

Winning Strategies for Streaming Rewriting Games^{*}

Christian Coester¹, Thomas Schwentick², and Martin Schuster³

¹ University of Oxford

² TU Dortmund

³ University of Edinburgh

Abstract. Context-free games on strings are two-player rewriting games based on a set of production rules and a regular target language. In each round, the first player selects a position of the current string; then the second player replaces the symbol at that position according to one of the production rules. The first player wins as soon as the current string belongs to the target language. In this paper the one-pass setting for context-free games is studied, where the knowledge of the first player is incomplete: She selects positions in a left-to-right fashion and only sees the current symbol and the symbols from previous rounds. The paper studies conditions under which dominant and undominated strategies exist for the first player, and when they can be chosen from restricted types of strategies that can be computed efficiently.

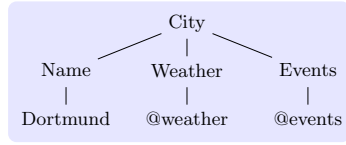
Keywords: context-free games · rewriting games · streaming · incomplete information · strategies.

1 Introduction

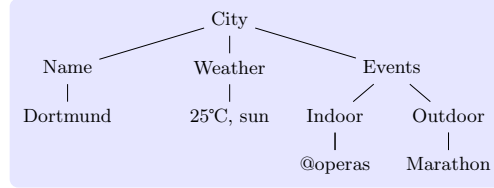
Context-free games on strings are rewriting games based on a set of *production rules* and a regular *target language*. They are played by two players, *Juliet* and *Romeo*, and consist of several rounds. In each round, first Juliet selects a position of the current string; then Romeo replaces the symbol at that position according to one of the production rules. Juliet wins as soon as the current string belongs to the target language. Context-free games were introduced by Muscholl, Schwentick and Segoufin [14] as an abstraction of *Active XML*.

Active XML (AXML) is a framework that extends XML by “active nodes”. In AXML documents, some of the data is given explicitly while other parts are given by means of embedded calls to web services [11]. These embedded calls can be invoked to materialise more data. As an example (adapted from [11,14]), consider a document for the web page of a local newspaper. The document may contain some explicit data, such as the name of the city, whereas information about the weather and local events is given by means of calls to a weather forecast service and an events service (see Figure 1a). By invoking these calls, the data is materialised, i.e. replaced by concrete weather and events data (Figure 1b). The data returned by the service call may contain further service calls.

^{*} The first author is supported by EPSRC Award 1652110.



(a) AXML document before invoking calls



(b) AXML document after invoking calls

Fig. 1. An AXML document before and after the invocation of service calls

It might not be necessary to invoke all possible service calls. In the example of Figure 1, data about the weather might be relevant only if there are outdoor events and otherwise it does not need to be materialised. The choice which data needs to be materialised by the sender and the receiver may be influenced by considerations about performance, capabilities, security and functionalities and can be specified, for instance, by a DTD [11]. An overview about AXML is given in [1].

The question whether a document can be rewritten so that it satisfies the specification then basically translates to the *winning problem* for context-free games: given a game and a string⁴, does Juliet have a winning strategy? In general, this problem is undecidable, however it becomes decidable if Juliet has to follow a *left-to-right-strategy* [14]. With such a strategy, Juliet basically traverses the string from left to right and decides, for each symbol, whether to play *Read* (keep the symbol and go to the next symbol) or *Call* (let Romeo replace the symbol).

With applications in mind where the AXML document comes as a data stream, Abiteboul, Milo and Benjelloun [3] initiated the study of a further strategy restriction, called *one-pass strategies*: Juliet still has to process the string from left-to-right, but now she does not even see the remaining part of the string, beyond the current symbol.

Due to the lack of knowledge of Juliet, one-pass strategies are more difficult to analyse and have less desirable properties than left-to-right strategies. For instance, in the sandbox game with one replacement rule $a \rightarrow b$ and the target language $\{ab, bc\}$, Juliet has a winning strategy that wins on the word ab (Read the initial a) and one that wins on ac (Call the initial a), but none that wins on both [3]. This example shows that even for some extremely simple games and input strings, there is no *dominant* strategy for Juliet, i.e., a strategy that wins on all words on which she has a winning strategy at all. However, both mentioned strategies are optimal in the sense that they can not be strictly improved; we call such strategies *undominated*.⁵

⁴ The restriction to strings instead of trees was justified in [11].

⁵ In [3], dominant and undominated strategies were called optimum and optimal respectively.

Since our focus will be on one-pass strategies, we will often just say strategy for short. We consider several restricted types of strategies. A strategy is *forgetful* if it does not need to remember all decisions it made, but only the (prefix of the) current string. Abiteboul et al. [3] also introduced *regular strategies*, a simple type of one-pass strategies defined by a finite state automaton, and therefore efficiently computable. Some strategies that are both regular and forgetful are of a particularly simple form such that they computed by an automaton that is derived from the minimal automaton for the target language. We call this most restricted type *strongly regular*. We refer to Section 2 for precise definitions of these notions.

We study the following questions.

- Under which circumstances does an undominated one-pass strategy exist?
- When can a dominant or undominated strategy even be chosen from one of the restricted types (regular, forgetful or strongly regular)?

This paper is based on the Master’s thesis of the first author, supervised by the other two authors [8]. The thesis contains further results, investigating also the computational complexity of some related problems.

1.1 Our results

Regarding the existence of undominated strategies, we show that the situation is much better than for dominant strategies. Although it remains unclear whether undominated strategies exist for *all* context-free games, we identify important classes of games in which they do exist. More precisely, we give a semantical restriction, the bounded depth property, that guarantees existence of undominated strategies, and a fairly natural syntactical restriction, prefix-freeness⁶, that guarantees existence of an undominated strategy which can even be chosen regular. For two other families of restricted games, non-recursive games (i.e., the production rules do not allow recursive generation of a symbol from itself) and games with finite target language, we show that if they have a dominant strategy, then even a strongly regular one.

