On Effective Construction of the Greatest Solution of Language Inequality $XA \subseteq BX$

Olivier Ly [a] Zhilin Wu [b,*]

[a] *LaBRI, Université Bordeaux/CNRS, 33400, Talence, France*

[b] *State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, P.O.Box 8718, 100190, Beijing, China*

Abstract

In this paper, we consider effective constructions of the greatest solution of the language inequality $XA \subseteq BX$. It has been proved by Kunc in 2005 that the greatest solution of $XA \subseteq BX$ is regular provided that B is regular, no matter what A is. However this proof is based on Kruskal's tree theorem, and does not provide any effective way to construct the greatest solution.

We focus on this gap in this paper. We give an effective construction of the greatest solution for the following two cases:

(i) A, B are regular and there exists $k \geq 1$ such that $\mathrm{pref}(B)A^k \cap B^{\leq k}\mathrm{pref}(B) = \emptyset$, where $\mathrm{pref}(B)$ is the set of prefixes of words in B,

(ii) A, B are regular and B is a code with finite decoding delay.

Our construction takes the point of view of games. As shown by Kunc in his regularity proof, the construction of the greatest solution can be reduced to the construction of the winning region of a two-player game. Our contribution is to show that the winning regions of the two-player game for the two cases can be constructed effectively.

The main ingredient of the construction for the first case is a shrinking lemma for the words on which one of the players has a winning strategy. While the construction for the second case is based on the observation that the two-player game can be reduced to a two-player reachability game played on the transition graph of a one-counter machine.

Key words: Language equations, Regular languages, Codes, Pushdown games, One counter machine.

1. Introduction

Language equations appear as a natural generalization of word equations, and exist in computer science from the early beginning of formal language theory. One can think about Arden's lemma for instance, or context-free languages which are components of the least solutions of systems of polynomial equations. Actually, many natural classes of formal languages have gotten characterizations in terms of equations (see [17,18]).

However, even simple equations may appear to be very difficult. This is the case of the equation $XL = LX$ where X is unknown: The long-standing Conway's problem asks whether the greatest solution of this equation is regular provided that L is regular ([6], see also [9,5,7,8]).

* Corresponding author.
Email addresses: `ly@labri.fr` (Olivier Ly), `wuzl@ios.ac.cn` (Zhilin Wu).

Many advances have been achieved in this domain in the last few years ([9,7,8]). Recently Conway's problem has gotten a solution: Surprisingly, it has been proved by Kunc in [13] that there exists a *finite* language L such that the greatest solution of $XL = LX$ is not recursively enumerable.

For the language inequality $XA \subseteq BX$, it has been proved by Kunc in [12] that the greatest solution of $XA \subseteq BX$ is regular provided that B is regular, whatever A is. But the situation is tight: If one imposes on X to be contained in some given star-free language, then the greatest solution of $XA \subseteq BX$ can become non-recursively enumerable ([11]).

Kunc's regularity proof is obtained by showing that the greatest solution of $XA \subseteq BX$ is upward-closed with respect to a well-quasi-ordering, where the well-quasi-orderedness follows from Kruskal's tree theorem ([10]). The proof is not constructive, i.e., it does not give any effective construction of the greatest solution. *We focus on the effective construction of the greatest solution in this paper*. We give such an *effective construction* for the following two cases:

(i) A, B are regular and there exists $k \geq 1$ such that $\mathrm{pref}(B)A^k \cap B^{\leq k}\mathrm{pref}(B) = \emptyset$, where $\mathrm{pref}(B)$ is the set of prefixes of words in B,

(ii) A, B are regular and B is a code with finite decoding delay (cf. Section 2).

Note that the first case above subsumes the situation that A and B are both finite and $\max_{v \in B} |v| < \min_{u \in A} |u|$ considered in [14]: Let $k = \max_{v \in B} |v| + 1$. It holds that $\mathrm{pref}(B)A^k \cap B^{\leq k}\mathrm{pref}(B) = \emptyset$.

As in [12,14], our construction takes the point of view of games. We consider a game $\mathcal{G}(A, B)$ with two players: *Attacker* and *Defender*. Configurations of the game are words. The game consists of a succession of rounds as follows: First, *Attacker* chooses a word $u \in A$ and appends it to x, where x is the current configuration of the game. If xu has no prefix in B then *Attacker* wins and the game stops. Otherwise, *Defender* chooses a prefix of xu which belongs to B, say v, and cuts it from xu, driving the game to a new configuration x' (i.e. $xu = vx'$) for the next round. *Defender* wins if the game consists of infinitely many rounds. Whether a given word belongs to the greatest solution of $XA \subseteq BX$ is equivalent to the existence of a winning strategy for *Defender* over that word, and the greatest solution of $XA \subseteq BX$ is exactly the winning region of *Defender* (see [12]).

The main ingredient of the effective construction for the first case is a shrinking lemma for words on which *Attacker* has a winning strategy (see Section 3.1), from which it can be deduced that the winning region for *Defender* is a union of equivalence classes of a right congruence of finite index over Σ^*.

Codes with finite decoding delay are particular codes which generalize prefix codes (cf. Section 2, also [1]).

The idea of using codes to simplify the discussion of language equation (or inequality) problems is not new. In [19,7], Conway's problem was solved positively for regular prefix codes and codes, i.e. it was shown that the greatest solution for the language equation $XL = LX$ is regular if L is a regular prefix code or code. On the other hand, it was shown that the greatest solution for the language equation $XA = BX$ is regular provided that A, B are finite biprefix codes [4].

Under the assumption that A, B are regular and B is a code with finite decoding delay, we observe that the game $\mathcal{G}(A, B)$ can be reduced to a two-player reachability game played on the transition graph of a *one-counter machine*. For the situation that A, B are finite, the state space of the one-counter machine is finite and the transition relation is finitely branching. Since such a one-counter machine is a special case of pushdown automata and it is well-known that the winning region of a pushdown game is regular and can be constructed effectively [21,3,20], it follows that the greatest solution of $XA \subseteq BX$ can be constructed effectively from A, B for the situation that A, B are finite and B is a code with finite decoding delay. Nevertheless, if A, B are infinite, then the state space of the one-counter machine is infinite and the transition relation is infinitely branching, which goes beyond the scope of pushdown automata.

To tackle the difficulty, we first show that a congruence can be defined to make the state space finite. We thus obtain a one-counter machine with finite state space, but still with infinitely-branching transition relation [1]. We go one-step further to illustrate how the one-counter game can be simplified so that the transition relation can be trimmed into a finitely-branching one, without modifying the winning regions. Then the effectiveness of the greatest solution follows from the classical results on pushdown games as mentioned before.

This paper is organized as follows. Preliminaries are given in the next section. Then in Section 3, the effective construction for the first case is presented. Section 4 considers the second case, where the game $\mathcal{G}(A, B)$ is reduced

[1] Reachability games played on counter machines, or vector addition systems with states, including a different kind of infinitely-branching transition relation, have been considered in [2].

step-by-step to a one-counter reachability game of finite state space and finitely-branching transition relation. Finally in Section 5, some conclusion is given and the future work is discussed.

Throughout this paper, we assume that the languages A, B in $XA \subseteq BX$ are *regular*.

2. Preliminaries

A finite alphabet Σ is fixed in this paper.

Let v be a prefix (respectively a suffix) of w. We denote by $v^{-1}w$ (respectively wv^{-1}) the unique word v' such that $w = vv'$ (respectively $w = v'v$).

Suppose $L, M \subseteq \Sigma^*$. Let LM denote the concatenation of L and M, $L \setminus M = \{v \in \Sigma^* \mid v \in L, v \notin M\}$ and $\overline{L} = \Sigma^* \setminus L$. Let $M^{-1}L$ and LM^{-1} denote respectively the left and right quotient of L by M, i.e. $M^{-1}L = \{v^{-1}w \mid v \text{ is a prefix of } w, v \in M, w \in L\}$, $LM^{-1} = \{vw^{-1} \mid w \text{ is a suffix of } v, v \in L, w \in M\}$. Let $\mathrm{pref}(L)$ denote the set of prefixes of words in L. Let $L^0 = \{\varepsilon\}$ and $L^i = L^{i-1}L$ for any $i > 0$. Moreover, for any $i \geq 0$, let $L^{\geq i}$ (respectively $L^{\leq i}$) denote the union of the languages L^j for $j : j \geq i$ (respectively for $j : 0 \leq j \leq i$).

Suppose $L \subseteq \Sigma^*$ is a regular language. Let $\sim_L$ denote the Myhill-Nerode equivalence relation of L, that is, for every $x, y \in \Sigma^*$, $x \sim_L y$ iff for every $z \in \Sigma^*$, $xz \in L$ iff $yz \in L$. In addition, let $[x]_L$ denote the equivalence class of $\sim_L$ containing x, $\mathcal{E}(L)$ denote the set of equivalence classes of $\sim_L$, and N_L denote the cardinality of $\mathcal{E}(L)$. For every $x, y \in \overline{\mathrm{pref}(L)}$, we have $x \sim_L y$; and for every $x \in \overline{\mathrm{pref}(L)}, z \in \mathrm{pref}(L)$, we have $x \not\sim_L z$. So if $\overline{\mathrm{pref}(L)} \neq \emptyset$, then $\overline{\mathrm{pref}(L)}$ is an equivalence class of $\sim_L$. Let $\perp$ denote this equivalence class of $\sim_L$.

Given $A, B \subseteq \Sigma^*$, let $\mathcal{C}(A, B)$ denote the greatest solution of $XA \subseteq BX$.

Let $w \in \Sigma^*$. A *play* in the game $\mathcal{G}(A, B)$ starting from w is

- either an infinite sequence $(u_1, v_1)(u_2, v_2)\dots$ such that $u_1, u_2, \dots \in A$, $v_1, v_2, \dots \in B$, and for every $i \geq 1$, $v_1 \dots v_i$ is a prefix of $wu_1 \dots u_i$,
- or a finite sequence $(u_1, v_1)\dots(u_k, v_k)(u_{k+1}, ?)$ (where $k \geq 0$) such that $u_1, \dots, u_{k+1} \in A$, $v_1, \dots, v_k \in B$, for every $i : 1 \leq i \leq k$, $v_1 \dots v_i$ is a prefix of $wu_1 \dots u_i$, and $(v_1 \dots v_k)^{-1}(wu_1 \dots u_{k+1}) \notin B\Sigma^*$.

Attacker is the *winner* of every play of finitely many rounds and *Defender* is the *winner* of every play of infinitely many rounds.

Let $(u_1, v_1)\dots$ be a play of the game $\mathcal{G}(A, B)$ starting from w. Then a *prefix* of the play $(u_1, v_1)\dots$ is either a sequence $(u_1, v_1)\dots(u_k, v_k)$ or a sequence $(u_1, v_1)\dots(u_k, v_k)u_{k+1}$.

Let $w \in \Sigma^*$, f be a partial function from $\Sigma^* \times (A \times B)^*$ to A. Then a prefix of a play of the form $(u_1, v_1)\dots(u_k, v_k)$ in the game $\mathcal{G}(A, B)$ starting from w is said to be *consistent* with f, if $u_1 = f(w, \varepsilon)$ and for every $i : 1 \leq i < k$, $u_{i+1} = f(w, (u_1, v_1)\dots(u_i, v_i))$. A play is said to be consistent with f if every prefix of the play of the form $(u_1, v_1)\dots(u_k, v_k)$ is consistent with f. A *strategy* for *Attacker* in the game $\mathcal{G}(A, B)$ starting from w is a partial function f from $\Sigma^* \times (A \times B)^*$ to A such that for every prefix of a play of the form $(u_1, v_1)\dots(u_k, v_k)$ consistent with f, $f(w, (u_1, v_1)\dots(u_k, v_k))$ is defined. A *winning strategy* for *Attacker* in the game $\mathcal{G}(A, B)$ starting from w is a strategy f such that every play consistent with f in the game $\mathcal{G}(A, B)$ starting from w is finite (thus winning for *Attacker*). Strategies and winning strategies for *Defender* can be defined similarly, with the modification that f is changed to a partial function from $\Sigma^* \times ((A \times B)^*A)$ to B, and prefixes of plays of the form $(u_1, v_1)\dots(u_k, v_k)u_{k+1}$ are considered. The *winning region* for *Attacker* (respectively *Defender*), denoted by $\mathrm{Win}_\alpha(\mathcal{G}(A, B))$ (respectively $\mathrm{Win}_\beta(\mathcal{G}(A, B))$), is the set of words $w \in \Sigma^*$ such that *Attacker* (respectively *Defender*) has a winning strategy in the game $\mathcal{G}(A, B)$ starting from w.

It was proved in [12] that for every $w \in \Sigma^*$, *Defender* has a winning strategy in the game $\mathcal{G}(A, B)$ starting from w iff $w \in \mathcal{C}(A, B)$. Combining this with Martin's determinacy theorem ([15]), we get the following result.

Theorem 1 ([12]) *The game $\mathcal{G}(A, B)$ is determined. More precisely, we have the following.*
- *Attacker has a winning strategy in the game $\mathcal{G}(A, B)$ starting from w iff $w \notin \mathcal{C}(A, B)$.*
- *Defender has a winning strategy in the game $\mathcal{G}(A, B)$ starting from w iff $w \in \mathcal{C}(A, B)$.*

Observation 2 *Let $w \in \Sigma^*$ and f be a winning strategy for Attacker in the game $\mathcal{G}(A, B)$ starting from w. Then there is a number $M_f \geq 1$ such that every (finite) play consistent with f has length at most M_f.*

A *strong strategy* for *Attacker* (respectively *Defender*) is a strategy in the game $\mathcal{G}(A, B)$ modified such that *Attacker* (respectively *Defender*) can choose the concatenations of several words of A (respectively B) in the same round.

Formally, a strong strategy for *Attacker* (respectively *Defender*) is a strategy in the game $\mathcal{G}(A^+, B)$ (respectively $\mathcal{G}(A, B^+)$) instead of $\mathcal{G}(A, B)$.

Proposition 3 *In the game $\mathcal{G}(A, B)$, Attacker (respectively Defender) has a winning strong strategy iff he (respectively she) has a winning strategy.*

PROOF. We illustrate the proof for *Attacker*.

"If" direction:

The proof is trivial, since a winning strategy of *Attacker* is a winning strong strategy of *Attacker*.

"Only if" direction:

Suppose f is a winning strategy for *Attacker* in the game $\mathcal{G}(A^+, B)$ starting from w. We construct a strategy f' for *Attacker* in the game $\mathcal{G}(A, B)$ starting from w as follows.

- Suppose $f(w, \varepsilon) = u_1 \dots u_{i_1}$. Then $f'(w, \varepsilon) = u_1$.
- Suppose that in the first round of $\mathcal{G}(A, B)$, the choice of *Defender* is v_1, and $f(w, (u_1 \dots u_{i_1}, v_1)) = u_{i_1+1} \dots u_{i_2}$. Define $f'(w, (u_1, v_1)) = u_2$.
- In general, suppose that in the game $\mathcal{G}(A, B)$ starting from w, j rounds have been played, the choices of *Defender* are $v_1, \dots, v_j$, and for every $0 \leq k < j$, $f'(w, (u_1, v_1) \dots (u_k, v_k)) = u_{k+1}$. Our goal is to define $f'(w, (u_1, v_1) \dots (u_j, v_j))$. For this purpose, consider the game $\mathcal{G}(A^+, B)$ starting from w. Suppose that in the j rounds of $\mathcal{G}(A^+, B)$, the choices of *Defender* are $v_1, \dots, v_j$, and *Attacker* has played by following f. Then for every $k : 0 \leq k \leq j$, $f(w, (u_1 \dots u_{i_1}, v_1) \dots (u_{i_{k-1}+1} \dots u_{i_k}, v_k)) = u_{i_k+1} \dots u_{i_{k+1}}$ ($i_0 = 0$ by convention), where $u_1, \dots, u_{i_{j+1}} \in A$. Evidently, $i_{j+1} \geq j + 1$. We define $f'(w, (u_1, v_1) \dots (u_j, v_j)) = u_{j+1}$.

Because f is a winning strategy in the game $\mathcal{G}(A^+, B)$ starting from w, from Observation 2, we know that there is $M_f \geq 1$ such that every play in the game $\mathcal{G}(A^+, B)$ consistent with f has length at most M_f. Therefore, every play in the game $\mathcal{G}(A, B)$ consistent with f' has also length at most M_f. We conclude that f' is a winning strategy for *Attacker*. $\Box$

It is easy to show the following upper bound for $\mathcal{C}(A, B)$.

Proposition 4 $\mathcal{C}(A, B) \subseteq B^*\mathrm{pref}(B)$.

As a result of Proposition 4, in the rest of this paper, we will concentrate on the set of configurations belonging to $B^*\mathrm{pref}(B)$. We also would like to remark that the set $B^*\mathrm{pref}(B)$ is prefix-closed.

In the rest of this paper, for the language inequality $XA \subseteq BX$, it is assumed that

$$A, B \text{ are regular}, A, B \neq \emptyset, A, B \subseteq \Sigma^+.$$

Note that the assumption that $A, B \subseteq \Sigma^+$ is justified by the following observation.

Observation 5 *If $\varepsilon \in B$, then the greatest solution of $XA \subseteq BX$ is Σ^*; on the other hand, if $\varepsilon \in A$ and $\varepsilon \notin B$, then the greatest solution is $\emptyset$.*

PROOF. If $\varepsilon \in B$, then *Defender* has the following winning strategy in the game $\mathcal{G}(A, B)$ starting from any word: Cut the current configuration by the empty word ε in each round, no matter what *Attacker* chooses. Therefore, the winning region of *Defender*, that is, the greatest solution, is Σ^*.

