Streaming Rewriting Games: Winning Strategies and Complexity

Christian Coester*¹, Thomas Schwentick², and Martin Schuster³

¹University of Oxford
²TU Dortmund
³University of Edinburgh

April 30, 2018

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 or undominated strategies for the first player exist and investigates the complexity of some related algorithmic problems.

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 [10] 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 [8]. These embedded calls can be invoked to materialise more data. As an example (adapted from [8, 10]; see Figure 1), 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 (see Figure 1b). The data returned by the service call may contain further 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 [8]. 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 string [1], 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 [10]. 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). It turns out that, with this restriction, the set of strings, on which Juliet can win, is regular. Furthermore, she even has a uniform strategy that wins on all these strings and can be computed by a finite automaton (in a way that will be made more precise in Section 3).

With applications in mind, in which 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 $bc$. This example shows that even for some extremely simple games and input strings, there is no dominant strategy [2] 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. In this paper, we

---

1The restriction to strings instead of trees was justified in [8].
2Such strategies were called optimum in [3]. Undominated strategies were called optimal there.
consider a third kind of “optimality” which lies between the two former notions. We define a
linear order $\leq_{sl}$ (the shortlex order) on strings and languages and call a strategy \textit{weakly dominant}
if its winning set is maximal with respect to this order. In the example above, assuming the usual
order on the alphabet, the strategy which plays \textit{Call} on the initial \textit{a} is weakly dominant.

Abiteboul et al. \cite{3} also introduced \textit{regular strategies}, a simple type of one-pass strategies
defined by a finite state automaton, and therefore efficiently computable. In this paper, we also
study a particularly simple form of regular strategies, called \textit{strongly regular}, computed by an
automaton that is derived from the minimal automaton for the target language. We refer to
Section 2 for precise definitions of these notions.

\textbf{Contributions} \quad Since dominant strategies are so elusive, we broaden our view towards weakly
dominant and undominated strategies and, at the same time, study conditions of games that
guarantee such strategies. In particular, in Section 3 we study the following three questions.

- Under which circumstances does a context-free game have a dominant, weakly dominant
  or undominated one-pass strategy?

- When can such a strategy even be chosen regular or strongly regular?

- When can such a strategy be \textit{forgetful}, in that it does not need to remember all decisions it
  made, but only the (prefix of the) current string?

We identify various conditions on games that yield positive results, summarised in Theorem 3.2.
The first one, the \textit{bounded-depth property}, is of a semantical nature and guarantees the existence
of weakly dominant strategies. The other conditions are syntactical: \textit{prefix-free games}, \textit{non-
recursive games} and games with a finite target language have the bounded-depth property and
therefore weakly dominant strategies. For the most natural condition, prefix-free games, there
is always a regular weakly dominant strategy. If a non-recursive game or a game with a finite
target language has a dominant strategy then it even has a strongly regular one. It remains open
whether all context-free games have undominated strategies and, for that matter, whether there
exist games which lack the bounded-depth property.

We complement these results by further negative results (Theorem 3.14): there exist games
with a dominant strategy, but without a forgetful one; and there exist games with a regular and
forgetful dominant strategy, but without a strongly regular one, and similarly for undominated
strategies. Figure 2 gives an illustration of our results of Section 3.

In the second part, in Section 4 we determine the complexity of the following decision
problems for regular strategies.

- Given a game, a regular strategy and a word, does the strategy win on the word?

- Given a game and a word, does a (strongly) regular winning strategy for the word exist?

- Given two regular strategies, does one dominate the other?

The results are summarised in Theorem 4.1.

This paper is based on the Master’s thesis of the first author, supervised by the other two
authors \cite{5}. The thesis contains further results, some of which will be mentioned later.

\footnote{As explained later, every game can be transformed into a very similar prefix-free game.}
Related work Further background about AXML is given in [1, 2, 3]. Context-free games were introduced in [4], which is the conference paper corresponding to [5]. 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. One-pass strategies and (forgetful) regular strategies were introduced in [6]. 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 [7]. Extended settings of context-free games with nested words (resembling the tree structure of (A)XML documents) are examined in [8].

2 Preliminaries

We denote the set of strings over an alphabet \( \Sigma \) by \( \Sigma^* \) and the set of non-empty strings by \( \Sigma^+ \). \( \Sigma^k \) 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 \( \mathcal{A} = (Q, \Sigma, \delta, \epsilon, F) \), where \( Q \) is the set of states, \( \Sigma \) the alphabet, \( \delta \subseteq Q \times \Sigma \times Q \) the transition relation, \( \epsilon \in Q \) 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 \( q_0, \ldots, q_n \) 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 \( \delta \) as transition function \( Q \times \Sigma \to Q \) and also use the extended transition function \( \delta^* : Q \times \Sigma^+ \to Q \), 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 \( \Sigma \), a minimal\(^4\) DFA \( T = (Q, \Sigma, \delta, \epsilon, F) \), and a binary relation \( R \subseteq \Sigma \times \Sigma^+ \) such that for each \( a \in \Sigma \), the replacement language \( L_a \) is a 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 \( a \to R_a \). We note that the definition of context-free games assures \( e \notin L_a \).

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 \( \Sigma_f = \{ \alpha \mid a \in \Sigma_f \} \) be a disjoint copy of the set \( \Sigma_f \) of function symbols, and let \( \tilde{\Sigma} = \Sigma \cup \Sigma_f \). A configuration is a tuple \( (\alpha, u) \in \tilde{\Sigma} \times \Sigma^+ \). If \( u \) is non-empty, i.e., \( u = av \) for \( a \in \Sigma \) and \( v \in \Sigma^+ \), then we also denote this configuration by \( (\alpha, a, v) \), consisting of a history string \( \alpha \), a current symbol \( a \in \Sigma \) and a remaining string \( v \in \Sigma^+ \). We denote the set of all (syntactically) possible configurations by \( K \). 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 \( \hat{b} \in \Sigma_f \) then this shall denote that Juliet’s \( i \)-th move was to call \( b \). The remaining string \( v \) is the string of symbols that have not been revealed to Juliet yet. By \( \tilde{\zeta} : \tilde{\Sigma} \to \Sigma^* \) we denote\(^5\) the homomorphism which deletes all symbols from \( \Sigma_f \) and is the identity on \( \Sigma \). We call \( \delta^*(s, \tilde{\zeta}\alpha) \) the T-state of the configuration \((\alpha, u)\).

\( ^4 \)The assumption that \( T \) is minimal will be convenient at times.

\( ^5 \)We usually omit brackets and write, e.g., \( \tilde{\zeta}(\alpha\beta) \) for \( \tilde{\zeta}(\alpha, \beta) \).
A play is a sequence of configurations, connected by moves. In one move at a configuration $(\alpha, a, v)$ 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 $\Pi = (K_0, K_1, K_2, \ldots)$ of configurations with the following properties:

(a) The initial configuration is of the form $K_0 = (\varepsilon, w)$, 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 $K_{n+1} = (\alpha \hat{a}, xv)$ 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 $x$.

(c) If $K_n = (\alpha, \varepsilon)$, then $K_n$ is the last configuration of the sequence. Its history string $\alpha$ is called the final history string of $\Pi$. Its final string is $\hat{\alpha}$.

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

The (unique) play of $\sigma$ and $\tau$ on $w$ is a play $\Pi(\sigma, \tau, w) = (K_0, K_1, K_2, \ldots)$ 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 $\sigma$ is terminating if each of its plays is finite. The depth of $\sigma$ is the supremum of depths of plays of $\sigma$. 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 $\sigma$ where each play of $\sigma$ has finite depth but depths are arbitrarily large.

A strategy $\sigma$ wins on a string $w \in \Sigma^*$ if every play of $\sigma$ on $w$ is winning (for Juliet). By $W(\sigma) = W_G(\sigma)$ we denote the set of words on which $\sigma$ wins in $G$. In contrast, $\sigma$ loses on $w$ if

---

6Even though we think of Romeo as an omniscient adversary, it is not necessary to provide the remaining string as an argument to $\tau$: The remaining string is uniquely determined by the input word and his own and Juliet’s previous moves.
there exists a losing play of \( \sigma \) on \( w \). Note that \( \sigma \) neither wins nor loses on \( w \) if there exists an infinite play of \( \sigma \) on \( w \) but no losing play of \( \sigma \) on \( w \).

A strategy \( \sigma \) dominates a strategy \( \sigma' \) if \( W(\sigma') \subseteq W(\sigma) \). It strictly dominates \( \sigma' \), if \( W(\sigma') \subset W(\sigma) \). A strategy \( \sigma \) is dominant if it dominates every other (one-pass) strategy. It is undominated if there is no other strategy that strictly dominates it.

We also consider a third, intermediate, type of optimality. To define it, we fix some total order \( \leq \) of the alphabet \( \Sigma \). 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 if \( |v| = |w| \) and \( v \) precedes \( w \) in the lexicographical order. We extend this to a total order \( \leq_{\text{sl}} \) 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 \( \leq_{\text{sl}} \)) 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 \( V \subseteq W \) then \( V <_{\text{sl}} W \). A strategy \( \sigma \) is weakly dominant if, for every strategy \( \sigma' \) 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 \( \leq_{\text{sl}} \).

A strategy \( \sigma \) is regular if the set \( L \) of strings \( \alpha a \) with \( \sigma(\alpha, a) = \text{Call} \) is regular. In this case, a DFA \( A \) for \( L \) is called a strategy automaton for \( \sigma = \sigma_A \). A strategy is forgetful if its decisions are independent of symbols from \( \Sigma_f \) in the history string, i.e., if \( \sigma(\alpha, a) = \sigma(\beta, a) \) whenever \( \exists \alpha = \exists \beta \). It is not hard to show that a strategy is regular and forgetful if and only if \( L' = \{ \alpha a | \sigma(\alpha, a) = \text{Call} \} \) is regular. A DFA \( A \) for \( L' \) is also called a strategy automaton, and we write \( \sigma_A = \sigma \) 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 \( A = (Q \cup \{ \text{Call} \}, \Sigma, \delta_A, s, \{ \text{Call} \}) \) with \( \delta_A(q, a) \in \{ \delta(q, a), \text{Call} \} \), for each \( q \) and \( a \), then \( \sigma_A \) is called strongly regular.

The following lemma yields a convenient property of one-pass strategies.

**Lemma 2.1.** For each game \( G \) and strategy \( \sigma \) there is a terminating strategy \( \sigma' \) s.t. \( W(\sigma) \subseteq W(\sigma') \).

**Proof.** The strategy \( \sigma' \) plays like \( \sigma \) except that once it is apparent that Romeo can force an infinite play against \( \sigma \), all remaining moves of \( \sigma' \) are Read.

For a formal definition, consider some \( \alpha \in \Sigma^* \) and \( a \in \Sigma_f \). If there exists a play \( \Pi \) of \( \sigma \) that contains a configuration \( (\alpha, a, v) \) for some \( v \in \Sigma^* \) such that no later configuration with remaining string \( v \) occurs in \( \Pi \), then let \( \sigma'(\alpha, a) = \text{Read} \) and \( \sigma'(\alpha a \beta, b) = \text{Read} \) for each \( \beta \in \Sigma^* \) and \( b \in \Sigma_f \). For all elements of the domain for which \( \sigma' \) is not already defined by this, we define \( \sigma' \) like \( \sigma \). Clearly, \( \sigma' \) is terminating and \( W(\sigma) \subseteq W(\sigma') \).

\( \square \)

### 3 Existence of dominant and undominated strategies

One-pass strategies are a restriction of left-to-right strategies, where Juliet’s moves can depend on the whole current configuration, including the remaining string. Every context-free game has a dominant left-to-right strategy for Juliet, i.e., a strategy that dominates all other left-to-right strategies \([9,10]\). Furthermore, there is such a strategy whose moves can be computed by a DFA that reads the current \( T \)-state, the current symbol and the remaining string. Due to the ignorance
of Juliet regarding the remaining string, this result fails to hold for one-pass strategies. It was shown in [3, Example 2.6] that dominant (one-pass) strategies need not exist, even for very restricted games.

In this section, we investigate dominant and undominated strategies more deeply. The main results are illustrated in Figure 2. We show that the situation for undominated strategies is much better than for dominant ones. Although it remains unclear whether such strategies exist for all 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-freeness, that guarantees existence of regular undominated strategies. For two other families of restricted games, non-recursive games and games with finite target language, we show that if they have a dominant strategy, there is even a strongly regular one. Finally, games with a unary alphabet also guarantee undominated strategies.

We complement these positive results by some negative findings: in general, the existence of a (regular) dominant strategy does not imply that there is a forgetful one. Furthermore, the existence of a regular and forgetful dominant strategy does not imply that there is a strongly regular one. The same holds with respect to undominated strategies, even for very restricted games.

3.1 Definition. • A game $G = (\Sigma, R, T)$ 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 $\sigma$ for $G$ and each $k \in \mathbb{N}$ there exists a one-pass strategy $\sigma_k$ that wins on each $w \in W(\sigma) \cap \Sigma^{\leq k}$ with plays of depth at most $B_{|w|}$.  

• A game $G = (\Sigma, R, T)$ is prefix-free if each replacement language $L_u$ is prefix-free, that is, there are no $u, v \in L_u$ where $u$ is a proper prefix of $v$.

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

Figure 2: Illustration of existence and implication results with respect to dominant and undominated strategies. Arrows from above indicate unconditional existence, arrows between boxes indicate implication of existence. Crossed out arrows indicate negative results. Arrow types indicate results for restricted game types as shown above. Upward and right-to-left solid arrows are not drawn, they all exist by definition.
• A game $G = (\Sigma, R, T)$ is unary if $|\Sigma| = 1$.

### 3.1 Positive results

Our positive results are summarised in the following theorem.

**Theorem 3.2.** (a) Every game with the bounded depth property has a weakly dominant strategy.

(b) Every prefix-free game has a regular weakly dominant 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.

(e) Every unary game has an undominated strategy.

Since every non-recursive game trivially has the bounded-depth property, it also has an undominated strategy by part (a). Games with a finite target language constitute a further class with the bounded-depth property and with regular undominated strategies [5, Corollary 4.15].

A concept used in the proofs of parts (a) and (d) is the convergence of a sequence of one-pass strategies.

#### 3.3 Definition. A sequence $(\sigma_k)_{k \in \mathbb{N}}$ of strategies converges to a strategy $\sigma$ if for each $n \in \mathbb{N}$ there exists $k_0 \in \mathbb{N}$ such that for each $k \geq k_0$ and $(\alpha, a) \in \Sigma^{\leq n} \times \Sigma$ it holds that $\sigma(\alpha, a) = \sigma_k(\alpha, a)$.

This is the same as the convergence in the metric space of strategies where the metric $d$ is given by $d(\sigma, \sigma') = 0$ and $d(\sigma, \sigma^0) = \frac{1}{n}$ for $\sigma \neq \sigma^0$, where $n$ is minimal such that there exists $(\alpha, a) \in \Sigma^n \times \Sigma$ and $\sigma(\alpha, a) \neq \sigma^0(\alpha, a)$.

**Lemma 3.4 (Convergence Lemma).** 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 $\sigma$. Let $L_1 \subseteq L_2 \subseteq \cdots$ be an infinite sequence of languages such that, for every $k$, $L_k \subseteq W(\sigma_k)$ and let $L \overset{\text{def}}{=} \bigcup_{k \in \mathbb{N}} L_k$.

Then $\sigma$ wins on every word $w \in L$ for which it terminates.

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

#### 3.1.1 Games with the bounded depth property

Before we prove Theorem 3.2 (a), we define some further notation. For a strategy $\sigma$ and some $i \geq 0$, we denote by $\sigma\mid_i$ the restriction of $\sigma$ to the first $i$ rounds of the game. Thus $\sigma\mid_i$ is a mapping $\sigma\mid_i : \Sigma^{\leq i-1} \times \Sigma \to \{\text{Call, Read}\}$ and $\sigma\mid_0$ is the mapping with empty domain.
Proof of Theorem 3.2 (a). Let \( G \) be a context-free game with the bounded depth property and let \( (B_k)_{k \in \mathbb{N}} \subseteq \mathbb{N} \) 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 \( L \overset{\text{def}}{=} \bigcup_k L_k \).

Let \( L_0 = \{ \varepsilon \} \) if \( \varepsilon \) 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 \( \sigma \) 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 \( \sigma \) such that \( L_k \subseteq W(\sigma) \).

