v.1)) & \text{if } \gamma'(v) = \gamma(v) = (\mathsf{ok}, v.0, v.1) \\ 1 & \text{if } \gamma'(v) = \star, \gamma(v) = (\mathsf{error}, v.0, v.1) \end{cases}$$

Note that these are the only cases since by construction of $\mathcal{T}'$, it does not have any node labelled error, and all of its labels ok match with the same node in $\mathcal{T}'$.

Observe that if a tree's root is labelled with ok then it does not belong to the set of its subtrees whose root is labelled with error. From the definition of μ, one then obtains that:

$$\mu(\widehat{\mathcal{T}_{err}} \mid v) = \begin{cases} 0 & \text{if } \gamma(v) = \star (= \gamma'(v)) \\ \frac{1}{2}(\mu\left(\widehat{\mathcal{T}_{err}} \mid v.0\right) + \mu\left(\widehat{\mathcal{T}_{err}} \mid v.1\right)) & \text{if } \gamma(v) = (\mathsf{ok}, v.0, v.1) (= \gamma'(v)) \\ 1 & \text{if } \gamma(v) = (\mathsf{error}, v.0, v.1) \text{ (then } \gamma'(v) = \star) \end{cases}$$

It follows that $\mu\left(\widehat{\mathcal{T}_{err}} \mid v\right) \leq \mathsf{bd}(v, v)$ for all $v \in \mathcal{U}'$. In particular, since root $\in \mathcal{U}'$, $\mu\left(\widehat{\mathcal{T}_{err}}\right) \leq \mathsf{bd}(\text{root}, \text{root}) \leq \varepsilon$, which completes the proof. □

6 Conclusion

In this paper, we extended the acceptance game [15] between a tree automaton and a labelled binary tree to a quantitative setting that lead to a more 'robust' notion of acceptance (instead of a Boolean one). Intuitively, our main result (cf. Theorem 4.1) shows how the winning strategy $\mathsf{AWin}^{\epsilon}_{\exists}$ of Verifier in an ϵ-acceptance game can be seen as the winning strategy of Verifier playing the traditional acceptance and bisimulation distance game simultaneously. Furthermore, the two examples from Section 5 suggest a strong connection between the optimal ε value for the game and Lebesgue measures over trees. In both examples, this optimal ε value corresponds to the measure of the set of subtrees that are not valid from the point of view of the execution of the automaton.

Future work Our work is built on the acceptance game [15] between an F-automaton and an F-coalgebra. As the first step, it is natural to extend the proof of main theorem to the level of coalgebras. This will allow one to recover various instances of ϵ-acceptance games over variety of structures like arbitrary n-ary trees, probabilistic trees, etcetera. More concretely, our work suggests that distance lifting (by an appropriate choice of f) results in a measure on binary trees. In the future, this connection needs a detailed treatment; the exact relationship between our distance lifting and measures on binary tree is left as future work.

References

[1] David Janin and Igor Walukiewicz. "Automata for the modal μ-calculus and related results". In: *Mathematical Foundations of Computer Science 1995*. Ed. by Jiří Wiedermann and Petr Hájek. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 552–562. ISBN: 978-3-540-44768-9.

[2] Robin Milner. *Communication and concurrency*. PHI Series in computer science. Prentice Hall, 1989. ISBN: 978-0-13-115007-2.

[3] David Park. "Concurrency and automata on infinite sequences". In: *Theoretical Computer Science*. Ed. by Peter Deussen. Berlin, Heidelberg: Springer Berlin Heidelberg, 1981, pp. 167–183. ISBN: 978-3-540-38561-5.

[4] Josée Desharnais et al. "Metrics for Labeled Markov Systems". In: *CONCUR'99 Concurrency Theory*. Ed. by Jos C. M. Baeten and Sjouke Mauw. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 258–273. ISBN: 978-3-540-48320-5.