On the other hand, if $\varepsilon \in A$ and $\varepsilon \notin B$, then *Attacker* has the following winning strategy in the game $\mathcal{G}(A, B)$ starting from any word: Append the empty word ε in each round, no matter what *Defender* chooses. Because $\varepsilon \notin B$, *Defender* has to cut at least one letter from the current configuration in each round, and the length of the configuration (word) decreases strictly in each round. Therefore, *Defender* loses the game starting from any word, and the winning region of *Defender*, i.e. the greatest solution, is $\emptyset$. $\Box$

Definition 6 (Finitely lengthening) *Let $A, B \subseteq \Sigma^+$ and $k \geq 1$. Then the pair of languages (A, B) is said to be k lengthening if $\mathrm{pref}(B)A^k \cap B^{\leq k}\mathrm{pref}(B) = \emptyset$. Intuitively, if (A, B) is k lengthening, then for every $w \in \mathrm{pref}(B)$, the words obtained by concatenating any k choices of Attacker to the right of w, will take Defender at least $(k + 1)$ rounds to cut. In addition, (A, B) is said to be* finitely lengthening *if (A, B) is k lengthening for some $k \geq 1$.*

Definition 7 (Codes and finite decoding delay) *Let $L \subseteq \Sigma^+$. L is a* code *if for every $x_1, \ldots, x_n, x'_1, \ldots, x'_m \in L$, $x_1 \ldots x_n = x'_1 \ldots x'_m$ implies that $n = m$ and $x_i = x'_i$ for all i. L is called a* prefix code *if for every $u, v \in L$, u is not a strict prefix of v. L is called a code with* finite decoding delay *if there is a natural number $d \geq 0$ such that*

$$\forall x, x' \in L, \forall y \in L^d, \forall u \in \Sigma^*, xyu \in x'L^* \Rightarrow x = x'.$$

If L has finite decoding delay, then the smallest integer d satisfying the above condition is called the decoding delay of L.

Intuitively, suppose that L is a code with decoding delay d, then given a word $x \in L^*$ (which has a unique decomposition into words in L), if the words $v_1, \ldots, v_{d+1} \in L$ have been found during the decoding of x such that $x = v_1 \ldots v_{d+1}x'$ for some $x' \in \Sigma^*$, then the unique correct decoding of x into words in L must start with v_1.

A code has decoding delay 0 iff it is a prefix code ([1]). On the other hand, it is not hard to verify that $L = \{a, abc, c\}$ is a code with decoding delay 1.

Definition 8 (Strategy trees) *For each $x \in B^*\mathrm{pref}(B)$, define the* strategy tree [2] *of x, denoted by $S(x)$, as follows:*
- *The nodes of $S(x)$ are the sequences $[v_1, \ldots, v_n]$ such that $v_1, \ldots, v_n \in B$, and $x = v_1 \ldots v_n y$ for some $y \in B^*\mathrm{pref}(B)$.*
- *For every node $[v_1, \ldots, v_{n-1}]$ and node $[v_1, \ldots, v_n]$ in $S(x)$, $[v_1, \ldots, v_{n-1}]$ is the parent of $[v_1, \ldots, v_n]$. In particular, the root of $S(x)$ is ε, and the leaves of $S(x)$ are the nodes $[v_1, \ldots, v_n]$ such that $x = v_1 \ldots v_n y$ with $y \in \mathrm{pref}(B) \setminus B^+\mathrm{pref}(B)$.*
- *For every node $[v_1, \ldots, v_n]$ in $S(x)$ such that $x = v_1 \ldots v_n y$, the label of $[v_1, \ldots, v_n]$ is $[y]_B$.*

Note that in the above definition of $S(x)$, if $x = v_1 \ldots v_n y$ such that $v_1, \ldots, v_n \in B$ and $y \notin B^*\mathrm{pref}(B)$, then $[v_1, \ldots, v_n]$ is not included as a node of $S(x)$ [3].

Definition 9 (Subtrees, tree morphisms and isomorphisms) *Let Γ be a finite alphabet.*
- *Suppose that T is a Γ-labeled finite tree and x is a node in T. Then let $T|_x$ denote the subtree of T rooted at x.*
- *Let T_1, T_2 be two Γ-labeled finite trees.*
  - *A* morphism *π from T_1 to T_2 is a mapping from T_1 to T_2 which preserves the root, the parent-child relation and the labels of nodes. Let $\lesssim$ and $\simeq$ denote respectively the quasi-order and the equivalence relation induced by the morphisms over Γ-labeled finite trees. More specifically, $T_1 \lesssim T_2$ iff there is a morphism from T_1 to T_2, and $T_1 \simeq T_2$ iff $T_1 \lesssim T_2$ and $T_2 \lesssim T_1$.*
  - *An* isomorphism *from T_1 to T_2 is a bijective morphism from T_1 to T_2. If there is an isomorphism from T_1 to T_2, then T_1 and T_2 are said to be* isomorphic*, denoted by $T_1 \cong T_2$.*

Let $\approx_S$ denote the equivalence relation on $B^*\mathrm{pref}(B)$ defined as follows: $x \approx_S y$ iff $S(x) \cong S(y)$.

Proposition 10 *Let $x, y \in B^*\mathrm{pref}(B)$ such that $x \approx_S y$. Then for every $z \in \Sigma^*$, $xz \in B^*\mathrm{pref}(B)$ iff $yz \in B^*\mathrm{pref}(B)$. Moreover, if $xz, yz \in B^*\mathrm{pref}(B)$, then $S(xz) \cong S(yz)$.*

PROOF. Suppose $x, y \in B^*\mathrm{pref}(B)$, $x \approx_S y$, and $z \in \Sigma^*$.

At first, we show that $xz \in B^*\mathrm{pref}(B)$ iff $yz \in B^*\mathrm{pref}(B)$. By symmetry, it is sufficient to show that $xz \in B^*\mathrm{pref}(B)$ implies that $yz \in B^*\mathrm{pref}(B)$.

Suppose $xz \in B^*\mathrm{pref}(B)$. Then $xz = v_1 \ldots v_i z'$ for some $v_1, \ldots, v_i \in B$ and $z' \in \mathrm{pref}(B)$.

There are the following two situations, z is a suffix of z' or z' is a suffix of z.

If z is a suffix of z', then $x = v_1 \ldots v_i x'$ and $z' = x'z$ for some $x' \in \Sigma^*$. Because $S(x) \cong S(y)$, then there are $v'_1, \ldots, v'_i \in B$ and y' such that $y = v'_1 \ldots v'_i y'$ and $x' \sim_B y'$. Therefore, $x'z \sim_B y'z$. Because $z' = x'z \in \mathrm{pref}(B)$, it follows that $y'z \in \mathrm{pref}(B)$. Consequently, $yz = v'_1 \ldots v'_i(y'z) \in B^*\mathrm{pref}(B)$.

On the other hand, if z' is a suffix of z, then $x = v_1 \ldots v_{j-1}x'$, $z = x''v_{j+1} \ldots v_i z'$, and $v_j = x'x''$ for some $j : 1 \leq j \leq i$ and $x', x'' \in \Sigma^*$. From the fact that $S(x) \cong S(y)$, we know that there are $v'_1, \ldots, v'_{j-1} \in B$ and $y' \in \Sigma^*$ such that $y = v'_1 \ldots v'_{j-1}y'$ and $x' \sim_B y'$. Then $y'x'' \in B$, since $x'x'' = v_j \in B$. Therefore, $yz = v'_1 \ldots v'_{j-1}(y'x'')v_{j+1} \ldots v_i z' \in B^*\mathrm{pref}(B)$.

---

[2] The concept of strategy trees was introduced in [12], where each node of the strategy tree is labeled by a set of equivalence classes of $\sim_B$, instead of a single one.

[3] In the definition of [12], $[v_1, \ldots, v_n]$ is also included as a node of $S(x)$, which is another difference between our definition of strategy trees and that in [12].

Suppose $xz, yz \in B^*\text{pref}(B)$. Then $S(xz)$ is obtained from $S(x)$ as follows: For each node $[v_1, \dots, v_i]$ in $S(x)$ such that $x = v_1 \dots v_i x'$ and $x' \in \text{pref}(B)$, do the following:

For every nonempty prefix z' of z such that $x'z' \in B$ and $z'' = (z')^{-1}z \in B^*\text{pref}(B)$, add $[v_1, \dots, v_i, x'z']$ labeled by $[z'']_B$ as a child of $[v_1, \dots, v_i]$, and add the strategy tree $S(z'')$ as the subtree of the node $[v_1, \dots, v_i, x'z']$, which means that every node $[v'_1, \dots, v'_j]$ in $S(z'')$ becomes a node $[v_1, \dots, v_i, x'z', v'_1, \dots, v'_j]$ in $S(xz)$, with the label preserved.

Similarly, $S(yz)$ can be obtained from $S(y)$.

From the above procedure to obtain $S(xz)$ and $S(yz)$ from respectively $S(x)$ and $S(y)$ and the fact that $S(x)$ and $S(y)$ are isomorphic, it is not hard to see that $S(xz)$ and $S(yz)$ are isomorphic as well. □

The following result follows from Proposition 10.

Proposition 11 *Let $x, y \in B^*\text{pref}(B)$ such that $S(x) \cong S(y)$. Then Defender has a winning strategy in $\mathcal{G}(A, B)$ starting from x iff Defender has a winning strategy in $\mathcal{G}(A, B)$ starting from y.*

3. The case that (A, B) is finitely lengthening

At first, we would like to point out a fact that the condition that (A, B) is finitely lengthening subsumes the condition that A, B are finite and $\max_{b \in B} |b| < \min_{a \in A} |a|$ considered in [14]: Let $k = \max_{v \in B} |v| + 1$, then $\text{pref}(B)A^k \cap B^{\leq k}\text{pref}(B) = \emptyset$. Moreover, this subsumption is strict, since A and B are not required to be finite for finitely lengthening (A, B). For instance, let $A = \{bba\}$, $B = a^*b$ and $k = 1$, then it is not hard to verify that $\text{pref}(B)A^k \cap B^{\leq k}\text{pref}(B) = \text{pref}(B)A \cap B^{\leq 1}\text{pref}(B) = \emptyset$.

Let us assume that $k \geq 1$ and (A, B) is k lengthening in the rest of this section.

Proposition 12 *Suppose that (A, B) is k lengthening. Then $(B^*\text{pref}(B))A^{kn} \cap B^{\leq (k+1)n-1}\text{pref}(B) = \emptyset$ for every $n \geq 1$.*

PROOF.

At first, we show that $\text{pref}(B)A^{kn} \cap B^{\leq (k+1)n-1}\text{pref}(B) = \emptyset$ implies $(B^*\text{pref}(B))A^{kn} \cap B^{\leq (k+1)n-1}\text{pref}(B) = \emptyset$.

Suppose $(B^*\text{pref}(B))A^{kn} \cap B^{\leq (k+1)n-1}\text{pref}(B) \neq \emptyset$. Then there are $x \in B^*\text{pref}(B)$, $u_1, \dots, u_{kn} \in A$, $t \leq (k+1)n - 1$, $v_1, \dots, v_t \in B$, $y \in \text{pref}(B)$ such that $xu_1 \dots u_{kn} = v_1 \dots v_t y$.

- If x is a prefix of $v_1 \dots v_t$, then $x = v_1 \dots v_{j-1}v'_j$, $u_1 \dots u_{kn} = v''_j v_{j+1} \dots v_t y$, and $v_j = v'_j v''_j$ for some $j : 1 \leq j \leq t$ and $v'_j, v''_j \in \Sigma^*$. It follows that $v'_j u_1 \dots u_{kn} = v_j \dots v_t y$. Therefore, $\text{pref}(B)A^{kn} \cap B^{\leq (k+1)n-1}\text{pref}(B) \neq \emptyset$.
- If $v_1 \dots v_t$ is a prefix of x, then $x = v_1 \dots v_t y'$, $u_1 \dots u_{kn} = y''$, and $y = y'y''$ for some $y', y'' \in \Sigma^*$. It follows that $y'u_1 \dots u_{kn} = y'y'' = y \in \text{pref}(B)$, so $\text{pref}(B)A^{kn} \cap B^{\leq (k+1)n-1}\text{pref}(B) \neq \emptyset$ as well.

Next, we show that $\text{pref}(B)A^{kn} \cap B^{\leq (k+1)n-1}\text{pref}(B) = \emptyset$ and complete the proof.

The proof goes by induction on n.

Induction base $n = 1$: Follows from the assumption.

Induction step $n > 1$: To the contrary, suppose that $\text{pref}(B)A^{kn} \cap B^{\leq (k+1)n-1}\text{pref}(B) \neq \emptyset$. Then there are $x \in \text{pref}(B)$, $u_1, \dots, u_{kn} \in A$, $i \leq (k+1)n - 1$, $v_1, \dots, v_i \in B$, and $y \in \text{pref}(B)$ such that $xu_1 \dots u_{kn} = v_1 \dots v_i y$.

- If $xu_1 \dots u_k$ is a prefix of $v_1 \dots v_i$, then $xu_1 \dots u_k = v_1 \dots v_{j-1}v'_j$, $u_{k+1} \dots u_{kn} = v''_j v_{j+1} \dots v_i y$, and $v_j = v'_j v''_j$ for some $j : 1 \leq j \leq i$, $v'_j, v''_j \in \Sigma^*$. It follows that $v'_j u_{k+1} \dots u_{kn} = v_j \dots v_i y$. Because $\text{pref}(B)A^k \cap B^{\leq k}\text{pref}(B) = \emptyset$ from the assumption, and $\text{pref}(B)A^{k(n-1)} \cap B^{\leq (k+1)(n-1)-1}\text{pref}(B) = \emptyset$ according to the induction hypothesis, it follows that $j - 1 > k$ and $i - j + 1 \geq (k+1)(n-1)$. Therefore, $i \geq (k+1)n$, a contradiction.
- If $v_1 \dots v_i$ is a prefix of $xu_1 \dots u_k$, then $xu_1 \dots u_k = v_1 \dots v_i y'$, $u_{k+1} \dots u_{kn} = y''$ and $y = y'y''$ for some $y', y'' \in \Sigma^*$. Thus, $y'u_{k+1} \dots u_{kn} = y'y'' = y \in \text{pref}(B)$. Therefore, $\text{pref}(B)A^{k(n-1)} \cap \text{pref}(B) \neq \emptyset$. We deduce that $\text{pref}(B)A^{k(n-1)} \cap B^{\leq (k+1)(n-1)-1}\text{pref}(B) \neq \emptyset$, a contradiction to the induction hypothesis. □

Proposition 12 tells us that for every word $w \in B^*\mathrm{pref}(B)$, the words obtained by concatenating any kn choices of *Attacker* to the right of w, will take *Defender* at least $(k+1)n$ rounds to cut.

3.1. *A Shrinking Lemma on Attacker's Strategies*

We fix a number $N_1 = 2k(k+1)\max\left\{k+2, 2^{N_B 2^{N_B}} + 1\right\}$ (recall that N_B is the number of equivalence classes of $\sim_B$) in this section, whose purpose will become clear later.

Let $x \in \Sigma^*$. Define the *visibility* of *Defender* through x, denoted by $\mathrm{Vis}(x)$, as follows:

$$\mathrm{Vis}(x) := \{[y]_B \mid y \in \mathrm{pref}(B) \text{ and } x \in B^*y\}.$$

Intuitively, $\mathrm{Vis}(x)$ consists of the labels $[y]_B$ of all nodes $[v_1, \ldots, v_n]$ in $S(x)$ such that $x = v_1 \ldots v_n y$ and $y \in \mathrm{pref}(B)$.

Note that $\mathrm{Vis}(x) = \emptyset$ for every $x \notin B^*\mathrm{pref}(B)$ and $\perp \notin \mathrm{Vis}(x)$ for every $x \in B^*\mathrm{pref}(B)$.

Proposition 13 *Let $x, y, w \in \Sigma^*$ such that Vis(x) = Vis(y). Then Vis(xw) = Vis(yw).*

PROOF. Suppose $x, y, w \in \Sigma^*$ and $\mathrm{Vis}(x) = \mathrm{Vis}(y)$. In the following, we will show that $\mathrm{Vis}(xw) \subseteq \mathrm{Vis}(yw)$. The argument for $\mathrm{Vis}(yw) \subseteq \mathrm{Vis}(xw)$ is symmetric.

Let $z \in \mathrm{pref}(B)$ such that $xw \in B^*z$. We show that $[z]_B \in \mathrm{Vis}(yw)$.

Suppose $xw = v_1 \ldots v_r z$ for $v_1, \ldots, v_r \in B$.

There are the following two cases.

- $x = v_1 \ldots v_j v'_j$, $w = v''_j v_{j+1} \ldots v_r z$, and $v_j = v'_j v''_j$ for some $j : 1 \leq j \leq r$, $v'_j, v''_j \in \Sigma^*$.
- $x = v_1 \ldots v_r z'$, $w = z''$, and $z = z'z''$ for $z', z'' \in \Sigma^*$.

For the first case above, because $[v'_j]_B \in \mathrm{Vis}(x) = \mathrm{Vis}(y)$, there is $y' \in \mathrm{pref}(B)$ such that $y \in B^*y'$ and $[v'_j]_B = [y']_B$. From the fact that $v'_j v''_j = v_j \in B$, it is deduced that $y'v''_j \in B$. Therefore, $yw = (yv''_j)v_{j+1} \ldots v_r z \in B^*(y'v''_j)v_{j+1} \ldots v_r z \subseteq B^*z$, we conclude that $[z]_B \in \mathrm{Vis}(yw)$.