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

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 \( \sigma \) 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 \( \rho_k : \Sigma^{\leq k-1} \times \Sigma \to \{ \text{Call, Read} \} \) such that for every \( k \geq 0 \),
\begin{itemize}
  
  \item \( \rho_{k+1} \) extends \( \rho_k \); more precisely: \( \rho_{k+1}|_k = \rho_k |_k \), and
  
  \item for each \( \ell \geq k \) there exists \( \sigma^k_\ell \in S_\ell \) with \( \rho_k = \sigma^k_\ell |_k \).
\end{itemize}

Let \( \rho_0 \) be the mapping with empty domain. Fix \( k \) such that \( \rho_0, \ldots, \rho_k \) are defined and have the stated properties. Since there are only finitely many mappings \( \Sigma^{\leq k} \times \Sigma \to \{ \text{Call, Read} \} \), one of them has to occur infinitely often within \( \{ \sigma^k_\ell |_{\ell \geq k} \} \). Let \( \rho_{k+1} \) be such a mapping. For \( \ell' \geq k+1 \) we can choose \( \sigma^{k+1}_{\ell'} = \sigma^k_{\ell'} \) for some \( \ell \geq \ell' \) with \( \rho_{k+1} = \sigma^k_{\ell} |_{\ell+1} \). This defines a sequence \( \rho_0, \rho_1, \ldots \) with the properties above. Let \( \hat{\sigma} \) be the strategy that is uniquely determined by \( \hat{\sigma}|_k = \rho_k \), for every \( k \). Clearly \( (\rho_k)_{k \in \mathbb{N}} \) converges\(^7\) to \( \hat{\sigma} \).

Thanks to Lemma 3.4 it suffices to show that \( \hat{\sigma} \) terminates on \( L \). Let thus \( w \in L \) and \( \tau \) be a Romeo strategy. We show that the depth of \( \Pi \overset{\text{def}}{=} \Pi(\hat{\sigma}, \tau, w) \) is at most \( B_{|w|} \). Otherwise let \( k \) be such that in the \( k \)th round Juliet does a \textit{Call} move of nesting depth \( B_{|w|} + 1 \). However, \( \hat{\sigma}|_k = \rho_k = \sigma^k_{\ell} |_k \), where \( \ell = \max \{ k, |w| \} \), and \( \sigma^k_{\ell} \in S_\ell \) has depth at most \( B_{|w|} \) on \( w \in L_\ell \), a contradiction. Therefore the depth of \( \Pi \) is at most \( B_{|w|} \) and by König’s Lemma \( \Pi \) is thus finite, completing the proof. \( \square \)

3.1.2 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. In this sense, every game \( G = (\Sigma, R, T) \) can be transformed into a prefix-free game \( G' = (\Sigma', R', T') \) by letting \( \Sigma' = \Sigma \cup \{ S \} \) for some new symbol \( S \notin \Sigma \) that shall denote the end of replacement strings,

\(^7\)Since the \( \rho_k \) are only partially defined, one might consider the strategies \( \sigma_k \) that result from the \( \rho_k \) which take the value \textit{Call} whenever \( \rho_k \) is undefined.
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 $\Sigma$ contains (besides other symbols) numbers $1, \ldots, N$ 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.

We give the proof of Theorem 3.2(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 $a_1, \ldots, a_n$. 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 symbol $a_i$ 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 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 $\rho : \Sigma^* \to Q$ such that $\rho(\varepsilon) = s$ and $(\rho(w), a, \rho(wa)) \in \delta$ for each $w \in \Sigma^*$ and $a \in \Sigma$.

Given a strategy $\rho$ for 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 $\rho$. A strategy $\rho$ is weakly dominant if, for every strategy $\rho'$, it holds $W_{\mathcal{N}}(\rho') \leq_{sl} W_{\mathcal{N}}(\rho)$.

For an NFA $\mathcal{N} = (Q, \Sigma, \delta, s, F)$ and a state $q \in Q$, we write $\mathcal{N}^q$ for the NFA $(Q, \Sigma, \delta, q, F)$ with starting state $q$. We say that a word $w$ is universally accepted by an NFA $\mathcal{N}$, if every run of $\mathcal{N}$ on $w$ is accepting.

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

**Lemma 3.5.** For each NFA $\mathcal{N}$, there exists a DFA $D$ obtained by removing transitions from $\mathcal{N}$ such that $\rho_D$ is a weakly dominant strategy for ONLINENFA($\mathcal{N}$).

**Proof.** We define a sequence $\mathcal{N}_0, \mathcal{N}_1, \ldots$ of NFAs, starting from $\mathcal{N}_0 = \mathcal{N}$. Each $\mathcal{N}_{n+1}$ results from $\mathcal{N}_n$ by removing transitions in the following way. For a state $q$, let $L^q_n$ be the set of words of length at most $n$ that are non-deterministically accepted by $\mathcal{N}_n^q$. For each state $q$ and symbol $a$, we only keep the transitions to those successor states $p$ that maximise $L^n_p$ with respect to $\leq_{sl}$. 

10
Since there are only finitely many transitions in $N$, the sequence $N_0, N_1, \ldots$ becomes stationary after finitely many steps. By $\tilde{N}$ we denote the last NFA of the sequence.

**Claim 3.6.** For each $q$ and $n$, all words in $L_n^q$ are non-deterministically and universally accepted by $\tilde{N}^q$.

**Proof.** We show a slightly stronger claim by induction on $n$: for each $q$, each $n \geq 0$, all words in $L_n^q$ are non-deterministically and universally accepted by $\tilde{N}^q$. For $n = 0$ the statement is trivial.

Let $aw \in L_{n+1}^q$ with $a \in \Sigma$ and $w \in \Sigma^\leq n$. Let $p$ be a state reached after reading the initial $a$ in some accepting run of $N_{n+1}$ on $aw$. Then $w \in L_{\rho}^p$. By definition of $N_{n+1}$, $L_{\rho}^n = L_{\rho}^p$ for every state $r$ reachable from $q$ by reading $a$ in $N_{n+1}$. Therefore, by the induction hypothesis, $w$ is universally accepted by $N_{\rho}^{q'}$, for any such $r$. Thus, $aw$ is universally accepted by $N_{\rho}^{q'}$.

Since each $N_{i+1}$ results from $N_i$ by removing transitions, every word that is universally accepted by $N_i$ is also universally accepted by $N_{i+1}$. On the other hand, since each state has a transition for each symbol, every universally accepted word is also nondeterministically accepted. Therefore, for each $q$, $n$, all words in $L_n^q$ are non-deterministically and universally accepted by $\tilde{N}^q$, for every $m \geq n$, and therefore also by $\tilde{N}^q$.

**Claim 3.7.** For each strategy $\rho$ for $\text{ONLINEFNA}(N)$, $W_{\mathcal{N}}(\rho) \leq_{sl} L(\tilde{N})$.

**Proof.** We show a more precise statement: for each $q$, each $n \geq 0$, and each strategy $\rho$ for $\text{ONLINEFNA}(N^q)$, $W_{\mathcal{N}}(\rho) \cap \Sigma^\leq n \leq_{sl} L_n^q$. Claim 3.7 easily follows from this statement and Claim 3.6.

Towards a contradiction, assume that there are $n, q, \rho$ with $W_{\mathcal{N}}(\rho) \cap \Sigma^\leq n >_{sl} L_n^q$. Let $n$ be minimal with this property and let $w \in (W_{\mathcal{N}}(\rho) \cap \Sigma^\leq n) \setminus L_n^q$ be chosen minimal with respect to $\leq_{sl}$. Clearly, for all $w' <_{sl} w$ it holds $w' \in (W_{\mathcal{N}}(\rho) \cap \Sigma^\leq n) \iff w' \in L_n^q$.

Clearly $n > 0$, so we can write $w = au$ for $a \in \Sigma$ and $u \in \Sigma^\leq n-1$. Let $p = \rho(a)$. Since $n$ is minimal, for the strategy $\rho'$ induced by $\rho$ for $N^p$, defined by $\rho'(v) = \rho(av)$, it holds

$$W_{\mathcal{N}}(\rho') \cap \Sigma^\leq n-1 \leq_{sl} L_{n-1}^p. \quad (1)$$

Let $r$ be some state for which $(q, a, r)$ is a transition in $N_n$. We claim

$$W_{\mathcal{N}}(\rho') \cap \Sigma^\leq n-1 >_{sl} L_{n-1}^p. \quad (2)$$

Towards a contradiction assume otherwise. Since $u \in W_{\mathcal{N}}(\rho') \cap \Sigma^\leq n-1$ and $u \notin L_{n-1}^p$, there must be some $u' <_{sl} u$ with $u' \notin W_{\mathcal{N}}(\rho') \cap \Sigma^\leq n-1$ and $u' \in L_{n-1}^p$. However, then $au' \notin W_{\mathcal{N}}(\rho) \cap \Sigma^\leq n$ and $au' \in q^\rho$, a contradiction, since $au' \leq_{sl} au = w$.

(1) and (2) imply that $(q, a, p)$ is not a transition in $N_n$. Let $m \leq n-1$ be maximal such that $(q, a, p)$ is a transition in $N_m$. Then $L_{m}^p <_{sl} L_{m}^\rho$ and hence $L_{n-1}^p <_{sl} L_{n-1}^\rho$, contradicting (1) and (2).

Let $D$ be any DFA resulting from $\tilde{N}$ by removing some further transitions. By Claim 3.6, $L(D) = L(\tilde{N})$. Therefore, by Claim 3.7, $\rho_D$ is weakly dominant, concluding the proof of Lemma 3.8.
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 $a \in \Sigma$, $A_a = (Q_a, \Sigma, \delta_a, s_a, \{f_a\})$ be a minimal DFA for the replacement language $L_a$ of $a$. Since $L_a$ is prefix-free, $A_a$ has a unique accepting state $f_a$.

For a strategy $\sigma$ (for Juliet or Romeo) and a string $\alpha \in \Sigma^*$, we define the substrategy $\sigma^\alpha$ of $\sigma$ 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 $\sigma$ on $w$ if the initial state of $T$ were $q$. More precisely, it is the set of states of the form $\delta^*(q, \cdot; \alpha)$ where $\alpha$ is a final history string of a play of $\sigma$ 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 $\sigma$ if $\text{states}(p, a, \sigma) \subseteq S$. 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 $\sigma$ is the set of all its effect triples. Finally, we define the effect set $E(\sigma)$ of a strategy $\sigma$ as $E(\sigma) \overset{\text{def}}{=} \bigcup_{\alpha \in \Sigma^*} \text{sse}(\sigma^\alpha)$. That is, $E(\sigma)$ contains all effect triples that are induced by substrategies of $\sigma$.

With a set $E$ of effect triples we associate an NFA $N_E = (\mathcal{P}(Q), \Sigma, \delta_E, \{s\}, \mathcal{P}(F))$, where $\delta_E$ 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 3.8.** Let $G = (\Sigma, R, T)$ be a prefix-free game, $E$ a set of effect triples, and $\sigma$ a terminating strategy such that $E(\sigma) \subseteq E$. Then there is a strategy $\rho$ for $\text{ONLINE} \text{NFA}(N_E)$ such that $W_G(\sigma) = W_{N_E}(\rho)$.

**Proof.** It is straightforward to verify that $\rho(w) \overset{\text{def}}{=} \text{states}(s, w, \sigma)$ yields a well-defined strategy $\rho$ for $\text{ONLINE} \text{NFA}(N_E)$. The proposition follows, since $w \in W_{N_E}(\rho)$ if and only if $\rho(w) \subseteq F$, and $w \in W_G(\sigma)$ if and only if $\text{states}(s, w, \sigma) \subseteq F$. \hfill $\square$

We say that a strategy automaton $A = (Q_A, \Sigma, \delta_A, s_A, F_A)$ is $(p, a, S)$-inducing if $\sigma_A$ is terminating, $a \in \Sigma_f$, and the following conditions hold.

- For each $u \in L_a$, $\text{states}(p, u, \sigma_A) \subseteq S$.
- There are disjoint subsets $Q_{A,q} \subseteq Q_A$, for $q \in S$, such that for every play of $\sigma$ on some $u \in L_a$ with final history string $\alpha$, it holds $\delta_A^2(s_A, \alpha) \in Q_{A,q} \iff \delta^*(p, \cdot; \alpha) = q$.
  Furthermore, there is no proper prefix $\beta$ of $\alpha$ for which $\delta_A(s_A, \beta) \in Q_{A,r}$, for any $r$.

**Proposition 3.9.** 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 $A_t$. Then there is a strategy automaton $A$ for $G$ such that, for each strategy $\rho$ for $\text{ONLINE} \text{NFA}(N_E)$, $W_{N_E}(\rho) \leq_{ut} W_G(\sigma_A)$.

**Proof.** Let $D$ be the DFA guaranteed by Lemma 3.5 giving the weakly dominant strategy $\rho_D$ for $\text{ONLINE} \text{NFA}(N_E)$. It suffices to construct $A$ such that $W_{N_E}(\rho_D) \subseteq W_G(\sigma_A)$.

The strategy automaton $A$ has a non-accepting state $(p, S)$ for each $S \subseteq Q$ and $p \in S$, further states of the form $(S, r)$, where $r$ is a state of some $A_t$, and $(S, r)$ is accepting if and only if $r$ is accepting in $A_t$, and an accepting state Call. The initial state is $(s, \{s\})$. We construct $A$ such that,
if \( \alpha \) is the final history string of a play of \( \sigma_A \) on \( w \), then \( \delta_A^*(s_A, \alpha) = (p, S) \), where \( S = \rho_D(w) \), \( p = \delta^*(s; \alpha) \), and \( p \in S \). Thus, whenever a subplay on a prefix of the input word is completed, \( A \) knows the state \( p \) that has been reached, and the state \( S \) that \( p \) has reached in \( N \).

To this end, \( \delta_A \) is defined as follows. Let \((p, S)\) be the current state and \( u \) the next symbol and let \( S' = \delta_D(S, a) \). If \( \delta(p, a) \in S' \) then from \( p \) a state in \( S' \) can be reached by playing \text{Read} \ on \( a \) and thus \( \delta_A((p, S), a) = (\delta(p, a), S') \). Otherwise, \( \delta_A((p, S), a) = \text{Call} \). In the latter case, we can choose some \( t = (p, a, S^\prime) \in E \) with \( S^\prime \subseteq S' \), corresponding to a \( \sigma \)-inducing automaton \( A \).

To continue the play after playing \text{Call} \ on \( a \), let \( \delta_A((p, S), a) = (S', s_i) \), where \( s_i \) is the initial state of \( A \). That is, a subrun is started in which \( A_i \) is simulated on the replacement word of \( a \), and \( S' \) is laid aside.

To carry out the simulation of \( A_i \), we simply define \( \delta_A((S', x), b) = (S', \delta_i(x, b)) \) for every state \( x \) of \( A_i \) and every \( b \in \Sigma \), unless \( \delta_i(x, b) \in Q_{A_i, q} \) for some \( q \). If \( \delta_i(x, b) \in Q_{A_i, q} \), then \( \delta_A((S', x), b) = (q, S') \), ending the subgame.

By induction on the length of \( w \), it is straightforward to show that \( A \) meets the above specification. In particular, whenever \( \rho_D \) wins for a word \( w \) in \text{ONLINEFNA}(G_N), A \) yields a strategy that ends in a state of the form \((p, S)\) with \( S \subseteq F \) and \( p \in S \) and it holds \( \delta^*(s; \alpha) = p \in F \), for the final history string \( \alpha \). Therefore, \( W_{G_N}(\rho_D) \subseteq W_G(\sigma) \) indeed holds.

The next lemma gives a crucial property of prefix-free games. Intuitively, it says that Juliet always knows the maximal prefix \( u \) of the input word that is completely processed, the history string \( \alpha \) corresponding to the subgame on \( u \), and the next symbol \( b \) of the input word that is currently processing.

**Lemma 3.10.** Let \( (\gamma, c, v) \) be a configuration in a play \( \Pi(\sigma, \tau, w) \) of a prefix-free game \( G = (\Sigma, R, T) \). Then \( \gamma \) and \( c \) uniquely determine \( u \in \Sigma^*, b \in \Sigma \), and a prefix \( \alpha \) of \( \gamma \) such that \( ub \) is a prefix of \( w \), \( \alpha \) is the final history string of \( \Pi(\sigma, \tau, u) \), and \( u \) has maximal length.

**Proof.** Let \( (K_0, K_1, \ldots, K_n) \) be the prefix of \( \Pi(\sigma, \tau, w) \) with \( K_i = (\gamma_i, c_i, v_i) \), for each \( i \), and \( (\gamma_0, c_0) = (\gamma, c) \). We show by backwards induction that for \( i = n, n - 1, \ldots, 0 \), there are unique \( u_i, b_i, \alpha_i \) such that \( \gamma_i \) is a prefix of \( \gamma \), \( u_i b_i \) is a prefix of \( c_i v_i \), \( \alpha_i \) is the final history string of \( \Pi(\sigma^\gamma, \tau^\gamma, u_i) \), and \( u_i \) has maximal length. In a nutshell, from \( K_i \) onwards, the string to be played on is \( c_i v_i \). The string \( u_i b_i \) is the prefix of \( c_i v_i \) that has been “touched” by the play from \( K_i \) to \( K_{n-1} \). The subplay on \( u_i \) yielded the additional history string \( \alpha_i \). The subplay on \( b_i \) is ongoing at \( K_{n-1} \). The lemma then follows by choosing \( (u_i, b_i, \alpha_i) = (u_0, b_0, \alpha_0) \).