- Theorem 1.** (a) *Every game with the bounded depth property has an undominated strategy.*
 (b) *Every prefix-free game has a regular undominated strategy.*
 (c) *Every game with a finite target language that has a dominant strategy has a strongly regular dominant strategy.*
 (d) *Every non-recursive game with a dominant strategy has a strongly regular dominant strategy.*

Games with a finite target language as well as non-recursive games also have the bounded depth property, so these two classes also have an undominated strategy by part (a) (cf. [8, Theorem 3.6]).

⁶ As explained later, every game can be transformed into a very similar prefix-free game.

We complement the implication results of Theorem 1 (c) and (d) by negative results, showing that these implications do not generalise to arbitrary games or to undominated strategies. More strongly, the following theorem states that even much weaker implications do not hold in general.

Theorem 2. (a) *There exists a game G_1 with a regular dominant strategy but no forgetful dominant strategy.*
 (b) *There exists a game G_2 with a forgetful regular dominant strategy but no strongly regular dominant strategy.*
 (c) *The statements (a) and (b) hold also when “dominant” is replaced by “undominated”. In this case, G_1 and G_2 can even be chosen as non-recursive games with a finite target language.*

1.2 Related work

Further background about AXML is given in [1,2,11]. Context-free games were introduced in [13], which is the conference paper corresponding to [14]. The article studies the decidability and complexity of deciding whether a winning unrestricted or left-to-right strategy exists for a word in the general case and several restricted cases. More recently, Holík et al. gave a new algorithm for determining the winner of a left-to-right context-free game and determining a winning strategy [10]. One-pass strategies and (forgetful) regular strategies were introduced in [3]. The complexity of deciding, for a given context-free game, whether Juliet has a winning left-to-right strategy for every word for which she has a winning unrestricted strategy is studied in [5]. Extended settings of context-free games with nested words (resembling the tree structure of (A)XML documents) are examined in [15,12].

The article [14] also showed a tight connection between context-free games and pushdown games [17,7,16]. In [4,6], variants of these pushdown games in settings of imperfect information are studied.

1.3 Organization

In Section 2 we provide several definitions and some basic lemmas. We prove Theorem 1 (a) in Section 3. In the two subsequent sections, we sketch the proofs of parts (b), (c) and (d). We also explain our motivation for prefix-freeness in Section 4, which we consider the most relevant case and with the most interesting proof. Section 6 contains proofs of Theorem 2 (a) and (b). Most proof details of Theorem 1 as well as the proof of Theorem 2 (c) can be found in the full version of our paper [9].

2 Preliminaries

We denote the set of strings over an alphabet Σ by Σ^* and the set of non-empty strings by Σ^+ . Σ^k denotes the set of strings of length k and $\Sigma^{\leq k}$ the set of strings of length at most k .

A *nondeterministic finite automaton (NFA)* is a tuple $\mathcal{A} = (Q, \Sigma, \delta, s, F)$, where Q is the set of states, Σ the alphabet, $\delta \subseteq Q \times \Sigma \times Q$ the transition relation, $s \in Q$ the initial state and $F \subseteq Q$ the set of accepting states. A *run* on a string $w = w_1 \cdots w_n$ is a sequence q_0, \dots, q_n of states such that $q_0 = s$ and, for each $i \leq n$, $(q_{i-1}, w_i, q_i) \in \delta$. A run is *accepting* if $q_n \in F$. A word w is in the language $L(\mathcal{A})$ of \mathcal{A} if \mathcal{A} has an accepting run on w . If \mathcal{A} is deterministic, i.e., for each p and a , there is exactly one state q such that $(p, a, q) \in \delta$, then we consider δ as *transition function* $Q \times \Sigma \rightarrow Q$ and also use the *extended transition function* $\delta^* : Q \times \Sigma^* \rightarrow Q$, as usual.

Context-free games A *context-free game*, or a *game* for short, is a tuple $G = (\Sigma, R, T)$ consisting of a finite alphabet Σ , a minimal⁷ DFA $T = (Q, \Sigma, \delta, s, F)$, and a binary relation $R \subseteq \Sigma \times \Sigma^+$ such that for each $a \in \Sigma$, the *replacement language* $L_a \stackrel{\text{def}}{=} \{v \in \Sigma^+ \mid (a, v) \in R\}$ of a is regular. We call $L(T)$ the *target language* of G . By $\Sigma_f = \{a \in \Sigma \mid \exists v \in \Sigma^+ : (a, v) \in R\}$ we denote the set of *function symbols*, i.e. the symbols occurring as the left hand side of a rule. The languages L_a are usually represented by regular expressions R_a for each $a \in \Sigma_f$, and we specify R often by expressions of the form $a \rightarrow R_a$. We note that the definition of context-free games assures $\epsilon \notin L_a$.

The semantics of context-free games formalises the intuition given in the introduction. In a configuration, we summarise the information about a current situation of a play together with some information about the history of the play. For the latter, let $\widehat{\Sigma}_f = \{\widehat{a} \mid a \in \Sigma_f\}$ be a disjoint copy of the set Σ_f of function symbols, and let $\overline{\Sigma} = \Sigma \dot{\cup} \widehat{\Sigma}_f$. A *configuration* is a tuple $(\alpha, u) \in \overline{\Sigma}^* \times \Sigma^*$. If u is non-empty, i.e. $u = av$ for $a \in \Sigma$ and $v \in \Sigma^*$, then we also denote this configuration by (α, a, v) , consisting of a *history string* α , a *current symbol* $a \in \Sigma$ and a *remaining string* $v \in \Sigma^*$. We denote the set of all (syntactically) possible configurations by \mathcal{K} . Intuitively, if the i th symbol of the history string is $b \in \Sigma$ then this shall denote that Juliet's i th move was to read the symbol b , and if it is $\widehat{b} \in \widehat{\Sigma}_f$ then this shall denote that Juliet's i th move was to call b . The remaining string is the string of symbols that have not been revealed to Juliet yet. By $\natural : \overline{\Sigma}^* \rightarrow \Sigma^*$ we denote⁸ the homomorphism which deletes all symbols from $\widehat{\Sigma}_f$ and is the identity on Σ . We call $\delta^*(s, \natural\alpha)$ the *T-state* of the configuration (α, u) .