For the second case above, because $[z']_B \in \mathrm{Vis}(x) = \mathrm{Vis}(y)$, there is $y' \in \mathrm{pref}(B)$ such that $y \in B^*y'$ and $[z']_B = [y']_B$. Therefore, $[z]_B = [z'z'']_B = [y'z'']_B$ and $yw = yz'' \in B^*y'z''$. It follows that $[z]_B = [y'z'']_B \in \mathrm{Vis}(yw)$. □

Definition 14 (N_1-visibility tree) *Let $x \in B^*\mathrm{pref}(B)$, define the N_1-visibility tree of x, denoted by $S_{N_1}(x)$, as follows:*

$S_{N_1}(x)$ is obtained from the strategy tree $S(x)$ by the following two steps:

(i) remove all the nodes at depth (strictly) greater than N_1 (the root has depth 0),

(ii) relabel every node $[v_1, \ldots, v_{N_1}]$ (at depth N_1) such that $x = v_1 \ldots v_{N_1} y$ by Vis(y).

Note that $S_{N_1}(x)$ is a $2^{\mathcal{E}(B)}$-labeled finite tree.

Proposition 15 *Let $x, y \in B^*\mathrm{pref}(B)$ and π be a morphism from $S_{N_1}(x)$ to $S_{N_1}(y)$. For every node $[v_1, \ldots, v_n]$ in $S_{N_1}(x)$, suppose $x = v_1 \ldots v_n x'$, $\pi([v_1, \ldots, v_n]) = [v'_1, \ldots, v'_n]$, and $y = v'_1 \ldots v'_n y'$, then we have Vis$(x') \subseteq$ Vis(y').*

PROOF. Suppose $[x'']_B \in \mathrm{Vis}(x')$, that is, $x' \in B^*x''$ with $x'' \in \mathrm{pref}(B)$. We show that $[x'']_B \in \mathrm{Vis}(y')$.

Since $x' \in B^*x''$, there are $v_{n+1}, \ldots, v_m \in B$ (where $m \geq n$) such that $x' = v_{n+1} \ldots v_m x''$. Then $x = v_1 \ldots v_n x' = v_1 \ldots v_n v_{n+1} \ldots v_m x''$.

If $m \leq N_1$, then $[v_1, \ldots, v_m]$ is a node in $S_{N_1}(x)$, so there are $v'_{n+1}, \ldots, v'_m \in B$ such that $[v'_1, \ldots, v'_m]$ is a node in $S_{N_1}(y)$ and $\pi([v_1, \ldots, v_m]) = [v'_1, \ldots, v'_m]$. Suppose $y = v'_1 \ldots v'_m y''$. Then $y' = v'_{n+1} \ldots v'_m y''$. Since $[y'']_B$, the label of $[v'_1, \ldots, v'_m]$, is equal to $[x'']_B$, the label of $[v_1, \ldots, v_m]$, it follows that $y'' \in \mathrm{pref}(B)$ and $[x'']_B = [y'']_B \in \mathrm{Vis}(y')$.

On the other hand, if $m > N_1$, then $[v_1, \ldots, v_{N_1}]$ is a node in $S_{N_1}(x)$, so there are $v'_{n+1}, \ldots, v'_{N_1} \in B$ such that $\pi([v_1, \ldots, v_{N_1}]) = [v'_1, \ldots, v'_{N_1}]$ and $y' = v'_{n+1} \ldots v'_{N_1} y''$ for some $y'' \in B^*\mathrm{pref}(B)$. It follows that $\mathrm{Vis}(y'')$, the label of $[v'_1, \ldots, v'_{N_1}]$ in $S_{N_1}(y)$, is equal to $\mathrm{Vis}(v_{N_1+1} \ldots v_m x'')$, the label of $[v_1, \ldots, v_{N_1}]$ in $S_{N_1}(x)$. Therefore, $[x'']_B \in \mathrm{Vis}(v_{N_1+1} \ldots v_m x'') = \mathrm{Vis}(y'') \subseteq \mathrm{Vis}(v'_{n+1} \ldots v'_{N_1} y'') = \mathrm{Vis}(y')$. □

Definition 16 (Reduced N_1-visibility tree) *Let $x \in B^*\text{pref}(B)$. The reduced N_1-visibility tree of x, denoted by $RS_{N_1}(x)$, is obtained from $S_{N_1}(x)$ by the following algorithm.*

Initially let $i = N_1$, $T_{N_1}(x) = S_{N_1}(x)$, repeat the following procedure until $i = 0$.

$T_{i-1}(x)$ is obtained from $T_i(x)$ by applying the following operations:

For each node $[v_1, \ldots, v_{i-1}]$ in $T_i(x)$, select a subset of the children of $[v_1, \ldots, v_{i-1}]$, say $[v_1, \ldots, v_{i-1}, v_{i,1}]$, $\ldots$, $[v_1, \ldots, v_{i-1}, v_{i,r}]$, such that

- *for each child $[v_1, \ldots, v_{i-1}, v'_i]$ of $[v_1, \ldots, v_{i-1}]$, there is $j : 1 \leq j \leq r$ such that $T_i(x)|_{[v_1,\ldots,v_{i-1},v'_i]} \lesssim T_i(x)|_{[v_1,\ldots,v_{i-1},v_{i,j}]}$,*
- *the subtrees $T_i(x)|_{[v_1,\ldots,v_{i-1},v_{i,1}]}, \ldots, T_i(x)|_{[v_1,\ldots,v_{i-1},v_{i,r}]}$ form an antichain of $\lesssim$.*

Keep the subtrees of $[v_1, \ldots, v_{i-1}]$ rooted at $[v_1, \ldots, v_{i-1}, v_{i,1}], \ldots, [v_1, \ldots, v_{i-1}, v_{i,r}]$ and remove all the other subtrees of $[v_1, \ldots, v_{i-1}]$.

Set $i := i - 1$. If $i = 0$, set $RS_{N_1}(x) := T_0(x)$.

By an induction on the depth of trees, it is not hard to show that there are only finitely many non-isomorphic reduced N_1-visibility trees.

Proposition 17 *Let $x, y \in B^*\text{pref}(B)$. Then $RS_{N_1}(x) \cong RS_{N_1}(y)$ iff $RS_{N_1}(x) \simeq RS_{N_1}(y)$.*

PROOF. The "Only if" direction is trivial.

"If" direction:

Suppose $RS_{N_1}(x) \simeq RS_{N_1}(y)$, i.e. there are two morphisms $\pi_1 : RS_{N_1}(x) \to RS_{N_1}(y)$ and $\pi_2 : RS_{N_1}(y) \to RS_{N_1}(x)$.

In the following, we will show that for every node $[v_1, \ldots, v_i]$ in $RS_{N_1}(x)$, $\pi_2(\pi_1([v_1, \ldots, v_i])) = [v_1, \ldots, v_i]$. Symmetrically, we can also show that for every node $[v'_1, \ldots, v'_i]$ in $RS_{N_1}(y)$, $\pi_1(\pi_2([v'_1, \ldots, v'_i])) = [v'_1, \ldots, v'_i]$. From these two facts, it is deduced that π_1 and π_2 are both injective mappings. We conclude that π_1 and π_2 are in fact isomorphisms, so $RS_{N_1}(x) \cong RS_{N_1}(y)$.

The proof of $\pi_2(\pi_1([v_1, \ldots, v_i])) = [v_1, \ldots, v_i]$ is by an induction on i.

Suppose $[v_1, \ldots, v_i]$ is a node in $RS_{N_1}(x)$.

Induction base: $i = 0$. From the definition of morphisms, we know that $\pi_2(\pi_1(\varepsilon)) = \varepsilon$.

Induction step: $i > 0$.

Let $[v_1, \ldots, v_{i-1}, v_{i,0}], [v_1, \ldots, v_{i-1}, v_{i,1}], \ldots, [v_1, \ldots, v_{i-1}, v_{i,r}]$ be a list of all the children of $[v_1, \ldots, v_{i-1}]$ in $RS_{N_1}(x)$ with $v_{i,0} = v_i$. Then by the induction hypothesis, $\pi_2(\pi_1([v_1, \ldots, v_{i-1}])) = [v_1, \ldots, v_{i-1}]$.

From the definition of morphisms, we know that π_1 maps the children of $[v_1, \ldots, v_{i-1}]$ to those of $\pi_1([v_1, \ldots, v_{i-1}])$ and π_2 maps the children of $\pi_1([v_1, \ldots, v_{i-1}])$ to those of $[v_1, \ldots, v_{i-1}]$. Therefore, $\pi_2(\pi_1([v_1, \ldots, v_i]))$ is a child of $[v_1, \ldots, v_{i-1}]$.

On the one hand, we have $RS_{N_1}(x)|_{[v_1,\ldots,v_i]} \lesssim RS_{N_1}(y)|_{\pi_1([v_1,\ldots,v_i])} \lesssim RS_{N_1}(x)|_{\pi_2(\pi_1([v_1,\ldots,v_i]))}$. On the other hand, according to the construction of $RS_{N_1}(x)$ from $S_{N_1}(x)$, it is impossible that there is a morphism between the two subtrees of $RS_{N_1}(x)$ rooted at two distinct children of $[v_1, \ldots, v_{i-1}]$. We conclude that $\pi_2(\pi_1([v_1, \ldots, v_i])) = [v_1, \ldots, v_i]$. $\square$

Proposition 18 *Let $x, y \in B^*\text{pref}(B)$. Then $RS_{N_1}(x) \cong RS_{N_1}(y)$ iff $S_{N_1}(x) \simeq S_{N_1}(y)$.*

PROOF. From Proposition 17, we know that $RS_{N_1}(x) \cong RS_{N_1}(y)$ iff $RS_{N_1}(x) \simeq RS_{N_1}(y)$.

Since $RS_{N_1}(x)$ is a subgraph of $S_{N_1}(x)$, it is evident that there is a morphism from $RS_{N_1}(x)$ to $S_{N_1}(x)$.

Moreover, according to the construction of $RS_{N_1}(x)$ from $S_{N_1}(x)$, it is not hard to see that there is also a morphism from $S_{N_1}(x)$ to $RS_{N_1}(x)$.

Therefore, $S_{N_1}(x) \simeq RS_{N_1}(x)$. Similarly, $S_{N_1}(y) \simeq RS_{N_1}(y)$.

Since $\simeq$ is an equivalence relation, it follows that $RS_{N_1}(x) \cong RS_{N_1}(y)$ iff $RS_{N_1}(x) \simeq RS_{N_1}(y)$ iff $S_{N_1}(x) \simeq S_{N_1}(y)$. $\square$

Definition 19 (B-relation) *Let $w, w' \in \Sigma^*$. Then w, w' are said to be B-*related*, denoted by $w \leftrightarrow_B w'$, iff*

- *either $w, w' \notin B^*\text{pref}(B)$,*

– *or* $w, w' \in B^*\mathrm{pref}(B)$ *and* $S_{N_1}(w) \simeq S_{N_1}(w')$.

Lemma 20 *The relation* $\leftrightarrow_B$ *is a right congruence of finite index.*

PROOF. From Proposition 18 and the fact that there are only finitely many non-isomorphic reduced N_1-visibility trees, we know that $\leftrightarrow_B$ is of finite index.

It remains to show that $\leftrightarrow_B$ is a right congruence.

Suppose that $w \leftrightarrow_B w'$ and $x \in \Sigma^*$.

If $w, w' \notin B^*\mathrm{pref}(B)$, then obviously $wx, w'x \notin B^*\mathrm{pref}(B)$. Therefore, $wx \leftrightarrow_B w'x$.

In the following, we assume that $w, w' \in B^*\mathrm{pref}(B)$. Then $S_{N_1}(w) \simeq S_{N_1}(w')$. Let π be a morphism from $S_{N_1}(w)$ to $S_{N_1}(w')$.

We first prove that $wx \in B^*\mathrm{pref}(B)$ iff $w'x \in B^*\mathrm{pref}(B)$. By symmetry, it is sufficient to show that $wx \in B^*\mathrm{pref}(B)$ implies $w'x \in B^*\mathrm{pref}(B)$. Suppose $wx \in B^*\mathrm{pref}(B)$. Then $wx = v_1 \dots v_n y$ for $v_1, \dots, v_n \in B$ and $y \in \mathrm{pref}(B)$.

– If w is a prefix of $v_1 \dots v_n$, then there are $j : 1 \leq j \leq n$, $v'_j, v''_j \in \Sigma^*$ such that $w = v_1 \dots v_{j-1} v'_j$, $x = v''_j v_{j+1} \dots v_n y$, and $v_j = v'_j v''_j$. So $v'_j \in \mathrm{pref}(B)$ and $[v'_j]_B \in \mathrm{Vis}(w)$. From Proposition 15, we know that $\mathrm{Vis}(w) \subseteq \mathrm{Vis}(w')$. Therefore, $[v'_j]_B \in \mathrm{Vis}(w')$, so there is $z \in \mathrm{pref}(B)$ such that $w' \in B^*z$ and $[v'_j]_B = [z]_B$. From this, we deduce that $zv''_j \in B$ and $w'x \in B^*(zv''_j)v_{j+1} \dots v_n y \subseteq B^*\mathrm{pref}(B)$.
– If $v_1 \dots v_n$ is a prefix of w, then there is $x' \in \Sigma^*$ such that $w = v_1 \dots v_n x'$ and $x'x = y$. So $[x']_B \in \mathrm{Vis}(w)$. From Proposition 15, we know that $\mathrm{Vis}(w) \subseteq \mathrm{Vis}(w')$. Therefore, there is $z \in \mathrm{pref}(B)$ such that $w' \in B^*z$ and $[x']_B = [z]_B$. From the fact that $x'x \sim_B zx$ and $x'x = y \in \mathrm{pref}(B)$, we deduce that $w'x \in B^*(zx) \subseteq B^*\mathrm{pref}(B)$.

If $wx, w'x \notin B^*\mathrm{pref}(B)$, then we are done.

Now suppose that $wx, w'x \in B^*\mathrm{pref}(B)$, we show that $S_{N_1}(wx) \simeq S_{N_1}(w'x)$.

In the following, we show that there is a morphism from $S_{N_1}(wx)$ to $S_{N_1}(w'x)$. The argument for the existence of a morphism from $S_{N_1}(w'x)$ to $S_{N_1}(wx)$ is symmetric.

Let $[v_1, \dots, v_i]$ be a non-root node in $S_{N_1}(wx)$. Then there is $x' \in B^*\mathrm{pref}(B)$ such that $wx = v_1 \dots v_i x'$. There are two situations:

– x is a suffix of x'. Then $w = v_1 \dots v_i y$ and $x' = yx$ for some $y \in \Sigma^*$. Therefore, $y \in B^*\mathrm{pref}(B)$, and we deduce that $[v_1, \dots, v_i]$ is a node in $S_{N_1}(w)$.
– x' is a suffix of x. Then $v_1 \dots v_i = wx''$ and $x''x' = x$ for some $x'' \in \Sigma^*$.

We define a mapping $\pi' : S_{N_1}(wx) \to S_{N_1}(w'x)$ as follows.

– For each node $[v_1, \dots, v_i]$ in $S_{N_1}(wx)$ such that $[v_1, \dots, v_i]$ is also a node in $S_{N_1}(w)$, let $\pi'([v_1, \dots, v_i]) = \pi([v_1, \dots, v_i])$. The definition is justified by the fact that $\pi([v_1, \dots, v_i])$ is a node in $S_{N_1}(w'x)$. The argument goes as follows: Let $y \in \Sigma^*$ such that $w = v_1 \dots v_i y$. Since $[v_1, \dots, v_i]$ is a node in $S_{N_1}(wx)$, it follows that $yx \in B^*\mathrm{pref}(B)$. Let $y' \in \Sigma^*$ such that $w' = \pi([v_1, \dots, v_i])y'$. From Proposition 15, we know that $\mathrm{Vis}(y) \subseteq \mathrm{Vis}(y')$. From the facts $yx \in B^*\mathrm{pref}(B)$ and $\mathrm{Vis}(y) \subseteq \mathrm{Vis}(y')$, we deduce that $y'x \in B^*\mathrm{pref}(B)$. Because $w'x = \pi([v_1, \dots, v_i])y'x$, it follows that $\pi([v_1, \dots, v_i])$ is a node in $S_{N_1}(w'x)$.
– For each node $[v_1, \dots, v_i]$ in $S_{N_1}(wx)$ such that $v_1 \dots v_i = wx''$ and $x''x' = x$ for some $x'', x' \in \Sigma^*$, there are $j : 1 \leq j \leq i$, $v_{j,1}, v_{j,2} \in \Sigma^*$ such that $w = v_1 \dots v_{j-1} v_{j,1}$, $v_j = v_{j,1} v_{j,2}$, and $x'' = v_{j,2} v_{j+1} \dots v_i$. Then $[v_1, \dots, v_{j-1}]$ is a node in $S_{N_1}(w)$. Let $\pi([v_1, \dots, v_{j-1}]) = [v'_1, \dots, v'_{j-1}]$. There is $y \in B^*\mathrm{pref}(B)$ such that $w' = v'_1 \dots v'_{j-1} y$ and $[y]_B = [v_{j,1}]_B$. Since $v_{j,1} v_{j,2} = v_j \in B$, we deduce that $yv_{j,2} \in B$, and $w'x = v'_1 \dots v'_{j-1} yx = v'_1 \dots v'_{j-1} yx''x' = v'_1 \dots v'_{j-1}(yv_{j,2})v_{j+1} \dots v_i x'$. Because $x' \in B^*\mathrm{pref}(B)$, we deduce that $[v'_1, \dots, v'_{j-1}, yv_{j,2}, v_{j+1}, \dots, v_i]$ is a node in $S_{N_1}(w'x)$, let $\pi'([v_1, \dots, v_i]) = [v'_1, \dots, v'_{j-1}, yv_{j,2}, v_{j+1}, \dots, v_i]$.