The base case is \( u_n = \alpha_n = \varepsilon \) and \( b_n = c_n \). For the induction step, either \( \gamma = \gamma_{i-1} c_{i-1} \) or \( \gamma = \gamma_{i-1} c_{i-1} \). In the former case, \( u_{i-1} = c_{i-1} u_i, b_{i-1} = b_i, \alpha_{i-1} = \alpha_{i-1} \). Otherwise, we consider two sub-cases. If \( u_i = x x' \) for some \( x \in L_{c_i} \) and \( x' \in \Sigma^* \) (here, \( x \) is uniquely determined thanks to prefix-freeness of \( \Sigma_L \)), then \( \tau(\gamma_{i-1}, c_{i-1}) = x \). Thus, we have \( K_i = (\gamma_{i-1} c_{i-1}, \tau(\gamma_{i-1}, c_{i-1}) x' b_i) \) for some \( z \) and hence \( K_{i-1} = (\gamma_{i-1}, c_{i-1} x' b_i) \). Then \( u_{i-1} = c_{i-1} x', b_{i-1} = b_i, \alpha_{i-1} = c_{i-1} \). Otherwise, the subgame for \( c_{i-1} \) has not ended in \( K_{n-1} \), and thus \( u_{i-1} = \alpha_{i-1} = \varepsilon, b_{i-1} = c_{i-1} \). It is easy to verify in all cases that \( (u_{i-1}, b_{i-1}, \alpha_{i-1}) \) is the unique solution for the required properties.

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 3.2(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 3.11.** In prefix-free games, each effect triple of a terminating strategy is also an effect triple of a strategy of bounded depth.

**Proof.** Let $E$ denote the set effect triples of all strategies of bounded depth. For $t \in E$ fix such a bounded depth strategy $\sigma_t$ with effect triple $t$. Since $E \subseteq Q \times \Sigma \times P(Q)$ is finite, there exists $B$ such that for each $t \in E$, the depth of plays of $\sigma_t$ is at most $B$.

We claim that if $(p, a, S) \notin E$ is an effect triple of strategy $\hat{\sigma}$, then there is a Romeo strategy $\tau$ such that $\Pi(\hat{\sigma}, \tau, a)$ contains a configuration $(\hat{\alpha}a, b, w)$ with $(q, b, S') \notin E$ for $q = \hat{\delta}^*(p, \hat{\gamma}a)$ and $S' = \text{states}(q, b, \hat{\delta}^\alpha a)$. This means that Romeo can repeatedly force a configuration where $\hat{\sigma}$'s substrategy on the current symbol has an effect triple outside of $E$, and hence $\hat{\sigma}$ would not be terminating. Thus, each effect triple of a terminating strategy is in $E$.

It remains to show the claim. Let $(p, a, S) \notin E$ be an effect triple of a strategy $\hat{\sigma}$. Suppose for the sake of contradiction that for each configuration of the form $(\hat{\alpha}a, b, w)$ occurring in some play of $\hat{\sigma}$ on $a$ it holds that $(q, b, S') \in E$, where $q = \hat{\delta}^*(p, \hat{\gamma}a)$ and $S' = \text{states}(q, b, \hat{\delta}^\alpha a)$. We claim that in this case, $(p, a, S)$ is an effect triple of a strategy of depth at most $B + 1$, in contradiction to $(p, a, S) \notin E$.

We first define a strategy $\sigma$ that plays on input words from $L_a$. We define $\sigma(\gamma, c)$ for $(\gamma, c) \in \Sigma' \times \Sigma_f$ such that a configuration of the form $(\gamma, c, v)$ occurs in some play of $\sigma$ on an input word from $L_a$ (given the definition of $\sigma$ for previous configurations). Consider $u, b, \alpha$ determined by $(\gamma, c)$ as per Lemma 3.10. The definition of $\sigma$ will ensure by induction on the length of $u$ that $\text{states}(p, u, \sigma) \subseteq \text{states}(p, u, \hat{\sigma}^\alpha)$. Clearly this holds for $u = \varepsilon$.

Let $q = \hat{\delta}^*(p, \hat{\gamma}a)$. Since $q \in \text{states}(p, u, \sigma) \subseteq \text{states}(p, u, \hat{\sigma}^\alpha)$, there is a history string $\alpha'$ that is the final history string of a play of $\hat{\sigma}^\alpha$ on $u$ such that $q = \hat{\delta}^*(p, \hat{\gamma}\alpha')$. Choose such $\alpha'$ lexicographically minimal. Since $\hat{\sigma}(\gamma, a) = \text{Call}$ (because $(p, a, S) \notin E$) and $ubw \in L_a$ for some $w$, the configuration $(\hat{\alpha}a', b, w)$ occurs in some play of $\hat{\sigma}$ on $a$. Let $S' = \text{states}(q, b, \hat{\sigma}^\alpha a')$. By assumption, $(q, b, S') \in E$. For each non-final configuration $(\beta, c, z)$ occurring in some play of $\sigma_{q, b, S'}$, let $\sigma(\alpha', c) = \sigma_{q, b, S'}(\beta, c)$. Note that $\gamma = \alpha' \alpha$ for some such $\beta$. This defines $\sigma$ until $ub$ is completely processed. Since $S' \subseteq \text{states}(p, ub, \sigma^\alpha)$, we have $\text{states}(p, ub, \sigma) \subseteq \text{states}(p, ub, \hat{\sigma}^\alpha)$, which completes the induction step.

Consider now the strategy $\sigma'$ given by $\sigma'(\varepsilon, a) = \text{Call}$, $\sigma'(\hat{\alpha}a, c) = \sigma(\gamma, c)$ if the latter is defined, and defined as $\text{Read}$ everywhere else. Since

$$\text{states}(p, u, \sigma) \subseteq \text{states}(p, u, \hat{\sigma}^\alpha) \subseteq \text{states}(p, a, \hat{\sigma}) = S$$

for all $u \in L_a$, it follows $\text{states}(p, a, \sigma') \subseteq S$ and hence $(p, a, S)$ is an effect triple of $\sigma'$. But the depth of $\sigma'$ is at most $B + 1$, contradicting $(p, a, S) \notin E$.

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

**Proposition 3.12.** Let $G = (X, R, T)$ be a prefix-free game and $(p, a, S)$ a non-trivial effect triple of some terminating strategy $\sigma$. Then there exists a $(p, a, S)$-inducing strategy automaton.
Proof. Due to Proposition 3.11 we can assume that the depth of $\sigma$ is bounded by some finite $k$. We proceed by induction on $k$.

Since $(p, a, S)$ is non-trivial, $\sigma(e, a) = Call$. Let $\sigma'$ be the strategy defined by $\sigma'(a, b) = \sigma(\alpha, b)$ if a configuration $(\alpha, b, u)$ can occur in a play of $\sigma$ on $a$, for some $u$, and $\sigma'(a, b) = Read$ otherwise. Note that the depth of $\sigma'$ is at most $k - 1$. Moreover, for every $u \in L_\sigma$, it holds $states(p, u, \sigma') \subseteq S$, since $states(p, a, \sigma) \subseteq S$.

Let $T'$ be the target automaton $(Q \times \bar{Q}, \Sigma, \delta \times \delta_a, (p, s_a), S \times \{f_a\})$, that is, the product automaton of $T''$ and $A_a$ whose accepting states are chosen to require $T''$ to end in $S$ and $A_a$ to end in $f_a$. Let $G'$ be the game $(\Sigma, R, T')$. We consider $\sigma'$ as a strategy for $G'$. Since $states(p, u, \sigma') \subseteq S$ for each $u \in L_\sigma$, we have $L_\sigma \subseteq W_G(\sigma')$.

Let $E$ be the set of effect triples $(q, b, S')$ of strategies for $G'$ with depth at most $k - 1$. For each non-trivial $t \in E$, a $t$-inducing strategy automaton exists by induction. Let $A$ be the strategy automaton for the set $E$, defined in the proof of Proposition 3.9.

By Proposition 3.8 there is a strategy $\rho$ for ONLINEFA($\bar{N}_E$) such that $W_G(\sigma') = W_{\bar{N}_E}(\rho)$. By Proposition 3.9, $W_{\bar{N}_E}(\rho) \leq_W W_G(\sigma_A)$. Altogether, we have $L_\sigma \leq_W W_G(\sigma_A)$ and, by definition of $G'$, $W_G(\sigma_A) \subseteq L_\sigma$, hence $W_G(\sigma_A) = L_\sigma$. Therefore, $A$ is $(p, a, S)$-inducing if we choose $Q_q = \{(q, f_a, S') \mid S' \ni (q, f_a)\}$ for every $q \in S$. \qed

Now we are ready to prove Theorem 3.2(b).

Proof of Theorem 3.2(b). Let $G = (\Sigma, R, T)$ be prefix-free. Let $E$ be the set of effect triples of terminating strategies. By Proposition 3.12 there is a $t$-inducing automaton for each non-trivial $t \in E$. Let $\sigma_A$ be the regular strategy as guaranteed by Proposition 3.9. We show that $\sigma_A$ is weakly dominant.

To this end, let $\sigma$ be any terminating strategy for $G$. Since $E(\sigma) \subseteq E$, Proposition 3.8 guarantees a strategy $\rho$ for ONLINEFA($\bar{N}_E$) such that $W_G(\sigma) = W_{\bar{N}_E}(\rho)$. By Proposition 3.9, $W_{\bar{N}_E}(\rho) \leq_W W_G(\sigma_A)$, and therefore, altogether $W_G(\sigma) \leq_W W_G(\sigma_A)$ as required. \qed

3.1.3 Games with finite target language

Proof of Theorem 3.2(c). Let $G = (\Sigma, R, T)$ with $T = (Q, \Sigma, \delta, s, F)$ such that $L(T)$ is finite, and let $\sigma$ be a dominant strategy for $G$. Thanks to Lemma 2.1 we can assume that all plays of $\sigma$ are finite.

We say that $\sigma$ has a $(q, a)$-conflict, for $q \in Q$ and $a \in \Sigma_f$, if there are words $w_1, w_2 \in W(\sigma)$ and strategies $t_1, t_2$ of Romeo such that $\Pi(\sigma, t_1, w_1)$ reaches some configuration $(\alpha_1, a, u_1)$ with $\delta^*(s, \alpha_1) = q$ and $\sigma(\alpha_1, a) = Read$, and $\Pi(\sigma, t_2, w_2)$ reaches some configuration $(\alpha_2, a, u_2)$ with $\delta^*(s, \alpha_2) = q$ and $\sigma(\alpha_2, a) = Call$. By $C(\sigma)$ we denote the set of pairs $(q, a)$, for which $\sigma$ has a $(q, a)$-conflict.

If $C(\sigma) = \emptyset$, then $\sigma$ is already almost strongly regular. The only exceptions might be configurations that are not reached by plays on strings from $W(\sigma)$ and we can easily change $\sigma$ on those configurations to make it entirely strongly regular without affecting $W(\sigma)$.

Thus we only need to show that a dominant strategy $\sigma$ with some conflicts can be transformed into a dominant strategy $\sigma'$ with $C(\sigma') \subseteq C(\sigma)$. By repeating this process at most $|Q||\Sigma_f|$ times, we reach a strategy without conflicts.
To this end, let $\sigma$ have a $(q,a)$-conflict, for some $q$ and $a$, with $\tau_1, \tau_2, \alpha_1, \alpha_2, u_1, u_2$ as above such that $\alpha_1$ has minimal possible length. Let $v_1 = \xi \alpha_1$.

We first claim that $\sigma$ only plays Read moves on the input word $v_1$. As a first step towards this claim, we show $v_1au_1 \in W(\sigma)$. Indeed, the strategy $\tilde{\sigma}$ that plays Read on $v_1$ and is defined by $\tilde{\sigma}(v_1\beta,x) = \sigma(\alpha_1 \beta,x)$ afterwards (and arbitrarily on other configurations) clearly wins on $v_1au_1$, since $\sigma$ wins from $(\alpha_1, a, u_1)$ on. As $\sigma$ is dominant, therefore $v_1au_1 \in W(\sigma)$ holds. As $\sigma$ plays Read on all $\Sigma$-symbols of $\alpha_1$, the assumption that $\sigma$ plays Call on some prefix of $v_1$ would yield a smaller instance of a conflict (with respect to the length of $\alpha_1$), and therefore, the claim that $\sigma$ only plays Read moves on the prefix of $v_1$ is shown.

We say that $\sigma$ is $(q,a)$-conflict-free after $v_1$ if either $\sigma(v_1,a) = \text{Read}$ and there is no $(q,a)$-conflicting Call-move in plays of $\sigma$ on any word $v_1au \in W(\sigma)$, for some $u$, or vice versa with Read and Call reversed. In both cases, a strategy without any $(q,a)$-conflicts can be obtained by “copying” the substrategy of the subgame starting from $(v_1,a,u)$ to any conflicting configuration.

However, we need to show how to guarantee $(q,a)$-conflict-freeness after $v_1$. Actually, thanks to the finiteness of $L(T)$ there can be no such conflict if $\sigma(v_1,a) = \text{Read}$, since $T$ can not return to state $q$ after taking a transition from $q$. But in the case where $\sigma(v_1,a) = \text{Call}$, there could be a conflicting Read-move at some configuration $(\alpha,a,u')$ after $(v_1,a,u)$. However, again thanks to the finiteness of $L(T)$, there can not be any Read moves between $(v_1,a,u)$ and $(\alpha,a,u')$ and therefore, in particular, $u$ must be a suffix of $u'$. In this case we show that actually $u' = u$ must hold. By copying the strategy of $\sigma$ of the subgame on $(\alpha,a,u)$ to the subgame on $(v_1,a,u)$, a strategy that matches the first case above can be obtained.

In all cases, it can be concluded from the dominance of $\sigma$ that the new strategies obtained by the described modifications are again dominant.

We now turn to a more detailed description of the proof.

We first show how to get rid of $(q,a)$-conflicts after $v_1$. Let us thus assume that $\sigma(v_1,a) = \text{Call}$, $v_1au \in W(\sigma)$, and for some strategy $\tau$ of Romeo, $\Pi(\sigma, \tau, v_1au)$ contains a configuration $(\alpha,a,u')$ with $\delta^*(s, \xi \alpha) = q$ and $\sigma(\alpha,a) = \text{Read}$, for some $u'$. As already argued above, there can not be any Read moves between $(v_1,a,u)$ and $(\alpha,a,u')$ and therefore $u$ must be a suffix of $u'$. We claim that $u' = u$.

Towards a contradiction, let us assume that $u' = u''u$ for some $u'' \neq \epsilon$. We show by induction on $k$ that $v_1a(u'')^ku \in W(\sigma)$ for each $k$. This is true for $k = 0$ by assumption. Suppose now that $v_1a(u'')^ku \in W(\sigma)$. Similarly as before, the play $\Pi(\sigma, \tau, v_1au) u''u)$ reaches configuration $(v_1,a, (u'')^ku)$ and later on $(\alpha,a, (u'')^{k+1}u)$. From this configuration on, $\sigma$ is winning. But then the strategy that plays from $(v_1,a, (u'')^{k+1}u)$ on like $\sigma$ plays from $(\alpha,a, (u'')^{k+1}u)$ on wins on the string $v_1a(u'')^{k+1}u$. By dominance of $\sigma$, it follows that $v_1a(u'')^{k+1}u \in W(\sigma)$, completing the inductive step.

However, now we can choose $k$ larger than the length $\ell$ of the longest string in $L(T)$ to get the desired contradiction: $v_1a(u'')^ku$ has more than $\ell$ symbols and each of them contributes at least one symbol from $\Sigma$ to the final history string, which therefore can not yield a win for Juliet. Altogether, we have established that $u' = u$.

We now define $\tilde{\sigma}$ by

$$
\tilde{\sigma}(\beta,x) =
\begin{cases}
\sigma(\alpha \beta', x) & \text{if } \beta = v_1 \beta', \text{ and } \beta' x \text{ begins with } a, \\
\sigma(\beta, x) & \text{otherwise}.
\end{cases}
$$
That $W(\sigma) \subseteq W(\tilde{\sigma})$ is easy to show: if the difference between $\sigma$ and $\tilde{\sigma}$ actually matters for some word $w \in W(\sigma)$, then $w$ must be of the form $v_1aw'$ and the first configuration in which $\sigma$ and $\tilde{\sigma}$ differ is $(v_1,a,w')$. However, since Romeo can enforce the configuration $(\alpha,a,w')$, $\sigma$ is winning from that configuration on. Since $\tilde{\sigma}$ plays from $(v_1,a,w')$ on like $\sigma$ plays from $(\alpha,a,w')$ on, we can conclude that $\tilde{\sigma}$ is winning from $(v_1,a,w')$, as desired.

