Topological Characterisation of Multi-buffer Simulation

Milka Hutagalung(✉)

University of Kassel, Kassel, Germany
milka.hutagalung@uni-kassel.de

Abstract. Multi-buffer simulation is an extension of simulation pre-order that can be used to approximate inclusion of languages recognised by Büchi automata up to their trace closures. It has been shown that multi-buffer simulation with unbounded buffers can be characterised with the existence of a continuous function f that witnesses trace closure inclusion. In this paper, we show that such a characterisation can be refined to the case where we only consider bounded buffers by requiring the function f to be Lipschitz continuous. This characterisation only holds for some restricted classes of automata. One of the automata should only produce words where each letter does not commute unboundedly to the left or right. We will show that such an automaton can be characterised with a cyclic-path-connected automaton, which is a refinement of a syntactic characterisation of an automaton that has a regular trace closure.

1 Introduction

Simulation is a pre-order relation that relates two automata $\mathcal{A}$, $\mathcal{B}$ in the sense that one automaton simulates the other. It is used to minimise and approximate language inclusion between automata on words and trees [1,3,4,6].

Multi-buffer simulation is introduced in [9] as an extension of simulation for non-deterministic Büchi automata [7]. It extends the framework of the standard simulation with n FIFO buffers of capacities $k_1, \ldots, k_n \in \mathbb{N} \cup \{\omega\}$. The buffers are associated with the alphabets $\Sigma_1, \ldots, \Sigma_n \subseteq \Sigma$, respectively. Spoiler plays as in the standard simulation. He moves his pebble by reading a letter one by one. However, Duplicator can skip her turn, and push the letter that is chosen by Spoiler to the associated buffers. Duplicator can move and pop some letters from the buffers in some round later. In [9], it is shown that multi-buffer simulation is undecidable in general but decidable if all buffers have bounded capacities, i.e. when $k_1, \ldots, k_n \in \mathbb{N}$. Multi-buffer simulation can be used to approximate inclusion of Mazurkiewicz trace closure. If we have multi-buffer simulation $\mathcal{A} \sqsubseteq^{k,\ldots,k} \mathcal{B}$ for some $k \in \mathbb{N} \cup \{\omega\}$, then we have $L(\mathcal{A}) \subseteq [L(\mathcal{B})]_I$ that is equivalent to the inclusion of Mazurkiewicz trace closure $[L(\mathcal{A})]_I \subseteq [L(\mathcal{B})]_I$, which is known to be undecidable [11] and even highly undecidable [5].

The winning strategy for Duplicator in multi-buffer simulation game can be characterised with a continuous function [9]. We have multi-buffer simulation

M. Hague and I. Potapov (Eds.): RP 2017, LNCS 10506, pp. 101–117, 2017.
DOI: 10.1007/978-3-319-67089-8_8

$\mathcal{A} \sqsubseteq^{\omega,\ldots,\omega} \mathcal{B}$ iff there exists a continuous function f that maps the accepting runs of $\mathcal{A}$ to the ones of $\mathcal{B}$ over trace equivalent words. Intuitively, this characterisation could also be lifted to the case of bounded buffer: $\mathcal{A} \sqsubseteq^{k,\ldots,k} \mathcal{B}$, for some $k \in \mathbb{N}$ iff there exists such a Lipschitz continuous function f. Unfortunately this is not the case. There are $\mathcal{A}$, $\mathcal{B}$ in which such a Lipschitz continuous function f exists but buffered simulation with bounded buffers does not hold, i.e. $\mathcal{A} \not\sqsubseteq^{k,\ldots,k} \mathcal{B}$ for any $k \in \mathbb{N}$. Hence one may ask whether we can add some restriction on the structure of $\mathcal{A}$, $\mathcal{B}$ such that the characterisation holds. This would give a good theoretical justification for multi-buffer simulation with bounded buffers.

We answer this question in this work. We first show that the characterisation with Lipschitz continuity fails in two cases. The first one is the case where Spoiler can form a non-accepting run that cannot be mimicked by Duplicator, which is irrelevant to the use of multi-buffer simulation. We can avoid this by restricting Duplicator's automaton to be complete. The second one is the case where Spoiler can produce a word, in which one of its letters, suppose a, can commute unboundedly to the left or right. Spoiler might read a word where a occurs at a very late position, but in a trace equivalent word that should be produced by Duplicator, a occurs at a very early position. In this case, Duplicator needs to store unboundedly many irrelevant letters before she can read a, and eventually violates the capacity constraint. We will show that we can avoid this by restricting Spoiler's automaton to only produce words where each of its letters cannot commute unboundedly, i.e. there exists a bound $k \in \mathbb{N}$, such that each letter commutes at most k steps to the left or right.

Note that the first restriction is a syntactic restriction, but the second one is not. We cannot check syntactically whether Spoiler's automaton $\mathcal{A}$ admits such a bound k by looking at the structure of $\mathcal{A}$. Hence, it is reasonable to ask whether we can have an equivalent syntactic restriction. For this purpose, we will show that we can lift the syntactic characterisation of *loop-connected* automaton, a syntactic characterisation of an automaton that has a regular trace closure [2].

2 Preliminaries

For any alphabet Σ, we denote the set of finite words over Σ with Σ^*, the set of infinite words over Σ with Σ^ω, and $\Sigma^\infty = \Sigma^* \cup \Sigma^\omega$. For any word $w \in \Sigma^\infty$ of length $n \in \mathbb{N} \cup \{\infty\}$, we denote with $|w| = n$ the length of w, $|w|_a$ the number of a in w, $\mathsf{Pos}(w) \subseteq \mathbb{N}$ the set of positions in w, $w(i)$ the letter of w at position i, and $\Sigma_w = \{w(i) \mid i \in \mathsf{Pos}(w)\}$ the alphabet of w.

A *non-deterministic Büchi automaton* (NBA) is a tuple $\mathcal{A} = (Q, \Sigma, q_I, E, F)$, where Q is a finite set of *states*, Σ is an alphabet, $q_I \in Q$ is the *initial state*, $E \subseteq Q \times \Sigma \times Q$ is the *transition relation*, and $F \subseteq Q$ is the set of *final states*. We denote with $|\mathcal{A}|$ the number of states of $\mathcal{A}$. We sometimes write $p \xrightarrow{a} p'$ if $(p, a, p') \in E$. A *run* of $\mathcal{A}$ on $a_0 a_1 \ldots \in \Sigma^\infty$ is an alternating sequence of states and letters $\rho = q_0 a_0 q_1 a_1 \ldots$ with q_0 being the initial state of $\mathcal{A}$ and $(q_i, a_i, q_{i+1}) \in E$ for all $i \geq 0$. The run ρ is *accepting* if $q_i \in F$ for infinitely many $i \in \mathbb{N}$. The set of runs and accepting runs are respectively denoted with $\mathsf{Run}(\mathcal{A})$ and

$\mathsf{AccRun}(\mathcal{A})$. For any run $\rho = q_0 a_0 q_1 a_1 \ldots$, the word of ρ is $\mathsf{word}(\rho) = a_0 a_1 \ldots \in \Sigma^\infty$, and the *language* of $\mathcal{A}$ is $L(\mathcal{A}) = \{\mathsf{word}(\rho) \mid \rho \in \mathsf{AccRun}(\mathcal{A})\}$. Moreover, for any finite run $r = q_0 a_0 q_1 a_1 \ldots q_n$, the length of r is $|r| = n$.

2.1 Mazurkiewicz Traces

An *independence alphabet* is a pair (Σ, I), where Σ is a finite alphabet and $I \subseteq \Sigma \times \Sigma$ is an irreflexive and symmetric relation, called *independence relation*. The relation $D = \Sigma \times \Sigma \setminus I$ is called the *dependence relation*, and the graph $G = (\Sigma, E)$, where $E = \{(a, b) \mid (a, b) \in D \text{ and } a \neq b\}$ is called the *dependency graph* of (Σ, I). The tuple $\hat{\Sigma} = (\Sigma_1, \ldots, \Sigma_n)$ where the set $\{\Sigma_1, \ldots, \Sigma_n\}$ is the set of maximal cliques in G is called the *distributed alphabet* of (Σ, I).

Given an independence alphabet (Σ, I), let $\hat{\Sigma} = (\Sigma_1, \ldots, \Sigma_n)$ be the corresponding distributed alphabet, and let $\pi_i : \Sigma^\infty \to \Sigma_i^\infty$ be a projection from the word over Σ to the word over Σ_i for all $i \in \{1, \ldots, n\}$. The projection $\pi_i(w)$ is obtained by deleting from w all letters that do not belong to Σ_i. For any $w, w' \in \Sigma^\infty$ over (Σ, I), we say w is *trace equivalent* with w', i.e. $w \sim_I w'$, iff $\pi_i(w) = \pi_i(w')$ for all $i \in \{1, \ldots, n\}$. For example if $\Sigma = \{a, b, c\}$, $I = \{(b, c), (c, b)\}$ then $\hat{\Sigma} = (\{a, b\}, \{a, c\})$, and we have $a(bc)^\omega \sim_I a(cb)^\omega$. For any NBA $\mathcal{A}$ over (Σ, I), the *trace closure* of $\mathcal{A}$ is the language $[L(\mathcal{A})]_I = \{w \in \Sigma^\omega \mid \exists w' \in L(\mathcal{A}) \colon w \sim_I w'\}$.