A play is a sequence of configurations, connected by moves. In one move at a configuration (α, a, v) Juliet can either “read” a or “call” a . In the latter case, Romeo can replace a by a string from L_a . More formally, a *play* of a game is a finite or infinite sequence $\Pi = (K_0, K_1, K_2, \dots)$ of configurations with the following properties:

- (a) The *initial configuration* is of the form $K_0 = (\epsilon, w)$, where $w \in \Sigma^*$ is called the *input word*.
- (b) If $K_n = (\alpha, a, v)$, then either $K_{n+1} = (\alpha a, v)$ or $K_{n+1} = (\alpha \widehat{a}, xv)$ with $x \in L_a$. In the former case we say that Juliet plays a *Read move*, otherwise she plays a *Call move* and Romeo replies by x .

⁷ The assumption that T is minimal will be convenient at times.

⁸ We usually omit brackets and write, e.g., $\natural\alpha\beta$ for $\natural(\alpha\beta)$.

- (c) If $K_n = (\alpha, \epsilon)$, then K_n is the last configuration of the sequence. Its history string α is called the *final history string* of Π . Its *final string* is $\sharp\alpha$.

A play is *winning for Juliet* (and *losing for Romeo*) if it is finite and its final string is in the target language $L(T)$. A play is *losing for Juliet* (and *winning for Romeo*) if it is finite and its final string is not in $L(T)$. An infinite play is neither winning nor losing for any player.

Strategies As mentioned in the introduction, we are interested in so-called one-pass strategies for Juliet, where Juliet's decisions do not depend on any symbols of the remaining string beyond the current symbol.

A *one-pass strategy for Juliet* is a map $\sigma: \bar{\Sigma}^* \times \Sigma_f \rightarrow \{\text{Call}, \text{Read}\}$, where the argument corresponds to the first two components of a configuration. A *strategy for Romeo* is a map $\tau: \bar{\Sigma}^* \times \Sigma_f \rightarrow \Sigma^+$ where $\tau(\alpha, a) \in L_a$ for each $(\alpha, a) \in \bar{\Sigma}^* \times \Sigma_f$.⁹ We generally denote strategies for Juliet by $\sigma, \sigma', \sigma_1, \dots$ and Romeo strategies by $\tau, \tau', \tau_1, \dots$. We often just use the term *strategy* to refer to a one-pass strategy for Juliet.

The *play of σ and τ on w* , denoted $\Pi(\sigma, \tau, w)$, is the (unique) play (K_0, K_1, \dots) with input word w satisfying that

- if $K_n = (\alpha, a, v)$ and $\sigma(\alpha, a) = \text{Read}$, then $K_{n+1} = (\alpha a, v)$,
- if $K_n = (\alpha, a, v)$ and $\sigma(\alpha, a) = \text{Call}$, then $K_{n+1} = (\alpha \hat{a}, \tau(\alpha, a)v)$.

The *depth* of a finite play is its maximum nesting depth of *Call* moves. E.g., if Romeo replaces some symbol a by a string u and Juliet calls a symbol in u , the nesting depth of this latter *Call* move is 2.

A strategy σ is *terminating* if each of its plays is finite. The *depth* of σ is the supremum of depths of plays of σ . Note that each strategy with finite depth is terminating. The converse, however, is not true and it is easy to construct counter-examples of a game and a strategy σ where each play of σ has finite depth but depths are arbitrarily large.

A strategy σ *wins* on a string $w \in \Sigma^*$ if every play of σ on w is winning (for Juliet). By $W(\sigma) = W_G(\sigma)$ we denote the set of words on which σ wins in G . In contrast, σ *loses* on w if there exists a losing play of σ on w . Note that σ neither wins nor loses on w if there exists an infinite play of σ on w but no losing play of σ on w .

A strategy σ *dominates* a strategy σ' if $W(\sigma') \subseteq W(\sigma)$. A strategy σ is *dominant* if it dominates every other (one-pass) strategy. It is *undominated* if there is no strategy σ' with $W(\sigma) \subsetneq W(\sigma')$.

In the proofs of Theorem 1 (a) and (b), we will actually show a slightly stronger form of optimality than “undominated”, which we call weakly dominant. To define it, fix some total order $<$ of the alphabet Σ . We order strings by *shortlex order*, i.e. for two strings $v, w \in \Sigma^*$ we define $v <_{\text{sl}} w$ if $|v| < |w|$ or

⁹ Even though we think of Romeo as an omniscient adversary, it is not necessary to provide the remaining string as an argument to τ : The remaining string is uniquely determined by the input word and his own and Juliet's previous moves.

if $|v| = |w|$ and v precedes w in the lexicographical order. We extend this to a total order \leq_{sl} on sets of words as follows. Let $V, W \subseteq \Sigma^*$ be two sets with $V \neq W$. Their order is determined by the minimal string w (with respect to shortlex order \leq_{sl}) that is contained in only one of the two sets. If $w \in W$, then $V <_{\text{sl}} W$; otherwise $W <_{\text{sl}} V$. We observe that if $V \subsetneq W$ then $V <_{\text{sl}} W$. A strategy σ is *weakly dominant* if, for every strategy σ' it holds $W(\sigma') \leq_{\text{sl}} W(\sigma)$. Thus, a weakly dominant strategy can be seen as a best undominated strategy with respect to \leq_{sl} .

The following lemma is convenient as it will often allow us to assume, without loss of generality, that a given strategy is terminating.

Lemma 1. *Each strategy is dominated by a terminating strategy.*

Proof. Given a strategy σ , we construct a terminating strategy σ' with $W(\sigma) \subseteq W(\sigma')$ as follows.