Therefore, in the following we can assume that $\sigma$ is $(q,a)$-conflict-free after $v_1$.

Now we are ready to get rid of the $(q,a)$-conflict altogether. We define a strategy $\sigma'$ that copies the substrategy of $\sigma$ from configurations of the form $(v_1,a,u)$ onwards to any conflicting configuration. Formally,

$$\sigma'(\beta,x) = \begin{cases} 
\sigma(v_1\beta'',x) & \text{if } \beta = \beta''\beta''', \delta^*(s,\tilde{\beta}') = q, \beta''x \text{ begins with } a \text{ or } \tilde{a}, \\
\sigma(\beta,x) & \text{otherwise.}
\end{cases}$$

Since the sub-plays of $\sigma$ from $(v_1,a,u)$ on do not contain any configurations that can cause a $(q,a)$-conflict, and since clearly no new conflicts are induced, we can conclude that $C(\sigma') \subseteq C(\sigma)$.

To show that $\sigma'$ is again dominant it suffices to show $W(\sigma) \subseteq W(\sigma')$. If the difference between $\sigma$ and $\sigma'$ matters for some word $w \in W(\sigma)$, then the first configuration in which $\sigma$ and $\sigma'$ differ is of the form $(\beta',a,u)$, with $\beta'$ as in the definition of $\sigma'$ and $u \in \Sigma^*$. Since $\sigma$ is winning from $(\beta',a,u)$ on, a strategy that plays $Read$ on $v_1$ and plays from $(v_1,a,u)$ on like $\sigma$ from $(\beta',a,u)$ on wins on the word $v_1au$. Since $\sigma$ is dominant, $\sigma$ wins on $v_1au$ as well. Since $\sigma'$ plays from $(\beta',a,u)$ on like $\sigma$ from $(v_1,a,u)$, it also wins, as desired. $\square$

### 3.1.4 Non-recursive games

**Proof of Theorem 3.2 (d).** Let $G = (\Sigma,R,T)$ with $T = (Q,\Sigma,\delta,s,F)$. Starting with some dominant strategy $\sigma_1$, we inductively construct a sequence $(\sigma_1,\sigma_2,\ldots)$ of dominant strategies that is either finite and ends with a strategy that has no $(q,a)$-conflict, or is infinite and converges to such a strategy. To construct a strategy $\sigma_{k+1}$, the idea is to modify $\sigma_k$ so as to move the earliest $(q,a)$-conflict to a later time in the future.

We describe next how $\sigma_{k+1}$ is defined from $\sigma_k$. We distinguish two cases.

The first case is that for all $q \in Q$, $a \in \Sigma_f$ and all configurations $(\alpha_1,a,u_1)$ and $(\alpha_2,a,u_2)$ with $q = \delta^*(s,\tilde{\alpha}_1) = \delta^*(s,\tilde{\alpha}_2)$ that occur in plays of $\sigma_k$ on strings from $W(\alpha_k)$, it holds that $\sigma_k(\alpha_1,a) = \sigma_k(\alpha_2,a)$. In this case $\sigma_k$ is the last strategy of the sequence.

In the other case, let $w_1,w_2 \in W(\sigma)$ and strategies $\tau_1,\tau_2$ of Romeo such that $\Pi(\sigma_k,\tau_1,w_1)$ reaches some configuration $(\alpha_1,a,u_1)$ with $\delta^*(s,\tilde{\alpha}_1) = q$, $\Pi(\sigma_k,\tau_2,w_2)$ reaches some configuration $(\alpha_2,a,u_2)$ with $\delta^*(s,\tilde{\alpha}_2) = q$ and $\sigma_k(\alpha_1,a) \neq \sigma_k(\alpha_2,a)$, such that $|\alpha_2| \geq |\alpha_1|$ and $(|\alpha_2|,|\alpha_1|)$ is lexicographically minimal.

As in the proof of Theorem 3.2 (c), it can be shown that $\sigma_k$ only plays $Read$ moves on $\tilde{\alpha}_1$ and that it wins on $\tilde{\alpha}_1au_1$. By minimality of $(|\alpha_2|,|\alpha_1|)$ with $|\alpha_2| \geq |\alpha_1|$, we can conclude that actually $\alpha_1 = \tilde{\alpha}_1$.

Now we are ready to define $\sigma_{k+1}$. It plays like $\sigma_k$ except that if it reaches a configuration with $T$-state $q$ and current symbol $a$ after precisely $|\alpha_2|$ moves, then it continues to play like $\sigma_k$ for the
We note that the proof uses non-recursiveness only to guarantee that $\sigma$ does not win but some strategy dominating $\sigma$ the current word. Thus, we can restrict our attention to the case where the replacement language playing Call cannot yield an advantage in this case, since Romeo can choose to never change $j$ state (transitions are precisely the edges $Q\Sigma$ helpful lemma. Let in the following always undominated strategy. Before we turn to this proof, we first present some observations and a 3.1.5 Unary games

$\sigma_k(\alpha, b) = \begin{cases} 
\sigma_k(\alpha, b'), & \text{if } \alpha_1 \text{ is not a prefix of } \alpha, \\
\sigma_k(\beta\alpha', b), & \text{if } \alpha = \alpha_1\alpha' \text{ for some } \alpha' \in \Sigma^*.
\end{cases}
$

On the input word $\alpha_1av$, $\overline{\sigma}$ initially plays only Read moves on the prefix $\alpha_1$. Upon reaching the configuration $(\alpha_1, av)$, it starts playing like $\sigma_2$ would play after reaching the configuration $(\beta, av)$. Since $\sigma_k$ wins from $(\beta, av)$ on, $\overline{\sigma}$ wins on $\alpha_1av$, the desired contradiction. Altogether, $(\sigma_1, \sigma_2, \ldots)$ is indeed a sequence of dominant strategies. If it is finite, then its last strategy $\sigma_k$ has no two conflicting configurations that can occur in plays on words from $W(\sigma_k)$. Thus, it can be turned into a strongly regular dominant strategy by adapting it only for configurations that cannot occur in plays on words where it does not win.

If $(\sigma_1, \sigma_2, \ldots)$ is infinite, it converges towards a strategy $\sigma'$, because, for each $n$, $\alpha_2$ can be of length $n$ at most $|Q||\Sigma|$ times in the above construction. Since $G$ is non-recursive, every play terminates, and therefore Lemma 3.4 guarantees that $\sigma'$ is dominant as well. Finally, $\sigma'$ can be transformed into a strongly regular strategy just as in the finite case. □

We note that the proof uses non-recursiveness only to guarantee that $\sigma'$ has the same winning set as every $\sigma_k$.

3.1.5 Unary games

To complete the proof of Theorem 3.2 we show in this subsection that every unary game has an undominated strategy. Before we turn to this proof, we first present some observations and a helpful lemma. Let in the following always $\Sigma = \{a\}$ and the DFA $T$ for the unary target language is minimal, we can assume that it is of the following form: $Q = \{0, 1, \ldots, m - 1\}$, for some $m$, with $1$ as initial state; the transitions are precisely the edges $(i - 1, i)$ for $i = 1, \ldots, m - 1$ and the edge $(m - 1, j)$ for some state $j \in Q$.

We can assume that $a \notin R_n$, since otherwise always playing Read is a dominant strategy. Indeed, playing Call cannot yield an advantage in this case, since Romeo can choose to never change the current word. Thus, we can restrict our attention to the case where the replacement language contains only words of length at least 2.

We call a strategy $\sigma$ almost undominated if there are only finitely many words on which $\sigma$ does not win but some strategy dominating $\sigma$ does win.
Lemma 3.13. If a game has an almost undominated strategy then it also has an undominated strategy.

Proof. Let \( \sigma_0 \) be an almost undominated strategy, let \( W_0 \) be the finite set of words on which \( \sigma_0 \) does not win but some strategy dominating \( \sigma_0 \) does win, and let \( n = |W_0| \). For each \( i > 0 \), we define \( \sigma_i \) as \( \sigma_{i-1} \) if \( \sigma_{i-1} \) is undominated or as some strategy that dominates \( \sigma_{i-1} \) and wins on the smallest word in \( W_{i-1} \). We define \( W_i \) as the subset of \( W_{i-1} \) of words on which \( \sigma_{i} \) does not win but some other strategy dominating \( \sigma \) does win. Clearly, \( W_n \) is empty and thus \( \sigma_n \) is undominated. \( \square \)

Proof of Theorem 3.2 (e). Thanks to Lemma 3.13 it suffices to show that any unary game has an almost undominated strategy. We may assume that the target language \( L(T) \) is \( m \)-periodic, meaning that \( \delta(m-1, a) = 0 \). The theorem for the general case where \( \delta(m-1, a) > 0 \) is then implied as follows: the target language of such a general automaton differs from some \( (m-j) \)-periodic language by at most the \( j \) shortest words. Hence, if a strategy is almost undominated for the associated \( (m-j) \)-periodic language, then it is also almost undominated for the actual target language. We can also assume that \( |R_0| > 1 \), since otherwise \( G \) is prefix-free.

In the following, we identify a string \( d^\ell \) with the number \( \ell \). Let \( n, k \geq 1 \) be such that \( 1 + n \) and \( 1 + n + k \) are the shortest words in \( R_0 \).

Since every Call extends the remaining string by at least \( n \) symbols, for each strategy there is a strategy with (at least) the same winning set, which always plays at least \( n \) Reads after each Call. We denote such a combined move by \( CR^\infty \). We therefore assume in the following that all strategies of Juliet that we consider are of this normal form and we only consider configurations and history strings where every \( CR^\infty \) is complete. Moreover, whenever the successor state of the \( T \)-state of the current configuration is accepting, Juliet plays Read (because if this Read reaches the end of the word, then Juliet wins, and otherwise she does not hurt her chances to win by delaying Calls until later).

With each configuration \( K \) we associate a profile \( (c, r, \ell) \) consisting of the number \( c \) of \( CR^\infty \)s that were played so far, the number \( r \) of additional Reads played so far (outside \( CR^\infty \) moves)\(^8\) and the length \( \ell \) of the remaining string. We note that history strings for games with unary alphabet are very uniform. Indeed, for arbitrary strategies \( \sigma \) of Juliet, strategies \( \tau, \tau' \) of Romeo, strings \( w, w' \) and \( i \geq 0 \) it holds: if the plays \( \Pi(\sigma, \tau, w) \) and \( \Pi(\sigma, \tau', w') \) last for at least \( i \) moves, then their history strings after \( i \) moves are identical. In particular, for every fixed \( \sigma \), each profile determines uniquely its underlying configuration.

By \( \sigma_0 \) we denote Juliet’s strategy that always reads, and by \( \tau_0 \) and \( \tau_1 \) we denote the strategies of Romeo that always play \( 1 + n \) and always play \( 1 + n + k \), respectively. We say that a strategy \( \sigma \) is patient if there is a string \( w \) such that \( \Pi(\sigma, \tau_0, w) \) has at least \( m \) Call-moves.

Let \( L \) be the union of \( L(T) \) and the set of all strings of the form \( w + ik \) with \( w \in L(T), w \geq m, \) and \( i \in \{1, \ldots, m - 1 \} \). We note that \( L \) is ultimately \( k \)-periodic. More precisely, for all words \( w \) of length at least \( (k + 1)|Q| \) it holds that \( w \in L \) if and only if \( w + k \in L \).

The existence of an almost undominated strategy can be concluded from the following three claims:

\(^8\)Thus the total number of Reads before \( K \) is \( cn + r \).
(a) If $\sigma_0$ is not undominated then there is a patient strategy that dominates $\sigma_0$.

(b) If there is a patient strategy that dominates $\sigma_0$, then there is a strategy that wins on $L$.

(c) Every strategy that wins on $L$ is almost undominated.

Proof of (a):
The proof is by contradiction. We assume that $\sigma_0$ is not undominated but that every strategy $\sigma$ that dominates it, uses at most $C$ Call-moves, for some $C < m$. Let $\sigma$ be a strategy dominating $\sigma_0$ that has maximally $C$ Call moves. Let $w$ be sufficiently large and $K$ be a configuration in the play $\Pi(\sigma, x_0, w)$ with $T$-state 0 with $C$ Call moves in its history string $\alpha$. Let $\sigma'$ be the strategy that plays $\sigma'(\beta, a) = \sigma(\beta, a)$ for strict prefixes $\beta$ of $\alpha$ and $\sigma'(a\beta, a) = \sigma(\beta, a)$, otherwise. It is easy to see that $\sigma'$ has more than $C$ Call moves and dominates $\sigma_0$, the desired contradiction.

Proof of (b):
Let $\sigma$ be a patient strategy that dominates $\sigma_0$. Let $K$ be some configuration with profile $(m, r, \ell)$ and for which Read was played in the last move. We claim that $\sigma$ wins from $K$ on, if $r + \ell - ik < L(T)$, for some $i \in \{0, \ldots, m-1\}$ with $ik \ll \ell$.

To this end, let $\tau^j$ denote the Romeo strategy that plays $1 + n + k$ during its first $j$ moves and $1 + n$ afterwards.

Let thus $w = r + \ell - ik$. Since $w \in L(T)$, $\sigma$ wins on $w$. In particular, $\sigma$ wins against strategy $\tau'$.

It is easy to see that the play $\Pi(\sigma, \tau', w)$ reaches a configuration with the same profile as $K$ and thus, as explained before, $K$. Since $\sigma$ wins on $w$, it thus wins all plays from $K$ on.

Next, we are going to construct from $\sigma$ a strategy $\sigma'$ that wins on $L$. To this end, let $w \in L(T)$ be such that $\Pi(\sigma, x_0, w)$ has at least $m$ Call-moves. Let $(m, r, \ell)$ be the profile of the configuration $K$ that is reached in $\Pi(\sigma, x_0, w)$ after the $m$-th Call move. By the above claim, $\sigma$ wins from every configuration $(m, r, \ell')$, for which $r + \ell' - ik < L(T)$, for some $i \in \{0, \ldots, m-1\}$ with $ik < \ell'$. The $T$-state of $K$ is $r' = r \mod m$.

Strategy $\sigma'$ is defined as follows. It first does $r'$ Read moves, yielding some configuration $K'$ with $T$-state $r'$. From there on, it copies the strategy of $\sigma$ in plays that start from $K$. More precisely, let $\alpha$ be the history string of $K$ and $\alpha' = r'$ be the history string of $K'$. Then, for every history string $\beta$, $\sigma'$ plays on $\alpha' \cdot \beta$ like $\sigma$ on $\alpha \cdot \beta$. We can conclude that $\sigma'$ not only wins on every string $v \in L(T)$ but also on every string of the form $v + ik$ with $v \in L(T)$, $v \geq r'$ and $i \in \{0, \ldots, m-1\}$.

Therefore, $\sigma'$ is a winning strategy for $L$, and (b) holds.

Proof of (c):
We show that there is no strategy for Juliet that wins on $L$ and some word $w \notin L$ of length at least $(k + 1)m$.

Let $d$ be the ultimate number of words in $L$ per $k$ words, that is let $d := |L \cap \{km + 1, \ldots, km + k\}|$ be the number of words in $L$ per every $k$ adjacent words. Suppose some strategy $\sigma'$ wins on $L$ and on some word $w \notin L$ of length at least $km$, and let $L' = L \cup \{w\}$. In particular, the play $\Pi(\sigma', \tau, v)$ terminates, for every $v \in L'$ and every strategy of Romeo. Towards a contradiction,
we prove by induction that, for every \( t \geq 1 \), \( \Pi(\sigma', \tau_1, w) \) reaches some configuration \( K_m \) with profile \((t, r, \ell)\), for some \( r, \ell \). Clearly, this contradicts the assumption that \( \Pi(\sigma', \tau_1, w) \) terminates. For \( m = 1 \), the claim follows since \( w \notin L(T) \).

For the inductive step, let \( K \) be the first configuration in \( \Pi(\sigma', \tau_1, w) \) with profile \((m, r, \ell)\), for some \( r, \ell \). Clearly, the move that leads to \( K \) is a \( CR^n \)-move and \( \ell > k \), since Romeo played \( 1 + n + k \). If \( \Pi(\sigma', \tau_1, w) \) makes a further \( CR^n \)-move after \( K \), we are done with the inductive step, thus we assume it does not. Since \( \ell > k \), the play must therefore reach a configuration \( K'' \) with profile \((m, r', k)\), for some \( r' \). Let \( \alpha \) be its history string.