Given an NBA $\mathcal{A}$ over (Σ, I), there is an important result regarding the regularity of $[L(\mathcal{A})]_I$. This result uses the notion of *connected word*. A word $w \in \Sigma^\infty$ over (Σ, I) is called *connected* if the subgraph of the dependency graph induced by Σ_w is connected [10]. We denote such a subgraph with G_w, and call it the *dependency graph of* w. For example, the word $w = a(bc)^\omega$ over $\Sigma = \{a, b, c\}$ and $I = \{(b, c), (c, b)\}$, is connected, but its infinite suffix $(bc)^\omega$ is not. The automaton $\mathcal{A}$ is called *loop-connected* if every cycle in $\mathcal{A}$ produces a connected word. For any NBA $\mathcal{A}$ over (Σ, I), $[L(\mathcal{A})]_I$ is regular iff $\mathcal{A}$ is loop-connected [2].

2.2 Multi-buffer Simulation

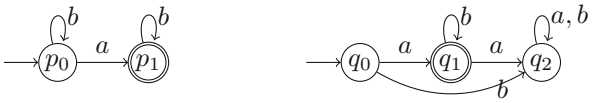
Given two NBA $\mathcal{A}$, $\mathcal{B}$ over (Σ, I), let $\hat{\Sigma} = (\Sigma_1, \ldots, \Sigma_n)$ be the distributed alphabet of (Σ, I), and $\kappa = (k_1, \ldots, k_n)$ a vector over $\mathbb{N} \cup \{\omega\}$, the *multi-buffer simulation game* $\mathcal{G}^{\kappa, \hat{\Sigma}}(\mathcal{A}, \mathcal{B})$, or simply $\mathcal{G}^{\kappa}(\mathcal{A}, \mathcal{B})$ is played between Spoiler and Duplicator in the automata $\mathcal{A}$, $\mathcal{B}$ with n buffers of capacity $k_1, \ldots, k_n$, and the buffers are associated with the alphabets $\Sigma_1, \ldots, \Sigma_n$, respectively. Initially, two pebbles are placed each on the initial states of $\mathcal{A}$ and $\mathcal{B}$. Spoiler moves the pebble in $\mathcal{A}$ by reading a letter $a \in \Sigma$, and pushes a copy of the a-symbol to each buffer i, in which $a \in \Sigma_i$. Duplicator either skips her turn or moves the pebble in $\mathcal{B}$ by reading a word $b_1 \ldots b_m$. While doing so, for every $i \in \{1, \ldots, m\}$, starting from $i = 1$, she pops b_i from each buffer that is associated with b_i. More formally, a configuration is a tuple $(p, \beta_1, \ldots, \beta_n, q) \in Q^{\mathcal{A}} \times \Sigma_1^* \times \ldots \times \Sigma_n^* \times Q^{\mathcal{B}}$, where $|\beta_i| \leq k_i$ for all $i \in \{1, \ldots, n\}$. The initial configuration is $(p_0, \epsilon, \ldots, \epsilon, q_0)$, where p_0, q_0 are the initial states of $\mathcal{A}$, $\mathcal{B}$, and in every configuration $(p, \beta_1, \ldots, \beta_n, q)$,

– Spoiler chooses a letter $a \in \Sigma$, a state $p' \in Q^{\mathcal{A}}$, such that $p \xrightarrow{a} p'$,

– Duplicator chooses a finite path $q \xrightarrow{b_1} q_1 \xrightarrow{b_2} q_2 \ldots \xrightarrow{b_m} q_m$ from q in $\mathcal{B}$, such that $\pi_i(a\beta_i) = \pi_i(\beta_i' b_1 \ldots b_m)$ for all $i \in \{1, \ldots, n\}$. The next configuration is $(p', \beta_1', \ldots, \beta_k', q')$.

If one of the players gets stuck, then the opponent wins, otherwise Spoiler and Duplicator respectively form infinite runs ρ in $\mathcal{A}$ and ρ' in $\mathcal{B}$. In this case, Duplicator wins iff ρ is not accepting or ρ' is accepting and every letter that is pushed by Spoiler into a buffer is eventually popped by Duplicator. We write $\mathcal{A} \sqsubseteq^{\kappa} \mathcal{B}$ if Duplicator wins $\mathcal{G}^{\kappa}(\mathcal{A}, \mathcal{B})$, and in this case it implies $L(\mathcal{A}) \subseteq [L(\mathcal{B})]_I$.

Example 1. Consider the following two NBA $\mathcal{A}$, $\mathcal{B}$ over the independence alphabet (Σ, I), in which $\hat{\Sigma} = (\{a\}, \{b\})$, i.e. $\Sigma = \{a, b\}$, $I = \{(a, b), (b, a)\}$.



We have $\mathcal{A} \sqsubseteq^{0,\omega} \mathcal{B}$, since Duplicator has the following winning strategy in $\mathcal{G}^{0,\omega}(\mathcal{A}, \mathcal{B})$: she skips her moves, except when Spoiler reads a. In this case, Duplicator goes to q_1: she pops all the bs from the second buffer, and a from the first buffer. From this state, if Spoiler reads b then Duplicator also reads b by looping in q_1 and pops b from the buffer. Duplicator wins since either Spoiler forms a non-accepting run, or Duplicator forms an accepting run and every letter that is pushed by Spoiler into a buffer is eventually popped by Duplicator. Duplicator however loses the game $\mathcal{G}^{0,k}(\mathcal{A}, \mathcal{B})$ for any $k \in \mathbb{N}$, since Spoiler can loop in p_0 indefinitely and push unboundedly many b before he goes to p_1. In this case, Duplicator eventually violates the buffer constraint.

3 Topological Characterisation

Given two NBA $\mathcal{A}$, $\mathcal{B}$, and a function $f : R_1 \to R_2$, $R_1 \subseteq \mathsf{Run}(\mathcal{A})$, $R_2 \subseteq \mathsf{Run}(\mathcal{B})$, let us call f *trace preserving* if for all $\rho \in \mathsf{Dom}(f)$, $\mathsf{word}(\rho) \sim_I \mathsf{word}(f(\rho))$. Trace closure inclusion $L(\mathcal{A}) \subseteq [L(\mathcal{B})]_I$ can be characterised with a trace preserving $f : \mathsf{AccRun}(\mathcal{A}) \to \mathsf{AccRun}(\mathcal{B})$. This is because such a function f exists iff for every $\rho \in \mathsf{AccRun}(\mathcal{A})$, there exists $\rho' \in \mathsf{AccRun}(\mathcal{B})$ over trace equivalent words.

Proposition 1. *$L(\mathcal{A}) \subseteq [L(\mathcal{B})]_I$ iff there exists a trace preserving function $f : \mathsf{AccRun}(\mathcal{A}) \to \mathsf{AccRun}(\mathcal{B})$.*

For such a function $f : R_1 \to R_2$, we can define its continuity by considering the standard metric for infinite words. This is because every run $\rho \in \mathsf{Run}(\mathcal{A})$ can be seen as an infinite word over $\Sigma' = Q^{\mathcal{A}} \cdot \Sigma$. We consider the metric $d : \mathsf{AccRun}(\mathcal{A})^2 \to [0, 1]$, where $d(\rho, \rho') = 0$ if $\rho = \rho'$, and

$d(\rho, \rho') = 2^{-min\{i \mid p_i a_i \neq q_i b_i\}}$ if $\rho = p_0 a_0\, p_1 a_1 \ldots$, $\rho' = q_0 b_0\, q_1 b_1 \ldots$ are different. Intuitively, the distance between two runs is small if they share a long common prefix.

In [9], it is shown that we can refine the characterisation in Proposition 1 for multi-buffer simulation $\mathcal{A} \sqsubseteq^{\omega,\ldots,\omega} \mathcal{B}$ by requiring the function f to be continuous. Recall that f is continuous if for any two distinct runs $\rho, \rho' \in \mathsf{Dom}(f)$ that are very close, they are mapped into two runs $f(\rho)$, $f(\rho')$ that are also very close.

Proposition 2 [9]. *$\mathcal{A} \sqsubseteq^{\omega,\ldots,\omega} \mathcal{B}$ iff there exists a continuous trace preserving function $f : \mathsf{AccRun}(\mathcal{A}) \to \mathsf{AccRun}(\mathcal{B})$.*

Consider again the NBA $\mathcal{A}$, $\mathcal{B}$ from Example 1. We have a continuous trace preserving function $f : \mathsf{AccRun}(\mathcal{A}) \to \mathsf{AccRun}(\mathcal{B})$ that maps every accepting run of $\mathcal{A}$, i.e. over b^*ab^ω, to the one of $\mathcal{B}$ over ab^ω. This function is trace preserving since for every $n \geq 0$, $b^n ab^\omega \sim_I ab^\omega$. It is also continuous since there is only one accepting run in $\mathcal{B}$, therefore the distance between two outputs of f is always 0, i.e. trivially very small.

Such a characterisation of winning strategies with continuous functions is far from new. For example, in the *delay game* [8], it is shown that the winning strategy for Duplicator can be characterised with a continuous function, and in the case of finite delay, the characterisation can be lifted to the one that consider a Lipschitz continuous function. Recall that a function is Lipschitz continuous if there exists a constant $C \in \mathbb{R}$, such that for any two inputs of distance d, their outputs' distance is at most $C \cdot d$.

