Optimally Resilient Strategies in Pushdown Safety Games

Daniel Neider
Max Planck Institute for Software Systems (MPI-SWS), Kaiserslautern, Germany
neider@mpi-sws.org

Patrick Totzke
University of Liverpool, UK
totzke@liverpool.ac.uk

Martin Zimmermann
University of Liverpool, UK
martin.zimmermann@liverpool.ac.uk

Abstract

Infinite-duration games with disturbances extend the classical framework of infinite-duration games, which captures the reactive synthesis problem, with a discrete measure of resilience against non-antagonistic external influence. This concerns events where the observed system behavior differs from the intended one prescribed by the controller. For games played on finite arenas it is known that computing optimally resilient strategies only incurs a polynomial overhead over solving classical games.

This paper studies safety games with disturbances played on infinite arenas induced by pushdown systems. We show how to compute optimally resilient strategies in triply-exponential time. For the subclass of safety games played on one-counter configuration graphs, we show that determining the degree of resilience of the initial configuration is PSPACE-complete and that optimally resilient strategies can be computed in doubly-exponential time.

2012 ACM Subject Classification Theory of computation → Automata over infinite objects

Keywords and phrases Controller Synthesis, Infinite Games, Resilient Strategies, Pushdown Games

Digital Object Identifier 10.4230/LIPIcs.MFCS.2020.74

Related Version Full version available at `https://arxiv.org/abs/1912.04771`.

Funding *Martin Zimmermann*: Supported by the EPSRC grant EP/S032207/1.

1 Introduction

Infinite games on finite arenas are a popular approach to the synthesis of reactive controllers from logical specifications. Originally proposed by Büchi and Landweber in 1969 [7], many variations of this classical framework have been studied, including stochastic games [12], games with partial information [14], games with delays [20], and games over infinite arenas such as pushdown graphs [42] and automatic structures [27, 28]. Other variations of this framework stem from the desire to synthesize controllers that exhibit certain user-desired properties. Examples of such properties range from controllers that need to achieve their task, e.g., reaching a goal, as quickly as possible [8] to controllers that are "robust" or "resilient" with respect to the environment in which they are deployed [3, 26, 4, 21, 25, 37, 38, 39]. Furthermore, infinite games have a plethora of applications in logic, automata theory and verification beyond the synthesis of reactive controllers. In this paper, we are concerned with the synthesis application and study infinite games with so-called *unmodeled intermittent disturbances* [13] played on configuration graphs of pushdown machines (pushdown graphs).

Pushdown graphs are finitely represented infinite graphs, typically the simplest class of such graphs one studies. Despite being conceptually simple, they have natural applications in program analysis, static code analysis, and compiler optimization [31, 32] due to their ability to capture recursion, e.g., the call stack of a procedural program. Furthermore, pushdown graphs are known to be well-behaved, and many problems on pushdown graphs are decidable (see, e.g., [5, 33, 34, 36]). In particular, Walukiewicz showed that solving parity games played on pushdown graphs is ExpTime-complete [42], paving the way for effective synthesis of *recursive* controllers. Also, Walukiewicz's result started a long and fruitful line of work on games on pushdown graphs [8, 9, 10, 24, 35]. Of particular interest is the special case of games on configuration graphs of one-counter machines, i.e., pushdown machines with a single stack symbol, which is known to be PSpace-complete [35, 23].

Games with unmodeled intermittent disturbances were originally introduced by Dallal, Neider, and Tabuada [13] to synthesize resilient controllers. The observation underlying this type of infinite game is that modeling the real-world environment of a controller in sufficiently great detail is often extremely challenging, either because parts of the environment are unknown or because simulating the environment is costly. Moreover, even if a high-resolution model of the environment is available, the resulting games often become prohibitively large. To alleviate this serious obstacle, Dallal, Neider, and Tabuada proposed to augment classical games with what they call unmodeled intermittent disturbances (in the following just called *disturbances* for the sake of brevity). Intuitively, such disturbances modify the outcome of a control action, thus modeling that the intended action of the controller did not have the desired consequences. Note, however, that disturbances are not under the control of the environment and, thus, are not antagonistic. Similarly, one does not consider the occurrence of disturbances as random events, as coming up with an appropriate stochastic error model is typically hard. Instead, the reader should understand them as rare events, such as a robot arm failing to grab an object due a physical phenomenon that has not been fully modeled.

The original work of Dallal, Neider, and Tabuada [13] provides a method to compute *optimally resilient strategies* for safety games over finite arenas, which intuitively are winning strategies that can tolerate as many disturbances as possible. In follow-up work, Neider, Weinert, and Zimmermann [30] have shown that computing optimally resilient strategies in finite arenas only incurs a polynomial overhead over solving classical games (under some mild assumptions on the winning condition), i.e., whenever a class of games is solvable without disturbances, then it is also solvable with disturbances. In particular, they have developed an algorithm that is effective for all standard winning conditions such as Rabin, Muller, and parity. Note, however, that both approaches crucially rely on the arena being finite.

The natural question, which we address here, is how to compute optimally resilient strategies for games on infinite arenas. As this is a very ambitious goal in its full generality, we restrict ourselves here to the setting of *safety games* played on *pushdown graphs*.[1]

As argued before, pushdown games are a natural starting point for investigating effective algorithms for games on infinite graphs, and safety specifications are a fundamental class of specifications in practice [15]. While this setting might seem restrictive, recall that both the ExpTime-hardness of solving pushdown games [42] and the PSpace-hardness of solving one-counter games [35] already hold for the safety condition. Thus, the complexity of solving

[1] Some of our results do carry over to other winning conditions, such as reachability and parity, or do not require the underlying arena to be a pushdown graph. If this is the case, we present our arguments and state our results as general as possible. Also, we discuss the additional challenges one has to overcome to generalize all our results to reachability and parity conditions.

pushdown games stems from the transition from finite to infinite graphs, not from the expressiveness of the winning condition. The setting we consider here is still expressive enough to model interesting applications such as reasoning about exception handling in recursive programs. Here, one is interested in determining how many exceptions the program can tolerate while still satisfying a given specification.

To capture the optimization aspect of the problem at hand, we re-use Neider, Weinert, and Zimmermann's notion of *resilience values* [30], which assigns to every vertex v of the arena an ordinal $r_{\mathcal{G}}(v) \leq \omega + 1$, where $\mathcal{G}$ denotes the game in question and ω is the first infinite ordinal. Intuitively, $r_{\mathcal{G}}(v)$ denotes how many disturbances can be tolerated by an optimally resilient strategy from v. This value can be $k \in \omega$ ($k-1$ disturbances can be tolerated, but not k), ω (finitely many disturbances can be tolerated, but not infinitely many), or $\omega + 1$ (infinitely many disturbances can be tolerated). When moving from finite to infinite arenas, however, various conceptual and technical complications arise, which make computing the resilience values of vertices and, by extension, resilient strategies challenging.

For instance, safety games over infinite arenas no longer guarantee the existence of optimally resilient strategies, i.e., in an infinite arena, one does not necessarily have a strategy that can tolerate an arbitrary finite number of disturbances from a vertex with resilience ω. Instead one has, for every $k \in \omega$, a strategy that can tolerate k disturbances, but not $k+1$.

Another complication is the fact that it is no longer possible to globally bound the finite resilience values in infinite arenas. In contrast, in the case of finite arenas, the number of vertices is a trivial bound on the finite resilience values [30]. Hence, fixed-point algorithms like the ones devised for finite arenas [13, 30] and algorithms based on exhaustive search do not necessarily terminate.