By \( w_1, \ldots, w_d \) we denote the minimal \( d \) words that are longer than \( w \), and for which \( \sigma' \) is a winning strategy. For every \( i \leq d \), let \( c_i = |w_i| - |w| \). By the choice of \( d \) and since \( L \) is \( k \)-periodic beyond \( km \), it follows that \( c_d \leq k - 1 \).

Let \( \tau' \) denote the strategy that deviates from \( \tau_1 \) (only) by answering the \( CR^n \)-move at configuration \( K' \) by \( 1 + n \). It is easy to see that, for each \( i \leq d \), the play \( \Pi(\sigma', \tau', w_i) \) reaches, following the same history string \( \alpha \), a configuration \( K'' \) with profile \((t, r', c_i)\). Since we assume that \( \sigma' \) wins from all these configurations, and there can be at most \( d \) accepting states within the next \( k \) states from the state of \( T \) for \( K'' \), \( \sigma' \) has to play \( CR^n \) at least once during the next \( k \) steps. The play \( \Pi(\sigma', \tau_1, w) \) thus reaches a configuration with profile \((t + 1, r'', \ell')\) with some \( r'' \) and \( \ell' \), completing the inductive step.

\[ \square \]

### 3.2 Negative results

In the previous subsection we showed the strongly positive result that, on non-recursive games or games with finite target languages, the existence of a dominant strategy always implies that there exists a strongly regular dominant strategy. We now turn to the negative results of this section and show that the above implication does not generalise to arbitrary games or to undominated strategies; in fact, the following theorem states that even much weaker statements do not hold in general.

**Theorem 3.14.** (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.

**Proof.** (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 \( \tilde{a} \). 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.

---

9We recall that \( \tau_1 \) is the strategy of Romeo that always chooses the answer string \( 1 + n + k \).
(b) Let $G_2 = (\Sigma, R, T)$ with $\Sigma = \{a, b, c, d\}$, rule set $R$ given by
\[
\begin{align*}
    a &\rightarrow b \\
    c &\rightarrow ac \\
    d &\rightarrow bad
\end{align*}
\]
and the target language automaton $T = (Q, \Sigma, \delta_T, q_0, F)$ depicted in Figure 3a. We claim that the regular forgetful strategy $\sigma_A$ based on the automaton $A$ shown in Figure 3b fulfils $W(\sigma_A) = \Sigma^*$ and is thus dominant. Indeed, by induction on the length of $w$, the following is easy to show: for each input word $w$, the strategy $\sigma_A$ yields a terminating play with a final string $u$ such that: $\delta_T(q_0, u) = \delta_A(q_0, u)$, if $u$ does not end with $a$, and $\delta_T(q_0, u) = q_0$ and $\delta_A(q_0, u) = q_0'$, otherwise.

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

(c) For the analogon of (a), let $G_1 = (\Sigma, R, T)$ over the alphabet $\Sigma = \{a, b, c, d, e\}$ with the set of rules $R$ given by
\[
\begin{align*}
    a &\rightarrow b + c \\
    b &\rightarrow cd \\
    c &\rightarrow e
\end{align*}
\]
and target language $L(T) = \{ e, cd \}$. It is not hard to see that the strategy that plays Call on $(\varepsilon, a)$, $(\varepsilon, b)$, $(\varepsilon, c)$, $(\hat{a}, b)$ and $(\hat{a}, c)$ and Read otherwise is regular, wins on $\{ a, b, c, e \}$, and is undominated.

Towards a contradiction, let us assume that $\sigma$ is a forgetful strategy for $G_1$ that is undominated. We show first that $\sigma$ does not win on $a$. Suppose $\sigma$ does win on $a$. Then $\sigma(\varepsilon, a) = \sigma(\varepsilon, b) = \text{Call}$. If $\sigma(\varepsilon, c) = \text{Read}$, then the play $(\varepsilon, a), (\hat{a}, c), (\hat{ac}, \varepsilon)$ of $\sigma$ on $a$ is losing – contradiction. If $\sigma(\varepsilon, c) = \text{Call}$, then the play

$$(\varepsilon, a), (\hat{a}, b), (\hat{ab}, cd), (\hat{ab}c, ed), (\hat{abc}e, d), (\hat{abc}ed, \varepsilon)$$

of $\sigma$ on $a$ is losing – contradiction.

It is also easy to see that no strategy can win on a word that starts with $a$ and has length at least 2. A one-pass strategy $\sigma'$ that wins on all words from $W(\sigma)$ and also on $a$ can be defined by

$$\sigma'(\alpha, f) = \begin{cases} \text{Call}, & \text{if } \alpha f \in \{ a, \hat{ab}, \hat{ac} \}, \\ \text{Read}, & \text{if } \alpha f = \hat{abc}, \\ \sigma(\alpha, f), & \text{otherwise} \end{cases}$$

for $\alpha \in \Sigma^*$, $f \in \Sigma_f = \{ a, b, c \}$. Since $W(\sigma) \subseteq W(\sigma) \cup \{ a \} = W(\sigma')$, it follows that $\sigma$ is not undominated.

For the analogon of (b), consider the game $G_2 = (\Sigma, R, T)$ over the alphabet $\Sigma = \{ a, b, c \}$ with rules given by

$$a \rightarrow bb + cbc$$
$$b \rightarrow cc$$

and target language automaton $T$ as per Figure 4b. The finite target language is $L(T) = \{ bbc, bcc, cbc, ccc \}$.

We claim that the regular forgetful strategy $\sigma_A$ based on the automaton $A$ shown in Figure 4b is undominated. It can be checked easily that $W(\sigma_A) = \{ a, bb, bcc, cbc, ccc \}$. The only words that are not in $W(\sigma_A)$ even though a strategy exists that wins on them are $bc$, $bcc$ and $cb$. However, a one-pass strategy that wins on $bc$ cannot win on $bcc$; a strategy that wins on $bcc$ cannot win on $bb$; a strategy that wins on $cb$ cannot win on $ccc$. Therefore, every one-pass strategy that wins on a word on which $\sigma_A$ does not win also loses on a word on which $\sigma_A$ wins. Hence, there exists no strategy $\sigma$ with $W(\sigma_A) \subseteq W(\sigma)$, which means that $\sigma$ is undominated.

It remains to show that there exists no undominated strongly regular strategy for $G$. Let $B = (Q \cup \{ \text{Call} \}, \Sigma, \delta_B, q_0, \{ \text{Call} \})$ be the strategy automaton of a strongly regular strategy $\sigma_B$. Observe that $\sigma_B$ cannot win on $a$: In order to win on $a$, a strategy must play Call in the configurations $(\varepsilon, a, \varepsilon)$ and $(\hat{ab}, b, \varepsilon)$ and it must play Read in the configurations $(\hat{a}, b, b)$ and $(\hat{ac}, b, c)$. But a strongly regular strategy must play the same in the configurations $(\hat{ab}, b, \varepsilon)$ and $(\hat{ac}, b, c)$, because both have $T$-state $q_1$ and current symbol $b$. However, we can find a
one-pass strategy $\sigma$ with $W(\sigma_B) \subset W(\sigma) \cup \{a\} = W(\sigma)$. Indeed, such a strategy can be defined by

\[
\sigma(\alpha, f) = \begin{cases} 
\text{Call,} & \text{if } (\alpha, f) = (\varepsilon, a) \text{ or } (\alpha, f) = (\hat{\alpha}b, b), \\
\text{Read,} & \text{if } (\alpha, f) = (\hat{\alpha}a, b) \text{ or } (\alpha, f) = (\hat{\alpha}c, b), \\
\sigma_B(\alpha, f), & \text{otherwise}
\end{cases}
\]

for $\alpha \in \Sigma^*$ and $f \in \Sigma_f = \{a, b\}$. Hence, $\sigma_B$ is not undominated. \qed

4 Complexity results for regular strategies

For classes of games where (weakly) dominant or undominated strategies are not guaranteed to exist, it would be desirable to have algorithms determining, given a game $G$, whether a strategy of a certain type exists on $G$, or whether a given strategy is (weakly) dominant or undominated on $G$. Unfortunately, in general, these problems are not even known to be decidable.

In this section, we start the investigation of algorithmic problems related to one-pass strategies by studying the complexity of three somewhat more restricted types of problems. The first is to
test for a given game, input word and strategy whether the strategy wins on the input word. The
second is similar, but the strategy is not part of the input: The question in this case is whether a
strategy exists that wins on a given input word in a given game. The third problem is relevant in
the context of determining dominant or undominated strategies: Given a game and two strategies,
is one strategy better than the other? That is, we examine the following decision problems for
automata-based classes $S$ of strategies:

| Problem          | Input:                      | Question:                                          |
|------------------|-----------------------------|---------------------------------------------------|
| IsWinning        | Game $G = (\Sigma, R, T)$, word $w \in \Sigma^*$, strategy automaton $A$ for $G$ | Is $w \in W(\sigma_A)$?                           |
| ExistsWinning($S$) | Game $G = (\Sigma, R, T)$, word $w \in \Sigma^*$, strategy automaton $A$ for $G$ | Does a strategy $\sigma \in S$ for $G$ exist such that $w \in W(\sigma)$? |
| IsDominated      | Game $G$, strategy automata $A_1$ and $A_2$ for $G$ | Is $W(\sigma_{A_1}) \subseteq W(\sigma_{A_2})$? |

For the IsWinning and IsDominated problems (and the more general problem whether a
given strategy is dominant or undominated), we restrict our attention to regular strategies, since
non-regular strategies do not in general have a finite encoding (as there are uncountably many of
them). The ExistsWinning problem may be examined for several types of strategies (regular,
forgetful, etc.); however, we mostly focus on the set $S_{\text{REG}}$ of strongly regular strategies.

The main results of this section are summarised in the following theorem.

**Theorem 4.1.** (a) $\text{IsWinning} \in \text{P}$. 

(b) $\text{ExistsWinning}(S_{\text{REG}})$ is $\text{NP}$-complete. The lower bound already holds for non-recursive
games whose replacement languages are finite and prefix-free.

(c) $\text{IsDominated}$ is $\text{PSPACE}$-complete. The lower bound already holds for strongly regular
strategies and for non-recursive games whose plays all have depth at most 2, and whose
replacement words all have length 1.

The remainder of this section is structured as follows. As an important tool for the upper
bounds, we first show that relations describing Romeo’s power against a given regular strategy
can be computed in polynomial time (Proposition 4.2). Then we show that, from these relations,
an NFA for those words for which Romeo has a winning strategy or can force infinite play
can be computed, again in polynomial time (Proposition 4.6). Afterwards the upper bounds of
Theorem 4.1 are shown as Proposition 4.7. The lower bounds are then shown as Propositions 4.8
and 4.9. The section is wrapped up by some additional results from [5].

In this section, we use some additional notation. Let $A = (Q, \Sigma, \delta, s, F)$ be a strategy automaton.
For a configuration $(\alpha, u)$ arising in a play following strategy $\sigma_A$, we refer to $\delta^*(s, \alpha)$ as its
$A$-state. For each $q \in Q$, we denote by $A_q = (Q, \Sigma, \delta, q, F)$ the automaton derived from $A$ by
setting its initial state to $q$. For a word $w$ and strategies $\sigma$ and $\tau$ of Juliet and Romeo, we
write $h(\Pi(\sigma, \tau, w))$ for the final history string of $\Pi(\sigma, \tau, w)$. Whenever we do so we imply that
$\Pi(\sigma, \tau, w)$ is finite. For a Romeo strategy $\tau$ and a history string $\alpha$, we denote by $\tau^\alpha$ the strategy
given by $\tau^\alpha(\beta, a) = \tau(\alpha \beta, a)$.

**Proposition 4.2.** For a strategy automaton $A$ and a game $G = (\Sigma, R, T)$, the following two
relations can be computed in polynomial time.
We show both implications separately by induction.

The semantics of the Move "a" following conditions holds for some alphabet symbols occur as atomic subexpressions.

Let $S$ be the set of all subexpressions of the regular expressions $R_a$ for all $a \in \Sigma_f$. We will define a relation $Move \subseteq Q \times (S \cup \Sigma) \times Q$, where a triple $(q, a, q') \in Move$ with $a \in \Sigma$ shall denote that a play of $\sigma_A$ on $a$ may terminate in state $q'$ (for some strategy of Romeo). We note that the union $S \cup \Sigma$ is not disjoint since replacement words are non-empty by definition, and therefore some alphabet symbols occur as atomic subexpressions.

For $(q, r, q') \in Q \times (S \cup \Sigma) \times Q$, we define inductively that $(q, r, q') \in Move$ if any of the following conditions holds for some $a \in \Sigma, r_1, r_2 \in S$ and $q'' \in Q$:

(a) $r = a$ and $q' = \delta(q, a) \notin F$

(b) $r = a, \delta(q, a) \in F$ and $(\delta(q, a), R_a, q') \in Move$

(c) $r = \varepsilon$ and $q = q'$

(d) $r = r_1r_2$ and $(q, r_1, q''), (q'', r_2, q') \in Move$

(e) $r = r_1 + r_2$ and $(q, r_1, q') \in Move$

(f) $r = r_1 + r_2$ and $(q, r_2, q') \in Move$

(g) $r = r_1^*$ and $q' = q$

(h) $r = r_1^*$ and $(q, r_1, q''), (q'', r_1^*, q') \in Move$

The semantics of the Move relation is characterised by the following claim.

Claim 4.3. For $q, q' \in Q$ and $r \in S \cup \Sigma$, the following are equivalent:

- $(q, r, q') \in Move$
- There exist $w \in L(r)$ and a Romeo strategy $\tau$ such that $\delta^*(q, \alpha) = q'$ holds for $\alpha = h(\Pi(\sigma_A, \tau, w))$.

We show both implications separately by induction.

"$\Rightarrow$": The proof of the implication from the first to the second statement is by structural induction on elements of Move.

If $r = a \in \Sigma$ and $q' = \delta(q, a) \notin F$, then the second statement holds for $w = a$ and an arbitrary Romeo strategy $\tau$, since $\sigma_A$ plays Read on $a$. 
If \( r = a \in \Sigma \), \( \delta(q, a) \in F \) and \((\delta(q, \hat{a}), R_a, q') \in Move\), then by the induction hypothesis there exist \( w' \in L(R_a) \) and a Romeo strategy \( \tau' \) such that \( h(\Pi(\sigma_{A\hat{a}}, \tau', w')) = \beta \) with \( \delta^*(\delta(q, \hat{a}), \beta) = q' \). The second statement holds for \( w = a \) and any Romeo strategy \( \tau \) with \( \tau(e, a) = w' \) and \( \tau(\hat{a}a, b) = \tau'(\alpha, b) \) for \( \alpha \in \Sigma^* \) and \( b \in \Sigma_f \). That is, if the input word is \( a \) and Juliet’s first move is \( Call \), then \( \tau \) replaces \( a \) by \( w' \) and continues to play like \( \tau' \) plays on the input word \( w' \).

If \( r = \varepsilon \) and \( q = q' \), then the second statement holds for \( w = \varepsilon \) and any \( \tau \). If \( r = r_1 r_2 \) and \((q, r_1, q'''), (q', r_2, q') \in Move\), then by the induction hypothesis there exist words \( w_1 \in L(r_1) \), \( w_2 \in L(r_2) \) and Romeo strategies \( \tau_1 \) and \( \tau_2 \) for which it holds that \( h(\Pi(\sigma_{A_{q'''}}, \tau_1, w_1))) = \beta \) with \( \delta^*(\delta(q, \hat{a}), \beta) = q'' \) and \( h(\Pi(\sigma_{A_{q'}}, \tau_2, w_2)) = \beta' \) with \( \delta^*(\delta(q', \hat{a}) = q' \). The second statement holds for \( w = w_1 w_2 \) and the Romeo strategy that plays like \( \tau_1 \) on \( w_1 \) and like \( \tau_2 \) thereafter.

The remaining cases are similar.

“\( \Rightarrow \)”: Let \( w \in L(r) \) and a Romeo strategy \( \tau \) be such that \( \delta^*(q, \alpha) = q' \) for \( \alpha = h(\Pi(\sigma_{A_{q'}}, \tau, w)) \). Let \( x = \tau \alpha \in \Sigma^* \) be the final string of this play. For \( a \in \Sigma_f \), let \( n_a \) be the number of \( \text{Calls} \) of \( a \) occurring during this play. We prove the implication by induction on \( |x| + |r| + \sum_{a \in \Sigma_f} n_a |R_a| \).

We distinguish several cases and deduce the first statement in each case either directly or by applying the induction hypothesis.