We would like to have such a topological characterisation for multi-buffer simulation. The characterisation with a continuous function holds for multi-buffer simulation as we can see in Proposition 2. However, the characterisation with a Lipschitz continuous function fails.

Example 2. Consider the following two automata $\mathcal{A}, \mathcal{B}$ over the independence alphabet (Σ, I) with $\hat{\Sigma} = (\{a, b\})$, i.e. $\Sigma = \{a, b\}$ and $I = \emptyset$,

In this case, we have a Lipschitz continuous and trace preserving function $f : \mathsf{AccRun}(\mathcal{A}) \to \mathsf{AccRun}(\mathcal{B})$ that maps the only accepting run of $\mathcal{A}$ to the one of $\mathcal{B}$. This function is trace preserving and also Lipschitz continuous with constant 0. However, Spoiler wins the game $\mathcal{G}^k(\mathcal{A}, \mathcal{B})$ for any $k \in \mathbb{N}$. He wins by playing the word b^ω. For every $k \in \mathbb{N}$, Duplicator eventually fills the buffer more than its capacity in round $k + 1$, and loses the game $\mathcal{G}^k(\mathcal{A}, \mathcal{B})$.

Example 3. Consider again the NBA $\mathcal{A}$, $\mathcal{B}$ from Example 1. We have a trace preserving and continuous function $f : \mathsf{AccRun}(\mathcal{A}) \to \mathsf{AccRun}(\mathcal{B})$ as shown before. It is also Lipschitz continuous with Lipschitz constant 0. However, Spoiler wins the game $\mathcal{G}^{k,k}(\mathcal{A}, \mathcal{B})$ for all $k \in \mathbb{N}$. He wins by first reading $bbb\ldots$ indefinitely. Duplicator either skips her move forever, or eventually moves by reading b. If Duplicator eventually moves by reading b, she would never form an accepting

run, and Spoiler can continue read ab^ω and form an accepting run. However if Duplicator never moves, then she violates the buffer constraint in round $k+1$. Hence in both cases Duplicator loses.

We will show that there are some restricted classes of $\mathcal{A}$, $\mathcal{B}$ where we can lift the characterisation in Proposition 2 to the case of bounded buffers by considering a Lipschitz continuous function.

4 Characterisation of $\sqsubseteq^{k,\ldots,k}$, $k \in \mathbb{N}$

First note that if multi-buffer simulation $\mathcal{A} \sqsubseteq^{k,\ldots,k} \mathcal{B}$ holds with some bounded capacity $k \in \mathbb{N}$, then we can construct a Lipschitz continuous trace preserving function $f : \mathsf{AccRun}(\mathcal{A}) \to \mathsf{AccRun}(\mathcal{B})$. For every $\rho \in \mathsf{AccRun}(\mathcal{A})$, we define $f(\rho)$ as the run that is formed by Duplicator in $\mathcal{G}^{k,\ldots,k}(\mathcal{A},\mathcal{B})$, assuming that Spoiler plays ρ and Duplicator plays according to the winning strategy. Such a function is trace preserving since it is derived from a winning strategy of Duplicator, and it is Lipschitz continuous with Lipschitz constant $C = k + \ldots + k$ since for any output run $f(\rho)$ the i-th letter of $f(\rho)$ is determined by the first $C + i$ letters of ρ.

Lemma 1. *If $\mathcal{A} \sqsubseteq^{k,\ldots,k} \mathcal{B}$ for some $k \in \mathbb{N}$, then there exists a Lipschitz continuous trace preserving function $f : \mathsf{AccRun}(\mathcal{A}) \to \mathsf{AccRun}(\mathcal{B})$.*

Proof. For every $\rho \in \mathsf{AccRun}(\mathcal{A})$, we define $f(\rho)$ as the run that is formed by Duplicator in $\mathcal{G}^{k,\ldots,k}(\mathcal{A},\mathcal{B})$, assuming that Spoiler plays ρ and Duplicator plays according to the winning strategy. The function f is trace preserving since it is derived from a winning strategy of Duplicator.

Let $n \in \mathbb{N}$ be some number and $C = k + \ldots + k$. If Spoiler plays $\rho \in \mathsf{AccRun}(\mathcal{A})$, then since Duplicator wins $\mathcal{G}^{k,\ldots,k}(\mathcal{A},\mathcal{B})$, in round $n + C$, Duplicator forms a finite run of length at least n. If there is $\rho' \in \mathsf{AccRun}(\mathcal{A})$ with $d(\rho,\rho') \leq 2^{-(n+C+1)}$, then in the first $n + C$ rounds, Duplicator does not see any difference whether Spoiler actually plays ρ or ρ'. Duplicator makes the same moves in response to ρ or ρ'. The output runs $f(\rho)$ and $f(\rho')$ share the same prefix of length n, i.e. $d(f(\rho), f(\rho')) \leq 2^{-(n+1)}$. This implies $d(f(\rho), f(\rho')) \leq 2^C \cdot d(\rho,\rho')$. The function f is Lipschitz continuous with Lipschitz constant 2^C.

As we can see in the previous section, the reverse direction of this lemma does not hold. In Example 2, the reason why Duplicator loses is because Spoiler can play a non-accepting run that cannot be mimicked by Duplicator. We can easily avoid this by assuming that Duplicator's automaton is complete: for every $q \in Q^{\mathcal{B}}$ and $a \in \Sigma$, there is $q' \in Q^{\mathcal{B}}$ such that $(q,a,q') \in E^{\mathcal{B}}$.

In Example 3, the reason why Duplicator loses is different. The automaton of Duplicator is complete. But in this case, Spoiler can produce b^*ab^ω, in which the letter a can commute unboundedly to the left or right. The letter a can be read by Spoiler in a very late round, but has to be read by Duplicator

in an early round. In order to read its trace equivalent word: ab^ω, DUPLICATOR first needs to store indefinitely many bs that are read by SPOILER. To avoid this, we need to restrict words that are produced by SPOILER. He should only produce words, in which each letter cannot commute unboundedly to the left or right. To formalise this restriction we introduce the notion of *corresponding relation*.

For any two words $w, v \in \Sigma^\infty$, the corresponding relation $\mathsf{Corr}_{w,v}$ relates the position of w and v that are over the same letter and have the same order with respect to the letter.

Definition 1. *For any $w, v \in \Sigma^\infty$, the corresponding relation $\mathsf{Corr}_{w,v} \subseteq \mathsf{Pos}(w) \times \mathsf{Pos}(v)$ is defined as $\mathsf{Corr}_{w,v} = \{(i,j) \mid \exists a \in \Sigma,\ w(i) = v(j) = a,\ |w(1)\dots w(i)|_a = |v(1)\dots v(j)|_a\}$.*

Consider (Σ, I) where $\Sigma = \{a,b,c\}$, $I = \{(b,c), (c,b)\}$. $w = a(bc)^\omega$ and $v = a(cb)^\omega$. We have $\mathsf{Corr}_{w,v} = \{(1,1)\} \cup \{(i, i+1) | \, i > 1 \text{ is even}\} \cup \{(i, i-1) | \, i > 1 \text{ is odd}\}$.

If w, v are two words over (Σ, I) and $w \sim_I v$, then $\mathsf{Corr}_{w,v}$ is a bijection. This is because for every $a \in \Sigma$, the number of a in w and v are the same. We will use the relation $\mathsf{Corr}_{w,v}$, in which $w \sim_I v$, to determine how long a letter in w or v can commute.

Definition 2. *Given a word $w \in \Sigma^\infty$ over an independence alphabet (Σ, I), and $i \in \mathsf{Pos}(w)$, let $S_w(i) = \{j - i \mid (i,j) \in \mathsf{Corr}_{w,v},\ w \sim_I v\}$. We define $\mathsf{Deg}^+_w(i) = \mathsf{max}\{k \in \mathbb{N} \cup \{\infty\} : k \in S_w(i)\}$ and $\mathsf{Deg}^-_w(i) = \mathsf{max}\{k \in \mathbb{N} \cup \{\infty\} : -k \in S_w(i)\}$. The* corresponding degree *of a letter at position i in w is $\mathsf{Deg}_w(i) = \mathsf{max}\{\mathsf{Deg}^+_w(i), \mathsf{Deg}^-_w(i)\}$.*

Consider the word $w = a(bc)^\omega$ over $\Sigma = \{a,b,c\}$, $I = \{(b,c), (c,b)\}$. We have $\mathsf{Deg}_w(1) = 0$ since for all $v \sim_I w$, $(1,1) \in \mathsf{Corr}_{w,v}$. We have $\mathsf{Deg}_w(2) = \infty$ since for all $k \in \mathbb{N}$, there is $v_k = ac^{k+1}(bc)^\omega$, such that $v_k \sim_I w$ and $(2, k+3) \in \mathsf{Corr}_{w,v_k}$. This implies $\mathsf{Deg}^+_w(2) > k$ for all $k \in \mathbb{N}$, and hence $\mathsf{Deg}^+_w(2) = \infty$.

The corresponding degree $\mathsf{Deg}_w(i)$ tells us the maximum length of how long the letter at position i in w can commute. Moreover, for a set of words $L \in \Sigma^\infty$, we define the corresponding degree of L as $\mathsf{Deg}(L) = \mathsf{max}\{k \in \mathbb{N} \cup \{\infty\} \mid w \in L, i \in \mathsf{Pos}(w), k \in \mathsf{Deg}_w(i)\}$. If $\mathsf{Deg}(L) = \infty$, then for any $k \in \mathbb{N}$, there is a word w_k in L such that one letter of w can commute more than k steps to the left or right. For example, consider the automaton $\mathcal{A}$ in Example 1. We have $\mathsf{Deg}(L(\mathcal{A})) = \infty$ since there is $w = ab^\omega \in L(\mathcal{A})$ and its first letter can commute more than k steps to the right for any $k \in \mathbb{N}$.