Now we show that π' is a morphism.

Preservation of the parent-child relationship:

For every pair of nodes $[v_1, \dots, v_{i-1}]$ and $[v_1, \dots, v_i]$ in $S_{N_1}(wx)$, we show that $\pi'([v_1, \dots, v_{i-1}])$ is the parent of $\pi'([v_1, \dots, v_i])$.

– If $[v_1, \dots, v_{i-1}]$ and $[v_1, \dots, v_i]$ are both nodes in $S_{N_1}(w)$, then $\pi'([v_1, \dots, v_{i-1}]) = \pi([v_1, \dots, v_{i-1}])$ and $\pi'([v_1, \dots, v_i]) = \pi([v_1, \dots, v_i])$ are both nodes in $S_{N_1}(w')$. So $\pi([v_1, \dots, v_{i-1}])$ is the parent of $\pi([v_1, \dots, v_i])$ in $S_{N_1}(w'x)$.

– If $[v_1,\ldots,v_{i-1}]$ and $[v_1,\ldots,v_i]$ satisfy that $wx'' = v_1\ldots v_{i-1}$ and $x = x''x'$ for some $x'', x' \in \Sigma^*$, then from the definition of π', we know that $\pi'([v_1,\ldots,v_{i-1}]) = [v'_1,\ldots,v'_{j-1}, yv_{j,2}, v_{j+1},\ldots,v_{i-1}]$ and $\pi'([v_1,\ldots,v_i]) = [v'_1,\ldots,v'_{j-1}, yv_{j,2}, v_{j+1},\ldots,v_i]$ with $j, v'_1,\ldots,v'_{j-1}, y, v_{j,2}$ satisfying the conditions specified in the definition of π'. Evidently, $[v'_1,\ldots,v'_{j-1}, yv_{j,2}, v_{j+1},\ldots,v_{i-1}]$ is the parent of $[v'_1,\ldots,v'_{j-1}, yv_{j,2}, v_{j+1},\ldots,v_i]$ in $S_{N_1}(w'x)$.
– If $[v_1,\ldots,v_{i-1}]$ and $[v_1,\ldots,v_i]$ satisfy that $v_1\ldots v_{i-1}$ is a prefix of w and w is a prefix of $v_1\ldots v_i$, then from the definition of π', we know that $\pi'([v_1,\ldots,v_{i-1}]) = \pi([v_1,\ldots,v_{i-1}]) = [v'_1,\ldots,v'_{i-1}]$, and $\pi'([v_1,\ldots,v_i]) = [v'_1,\ldots,v'_{i-1}, yv_{i,2}]$ with $y, v_{i,2}$ satisfying the conditions specified in the definition of π'. Evidently, $[v'_1,\ldots,v'_{i-1}]$ is the parent of $[v'_1,\ldots,v'_{i-1}, yv_{i,2}]$ in $S_{N_1}(w'x)$.

Preservation of the label:
Let $[v_1,\ldots,v_i]$ be a node in $S_{N_1}(wx)$. Then there is $x' \in B^*\mathrm{pref}(B)$ such that $wx = v_1\ldots v_i x'$.
– If w is a prefix of $v_1\ldots v_i$, then $\pi'([v_1,\ldots,v_i]) = [v'_1,\ldots,v'_{j-1}, yv_{j,2}, v_{j+1},\ldots,v_i]$, with $j, v'_1,\ldots,v'_{j-1}, y, v_{j,2}$ satisfying the conditions specified in the definition of π'. Because $w'x = v'_1\ldots v'_{j-1} y v_{j,2} v_{j+1}\ldots v_i x'$, we deduce that $[v_1,\ldots,v_i]$ and $\pi'([v_1,\ldots,v_i])$ are either labeled by $[x']_B$ (if $i < N_1$) or Vis(x') (if $i = N_1$), so they have the same label.
– If $v_1\ldots v_i$ is a prefix of w, then from the definition of π', we know that $\pi'([v_1,\ldots,v_i]) = \pi([v_1,\ldots,v_i]) = [v'_1,\ldots,v'_i]$. Let $w = v_1\ldots v_i y$ and $w' = v'_1\ldots v'_i y'$ for $y, y' \in \Sigma^*$.
  · If $i < N_1$, then $[y]_B$, the label of $[v_1,\ldots,v_i]$ in $S_{N_1}(w)$, is equal to $[y']_B$, the label of $[v'_1,\ldots,v'_i] = \pi([v_1,\ldots,v_i])$ in $S_{N_1}(w')$. From this, we know that $[yx]_B$, the label of $[v_1,\ldots,v_i]$ in $S_{N_1}(wx)$, and $[y'x]_B$, the label of $\pi'([v_1,\ldots,v_i]) = [v'_1,\ldots,v'_i]$ in $S_{N_1}(w'x)$, are the same.
  · If $i = N_1$, then Vis(y), the label of $[v_1,\ldots,v_i]$ in $S_{N_1}(w)$, is equal to Vis(y'), the label of $[v'_1,\ldots,v'_i] = \pi([v_1,\ldots,v_i])$ in $S_{N_1}(w')$. From Proposition 13 and the fact Vis$(y) =$ Vis(y'), we know that Vis$(yx) =$ Vis$(y'x)$. So Vis(yx), the label of $[v_1,\ldots,v_i]$ in $S_{N_1}(wx)$, and Vis$(y'x)$, the label of $\pi'([v_1,\ldots,v_i]) = [v'_1,\ldots,v'_i]$ in $S_{N_1}(w'x)$, are the same. □

Lemma 21 (Shrinking Lemma) *Given two B-related words $w, w' \in B^*\mathrm{pref}(B)$, and a strategy f for Attacker in the game $\mathcal{G}(A,B)$ starting from w, there exists a strong strategy f' for Attacker in the game $\mathcal{G}(A,B)$ starting from w' with the following property: Whatever the plays of Defender, by following f', in a finite number of rounds,*
– *either Attacker wins,*
– *or he drives the game from w' to a configuration w'_1 such that there exists a non-void prefix of a play consistent with f in the game $\mathcal{G}(A,B)$ starting from w, driving the game from w to a configuration w_1 which is B-related to w'_1.*

Before going into its proof, let us state the main consequence of this result.

Theorem 22 *Let $w, w' \in B^*\mathrm{pref}(B)$ be two B-related words. Then Attacker has a winning strategy in the game $\mathcal{G}(A,B)$ starting from w iff he has one in $\mathcal{G}(A,B)$ starting from w'.*

PROOF. By symmetry, it is sufficient to show that if *Attacker* has a winning strategy in the game $\mathcal{G}(A,B)$ starting from w, then *Attacker* also has a winning strategy in the game $\mathcal{G}(A,B)$ starting from w'.

Let f be a winning strategy of *Attacker* in the game $\mathcal{G}(A,B)$ starting from w.

Lemma 21 provides us with a strong strategy f' in the game $\mathcal{G}(A,B)$ starting from w', satisfying the following condition: Let *Attacker* start playing according to this strategy. Then in a finite number of rounds,
– either *Attacker* wins (that is what we want and the game stops here),
– or else, he drives the game to a word w'_1 such that there exists a non-void prefix of a play consistent with f in the game $\mathcal{G}(A,B)$ starting from w, driving the game to a new word w_1 which is B-related to w'_1.

The strategy f induces a winning strategy in the game $\mathcal{G}(A,B)$ starting from this new word w_1, we then start again the process with the words w_1 and w'_1, and so on.

Therefore, we obtain a sequence $w_1, w_2, \ldots, w_l, \ldots$ of words which are configurations of a prefix of a play consistent with f in the game $\mathcal{G}(A,B)$ starting from w. Let us note that for every $i \geq 1$, the two configurations w_i and w_{i+1} are separated by at least one round. In particular, when the game arrives at the word w_l, at least l rounds have been played. Because f is a winning strategy for *Attacker* in the game $\mathcal{G}(A,B)$ starting from w, from Observation 2 we know that there exists a number M_f such that *Attacker* wins for sure in less than M_f rounds in the game $\mathcal{G}(A,B)$

starting from w, no matter what *Defender* chooses. Therefore, l is bounded by M_f. This implies that our process stops after at most M_f applications of Lemma 21. Consequently, *Attacker* wins for sure no matter what *Defender* plays in the game $\mathcal{G}(A^+, B)$ starting from w', in other words, *Attacker* has a winning strong strategy in the game $\mathcal{G}(A, B)$ starting from w'. From Proposition 3, we conclude that *Attacker* has a winning strategy in the game $\mathcal{G}(A, B)$ starting from w'. □

For the proof of Lemma 21, we need the concept of waiting loops.

Definition 23 (Waiting Loop) *Let $w = w_1w_2w_3w_4$. Then w_3 is called a waiting loop of w_1 with respect to w_2 in w if the following three conditions hold,*
- $w_1 \in B^{\geq N_1}\mathrm{pref}(B) \setminus B^{\leq N_1-1}\mathrm{pref}(B)$ *(this condition ensures that the labels of all the non-leaf nodes in $S_{N_1}(w_1)$ are $\bot$ and all the leaves in $S_{N_1}(w_1)$ are of depth N_1 and labeled by Vis(y) for some $y \in B^*\mathrm{pref}(B)$),*
- *w_3 is nonempty,*
- *for every $y \in \mathrm{pref}(B)$ such that $w_1 \in B^*y$, Vis$(yw_2) =$ Vis(yw_2w_3).*

Proposition 24 *Suppose $w = w_1w_2w_3w_4$, and for every $y \in \mathrm{pref}(B)$ such that $w_1 \in B^*y$, it holds that Vis$(yw_2) =$ Vis(yw_2w_3). Then for every $z \in B^*\mathrm{pref}(B)$ such that $w_1 = v_1 \dots v_iz$ with $v_1, \dots, v_i \in B$, we have Vis$(zw_2) =$ Vis(zw_2w_3).*

PROOF. Let $z \in B^*\mathrm{pref}(B)$, $w_1 = v_1 \dots v_iz$, and $v_1, \dots, v_i \in B$.

In the following, we will show that $\mathrm{Vis}(zw_2) \subseteq \mathrm{Vis}(zw_2w_3)$. The proof of $\mathrm{Vis}(zw_2w_3) \subseteq \mathrm{Vis}(zw_2)$ is similar.

Suppose $z' \in \mathrm{pref}(B)$ and $zw_2 \in B^*z'$. We show that $[z']_B \in \mathrm{Vis}(zw_2w_3)$.

Let $zw_2 = v_1 \dots v_nz'$ for $n \in \mathbb{N}$ and $v_1, \dots, v_n \in B$.

If z' is a suffix of w_2, then $z = v_1 \dots v_{j-1}v'_j$, $w_2 = v''_jv_{j+1} \dots v_nz'$, and $v_j = v'_jv''_j$ for some $j : 1 \leq j \leq n$ and $v'_j, v''_j \in \Sigma^*$. It follows that $v'_j \in \mathrm{pref}(B)$ and $w_1 \in B^*z = B^*v_1 \dots v_{j-1}v'_j \subseteq B^*v'_j$. So $\mathrm{Vis}(v'_jw_2) = \mathrm{Vis}(v'_jw_2w_3)$ from the assumption. Therefore, $[z']_B \in \mathrm{Vis}(v'_jw_2) = \mathrm{Vis}(v'_jw_2w_3)$. Because $zw_2w_3 = v_1 \dots v_{j-1}(v'_jw_2w_3)$, we conclude that $[z']_B \in \mathrm{Vis}(zw_2w_3)$.

If w_2 is a suffix of z', then $z = v_1 \dots v_nz''$ and $z''w_2 = z'$ for some $z'' \in \Sigma^*$. It follows that $z'' \in \mathrm{pref}(B)$ and $w_1 \in B^*z = B^*(v_1 \dots v_n)z'' \subseteq B^*z''$. So $\mathrm{Vis}(z''w_2) = \mathrm{Vis}(z''w_2w_3)$ according to the assumption. Since $[z']_B \in \mathrm{Vis}(z''w_2)$, we have $[z']_B \in \mathrm{Vis}(z''w_2w_3)$. Because $zw_2w_3 = v_1 \dots v_n(z''w_2w_3)$, we conclude that $[z']_B \in \mathrm{Vis}(zw_2w_3)$. □

Proposition 25 *Let $w = w_1w_2w_3w_4$, and w_3 be a waiting loop of w_1 with respect to w_2 in w. Then for every $y \in B^{\geq N_1}\mathrm{pref}(B) \setminus B^{\leq N_1-1}\mathrm{pref}(B)$ such that $w_1 = v_1 \dots v_iy$ for $v_1, \dots, v_i \in B$, it holds that w_3 is a waiting loop of y with respect to w_2 in $yw_2w_3w_4$.*

PROOF. Suppose $w = w_1w_2w_3w_4$, w_3 is a waiting loop of w_1 with respect to w_2 in w, $y \in B^{\geq N_1}\mathrm{pref}(B) \setminus B^{\leq N_1-1}\mathrm{pref}(B)$, and $w_1 = v_1 \dots v_iy$ for $v_1, \dots, v_i \in B$.

To prove that w_3 is a waiting loop of y with respect to w_2 in $yw_2w_3w_4$, it is sufficient to show that for every $z \in \mathrm{pref}(B)$ such that $y \in B^*z$, $\mathrm{Vis}(zw_2) = \mathrm{Vis}(zw_2w_3)$.

Let $z \in \mathrm{pref}(B)$ such that $y \in B^*z$. Because $w_1 = v_1 \dots v_iy \in B^*y$, we have $w_1 \in B^*z$. From the assumption that w_3 is a waiting loop of w_1 with respect to w_2 in w, we deduce that $\mathrm{Vis}(zw_2) = \mathrm{Vis}(zw_2w_3)$. □

Lemma 26 (Waiting Loops and B-Relation) *Let $w = w_1w_2w_3w_4$, and w_3 be a waiting loop of w_1 with respect to w_2 in w. Then w is B-related to every word in $w_1w_2w_3^*w_4$.*

PROOF.

Let $w = w_1w_2w_3w_4$, and w_3 be a waiting loop of w_1 with respect to w_2 in w.

We first prove the following claim.

Claim. For each $i \geq 0$ and every $y \in B^*\mathrm{pref}(B)$ such that $w_1 = v_1 \dots v_jy$ and $[v_1, \dots, v_j]$ is a node in $S_{N_1}(w_1)$, it holds that $\mathrm{Vis}(yw_2) = \mathrm{Vis}(yw_2w_3) = \mathrm{Vis}(yw_2w_3^i)$.

We prove this claim by an induction on i.

Induction base $i = 0$: From Proposition 24, we know that $\text{Vis}(yw_2) = \text{Vis}(yw_2w_3)$.

Induction step $i \geq 1$:

By the induction hypothesis, $\text{Vis}(yw_2) = \text{Vis}(yw_2w_3^{i-1})$. Then from Proposition 13, we know that $\text{Vis}(yw_2w_3) = \text{Vis}(yw_2w_3^{i-1}w_3) = \text{Vis}(yw_2w_3^i)$. Therefore, $\text{Vis}(yw_2) = \text{Vis}(yw_2w_3^i)$.

The proof of the claim is complete.

In order to prove that w is B-related to every word in $w_1w_2w_3^*w_4$, it is sufficient to prove that $w_1w_2 \leftrightarrow_B w_1w_2w_3^i$ for every $i \geq 1$, since $\leftrightarrow_B$ is a right congruence.

From the claim, it follows that for every $i \geq 1$, $\text{Vis}(w_1w_2) = \text{Vis}(w_1w_2w_3^i)$. So $w_1w_2 \in B^*\text{pref}(B)$ iff $w_1w_2w_3^i \in B^*\text{pref}(B)$.

If $w_1w_2, w_1w_2w_3^i \notin B^*\text{pref}(B)$, then we are done.

So in the following, we assume that $w_1w_2, w_1w_2w_3^i \in B^*\text{pref}(B)$.

It remains to prove that $S_{N_1}(w_1w_2) \simeq S_{N_1}(w_1w_2w_3^i)$, in order to show that $w_1w_2 \leftrightarrow_B w_1w_2w_3^i$.

Because w_3 is a waiting loop of w_1 with respect to w_2 in w, it follows that $w_1 \in B^{\geq N_1}\text{pref}(B) \setminus B^{\leq N_1-1}\text{pref}(B)$. Then every leaf in $S_{N_1}(w_1)$, $S_{N_1}(w_1w_2)$, and $S_{N_1}(w_1w_2w_3^i)$ must be of depth exactly N_1. So $S_{N_1}(w_1w_2)$(respectively $S_{N_1}(w_1w_2w_3^i)$) is obtained from $S_{N_1}(w_1)$ through the following two-step procedure:

(i) Replace the label of each leaf $[v_1, \ldots, v_{N_1}]$, say $\text{Vis}(y)$ such that $w_1 = v_1 \ldots v_{N_1}y$ and $y \in B^*\text{pref}(B)$, by $\text{Vis}(yw_2)$ (respectively $\text{Vis}(yw_2w_3^i)$).

(ii) Remove all the subtrees such that all the leaves $[v_1, \ldots, v_{N_1}]$ in the subtree are labeled by the empty set. In other words, remove all the subtrees in which all the leaves $[v_1, \ldots, v_{N_1}]$ such that $w_1 = v_1 \ldots v_{N_1}y$ satisfy that $\text{Vis}(yw_2) = \emptyset$ (respectively $\text{Vis}(yw_2w_3^i) = \emptyset$).