Our Contributions

In the rest of this paper, we study resilience in pushdown safety games, which we introduce in Section 2.

First, we show in Section 3 that no vertex of a finitely branching safety game (which covers pushdown games in particular) can have resilience ω. As a corollary, we show that Player 0 has positional optimally resilient strategies in finitely branching safety games. In contrast, we show that Player 0 does not necessarily have an optimally resilient strategy in infinitely branching safety games, for the reasons explained earlier.

In Sections 4 to 6, we consider the problem of determining the resilience of the initial vertex of a given pushdown safety game. First, we show in Section 4 how to characterize resilience values using classical games (without disturbances): While the notion of resilience is not defined via strategies of the antagonist, we show that one can nevertheless give control over disturbances to the antagonist, if one additionally adjusts the winning condition to control the number of occurrences of disturbances. For certain resilience values, but not all, this adjustment leads to a polynomial time reduction to solving classical games on pushdown games. The values that can be characterized in safety games are fixed finite values k and $\omega + 1$, but not ω.

We then prove that the resilience value of the initial vertex in pushdown safety games can determined in triply-exponential time (Sections 5) and that of the initial vertex in one-counter safety games in polynomial space (Section 6). The latter result is tight, as associated decision problems are shown to be PSpace-complete. To show membership, we use the following approach: We prove the existence of an upper bound on the resilience value of the initial vertex in case it is finite. With such an upper bound b, we can use the characterizations developed in Section 4 to perform an exhaustive search on the finite search space (the resilience

is either in $\{0, 1, \ldots, b\}$ or $\omega + 1$, as we have ruled out ω). For general pushdown games, this search can be implemented in triply-exponential time, as the bound b is doubly-exponential. However, relying on the simplicity of configuration graphs of one-counter systems and on the fact that the bound b is only exponential in this case, we are able to show that the search can be implemented in polynomial space for one-counter safety games. Proving the last result requires the combination of a wide range of techniques, including results from the theory of quantitative pushdown games [17], positional determinacy for quantitative pushdown games, and specifically tailored "hill-cutting" [5, 40] and "summarization" arguments [31, 19], which we generalize from individual paths in pushdown systems to strategies. Also, we show that a strategy that is optimally-resilient from the initial vertex can be computed in exponential space (triply-exponential time) for one-counter safety games (pushdown safety games).

Section 7 concludes and discusses directions for future work. Finally, in the full version [29], we present an application of our results, namely, a connection between optimally resilient strategies in pushdown safety games and optimal strategies (in the number of steps to the target) in pushdown reachability games [8, 10]. There, we also discuss which of our results obtained here carry over to pushdown reachability games and discuss the obstacles preventing us from generalizing the other results from safety to reachability.

Related Work

Resilience, and closely related notions like fault-tolerance and robustness, are not a novel concept in the context of reactive systems synthesis, with numerous formalizations having been proposed. So as to not clutter this paper too much, we refer the reader to Dallal, Neider, and Tabuada [13] as well as Neider, Weinert, and Zimmermann [30] for a comprehensive discussion of how these notions are related to the concept of unmodeled intermittent disturbances. Other notions of resilience against environmental impacts not discussed there include an approach based on imperfect information games that quantifies the resilience of controllers to noise in the input signal [2, 41] (see also the references).

Finally, let us mention that one can implement the characterization of finite resilience values presented in Section 4 by energy conditions [6, 11]. However, solving energy games on pushdown graphs is undecidable [1] and so we do not pursue this approach here. Similarly unfeasible are stochastic methods to quantify resilience in pushdown games. Indeed, checking even the most basic, almost-sure reachability conditions for stochastic games on pushdown graphs is undecidable already for single state systems or single-player games [16].

2 Preliminaries

We use the ordinals $0 < 1 < 2 < \cdots < \omega < \omega + 1 < \omega + 2$ to define resilience values. For convenience of notation, we also denote the cardinality of ω by ω.

2.1 Infinite Games with Disturbances

An *arena* (with unmodeled intermittent disturbances) $\mathcal{A} = (V, V_0, V_1, E, D)$ consists of a countable directed graph (V, E), a partition $\{V_0, V_1\}$ of V into the set of vertices V_0 of Player 0 and the set of vertices V_1 of Player 1, and a set $D \subseteq V_0 \times V$ of disturbance edges. Note that only vertices of Player 0 may have outgoing disturbance edges. We require that every vertex $v \in V$ has a successor v' with $(v, v') \in E$ to avoid finite plays. A vertex $v \in V$ is a *sink* if it has a single outgoing edge $(v, v) \in E$ leading back to itself but no outgoing disturbance edges.

A *play* in $\mathcal{A}$ is an infinite sequence $\rho = (v_0, b_0)(v_1, b_1)(v_2, b_2) \cdots \in (V \times \{0,1\})^\omega$ such that $b_0 = 0$ and for all $j > 0$: $b_j = 0$ implies $(v_{j-1}, v_j) \in E$, and $b_j = 1$ implies $(v_{j-1}, v_j) \in D$. Hence, the additional bits b_j for $j > 0$ denote whether a standard edge or a disturbance edge has been taken to move from v_{j-1} to v_j. We say ρ starts in v_0. A play prefix $(v_0, b_0) \cdots (v_j, b_j)$ is defined similarly and ends in v_j. The number of disturbances in a play $\rho = (v_0, b_0)(v_1, b_1)(v_2, b_2) \cdots$ is defined as $\#_D(\rho) = |\{j \in \omega \mid b_j = 1\}|$, which is either some $k \in \omega$ (if there are finitely many disturbances, namely k) or it is equal to ω (if there are infinitely many). A play ρ is disturbance-free, if $\#_D(\rho) = 0$.

A *game* (with unmodeled intermittent disturbances) $\mathcal{G} = (\mathcal{A}, \mathrm{Win})$ consists of an arena with set V of vertices and a winning condition $\mathrm{Win} \subseteq V^\omega$. A play $\rho = (v_0, b_0)(v_1, b_1)(v_2, b_2) \cdots$ is winning for Player 0 if $v_0 v_1 v_2 \cdots \in \mathrm{Win}$, otherwise it is winning for Player 1. Hence, winning is oblivious to occurrences of disturbances.

In this work, we focus on safety conditions, but also use the Büchi condition in proofs. Both are induced by a subset F of the set V of vertices.

- $\mathrm{Safety}(F)$ containing the sequences $v_0 v_1 v_2 \cdots \in V^\omega$ with $v_j \notin F$ for every $j \in \omega$ denotes the safety condition induced by F, which requires to avoid F.
- $\mathrm{Büchi}(F)$ containing the sequences $v_0 v_1 v_2 \cdots \in V^\omega$ with $v_j \in F$ for infinitely many $j \in \omega$ denotes the Büchi condition induced by F, which requires to visit F infinitely often.

A game $(\mathcal{A}, \mathrm{Win})$ is a *safety game* if $\mathrm{Win} = \mathrm{Safety}(F)$ for some subset F of the vertices of $\mathcal{A}$.