- If \( r = \varepsilon \), then \( w = \varepsilon \) and therefore \( q' = q \). Hence, \((q, r, q') \in Move\) follows directly from the definition, since condition (a) is satisfied.
- If \( r = a \in \Sigma \) and \( \delta(q, a) \notin F \), then \( w = a \) and \( q' = \delta(q, a) \). Again \((q, r, q') \in Move\) follows directly, since condition (a) of the definition is satisfied.
- If \( r = a \in \Sigma \) and \( \delta(q, a) \in F \), then \( w = a \) and for \( \alpha' = h(\Pi(\sigma_{A_{q'}}, \tau, \tau(\alpha))) \) it holds that \( \delta^*(\delta(q, \hat{a}), \alpha') = q' \). We can apply the induction hypothesis and get \((\delta(q, \hat{a}), R_a, q') \in Move\). Thus, \((q, r, q') \in Move\) holds by condition (b) of the definition.
- If \( r = r_1^* \) then we can write \( w = w_1 w_2 \ldots w_n \) with \( n \in \mathbb{N}_0 \) and \( w_j \in L(r_1) \setminus \{\varepsilon\} \). If \( n = 0 \), then \( w = \varepsilon \) and \( q = q' \), so \((q, r, q') \in Move\) by condition (c). If \( n \geq 1 \), let \( \alpha_1 \overset{def}{=} h(\Pi(\sigma_{A_{q'}}, \tau, w_1)) \) and let \( q'' = \delta(q, \alpha_1) \). Since \( |r_1^*| = |r_1| + 1 > |r_1| \), we can apply the induction hypothesis, which yields \((q, r_1, q'') \in Move\). Observe that \( h(\Pi(\sigma_{A_{q''}}, \tau, w_2 w_3 \ldots w_n)) = \alpha_2 \) with \( \delta^*(q'', \alpha_2) = q' \), and its final string is the string \( x' \) with \( \alpha_1 x' x = x \). From \( |\alpha_1| > |w_1| > 0 \) it follows that \( |x| > |x'| \), so another application of the induction hypothesis yields \((q'', r_1^*, q') \in Move\). We get \((q, r, q') \in Move\) from condition (d) of the definition.
- The cases \( r = r_1 r_2 \) and \( r = r_1 + r_2 \) are similar to and slightly easier than the previous case.

This concludes the proof of Claim 4.3.

The basic idea for the detection of infinite plays is as follows: for a play to be infinite there must be a function symbol \( a \in \Sigma_f \) and a state \( q \in Q \) such that \( a \) is called in \( q \) and in the “sub-play” on the replacement word of \( a \) there occurs another configuration in which \( A \) has state \( q \).
and the current symbol is $a$. To detect the possibility of such a play, we introduce a relation $\text{Next} \subseteq Q \times (S \cup \Sigma) \times Q \times \Sigma$. A tuple $(q, r, q', a)$ shall be in $\text{Next}$ if and only if there exists a play of $\sigma_{A_q}$ on some word from $L(r)$ that contains a configuration with $A_q$-state $q'$ and current symbol $a$.

To this end, we define inductively, for $(q, r, q', a) \in Q \times (S \cup \Sigma) \times Q \times \Sigma$, that $(q, r, q', a) \in \text{Next}$ if any of the following conditions holds for some $b \in \Sigma, r_1, r_2 \in S$ and $q'' \in Q$:

(a) $r = a$ and $q = q'$
(b) $r = b, \delta(q, b) \in F$ and $(\delta(q, \hat{b}), R_b, q', a) \in \text{Next}$
(c) $r = r_1 r_2$ and $(q, r_1, q', a) \in \text{Next}$
(d) $r = r_1 r_2, (q, r_1, q'') \in \text{Move}$ and $(q'', r_2, q', a) \in \text{Next}$
(e) $r = r_1 + r_2$ and $(q, r_1, q', a) \in \text{Next}$
(f) $r = r_1 + r_2$ and $(q, r_2, q', a) \in \text{Next}$
(g) $r = r_1', (q, r_1', q'') \in \text{Move}$ and $(q'', r_1, q', a) \in \text{Next}$

Claim 4.4. For $q, q' \in Q, r \in S \cup \Sigma$ and $a \in \Sigma$, the following are equivalent:

- $(q, r, q', a) \in \text{Next}$
- There exist $w \in L(r)$ and a Romeo strategy $\tau$ such that $\Pi(\sigma_{A_q}, \tau, w)$ contains a configuration with $A_q$-state $q'$ and current symbol $a$.

“$\Rightarrow$” : The proof is by structural induction on elements of $\text{Next}$ and uses the same techniques that were already in the proof of the implication “$\Rightarrow$” of Claim 4.3.

“$\Leftarrow$” : Let $w \in L(r)$ and $\tau$ be a Romeo strategy such that $\Pi(\sigma_{A_q}, \tau, w)$ contains a configuration with $A_q$-state $q'$ and current symbol $a$ and $w$ is of minimal length with these properties. Let $\alpha$ be the history string of the first such configuration. For $b \in \Sigma_f$ let $n_b$ be the number of occurrences of $b$ in $\alpha$, i.e. the number of Calls of $b$ before this configuration is reached. The proof is by induction on $|r| + \sum_{b \in \Sigma} n_b |R_b|$. Note that $w \neq \varepsilon$, because the only configuration in a play on $\varepsilon$ is $(\varepsilon, \varepsilon)$, which does not have current symbol $a$. So in particular $r \neq \varepsilon$. There are several cases.

- If $r = a$ and $q = q'$, then $(q, r, q', a) \in \text{Next}$ by condition (a) of the definition.
- If $r = b \in \Sigma, b \neq a$ or $q \neq q'$, then $w = b$ and the initial configuration $(\varepsilon, b)$ does not have both $A_q$-state $q'$ and current symbol $a$. It is necessary that $\delta(q, b) \in F$ because otherwise also the second and last configuration $(b, \varepsilon)$ does not have current symbol $a$. Thus, $\alpha = \hat{b} \beta$ for some $\beta \in \Sigma^*$ and $\Pi(\sigma_{A_{\hat{b} \beta}}, \tau(\varepsilon, b))$ must have a configuration with $A_{\delta(q, \hat{b})}$-state $q'$ and current symbol $a$. By the induction hypothesis, $(\delta(q, \hat{b}), R_b, q', a) \in \text{Next}$. Hence, $(q, r, q', a) \in \text{Next}$ by condition (b) of the definition.
• If \( r = r_1^s \), then, since \( w \neq \epsilon \) by the argument above, \( w = w_1 \ldots w_n \) for some \( n \geq 1 \) and \( w_j \in L(r_1) \setminus \{ \epsilon \} \). By minimality of the length of \( w \), the play \( \Pi(\sigma_{A_q}, \tau, w_1 \ldots w_{n-1}) \) must be finite. Let \( \beta \) be its history string, and let \( q'' = \delta'(q, \beta) \). By Claim 4.3, \( (q, r_1, q'') \in \text{Move} \). Moreover, \( \Pi(\sigma_{A_q}, \tau^{\beta}, w_n) \) contains a configuration with \( A_q \)-state \( q' \) and current symbol \( a \). By the induction hypothesis, \( (q'', r_1, q', a) \in \text{Next} \). It follows from condition (g) of the definition that \( (q, r, q', a) \in \text{Next} \).

• The cases \( r = r_1 r_2 \) and \( r = r_1 + r_2 \) are similar.

This concludes the proof of Claim 4.4.

Finally, we define the relation \( \text{Inf} \in Q \times (\Sigma \cup E) \) that shall contain a pair \( (q, r) \) if and only if an infinite play of \( \sigma_{A_q} \) on some word from \( L(r) \) exists. For \( q \in Q \) and \( r \in S \cup \Sigma \), we define inductively that \( (q, r) \in \text{Inf} \) if any of the following conditions holds for some \( a \in \Sigma \), \( r_1, r_2 \in S \) and \( q \in Q \):

(a) \( r = a, \delta(q, a) \in F \) and \( (\delta(q, \hat{a}), R, q, a) \in \text{Next} \)

(b) \( r = a, \delta(q, a) \in F \) and \( (\delta(q, \hat{a}), R) \in \text{Inf} \)

(c) \( r = r_1 r_2 \) and \( (q, r_1) \in \text{Inf} \)

(d) \( r = r_1 r_2, (q, r_1, q') \in \text{Move} \) and \( (q', r_2) \in \text{Inf} \)

(e) \( r = r_1 + r_2 \) and \( (q, r_1) \in \text{Inf} \)

(f) \( r = r_1 + r_2 \) and \( (q, r_2) \in \text{Inf} \)

(g) \( r = r_1^*, (q, r_1^*, q') \in \text{Move} \) and \( (q', r_1) \in \text{Inf} \)

Claim 4.5. For \( q \in Q \) and \( r \in S \cup \Sigma \), the following are equivalent:

• \((q, r) \in \text{Inf}\)

• There exist \( w \in L(r) \) and a Romeo strategy \( \tau \) such that \( \Pi(\sigma_{A_q}, \tau, w) \) is infinite.

“\( \implies \)”: The proof is by structural induction on elements of \( \text{Inf} \).

We first consider the case that \( r = a \in \Sigma, \delta(q, a) \in F \) and \( (\delta(q, \hat{a}), R_0, q, a) \in \text{Next} \). The idea of a strategy for Romeo that achieves infinite play against \( \sigma_{A_q} \) on \( a \) is to reach a configuration with \( A_q \)-state \( q \) and current symbol \( a \) (like the initial configuration), and then to repeat the moves from the beginning so that such a configuration occurs over and over again.

Let \( w \in L(R_0) \) and a Romeo strategy \( \tau \) be such that the play \( \Pi(\sigma_{A_{a(q,a)}}, \tau, w) = (K_0, K_1, \ldots) \) contains a configuration \( K_n \) with \( A_{\delta(q, a)} \)-state \( q \) and current symbol \( a \). Such \( w \) and \( \tau \) exist by Claim 4.4. For \( i = 0, \ldots, n \), let \( \alpha_i \in \Sigma^* \) and \( v_i \in \Sigma^+ \) be such that \( K_i = (\alpha_i, v_i) \). Thus, \( \alpha_0 = \epsilon, v_0 = w, \delta'(q, \hat{a} \alpha_n) = q \) and \( v_n = av \) for some \( v \in \Sigma^+ \). Let \( \tau' \) be any Romeo strategy
with \( \tau'((\widehat{\alpha}_n)^k, a) = w = v_0 \) and \( \tau'((\widehat{\alpha}_n)^k \widehat{\alpha}_i, b) = \tau(\alpha_i, b) \) for each \( k \in \mathbb{N}_0 \), \( b \in \Sigma_f \) and \( i = 1, \ldots, n-1 \). We claim that \( \Pi(\sigma_{A_q}, \tau', a) \) is the infinite sequence

\[
((\widehat{\alpha}_n)^0, a^0), ((\widehat{\alpha}_n)^0 \widehat{\alpha}_0, v_0^0), ((\widehat{\alpha}_n)^0 \widehat{\alpha}_1, v_1^0), \ldots, ((\widehat{\alpha}_n)^0 \widehat{\alpha}_{n-1}, v_{n-1}^0),
\]

\[
((\widehat{\alpha}_n)^1, a^1), ((\widehat{\alpha}_n)^1 \widehat{\alpha}_0, v_0^1), ((\widehat{\alpha}_n)^1 \widehat{\alpha}_1, v_1^1), \ldots, ((\widehat{\alpha}_n)^1 \widehat{\alpha}_{n-1}, v_{n-1}^1),
\]

\[
((\widehat{\alpha}_n)^2, a^2), \ldots.
\]

The initial configuration is correct, since \( ((\widehat{\alpha}_n)^0, a^0) = (\varepsilon, a) \). By induction on \( k \in \mathbb{N}_0 \) it is clear that \( \delta^*(q, (\widehat{\alpha}_n)^k) = q \). Therefore,

\[
\delta^*(q, (\widehat{\alpha}_n)^k a) = \delta(\delta^*(q, (\widehat{\alpha}_n)^k), a) = \delta(q, a) \in F,
\]

so \( \sigma_{A_q} \) calls in a configuration \( ((\widehat{\alpha}_n)^k, a^k) \). The next configuration is therefore

\[
((\widehat{\alpha}_n)^k \widehat{\alpha}, \tau'((\widehat{\alpha}_n)^k, a)^k) = ((\widehat{\alpha}_n)^k \widehat{\alpha}, w^k) = ((\widehat{\alpha}_n)^k \widehat{\alpha}_0, v_0^k),
\]

which conforms to the sequence above.

Consider now a configuration \( K = ((\widehat{\alpha}_n)^k \widehat{\alpha}_i, v_i v^k) \) for \( k \in \mathbb{N}_0 \) and \( i \in \{0, \ldots, n-1\} \). We can write \( v_i = b_i v_i' \) where \( b_i \in \Sigma \) and \( v_i' \in \Sigma^* \). The strategy \( \sigma_{A_q} \) plays Call if it holds that \( \delta^*(q, (\widehat{\alpha}_n)^k \widehat{\alpha}_i b_i) \in F \), i.e. if \( \delta^*(\delta(q, \alpha), \alpha_i b_i) \in F \), which is equivalent to \( \sigma_{A_{\{q, \alpha\}}} \) playing Call in the configuration \( K_i = (\alpha, b_i v_i') \). In that case, \( \alpha_{i+1} = \alpha_i b_i \) and \( v_{i+1} = \tau(\alpha_i, b_i) v_i' = \tau'((\widehat{\alpha}_n)^k \widehat{\alpha}_i, b_i) v_i' \). The configuration after \( K \) is then \( ((\widehat{\alpha}_n)^k \widehat{\alpha}_i, v_{i+1} v^k) \), as above. Note that this also conforms to the sequence for \( i = n-1 \) since \( v_n = a v \). The other case is that \( \sigma_{A_q} \) plays Read and in \( K \) and \( \sigma_{A_{\{q, \alpha\}}} \) plays Read in \( K_i \). This case is analogous. We have shown that if condition \( [\square] \) from the definition of \( Inf \) holds, then \( \Pi(\sigma_{A_q}, \tau', a) \) is the infinite sequence from above. This finishes the base case of the structural induction.

The inductive cases are considerably easier. If \( r = a \in \Sigma \), \( \delta(q, a) \in F \) and \( (\delta(q, \alpha), R_a) \in Inf \), then by the induction hypothesis there exist \( w \in L(R_a) \) and a Romeo strategy \( \tau \) such that \( \Pi(\sigma_{A_q}, \tau, w) \) is infinite. Consider a Romeo strategy \( \tau' \) with \( \tau'(\varepsilon, a) = w \) and \( \tau'(\alpha, b) = \tau(\alpha, b) \) for \( \alpha \in \Sigma^* \) and \( b \in \Sigma_f \). The play \( \Pi(\sigma_{A_q}, \tau', a) \) is clearly infinite. In the other inductive cases, the second statement follows similarly from the induction hypothesis.

\[ \begin{align*}
\text{“} & \text{⇐} \text{”:} \quad \text{Let} \ w_1 \in L(r) \ \text{and a Romeo strategy} \ \tau_1 \ \text{be such that} \ \Pi(\sigma_{A_q}, \tau_1, w_1) \ \text{is infinite. The idea is} \\
\text{to make use of the fact that there must exist a pair} \ (q_m, a_m) \in Q \times \Sigma \ \text{such that} \ \Pi(\sigma_{A_q}, \tau_1, w_1) \ \text{contains infinitely many configurations with} \ A_q \text{-state} \ q_m \ \text{and current symbol} \ a_m. \\
\text{Let} \ u_1 \ \text{be the longest prefix of} \ w_1 \ \text{such that} \ \Pi(\sigma_{A_q}, \tau_1, u_1) \ \text{is finite. Define} \ \alpha_1 \in \Sigma^* \ \text{by} \\
\alpha_1 \overset{a_1}{=} h(\Pi(\sigma_{A_q}, \tau_1, u_1)) \ \text{and let} \ a_1 \in \Sigma \ \text{and} \ v_1 \in \Sigma^* \ \text{be such that} \ w_1 = u_1 a_1 v_1. \ \text{This means} \ \\
\text{that} \ \Pi(\sigma_{A_q}, \tau_1, w_1) \ \text{contains the configuration} \ (\alpha_1, a_1 v_1) \ \text{with} \ A_q \text{-state} \ q_1 = \delta^*(q, a_1) \ \text{and, by maximality of} \ u_1, \ \text{the play} \ \Pi(\sigma_{A_q}, \tau_1, a_1 v_1) \ \text{is infinite. Thus} \ \delta(q_1, a_1) \in F \ \text{and,} \\
\text{letting} \ \tau_2 = \tau_{a_1} \ \text{and} \ w_2 = \tau_1(\alpha_1, a_1), \ \text{the play} \ \Pi(\sigma_{A_q}, \tau_2, w_2) \ \text{must also be infinite.} \\
\text{By repeating the argument we obtain infinite sequences} \ (w_i), (u_i), (v_i) \ \subseteq \Sigma^*, \ (a_i) \ \subseteq \Sigma, \ (\alpha_i) \ \subseteq \Sigma^*, \ (q_i) \ \subseteq Q \ \text{and} \ (\tau_i) \ \text{of Romeo strategies such that} \\
\quad \bullet \ w_{i+1} = \tau_i(\alpha_i, a_i) = u_{i+1} a_i v_{i+1},
\end{align*} \]
Move when no new tuple was added during its last iteration. Since $Q$.