For any NBA $\mathcal{A}$ over (Σ, I), let $\mathsf{Tr}(\mathcal{A})$ be the set of words over a finite or infinite path of $\mathcal{A}$, i.e. $\mathsf{Tr}(\mathcal{A}) = \{a_1 a_2 \dots \in \Sigma^\infty \mid \exists p_1, p_2, \dots \in Q^{\mathcal{A}} : (p_1, a_1, p_2), (p_2, a_2, p_3), \dots \in E^{\mathcal{A}}\}$. We will show that for any two NBA $\mathcal{A}$, $\mathcal{B}$, if the corresponding degree of $\mathsf{Tr}(\mathcal{A})$ is finite and $\mathcal{B}$ is complete, then the reverse direction of Lemma 1 also holds. We will show this by using the *delay* game from [8] as an intermediate game. The delay game is similar to the multi-buffer simulation game. However the winning condition is given by a function $f : R_1 \to R_2$, where $R_1 \subseteq \mathsf{Run}(\mathcal{A})$ and $R_2 \subseteq \mathsf{Run}(\mathcal{B})$. In this game, DUPLICATOR can read any letter freely, even the one that is not yet read by SPOILER.

A *delay game* $\Gamma^k(\mathcal{A}, \mathcal{B}, f)$ is played between Spoiler and Duplicator in $\mathcal{A}$, $\mathcal{B}$ in which the configuration is a pair $(r_\mathcal{A}, r_\mathcal{B})$ of finite runs of $\mathcal{A}$, $\mathcal{B}$ with $0 \leq |r_\mathcal{A}| - |r_\mathcal{B}| \leq k$. The initial configuration is the pair (p_I, q_I) of the initial states of $\mathcal{A}$, $\mathcal{B}$. In every round $i > 0$ with a configuration $(r_\mathcal{A}, r_\mathcal{B})$,

- Spoiler extends $r_\mathcal{A}$ to some finite run $r'_\mathcal{A} := r_\mathcal{A} ap$ in $\mathcal{A}$, and
- Duplicator extends $r_\mathcal{B}$ to some finite run $r'_\mathcal{B} := r_\mathcal{B} b_1 q_1 b_2 \dots b_n q_n$ in $\mathcal{B}$.

The next configuration is $(r'_\mathcal{A}, r'_\mathcal{B})$. If one of the players gets stuck, then the opponent wins. Otherwise, the play goes on infinitely many rounds and produces two infinite runs ρ, ρ' of $\mathcal{A}$, $\mathcal{B}$, respectively. Duplicator wins iff whenever $\rho \in \mathsf{Dom}(f)$ then $\rho' = f(\rho)$.

Example 4. Consider the automata $\mathcal{A}$, $\mathcal{B}$ from Example 1. Let $f : \mathsf{AccRun}(\mathcal{A}) \to \mathsf{AccRun}(\mathcal{B})$ be a trace preserving function. In this case, there is such a unique f, i.e. $f(\rho) = q_0 a (q_1 b)^\omega$ for all $\rho \in \mathsf{AccRun}(\mathcal{A})$. Duplicator wins the delay game $\Gamma^0(\mathcal{A}, \mathcal{B}, f)$ with the following winning strategy: first she reads a, then she reads $bbb\dots$ for the rest of the play. Since Duplicator always forms the image of Spoiler's run without any delay, she wins $\Gamma^0(\mathcal{A}, \mathcal{B}, f)$.

The existence of winning strategy for the Duplicator in the delay game with finite delay basically corresponds to the Lipschitz continuity of the function that defines the winning condition.

Lemma 2. *For any two NBA $\mathcal{A}$, $\mathcal{B}$ in which $\mathcal{B}$ is complete,* Duplicator *wins $\Gamma^C(\mathcal{A}, \mathcal{B}, f)$ iff f is Lipschitz continuous with constant 2^C.*

Proof ($\Leftarrow$). Consider the following winning strategy for Duplicator. Suppose we are at configuration $(r_\mathcal{A}, r_\mathcal{B})$ and Spoiler extends his run to $r'_\mathcal{A} := r_\mathcal{A} ap$. If $r'_\mathcal{A}$ cannot be extended to any $\rho \in \mathsf{Dom}(f)$, then Duplicator extends her run to $r'_\mathcal{B}$, such that $\mathsf{word}(r'_\mathcal{A}) \sim_I \mathsf{word}(r'_B)$. This is possible since $\mathcal{B}$ is complete. If $r'_\mathcal{A}$ can be extended to some $\rho \in \mathsf{Dom}(f)$ and there exists $r'_\mathcal{B} := r_\mathcal{B} b_1 q_1 \dots b_n q_n$, such that for every such a run ρ, $f(\rho)$ is started with $r'_\mathcal{B}$, then Duplicator extends her run to such a maximal $r'_\mathcal{B}$. Otherwise, Duplicator skips her turn.

If Duplicator plays according to this strategy, then there is no round with a configuration $(r_\mathcal{A}, r_\mathcal{B})$, in which $|r_\mathcal{A}| - |r_\mathcal{B}| > C$. Suppose there is such a round m. Duplicator does not extend her run to some run longer than $r_\mathcal{B}$ in round m, because there exist two runs $\rho_1, \rho_2 \in \mathsf{Dom}(f)$ that can be extended from $r_\mathcal{A}$, i.e. $d(\rho_1, \rho_2) \leq 2^{-(|r_\mathcal{A}|+1)}$, and both $f(\rho_1), f(\rho_2)$ can be extended from $r_\mathcal{B}$, but not from any run longer than $r_\mathcal{B}$, i.e. $d(f(\rho_1), f(\rho_2)) = 2^{-(|r_\mathcal{B}|+1)}$. Since $|r_\mathcal{A}| - |r_\mathcal{B}| > C$, we have $d(f(\rho_1), f(\rho_2)) > 2^C \cdot d(\rho_1, \rho_2)$. This contradicts that f is Lipschitz continuous with constant 2^C.

Since in every round the length difference between Spoiler and Duplicator's runs is at most $C \in \mathbb{N}$, then if Spoiler forms an infinite run, Duplicator also forms an infinite run. Moreover, since the invariant holds that in any round i with a configuration $(r^{(i)}_\mathcal{A}, r^{(i)}_\mathcal{B})$, if ρ is started with $r^{(i)}_\mathcal{A}$ then ρ' is started with $r^{(i)}_\mathcal{B}$, then whenever Spoiler plays $\rho \in \mathsf{Dom}(f)$, Duplicator forms the f-image of ρ, i.e. $\rho' = f(\rho)$. Duplicator wins $\Gamma^C(\mathcal{A}, \mathcal{B}, f)$.

($\Rightarrow$) Let $k \in \mathbb{N}$ be some number. If f is not Lipschitz continuous, then there exist $\rho_1, \rho_2 \in \mathsf{AccRun}(\mathcal{A})$, such that $d(f(\rho_1), f(\rho_2)) > 2^k \cdot d(\rho_1, \rho_2)$. Otherwise, f is Lipschitz continuous and k is a Lipschitz constant of f. Let $n \in \mathbb{N}$, such that $2^{-n} = d(\rho_1, \rho_2)$. The winning strategy for Spoiler is to first play ρ_1. On round $n-k+1$, if Duplicator forms a finite run r' that is not a prefix of $f(\rho_1)$, then Spoiler keeps playing ρ_1 for the rest of the play. Otherwise, he continues by playing ρ_2. This is possible, since ρ_1, ρ_2 share the same prefix of length n. In the first case, Spoiler wins because Duplicator does not form the f-image of ρ_1. In the second case, since $d(f(\rho_1), f(\rho_2)) > 2^{k-n}$, i.e. $f(\rho_1), f(\rho_2)$ share the same prefix of length less than $n-k$, and $|r'| > n-k$, so r' is not a prefix of $f(\rho_0)$. In this case, Duplicator does not form the f-image of ρ_2.

Unfortunately, the winning strategy for Duplicator in the delay game may not be suitable for the buffer game. Consider Duplicator's winning strategy in Example 4. It is not winning in $\mathcal{G}^{\omega,\omega}(\mathcal{A}, \mathcal{B})$, because in the first round, Spoiler might not read a, and hence Duplicator cannot pop a from the buffer. In the multi-buffer game over $\hat{\Sigma} = (\Sigma_1, \ldots, \Sigma_n)$, if w, w' are the words produced by Spoiler and Duplicator in some round, then $\pi_i(w)$ is a prefix of $\pi_i(w')$ for all $i \in \{1, \ldots, n\}$. This is not always the case in the delay game. Duplicator can read letters that are not yet read by Spoiler. We need to translate the winning strategy from the delay to the buffer game. Duplicator should just take the longest output in the delay game that is allowed in the buffer game.