From the claim and the above constructions of $S_{N_1}(w_1w_2)$ and $S_{N_1}(w_1w_2w_3^i)$ from $S_{N_1}(w_1)$, it follows that $S_{N_1}(w_1w_2) \simeq S_{N_1}(w_1w_2w_3^i)$ (in fact, they are isomorphic). □

Lemma 27 (Existence of waiting loops, Version 1) *For every $w \in \Sigma^*$ and every prefix w_1 of w such that $w_1 \in B^{\geq N_1}\text{pref}(B) \setminus B^{\leq N_1-1}\text{pref}(B)$ and the length of $w_1^{-1}w$ is at least $2^{N_B 2^{N_B}} + 1$, there exists a decomposition of w into $w_1w_2w_3w_4$ such that w_3 is a waiting loop of w_1 with respect to w_2 in w.*

PROOF. Let $w_1 \in B^{\geq N_1}\text{pref}(B) \setminus B^{\leq N_1-1}\text{pref}(B)$ be a prefix of w, and $w_1^{-1}w = \sigma_1 \ldots \sigma_r$ with $r \geq 2^{N_B 2^{N_B}} + 1$ and $\sigma_1, \ldots, \sigma_r \in \Sigma$.

In the following, we will show that there are $i, j : 1 \leq i < j \leq r$ such that for each $y \in \text{pref}(B)$ such that $w_1 \in B^*y$, $\text{Vis}(y\sigma_1 \ldots \sigma_i) = \text{Vis}(y\sigma_1 \ldots \sigma_j)$. If this holds, let $w_2 = \sigma_1 \ldots \sigma_i$, $w_3 = \sigma_{i+1} \ldots \sigma_j$, and $w_4 = \sigma_{j+1} \ldots \sigma_r$, then w_3 is a waiting loop of w_1 with respect to w_2 in w.

Let $[v_{1,1}, \ldots, v_{1,i_1}], \ldots, [v_{l,1}, \ldots, v_{l,i_l}]$ be a collection of the nodes in $S(w_1)$ and $y_1, \ldots, y_l \in B^*\text{pref}(B)$ such that

- for every $j : 1 \leq j \leq l$, $w_1 = v_{j,1} \ldots v_{j,i_j} y_j$;
- for every $y \in B^*\text{pref}(B)$ such that $w_1 \in B^*y$, there is $l' : 1 \leq l' \leq l$ such that $\text{Vis}(y) = \text{Vis}(y_{l'})$.

It is not hard to see that such a collection of nodes with $l \leq 2^{N_B}$ exists.

Consider the following sequence of tuples

$$(\text{Vis}(y_1\sigma_1), \ldots, \text{Vis}(y_l\sigma_1)), \ldots, (\text{Vis}(y_1\sigma_1 \ldots \sigma_r), \ldots, \text{Vis}(y_l\sigma_1 \ldots \sigma_r)).$$

For each $i : 1 \leq i \leq r$, it holds that $(\text{Vis}(y_1\sigma_1 \ldots \sigma_i), \ldots, \text{Vis}(y_l\sigma_1 \ldots \sigma_i)) \in \left(2^{\mathcal{E}(B)}\right)^l$. Because $l \leq 2^{N_B}$ and $r \geq 2^{N_B 2^{N_B}} + 1$, it follows that there are $i, j : 1 \leq i < j \leq r$ such that

$$(\text{Vis}(y_1\sigma_1 \ldots \sigma_i), \ldots, \text{Vis}(y_l\sigma_1 \ldots \sigma_i)) = (\text{Vis}(y_1\sigma_1 \ldots \sigma_j), \ldots, \text{Vis}(y_l\sigma_1 \ldots \sigma_j)).$$

In the following, we will complete the proof by showing that for every $y \in \text{pref}(B)$ such that $w_1 \in B^*y$, it holds that $\text{Vis}(y\sigma_1 \ldots \sigma_i) = \text{Vis}(y\sigma_1 \ldots \sigma_j)$.

Suppose $y \in \text{pref}(B)$ and $w_1 \in B^*y$. Then there is $l' : 1 \leq l' \leq l$ such that $\text{Vis}(y) = \text{Vis}(y_{l'})$. From Proposition 13, it follows that $\text{Vis}(y\sigma_1 \ldots \sigma_i) = \text{Vis}(y_{l'}\sigma_1 \ldots \sigma_i) = \text{Vis}(y_{l'}\sigma_1 \ldots \sigma_j) = \text{Vis}(y\sigma_1 \ldots \sigma_j)$. □

Similarly, we can show the following result.

Lemma 28 (Existence of waiting loops, Version 2) *Let $w = c_1c_2\dots c_n$ be a decomposition of w into n factors. Let n_1 be such that $c_1\dots c_{n_1} \in B^{\geq N_1}\mathrm{pref}(B) \setminus B^{\leq N_1-1}\mathrm{pref}(B)$ and $n - n_1 \geq 2^{N_B 2^{N_B}} + 1$. Then there is a decomposition of w into $w_1w_2w_3w_4$ such that w_3 is a waiting loop of w_1 with respect to w_2 in w, and for every $i = 1, 2, 3, 4$, $w_i = c_{n_{i-1}+1}\dots c_{n_i}$, where $0 = n_0 < n_1 \leq n_2 < n_3 \leq n_4 = n$.*

Now we are ready to prove Lemma 21.

PROOF. [of Lemma 21]

Suppose $w \leftrightarrow_B w'$. Then there is a morphism $\pi : S_{N_1}(w') \to S_{N_1}(w)$.

We describe round by round a strong strategy f' in the game $\mathcal{G}(A,B)$ starting from w'.

During this description, we shall use f as an *oracle* to which we provide choices of *Defender* and which tells us what f suggests for *Attacker*'s choices.

There are three stages in the strategy f'.

At first we remark that if during the three stages, the game arrives at a configuration belonging to $\overline{B^*\mathrm{pref}(B)}$, then *Attacker* wins and the description of f' ends there.

1. Informally, the first stage starts at the beginning and goes on until the word obtained by concatenating all choices of *Attacker* is sufficiently long, actually in $B^{\geq N_1}\mathrm{pref}(B) \setminus B^{\leq N_1-1}\mathrm{pref}(B)$.

Let $N_1' = k \max\left\{k+2, 2^{N_B 2^{N_B}} + 1\right\}$.

Precisely, the first stage consists of kn_1 rounds of the game $(u_1,v_1),(u_2,v_2),\dots,(u_{kn_1},v_{kn_1})$, where n_1 is such that $(k+1)(n_1-1) < N_1 + N_1' + 1 \leq (k+1)n_1$. It follows that

$$n_1 = \ulcorner (N_1+N_1'+1)/(k+1) \urcorner = \ulcorner ((k(2k+3))/(k+1))\ \max\left\{k+2, 2^{N_B 2^{N_B}} + 1\right\} + 1/(k+1) \urcorner.$$

Note that

$$\begin{aligned} N_1 - kn_1 &\geq 2k(k+1)\max\left\{k+2, 2^{N_B 2^{N_B}}+1\right\} - k^2\frac{2k+3}{k+1}\max\left\{k+2, 2^{N_B 2^{N_B}}+1\right\} - \frac{k}{k+1} - k \\ &= \frac{k(k+2)}{k+1}\max\left\{k+2, 2^{N_B 2^{N_B}}+1\right\} - \frac{k}{k+1} - k \\ &= k\max\left\{k+2, 2^{N_B 2^{N_B}}+1\right\} + \frac{k}{k+1}\left(\max\left\{k+2, 2^{N_B 2^{N_B}}+1\right\} - (k+2)\right) \\ &\geq k\max\left\{k+2, 2^{N_B 2^{N_B}}+1\right\} = N_1'. \end{aligned}$$

We also would like to remark that $k+2$ in $\max\{k+2, 2^{N_B 2^{N_B}}+1\}$ is used to get the last inequality above. The inequality $N_1 - kn_1 \geq N_1'$ guarantees that in the game $\mathcal{G}(A,B)$ starting from w', *Attacker* is able to follow f by utilizing the morphism π in this stage and Stage 2 below.

During the kn_1 rounds of the game starting from w', *Attacker* utilizes the morphism π and follows f as follows.

Suppose $f(w,\varepsilon) = u_1 \in A$. Then in the first round of the game $\mathcal{G}(A,B)$ starting from w', *Attacker* just follows f and chooses u_1. Let v_1' be the choice of *Defender* in the first round of the game $\mathcal{G}(A,B)$ starting from w'.

If w' is completely erased by v_1', then there are $x, y \in \Sigma^*$ such that $v_1' = w'x$ and $u_1 = xy$. So the game $\mathcal{G}(A,B)$ starting from w' reaches the configuration $(v_1')^{-1}(w'u_1) = y$. Because $\pi(\varepsilon) = \varepsilon$ and π preserves the labels of nodes, we deduce that $[w']_B = [w]_B$. Therefore, $wx \in B$ as well, since $w'x = v_1' \in B$. Suppose in the first round of the game $\mathcal{G}(A,B)$ starting from w, *Attacker* has followed f and chosen u_1, and *Defender* has chosen wx. Then the game $\mathcal{G}(A,B)$ starting from w reaches the configuration $(wx)^{-1}(wu_1) = y$. Therefore, the game $\mathcal{G}(A,B)$ starting from w' and the game $\mathcal{G}(A,B)$ starting from w reach the same configuration after the first round. Consequently, in this situation, after the first round of the game $\mathcal{G}(A,B)$ starting from w', *Attacker* is able to completely follow f, so the description of f' ends here.

If w' is not completely erased by v_1', then the description of f' continues.

For the general situation, suppose that i rounds (where $1 \le i < kn_1$) have been played in the game $\mathcal{G}(A, B)$ starting from w', and in these i rounds, *Attacker* has followed the strategy f by utilizing the morphism π. In addition, w' has not been erased completely after the i rounds.

Let $(u_1, v'_1) \dots (u_i, v'_i)$ be the choices of *Attacker* and *Defender* in these first i rounds of the game $\mathcal{G}(A, B)$ starting from w', then $[v'_1, \dots, v'_i]$ is a node in $S_{N_1}(w')$. So there are $v_1, \dots, v_i \in B$ such that $\pi([v'_1, \dots, v'_i]) = [v_1, \dots, v_i]$. In the first i rounds of the game $\mathcal{G}(A, B)$ starting from w', *Attacker* has followed the strategy f in the game $\mathcal{G}(A, B)$ starting from w by utilizing the morphism π. Therefore, $(u_1, v_1) \dots (u_i, v_i)$ is a prefix of a play consistent with f in the game $\mathcal{G}(A, B)$ starting from w. Let $u_{i+1} = f(w, (u_1, v_1) \dots (u_i, v_i))$. Then in the strategy f', we let *Attacker* choose u_{i+1} in the $(i+1)$-st round of the game $\mathcal{G}(A, B)$ starting from w'. Let $v'_{i+1} \in B$ be the choice of *Defender* in the $(i+1)$-st round of the game $\mathcal{G}(A, B)$ starting from w'.

If w' is completely erased by $v'_1 \dots v'_i v'_{i+1}$, since w' is not completely erased by $v'_1 \dots v'_i$, it follows that there are $x', y' \in \Sigma^*$ such that $w' = v'_1 \dots v'_i x'$, $v'_1 \dots v'_i v'_{i+1} = w'y'$, and $v'_{i+1} = x'y'$. Then after the $(i+1)$ rounds, the game $\mathcal{G}(A, B)$ starting from w' reaches the configuration $(y')^{-1}(u_1 \dots u_{i+1})$. From the above discussion, we know that $\pi([v'_1, \dots, v'_i]) = [v_1, \dots, v_i]$. Let $x = (v_1 \dots v_i)^{-1} w$. Then $[x']_B = [x]_B$. So $xy' \in B$, since $x'y' = v'_{i+1} \in B$. Therefore, $(u_1, v_1) \dots (u_i, v_i)(u_{i+1}, xy')$ is a prefix of a play consistent with f in the game $\mathcal{G}(A, B)$ starting from w. Then after $i+1$ rounds, the game $\mathcal{G}(A, B)$ starting from w reaches the configuration $(y')^{-1}(u_1 \dots u_{i+1})$. Therefore, the game $\mathcal{G}(A, B)$ starting from w' and the game $\mathcal{G}(A, B)$ starting from w reach the same configuration after the $i+1$ rounds. Consequently, in this situation, after the $i+1$ rounds of the game $\mathcal{G}(A, B)$ starting from w', *Attacker* is able to completely follow f, so the description of f' ends here.

If w' is not completely erased by $v'_1 \dots v'_i v'_{i+1}$, then let $i := i + 1$, the description of f' continues.

If after the kn_1 rounds of the game $\mathcal{G}(A, B)$ starting from w', *Attacker* has not won yet and w' has not been completely erased, then we go to Stage 2. Note that the choices of *Defender* in the kn_1 rounds of the game $\mathcal{G}(A, B)$ starting from w' are $v'_1, \dots, v'_{kn_1}$, according to the above description.

Because $w'u_1 \dots u_{kn_1} \in B^*\mathrm{pref}(B)$ (otherwise, *Attacker* wins), it follows from Proposition 12 that $w'u_1 \dots u_{kn_1} \in B^{\ge(k+1)n_1}\mathrm{pref}(B) \backslash B^{\le(k+1)n_1-1}\mathrm{pref}(B)$. From the fact that $(k+1)n_1 \ge N_1 + N'_1 + 1$, we deduce that $w'u_1 \dots u_{kn_1} \in B^{\ge N_1+N'_1+1}\mathrm{pref}(B) \setminus B^{\le N_1+N'_1}\mathrm{pref}(B)$.

2. In Stage 2, *Attacker* still follows f by utilizing the morphism π until a *waiting loop* is found.

The description of f' in Stage 2 is the same as that in Stage 1. The description of f' in Stage 2 ends in the (kn_2)-th round such that

(i) either w' has been completely erased after the kn_2 rounds,

(ii) or else, there exists $n'_2 : n_1 \le n'_2 < n_2$ such that $u_{kn'_2+1} \dots u_{kn_2}$ is a waiting loop of $w'u_1 \dots u_{kn_1}$ with respect to $u_{kn_1+1} \dots u_{kn'_2}$ in $(w'u_1 \dots u_{kn_1})(u_{kn_1+1} \dots u_{kn'_2})(u_{kn'_2+1} \dots u_{kn_2})(\varepsilon)$.

Because $N_1 - kn_1 \ge N'_1 \ge k(2^{N_B 2^{N_B}} + 1)$, from Lemma 28 (where we take each c_i as a concatenation of k words u_j), we know that by following the strategy f', in the game $\mathcal{G}(A, B)$ starting from w', a number n_2 exists such that $kn_1 \le kn_2 \le kn_1 + N'_1 \le N_1$ and n_2 satisfies the property stated above. Moreover, we choose n_2 to be minimal for the property.

If w' has not been completely erased after the kn_2 rounds, then let $v'_{kn_1+1}, \dots, v'_{kn_2}$ be the choices of *Defender* in Stage 2, and $\pi([v'_1, \dots, v'_{kn_2}]) = [v_1, \dots, v_{kn_2}]$, we go to Stage 3.

3. During Stage 3, *Attacker* no longer follows f. He plays the sequence $u_{kn'_2+1}, \dots, u_{kn_2}$ in loop until *Defender* erases w' completely, i.e., until the (n_3+1)-st round for some $n_3 : n_3 \ge kn_2$ such that w' is a prefix of $v'_1 \dots v'_{n_3} v'_{n_3+1}$ and $v'_1 \dots v'_{n_3}$ is a (proper) prefix of w', where $v'_{kn_2+1}, \dots, v'_{n_3+1}$ are all the choices of *Defender* after the (kn_2)-th round. It follows that there are $z'_1, z'_2 \in \Sigma^*$ such that $w' = v'_1 \dots v'_{n_3} z'_1$ and $v'_{n_3+1} = z'_1 z'_2$. Evidently, $z'_1 \in \mathrm{pref}(B)$. We would like to remark that because $B \subseteq \Sigma^+$, such a number $n_3 + 1$ exists.

Let us note that in the (n_3+1)-st round, *Attacker* may be inside the loop, i.e., he may be playing some u_r with $kn'_2 + 1 \le r < kn_2$. Then after the (n_3+1)-st round in the game $\mathcal{G}(A, B)$ starting from w', *Attacker* finishes the current loop. This drives the game $\mathcal{G}(A, B)$ starting from w' to some round kn_4. Let $v'_{n_3+2}, \dots, v'_{kn_4}$ be the choices of *Defender* from the (n_3+2)-nd round to the (kn_4)-th round. Because $kn_2 - kn'_2 \le kn_2 - kn_1 \le N'_1$, it follows that $kn_4 - n_3 \le N'_1$. Note that while *Attacker* is finishing his loop, starting from the (n_3+1)-st round, *Defender* erases the choices of *Attacker*, actually, the choices $u_1, \dots, u_{kn_1}$ of *Attacker* in Stage 1. Because $z'_1 \in \mathrm{pref}(B)$ and we know from Proposition 12 that $\mathrm{pref}(B)A^{kn_1} \cap B^{\leq (k+1)n_1-1}\mathrm{pref}(B) = \emptyset$, it follows that

$$z'_1 u_1 \dots u_{kn_1} \in B^{\geq (k+1)n_1}\mathrm{pref}(B) \setminus B^{\leq (k+1)n_1-1}\mathrm{pref}(B) \subseteq B^{\geq N_1+N'_1+1}\mathrm{pref}(B) \setminus B^{\leq N_1+N'_1}\mathrm{pref}(B).$$

Let $u = (v'_1 \dots v'_{kn_4})^{-1}(w' u_1 \dots u_{kn_1}) = (z'_2 v'_{n_3+2} \dots v'_{kn_4})^{-1}(u_1 \dots u_{kn_1}) = \left(v'_{n_3+1} \dots v'_{kn_4}\right)^{-1}(z'_1 u_1 \dots u_{kn_1})$. From the fact that $kn_4 - n_3 \leq N'_1$, it is deduced that $u \in B^{\geq N_1+1}\mathrm{pref}(B) \setminus B^{\leq N_1}\mathrm{pref}(B)$.