A *strategy* for Player $i \in \{0,1\}$ is a function $\sigma \colon V^* V_i \to V$ such that $(v_j, \sigma(v_0 \cdots v_j)) \in E$ for every $v_0 \cdots v_j \in V^* V_i$. A play $(v_0, b_0)(v_1, b_1)(v_2, b_2) \cdots$ is *consistent with* σ if $v_{j+1} = \sigma(v_0 \cdots v_j)$ for every j with $v_j \in V_i$ and $b_{j+1} = 0$, i.e., if the next vertex is the one prescribed by the strategy unless a disturbance edge is used. A strategy σ is positional, if $\sigma(v_0 \cdots v_j) = \sigma(v_j)$ for all $v_0 \cdots v_j \in V^* V_i$.

2.2 Pushdown Games

A *pushdown system* (PDS) $\mathcal{P} = (Q, \Gamma, \mathcal{E}, q_I)$ consists of a finite set Q of states with an initial state $q_I \in Q$, a stack alphabet Γ with a designated stack bottom symbol $\bot \notin \Gamma$, and a transition relation $\mathcal{E} \subseteq Q \times \Gamma_\bot \times Q \times \Gamma_\bot^{\leq 2}$, where $\Gamma_\bot = \Gamma \cup \{\bot\}$ and $\Gamma_\bot^{\leq 2} = \{w \in \Gamma_\bot^* \mid |w| \leq 2\}$. We require $\mathcal{E}$ to neither write nor delete $\bot$ from the stack. Also, we assume every PDS to be deadlock-free, i.e., for every $q \in Q$ and $A \in \Gamma_\bot$ there exist $q' \in Q$ and $w \in \Gamma_\bot^{\leq 2}$ such that $(q, A, q', w) \in \mathcal{E}$. Finally, $\mathcal{P}$ is a *one-counter system* (OCS) if $|\Gamma| = 1$.

A stack content is a word in $\Gamma^* \bot$ where the leftmost symbol is assumed to be the top of the stack. A *configuration* of $\mathcal{P}$ is a pair (q, γ) consisting of a state $q \in Q$ and a stack content $\gamma \in \Gamma^* \bot$. The stack height of a configuration (q, γ) is defined by $\mathrm{sh}(q, \gamma) = |\gamma| - 1$. Given two configurations (q, γ) and (q', γ') we write $(q, \gamma) \vdash_{\mathcal{E}} (q', \gamma')$ if there exists a transition $(q, \gamma_0, q', w) \in \mathcal{E}$ with $\gamma' = w \gamma_1 \cdots \gamma_{|\gamma|-1}$.

Fix a PDS $\mathcal{P} = (Q, \Gamma, \mathcal{E}, q_I)$, a partition $\{Q_0, Q_1\}$ of Q and an additional transition relation $\Delta \subseteq Q_0 \times \Gamma_\bot \times Q \times \Gamma_\bot^{\leq 2}$, which is also required to neither write nor delete $\bot$ from the stack. These induce the (pushdown) arena (V, V_0, V_1, E, D) with

- $V = \{(q, \gamma) \mid q \in Q, \gamma \in \Gamma^* \bot\}$ is the set of configurations of $\mathcal{P}$,
- $V_i = \{(q, \gamma) \in V \mid q \in Q_i\}$ for $i \in \{0,1\}$ is the set of configurations whose state is in Q_i,
- $E = \{(v, v') \mid v \vdash_{\mathcal{E}} v'\}$ is the set of edges, induced by the transition relation $\mathcal{E}$, and
- $D = \{(v, v') \mid v \vdash_{\Delta} v'\}$ is the set of disturbance edges, which is induced by the transition relation Δ, where $\vdash_\Delta$ is defined analogously to $\vdash_{\mathcal{E}}$.

Typically, we are interested in the initial vertex of the arena, which is defined as $(q_I, \bot)$.

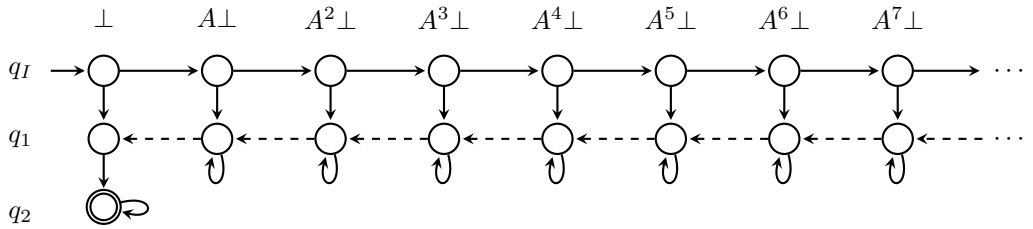

Figure 1 A one-counter arena, restricted to vertices reachable from the initial vertex $(q_I, \bot)$. All vertices are in V_0, disturbance edges are drawn as dashed arrows, and doubly-lined vertices are in F.

A *pushdown safety game* is a safety game whose arena is induced by a pushdown system $\mathcal{P}$ and whose winning condition is induced by a subset of $\mathcal{P}$'s states, i.e., $F \subseteq Q$ induces the set $\{(q, \gamma) \in V \mid q \in F\}$ of vertices. One-counter safety games are defined analogously.

When using a pushdown game as an input for an algorithm, we represent it by the underlying PDS, the partition of its states, the additional transition relation for the disturbance edges, and a subset of the states inducing the winning condition. We define the size of the input as $|Q| + |\Gamma|$, as all these objects can be represented in polynomial size in the number of states and stack symbols of the underlying PDS.

2.3 Infinite Games without Disturbances

For technical convenience, we characterize the classical notion of infinite games, i.e., those without disturbances, (see, e.g., [18]) as a special case of games with disturbances. Let $\mathcal{G}$ be a game with vertex set V. A strategy σ for Player i in $\mathcal{G}$ is said to be a winning strategy for her from $v \in V$, if every disturbance-free play that starts in v and that is consistent with σ is winning for Player i. The winning region $\mathcal{W}_i(\mathcal{G})$ of Player i in $\mathcal{G}$ contains those vertices from which Player i has a winning strategy. Thus, the winning regions of $\mathcal{G}$ are independent of the disturbance edges, i.e., we obtain the classical notion of infinite games. Player i wins $\mathcal{G}$ from v, if $v \in \mathcal{W}_i(\mathcal{G})$.

2.4 Resilient Strategies

Let $\mathcal{G}$ be a game with vertex set V and let $\alpha \in \omega + 2$. A strategy σ for Player 0 in $\mathcal{G}$ is *α-resilient* from $v \in V$ if every play ρ that starts in v, that is consistent with σ, and with $\#_D(\rho) < \alpha$, is winning for Player 0. Thus, a k-resilient strategy with $k \in \omega$ is winning even under at most $k-1$ disturbances, an ω-resilient strategy is winning even under any finite number of disturbances, and an $(\omega+1)$-resilient strategy is winning even under infinitely many disturbances.

We define the *resilience* of a vertex v of $\mathcal{G}$ as

$$r_{\mathcal{G}}(v) = \sup\{\alpha \in \omega + 2 \mid \text{Player 0 has an} \alpha\text{-resilient strategy for } \mathcal{G} \text{ from } v\}.$$

Note that the definition is not antagonistic, i.e., it is not defined via strategies of Player 1. A strategy σ is *optimally resilient* if it is $r_{\mathcal{G}}(v)$-resilient from every vertex v.

▶ **Example 1.** Consider the game $\mathcal{G} = (\mathcal{A}, \text{Safety}(F))$ where $\mathcal{A}$ is the arena from Figure 1 and $\text{Safety}(F)$ is the safety condition induced by $F = \{q_2\}$.