Lemma 3. *Let $f : \mathsf{AccRun}(\mathcal{A}) \to \mathsf{AccRun}(\mathcal{B})$ be a trace preserving function for two NBA $\mathcal{A}$, $\mathcal{B}$, in which $\mathcal{A}$ is complete. If* Duplicator *wins $\Gamma^k(\mathcal{A}, \mathcal{B}, f)$ for some $k \in \mathbb{N}$, then she wins $\mathcal{G}^{\omega,\ldots,\omega}(\mathcal{A}, \mathcal{B})$.*

Proof. Suppose the NBA $\mathcal{A}$, $\mathcal{B}$ are over (Σ, I) with distributed alphabet $\hat{\Sigma} = (\Sigma_1, \ldots, \Sigma_n)$. The translation is as follows. Suppose in $\mathcal{G}^{\omega,\ldots,\omega}(\mathcal{A}, \mathcal{B})$, Spoiler and Duplicator form finite runs $r_\mathcal{A}$ and $r_\mathcal{B}$. Let $w = \mathsf{word}(r_\mathcal{A})$ and $w' = \mathsf{word}(r_\mathcal{B})$. If the strategy in the delay game tells Duplicator to extend $r_\mathcal{B}$ to $r'_\mathcal{B} := r_\mathcal{B} b_1 p_1 \ldots b_m p_m$, $m \geq 0$, then in the buffer game, we extend $r_\mathcal{B}$ to $r_\mathcal{B} b_1 q_1 \ldots b_{m'} q_{m'}$, $m' \leq m$, the maximal prefix of $r'_\mathcal{B}$, such that $\pi_i(w' b_1 \ldots b_{m'})$ is a prefix of $\pi_i(w)$ for all $i \in \{1, \ldots, n\}$.

Let ρ, ρ_1 be the accepting runs that are formed by Spoiler and Duplicator in $\Gamma^k(\mathcal{A}, \mathcal{B}, f)$, and ρ, ρ_2 be the runs that are formed in $\mathcal{G}^{\omega,\ldots,\omega}(\mathcal{A}, \mathcal{B})$ according to the translation. Since we always extend Duplicator's run in the buffer game by taking the prefix of the original extension in the delay game, any finite prefix of ρ_2 is also a prefix of ρ_1. The converse also holds: any finite prefix of ρ_1 is a prefix of ρ_2. Suppose $r_\mathcal{B}$ is a finite prefix of ρ_1. There is $r_\mathcal{A}$, a finite prefix of ρ, such that in the delay game, when Spoiler extends his run to $r_\mathcal{A}$, Duplicator extends her run to $r_\mathcal{B}$. Let $w = \mathsf{word}(r_\mathcal{A})$ and $w' = \mathsf{word}(r_\mathcal{B})$. Since f is trace preserving, there is $r'_\mathcal{A} := r_\mathcal{A} a_1 p_1 \ldots a_k p_k$, $k \geq 0$, a finite prefix of ρ, such that $\pi_i(w')$ is a prefix of $\pi_i(w a_1 \ldots a_k)$ for all $i \in \{1, \ldots, n\}$. Hence when Spoiler forms $r'_\mathcal{A}$, Duplicator extends her run to $r_\mathcal{B}$ in the delay game. This implies

that $r_{\mathcal{B}}$ is also a prefix of ρ_2. We have $\rho_1 = \rho_2$. Thus, whenever Spoiler plays an accepting run ρ in $\mathcal{G}^{\omega,\ldots,\omega}(\mathcal{A},\mathcal{B})$, then Duplicator forms an accepting run $\rho' = f(\rho)$.

If we additionally assume that the corresponding degree of $\mathsf{Tr}(\mathcal{A})$ is finite, then we can show that Duplicator also wins the buffer game with some bounded capacity. Such an assumption is needed to avoid the case where Spoiler produces a word in which one of its letter commutes unboundedly, and makes Duplicator store unboundedly many irrelevant letters before she reads the corresponding one, as we exemplify in Example 3.

Lemma 4. *Let $f : \mathsf{AccRun}(\mathcal{A}) \to \mathsf{AccRun}(\mathcal{B})$ be a trace preserving function for two NBA $\mathcal{A}$, $\mathcal{B}$, in which the corresponding degree of $\mathsf{Tr}(\mathcal{A})$ is finite, and $\mathcal{B}$ is complete. If* Duplicator *wins $\Gamma^k(\mathcal{A},\mathcal{B},f)$ for some $k \in \mathbb{N}$, then she wins $\mathcal{G}^{\kappa}(\mathcal{A},\mathcal{B})$ for some $\kappa \in \mathbb{N}^*$.*

Proof. Let $k' \in \mathbb{N}$ be the corresponding degree of $\mathsf{Tr}(\mathcal{A})$, i.e. $k' = \mathsf{Deg}(\mathsf{Tr}(\mathcal{A}))$. Suppose Duplicator wins for some $k \in \mathbb{N}$. By Lemma 2 we can assume that Duplicator wins $\Gamma^k(\mathcal{A},\mathcal{B},f)$ with the winning strategy as defined before. Consider the translation in which Duplicator output the maximal prefix that is allowed in the buffer game. We will show that the translated winning strategy in Lemma 3 is not only winning in $\mathcal{G}^{\omega,\ldots,\omega}(\mathcal{A},\mathcal{B})$, but also in $\mathcal{G}^{k'+k,\ldots,k'+k}(\mathcal{A},\mathcal{B})$. We will show this by contradiction. Suppose while playing in $\mathcal{G}^{\omega,\ldots,\omega}(\mathcal{A},\mathcal{B})$, there exists a round, such that one of the buffers is filled with $k+k'+1$ letters. Let $r_{\mathcal{A}}, r_{\mathcal{B}}$ be the runs that are formed by Spoiler and Duplicator in this round. We have $|r_{\mathcal{A}}| - |r_{\mathcal{B}}| > k + k'$. Duplicator does not extend her run longer than $r_{\mathcal{B}}$, because either the winning strategy in $\Gamma^k(\mathcal{A},\mathcal{B},f)$ tells her to extend to $r_{\mathcal{B}}$, or it actually tells her to extend to some run $r'_{\mathcal{B}}$ longer than $r_{\mathcal{B}}$, but $r_{\mathcal{B}}$ is the maximal prefix of $r'_{\mathcal{B}}$ that satisfies

$$\pi_i(\mathsf{word}(r_{\mathcal{B}})) \text{ is a prefix of } \pi_i(\mathsf{word}(r_{\mathcal{A}})), \tag{1}$$

for all $i \in \{1,\ldots,n\}$. In the first case, since it implies $|r_{\mathcal{A}}| - |r_{\mathcal{B}}| > k$, this contradicts the strategy is winning in $\Gamma^k(\mathcal{A},\mathcal{B},f)$. In the second case, suppose $r'_{\mathcal{B}}$ is extended from $r_{\mathcal{B}}$ by reading u, i.e. $r'_{\mathcal{B}} = r_{\mathcal{B}}u(1)p_1 \ldots u(\ell)p_\ell$, $\ell > 0$, and suppose $u(1) = a$. Since (1) holds, we have $|\mathsf{word}(r_{\mathcal{B}})|_a \leq |\mathsf{word}(r_{\mathcal{A}})|_a$. However since $r_{\mathcal{B}}$ is the maximal prefix of $r'_{\mathcal{B}}$ that satisfies (1), we have $|\mathsf{word}(r_{\mathcal{B}})|_a = |\mathsf{word}(r_{\mathcal{A}})|_a$, since otherwise $r_{\mathcal{B}}ap_1$ also satisfies (1) and contradicts the maximality of $r_{\mathcal{B}}$. Let w, w' be the words that are produced respectively by Spoiler and Duplicator in $\mathcal{G}^{\omega,\ldots,\omega}(\mathcal{A},\mathcal{B})$. Hence $\mathsf{word}(r_{\mathcal{A}})$ and $\mathsf{word}(r'_{\mathcal{B}})$ are prefixes of w and w', respectively. Since we assume f is trace preserving and Duplicator plays according to the winning strategy in Lemma 2, we have $w \sim_I w'$. Let $n_0 = |\mathsf{word}(r_{\mathcal{A}})|_a = |\mathsf{word}(r_{\mathcal{B}})|_a$, $i_0 \in \mathsf{Pos}(w)$, and $i_1 \in \mathsf{Pos}(w')$, such that $w(i_0) = w'(i_1) = a$ and $|w(1)\ldots w(i_0)|_a = |w'(1)\ldots w'(i_1)|_a = n_0 + 1$. Hence $i_0 > |r_{\mathcal{A}}|$ and $i_1 = |r_{\mathcal{B}}| + 1$. This implies $i_0 - i_1 > k'$ since $|r_{\mathcal{A}}| - |r_{\mathcal{B}}| > k'$. Since $(i_0, i_1) \in \mathsf{Corr}_{w,w'}$ and $w \in \mathsf{Tr}(\mathcal{A})$, this contradicts $\mathsf{Deg}(\mathsf{Tr}(\mathcal{A})) = k'$.

Hence if the corresponding degree of $\mathsf{Tr}(\mathcal{A})$ is finite and $\mathcal{B}$ is complete, the existence of a Lipschitz continuous trace preserving function implies that multi-buffer simulation holds for some bounded buffers. Together with Lemma 1, we have the following characterisation.