This completes the proof of Claim 4.5.

Clearly (a) follows from (b) and therefore it suffices to show (b).

Proof. (b) An NFA $N$ recognising $\Sigma^* \setminus W(\sigma_A)$ can be constructed from $G$ and $A$ in polynomial time.

Proof. Clearly (a) follows from (b) and therefore it suffices to show (b).
The idea behind the construction of the NFA $N$ recognising $\Sigma^* \setminus W(\sigma_A)$ is basically that the nondeterminism in $\mathcal{N}$ will be used to guess a counter-strategy (consisting of choosing replacement words with associated sub-strategies) to the fixed strategy $\sigma_A$. The groundwork for this was already laid in the form of the Move relation from the proof of Proposition 4.2. However, the target language $L(T)$ and the relation Inf from the proof of Proposition 4.2 need to be taken into account, which requires some additional constructions.

First, we consider the product automaton $A'$ of $A$ with $T$. Let $A = (Q_A, \Sigma, \delta_A, s_A, F_A)$ and $T = (Q_T, \Sigma, \delta_T, s_T, F_T)$. We define $A' = (Q_A', \Sigma, \delta_{A'}, s_{A'}, F_{A'})$, where $Q_{A'} = Q_A \times Q_T$ and $s_{A'} = (s_A, s_T)$ as per the standard product construction. Since $A'$ will take the place of a strategy automaton in the construction from the proof of Proposition 4.2, we set its accepting states to $F_{A'} = F_A \times Q_T$ and define its transition function $\delta_{A'}$ as follows for symbols $a \in \Sigma, \hat{a} \in \hat{\Sigma}$:

$$\delta_{A'}((p, q), a) = (\delta_A(p, a), \delta_T(q, a)),$$

$$\delta_{A'}((p, q), \hat{a}) = (\delta_A(p, \hat{a}), q).$$

It is easy to verify that $A'$ defines a strategy with the same winning set as $\sigma_A$, and that, on an input $\alpha \in \Sigma$, its second component simulates the DFA $T$ on $\alpha$. Clearly, $A'$ can also be computed from $G$ and $A$ in polynomial time. Since $A'$ is a strategy automaton of polynomial size, it is also possible to compute its Move and Inf relations from the proof of Proposition 4.2 in polynomial time.

We now define the NFA $\mathcal{N}$ for $\Sigma^* \setminus W(\sigma_A)$ by $\mathcal{N} = (Q_A \cup \{q_\infty\}, \Sigma, \delta_{\mathcal{N}}, s_{\mathcal{N}}, F_{\mathcal{N}})$, where $F_{\mathcal{N}} = (Q_A \times (Q_T \setminus F_T)) \cup \{q_\infty\}$ and the transition relation $\delta_{\mathcal{N}}$ is defined as the union of the sets $\delta_{\mathcal{N}} = \{(p, q), a, (p', q') \in \text{Move} \mid a \in \Sigma\}, \{(p, q), a, q_\infty \in \text{Inf} \mid a \in \Sigma\}$ and $\{(p_\infty, a, q_\infty) \mid a \in \Sigma\}$. We note that, whenever the $A$-component reaches a situation in which Romeo can enforce an infinite play, $\mathcal{N}$ can enter the accepting state $q_\infty$ and remain there. Clearly, $\mathcal{N}$ can be constructed in polynomial time. It thus only remains to show that $L(\mathcal{N}) = \Sigma^* \setminus W(\sigma_A)$. We prove both inclusions separately.

\textbf{“$\subseteq$”}: Suppose $w = a_1 \ldots a_n \in L(\mathcal{N})$ with $a_i \in \Sigma$. Let us assume first that there is an accepting run of $\mathcal{N}$ on $w$ which ends in a state of $Q_A \times (Q_T \setminus F_T)$. Then there exists a sequence $(p_0, q_0), \ldots, (p_n, q_n) \in Q_{A'}$ such that $(p_0, q_0) = (s_A, s_T), q_n \in Q_T \setminus F_T$, and $((p_{i-1}, q_{i-1}), a_i, (p_i, q_i)) \in \delta_{\mathcal{N}}$ for each $i = 1, \ldots, n$. By construction, it holds for each $i = 1, \ldots, n$ that $((p_{i-1}, q_{i-1}), a_i, (p_i, q_i)) \in \text{Move}$, and by Claim 4.3 as well as the above considerations, there exist Romeo strategies $\tau_1, \ldots, \tau_n$ such that $\Pi(\sigma_{A_{p_{i-1}}}, \tau_i, a_i)$ terminates with a final history string $\alpha_i$ such that $\delta_{\mathcal{N}}(p_{i-1}, \alpha_i) = p_i$ and $\delta_{\mathcal{N}}(q_{i-1}, \tau_\alpha_i) = q_i$ for each $i = 1, \ldots, n$. Then the play of $\sigma_A$ against the Romeo strategy that plays like $\tau_i$ on the $i$th symbol of the input word $w$ yields a final history string $\alpha = \alpha_1 \cdots \alpha_n$ with $\delta_{\mathcal{N}}(s_T, \tau_\alpha) = q_n \in Q_T \setminus F_T$.

In the other case there is an accepting run that ends in $q_\infty$. Let $(p, q)$ be the last state of $\mathcal{N}$ in this run before $q_\infty$ is entered by reading a symbol $a$. By construction of $\mathcal{N}$, $(p, a) \in \text{Inf}$. Similarly as in the first case there exists a strategy of Romeo leading into a configuration in which $A$ has state $p$ with $a$ as the next symbol. Romeo can enforce an infinite play from here and therefore $w \notin W(\sigma_A)$.
Suppose \( w \in \Sigma^* \setminus W(\sigma_A) \) for some \( w = a_1 \ldots a_n \) with \( a_i \in \Sigma \). Then there exists a Romeo strategy \( \tau \) such that \( \Pi(\sigma_A, \tau, w) \) is infinite or losing for Juliet. We first consider the case that it is losing for Juliet. For \( i = 1, \ldots, n \), let \( (q_i, a_i, a_{i+1} \ldots a_n) \) be the configuration at the time when \( a_1 \ldots a_i \) is completely processed, let \( p_i = \delta^+_A(s_T, \tau \alpha_i) \) its \( A \)-state and \( q_i = \delta^+_T(s_T, \tau \alpha_i) \) its \( T \)-state, and let \( \alpha_0 = \epsilon \) (the history string of the initial configuration). Then, for each \( i = 0, \ldots, n-1 \), \( h(\Pi(\sigma_A, \tau, a_i, a_{i+1})) = \beta_i \) such that \( \alpha_{i+1} = \alpha_i \beta_i \), and therefore \( \delta_A(p_i, \beta_i) = p_{i+1} \) and \( \delta_T(q_i, \tau \beta_i) = q_{i+1} \). Thus, by Claim 4.3 and the above considerations, \( (p_i, q_i, a_{i+1}, p_{i+1}, q_{i+1}) \in \text{Move} \) for each \( i = 0, \ldots, n-1 \), so there exists a run of \( N \) on \( w \) that terminates in \( (p_n, q_n) \). Since \( \Pi(\sigma_A, \tau, w) \) is losing for Juliet, it holds that \( q_n = \delta^+_T(s_T, \tau \alpha_n) \in Q_T \setminus F \), so \( w \in L(N) \).

In the other case, when \( \Pi(\sigma_A, \tau, w) \) is infinite, let \( m \) be maximal such that the play on \( a_1 \ldots a_m \) is finite. Let the states \( p_i \) and \( q_i \) be defined as before, for \( i = 1, \ldots, m \). Since the play on the next symbol is infinite, \( (p_m, a_{m+1}) \in \text{Inf} \), and therefore \( N \) has a run that reaches \( (p_m, q_m) \) and then enters \( q_m \) and remains there, so \( w \in L(N) \). \[ \square \]

Now we are ready to prove the upper bounds of Theorem 4.1.

**Proposition 4.7.**

(a) \( \text{ISWinning} \in \mathbb{P} \).

(b) \( \text{ExistsWinning(SReg)} \in \mathbb{NP} \).

(c) \( \text{ISDominated} \in \text{PSpace} \).

**Proof.**

(a) By Proposition 4.6, an NFA \( N \) recognizing \( \Sigma^* \setminus W(\sigma_A) \) can be constructed in polynomial time from \( A \) and \( G \). Then, the set of states of \( N \) that can be reached at the end of a run on \( w \) can be computed in polynomial time by simulating the power set automaton. We have \( w \in W(\sigma_A) \) if and only if none of these states is accepting.

(b) \( \text{ExistsWinning(SReg)} \) can be solved non-deterministically in polynomial time as follows. Let \( G = (\Sigma, R, T) \) and \( w \in \Sigma^* \) be the input. First, guess a strongly regular strategy automaton \( A \) by adding a state \( \text{Call} \) to \( T \) and nondeterministically guessing which transitions of \( T \) to reroute to \( \text{Call} \). Second, call the algorithm for ISWinning from (a) with input \( A, G \) and \( w \).

(c) Let \( G \) be a game and let \( A_1 \) and \( A_2 \) be strategy automata for \( G \). By Proposition 4.6, NFAs \( B_1 \) and \( B_2 \) recognizing \( \Sigma^* \setminus W(\sigma_{A_1}) \) and \( \Sigma^* \setminus W(\sigma_{A_2}) \), respectively, can be constructed in polynomial time. Clearly, \( W(\sigma_{A_1}) \subseteq W(\sigma_{A_2}) \) if and only if \( L(B_2) \subseteq L(B_1) \). Therefore, the relative domination can be decided with the help of the polynomial space algorithm that tests containment for NFAs \[ \square \]

We now turn to the corresponding lower bound results for Theorem 4.1.

**Proposition 4.8.** The problem \( \text{ExistsWinning(SReg)} \) is \( \mathbb{NP} \)-hard, even for non-recursive games whose replacement languages are finite and prefix-free.

**Proof.** We give a reduction from the \( \mathbb{NP} \)-complete problem 3SAT of determining whether a given Boolean formula in CNF with three literals per clause is satisfiable. To this end, let \( \varphi \) be an
arbitrary formula in 3CNF over variables $x_1, \ldots, x_n$ with clauses $K_1, \ldots, K_m$. We describe how to construct from $\varphi$ a game $G = (\Sigma, R, T)$ and string $w \in \Sigma^*$ in polynomial time, such that Juliet has a strongly regular winning strategy in $G$ on $w$ if and only if $\varphi$ is satisfiable.

The basic idea behind this reduction is to encode variable assignments into strategy automata of a strongly regular strategy. To this end, the input string $w$ is of the form $w_1w_2$, where the sub-game on $w_1$ is meant to choose a variable assignment and the sub-game on $w_2$ to verify that it is satisfying. We consider only certain valid strategies $\sigma$ for Juliet, each of which will induce a variable assignment $\theta_\sigma$ for $\varphi$.

The string $w_1$ is chosen as $w_1 = 0CDa_20CDa_30CD \cdots a_n0CD$.

The DFA $T = (Q, \Sigma, \delta, s_1, F)$ consists of sub-automata $T_1, \ldots, T_n$, where for each $i \leq n$, $T_i$ has states and transitions as specified by Figure 5. Furthermore, $Q$ contains a non-accepting sink state $q_e$. The set of accepting states is $F = \{f_1, \ldots, f_n\}$, and the initial state is $s = s_1$. The construction of $T$ is completed by adding the following transitions:

- $(t_{i-1}, a_i, s_i)$ for $i = 2, \ldots, n$ (connecting $T_{i-1}$ and $T_i$),
- $(q, a_i, s_i)$ for each $q \in Q \setminus (F \cup \{q_e, t_1, t_2, \ldots, t_{n-1}\})$ and $i = 1, \ldots, n$, and
- transitions leading to the error state $q_e$ for all remaining combinations of states and symbols.

Roughly speaking, every sub-string $0CD$ of $w_1$ induces a sub-game whose strategy is determined by one sub-automaton $T_i$ and the symbols $a_i$ are used to get from $T_{i-1}$ to $T_i$.

Before we further describe the construction of $G$, we first show that $T$ is a minimal automaton. This is because all states are reachable from the initial state $s_1$ and the states are pairwise non-equivalent:

- The sink state is not equivalent to any other state because it is the only state from which no accepting state is reachable.
- For any two states $p$ and $q$ with distinct roles within their components ($p$ and $q$ may belong to the same component $T_i$ or to two distinct components $T_j$ and $T_k$ with $j \neq k$) it is easy to find some symbol $e \in \Sigma$ such that $\delta(p, e) = q_e$ and $\delta(q, e) \neq q_e$ or vice versa.
• Two states $q_j$ and $q_k$ with the same role in their respective components $T_j$ and $T_k$ with $j < k$ are not equivalent: Let $u \in \{1, c, C, d, D, a_{k+1}, \ldots, a_n\}^*$ be of minimal length such that $\delta^*(q_k, u) = t_n$. Then $\delta^*(q_k, ua_1) = s_1$ and $\delta^*(q_j, ua_1) = q_e$.

Thus, the strongly regular strategy automata for $G$ are exactly the automata that can be obtained from $T$ by adding a state $Call$ as the only accepting state, removing transitions with function symbols from $T$ and adding new transitions with function symbols to $Call$ instead.

A strategy automaton for Juliet derived from $T$ is called valid if, for every $i$, $T_i$ has exactly one of the two transitions $(s_i, 0, b_i)$ and $(s_i, 1, d_i)$. A strategy is called valid if it is defined by a valid strategy automaton. With a valid strategy $\sigma$ we associate a variable assignment $\theta_\sigma$ as follows: $\theta_\sigma(x_i) = 0$ if and only if the transition $(s_i, 0, b_i)$ is missing (and $\theta_\sigma(x_i) = 1$ if the transition $(s_i, 1, d_i)$ is missing).

For the sub-game on $w_1$, the set $R$ has the following replacement rules.

$$
\begin{align*}
0 & \rightarrow b \\
1 & \rightarrow b \\
C & \rightarrow c1 \\
D & \rightarrow d1d
\end{align*}
$$

The symbols $b, c, d$ and $a_1, \ldots, a_n$ are not function symbols.

These rules ensure that any strongly regular strategy $\sigma$ with strategy automaton $A$ for Juliet rewrites $w_1 = 0CDa_20CDa_30CD \cdots a_n0CD$ into a string $w'_1$ with $\delta^*(s_1, w'_1) \neq q_e$ only if it is valid. It should be noted that this statement is independent of any strategy of Romeo, since the above four rules do not leave Romeo any choice.

The statement follows by an easy induction on prefixes of $w_1$ of the form $0CDa_20 \cdots a_n0CD$.

The crucial observation is that a sub-play on a subword $0CD$ starting from a state $s_i$ reaches the sink state $q_e$ if either both or none of $(s_i, 0, b_i)$ and $(s_i, 1, d_i)$ are present. And there is no way out of $q_e$ afterwards.

Indeed, if both transitions are present, the sub-play proceeds as follows: it reads 0. If the transition $(b_i, C, c_i)$ is missing, $C$ is replaced by $c1$ yielding the sink state after reading $c$. Otherwise, $C$ is read, and the subsequent $D$ either leads to $q_e$ or is necessarily replaced by $d1d$. In the latter case, $d$ is then read, bringing the play back to $s_i$. By assumption, 1 is read as well and now $d$ yields a transition into $q_e$.

On the other hand, if both transitions are missing, the sub-play proceeds as follows: it calls 0, getting $b$ which is read. It then calls $C$, getting $c1$. Reading $c$ leads back into $s_1$, where the subsequent 1 triggers another call and the resulting $b$ leads back to $f_i$. The call of $D$ yields $d1d$, reading $d$ leads into $t_i$. Either $q_e$ is now immediately reached by reading 1 or after calling it and reading $b$.

Altogether, we have established that any strongly regular winning strategy for Juliet on $w$ must be valid. We are ready to complete the construction of $R$ and finish the proof.