Consider some $\alpha \in \overline{\Sigma}^*$ and $a \in \Sigma_f$. If there exists a play Π of σ that contains a configuration (α, av) for some $v \in \Sigma^*$ such that no later configuration of the form $(\alpha\alpha', v)$ occurs in Π , then let $\sigma'(\alpha, a) = \text{Read}$ and $\sigma'(\alpha a \beta, b) = \text{Read}$ for each $\beta \in \overline{\Sigma}^*$ and $b \in \Sigma_f$. For all elements of the domain for which σ' is not already defined by this, we define σ' like σ . Clearly, σ' is terminating and $W(\sigma) \subseteq W(\sigma')$. \square

Restricted strategy types A strategy σ is *regular* if the set L of strings αa with $\sigma(\alpha, a) = \text{Call}$ is regular. In this case, a DFA \mathcal{A} for L is called a *strategy automaton* for $\sigma = \sigma_{\mathcal{A}}$. A strategy is *forgetful* if its decisions are independent of symbols from $\widehat{\Sigma}_f$ in the history string, i.e. if $\sigma(\alpha, a) = \sigma(\beta, a)$ whenever $\sharp\alpha = \sharp\beta$. Clearly, if σ is regular and forgetful, then $L' \stackrel{\text{def}}{=} \{\sharp\alpha a \mid \sigma(\alpha, a) = \text{Call}\}$ is regular. A DFA \mathcal{A} for L' is also called a strategy automaton, and we write $\sigma_{\mathcal{A}} = \sigma$ again.

We are particularly interested in the special case of regular forgetful strategies where Juliet's decisions depend only on the current T -state and the current symbol. More precisely, if $T = (Q, \Sigma, \delta, s, F)$ is the target automaton and the strategy automaton is of the form $\mathcal{A} = (Q \cup \{\text{Call}\}, \Sigma, \delta_{\mathcal{A}}, s, \{\text{Call}\})$ with $\delta_{\mathcal{A}}(q, a) \in \{\delta(q, a), \text{Call}\}$, for each q and a , then $\sigma_{\mathcal{A}}$ is called *strongly regular*.

Classes of games A game G has the *bounded depth property* if there exists a sequence $(B_k)_{k \in \mathbb{N}_0} \subseteq \mathbb{N}$ such that for each one-pass strategy σ for G and each $k \in \mathbb{N}$ there exists a one-pass strategy σ_k that wins on each $w \in W(\sigma) \cap \Sigma^{\leq k}$ with plays of depth at most $B_{|w|}$. Roughly speaking, the bounded depth property means the following: When input words are restricted to a finite set, Juliet can choose a strategy whose depth is bounded on words from her winning set, without losing strategic power.

A game is *prefix-free* if each replacement language L_a is prefix-free, that is, there are no $u, v \in L_a$ where u is a proper prefix of v .

A game is *non-recursive* if no symbol can be derived from itself by a sequence of rules, i.e. there do not exist $a_0, \dots, a_n \in \Sigma_f$, $n \geq 1$, such that $a_0 = a_n$ and for each $k = 1, \dots, n$ there exists a word in $L_{a_{k-1}}$ containing a_k .

Convergence of strategies A concept used in the proofs of parts (a) and (d) is the convergence of a sequence of one-pass strategies. A sequence $(\sigma_k)_{k \in \mathbb{N}}$ of strategies *converges to* a strategy σ if for each $n \in \mathbb{N}$ there exists $k_0 \in \mathbb{N}$ such that for each $k \geq k_0$ and $(\alpha, a) \in \overline{\Sigma}^{\leq n} \times \Sigma$ it holds that $\sigma(\alpha, a) = \sigma_k(\alpha, a)$.

Lemma 2. *Let G be a game and $(\sigma_k)_{k \in \mathbb{N}}$ be a sequence of one-pass strategies that converges to some one-pass strategy σ . Let $L_1 \subseteq L_2 \subseteq \dots$ be an infinite sequence of languages such that, for every k , $L_k \subseteq W(\sigma_k)$ and let $L \stackrel{\text{def}}{=} \bigcup_{k \in \mathbb{N}} L_k$.*

Then σ does not lose on any word $w \in L$.

It should be noted, however, that σ might fail to win on some of these words due to infinite plays.

Proof of Lemma 2. Towards a contradiction, suppose that σ loses on a word $w \in L$. Then there exists a strategy of Romeo with which he wins the (finite) play $\Pi = \Pi(\sigma, \tau, w) = (K_0, \dots, K_n)$. Let $k_0 \in \mathbb{N}$ be such that for each $k \geq k_0$ and $(\alpha, a) \in \overline{\Sigma}^{\leq n} \times \Sigma$ it holds that $\sigma(\alpha, a) = \sigma_k(\alpha, a)$. Let furthermore k_1 be such that $w \in L_{k_1}$ and let $k \stackrel{\text{def}}{=} \max(k_0, k_1)$. Then Π is also a play of σ_k on w . But then σ_k loses on $w \in L_{k_1} \subseteq L_k$, the desired contradiction. \square

3 Games with the bounded depth property

In this section, we show that each game with the bounded depth property admits a weakly dominant strategy, implying Theorem 1 (a). Since non-recursive games trivially have the bounded depth property, and also games with a finite target language (cf. [8, Lemma 3.5]) and prefix-free games (cf. Section 4) have the bounded depth property, it follows that any such game has an undominated strategy. In fact, all of these games satisfy the stronger version of the bounded depth property where $B_k = B$ does not depend on k .

For a strategy σ and some $i \geq 0$, we denote by $\sigma|_i$ the restriction of σ to the first i rounds of the game. Thus $\sigma|_i$ is a mapping $\sigma|_i: \overline{\Sigma}^{\leq i-1} \times \Sigma \rightarrow \{\text{Call}, \text{Read}\}$ and $\sigma|_0$ is the mapping with empty domain.

Proof of Theorem 1 (a). Let G be a context-free game with the bounded depth property and let $(B_k)_{k \in \mathbb{N}_0} \subseteq \mathbb{N}$ be its sequence of depth bounds.

We first define a language L which will serve as the winning set of the weakly dominant strategy that will be constructed below.

The definition of L is by induction. For each $k \geq 0$, we define a set $L_k \subseteq \Sigma^{\leq k}$ such that $L_k \subseteq L_{k+1}$, and finally let $L \stackrel{\text{def}}{=} \bigcup_k L_k$.

Let $L_0 = \{\epsilon\}$ if ϵ is in the target language of G , and $L_0 = \emptyset$ otherwise.

For $k \geq 0$, we define L_{k+1} as the maximal set with respect to \leq_{sl} of the form $W(\sigma) \cap \Sigma^{\leq k+1}$ for some strategy σ with $L_k \subseteq W(\sigma)$. It is easy to see that the following two properties hold by construction.