Theorem 1. *Let $\mathcal{A}$, $\mathcal{B}$ be two NBA, in which the corresponding degree of $\mathsf{Tr}(\mathcal{A})$ is finite and $\mathcal{B}$ is complete. $\mathcal{A} \sqsubseteq^{k,\dots,k} \mathcal{B}$ for some $k \in \mathbb{N}$ iff there exists a Lipschitz continuous trace preserving function $f : \mathsf{AccRun}(\mathcal{A}) \to \mathsf{AccRun}(\mathcal{B})$.*

5 Cyclic-Path-Connected Automata

Recall that an automaton $\mathcal{A}$ is loop-connected if every cycle in $\mathcal{A}$ produces a connected word. If $\mathcal{A}$ is not loop-connected, then the corresponding degree of $\mathsf{Tr}(\mathcal{A})$ is not finite. For example, consider the automaton $\mathcal{A}$ with two states q_0, q_1, over $\Sigma = \{a, b\}$ and $I = \{(a, b), (b, a)\}$, i.e. a, b is independent with each other. Suppose $E^{\mathcal{A}} = \{(q_0, a, q_1), (q_1, b, q_0)\}$. Hence, $\mathcal{A}$ is not loop-connected since ab is not connected. The corresponding degree of $\mathsf{Tr}(\mathcal{A})$ is also not finite since for every $k \in \mathbb{N}$, we can consider the word $(ab)^{k+1} \in \mathsf{Tr}(\mathcal{A})$, in which its first letter can commute more than k steps to the right. Hence $\mathsf{Deg}(\mathsf{Tr}(\mathcal{A})) = \infty$.

Lemma 5. *If the corresponding degree of $\mathsf{Tr}(\mathcal{A})$ is finite, then $\mathcal{A}$ is loop-connected.*

Proof. Suppose $\mathcal{A}$ is not loop-connected. There exists a cycle c in $\mathcal{A}$ over a non-connected word w. For any $k \in \mathbb{N}$, let $v = w^{k+1}$. Since w is not connected, there exists $\langle \Sigma_1, \Sigma_2 \rangle$ a partition of Σ_w, such that the dependency graph G_w consists of two non-connected components Σ_1 and Σ_2. Every letter in Σ_1 and Σ_2 commutes with each other, i.e. $w \sim_I \pi_1(w)\pi_2(w) \sim_I \pi_2(w)\pi_1(w)$, where π_i is the projection to Σ_i for $i \in \{1, 2\}$. This implies $v \sim_I \pi_1(w)^{k+1}\pi_2(w)^{k+1} \sim_I \pi_2(w)^{k+1}\pi_1(w)^{k+1}$. Let $\mathsf{b} \in \{1, 2\}$, such that $v(1) \in \Sigma_{\mathsf{b}}$. Let $v' = \pi_{\overline{\mathsf{b}}}(w)^{k+1}\pi_{\mathsf{b}}(w)^{k+1}$ and $n = |\pi_{\overline{\mathsf{b}}}(w)^{k+1}|$. We have $n > k$ since at least one letter of w belongs to $\Sigma_{\overline{\mathsf{b}}}$. The first letter of v corresponds to the $(n+1)$-th letter of v', i.e. $(1, n+1) \in \mathsf{Corr}_{v,v'}$. Hence for every $k \in \mathbb{N}$, there exists $v \in \mathsf{Tr}(\mathcal{A})$, such that $\mathsf{Deg}^+_v(1) > k$. This implies $\mathsf{Deg}^+_v(1) = \infty$, and hence $\mathsf{Deg}(\mathsf{Tr}(\mathcal{A})) = \infty$.

The converse of this lemma, however, does not hold. Consider the automaton $\mathcal{A}$ from Example 1. It is loop-connected, but we have seen that the corresponding degree of $\mathsf{Tr}(\mathcal{A})$ is not finite. We will show that we can characterise $\mathcal{A}$, in which the corresponding degree of $\mathsf{Tr}(\mathcal{A})$ is finite, by considering a more restrictive condition than loop-connected. Instead of only for the cycle, we require every path in $\mathcal{A}$ that contains a cycle to produce a connected word. We call such an automaton *cyclic-path-connected*.

Definition 3. *An automaton $\mathcal{A}$ over (Σ, I) is cyclic-path-connected if for every path $p \xrightarrow{u} q \xrightarrow{v} q \xrightarrow{w} r$, where $u, w \in \Sigma^*$ and $v \in \Sigma^+$, the word uvw is connected.*

Note that the automaton $\mathcal{A}$ in Example 1 is loop-connected, but not cyclic-path-connected. The word ba is over a cyclic-path of $\mathcal{A}$ but not connected with respect to the given independence alphabet. Note that the corresponding degree of $\mathsf{Tr}(\mathcal{A})$ is also not finite. This is because for every $k \in \mathbb{N}$ we can consider the word $b^{k+1}a \in \mathsf{Tr}(\mathcal{A})$. The last letter of such a word can commute more than k steps to the left, and hence $\mathsf{Deg}(\mathsf{Tr}(\mathcal{A})) = \infty$. This actually also holds in general.

Theorem 2. *If the corresponding degree of* $\mathsf{Tr}(\mathcal{A})$ *is finite, then* $\mathcal{A}$ *is cyclic-path-connected.*

Proof. Suppose $\mathcal{A}$ is not cyclic-path-connected. If $\mathcal{A}$ is also not loop-connected, then by Lemma 5, the corresponding degree of $\mathsf{Tr}(\mathcal{A})$ is not finite. Suppose $\mathcal{A}$ is loop-connected but not cyclic-path-connected. There exists a cyclic-path r over a non-connected word w. Without loss of generality, let $w = w'w''$, where w' is produced by a cycle. For any $k \in \mathbb{N}$, let $v = w'^{k+1}w''$. Since w is not connected, there exists $\langle \Sigma_1, \Sigma_2 \rangle$ a non-empty partition of Σ_w, such that $w \sim_I \pi_1(w)\pi_2(w) \sim_I \pi_2(w)\pi_1(w)$. This implies $v \sim_I w'^k\pi_1(w)\pi_2(w) \sim_I w'^k\pi_2(w)\ \pi_1(w)$. Let $\mathsf{b} \in \{1, 2\}$, such that $\Sigma_{w'} \subseteq \Sigma_{\mathsf{b}}$. There exists such a b since w' is connected. Let i_0 be the smallest position in v, such that $v(i_0) \in \Sigma_{\bar{\mathsf{b}}}$. Since $w' \neq \epsilon$ and $\Sigma_{w'} \cap \Sigma_{\bar{\mathsf{b}}} = \emptyset$, we have $i_0 > |w'^{k+1}| > k$. Since every letter in Σ_1 and Σ_2 commutes with each other, we have $v \sim_I \pi_{\bar{\mathsf{b}}}(w)w'^k\pi_{\mathsf{b}}(w)$. Let $v' = \pi_{\bar{\mathsf{b}}}(w)w'^k\pi_{\mathsf{b}}(w)$. The first letter of v' corresponds to the i_0-th letter of v i.e. $(i_0, 1) \in \mathsf{Corr}_{v,v'}$. Hence for every $k \in \mathbb{N}$, there exist $v \in \mathsf{Tr}(\mathcal{A})$ and $i_0 \in \mathsf{Pos}(v)$, such that $\mathsf{Deg}^-_v(i_0) > k$. This implies $\mathsf{Deg}^-_v(i_0) = \infty$, and hence $\mathsf{Deg}(\mathsf{Tr}(\mathcal{A})) = \infty$.

The converse of Theorem 2 also holds. However, we need a more involved technique. We will show this by considering a relation $\mathsf{Block}_w \subseteq \mathsf{Pos}(w) \times \mathsf{Pos}(w)$. This relation tells us positions of two letters in w that do not commute with each other.

Definition 4. *Let* $w \in \Sigma^\infty$ *be a word over an independence alphabet* (Σ, I). *The relation* $\mathsf{Block}_w \subseteq \mathsf{Pos}(w) \times \mathsf{Pos}(w)$ *is the transitive closure of* $D_w = \{(i, j) \mid i \leq j, (w(i), w(j)) \in D\}$, *where* $D = \Sigma^2 \setminus I$.

Consider the word $w = cdbca$ over an independence alphabet (Σ, I) with dependency graph $G : a-b-c-d$. We have $(1, 3), (3, 5) \in \mathsf{Block}_w$ since $(c, b) \in D$ and $(b, a) \in D$. By transitivity, we also have $(1, 5) \in \mathsf{Block}_w$.

If we have $(i, j) \in \mathsf{Block}_w$ for some two positions i, j of w over (Σ, I), then the letters at positions i and j in w do not commute with each other. This implies that their corresponding positions in some word w' that is trace equivalent with w, do not change order, since otherwise the letter at position i in w commute with the one at position j.

Lemma 6. *Let* $w \in \Sigma^\infty$ *be a word over* (Σ, I). *If* $(i, j) \in \mathsf{Block}_w$ *then* $\mathsf{Corr}_{w,w'}(i) \leq \mathsf{Corr}_{w,w'}(j)$ *for all* $w' \sim_I w$.