We have that $r_{\mathcal{G}}(q_I, A^n\bot) = \omega + 1$, $r_{\mathcal{G}}(q_1, A^n\bot) = n$ for all $n \in \omega$, and $r_{\mathcal{G}}(q_2, \bot) = 0$. Furthermore, the strategy that indefinitely stays in state q_I is optimally resilient.

3 Resilience in Infinite Safety Games

Player 0 has optimally resilient strategies in every safety game played in a finite arena [13]. In this section, we show that this result also holds for pushdown safety games, but fails for safety games in arbitrary infinite arenas. We start by observing that in safety games in infinite arenas, vertices with resilience ω may exist, unlike in safety games in finite arenas [13].

▶ **Example 2.** Consider the one-counter arena presented in Figure 1 with the safety condition induced by $F = \{q_2\}$, i.e., Player 0 wins if she avoids visiting a vertex with state q_2. As argued in Example 1, the resulting game $\mathcal{G}$ has vertices of resilience $\omega + 1$ and k, for each $k \in \omega$, i.e., all values but ω are assumed.

Let us add a vertex $v \in V_0$ to $\mathcal{G}$ with outgoing edges to all vertices of the form $(q_1, A^n\bot)$ to obtain the game $\mathcal{G}'$ (which is infinitely branching and therefore no longer a pushdown arena). Let σ_k, for $k > 0$, be a strategy that moves from v to $(q_1, A^k\bot)$. We have that $r_{\mathcal{G}'}(v) \geq \omega$, as σ_k is k-resilient from v. Consider an arbitrary strategy σ: From v, it moves to some $(q_1, A^k\bot)$ from which k disturbances force the play into the losing sink. Hence, σ is not $(k+1)$-resilient and therefore not ω-resilient. Thus, there is no optimally resilient strategy in $\mathcal{G}'$.

The underlying issue is that $r_{\mathcal{G}}(v) \geq \omega$ can be witnessed either

(a) by the existence of a strategy that is ω-resilient from v, or

(b) by the existence of a family $(\sigma_k)_{k\in\omega}$ of strategies where each σ_k is k-resilient from v, but not ω-resilient from v.

The second case only exists as ω is a limit ordinal (the only one we consider). For all $\alpha \neq \omega$, we have that $r_{\mathcal{G}}(v) = \alpha$ if and only if Player 0 has an α-resilient strategy from v. The games studied in previous work [13, 30] only exhibited the former case, as these only considered finite arenas. As witnessed in Example 2, this is no longer true in games in infinite arenas.

Note that there is a change of quantifiers between these two cases: by definition, an ω-resilient strategy is k-resilient for every $k \in \omega$, i.e., in the former case there is a uniform strategy that is k-resilient for every $k \in \omega$. In the latter case, for every $k \in \omega$, there is a strategy that is k-resilient, but not ω-resilient. Hence, in the following, we distinguish between these two cases. We say that a vertex v of a game $\mathcal{G}$ with $r_{\mathcal{G}}(v) = \omega$ has a uniform witness[2], if there is an ω-resilient strategy from v. A game with a vertex of resilience ω without a uniform witness has no optimally resilient strategy by definition.

For safety games in infinite arenas, the existence of optimally resilient strategies depends on the branching of the arena. We say that an arena (V, V_0, V_1, E, D) is *finitely branching* if the set $\{v' \mid (v, v') \in E\}$ of successors of v is finite for every $v \in V$. Otherwise, if there is a vertex with infinitely many successors, then the arena is *infinitely branching*. Note that pushdown arenas are finitely branching.

The following theorem shows that the games presented in Example 2 already exhibit all possible resilience values in safety games, and that infinite branching is necessary to obtain a vertex of resilience ω. We formulate the result for arbitrary infinite arenas, as the proof technique we use here does not rely on the arena being a pushdown arena.

▶ **Lemma 3.** *Let $\mathcal{G}$ be a safety game with vertex set V.*

1. *There is no $v \in V$ with $r_{\mathcal{G}}(v) = \omega$ that has a uniform witness.*
2. *If $\mathcal{A}$ is finitely branching, then there is no $v \in V$ with $r_{\mathcal{G}}(v) = \omega$.*

[2] Note that uniformity here refers to having a single strategy σ that is k-resilient from v for every k. It is *not* related to the concept of uniform winning strategies, i.e., strategies that are winning from every vertex in a winning region.

Finally, the main result of this section shows that optimally resilient strategies exist in all finitely branching safety games, i.e., in particular in pushdown safety games.

▶ **Theorem 4.** *Player 0 has positional optimally resilient strategies in finitely branching safety games.*

4 Characterizing Resilience Values via Classical Games

In this section, we characterize the existence of α-resilient strategies by games without disturbances. This generalizes a characterization for $\alpha = \omega + 1$ in finite arenas [30] to infinite arenas and all $\alpha \in \omega + 2$.

The main idea is to give Player 1 control over the disturbances and to restrict the number of their occurrences using the winning condition. Intuitively, when it is Player 0's turn at a vertex v, we let Player 1 first decide whether to simulate a disturbance edge from D or whether to allow Player 0 to pick a standard edge from E. To this end, we add v to Player 1's vertices and he can either move to some vertex v' such that the disturbance edge (v, v') exists. By doing his, he has to visit the fresh vertex (v, v'), which allows to keep track of the number of simulated disturbances. This vertex has exactly one outgoing edge leading to v'. On the other hand, if he does not simulate a disturbance edge, he moves from v to a fresh copy $\overline{v}$ of v from which Player 0 has edges leading to the successors of v. Finally, the moves at Player 1's original vertices are unchanged, but we subdivide the edge so that a play in the extended arena always alternates between vertices from V and auxiliary vertices.

Formally, given an arena $\mathcal{A} = (V, V_0, V_1, E, D)$, we define the rigged arena $\mathcal{A}_{\text{rig}} = (V', V_0', V_1', E', D')$ with $V' = V \cup A$ for the set

$$A = \{\overline{v} \mid v \in V_0\} \cup D \cup \{(v, v') \in E \mid v \in V_1\}$$

of auxiliary vertices, $V_0' = \{\overline{v} \mid v \in V_0\}$, $V_1' = V' \setminus V_0'$, $D' = \emptyset$, and E is the union of the following sets of edges:

- $\{(v, (v, v')), ((v, v'), v') \mid (v, v') \in D\}$: Player 1 simulates a disturbance edge $(v, v') \in D$ by moving from v to v' via the auxiliary vertex (v, v') that signifies that a disturbance is simulated.
- $\{(v, \overline{v}) \mid v \in V_0\}$: Player 1 does not simulate a disturbance edge and instead gives control to Player 0 by moving to the auxiliary vertex $\overline{v}$.
- $\{(\overline{v}, v') \mid v \in V_0 \text{ and } (v, v') \in E\}$: Player 0 has control at the auxiliary vertex $\overline{v}$ and simulates a standard move from $v \in V_0$ to v'.
- $\{(v, (v, v')), ((v, v'), v') \mid (v, v') \in E \text{ and } v \in V_1\}$: Player 1 simulates a standard move from $v \in V_1$ to v' by moving via the auxiliary vertex (v, v').