- (1) For each k , there is a strategy σ such that $L_k \subseteq W(\sigma)$.

(2) There is no strategy σ with $L <_{\text{sl}} W(\sigma)$.

Thanks to property (2), it suffices to construct a strategy $\hat{\sigma}$ with $L \subseteq W(\hat{\sigma})$.

For each $k \geq 0$, let S_k be the set of strategies σ with $L_k \subseteq W(\sigma)$ for which each play on a word $w \in L_k$ has depth at most $B_{|w|}$. Because of property (1) and since G has the bounded depth property, we have $S_k \neq \emptyset$ for every $k \geq 0$.

We will construct mappings $\rho_k: \overline{\Sigma}^{\leq k-1} \times \Sigma \rightarrow \{\text{Call}, \text{Read}\}$ such that for every $k \geq 0$,

- ρ_{k+1} extends ρ_k ; more precisely: $\rho_{k+1}|_k = \rho_k$, and
- for each $\ell \geq k$ there exists $\sigma_\ell^k \in S_\ell$ with $\rho_k = \sigma_\ell^k|_k$.

Let ρ_0 be the mapping with empty domain. Fix k such that ρ_0, \dots, ρ_k are defined and have the stated properties. Since there are only finitely many mappings $\overline{\Sigma}^{\leq k} \times \Sigma \rightarrow \{\text{Call}, \text{Read}\}$, one of them has to occur infinitely often within $(\sigma_\ell^k|_{k+1})_{\ell \geq k}$. Let ρ_{k+1} be such a mapping. For $\ell' \geq k+1$ we can choose $\sigma_{\ell'}^{k+1} = \sigma_\ell^k$ for some $\ell \geq \ell'$ with $\rho_{k+1} = \sigma_\ell^k|_{k+1}$. This defines a sequence ρ_0, ρ_1, \dots with the properties above. Let $\hat{\sigma}$ be the strategy that is uniquely determined by $\hat{\sigma}|_k = \rho_k$, for every k . Clearly $(\rho_k)_{k \in \mathbb{N}}$ converges¹⁰ to $\hat{\sigma}$.

Thanks to Lemma 2 it suffices to show that $\hat{\sigma}$ terminates on L . Let thus $w \in L$ and τ be a Romeo strategy. We show that the depth of $\Pi \stackrel{\text{def}}{=} \Pi(\hat{\sigma}, \tau, w)$ is at most $B_{|w|}$. Otherwise let k be such that in the k th round Juliet does a *Call* move of nesting depth $B_{|w|} + 1$. However, $\hat{\sigma}|_k = \rho_k = \sigma_\ell^k|_k$, where $\ell = \max\{k, |w|\}$, and $\sigma_\ell^k \in S_\ell$ has depth at most $B_{|w|}$ on $w \in L_\ell$, a contradiction. Therefore the depth of Π is at most $B_{|w|}$ and by König's Lemma Π is thus finite, completing the proof. \square

4 Prefix-free games

Prefix-freeness appears as a realistic constraint for a practical (Active XML) setting since it can be easily enforced by suffixing each replacement string with a special end-of-file symbol: *Every* game $G = (\Sigma, R, T)$ can be transformed into a prefix-free game $G' = (\Sigma', R', T')$ by letting $\Sigma' = \Sigma \dot{\cup} \{\$$ for some new symbol $\$ \notin \Sigma$ that shall denote the end of replacement strings, and further letting $R'_a = R_a \$$ for each $a \in \Sigma_f$ to enforce that replacement words end with $\$$, and adding a loop transition for the symbol $\$$ to each state of T (accomplishing that the symbol $\$$ is “ignored” by the target language). Another special case of prefix-free games, which is similar to the *one-pass with size* setting discussed in [3], are games where the alphabet Σ contains (besides other symbols) numbers $1, \dots, N$ for some $N \in \mathbb{N}$ and all replacement strings are of the form nx where $x \in \Sigma^+$ and $n = |x|$. Our result for prefix-free games also easily transfers to the setting where the input word is revealed to Juliet in a one-pass fashion, but Romeo's replacement words are revealed immediately.

¹⁰ Since the ρ_k are only partially defined, one might consider the strategies σ_k that result from the ρ_k which take the value *Call* whenever ρ_k is undefined.

We sketch the proof of Theorem 1 (b) in the following.

A context-free game on a string $w = a_1 \cdots a_n$ can be viewed as a sequence of n games on the single symbols a_1, \dots, a_n . Intuitively, in prefix-free games Juliet has the benefit to know when a subgame on some symbol a_i has ended and when the next subgame starts.

This allows us to view strategies of Juliet in a hierarchical way: they consist of a top-level strategy that chooses, whenever a subgame on some a_i starts, a strategy for this subgame. This choice may take the current history string into account. We will use this view to proceed in an inductive fashion: we establish that there are automata for the subgame strategies and then combine these automata with suitable automata for a “top-level” strategy.

It turns out that the choice of the top-level strategy boils down to an “online word problem” for NFAs which we introduce and study first.

The online word problem for NFAs In the online-version of the word problem for an NFA \mathcal{N} , denoted $\text{ONLINENFA}(\mathcal{N})$, the single player gets to know the symbols of a word one by one, and always needs to decide which transition \mathcal{N} should take before the next symbol is revealed. We only consider the case that $\mathcal{N} = (Q, \Sigma, \delta, s, F)$ has at least one transition for each symbol from each state. Formally, a strategy is a map $\rho: \Sigma^* \rightarrow Q$ such that $\rho(\epsilon) = s$ and $(\rho(w), a, \rho(wa)) \in \delta$ for each $w \in \Sigma^*$ and $a \in \Sigma$.

Given a strategy ρ for $\text{ONLINENFA}(\mathcal{N})$, we denote by $W_{\mathcal{N}}(\rho)$ the *winning set* of words that are accepted by \mathcal{N} if the player follows ρ . A strategy ρ is *weakly dominant* if, for every strategy ρ' , it holds $W_{\mathcal{N}}(\rho') \leq_{\text{sl}} W_{\mathcal{N}}(\rho)$.