Now, the sequence of choices which have been made by *Attacker* and *Defender* in the game $\mathcal{G}(A, B)$ starting from w' is

$$(u_1, v'_1) \dots (u_{kn'_2}, v'_{kn'_2})(u_{kn'_2+1}, v'_{kn'_2+1}) \dots (u_{kn_2}, v'_{kn_2})(u_{kn'_2+1}, v'_{kn_2+1}) \dots (u_r, v'_{n_3+1}) \dots (u_{kn_2}, v'_{kn_4}).$$

Recall that in the end of Stage 2, kn_2 rounds have been played and $\pi([v'_1, \dots, v'_{kn_2}]) = [v_1, \dots, v_{kn_2}]$. Because $z'_1 = (v'_1 \dots v'_{n_3})^{-1}w' \in \mathrm{pref}(B)$ and $n_3 \geq kn_2$, it follows that $[z'_1]_B \in \mathrm{Vis}((v'_1 \dots v'_{kn_2})^{-1}w')$. From Proposition 15, we know that $\mathrm{Vis}((v'_1 \dots v'_{kn_2})^{-1}w') \subseteq \mathrm{Vis}((v_1 \dots v_{kn_2})^{-1}w)$. So $[z'_1]_B \in \mathrm{Vis}((v_1 \dots v_{kn_2})^{-1}w)$. Consequently, there are $\bar{v}_1, \dots, \bar{v}_m \in B$ such that $[(v_1 \dots v_{kn_2}\bar{v}_1 \dots \bar{v}_m)^{-1}w]_B = [z'_1]_B$. Now, let us consider the following prefix of a play consistent with f in the game $\mathcal{G}(A, B)$ starting from w,

$$(u_1, v_1) \dots (u_{kn_2}, v_{kn_2})(\bar{u}_1, \bar{v}_1) \dots (\bar{u}_m, \bar{v}_m),$$

where for each $j : 1 \leq j \leq m$, $\bar{u}_j = f(w, (u_1, v_1) \dots (u_{kn_2}, v_{kn_2})(\bar{u}_1, \bar{v}_1) \dots (\bar{u}_{j-1}, \bar{v}_{j-1}))$. Suppose $z_1 = (v_1 \dots v_{kn_2}\bar{v}_1 \dots \bar{v}_m)^{-1}w$. Then $z_1 z'_2 \in B$, since $z'_1 z'_2 = v'_{n_3+1} \in B$ and $[z_1]_B = [z'_1]_B$. After $kn_2 + m$ rounds in the game $\mathcal{G}(A, B)$ starting from w, the configuration $z_1 u_1 \dots u_{kn_2}\bar{u}_1 \dots \bar{u}_m$ is reached. Let $v_{n_3+1} = z_1 z'_2$. Then

$$\begin{gathered}(v_{n_3+1} v'_{n_3+2} \dots v'_{kn_4})^{-1}(z_1 u_1 \dots u_{kn_2}\bar{u}_1 \dots \bar{u}_m) = \\ (z'_2 v'_{n_3+2} \dots v'_{kn_4})^{-1}(u_1 \dots u_{kn_2}\bar{u}_1 \dots \bar{u}_m) = \\ \left((z'_2 v'_{n_3+2} \dots v'_{kn_4})^{-1}(u_1 \dots u_{kn_1})\right)(u_{kn_1+1} \dots u_{kn_2}\bar{u}_1 \dots \bar{u}_m) = \\ u u_{kn_1+1} \dots u_{kn_2}\bar{u}_1 \dots \bar{u}_m.\end{gathered}$$

Thus, we can obtain the following prefix of a play consistent with f in the game $\mathcal{G}(A, B)$ starting from w,

$$(u_1, v_1) \dots (u_{kn_2}, v_{kn_2})(\bar{u}_1, \bar{v}_1) \dots (\bar{u}_m, \bar{v}_m)(\bar{u}'_{n_3+1}, v_{n_3+1})(\bar{u}'_{n_3+2}, v'_{n_3+2}) \dots (\bar{u}'_{kn_4}, v'_{kn_4}),$$

where $\bar{u}'_{n_3+1} = f(w, (u_1, v_1) \dots (u_{kn_2}, v_{kn_2})(\bar{u}_1, \bar{v}_1) \dots (\bar{u}_m, \bar{v}_m))$ and for each $j : n_3 + 2 \leq j \leq kn_4$,

$$\bar{u}'_j = f\left(w, (u_1, v_1) \dots (u_{kn_2}, v_{kn_2})(\bar{u}_1, \bar{v}_1) \dots (\bar{u}_m, \bar{v}_m)(\bar{u}'_{n_3+1}, v_{n_3+1})(\bar{u}'_{n_3+2}, v'_{n_3+2}) \dots (\bar{u}'_{j-1}, v'_{j-1})\right).$$

Therefore, after $kn_2 + m + kn_4 - n_3$ rounds above in the game $\mathcal{G}(A, B)$ starting from w, the following configuration is reached,

$$\begin{gathered}(v_1 \dots v_{kn_2}\bar{v}_1 \dots \bar{v}_m v_{n_3+1} v'_{n_3+2} \dots v'_{kn_4})^{-1}(w u_1 \dots u_{kn_2}\bar{u}_1 \dots \bar{u}_m \bar{u}'_{n_3+1}\bar{u}'_{n_3+2} \dots \bar{u}'_{kn_4}) = \\ (v_{n_3+1} v'_{n_3+2} \dots v'_{kn_4})^{-1}(z_1 u_1 \dots u_{kn_2}\bar{u}_1 \dots \bar{u}_m \bar{u}'_{n_3+1}\bar{u}'_{n_3+2} \dots \bar{u}'_{kn_4}) = \\ u u_{kn_1+1} \dots u_{kn_2}\bar{u}_1 \dots \bar{u}_m \bar{u}'_{n_3+1} \dots \bar{u}'_{kn_4}.\end{gathered}$$

Let $\bar{u}'_{kn_4+1} =$

$$f(w, (u_1, v_1) \dots (u_{kn_2}, v_{kn_2})(\bar{u}_1, \bar{v}_1) \dots (\bar{u}_m, \bar{v}_m)(\bar{u}'_{n_3+1}, v_{n_3+1})(\bar{u}'_{n_3+2}, v'_{n_3+2}) \dots (\bar{u}'_{kn_4}, v'_{kn_4})),$$

in other words, $\bar{u}'_{kn_4+1}$ is the choice of *Attacker* by following the strategy f in the $(kn_2 + m + kn_4 - n_3 + 1)$-st round of the game $\mathcal{G}(A, B)$ starting from w.

Let us go back to the description of f' in the game $\mathcal{G}(A, B)$ starting from w'. We are in the $(kn_4 + 1)$-st round, and *Attacker* is going to play. Let u'_{kn_4+1} denote $\bar{u}_1 \dots \bar{u}_m \bar{u}'_{n_3+1} \dots \bar{u}'_{kn_4+1}$. Then in f', we define the choice of *Attacker* in the $(kn_4 + 1)$-st round to be u'_{kn_4+1} (recall that our goal is to define a strong strategy f'). Let v'_{kn_4+1} be the choice of *Defender* in the $(kn_4 + 1)$-st round.

The description of f' ends here.

From the above description of f', we deduce that in the game $\mathcal{G}(A,B)$ starting from w' by following f', the following configuration is reached,

$$w'_1 = \overbrace{((v'_{kn_4+1})^{-1}u)}^{\bar{w}_1}\overbrace{u_{kn_1+1}\dots u_{kn'_2}}^{\bar{w}_2}\overbrace{(u_{kn'_2+1}\dots u_{kn_2})\dots(u_{kn'_2+1}\dots u_{kn_2})}^{\bar{w}_3^*}\overbrace{\bar{u}_1\dots\bar{u}_m\bar{u}'_{n_3+1}\dots\bar{u}'_{kn_4+1}}^{\bar{w}_4}.$$

On the other hand, in the game $\mathcal{G}(A,B)$ starting from w, let the choice of *Defender* in the $(kn_2+m+kn_4-n_3+1)$-st round be $v'_{kn_4+1} \in B$, then by following the strategy f, the following configuration is reached,

$$w_1 = \underbrace{((v'_{kn_4+1})^{-1}u)}_{\bar{w}_1}\underbrace{u_{kn_1+1}\dots u_{kn'_2}}_{\bar{w}_2}\underbrace{u_{kn'_2+1}\dots u_{kn_2}}_{\bar{w}_3}\underbrace{\bar{u}_1\dots\bar{u}_m\bar{u}'_{n_3+1}\dots\bar{u}'_{kn_4+1}}_{\bar{w}_4}.$$

Recall that $u = (v'_1\dots v'_{kn_4})^{-1}(w'u_1\dots u_{kn_1}) = (z'_2 v'_{n_3+2}\dots v'_{kn_4})^{-1}(u_1\dots u_{kn_1})$ is a suffix of $u_1\dots u_{kn_1}$, moreover, $u \in B^{\geq N_1+1}\mathrm{pref}(B) \setminus B^{\leq N_1}\mathrm{pref}(B)$. Then it follows that $\bar{w}_1 = (v'_{kn_4+1})^{-1}u \in B^{\geq N_1}\mathrm{pref}(B) \setminus B^{\leq N_1-1}\mathrm{pref}(B)$. In addition, $\bar{w}_1$ satisfies that $w'u_1\dots u_{kn_1} = (v'_1\dots v'_{kn_4}v'_{kn_4+1})\bar{w}_1$. On the other hand, $\bar{w}_3 = u_{kn'_2+1}\dots u_{kn_2}$ is a waiting loop of $w'u_1\dots u_{kn_1}$ with respect to $\bar{w}_2 = u_{kn_1+1}\dots u_{kn'_2}$ in $(w'u_1\dots u_{kn_1})\bar{w}_2\bar{w}_3\bar{w}_4$ (see Stage 2 above). Then from Proposition 25, we know that $\bar{w}_3$ is a waiting loop of $\bar{w}_1$ with respect to $\bar{w}_2$ in $\bar{w}_1\bar{w}_2\bar{w}_3\bar{w}_4 = w_1$. According to Lemma 26, we conclude that w_1 is B-related to w'_1.

The proof of the lemma is complete. □

3.2. *Effective construction of the greatest solution*

From Theorem 22, every pair of B-related words either both belong to $\mathcal{C}(A,B)$ or both do not.

From the definition, $\overline{B^*\mathrm{pref}(B)}$ is an equivalence class of $\leftrightarrow_B$. The other equivalence classes of $\leftrightarrow_B$ are determined completely by the reduced N_1-visibility trees.

For each reduced N_1-visibility tree T, it is not hard to show the following facts:

- The equivalence class of $\leftrightarrow_B$ corresponding to T is regular and a finite automaton for this equivalence class can be effectively constructed from T.
- It is decidable whether the equivalence class of $\leftrightarrow_B$ corresponding to T is a subset of $\mathcal{C}(A,B)$ or does not intersect with $\mathcal{C}(A,B)$: Pick an arbitrary word w from the equivalence class and decide whether $w \in \mathcal{C}(A,B)$, whose decidability follows from Kunc's regularity proof ([12]).

Because there are only finitely many non-isomorphic reduced N_1-visibility trees, it follows that $\mathcal{C}(A,B)$ can be effectively constructed from A, B.

4. The case that B is a code with finite decoding delay

In this section, for the language inequality $XA \subseteq BX$, it is assumed that

A, B are regular, and B is a code with decoding delay $d \geq 0$.

Moreover, the set of words $B^*\mathrm{pref}(B) \setminus B^{d+1}\Sigma^*$ is called the set of *bottom configurations* of $\mathcal{G}(A,B)$, denoted by $\mathrm{Conf}_{\mathrm{bt}}(A,B)$, and the set of words $\mathrm{Conf}_{\mathrm{bt}}(A,B) \cap B^d\Sigma^* = (B^*\mathrm{pref}(B) \cap B^d\Sigma^*) \setminus B^{d+1}\Sigma^*$ is called *border configurations* of $\mathcal{G}(A,B)$, denoted by $\mathrm{Conf}_{\mathrm{bd}}(A,B)$.

4.1. *Reduction into a two-player one-counter reachability game*

In the following, we first observe that with the assumption that A, B are regular and B is a code with finite decoding delay, the game $\mathcal{G}(A,B)$ can be reduced to a two-player reachability game played on the transition graph of some one-counter machine. If A, B are finite, then the one-counter machine has finite state space and finitely-branching transition relation, so the effectiveness of the greatest solution follows from the well-known results on two-player games played on the transition graph of pushdown automata ([21,3,20]). Nevertheless, if A, B are infinite, then the one-counter machine has infinite state space and infinitely-branching transition relation, which goes beyond the scope of pushdown automata. We solve the problem by showing that the one-counter reachability game can be reduced further to one with finite state space and finitely-branching transition relation.

4.1.1. *From $\mathcal{G}(A, B)$ to a two-player one-counter reachability game*

We first state a property of codes with finite decoding delay, which is pertinent to the reduction of the game $\mathcal{G}(A, B)$ into a one-counter reachability game.

Lemma 29 *Let B be a code with decoding delay $d \geq 0$. Then for every $x \in B^{d+1}\Sigma^* \cap B^*\mathrm{pref}(B)$, there are $v \in B$ and $y \in B^d\Sigma^* \cap B^*\mathrm{pref}(B)$ such that $x = vy$. In addition, for every $v' \in B, z \in B^*\mathrm{pref}(B)$ such that $x = v'z$, it holds that $v = v'$ and $y = z$.*

PROOF.

Let $x \in B^{d+1}\Sigma^* \cap B^*\mathrm{pref}(B)$. Then there are $v_1, \ldots, v_{d+1} \in B$ and $x' \in \Sigma^*$ such that $x = v_1 \ldots v_{d+1}x'$.

Because $x \in B^*\mathrm{pref}(B)$, there is $x'' \in \Sigma^*$ such that $xx'' \in B^+$. Let $v \in B$ such that $xx'' \in vB^*$. Then $xx'' = v_1 \ldots v_{d+1}x'x'' = v_1(v_2 \ldots v_{d+1})(x'x'') \in vB^*$. From the definition of finite decoding delays, we deduce that $v_1 = v$ and $v_2 \ldots v_{d+1}x'x'' \in B^*$. Let $y = v_2 \ldots v_{d+1}x'$. Then $y \in B^*\mathrm{pref}(B) \cap B^d\Sigma^*$ and $x = vy$.

Suppose $v' \in B, z \in B^*\mathrm{pref}(B)$ such that $x = v'z$. In the following, we will show that $v = v'$ and $y = z$.

From the fact that $z \in B^*\mathrm{pref}(B)$, we know that there is $z' \in \Sigma^*$ such that $zz' \in B^+$. Because $v'(zz') = xz' = v_1 \ldots v_{d+1}x'z' = v_1(v_2 \ldots v_{d+1})x'z'$, it follows that $v_1(v_2 \ldots v_{d+1})x'z' \in v'B^*$. From the definition of finite decoding delays, we deduce that $v' = v_1 = v$ and $y = z$. $\square$

Corollary 30 *Let B be a code with decoding delay $d \geq 0$. Then for every $x \in B^*\mathrm{pref}(B) \cap B^{d+1}\Sigma^*$, there is a unique pair (i, y) such that $i \geq 1$, $y \in \mathrm{Conf}_{\mathrm{bd}}(A, B)$ (recall that $\mathrm{Conf}_{\mathrm{bd}}(A, B) = (B^*\mathrm{pref}(B) \cap B^d\Sigma^*) \setminus B^{d+1}\Sigma^*$), and $x \in B^iy$.*

PROOF. Suppose $x \in B^*\mathrm{pref}(B) \cap B^{d+1}\Sigma^*$. Then according to Lemma 29, there is a unique pair (v_1, y_1) such that $v_1 \in B$, $y_1 \in B^*\mathrm{pref}(B) \cap B^d\Sigma^*$, and $x = v_1y_1$.

If $y_1 \in B^{d+1}\Sigma^*$, then we can apply Lemma 29 to y_1 to get $v_2 \in B$ and $y_2 \in B^*\mathrm{pref}(B) \cap B^d\Sigma^*$ such that $y_1 = v_2y_2$, and so on.

Evidently this process will terminate. Finally we get $v_1, \ldots, v_i \in B$ and $y_i \in (B^*\mathrm{pref}(B) \cap B^d\Sigma^*) \setminus B^{d+1}\Sigma^* = \mathrm{Conf}_{\mathrm{bd}}(A, B)$ such that $x = v_1 \ldots v_iy_i$. So we get a pair (i, y_i) such that $i \geq 1$, $y_i \in \mathrm{Conf}_{\mathrm{bd}}(A, B)$, and $x \in B^iy_i$.