Let $R_{\geq k}$ denote the set of sequences $v_0 v_1 v_2 \cdots \in (V')^\omega$ such that $|\{j \mid v_j \in D\}| \geq k$, i.e., those plays in which Player 1 simulates at least k disturbances. Finally, given a winning condition $\text{Win} \subseteq V^\omega$ for $\mathcal{A}$, we define the rigged winning condition

$$\text{Win}_{\text{rig}} = \{v_0 v_1 v_2 \cdots \in (V')^\omega \mid v_0 \in V \text{ and } v_0 v_2 v_4 \cdots \in \text{Win}\},$$

which contains all plays in $\mathcal{A}_{\text{rig}}$ that start in V and are in Win after removing the auxiliary vertices. Note that Büchi(D) contains those plays that simulate infinitely many disturbances.

▶ **Lemma 5.** *Let $\mathcal{G} = (\mathcal{A}, \text{Win})$ be a game, let v be a vertex of $\mathcal{G}$, and let $k \in \omega$.*

1. *Player 0 has an $(\omega+1)$-resilient strategy for $\mathcal{G}$ from v if and only if $v \in \mathcal{W}_0(\mathcal{A}_{\text{rig}}, \text{Win}_{\text{rig}})$.*
2. *Player 0 has an ω-resilient strategy for $\mathcal{G}$ from v if and only if $v \in \mathcal{W}_0(\mathcal{A}_{\text{rig}}, \text{Win}_{\text{rig}} \cup$ Büchi$(D))$.*
3. *Player 0 has a k-resilient strategy for $\mathcal{G}$ from v if and only if $v \in \mathcal{W}_0(\mathcal{A}_{\text{rig}}, \text{Win}_{\text{rig}} \cup R_{\geq k})$.*

5 Resilience in Pushdown Safety Games

The goal of this section is to develop an algorithm that determines the resilience of the initial vertex of a pushdown safety game. To this end, we rely on the characterizations presented in the previous section as as well as an upper bound on the possible finite resilience values that can be realized by the initial vertex of such a game. We begin by showing that the first two characterizations presented in Lemma 5 (for $\omega+1$ and ω) are effective for pushdown games. Intuitively, we prove that a pushdown machine $\mathcal{P}$ inducing an arena $\mathcal{A}$ can in polynomial time be turned into a pushdown machine $\mathcal{P}_{\text{rig}}$ inducing the arena $\mathcal{A}_{\text{rig}}$.

We state the result for parity conditions (see, e.g., [18] for a definition of parity conditions), which subsume safety conditions.

▶ **Lemma 6.** *The following problem is ExpTime-complete (and PSpace-complete if inputs are restricted to one-counter games): "Given a pushdown parity game $\mathcal{G}$ with initial vertex v_I and $\alpha \in \{\omega, \omega+1\}$, does Player 0 have an α-resilient strategy for $\mathcal{G}$ from v_I?". If yes, such a strategy can be computed in exponential time.*

Both ExpTime-hardness and PSpace-hardness already hold for pushdown safety games and one-counter safety games, respectively. The third characterization of Lemma 5 (for $k \in \omega$) is effective as well (even for parity games). Here the running time depends on k.

▶ **Lemma 7.** *The following problem is in 2ExpTime (in ExpSpace if the input is one-counter): "Given a pushdown parity game $\mathcal{G}$ with initial vertex v_I and $k \in \omega$ (encoded in binary), does Player 0 have a k-resilient strategy for $\mathcal{G}$ from v_I?". If yes, such a strategy can be computed in doubly-exponential time.*

There are no vertices of resilience ω in pushdown safety games (Lemma 3.2). Thus, the effective characterizations we have presented so far suffice to determine the resilience of the initial vertex in such a game: First, check whether it is $\omega+1$; if not, then it has to be finite. Hence, for increasing k, check whether the resilience is greater than k. As the resilience is finite, this algorithm will eventually terminate and report the resilience correctly. However, without an upper bound on the possible finite resilience values of the initial vertex, there is no bound on the running time, just a termination guarantee. In the remainder of this section, we present a tight doubly-exponential upper bound $b(\mathcal{P})$ on the resilience of the *initial vertex* in pushdown safety games in the case the resilience is finite. That is, if $r_{\mathcal{G}}(v_I) \in \omega$ then $r_{\mathcal{G}}(v_I) < b(\mathcal{P})$. Note that any proof of the upper bound has to depend on the vertex under consideration being initial, as we have shown that there is in general no upper bound on finite resilience values assumed in pushdown safety games (cf. Example 2). The bound $b(\mathcal{P})$ only depends on the pushdown system $\mathcal{P}$ inducing the game and yields an effective algorithm to determine the resilience of the initial vertex v_I, presented as Algorithm 1.

Algorithm 1 Computing the resilience of the initial vertex v_I of a pushdown safety game $\mathcal{G} = (\mathcal{A}, \text{Safety}(F))$ induced by a PDS $\mathcal{P}$.

```
1: if v_I ∈ W_0(A_rig, Safety(F)_rig) then
2:     return ω + 1
3: for k = 1 to b(P) do
4:     if v_I ∈ W_1(A_rig, Safety(F)_rig ∪ R_{≥k}) then
5:         return k − 1
```

Given a PDS $\mathcal{P}$ with set Q of states and set Γ of stack symbols let $\mathcal{P}_{\text{rig}}$ be the PDS obtained from $\mathcal{P}$ by implementing the transformation from an arena to the rigged arena.

The cardinality of the set Q' of states of $\mathcal{P}_{\text{rig}}$ is bounded quadratically in $|Q|$ and the set of stack symbols used by $\mathcal{P}_{\text{rig}}$ is still Γ. We define $b(\mathcal{P}) = |Q'| \cdot h(\mathcal{P}) \cdot |\Gamma|^{h(\mathcal{P})}$, where $h(\mathcal{P}) = |Q'| \cdot |\Gamma| \cdot 2^{|Q'|+1} + 1$. Note that $b(\mathcal{P}) \in 2^{2^{\mathcal{O}(|\mathcal{P}|^2)}}$ and $b(\mathcal{P}) \in 2^{\mathcal{O}(|\mathcal{P}|^2)}$ if $\mathcal{P}$ is an OCS.

▶ **Lemma 8.** *Let $\mathcal{G}$ be a pushdown safety game with initial vertex v_I. If $r_{\mathcal{G}}(v_I) \neq \omega + 1$, then $r_{\mathcal{G}}(v_I) < b(\mathcal{P})$, where $\mathcal{P}$ is the PDS underlying $\mathcal{G}$.*

This upper bound immediately implies correctness of Algorithm 1, which determines the resilience of the initial vertex of a pushdown safety game.

▶ **Theorem 9.** *The following problem can be solved in triply-exponential time: "Given a pushdown safety game $\mathcal{G}$ with initial vertex v_I, determine $r_{\mathcal{G}}(v_I)$". If yes, an $r_{\mathcal{G}}(v_I)$-resilient strategy can be computed in triply-exponential time.*

Note that there is a gap between the triply-exponential upper bound and the exponential lower bound obtained for the related decision problems for ω and $\omega + 1$ (Lemma 6).

The complexity for the special case of one-counter safety games is much smaller, i.e., the resilience of the initial vertex can be computed in exponential space, as the winner of one-counter safety games can be computed in polynomial space [35] and the upper bound on finite resilience values of the initial vertex is only exponential. Furthermore, a witnessing strategy can be computed in doubly-exponential time using Lemma 7. In the next section, we prove that one can do even better by exploiting the simple structure of one-counter arenas.