We are interested in strategies that can be computed by automata. A particularly simple such strategy for $\text{ONLINENFA}(\mathcal{N})$ can be obtained by transforming \mathcal{N} into a DFA \mathcal{D} by removing transitions. The associated strategy $\rho_{\mathcal{D}}$ is the one that only uses the transitions of \mathcal{D} . We prove that \mathcal{D} can be chosen such that $\rho_{\mathcal{D}}$ is weakly dominant.

Lemma 3. *For each NFA \mathcal{N} , there exists a DFA \mathcal{D} obtained by removing transitions from \mathcal{N} such that $\rho_{\mathcal{D}}$ is a weakly dominant strategy for $\text{ONLINENFA}(\mathcal{N})$.*

Game composition and game effects Let in the following, $G = (\Sigma, R, T)$ be a prefix-free game with $T = (Q, \Sigma, \delta, s, F)$. Let furthermore, for every $a \in \Sigma$, $\mathcal{A}_a = (Q_a, \Sigma, \delta_a, s_a, \{f_a\})$ be a minimal DFA for the replacement language L_a of a . Since L_a is prefix-free, \mathcal{A}_a has a unique accepting state f_a .

For a strategy σ (for Juliet or Romeo) and a string $\alpha \in \Sigma^*$, we define the *substrategy* σ^α of σ by $\sigma^\alpha(\beta, a) = \sigma(\alpha\beta, a)$.

In the following, $\text{states}(q, w, \sigma)$ denotes the set of T -states that can be reached at the end of a play of σ on w if the initial state of T were q . More precisely, it is the set of states of the form $\delta^*(q, \alpha)$ where α is a final history string of a play of σ on w .

An *effect triple* (p, a, S) consists of a state $p \in Q$, a symbol $a \in \Sigma$ and a set $S \subseteq Q$. We say that (p, a, S) is an effect triple of σ if $\text{states}(p, a, \sigma) \subseteq S$. That

is, starting from p and processing a according to σ , one is guaranteed to reach a state in S . We call (p, a, S) *trivial* if $\delta(p, a) \in S$, i.e. if it is an effect triple of a strategy that plays Read on a . The *single-symbol effect* $\text{sse}(\sigma)$ of a strategy σ is the set of all its effect triples. Finally, we define the *effect set* $E(\sigma)$ of a strategy σ as $E(\sigma) \stackrel{\text{def}}{=} \bigcup_{\alpha \in \Sigma^*} \text{sse}(\sigma^\alpha)$. That is, $E(\sigma)$ contains all effect triples that are induced by substrategies of σ .

For a set E of effect triples, consider the NFA $\mathcal{N}_E = (\mathcal{P}(Q), \Sigma, \delta_E, \{s\}, \mathcal{P}(F))$, where δ_E is defined as follows. For sets $S, S' \subseteq Q$ and $a \in \Sigma$, $(S, a, S') \in \delta_E$ if, for each $p \in S$, there is some $S'' \subseteq S'$ such that $(p, a, S'') \in E$.

Proposition 1. *Let $G = (\Sigma, R, T)$ be a prefix-free game, E a set of effect triples, and σ a terminating strategy such that $E(\sigma) \subseteq E$. Then there is a strategy ρ for $\text{ONLINENFA}(\mathcal{N}_E)$ such that $W_G(\sigma) = W_{\mathcal{N}_E}(\rho)$.*

Proof. It is straightforward to verify that $\rho(w) \stackrel{\text{def}}{=} \text{states}(s, w, \sigma)$ yields a well-defined strategy ρ for $\text{ONLINENFA}(\mathcal{N}_E)$. The proposition follows, since $w \in W_{\mathcal{N}_E}(\rho)$ if and only if $\rho(w) \subseteq F$, and $w \in W_G(\sigma)$ if and only if $\text{states}(s, w, \sigma) \subseteq F$.

We say that a strategy automaton $\mathcal{A} = (Q_{\mathcal{A}}, \Sigma, \delta_{\mathcal{A}}, s_{\mathcal{A}}, F_{\mathcal{A}})$ is (p, a, S) -*inducing* if $\sigma_{\mathcal{A}}$ is terminating, $a \in \Sigma_f$, and the following conditions hold.

- For each $u \in L_a$, $\text{states}(p, u, \sigma_{\mathcal{A}}) \subseteq S$.
- There are disjoint subsets $Q_{\mathcal{A},q} \subseteq Q_{\mathcal{A}}$, for $q \in S$, such that for every play of σ on some $u \in L_a$ with final history string α , it holds $\delta_{\mathcal{A}}^*(s_{\mathcal{A}}, \alpha) \in Q_{\mathcal{A},q} \Leftrightarrow \delta^*(p, \alpha) = q$.
Furthermore, there is no proper prefix β of α for which $\delta_{\mathcal{A}}^*(s_{\mathcal{A}}, \beta) \in Q_{\mathcal{A},r}$, for any r .

Proposition 2. *Let $G = (\Sigma, R, T)$ be a prefix-free game and E a set of effect triples such that for each non-trivial $t \in E$, there exists a t -inducing strategy automaton \mathcal{A}_t . Then there is a strategy automaton \mathcal{A} for G such that, for each strategy ρ for $\text{ONLINENFA}(\mathcal{N}_E)$, $W_{\mathcal{N}_E}(\rho) \leq_{sl} W_G(\sigma_{\mathcal{A}})$.*

A crucial ingredient is the following proposition, which will allow us to restrict our attention to strategies of finite depth. An almost identical proof can also be used to show that prefix-free games have the bounded depth property, and we could use this and Theorem 1 (a) to deduce immediately that they have weakly dominant strategies. However, we are aiming for the stronger result that they have *regular* weakly dominant strategies.

Proposition 3. *In prefix-free games, each effect triple of a terminating strategy is also an effect triple of a strategy of bounded depth.*

Proof idea. Let E denote the set of effect triples of bounded depth strategies. Consider an effect triple $(p, a, S) \notin E$ of a strategy $\hat{\sigma}$. If the effect triples of $\hat{\sigma}$'s substrategies on the symbols of a 's replacement word were all in E , then these substrategies could be replaced so as to obtain a bounded depth strategy σ for

(p, a, S) , contradicting $(p, a, S) \notin E$. Thus, for each $(p, a, S) \notin E$, Romeo can force a configuration in the play against $\hat{\sigma}$ on a where the effect triple of the substrategy is again not in E . But Romeo can do this repeatedly, so $\hat{\sigma}$ is not terminating. \square

The finite depth allows us to construct t -inducing strategy automata by induction on the depth of a strategy with effect triple t .