[5] Franck van Breugel and James Worrell. "Towards Quantitative Verification of Probabilistic Transition Systems". In: *Automata, Languages and Programming*. Ed. by Fernando Orejas, Paul G. Spirakis and Jan van Leeuwen. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 421–432. ISBN: 978-3-540-48224-6.

[6] Hubert Comon et al. *Tree Automata Techniques and Applications*. 2008, p. 262. URL: https://inria.hal.science/hal-03367725.

[7] Krishnendu Chatterjee, Laurent Doyen and Thomas A. Henzinger. “Quantitative languages”. In: *ACM Trans. Comput. Log.* 11.4 (2010), 23:1–23:38. DOI: 10.1145/1805950.1805953. URL: https://doi.org/10.1145/1805950.1805953.

[8] Moshe Y. Vardi. “Automatic Verification of Probabilistic Concurrent Finite-State Programs”. In: *26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985*. IEEE Computer Society, 1985, pp. 327–338. DOI: 10.1109/SFCS.1985.12. URL: https://doi.org/10.1109/SFCS.1985.12.

[9] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. The MIT Press, 2008. ISBN: 026202649X.

[10] Josée Desharnais et al. “Metrics for labelled Markov processes”. In: *Theor. Comput. Sci.* 318.3 (2004), pp. 323–354. DOI: 10.1016/J.TCS.2003.09.013. URL: https://doi.org/10.1016/j.tcs.2003.09.013.

[11] Paolo Baldan et al. “Coalgebraic Behavioral Metrics”. In: *Log. Methods Comput. Sci.* 14.3 (2018). DOI: 10.23638/LMCS-14(3:20)2018. URL: https://doi.org/10.23638/LMCS-14(3:20)2018.

[12] Shin-ya Katsumata and Tetsuya Sato. “Codensity Liftings of Monads”. In: *6th Conference on Algebra and Coalgebra in Computer Science (CALCO 2015)*. Ed. by Lawrence S. Moss and Pawel Sobocinski. Vol. 35. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015, pp. 156–170. ISBN: 978-3-939897-84-2. DOI: 10.4230/LIPIcs.CALCO.2015.156. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.CALCO.2015.156.

[13] Filippo Bonchi, Barbara König and Daniela Petrisan. “Up-To Techniques for Behavioural Metrics via Fibrations”. In: *29th International Conference on Concurrency Theory (CONCUR 2018)*. Ed. by Sven Schewe and Lijun Zhang. Vol. 118. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018, 17:1–17:17. ISBN: 978-3-95977-087-3. DOI: 10.4230/LIPIcs.CONCUR.2018.17. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.CONCUR.2018.17.

[14] Samuel Humeau, Daniela Petrisan and Jurriaan Rot. “Correspondences Between Codensity and Coupling-Based Liftings, a Practical Approach”. In: *33rd EACSL Annual Conference on Computer Science Logic (CSL 2025)*. Ed. by Jörg Endrullis and Sylvain Schmitz. Vol. 326. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025, 29:1–29:18. ISBN: 978-3-95977-362-1. DOI: 10.4230/LIPIcs.CSL.2025.29. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.CSL.2025.29.

[15] Clemens Kupke and Yde Venema. “Coalgebraic Automata Theory: Basic Results”. In: *Log. Methods Comput. Sci.* 4.4 (2008). DOI: 10.2168/LMCS-4(4:10)2008. URL: https://doi.org/10.2168/LMCS-4(4:10)2008.

[16] Bart Jacobs. *Introduction to Coalgebra: Towards Mathematics of States and Observation*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2016.

[17] Michael O. Rabin. “Decidability of Second-Order Theories and Automata on Infinite Trees”. In: *Transactions of the AMS* 141 (1969), pp. 1–35.

[18] H. Bauer. *Measure and Integration Theory*. De Gruyter studies in mathematics. W. de Gruyter, 2001. ISBN: 9783110167191. URL: https://books.google.co.uk/books?id=5mskCf1wKPkC.