To conclude this section, we claim that the bound $b(\mathcal{P})$ on the resilience of an initial vertex in a pushdown safety game with finite resilience is tight: There is an exponential lower bound for the one-counter case and a doubly-exponential lower bound for the pushdown case. Both constructions are generalizations of constructions that appeared in the literature previously [10]. To simplify our notation, let p_j denote the j-th prime number and define the primorial $p_k\# = \Pi_{j=1}^{k} p_j$ to be the product of the first k prime numbers. We have $p_k\# \geq 2^k$.

▶ **Lemma 10.** *Let $k \in \omega$.*

1. *There is a one-counter safety game $\mathcal{G}_k$ with initial state v_I such that $r_{\mathcal{G}}(v_I) = p_k\#$ and the underlying OCS has polynomially many states in k.*
2. *There is a pushdown safety game $\mathcal{G}'_k$ with initial state v_I such that $r_{\mathcal{G}}(v_I) = 2^{p_k\#} - 1$ and the underlying PDS has polynomially many states in k and two stack symbols.*

6 Resilience in One-counter Safety Games

In this section, we show that one can compute the resilience of the initial vertex in a one-counter safety game in polynomial space, significantly improving the exponential space requirement derived in the previous section.

▶ **Theorem 11.** *The following problem can be solved in polynomial space: "Given a one-counter safety game $\mathcal{G}$ with initial vertex v_I, determine $r_{\mathcal{G}}(v_I)$".*

To prove this result, we show that one can implement Algorithm 1 in polynomial space if the underlying pushdown system is one-counter. In this case, one can run the check "$v_I \in \mathcal{W}_0(\mathcal{A}_{\text{rig}}, \text{Safety}(F)_{\text{rig}})$" in Line 1 in polynomial space due to Lemma 6, and can implement the counter in Line 3 in polynomial space, as the upper bound $b(\mathcal{P})$ is exponential (see the definition on Page 10). It remains to show that one can check in polynomial space, for a given $k \leq b(\mathcal{P})$, if $v_I \in \mathcal{W}_1(\mathcal{A}_{\text{rig}}, \text{Safety}(F)_{\text{rig}} \cup R_{\geq k})$ holds. In the rest of this section we show that this is indeed possible.

Fix, the rigged game $\mathcal{G}_k = (\mathcal{A}_{\text{rig}}, \text{Safety}(F)_{\text{rig}} \cup R_{\geq k})$ for some $k \leq b(\mathcal{P})$ with $\mathcal{A}_{\text{rig}} = (V', V_0', V_1', E', \emptyset)$, with initial vertex v_I, where $\mathcal{P}$ is the OCS underlying the original game $\mathcal{G}$ that induces $\mathcal{G}_k$. We show that the existence of winning strategies for Player 1 in $\mathcal{G}_k$ can be witnessed by a finite graph structure, as follows.

A *strategy graph* for $\mathcal{G}_k$ is a tuple $(V^\circ, E^\circ, \mu_r^\circ, \mu_d^\circ)$ with $\mu_r^\circ \colon V^\circ \to \{0, \dots, k-1\}$ and $\mu_d^\circ \colon V^\circ \to \{0, \dots, |V^\circ|\}$ such that the following properties are satisfied:

1. (V°, E°) is a directed graph with $V^\circ \subseteq V'$, $E^\circ \subseteq E'$, $v_I \in V^\circ$, and $\text{sh}(v) \leq (2k)^{|Q|^2}$ for all $v \in V^\circ$. Note that $(2k)^{|Q|^2}$ is exponential in the size of the pushdown system $\mathcal{P}$ underlying $\mathcal{G}$, even though $k \leq b(\mathcal{P})$ may itself be exponential.
2. For all $v \in (V^\circ \cap V_0') \setminus F$ and all $(v, v') \in E'$, we have $(v, v') \in E^\circ$.
3. For all $v \in (V^\circ \cap V_1') \setminus F$ there is a unique outgoing edge $(v, v') \in E'$ with $(v, v') \in E^\circ$.
4. For all $(v, v') \in E^\circ$, we have $\mu_r^\circ(v) \geq \mu_r^\circ(v')$ with strict inequality if $v \in D$.
5. For all $(v, v') \in E^\circ$, we have $\mu_d^\circ(v) > \mu_d^\circ(v')$.

▶ **Lemma 12.** *Player* 1 *wins* $\mathcal{G}_k$ *from* v_I *if and only if there exists a strategy graph for* $\mathcal{G}_k$*.*

To simplify the proof, we transform $\mathcal{G}_k$ into a game $\mathcal{G}_k'$ where all reachable vertices in F are sinks of stack height zero. To do this, we replace all outgoing (standard and disturbance) edges of vertices $(q, A^n\bot) \in F$ with $n > 0$ by an edge to $(q, A^{n-1}\bot)$ (which is also in F) and the all outgoing (standard and disturbance) edges of vertices $(q, \bot) \in F$ by an edge to a sink vertex $(q_f, \bot)$, where q_f is a fresh state. Then, $\mathcal{G}_k'$ is the game played in the modified arena with winning condition $\text{Safety}(\{q_f\})_{\text{rig}} \cup R_{\geq k}$. Intuitively, once a vertex in F is reached in the modified arena, the players no longer have strategic choices; instead, the stack is emptied (without simulating any disturbances) and the unsafe sink vertex $(q_f, \bot)$ is reached.

It is straightforward to verify that we have $v \in \mathcal{W}_i(\mathcal{G}_k)$ if and only $v \in \mathcal{W}_i(\mathcal{G}_k')$ for every vertex of $\mathcal{A}_{\text{rig}}$ and $i \in \{0, 1\}$ by transferring winning strategies between the games. So, in the following, we assume without loss of generality, that the only vertices of $\mathcal{G}_k$ in F that are reachable from the initial vertex are sinks of stack height zero. In this situation, a play can no longer simulate a disturbance edge once a vertex in F has been reached.

To prove Lemma 12, we show that if Player 1 wins $\mathcal{G}_k$ with some arbitrary winning strategy, then also with a winning strategy that can be turned into a strategy graph. To simplify our notation, given a strategy τ, let $\text{maxSh}(\tau) = \sup_v \text{sh}(v)$, where v ranges over all vertices reachable by a play prefix starting in v_I that is consistent with τ, i.e., $\text{maxSh}(\tau)$ is the maximal stack height visited by a play that is starting in the initial vertex and consistent with τ. Using this, we show that Player 1 wins $\mathcal{G}_k$ from v_I if and only if he has a positional winning strategy from v_I with $\text{maxSh}(\tau) \leq (2k)^{|Q|^2}$. The latter can then be transformed into a strategy graph.

We only have to consider the implication from left to right, as the other one is trivial. Let Player 1 win $\mathcal{G}_k$ from v_I, i.e., he has a winning strategy τ for $\mathcal{G}_k$ from v_I. We proceed in two steps: First, We turn τ in a positional winning strategy τ' from v_I (Lemma 13). Then, we turn τ' into a positional winning strategy τ'' with $\text{maxSh}(\tau'') \leq (2k)^{|Q|^2}$ (Lemma 14).

For the first step, we generalize a standard argument for turning an arbitrary, not necessarily positional, winning strategy τ in a reachability game into a positional one: At a vertex $v \notin F$, consider all play prefixes that are consistent with τ and end in v, and mimic the move τ prescribes for a longest one (call it $\text{rep}(v)$). The resulting strategy τ' is obviously positional and winning as every play consistent with τ' and ending in some $v \notin F$ can be shown to be at most as long as the play $\text{rep}(v)$ whose moves are mimicked to define $\tau'(v)$. Here, we have to refine this argument to ensure that the resulting strategy τ' still simulates at most $k-1$ disturbances during each play.