Proof. Suppose $(i,j) \in \mathsf{Block}_w$. Let $k \in \mathsf{Pos}(w)$, such that $i \leq k < j$, $(i,k) \in \mathsf{Block}_w$, and $(w(k), w(j)) \in D$. Suppose there exists w', such that $\mathsf{Corr}_{w,w'}(k) > \mathsf{Corr}_{w,w'}(j)$. Let $k' = \mathsf{Corr}_{w,w'}(k)$ and $j' = \mathsf{Corr}_{w,w'}(j)$. Let $a, b \in \Sigma$, such that $w(k) = w'(k') = a$, $w(j) = w'(j') = b$, and let $\hat{\Sigma} = (\Sigma_1, \ldots, \Sigma_m)$ be the corresponding distributed alphabet of (Σ, I). Since we assume $(a,b) \in D$, there exists $\ell \in \{1, \ldots, m\}$, such that $a, b \in \Sigma_\ell$. Let $n_1 = |w(1) \ldots w(k)|_a = |w'(1) \ldots w'(k')|_a$ and $n_2 = |w(1) \ldots w(j)|_\mathsf{b} = |w'(1) \ldots w'(j')|_\mathsf{b}$. Note that in the projection of w to Σ_ℓ, i.e. $\pi_\ell(w)$, since $k < j$, there exist at least n_1 many as that occur before the n_2-th b. However, in the projection of w' to Σ_ℓ, i.e. $\pi_\ell(w')$, since $k' > j'$, the n_1-th a occurs after the n_2-th b. There are less than n_1 many as that occur before the n_2-th b. Hence, $\pi_\ell(w) \neq \pi_\ell(w')$. This contradicts $w' \sim_I w$. For all $w' \sim_I w$, we have $\mathsf{Corr}_{w,w'}(k) \leq \mathsf{Corr}_{w,w'}(j)$. By induction hypothesis, we also have $\mathsf{Corr}_{w,w'}(i) \leq \mathsf{Corr}_{w,w'}(k)$ for all $w' \sim_I w$. Thus, $\mathsf{Corr}_{w,w'}(i) \leq \mathsf{Corr}_{w,w'}(j)$ for all $w' \sim_I w$.

In contrast, if we have $(i,j) \notin \mathsf{Block}_w$ for some position $i < j$ in w over (Σ, I), then the letters at positions i and j commute with each other. If there are n many positions of such i then the letter at position j can commute n many steps to the left.

Lemma 7. *Let $w \in \Sigma^\infty$ be a word over (Σ, I). If there are positions $i_1, \ldots, i_n < j_0 \in \mathsf{Pos}(w)$, such that $(i_1, j_0), \ldots, (i_n, j_0) \notin \mathsf{Block}_w$, then there exists $w' \sim_I w$, such that $(j_0, j_0 - n) \in \mathsf{Corr}_{w,w'}$.*

Proof. Let $i_1, \ldots, i_n$ and j_0 be such positions in w. Without loss of generality, we can assume that i_n is the largest position such that $i_n < j_0$ and $(i_n, j_0) \notin \mathsf{Block}_w$. Hence for all k, $i_n < k < j_0$, $(k, j_0) \in \mathsf{Block}_w$. This implies $(w(i_n), w(k)) \notin D$ for all k, $i_n < k < j_0$, since otherwise $(i_n, j_0) \in \mathsf{Block}_w$. Let $u_1 = w(1)w(2) \ldots w(i_n - 1)$, $u_2 = w(i_n)w(i_n + 1) \ldots w(j_0)$, and $u_3 = w(j_0+1)w(j_0+2) \ldots$, such that $w = u_1u_2u_3$. Let $u_2' = w(i_n+1) \ldots w(j_0)\, w(i_n)$. We have $u_2 \sim_I u_2'$ since $(w(i_n), w(k)) \notin D$ for all k, $i_n + 1 \leq k \leq j_0$. Hence we have $w \sim_I u_1u_2'u_3$. Let $w' = u_1u_2'u_3$. The letter at position j_0 in w corresponds to the one at position $j_0 - 1$ in w', i.e. $(j_0, j_0 - 1) \in \mathsf{Corr}_{w,w'}$. By induction hypothesis, there exists $w'' \sim w'$, such that $(j_0 - 1, j_0 - n) \in \mathsf{Corr}_{w',w''}$. Hence we have $(j_0, j_0 - n) \in \mathsf{Corr}_{w,w''}$.

Let us define $\overline{\mathsf{Block}^+_w(i)} = \{j > i \mid (i,j) \notin \mathsf{Block}_w\}$ and $\overline{\mathsf{Block}^-_w(i)} = \{j < i \mid (j,i) \notin \mathsf{Block}_w\}$. We have $j \in \overline{\mathsf{Block}^+_w(i)}$ or $j \in \overline{\mathsf{Block}^-_w(i)}$ if the letter at position $j \neq i$ commutes with the one at position i. The positive and negative signs are used to indicate whether it occurs before or after the letter at position i. In the following, we show that the size of $\overline{\mathsf{Block}^+_w(i)}$ and $\overline{\mathsf{Block}^-_w(i)}$ give us the bound on how long the letter at position i commutes to the right and left, respectively.

Lemma 8. *For all $\mathsf{b} \in \{+, -\}$, $\mathsf{Deg}^\mathsf{b}_w(i) = |\overline{\mathsf{Block}^\mathsf{b}_w(i)}|$.*

Proof. We will show this for $\mathsf{b} = -$. A similar argument can also be used for $\mathsf{b} = +$. Let $k_1 = \mathsf{Deg}^-_w(i)$ and $k_2 = |\overline{\mathsf{Block}^-_w(i)}|$. If $k_1 < k_2$, then there are at

least $k_1 + 1$ many distinct positions in $\overline{\mathsf{Block}_w^-(i)}$, i.e. there are $i_1, \ldots, i_{k_1+1} < i$, such that $(i_1, i), \ldots, (i_{k_1+1}, i) \notin \mathsf{Block}_w$. By Lemma 7, there exists w', such that $w' \sim_I w$ and $(i, i - (k_1 + 1)) \in \mathsf{Corr}_{w,w'}$. This means $\mathsf{Deg}_w^-(i) > k_1$, which contradicts $\mathsf{Deg}_w^-(i) = k_1$.

If $k_1 > k_2$, then $\mathsf{Deg}_w^-(i) > k_2$. There exist a word w' and a position $j \in \mathsf{Pos}(w')$ such that $w \sim_I w'$, $(i, j) \in \mathsf{Corr}_{w,w'}$, and $i - j > k_2$. Consider the set $S = \{i' < i \mid \mathsf{Corr}_{w,w'}(i') > j\}$. We have $|S| = i - j$, and hence $|S| > k_2$. Since there are only k_2 many positions in $\overline{\mathsf{Block}_w^-(i)}$, there is at least one position that is not in $\overline{\mathsf{Block}_w^-(i)}$, but belongs to S. Let i' be such a position. We have $(i', i) \in \mathsf{Block}_w$ and $\mathsf{Corr}_{w,w'}(i') > \mathsf{Corr}_{w,w'}(i)$. This contradicts Lemma 6. We have $k_1 = k_2$ since $k_1 \not< k_2$ and $k_1 \not> k_2$.

One interesting property regarding the cyclic-path-connected automata is that whenever we have a path that goes through a cycle, then every letter on the path does not commute with at least one letter in the cycle. This is always the case since otherwise the cyclic-path will not be connected.

Lemma 9. *Let $\mathcal{A}$ be a cyclic-path-connected automaton, and $p \xrightarrow{w_1} q \xrightarrow{u} q \xrightarrow{w_2} p'$ a cyclic-path over $w = w_1uw_2$. Let $P_1 = \{1, \ldots, |w_1|\}$, $P_2 = \{|w_1| + 1, \ldots, |w_1u|\}$, $P_3 = \{|w_1u| + 1, \ldots, |w_1uw_2|\} \subseteq \mathsf{Pos}(w)$.*

- *For all $i \in P_1$, there exists $j \in P_2$, such that $(i, j) \in \mathsf{Block}_w$.*
- *For all $j \in P_3$, there exists $i \in P_2$, such that $(i, j) \in \mathsf{Block}_w$.*

Proof. We will show this for the first part. A similar argument can also be used to prove the second one. Let $i \in P_1$, $n = |w_1|$, $P_1' = \{i, i + 1, \ldots, n\}$, and $w_1' = w(i)w(i+1)\ldots w(n)$. Since the word $w_1'u$ is produced by a cyclic-path, $w_1'u$ is connected. There exists $j \in P_1' \cup P_2$, such that $(w(i), w(j)) \in D$. If $j \in P_2$, then by definition, $(i, j) \in \mathsf{Block}_w$. If $j \in P_1'$, then by induction hypothesis there exists $j' \in P_2$, such that $(j, j') \in \mathsf{Block}_w$. Since $(i, j) \in \mathsf{Block}_w$, we have $(i, j') \in \mathsf{Block}_w$.

Let us call a path r in $\mathcal{A}$ *wall* if for every path r' that is extended from r, every letter that is read before r and after r does not commute with each other.

Definition 5. *A path $r = p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} p_n$ over $w = a_1 \ldots a_n$ in $\mathcal{A}$ is called a* wall *if for every path r' that is extended from r, i.e. $r' = q \xrightarrow{w_1} p_0 \xrightarrow{w} p_n \xrightarrow{w_2} q'$, over $w' = w_1ww_2$, we have $(i, j) \in \mathsf{Block}_{w'}$ for all $i \in \{1, \ldots, |w_1|\}$, $j \in \{|w_1w| + 1, \ldots, |w_1ww_2|\}$.*

Moreover, let $C(\mathcal{A})$ bethe set of simple cycles of $\mathcal{A}$. By simple cycle, we mean a path that does not visit the same state twice except the first and the last state, i.e. $C(\mathcal{A}) = \{p_1 \xrightarrow{a_1} p_2 \xrightarrow{a_2} \ldots \xrightarrow{a_k} p_k \mid p_1 \neq p_2 \ldots \neq p_{k-1} \text{ and } p_1 = p_k\}$. If we have a path of length $|\mathcal{A}|^2 \cdot |C(\mathcal{A})|$, then there exists a simple cycle that is visited at least $n = |\mathcal{A}|$ many times. This is because for every path of length $|\mathcal{A}|$, there exists a state that is visited at least twice.