In addition, Lemma 29 guarantees that the words $v_1, \ldots, v_i \in B$ and $y_i \in \mathrm{Conf}_{\mathrm{bd}}(A, B)$ are unique. Therefore, there is one unique desired pair (i, y). $\square$

Definition 31 (Index and Remainder) *For each $x \in B^*\mathrm{pref}(B) \cap B^{d+1}\Sigma^*$, define the* index *of x, denoted by $\mathrm{idx}(x)$, and the* remainder *of x, denoted by $\mathrm{rmd}(x)$, as respectively the number $i \geq 1$ and the word $y \in \mathrm{Conf}_{\mathrm{bd}}(A, B)$ stated in Corollary 30. Moreover, if $x \in \mathrm{Conf}_{\mathrm{bt}}(A, B)$, i.e. $x \in B^*\mathrm{pref}(B) \setminus B^{d+1}\Sigma^*$, then $\mathrm{idx}(x) = 0$ and $\mathrm{rmd}(x) = x$ by convention.*

Now we illustrate how the game $\mathcal{G}(A, B)$ can be reduced to a one-counter game. From Proposition 4, we know that $\mathcal{C}(A, B) \subseteq B^*\mathrm{pref}(B)$. Therefore, in the game $\mathcal{G}(A, B)$, it is sufficient to consider the configurations belonging to $B^*\mathrm{pref}(B)$.

Lemma 32 *Let B be a code with decoding delay d and $x, y \in B^*\mathrm{pref}(B)$. If $\mathrm{idx}(x) = \mathrm{idx}(y)$ and $\mathrm{rmd}(x) = \mathrm{rmd}(y)$, then $x \in \mathrm{Win}_\beta(\mathcal{G}(A, B))$ iff $y \in \mathrm{Win}_\beta(\mathcal{G}(A, B))$.*

PROOF. If $\mathrm{idx}(x) = \mathrm{idx}(y)$ and $\mathrm{rmd}(x) = \mathrm{rmd}(y)$, then we know that $S(x) \cong S(y)$. From Proposition 11, we conclude that $x \in \mathrm{Win}_\beta(\mathcal{G}(A, B))$ iff $y \in \mathrm{Win}_\beta(\mathcal{G}(A, B))$. $\square$

Lemma 33 *Let $x \in B^*\mathrm{pref}(B)$ and $u \in A$ such that $xu \in B^*\mathrm{pref}(B)$. Then $\mathrm{rmd}(xu) = \mathrm{rmd}(\mathrm{rmd}(x)u)$ and $\mathrm{idx}(xu) = \mathrm{idx}(x) + \mathrm{idx}(\mathrm{rmd}(x)u)$.*

PROOF. If $\mathrm{idx}(x) = 0$ and $\mathrm{rmd}(x) = x$, then it is evident that the conclusion of the lemma holds.

In the following, we assume that $\mathrm{idx}(x) = i > 0$.

Let $x' = \mathrm{rmd}(x)$. Then $x' \in \mathrm{Conf}_{\mathrm{bd}}(A, B)$ and $x = v_1 \dots v_i x'$ for $v_1, \dots, v_i \in B$.

Because $xu \in B^*\mathrm{pref}(B)$, there are $y \in \Sigma^*$ and $v \in B$ such that $xuy \in vB^*$. From $xuy = v_1 \dots v_i x'uy$, we have $v_1 v_2 \dots v_i x'uy \in vB^*$. According to the fact that $x' \in B^d\Sigma^*$ and the definition of finite decoding delays, we deduce that $v_1 = v$ and $v_2 \dots v_i x'uy \in B^*$. Then there is $v' \in B$ such that $v_2 \dots v_i x'uy \in v'B^*$. By a similar argument, we deduce that $v' = v_2$ and $v_3 \dots v_i x'uy \in B^*$, and so on. At last, we deduce that $x'uy \in B^*$. This implies that $x'u \in B^*\mathrm{pref}(B) \cap B^d\Sigma^*$.

From the fact that $xu = v_1 \dots v_i(x'u)$ and the proof of Corollary 30, we know that $v_1, \dots, v_i$ are exactly the first i words from B to cut from the beginning of xu, in order to get the remainder of xu. Therefore, $\mathrm{rmd}(xu) = \mathrm{rmd}(x'u)$ and $\mathrm{idx}(xu) = i + \mathrm{idx}(x'u) = \mathrm{idx}(x) + \mathrm{idx}(\mathrm{rmd}(x)u)$. □

From Lemma 32 and Lemma 33, we reduce the game $\mathcal{G}(A, B)$ into a two-player one-counter reachability game, denoted $G = (V, W, \rightarrow)$, as follows,

– V is the set of game positions for *Attacker*,

$$V = \{(x, 0, \alpha) \mid x \in \mathrm{Conf}_{\mathrm{bt}}(A, B)\} \cup \{(x, i, \alpha) \mid x \in \mathrm{Conf}_{\mathrm{bd}}(A, B), i > 0\}.$$

– W is the set of game positions for *Defender*,

$$W = \{(\bot, \beta)\} \cup \{(x, 0, \beta) \mid x \in \mathrm{Conf}_{\mathrm{bt}}(A, B)\} \cup \{(x, i, \beta) \mid x \in \mathrm{Conf}_{\mathrm{bd}}(A, B), i > 0\}.$$

– $\rightarrow \subseteq V \times W \cup W \times V$ is defined as follows.

Let $(x, i, \alpha) \in V$, $(y, j, \beta) \in W$. Then

· $(x, i, \alpha) \rightarrow (\bot, \beta)$ iff there exists some $u \in A$ such that $xu \notin B^*\mathrm{pref}(B)$,

· $(x, i, \alpha) \rightarrow (y, j, \beta)$ iff there exists some $u \in A$ such that $y = \mathrm{rmd}(xu)$ and $j = i + \mathrm{idx}(xu)$.

Let $(x, i, \beta) \in W$, $(y, j, \alpha) \in V$. Then $(x, i, \beta) \rightarrow (y, j, \alpha)$ iff one of the following conditions holds,

· $i > 0, j = i - 1, y = x$.

· $i = 0$, $j = 0$ and there is $v \in B$ such that $x = vy$.

There are no arcs out of $(\bot, \beta)$.

The *dead points* of G are $(\bot, \beta)$ or those vertices $(x, 0, \beta) \in W$ without successors, which happens when $x \in \mathrm{pref}(B) \setminus B^+\mathrm{pref}(B)$.

Each play of the reachability game G starts from some vertex in $V \cup W$, and goes as follows: If the game reaches some vertex $(x, i, \alpha) \in V$, then *Attacker* selects a successor $(y, j, \beta) \in W$ of (x, i, α) and the game continues on (y, j, β). Similarly for *Defender* when the game reaches a vertex in W.

Attacker wins a play if some dead point in W (thus *Defender* is not able to move) is reached, and *Defender* wins every infinite play.

The winning strategies and regions can be defined in a standard way, similarly to those for parity games ([16]), e.g. a winning strategy for *Attacker* in the game G starting from (x, i, α) is a partial function f from $(VW)^*V$ to W such that for every prefix of a play, say $(x_0, j_0, \alpha)(x_1, j_1, \beta)(x_2, j_2, \alpha) \dots (x_{2k}, j_{2k}, \alpha)$ (where $x_0 = x$, $j_0 = i$, and $k \geq 0$), consistent with f, that is, for every $r : 0 \leq r < k$, $f((x_0, j_0, \alpha)(x_1, j_1, \beta) \dots (x_{2r}, j_{2r}, \alpha)) = (x_{2r+1}, j_{2r+1}, \beta)$, it holds that $f((x_0, j_0, \alpha)(x_1, j_1, \beta) \dots (x_{2k}, j_{2k}, \alpha))$ is defined and is a successor of (x_{2k}, j_{2k}, α) in G.

The winning region of *Attacker* and *Defender* are denoted as respectively $\mathrm{Win}_\alpha(G)$ and $\mathrm{Win}_\beta(G)$.

From Lemma 32, it follows that the winning regions of G correspond to those of $\mathcal{G}(A, B)$ as follows.

Lemma 34 *For each $x \in B^*\mathrm{pref}(B)$,*

$$x \in \mathrm{Win}_\beta(\mathcal{G}(A, B)) \textit{ iff } (\mathrm{rmd}(x), \mathrm{idx}(x), \alpha) \in \mathrm{Win}_\beta(G).$$

For each $(x, i, p) \in V \cup W$ (where $p \in \{\alpha, \beta\}$), let $\mathrm{suc}((x, i, p))$ denote the set of successors of (x, i, p) in G. The following result shows some regularity of the structure of the transition graph of G.

Lemma 35 *Let $x \in \mathrm{Conf}_{\mathrm{bd}}(A, B)$ and $j = i + r$ with $r > 0$. Then the mapping $\varphi : \mathrm{suc}((x, i, \alpha)) \rightarrow \mathrm{suc}((x, j, \alpha))$ defined by $\varphi((\bot, \beta)) = (\bot, \beta)$ and $\varphi((y, k, \beta)) = (y, k + r, \beta)$ for each $(y, k, \beta) \in \mathrm{suc}((x, i, \alpha))$ is a bijection.*

Remark 36 *If A, B are finite, then G is a one-counter reachability game with finite state space and finitely-branching transition relation, i.e. a pushdown game with unary pushdown alphabet. From the classical results on pushdown games [21,3,20], it follows that the winning regions of G are regular and can be constructed effectively.*

Nevertheless, if A, B are infinite, then G is a game with infinite state space and infinitely-branching transition relation, which goes beyond the scope of pushdown automata. But we can still reduce the game G to a game with the finite state space and the finitely-branching transition relation.

4.1.2. *Making the state space finite and the transition relation finitely-branching*

In this subsection, we first show that the right congruence $\approx_S$ over the state space of G, that is, $\mathrm{Conf}_{\mathrm{bt}}(A, B)$, satisfies that the quotient of $\mathrm{Conf}_{\mathrm{bt}}(A, B)$ with respect to $\approx_S$ is finite. Then we show how to trim the transition relation into a finitely-branching one.

It is easy to see that the strategy tree $S(x)$ for each $x \in \mathrm{Conf}_{\mathrm{bt}}(A, B)$ has depth at most d (the root has depth 0).

Lemma 37 *For each $x \in \mathrm{Conf}_{\mathrm{bt}}(A, B)$, no two distinct leaves of $S(x)$ have the same label.*

PROOF. To the contrary, suppose that there are two distinct leaves $[v_1, \dots, v_i]$ and $[v'_1, \dots, v'_j]$ of $S(x)$ such that $x = v_1 \dots v_i y = v'_1 \dots v'_j z$, $[y]_B = [z]_B$, and $y, z \in \mathrm{pref}(B) \setminus B^+\mathrm{pref}(B)$.

Because $y, z \in \mathrm{pref}(B)$ and $y \sim_B z$, it follows that there is $u \in \Sigma^*$ such that $yu, zu \in B$.

Then $xu = v_1 \dots v_i(yu) = v'_1 \dots v'_j(zu)$. Because B is a code, $xu \in B^*$ has a unique decomposition into words in B, so we have that $i = j$, $v_\ell = v'_\ell$ for each $\ell \leq i$, and $y = z$, a contradiction to the distinctness of $[v_1, \dots, v_i]$ and $[v'_1, \dots, v'_j]$. $\square$

From Lemma 37, we know that the number of leaves of $S(x)$ for every $x \in \mathrm{Conf}_{\mathrm{bt}}(A, B)$ is bounded by N_B. Because the depth of $S(x)$ is bounded by d, it follows that the number of nodes in $S(x)$ is bounded by $(d + 1)N_B$. Thus the number of non-isomorphic strategy trees for words in $\mathrm{Conf}_{\mathrm{bt}}(A, B)$ is bounded by a number, say N_S, which depends on d and N_B.

For each $x \in \mathrm{Conf}_{\mathrm{bt}}(A, B)$, let $[x]_S$ denote the equivalence class of $\approx_S$ containing x. The number of equivalence classes of $\approx_S$ on $\mathrm{Conf}_{\mathrm{bt}}(A, B)$ is bounded by N_S.

From Proposition 10, it is easy to deduce the following fact.

Proposition 38 *Suppose $x, y \in \mathrm{Conf}_{\mathrm{bt}}(A, B)$, $x \approx_S y$ and $z \in \Sigma^*$. Then $xz \in B^*\mathrm{pref}(B)$ iff $yz \in B^*\mathrm{pref}(B)$. Moreover, if $xz, yz \in B^*\mathrm{pref}(B)$, then $\mathrm{idx}(xz) = \mathrm{idx}(yz)$ and $\mathrm{rmd}(xz) \approx_S \mathrm{rmd}(yz)$.*

Lemma 39 *The transition relation $\rightarrow$ is compatible with $\approx_S$. Let $x, y \in \mathrm{Conf}_{\mathrm{bt}}(A, B)$ and $x \approx_S y$. Then*

(i) $(x, i, \alpha) \rightarrow (\bot, \beta)$ iff $(y, i, \alpha) \rightarrow (\bot, \beta)$ for any $i \in \mathbb{N}$.

(ii) Suppose $(x, i, \alpha) \rightarrow (z, j, \beta)$ for $z \in \mathrm{Conf}_{\mathrm{bt}}(A, B)$, $i, j \in \mathbb{N}$. Then there exists $z' \in \mathrm{Conf}_{\mathrm{bt}}(A, B)$ such that $(y, i, \alpha) \rightarrow (z', j, \beta)$ and $z \approx_S z'$.

(iii) Suppose $(x, i, \beta) \rightarrow (z, j, \alpha)$ for $z \in \mathrm{Conf}_{\mathrm{bt}}(A, B)$, $i, j \in \mathbb{N}$. Then there exists $z' \in \mathrm{Conf}_{\mathrm{bt}}(A, B)$ such that $(y, i, \beta) \rightarrow (z', j, \alpha)$ and $z \approx_S z'$.

PROOF.

(i). Since x and y have the isomorphic strategy trees, it follows from Proposition 38 that for any $u \in A$, $xu \in B^*\mathrm{pref}(B)$ iff $yu \in B^*\mathrm{pref}(B)$. Therefore, according to the definition of G, $(x, i, \alpha) \rightarrow (\bot, \beta)$ iff $(y, i, \alpha) \rightarrow (\bot, \beta)$ for any $i \in \mathbb{N}$.

(ii). Suppose $(x, i, \alpha) \rightarrow (z, j, \beta)$ such that there exists $u \in A$ satisfying that $xu \in B^*\mathrm{pref}(B)$, $j = i + \mathrm{idx}(xu)$, and $z = \mathrm{rmd}(xu)$. Since $x \approx_S y$, from Proposition 38, it follows that $\mathrm{idx}(xu) = \mathrm{idx}(yu)$ and $\mathrm{rmd}(xu) \approx_S \mathrm{rmd}(yu)$. Let $z' = \mathrm{rmd}(yu)$. Then $(y, i, \alpha) \rightarrow (z', j, \beta)$ in G and $z \approx_S z'$.

(iii) Suppose $(x, i, \beta) \rightarrow (z, j, \alpha)$.

If $i > 0$, then $z = x$ and $j = i - 1$. Let $z' = y$. Then we have $(y, i, \beta) \rightarrow (z', j, \beta)$ and $z \approx_S z'$.

Otherwise, $j = 0$ and there is $v \in B$ such that $x = vz$. Because $x \approx_S y$, we know that there are $v' \in B$ and $z' \in \mathrm{Conf}_{\mathrm{bt}}(A, B)$ such that $y = v'z'$ and $z \approx_S z'$. Then we have $(y, 0, \alpha) \rightarrow (z', 0, \beta)$ and $z \approx_S z'$. $\square$

Corollary 40 *Let $x \approx_S y$, $i \in \mathbb{N}$, and $p \in \{\alpha, \beta\}$ such that $(x, i, p), (y, i, p) \in V \cup W$. Then $(x, i, p) \in \mathrm{Win}_\beta(G)$ iff $(y, i, p) \in \mathrm{Win}_\beta(G)$.*

Therefore G can be reduced to the quotient of G with respect to $\approx_S$, denoted $G/{\approx_S} = (V/{\approx_S}, W/{\approx_S}, \rightarrow /{\approx_S})$, as follows,

- $V/{\approx_S} = \{([x]_S, i, \alpha) \mid (x, i, \alpha) \in V\}$,
- $W/{\approx_S} = \{(\bot, \beta)\} \cup \{([x]_S, i, \beta) \mid (x, i, \beta) \in W\}$,
- $([x]_S, i, \alpha) \rightarrow /{\approx_S} (\bot, \beta)$ iff $(x, i, \alpha) \rightarrow (\bot, \beta)$.
- $([x]_S, i, p) \rightarrow /{\approx_S} ([y]_S, j, q)$ iff there exist $y' \in [y]_S$ such that $(x, i, p) \rightarrow (y', j, q)$, where $p, q \in \{\alpha, \beta\}$.

Remark 41 *Although the game $G/{\approx_S}$ has finite state space, its transition relation $\rightarrow /{\approx_S}$ is still infinitely-branching. This infinity is due to the fact that A may be infinite, and in game $\mathcal{G}(A, B)$, Attacker may append an arbitrarily long word from A to the end of the current configuration. For pushdown automata, allowing to push into the stack the words from an infinite regular language does not increase the expressive power, since this kind of pushing can still be simulated by pushdown automata. This is not the case in general for one-counter automata. In the following, we will show that some transitions of $G/{\approx_S}$ can be trimmed to make the transition relation of $G/{\approx_S}$ finitely-branching.*

We finally trim the transition relation of $G/{\approx_S}$ into a finitely-branching one and reduce $G/{\approx_S}$ to a one-counter reachability game $(G/{\approx_S})_\bot = (V/{\approx_S}, W/{\approx_S}, \rightsquigarrow)$ as follows.