▶ **Lemma 13.** *If Player* 1 *wins* $\mathcal{G}_k$ *from* v_I *then he has a positional winning strategy for* $\mathcal{G}_k$ *from* v_I*.*

The second step of our construction is to bound the stack height reached by plays consistent with the winning strategy (while preserving positionality). To this end, we generalize a classical argument for pushdown safety games: In such games, Player 1, who has a reachability objective, has a positional winning strategy τ from v_I with exponentially bounded $\mathrm{maxSh}(\tau)$, if he wins at all from v_I. This is typically proven by a "hill-cutting" argument [5, 40] showing that a winning strategy exceeding this bound can be turned into one of smaller maximal stack height by removing infixes of plays that increase the stack without reaching states that have not been reached at smaller stack height already. Here, we again have to generalize this argument to additionally ensure that the number of simulated disturbances remains bounded by $k-1$. This is done using "summarizations" of paths in pushdown systems (see e.g. [31, 19]) that take the number of disturbances into account.

▶ **Lemma 14.** *If Player* 1 *wins* $\mathcal{G}_k$ *from* v_I *then he has a positional winning strategy from* v_I *with* $\mathrm{maxSh}(\tau) \leq (2k)^{|Q|^2}$*.*

A positional strategy as in Lemma 14 is essentially a strategy graph. So, we have proven Lemma 12: The existence of strategy graphs for $\mathcal{G}_k$ captures Player 1 winning $\mathcal{G}_k$. Hence, it remains to prove that we can decide the existence of strategy graphs in polynomial space. Here, we use the fact that k is at most $b(\mathcal{P}) \in \mathcal{O}(2^{|\mathcal{P}|^2})$, where $\mathcal{P}$ is the pushdown system underlying the game inducing $\mathcal{G}_k$, to guess and verify a strategy graph in polynomial space.

▶ **Lemma 15.** *The following problem is in* PSpace*: "Given a one-counter safety game* $\mathcal{G}$ *induced by a PDS* $\mathcal{P}$ *and* $k \leq b(\mathcal{P})$ *(encoded in binary), is there a strategy graph for* $\mathcal{G}_k$*?".*

While we consider one-counter systems with unit updates, i.e., each transition changes the counter value by at most one, our results are also applicable to one-counter systems where each transition updates the counter by some integer (encoded in binary). Such *binary updates* can be simulated by unit updates, albeit with an exponential blowup. Hence, the algorithm above computes the resilience of the initial vertex of a one-counter safety game with binary updates in exponential space. A matching lower bound follows from the ExpSpace hardness of solving disturbance-free one-counter safety games with binary updates [22].

7 Conclusion

In this work, we have investigated pushdown safety games with disturbances, thereby extending the theory of games with disturbances from finite to infinite arenas. In particular, we have determined the possible resilience values in safety games, presented effective characterizations for all possible values, and presented algorithms that determine the resilience of the initial vertex (and a witnessing strategy) in one-counter and pushdown safety games. As an application of our results, we obtain a polynomial space algorithm for computing optimal winning strategies for one-counter reachability games (see the full version for details [29]). This is, to the best of our knowledge, the first improvement over the general doubly-exponential time algorithm for pushdown reachability games due to Carayol and Hague [10].

The algorithm computing the resilience in one-counter safety games runs in polynomial space, which is optimal, as the corresponding decision problems are PSpace-complete. However, the algorithm for pushdown games has triply-exponential running time. Here, there is a gap, as some of the corresponding decision problems are ExpTime-complete (e.g., those for resilience $\omega+1$ and ω) while the complexity of others is open (e.g., that for finite resilience

values). In future work, we aim to close this gap. An interesting first step in that direction would be to determine the complexity of checking whether the resilience of the initial vertex is at least k, where k is part of the input and encoded in binary. Here, one has to keep in mind that algorithms for computing the resilience also yield algorithms computing optimal strategies in reachability games. The latter problem also has a complexity gap between the currently best algorithms and known lower bounds. Finally, another obvious open problem is to consider more general winning conditions, e.g., reachability (see the full version [29] for preliminary results) or parity.

The main obstacle is that one either has to develop an effective characterization of vertices with resilience ω without a uniform witness, or to obtain an upper bound on the finite resilience value an initial vertex can assume. The first option is challenging due to the quantifier change discussed in Section 5. Hence, the more promising route seems to be the second option. The main challenge here is to bound the number of disturbances that are necessary to prevent Player 0 from ever reaching the target states, i.e., Player 1 now has a safety objective in conjunction with a limited number of disturbances at his disposal.

References

1 Parosh Aziz Abdulla, Mohamed Faouzi Atig, Piotr Hofman, Richard Mayr, K. Narayan Kumar, and Patrick Totzke. Infinite-state energy games. In *CSL-LICS 2014*, pages 7:1–7:10. ACM, 2014. doi:10.1145/2603088.2603100.

2 Shaull Almagor and Orna Kupferman. Latticed-LTL synthesis in the presence of noisy inputs. *Discrete Event Dynamic Systems*, 27(3):547–572, 2017. doi:10.1007/s10626-017-0242-0.

3 Roderick Bloem, Krishnendu Chatterjee, Karin Greimel, Thomas A. Henzinger, Georg Hofferek, Barbara Jobstmann, Bettina Könighofer, and Robert Könighofer. Synthesizing robust systems. *Acta Inf.*, 51(3-4):193–220, 2014. doi:10.1007/s00236-013-0191-5.

4 Roderick Bloem, Krishnendu Chatterjee, Thomas A. Henzinger, and Barbara Jobstmann. Better quality in synthesis through quantitative objectives. In *CAV 2009*, volume 5643 of *LNCS*, pages 140–156. Springer, 2009. doi:10.1007/978-3-642-02658-4_14.

5 Stanislav Böhm, Stefan Göller, and Petr Jancar. Bisimulation equivalence and regularity for real-time one-counter automata. *J. Comput. Syst. Sci.*, 80(4):720–743, 2014. doi:10.1016/j.jcss.2013.11.003.

6 Patricia Bouyer, Ulrich Fahrenberg, Kim Guldstrand Larsen, Nicolas Markey, and Jirí Srba. Infinite runs in weighted timed automata with energy constraints. In *FORMATS 2008*, volume 5215 of *LNCS*, pages 33–47. Springer, 2008.

7 J. Richard Büchi and Lawrence H. Landweber. Solving sequential conditions by finite-state strategies. *Trans. Amer. Math. Soc.*, 138:295–311, 1969.

8 Thierry Cachat. Symbolic strategy synthesis for games on pushdown graphs. In *ICALP 2002*, volume 2380 of *LNCS*, pages 704–715. Springer, 2002. doi:10.1007/3-540-45465-9_60.

9 Thierry Cachat. Higher order pushdown automata, the caucal hierarchy of graphs and parity games. In *ICALP 2003*, volume 2719 of *LNCS*, pages 556–569. Springer, 2003. doi:10.1007/3-540-45061-0_45.

10 Arnaud Carayol and Matthew Hague. Optimal strategies in pushdown reachability games. In *MFCS 2018*, volume 117 of *LIPIcs*, pages 42:1–42:14. Schloss Dagstuhl - LZI, 2018. doi:10.4230/LIPIcs.MFCS.2018.42.