Proposition 4. *Let $G = (\Sigma, R, T)$ be a prefix-free game and (p, a, S) a non-trivial effect triple of some terminating strategy σ . Then there exists a (p, a, S) -inducing strategy automaton.*

Now we are ready to prove Theorem 1 (b).

Proof of Theorem 1 (b). Let $G = (\Sigma, R, T)$ be prefix-free. Let E be the set of effect triples of terminating strategies. By Proposition 4 there is a t -inducing automaton for each non-trivial $t \in E$. Let $\sigma_{\mathcal{A}}$ be the regular strategy as guaranteed by Proposition 2. We show that $\sigma_{\mathcal{A}}$ is weakly dominant.

To this end, let σ be any terminating strategy for G (cf. Lemma 1). Since $E(\sigma) \subseteq E$, Proposition 1 guarantees a strategy ρ for $\text{ONLINENFA}(\mathcal{N}_E)$ such that $W_G(\sigma) = W_{\mathcal{N}_E}(\rho)$. By Proposition 2, $W_{\mathcal{N}_E}(\rho) \leq_{\text{sl}} W_G(\sigma_{\mathcal{A}})$, and therefore, altogether $W_G(\sigma) \leq_{\text{sl}} W_G(\sigma_{\mathcal{A}})$ as required. \square

5 Strongly regular dominant strategies

In this section, we sketch the main ideas of the proof of Theorem 1 (c) and (d).

We say that a strategy σ has a (q, a) -conflict, for $q \in Q$ and $a \in \Sigma_f$, if there are configurations (α_1, a, u_1) and (α_2, a, u_2) in plays on words from $W(\sigma)$ such that $\delta^*(s, \downarrow \alpha_1) = \delta^*(s, \downarrow \alpha_2) = q$ and $\sigma(\alpha_1, a) \neq \sigma(\alpha_2, a)$. If σ has no conflicts, then changing it to a strongly regular strategy requires modification only on configurations that do not occur in plays on words from $W(\sigma)$.

Proof idea of Theorem 1 (c). We show that a dominant strategy σ with some conflicts can be transformed into a dominant strategy σ' with less conflicts. To do so, we find a configuration $(v, a, *)$ with $\delta^*(s, v) = q$ that has no (q, a) -conflict with any later configuration of the same play. The existence of such a strategy can be concluded from the fact that T is acyclic, hence the only way for a configuration to conflict with a later configuration is if all intermediate moves are *Call*. A strategy without (q, a) -conflict can be obtained by “copying” the substrategy starting from $(v, a, *)$ to any conflicting configuration. \square

Proof idea of Theorem 1 (d). We inductively construct a sequence $(\sigma_1, \sigma_2, \dots)$ of dominant strategies that is either finite and ends with a strategy that has no conflict, or is infinite and converges to such a strategy. In the convergent case, since each strategy in a non-recursive game is terminating, the limit strategy is also dominant by Lemma 2. To construct a strategy σ_{k+1} , the idea is to modify σ_k so as to shift the earliest conflict to a later time in the future. \square

6 Negative Results

Finally, we provide proofs for Theorem 2 (a) and (b).

Proof of Theorem 2 (a). The game $G_1 = (\Sigma, R, T)$ with $\Sigma = \{a\}$, the only replacement rule being $a \rightarrow aa$ and $L(T) = \{a^k \mid k \geq 2\}$ has the stated property. The strategy plays Call exactly if it has not seen any symbol \hat{a} . Since this strategy wins on every word, it is dominant. However, a forgetful strategy that plays Call on the first symbol a is bound to play Call forever, and therefore does not win on any word. On the other hand, a forgetful strategy that plays Read on the first symbol a does not win on the word a and is therefore not dominant, either. \square

Proof of Theorem 2 (b). Let $G_2 = (\Sigma, R, T)$ with $\Sigma = \{a, b, c, d\}$, rule set R given by

$$\begin{aligned} a &\rightarrow b \\ c &\rightarrow ac \\ d &\rightarrow bad \end{aligned}$$

and the target language automaton $T = (Q, \Sigma, \delta_T, q_0, F)$ depicted in Figure 2a. We claim that the regular forgetful strategy $\sigma_{\mathcal{A}}$ based on the automaton \mathcal{A} shown in Figure 2b fulfils $W(\sigma_{\mathcal{A}}) = \Sigma^*$ and is thus dominant. Indeed, by induction on

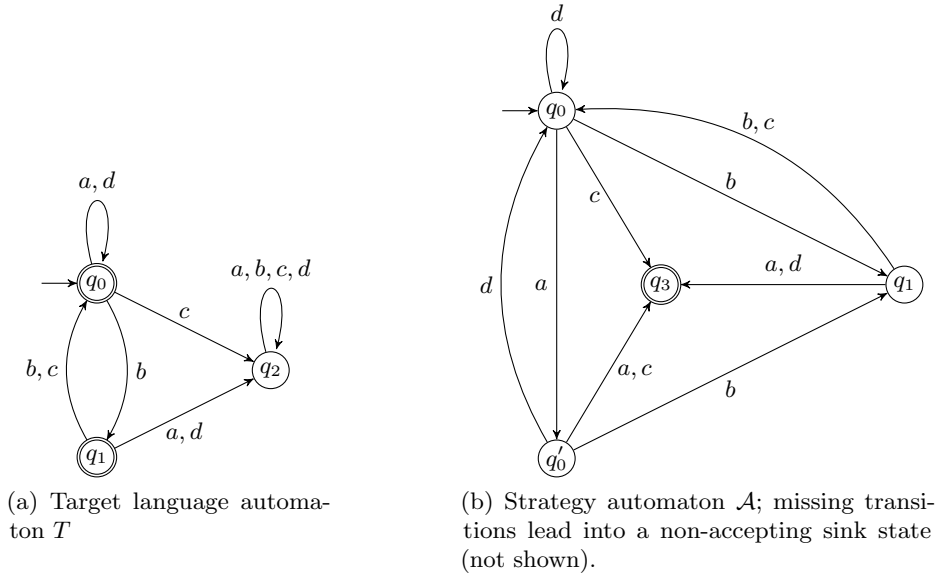


Fig. 2. Automata used in the proof of Theorem 2 (b)

the length of w , the following is easy to show: for each input word w , the strategy

$\sigma_{\mathcal{A}}$ yields a terminating play with a final string u such that: $\delta_T(q_0, u) = \delta_{\mathcal{A}}(q_0, u)$, if u does not end with a , and $\delta_T(q_0, u) = q_0$ and $\delta_{\mathcal{A}}(q_0, u) = q'_0$, otherwise.