- $([x]_S, i, \alpha) \rightsquigarrow (\bot, \beta)$ iff $([x]_S, i, \alpha) \rightarrow /{\approx_S} (\bot, \beta)$.
- $([x]_S, i, \alpha) \rightsquigarrow ([y]_S, j, \beta)$ iff $([x]_S, i, \alpha) \rightarrow /{\approx_S} ([y]_S, j, \beta)$ and j is the minimal j' such that $([x]_S, i, \alpha) \rightarrow /{\approx_S} ([y]_S, j', \beta)$.
- $([x]_S, i, \beta) \rightsquigarrow ([y]_S, j, \alpha)$ iff $([x]_S, i, \beta) \rightarrow /{\approx_S} ([y]_S, j, \alpha)$.

Lemma 42 *Suppose $([x]_S, i, \alpha) \in V/{\approx_S}$. Then $([x]_S, i, \alpha) \in \mathrm{Win}_\beta(G/{\approx_S})$ iff $([x]_S, i, \alpha) \in \mathrm{Win}_\beta((G/{\approx_S})_\bot)$.*

From Lemma 42, Corollary 40, and Lemma 34, we get the main result of this section.

Theorem 43 $\mathcal{C}(A, B) = \bigcup_{([x]_S, i, \alpha) \in \mathrm{Win}_\beta((G/\approx_S)_\bot)} B^i [x]_S.$

To prove Lemma 42, we introduce a concept of strong strategies of *Defender* in G and $G/{\approx_S}$.

Definition 44 (Strong strategies of *Defender* in G and $G/{\approx_S}$) *Strong strategies of Defender*[4] *in G starting from (x, i, α) are the same as strategies of Defender in G starting from (x, i, α), that is, they are functions f from $(VW)^+$ to V, with the difference that $(x_{2k}, j_{2k}, \alpha) = f((x_0, j_0, \alpha)(x_1, j_1, \beta) \dots (x_{2k-1}, j_{2k-1}, \beta))$ may not be a successor of $(x_{2k-1}, j_{2k-1}, \beta)$ in G. Instead, (x_{2k}, j_{2k}, α) satisfies the following condition:*

Either $x_{2k} = x_{2k-1}$ and $0 \leq j_{2k} < j_{2k-1}$, or $j_{2k} = 0$ and $(x_{2k-1}, 0, \beta) \rightarrow (x_{2k}, 0, \alpha)$.

Strong strategies of Defender in $G/{\approx_S}$ starting from $([x]_S, i, \alpha)$ are defined similarly. A strong strategy f of Defender in G or $G/{\approx_S}$ is winning if every play consistent with f is winning for Defender.

Intuitively, if the counter value of the current configuration $([x]_S, i, \beta)$ for *Defender* is greater than zero, then by applying a strong strategy, *Defender* may decrease the counter value arbitrarily in G and does not change the state, or decrease the counter value to zero and choose a successor of $([x]_S, 0, \beta)$.

From Lemma 34, it is easy to observe that every strong strategy of *Defender* in G induces a strong strategy of *Defender* in $\mathcal{G}(A, B)$. Therefore, from Proposition 3 and Lemma 34, we have the following result.

Lemma 45 *If Defender has a winning strong strategy in G starting from (x, i, α), then Defender has a winning strategy in G starting from (x, i, α).*

It is also easy to observe that every winning strong strategy of *Defender* in G induces a winning strong strategy in $G/{\approx_S}$, and vice versa. From Lemma 45, we deduce the following result.

Lemma 46 *If Defender has a winning strong strategy in $G/{\approx_S}$ starting from $([x]_S, i, \alpha)$, then Defender has a winning strategy in $G/{\approx_S}$ starting from $([x]_S, i, \alpha)$.*

Now we are ready to prove Lemma 42.

---

[4] We do not define strong strategies of *Attacker* in G and $G/{\approx_S}$ here, since they are not needed for the proof of Lemma 42.

PROOF. (Lemma 42)

It is sufficient to prove $([x]_S, i, \alpha) \in \mathrm{Win}_\beta(G/{\approx_S})$ iff $([x]_S, i, \alpha) \in \mathrm{Win}_\beta((G/{\approx_S})_\perp)$.

"*Only if*" direction:

From the fact that $(G/{\approx_S})_\perp$ is obtained from $G/{\approx_S}$ by restricting the choices of *Attacker*, we know that a winning strategy of *Defender* in $G/{\approx_S}$ starting from $([x]_S, i, \alpha)$ induces a winning strategy of *Defender* in $((G/{\approx_S})_\perp$ starting from $([x]_S, i, \alpha)$.

"*If*" direction:

Suppose f is a winning strategy of *Defender* in $(G/{\approx_S})_\perp$ starting from $([x]_S, i, \alpha)$. From Lemma 46, it is sufficient to show that there is a winning strong strategy f' of *Defender* in $G/{\approx_S}$ starting from $([x]_S, i, \alpha)$.

The description of f' is as follows.

In a play of $G/{\approx_S}$, f' follows f if all the choices of *Attacker* so far belong to $(G/{\approx_S})_\perp$ until some round such that the choice of *Attacker* in that round does not belong to $(G/{\approx_S})_\perp$.

Suppose k rounds have been played in $G/{\approx_S}$ starting from $([x]_S, i, \alpha)$ and all the choices of *Attacker* so far belong to $(G/{\approx_S})_\perp$. Let $([x_0]_S, j_0, \alpha) \dots ([x_{2k-1}]_S, j_{2k-1}, \beta)([x_{2k}]_S, j_{2k}, \alpha)$ (where $[x_0]_S = [x]_S$ and $j_0 = i$) be the history of the k rounds. Suppose in the $(k+1)$-st round of $G/{\approx_S}$ starting from $([x]_S, i, \alpha)$, *Attacker* chooses a successor of $([x_{2k}]_S, j_{2k}, \alpha)$, say $([x_{2k+1}]_S, j_{2k+1}, \beta)$, such that $([x_{2k}]_S, j_{2k}, \alpha) \rightsquigarrow ([x_{2k+1}]_S, j_{2k+1}, \beta)$ does not hold. Then there is $j' : j' < j_{2k+1}$ such that $([x_{2k}]_S, j_{2k}, \alpha) \rightsquigarrow ([x_{2k+1}]_S, j', \beta)$.

Suppose $f(([x_0]_S, j_0, \alpha) \dots ([x_{2k-1}]_S, j_{2k-1}, \beta)([x_{2k}]_S, j_{2k}, \alpha)([x_{2k+1}]_S, j', \beta)) = ([x_{2k+2}]_S, j'', \alpha)$.

Let $f'(([x_0]_S, j_0, \alpha) \dots ([x_{2k-1}]_S, j_{2k-1}, \beta)([x_{2k}]_S, j_{2k}, \alpha)([x_{2k+1}]_S, j_{2k+1}, \beta)) = ([x_{2k+2}]_S, j'', \alpha)$.

Notice that either $j' > 0$, $[x_{2k+2}]_S = [x_{2k+1}]_S$ and $j'' = j' - 1 < j_{2k+1}$, or $j'' = j' = 0$ and $([x_{2k+1}]_S, 0, \beta) \rightarrow_{/\approx_S} ([x_{2k+2}]_S, 0, \alpha)$. So f' defined above satisfies the condition stated in the definition of strong strategies.

The game $G/{\approx_S}$ starting from $([x]_S, i, \alpha)$ continues with the configuration $([x_{2k+2}]_S, j'', \alpha)$ in the beginning of the $(k+2)$-nd round. Starting from the $(k+2)$-nd round, f' still follows f, until some round such that the choice of *Attacker* in that round does not belong to $(G/{\approx_S})_\perp$. If this happens, then we can repeat the above argument to define f', and so on.

Since f is a winning strategy in $(G/{\approx_S})_\perp$, it follows that f' is a winning strong strategy in $G/{\approx_S}$. □

4.2. *Effective construction of the greatest solution*

In the last subsection, we have reduced $\mathcal{G}(A, B)$ to a one-counter reachability game $(G/{\approx_S})_\perp$ with finite state space and finitely-branching transition relation. In the following, we show that the reduction is effective. Because $(G/{\approx_S})_\perp$ is a reachability game played on the transition graph of a one-counter machine, it is sufficient to show that the a finite representation of the one-counter machine can be computed effectively from (A, B). To be more precise, the one-counter machine is defined as follows.

- The state space of the one-counter machine, denoted by Q, is the union of $\{(\perp, \beta)\}$ and the set of $([x]_S, p)$'s, where $x \in \mathrm{Conf}_{\mathrm{bt}}(A, B)$ and $p \in \{\alpha, \beta\}$.
- The transition relation of the one-counter machine, denoted by δ, is a subset of $Q \times Q \times \{= 0, \neq 0\} \times \mathcal{I}$, where $\{= 0, \neq 0\}$ is the set of guards testing whether the current counter value is zero, and $\mathcal{I}$ is a set of finitely many instructions $+c, -c$ for $c \in \mathbb{N}$. The transition relation δ is defined by the following rules.
  - If $([x]_S, 0, \alpha) \rightsquigarrow (\perp, \beta)$, then $(([x]_S, \alpha), (\perp, \beta), = 0, +0) \in \delta$.
  - If $([x]_S, 1, \alpha) \rightsquigarrow (\perp, \beta)$, then $(([x]_S, \alpha), (\perp, \beta), \neq 0, +0) \in \delta$.
  - If $([x]_S, 0, \alpha) \rightsquigarrow ([y]_S, c, \beta)$, then $(([x]_S, \alpha), ([y]_S, \beta), = 0, +c) \in \delta$.
  - If $([x]_S, 1, \alpha) \rightsquigarrow ([y]_S, c, \beta)$, then $(([x]_S, \alpha), ([y]_S, \beta), \neq 0, +(c-1)) \in \delta$.
  - If $([x]_S, 0, \beta) \rightsquigarrow ([y]_S, 0, \alpha)$, then $(([x]_S, \beta), ([y_S], \alpha), = 0, +0) \in \delta$.
  - If $([x]_S, 1, \beta) \rightsquigarrow ([x]_S, 0, \alpha)$, then $(([x]_S, \beta), ([x_S], \alpha), \neq 0, -1) \in \delta$.

In the following, we illustrate that both the state space and the transition relation of the one-counter machine can be computed effectively from (A, B).

Effective computation of the state space Q.

The equivalence classes of $\sim_B$ correspond to the states of the minimal automaton recognizing B. The non-isomorphic trees of depth at most d, labeled by the equivalence classes of $\sim_B$, can be effectively enumerated.

By an induction on $i \leq d$, we can show that for each such tree T, a finite automaton $\mathcal{A}_T$ can be constructed effectively from B to recognize all words $x \in \mathrm{Conf}_{\mathrm{bt}}(A, B)$ whose strategy tree is isomorphic to T. So each equivalence class of $\approx_S$ can be finitely represented by a finite state automaton $\mathcal{A}_T$.

Because Q is the union of $\{(\perp, \beta)\}$ and the set of $([x]_S, p)$'s with $x \in \mathrm{Conf}_{\mathrm{bt}}(A, B)$ and $p \in \{\alpha, \beta\}$, we conclude that a finite representation of the state space Q can be computed effectively from (A, B).

Effective computation of the transition relation δ.

It is sufficient to show how to compute from A, B the (finite) set of transitions of the form $([x]_S, 0, \alpha) \rightsquigarrow ([y]_S, c, \beta)$, $([x]_S, 1, \alpha) \rightsquigarrow ([y]_S, c, \beta)$, $([x]_S, 0, \alpha) \rightsquigarrow (\perp, \beta)$, $([x]_S, 1, \alpha) \rightsquigarrow (\perp, \beta)$, $([x]_S, 0, \beta) \rightsquigarrow ([y]_S, 0, \alpha)$, and $([x]_S, 1, \beta) \rightsquigarrow ([x]_S, 0, \alpha)$ in $(G/\approx_S)_\perp$. In the following, we only illustrate how to compute the transitions of the form $([x]_S, 0, \alpha) \rightsquigarrow ([y]_S, c, \beta)$. The computation of the other transitions is similar.

For each pair $([x]_S, \alpha), ([y]_S, \beta) \in Q$, we do the following computation.

(i) If $y \notin \mathrm{Conf}_{\mathrm{bd}}(A, B)$, then test whether $[x]_S A \cap [y]_S \neq \emptyset$. If the answer is yes, then set $([x]_S, 0, \alpha) \rightsquigarrow ([y]_S, 0, \beta)$.

(ii) If $y \in \mathrm{Conf}_{\mathrm{bd}}(A, B)$, then test whether $[x]_S A \cap B^*[y]_S \neq \emptyset$. If the answer is yes, then compute the minimal $c \geq 0$ such that $[x]_S A \cap B^c[y]_S \neq \emptyset$. Denote such a minimal c as c_0, and set $([x]_S, 0, \alpha) \rightsquigarrow ([y]_S, c_0, \beta)$.

Therefore, we have shown that a finite representation of the one-counter reachability game $(G/\approx_S)_\perp$ can be computed effectively from (A, B). Finally, from the classical results on pushdown games ([21,3,20]), we conclude that the greatest solution of $XA \subseteq BX$, which corresponds to the winning region of *Defender* in $(G/\approx_S)_\perp$, can be constructed effectively from (A, B).

5. Conclusion

In this paper, we gave an effective construction of the greatest solution for the language inequality $XA \subseteq BX$ for the two cases: (i) A, B are regular and there exist $k \geq 1$ such that $\mathrm{pref}(B)A^k \cap B^{\leq k}\mathrm{pref}(B) = \emptyset$, and (ii) A, B are regular and B is a code with decoding delay d. In both cases, we adopted the view of a two-player game and reduced the problem to the computation of winning region of one of the players. While the solution of the first case relied on a shrinking lemma for winning strategies, that of the second case was based on the observation that the game can be reduced to a two-player one-counter game. If A, B are infinite, then the one-counter game for the second case has infinite state space and infinitely-branching transition relation. We further reduced the game to a one-counter reachability game with finite state space and finitely-branching transition relation. Then it follows from the classical results on pushdown games that the greatest solution can be effectively constructed.

There are several directions for the future work. The first direction is to extend the approach proposed in this paper to the more general cases. The most interesting and promising case seems to be the case that B is a code without finite decoding delay. The second direction is to investigate whether the game-solving approach proposed in this paper can be used to construct effectively the greatest solution for language equations, e.g. $XA = BX$.

Acknowledgement. The authors would like to thank the anonymous referees for their invaluable comments and suggestions to improve the quality of the paper.

References

[1] J. Berstel and D. Perrin. *Theory of Codes*. Academic Press, Inc., 1985.
[2] T. Brázdil, P. Jančar, and A. Kučera. Reachability games on extended vector addition systems with states. In *ICALP*, pages 478–489, 2010.
[3] T. Cachat. Uniform solution of parity games on prefix-recognizable graphs. *Electronic Notes in Theoretical Computer Science*, 68(6), 2002.
[4] J. Cassaigne, J. Karhumäki, and P. Salmela. Conjugacy of finite biprefix codes. *Theoretical Computer Science*, 410(24-25):2345–2351, 2009.
[5] C. Choffrut, J. Karhumäki, and N. Ollinger. The commutation of finite sets: a challenging problem. *Theoretical Computer Science*, 273(1-2):69–79, 2002.
[6] J. Conway. *Regular Algebra and Finite Machines*. Chapman and Hall, 1971.
[7] J. Karhumäki, M. Latteux, and I. Petre. Commutation with codes. *Theoretical Computer Science*, 340(2):322–333, 2005.
[8] J. Karhumäki, M. Latteux, and I. Petre. Commutation with Ternary Sets of Words. *Theory of Computing Systems*, 38(2), 2005.
[9] J. Karhumäki and I. Petre. Conway's Problem for Three-Word Sets. *Theoretical Computer Science*, 289:705–725, 2002.

[10] J. Kruskal. Well-Quasi-Ordering, the Tree Theorem and Vazsonyi's Conjecture. *Transactions of American Mathematical Society*, 95:210–225, 1960.
[11] M. Kunc. On language inequalities $XK \subseteq LX$. In *DLT*, pages 327–337, 2005.
[12] M. Kunc. Regular Solutions of Language Inequalities and Well Quasi-Orders. *Theoretical Computer Science*, 348(2-3):277–293, 2005.
[13] M. Kunc. The power of commuting with finite sets of words. *Theory of Computing Systems*, 40(4):521–551, 2007.
[14] O. Ly. A Constructive Solution of the Language Inequation $XA \subseteq BX$. In *TALE'07, "Theory and Applications of Language Equations" 2007, General Publications series of Turku Centre for Computer Science, 44*, pages 76–84, 2007.
[15] D. A. Martin. Borel determinacy. *Annals of Mathematics*, 102(2):363–371, Sept. 1975.
[16] R. Mazala. Infinite games. In *Automata, Logics, and Infinite Games, LNCS 2500*, pages 23–42, 2001.
[17] A. Okhotin. Decision problems for language equations with Boolean operations. In *ICALP*, pages 239–251. LNCS 2719, 2003.
[18] A. Okhotin. A Characterization of the Arithmetical Hierarchy by Language Equations. In *DCFS*, pages 225–237, 2004.
[19] B. Ratoandromanana. Codes et motifs. *RAIRO - Theoretical Informatics and Applications*, 23(4):425–444, 1989.
[20] O. Serre. Note on winning positions on pushdown games with [omega]-regular conditions. *Information Processing Letters*, 85(6):285–291, 2003.
[21] I. Walukiewicz. Pushdown processes: Games and model-checking. *Information and Computation*, 164(2):234–263, 2001.