11 Arindam Chakrabarti, Luca de Alfaro, Thomas A. Henzinger, and Mariëlle Stoelinga. Resource interfaces. In *EMSOFT 2003*, volume 2855 of *LNCS*, pages 117–133. Springer, 2003.

12 Anne Condon. On algorithms for simple stochastic games. In *Advances in Computational Complexity Theory*, pages 51–73. American Mathematical Society, 1993.

13 Eric Dallal, Daniel Neider, and Paulo Tabuada. Synthesis of safety controllers robust to unmodeled intermittent disturbances. In *CDC 2016*, pages 7425–7430. IEEE, 2016. doi:10.1109/CDC.2016.7799416.

14 L. Doyen and J.-F. Raskin. Games with imperfect information: Theory and algorithms. In *Lectures in Game Theory for Computer Scientists*, pages 185–212. Cambridge University Press, 2011.

15 Matthew B. Dwyer, George S. Avrunin, and James C. Corbett. Patterns in property specifications for finite-state verification. In *ICSE 1999*, pages 411–420. ACM, 1999. doi:10.1145/302405.302672.

16 Kousha Etessami and Mihalis Yannakakis. Recursive markov decision processes and recursive stochastic games. *J. ACM*, 62(2):11:1–11:69, 2015. doi:10.1145/2699431.

17 Wladimir Fridman and Martin Zimmermann. Playing pushdown parity games in a hurry. In *GandALF 2012*, volume 96 of *EPTCS*, pages 183–196, 2012. doi:10.4204/EPTCS.96.14.

18 Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research*, volume 2500 of *LNCS*. Springer, 2002. doi:10.1007/3-540-36387-4.

19 Lukás Holík, Roland Meyer, and Sebastian Muskalla. Summaries for context-free games. In *FSTTCS*, volume 65 of *LIPIcs*, pages 41:1–41:16. Schloss Dagstuhl - LZI, 2016. doi:10.4230/LIPIcs.FSTTCS.2016.41.

20 Frederick A. Hosch and Lawrence H. Landweber. Finite delay solutions for sequential conditions. In *ICALP 1972*, pages 45–60. North-Holland, Amsterdam, 1972.

21 Chung-Hao Huang, Doron A. Peled, Sven Schewe, and Farn Wang. A game-theoretic foundation for the maximum software resilience against dense errors. *IEEE Trans. Software Eng.*, 42(7):605–622, 2016. doi:10.1109/TSE.2015.2510001.

22 Paul Hunter. Reachability in succinct one-counter games. In Mikołaj Bojańczyk, Slawomir Lasota, and Igor Potapov, editors, *RP 2015*, volume 9328 of *LNCS*, pages 37–49. Springer, 2015. doi:10.1007/978-3-319-24537-9_5.

23 Petr Jancar and Zdenek Sawa. A note on emptiness for alternating finite automata with a one-letter alphabet. *Inf. Process. Lett.*, 104(5):164–167, 2007. doi:10.1016/j.ipl.2007.06.006.

24 Orna Kupferman and Moshe Y. Vardi. An automata-theoretic approach to reasoning about infinite-state systems. In *CAV 2000*, volume 1855 of *LNCS*, pages 36–52. Springer, 2000. doi:10.1007/10722167_7.

25 Rupak Majumdar, Elaine Render, and Paulo Tabuada. A theory of robust omega-regular software synthesis. *ACM Trans. Embedded Comput. Syst.*, 13(3):48:1–48:27, 2013. doi:10.1145/2539036.2539044.

26 David E. Muller and Paul E. Schupp. The theory of ends, pushdown automata, and second-order logic. *Theor. Comput. Sci.*, 37:51–75, 1985. doi:10.1016/0304-3975(85)90087-8.

27 Daniel Neider. Reachability games on automatic graphs. In *CIAA 2010*, volume 6482 of *LNCS*, pages 222–230. Springer, 2010. doi:10.1007/978-3-642-18098-9_24.

28 Daniel Neider and Ufuk Topcu. An automaton learning approach to solving safety games over infinite graphs. In *TACAS 2016*, volume 9636 of *LNCS*, pages 204–221. Springer, 2016. doi:10.1007/978-3-662-49674-9_12.

29 Daniel Neider, Patrick Totzke, and Martin Zimmermann. Optimally resilient strategies in pushdown safety games. *arXiv*, 1912.04771, 2019.

30 Daniel Neider, Alexander Weinert, and Martin Zimmermann. Synthesizing optimally resilient controllers. In *CSL 2018*, volume 119 of *LIPIcs*, pages 34:1–34:17. Schloss Dagstuhl - LZI, 2018. doi:10.4230/LIPIcs.CSL.2018.34.

31 Thomas W. Reps, Susan Horwitz, and Shmuel Sagiv. Precise interprocedural dataflow analysis via graph reachability. In *POPL 1995*, pages 49–61. ACM Press, 1995. doi:10.1145/199448.199462.

32 Thomas W. Reps, Akash Lal, and Nicholas Kidd. Program analysis using weighted pushdown systems. In *FSTTCS 2007*, volume 4855 of *LNCS*, pages 23–51. Springer, 2007. doi:10.1007/978-3-540-77050-3_4.

33 Géraud Sénizergues. L(A)=L(B)? decidability results from complete formal systems. *Theor. Comput. Sci.*, 251(1-2):1–166, 2001. doi:10.1016/S0304-3975(00)00285-1.

34 Géraud Sénizergues. The bisimulation problem for equational graphs of finite out-degree. *SIAM J. Comput.*, 34(5):1025–1106, 2005. doi:10.1137/S0097539700377256.

35 Olivier Serre. Parity games played on transition graphs of one-counter processes. In *FOSSACS 2006*, volume 3921 of *LNCS*, pages 337–351. Springer, 2006. doi:10.1007/11690634_23.

36 Jiří Srba. Roadmap of infinite results. In Gheorghe Paun, Grzegorz Rozenberg, and Arto Salomaa, editors, *Current Trends in Theoretical Computer Science*, pages 337—-350. World Scientific, 2004. doi:10.1142/9789812562494_0054.

37 Paulo Tabuada, Sina Yamac Caliskan, Matthias Rungger, and Rupak Majumdar. Towards robustness for cyber-physical systems. *IEEE Trans. Automat. Contr.*, 59(12):3151–3163, 2014. doi:10.1109/TAC.2014.2351632.

38 Paulo Tabuada and Daniel Neider. Robust linear temporal logic. In *CSL 2016*, volume 62 of *LIPIcs*, pages 10:1–10:21. Schloss Dagstuhl - LZI, 2016. doi:10.4230/LIPIcs.CSL.2016.10.

39 Ufuk Topcu, Necmiye Ozay, Jun Liu, and Richard M. Murray. On synthesizing robust discrete controllers under modeling uncertainty. In *HSCC 2012*, pages 85–94. ACM, 2012. doi:10.1145/2185632.2185648.

40 Leslie G. Valiant. *Decision Procedures for Families of Deterministic Pushdown Automata*. PhD thesis, University of Warwick, 1973.

41 Yaron Velner and Alexander Rabinovich. Church synthesis problem for noisy input. In *FoSSaCS 2011*, volume 6604 of *LNCS*, pages 275–289. Springer, 2011. doi:10.1007/978-3-642-19805-2_19.

42 Igor Walukiewicz. Pushdown processes: Games and model-checking. *Inf. Comput.*, 164(2):234–263, 2001. doi:10.1006/inco.2000.2894.