Proposition 3. *For any automaton $\mathcal{A}$, if r is a path in $\mathcal{A}$ of length $|\mathcal{A}|^2 \cdot |C(\mathcal{A})|$, then there exists $c \in C(\mathcal{A})$ that is visited at least $|\mathcal{A}|$ many times in r.*

We use this proposition to show the following lemma.

Lemma 10. *If $\mathcal{A}$ is a cyclic-path-connected automaton, then every path of length $|\mathcal{A}|^2 \cdot |C(\mathcal{A})|$ is a wall.*

Proof. Suppose r is a path of length $|\mathcal{A}|^2 \cdot |C(\mathcal{A})|$ from p to p'. By Proposition 3, there exists a simple cycle $c \in C(\mathcal{A})$, suppose over u, that is visited at least $n = |\mathcal{A}|$ many times in r. The path r is over the word $w = v_0 u v_1 \dots u v_n$ for some $v_0, \dots, v_n \in \Sigma^*$. Let r' be a path extended from r, i.e. $r' = s \xrightarrow{w_1} p \xrightarrow{w} p' \xrightarrow{w_2} s'$, over the word $w' = w_1 w w_2$. Let $i_0 \in \{1, \dots, |w_1|\}$ and $j_0 \in \{|w_1 w| + 1, \dots, |w_1 w w_2|\}$. We will show that $(i_0, j_0) \in \mathsf{Block}_{w'}$. Let $m_k = |w_1 v_0 u \dots u v_{k-1}|$ and $P_k = \{m_k + 1, \dots, m_k + |u|\}$ for all $k \in \{1, \dots, n\}$. Since $\mathcal{A}$ is cyclic-path-connected and we can see r' as $s \xrightarrow{w_1 v_0} q \xrightarrow{u} q \xrightarrow{v_1 u \dots u v_n w_2} s'$, by the first part of Lemma 9, there exists $i_1 \in P_1$, such that $(i_0, i_1) \in \mathsf{Block}_{w'}$. Moreover, since $s \xrightarrow{w_1 v_0 u \dots u v_{n-1}} q \xrightarrow{u} q \xrightarrow{v_n w_2} s'$, by the second part of Lemma 9, there exists $j_1 \in P_n$, such that $(j_1, j_0) \in \mathsf{Block}_{w'}$. We will show that $(i_1, j_1) \in \mathsf{Block}_{w'}$.

Since the word u is connected and $w'(i_1), w'(j_1) \in \Sigma_u$, there exists a path from $w'(i_1)$ to $w'(j_1)$ in the dependency graph G_u. There exists such a path of length $m \leq n$ since $|G_u| = |\Sigma_u| \leq n$. Let ℓ be such a path, and $i_2 \in P_2, \dots, i_m \in P_m$, such that $\ell = w'(i_1) w'(i_2) \dots w'(i_m)$. Since an edge in the dependency graph represents dependency between letters, we have $(w'(i_1), w'(i_2)), \dots, (w'(i_{m-1}), w'(i_m)) \in D$. Since $i_1 < \dots < i_m$, we have (i_1, i_2), $\dots$, $(i_{m-1}, i_m) \in \mathsf{Block}_{w'}$. Moreover, since $w'(i_m) = w'(j_1)$ and $i_m \leq j_1$, we also have $(i_m, j_1) \in \mathsf{Block}_{w'}$. Hence $(i_1, j_1) \in \mathsf{Block}_{w'}$, and we have $(i_0, j_0) \in \mathsf{Block}_{w'}$.

This implies that for every finite or infinite path $p_1 a_1 p_2 a_2 \dots$ over $w = a_1 a_2 \dots$, in a cyclic-path-connected automaton with n states and m simple cycles, the letter a_i does not commute with a_j for all $j < i - n^2 m$ and $j > i + n^2 m$. This is because for any $i > 0$, the path $p_i a_i \dots p_{i+n^2 m}$ or $p_{i-n^2 m} \dots a_{i-1} p_i$ is a wall in $\mathcal{A}$. Hence there are at most $n^2 m$ many letters to the left or right of a_i that commute with a_i, i.e. $|\overline{\mathsf{Block}^{\mathsf{b}}_w(i)}| \leq n^2 m$, for any $\mathsf{b} \in \{+, -\}$.

Hence if $\mathcal{A}$ is cyclic-path-connected, then for any word $w \in \mathsf{Tr}(\mathcal{A})$ and position $i \in \mathsf{Pos}(w)$, $\max\{\mathsf{Deg}^+_w(i), \mathsf{Deg}^-_w(i)\} \leq |\mathcal{A}|^2 \cdot |C(\mathcal{A})|$. In other words, the corresponding degree of $\mathsf{Tr}(\mathcal{A})$ is finite. Since $\mathcal{A}$ is cyclic-path-conneted iff the corresponding degree of $\mathsf{Tr}(\mathcal{A})$ is finite, we can refine the characterisation in Theorem 1 into the following theorem.

Theorem 3. *For any two NBA $\mathcal{A}$, $\mathcal{B}$, in which $\mathcal{A}$ is cyclic-path-connected and $\mathcal{B}$ is complete, then $\mathcal{A} \sqsubseteq^{k,\dots,k} \mathcal{B}$ for some $k \in \mathbb{N}$ iff there exists a Lipschitz continuous trace preserving function $f : \mathsf{AccRun}(\mathcal{A}) \to \mathsf{AccRun}(\mathcal{B})$.*

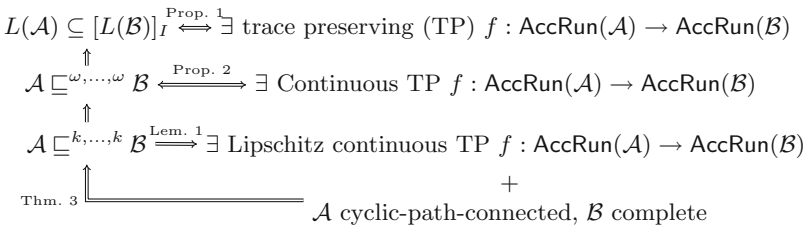


Fig. 1. Topological characterisation of multi-buffer simulation

6 Conclusion

We have shown that we can lift the characterisation of multi-buffer simulation with unbounded buffers to the one with bounded buffers if the automaton for Spoiler is cyclic-path-connected and the automaton for Duplicator is complete. In this case, multi-buffer simulation with bounded buffers can be characterised by the existence of a Lipschitz continuous function that witnesses trace closure inclusion. We summarise the topological characterisation of multi-buffer simulation in Fig. 1.

References

1. Abdulla, P.A., Bouajjani, A., Holík, L., Kaati, L., Vojnar, T.: Computing simulations over tree automata. In: Ramakrishnan, C.R., Rehof, J. (eds.) TACAS 2008. LNCS, vol. 4963, pp. 93–108. Springer, Heidelberg (2008). doi:10.1007/978-3-540-78800-3_8
2. Clerbout, M., Latteux, M.: Semi-commutations. Inf. Comput. **73**(1), 59–74 (1987)
3. Dill, D.L., Hu, A.J., Wong-Toi, H.: Checking for language inclusion using simulation preorders. In: Larsen, K.G., Skou, A. (eds.) CAV 1991. LNCS, vol. 575, pp. 255–265. Springer, Heidelberg (1992). doi:10.1007/3-540-55179-4_25
4. Etessami, K., Wilke, T., Schuller, R.A.: Fair simulation relations, parity games, and state space reduction for Büchi automata. In: Orejas, F., Spirakis, P.G., Leeuwen, J. (eds.) ICALP 2001. LNCS, vol. 2076, pp. 694–707. Springer, Heidelberg (2001). doi:10.1007/3-540-48224-5_57
5. Finkel, O.: Three applications to rational relations of the high undecidability of the infinite post correspondence problem in a regular ω-language. Int. J. Found. Comput. Sci. **23**(7), 1481–1498 (2012)
6. Fritz, C., Wilke, T.: Simulation relations for alternating Büchi automata. Theor. Comput. Sci. **338**(1), 275–314 (2005)
7. Henzinger, T.A., Kupferman, O., Rajamani, S.K.: Fair simulation. Inf. Comput. **173**(1), 64–81 (2002)
8. Holtmann, M., Kaiser, L., Thomas, W.: Degrees of lookahead in regular infinite games. Log. Methods Comput. Sci. 8(3) (2012)
9. Hutagalung, M., Hundeshagen, N., Kuske, D., Lange, M., Lozes, É.: Multi-buffer simulations for trace language inclusion. In: GandALF 2016, pp. 213–227 (2016)

10. Rozenberg, G., Salomaa, A. (eds.): Handbook of Formal Languages: Vol. 3: Beyond Words. Springer, New York (1977)
11. Sakarovitch, J.: The "last" decision problem for rational trace languages. In: Simon, I. (ed.) LATIN 1992. LNCS, vol. 583, pp. 460–473. Springer, Heidelberg (1992). doi:10.1007/BFb0023848