However, for a strategy automaton $\mathcal{B} = (Q \cup \{\text{Call}\}, \Sigma, \delta_{\mathcal{B}}, q_0, \{\text{Call}\})$ of a strongly regular strategy $\sigma_{\mathcal{B}}$, it holds that $W(\sigma_{\mathcal{B}}) \subsetneq \Sigma^*$, and thus no such $\sigma_{\mathcal{B}}$ is dominant. For a proof of this claim, we can assume that $\delta_{\mathcal{B}}(q_0, c) = \delta_{\mathcal{B}}(q_1, a) = \delta_{\mathcal{B}}(q_1, d) = \text{Call}$ since otherwise $\sigma_{\mathcal{B}}$ would lose on c , ba or bd . If $\delta_{\mathcal{B}}(q_0, a) = q_0$, then the play of $\sigma_{\mathcal{B}}$ on ac is infinite. On the other hand, if $\delta_{\mathcal{B}}(q_0, a) = \text{Call}$, then the play of $\sigma_{\mathcal{B}}$ on ad is infinite. \square

7 Conclusion and open questions

It remains unclear whether undominated strategies always exist. Our main positive result is that prefix-free replacement languages allow for an undominated strategy that is regular. Indeed, prefix-freeness seems to be a realistic solution in practice, because it can be achieved easily and with almost no overhead by suffixing replacement words with an end-of-file symbol. Also non-recursive rules or a finite target language lead to many positive properties, in particular because they have the bounded depth property. While finiteness of the target language may seem like a strong restriction, one particular instance of it is if there is a single target document that has to be reached [14]. Restrictions that bound the number of recursive replacements also seem plausible in practise.

It is actually conceivable that each game has the bounded depth property (even with a constant bounding sequence) and therefore undominated strategies. Another question is whether every game with an undominated strategy also has a regular one.

References

1. S. Abiteboul, O. Benjelloun, and T. Milo. The Active XML project: an overview. *The VLDB Journal*, 17(5):1019–1040, 2008.
2. S. Abiteboul, A. Bonifati, G. Cob  na, I. Manolescu, and T. Milo. Dynamic XML documents with distribution and replication. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, pages 527–538. ACM, 2003.
3. S. Abiteboul, T. Milo, and O. Benjelloun. Regular rewriting of Active XML and unambiguity. In *PODS*, pages 295–303. ACM, 2005.
4. B. Aminof, A. Legay, A. Murano, O. Serre, and M. Y. Vardi. Pushdown module checking with imperfect information. *Inf. Comput.*, 223:1–17, 2013. doi:10.1016/j.ic.2012.11.005.
5. H. Bj  rklund, M. Schuster, T. Schwentick, and J. Kulbatzki. On optimum left-to-right strategies for active context-free games. In *Joint 2013 EDBT/ICDT Conferences, ICDT ’13 Proceedings*, pages 105–116, 2013.
6. L. Bozzelli. New results on pushdown module checking with imperfect information. In *Proceedings of Second International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2011, Minori, Italy, 15-17th June 2011.*, pages 162–177, 2011. doi:10.4204/EPTCS.54.12.

7. T. Cachat. Symbolic strategy synthesis for games on pushdown graphs. In *Automata, Languages and Programming, 29th International Colloquium, ICALP 2002, Malaga, Spain, July 8-13, 2002, Proceedings*, pages 704–715, 2002. doi:10.1007/3-540-45465-9_60.
8. C. Coester. One-pass strategies for context-free games. Master's thesis, TU Dortmund, 2015. doi:10.17877/DE290R-16466.
9. Christian Coester, Thomas Schwentick, and Martin Schuster. Streaming rewriting games: Winning strategies and complexity. *CoRR*, 2018. URL: <http://arxiv.org/abs/1804.10292>.
10. L. Holík, R. Meyer, and S. Muskalla. Summaries for context-free games. In *36th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2016, December 13-15, 2016, Chennai, India*, pages 41:1–41:16, 2016. doi:10.4230/LIPIcs.FSTTCS.2016.41.
11. T. Milo, S. Abiteboul, B. Amann, O. Benjelloun, and F. D. Ngoc. Exchanging intensional XML data. *ACM Transactions on Database Systems*, 30(1):1–40, 2005.
12. M. Schuster. Transducer-based rewriting games for active XML. In *41st International Symposium on Mathematical Foundations of Computer Science, MFCS 2016, August 22-26, 2016 - Kraków, Poland*, pages 83:1–83:13, 2016. doi:10.4230/LIPIcs.MFCS.2016.83.
13. A. Muscholl, T. Schwentick, and L. Segoufin. Active context-free games. In *STACS 2004, 21st Annual Symposium on Theoretical Aspects of Computer Science*, pages 452–464, 2004.
14. A. Muscholl, T. Schwentick, and L. Segoufin. Active context-free games. *Theory of Computing Systems*, 39(1):237–276, 2006.
15. M. Schuster and T. Schwentick. Games for Active XML revisited. In *18th International Conference on Database Theory, ICDT 2015*, pages 60–75, 2015.
16. O. Serre. Note on winning positions on pushdown games with $[\omega]$ -regular conditions. *Inf. Process. Lett.*, 85(6):285–291, 2003. doi:10.1016/S0020-0190(02)00445-3.
17. I. Walukiewicz. Pushdown processes: Games and model-checking. *Inf. Comput.*, 164(2):234–263, 2001. doi:10.1006/inco.2000.2894.