To this end, for each $j \leq m$, let $K_j = (\ell_{j, 1} \lor \ell_{j, 2} \lor \ell_{j, 3})$ be a clause of $\phi$ with literals $\ell_{j, 1}, \ell_{j, 2}, \ell_{j, 3}$. By $w(K_j)$ we denote the string $w(\ell_{j, 1})w(\ell_{j, 2})w(\ell_{j, 3})$, where for an arbitrary literal $\ell$, we define
\[ w(\ell) = \begin{cases} a_i0, & \text{if } \ell = \neg x_i \text{ for some } i \\ a_i1, & \text{if } \ell = x_i \text{ for some } i. \end{cases} \]

Finally, \( w_2 = E \), where \( E \) is a function symbol with replacement rules

\[ E \to w(K_1) | w(K_2) | \ldots | w(K_m). \]

It is not hard to see that this construction is indeed feasible in polynomial time and that \( G \) is non-recursive with finite prefix-free replacement languages. It remains to be proven that Juliet has a strongly regular winning strategy on \( w \) in \( G \) if and only if \( \varphi \) is satisfiable.

For the “if” direction, let \( \theta \) be a satisfying assignment for \( \varphi \) and consider the strongly regular strategy \( \sigma_\theta \) with strategy automaton \( A_\theta \) derived from \( T \) by adding the accepting Call state and rerouting to it from each component \( T_i \) of \( T \) all transitions leading to \( q_e \) and, furthermore, the transition \((s_i,0,b_i)\) if \( \theta(x_i) = 0 \) and the transition \((s_i,1,d_i)\), otherwise. We claim that \( \sigma_\theta \) is a winning strategy for Juliet.

It is easy to see, that every sub-play on 0CD starting in a state \( s_i \) according to \( \sigma_\theta \) leads to state \( t_i \). Either through reading 0C, calling \( D \), reading \( d \), calling 1, reading \( b \) and \( d \). Or through calling 0, reading \( b \), calling \( C \) and reading \( c1D \).

Thus, with the help of the additional \( a_i \)-symbols, the sub-play on \( w_1 \), which starts in \( s_1 \), ends necessarily in \( t_n \). By construction of \( \sigma_\theta \), Juliet then calls the final \( E \) of \( w \). Let \( j \) be such that \( w(K_j) \) is Romeo’s answer to Juliet’s Call on \( E \). Since \( \theta \) satisfies \( \varphi \), at least one literal of \( K_j \) is set to “true” under \( \theta \), which by the construction of \( A_\theta \) and the definition of \( w(K_j) \) means that \( w(K_j) \) contains at least one substring of the form \( a_i0 \) (resp. \( a_i1 \)) for which the 0-transition (resp. 1-transition) is missing from \( T_i \) in \( A_\theta \). It is easy to see that, after reading \( a_i \), \( T \) is in state \( s_i \), so the subsequent 0 (resp. 1) is called by Juliet and thus rewritten into \( b \), inducing a transition into \( f_i \). Since \( f_i \) has no outgoing 0- or 1-transitions in \( A_\theta \) and the remainder of \( w(K_j) \) contains only symbols from \( \{0,1,a_1,\ldots,a_n\} \), it is clear that all remaining symbols 0 or 1 get rewritten to \( b \), and \( T \) never leaves \( f_i \) anymore. This implies that \( \sigma_\theta \) is winning on \( w \).

For the “only if” direction, assume that \( \sigma = \sigma_A \) is a strongly regular winning strategy for Juliet on \( w \) induced by a strategy automaton \( A_\sigma \). As shown above, \( \sigma \) must be valid and thus induces an assignment \( \theta_\sigma \) for the variables of \( \varphi \). We need to show that \( \theta_\sigma \) satisfies at least one literal in each clause of \( \varphi \), and is therefore a satisfying assignment for \( \varphi \).

We first prove that the assumption that \( \sigma \) is winning implies that the sub-play on \( w_1 \) must end in state \( t_n \). To show this, we consider again a sub-play on a string 0CD starting from a state \( s_i \). We need to consider two cases.

In the first case, where \((s_i,0,b_i)\) is present, \((b_{ij}C,c_i)\) must also be present since otherwise, reading 0 and calling \( C \) would bring the next symbol \( c \), which can not be called and brings \( T \) from \( b_{ij} \) to \( q_e \), contradicting our winning assumption. Thus, \( C \) is read, \( D \) is called, \( d \) is read, 1 is called and finally \( bd \) is read, reaching \( t_i \).

In the other case, where \((s_i,0,b_i)\) is absent, 0 is called, \( b \) is read, \( C \) is called, and \( c1 \) is read. If the transition \((d_{ij},D,t_i)\) were absent, the subsequent call of \( D \) would yield \( d1d \) and reading \( d \) would bring \( T \) into state \( q_e \), again a contradiction. Thus, \( D \) is read and the sub-play ends in \( t_i \), as desired.
By induction and, taking the intermediate non-function symbols $a_i$ into account, the play on $w_1$ thus indeed ends in $t_n$.

In the remaining play on $w_2 = E$, $\sigma$ has to call $E$, since $\delta(t_n, E) = q_\varphi$. Each counter-strategy of Romeo consists of answering this call by some $w(K_j)$. We show that, if Juliet wins according to $\sigma$ against a counter-strategy playing $w(K_j)$, then the assignment $\theta_\sigma$ satisfies clause $K_j$; since $\sigma$ is a winning strategy, this directly implies that $\theta_\sigma$ is a satisfying assignment for $\varphi$.

To this end, assume for the sake of contradiction that $\theta_\sigma$ does not satisfy $K_j$. This means that $\theta_\sigma(l_{j,k}) = 0$, for every $k \in \{1, 2, 3\}$. Thus, for each substring of the form $a_i0$ (resp. $a_i1$) of $w(K_j)$, the 0-transition (resp. 1-transition) of $T_i$ is present in $A_\sigma$. It is easy to see that, starting from $t_n$, the sub-play on $w(K_j)$ consists of six reading steps, and eventually ends in a non-accepting state of the form $s_k$. Therefore $\sigma$ does not win on $w$ against the counter-strategy of Romeo replacing $E$ with $w(K_j)$. This contradicts the assumption that $\sigma$ is winning on $w$ and completes the proof. □

**Proposition 4.9.** The problem ISDOMINATED is PSPACE-hard, even for strongly regular strategies and for non-recursive games whose plays all have depth at most 2, and whose replacement words all have length 1.

**Proof.** The proof is by a reduction from the PSPACE-complete universality problem for NFAs [7,6]: Given an NFA $A$ over the alphabet $\{0,1\}$, does $L(A) = \{0,1\}^*$?

The idea is to construct $G$ such that there is a strongly regular strategy $\sigma_1$ that loses on a string from $\{0,1\}^*$ if and only if it is accepted by $A$ and a second one, $\sigma_2$, that loses exactly on all words from $\{0,1\}^*$.

Then $W(\sigma_1) \subseteq W(\sigma_2)$ if and only if $L(A) = \{0,1\}^*$.

Let thus $A = (Q_A, \Sigma, q_A, \delta_A, F_A)$ be an NFA. We construct a game $G = (\Sigma, R, T)$ and strongly regular strategy automata $A_1$, $A_2$ for $G$ such that $L(A) = \{0,1\}^*$ holds if and only if $W(\sigma_{A_1}) \subseteq W(\sigma_{A_2})$.

Without loss of generality, we assume that $s \in F_A$. Otherwise $A$ is not universal due to $\varepsilon \notin L(A)$ and $A$ can be mapped to a fixed negative instance of ISDOMINATED. We also assume without loss of generality that all states of $A$ are reachable from the initial state $s$ (otherwise, unreachable states can be removed).

We construct $G = (\Sigma, R, T)$ with $\{0,1\} \subseteq \Sigma$ and automata $A_1$ and $A_2$ such that

(a) $L(A) = \{0,1\}^* \setminus W(\sigma_{A_1})$, i.e. $A$ recognizes the language of words in $\{0,1\}^*$ on which $\sigma_{A_1}$ does not win, and

(b) $W(\sigma_{A_2}) = \Sigma^* \setminus \{0,1\}^*$.

Then it holds that $L(A) = \{0,1\}^*$ if and only if $W(\sigma_{A_1}) \cap \{0,1\}^* = \emptyset$, which in turn holds if and only if $W(\sigma_{A_1}) \subseteq W(\sigma_{A_2})$, since the latter equals $\Sigma^* \setminus \{0,1\}^*$. Thus it suffices to show that $G$, $A_1$ and $A_2$ with (a) and (b) can be constructed in polynomial time. An example of the subsequently described construction is shown in Figure 5.

We define the alphabet of $G$ as $\Sigma = \{0,1,#\} \cup \{(a,p) \mid a \in \{0,1\}, p \in Q_A\} \cup \{($,$p) \mid p \in Q_A \setminus \{s\}\}$.
The set $R$ of rules of $G$ is given by
\[
    a \to \bigvee_{p \in Q_N} (a, p) \quad \text{for } a \in \{0, 1\}
\]
\[
    (a, p) \to \# \quad \text{for } a \in \{0, 1\} \text{ and } p \in Q_N
\]
\[
    (\$$, p) \to \# \quad \text{for } p \in Q_N \setminus \{s\}.
\]

We note that $G$ is non-recursive, all possible plays have depth at most 2 and all replacement words are of length 1.

The target language automaton of $G$ is $T = (Q, \Sigma, \delta, F)$, where $Q = Q_N \cup \{f\}$ for some new state $f \notin Q_N$, the set $F = \{f\} \cup Q_N \setminus F_N$ of accepting states consists of $f$ and the non-accepting states of $\mathcal{N}$, and the transition function $\delta$ is defined as follows, for $a \in \{0, 1\}$ and $p, q \in Q_N$:

- $\delta(p, a) = p$,
- $\delta(p, (a, q)) = \begin{cases} q & \text{if } (p, a, q) \in \delta_N, \\ f & \text{otherwise}, \end{cases}$
- $\delta(p, (\$$, p)) = s$ if $p \neq s$,
- all remaining transitions lead to the state $f$.

The automaton $T$ is in fact the minimal DFA for $L(T)$ because its states are reachable (by assumption on $\mathcal{N}$) and pairwise non-equivalent: The two states $s$ and $f$ are non-equivalent because $f$ is accepting and $s$ is non-accepting (since $s$ is accepting in $\mathcal{N}$ by assumption). Each remaining state $q \in Q_N \setminus \{s\}$ is not equivalent to any other state of $T$ because it is the only state from which reading $(\$$, q)$ leads to a non-accepting state (namely $s$). This is actually the only purpose of the symbols of the form $(\$$, q)$.

Since $T$ is minimal, strongly regular strategy automata $A_1$ and $A_2$ can be derived from $T$ by adding states $Call_1$ for $A_1$ and $Call_2$ for $A_2$, deleting some transitions and replacing them by transitions to the respective $Call$ state.

To achieve property [a], the idea is that each run of $\mathcal{N}$ on an input word $w \in \{0, 1\}^*$ should correspond to a play, in which Romeo chooses the replacement words according to the choices in the run. Therefore, Romeo shall have a winning strategy on $w$ if and only if there exists an accepting run of $\mathcal{N}$ on $w$.

To this end, we define $A_1$ as the automaton obtained from $T$ by keeping only the $(a, p)$-edges (for $a \in \{0, 1\}$ and $p \in Q_N$) and rerouting all others to state $Call_1$.

Claim: $A_1$ satisfies [a].

It is easy to show by induction on the length of an input word $w \in \{0, 1\}^*$ that $\mathcal{N}$ has a run ending in some state $q \in Q_N$ if and only if Romeo has a strategy for which the play against $\sigma_{A_1}$ yields a final word $v$ with $\delta^*(s, v) = q$. Therefore, Romeo has a winning strategy against $\sigma_{A_1}$ precisely on all words in $L(\mathcal{N})$, since each run ending in an accepting state $q$ of $\mathcal{N}$ corresponds to a play that yields the non-accepting state $q$ of $T$ and vice versa. It is important to note here, that plays in which Romeo chooses a rewriting to a symbol $(a, p)$, for which no corresponding transition in $\mathcal{N}$ exists, end in the accepting state $f$ of $T$ and are therefore losing for Romeo. This completes the proof of the claim.
Figure 6: Example of an NFA, and the rules, target language DFA, and strategy automata constructed from it, as in the proof of Proposition 4.9.
The automaton \( A_2 \) is obtained from \( T \) by keeping only the transitions of the form \( \delta(s, a) \) with \( a \in \{0, 1, \#\} \) and of the form \( \delta(f, a) \) for \( a \in \Sigma \), and rerouting all other transitions to state \( Call_2 \).

Claim: \( A_2 \) satisfies (b):

Let \( w \in \Sigma^* \). If \( w \in \{0, 1\}^* \), then the strategy \( \sigma_{A_2} \) plays only Read moves and the \( A_2 \)-state is \( s \) all the time, yielding a play that is won by Romeo since, by our assumption, \( s \notin F \). This shows that \( W(\sigma_{A_2}) \subseteq \Sigma^* \{0, 1\}^* \).

On the other hand, if \( w \in \Sigma^* \{0, 1\}^* \), let \( x \in \Sigma \{0, 1\} \) be the first symbol in \( w \) that is not in \( \{0, 1\} \). If \( x = \# \), the strategy \( \sigma_{A_2} \) reads \# and the new \( T \)-state is \( f \). If \( x \neq \# \) then \( \sigma_{A_2} \) calls \( x \), it gets replaced by \#, and again \( \sigma_{A_2} \) reads \# and reaches \( f \). The state \( f \) cannot be left any more, and since it is accepting, Juliet wins. Thus, \( w \in W(\sigma_{A_2}) \), which completes the proof of the claim.

The proof of Proposition 4.9 is now completed by the observation that \( G = (\Sigma, R, T) \), \( A_1 \) and \( A_2 \) can clearly be constructed in polynomial time.

Theorem 4.1 assumes (as per our standard definition) that target language automata are DFAs. In [5], also the case of NFAs was studied. The following complexities could be obtained.

- \textsc{IsWinning} is \textsc{PSPACE}-complete. [5, Theorem 6.2]
- \textsc{ExistsWinning(SReg)} is \textsc{PSPACE}-hard and in \textsc{NEXPTIME}. [5, Theorem 6.7]
- \textsc{IsDominated} is in \textsc{EXPSPACE}. [5, Corollary 6.10]

The problem \textsc{ExistsWinning} can also be studied for more general classes of strategies than strongly regular ones. We do not know in general whether this problem is decidable, even for regular (but not strongly regular) strategies. However, [5, Theorem 6.8] shows that it is decidable for several restricted games, including

- prefix-free games,
- non-recursive games with finite replacement sets,
- non-recursive games with finite target languages.

5 Conclusion and open questions

Whereas we have no positive results for general games, the results for restricted games are somewhat encouraging. Indeed, prefix-free replacement languages seem to be a realistic solution in practice, because they can be achieved easily and with almost no overhead by suffixing replacement words with an end-of-file symbol. Another setting in which many positive properties hold is if the target language is finite. While finiteness of the target language may seem like a very strong restriction, one particular instance of it is if there is a single target document that has to be reached [10]. Restrictions that bound the number of recursive replacements also seem plausible in practise.

We have seen that in several cases it is possible to use a one-pass strategy that is based on an automaton and there is hope that this is possible in general (for undominated strategies). However,
we can rule out that one can restrict oneself to strongly regular strategies without losing strategic power. We have also seen that it is not generally possible to find good strategies efficiently.

Many questions remain open for further study. First of all, it remains unclear whether undominated strategies always exist. It is actually conceivable that each game has the bounded-depth property and therefore also has undominated strategies. It is also not clear whether in the bounded-depth property the bounding sequence can always be replaced by some constant. Another question is whether every game with an undominated strategy also has a regular one. This is open for general games and for games with a forgetful undominated strategy. Some open problems regarding complexity were mentioned at the end of Section 4.

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. Cobéna, 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] H. Björklund, 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.

[5] C. Coester. One-pass strategies for context-free games. Master’s thesis, TU Dortmund, 2015. [doi:10.17877/DE290R-16466]

[6] H. B. Hunt, D. J. Rosenkrantz, and T. G. Szymanski. On the equivalence, containment, and covering problems for the regular and context-free languages. *Journal of Computer and System Sciences*, 12(2):222–268, 1976.

[7] A. R. Meyer and L. J. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *Proceedings of the 13th Annual Symposium on Switching and Automata Theory*, SWAT ’72, pages 125–129. IEEE Computer Society, 1972.

[8] 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.

[9] 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.

[10] A. Muscholl, T. Schwentick, and L. Segoufin. Active context-free games. *Theory of Computing Systems*, 39(1):237–276, 2006.
[11] M. Schuster and T. Schwentick. Games for Active XML revisited. In 18th International Conference on Database Theory, ICDT 2015, pages 60–75, 2015.

[12] Martin 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. URL: https://doi.org/10.4230/LIPIcs.MFCS.2016.83, doi:10.4230/LIPIcs.MFCS.2016.83