RWTH Aachen

Department of Computer Science
Technical Report

Time-optimal Winning Strategies for Poset Games

Martin Zimmermann

ISSN 0935–3232 · Aachener Informatik Berichte · AIB-2009-13

RWTH Aachen · Department of Computer Science · May 2009

Time-optimal Winning Strategies for Poset Games*

Martin Zimmermann**

Lehrstuhl Informatik 7, RWTH Aachen University, Germany
zimmermann@automata.rwth-aachen.de

Abstract. We introduce a novel winning condition for infinite two-player games on graphs which extends the request-response condition and better matches concrete applications in scheduling or project planning. In a poset game, a request has to be responded by multiple events in an ordering over time that is compatible with a given partial ordering of the events. Poset games are zero-sum, but there are plays that are more desirable than others, i.e., those in which the requests are served quickly. We show that optimal strategies (with respect to long term average accumulated waiting times) exist. These strategies are implementable with finite memory and are effectively computable.

1 Introduction

The use of two-player games of infinite duration has a long history in the synthesis of controllers for reactive systems (see [3] for an overview). Classically, the quality of a winning strategy is measured in the size of the memory needed to implement it. But often there are other natural quality measures: in many games (even if they are zero-sum) there are winning plays for Player 0 that are more desirable than others, often given by notions of waiting that reflect periods of waiting in the modeled system. In reachability games, this can be the number of steps before the play reaches one of the designated vertices, in Büchi games the number of steps between visits of the designated vertices, and in parity games the number of steps between visits of the minimal even color seen infinitely often.

Another winning condition with a natural notion of waiting is the request-response condition [6]. It is given by pairs (Q_j, P_j) of subsets of the graph's vertices. Player 0 wins a play if every visit of Q_j is eventually responded by a visit of P_j. The waiting time is given by the number of steps between a request and the next response. As there might be several request-response pairs, there is a trade-off between the pairs: it can be favorable to delay the response of a pair to answer another request more quickly. Wallmeier [5] defined the value of a play to be the long-term average accumulated waiting time and the value of a strategy to be the worst-case outcome. He then proved that optimal winning strategies exist and are effectively computable (see also [4, 7]).

However, request-response winning conditions are often too weak to express real-life requirements concisely, because a request is responded by a single event. Imagine an intersection with a level crossing: if a train approaches the crossing (a request), then all traffic lights have to be switched to red, then the boom barriers are lowered, the train gets an all-clear signal and crosses the intersection. Afterwards, the barriers are raised and the lights are switched to green. It

* This report is an extended version of [8].

** The author's work was supported by the ESF project *Games for Analysis and Synthesis of Interactive Computational Systems (GASICS).*

would be rather cumbersome to model this requirement using request-response conditions with a single event as response. Another example is motivated by project planning: a project consists of several subtasks (and their durations) and a partial ordering of the subtasks, e.g., specifying that the roof of a house cannot be constructed before the walls are built. A plan is then a linearization of this partial ordering.

These examples motivate to replace a response by a partially ordered set of events and require Player 0 to answer every request by an embedding of these events in time. This generalization of request-response games retains the natural definition of waiting times. Hence, the framework for request-response games can be adapted to the new type of games, called poset games.

We prove that optimal winning strategies for poset games exist, which are again finite-state and effectively computable. To this end, we adapt the proof presented in [4] for request-response games. However, the increased expressiveness of poset games requires substantial changes. As a request is no longer responded by a single event, there can be different requests that are answered to a different degree at a given position, i.e., the embeddings can overlap. This requires additional bookkeeping of the events that still have to be embedded and changes to the definition of waiting times. Informally, in request-response games, there is a single clock for every pair (Q_j, P_j) that is started when Q_j is visited and stopped as soon as P_j is visited afterwards; requests that are encountered, while the clock is already active, are ignored. This is no longer possible in poset games: here, we need a clock for every request, due to the overlapping of embeddings. Hence, we do not only have to bound the waiting times to obtain our result, but also the number of open requests, i.e., the number of active clocks.

This paper is structured as follows: Section 2 fixes our notation. Poset games are introduced in Section 3 and solved by a reduction to Büchi games in Section 4. Finally, in Section 5 the existence of optimal strategies is proven. Section 6 gives some open problems and pointers to further research.

2 Definitions

Throughout this paper let P be a set of *events*. The power set of a set S is denoted by 2^S, $\mathbb{N}$ is the set of non-negative integers, and let $[n]:=\{1,\dots,n\}$. The prefix-ordering on words is denoted by $\sqsubseteq$, its strict version by $\sqsubset$. Given a sequence $(w_n)_{n\in\mathbb{N}}$ of finite words such that $w_n \sqsubset w_{n+1}$ for all n, $\lim_{n\to\infty} w_n$ denotes the unique ω-word induced by the w_n. Let $(f_n)_{n\in\mathbb{N}}$ be a sequence of functions $f_n : A \to B$ and $f : A \to B$. We say that $(f_n)_{n\in\mathbb{N}}$ *converges* to f, $\lim_{n\to\infty} f_n = f$, if $\forall a \in A\, \exists n_a \in \mathbb{N}\, \forall n \geq n_a : f_n(a) = f(a)$.

Infinite Games. An *(initialized and labeled) arena* $G = (V, V_0, V_1, E, s_0, l_G)$ consists of a finite directed graph (V, E), a partition $\{V_0, V_1\}$ of V denoting the positions of *Player* 0 and *Player* 1, an *initial vertex* $s_0 \in V$, and a *labeling function* $l_G : V \to 2^P$. It is assumed that every vertex has at least one outgoing edge. A *play* $\rho = \rho_0\rho_1\rho_2\dots$ is an infinite path starting in s_0. A *strategy for Player* i is a (partial) mapping $\sigma : V^*V_i \to V$ such that $(s, \sigma(ws)) \in E$ for all $w \in V^*$ and all $s \in V_i$. A play ρ is *consistent with* σ if $\rho_{n+1} = \sigma(\rho_0, \dots \rho_n)$ for all

$\rho_n \in V_i$. The unique play consistent with the strategies σ for Player 0 and τ for Player 1 is denoted by $\rho(\sigma, \tau)$.

A *game* $\mathcal{G} = (G, \varphi)$ consists of an arena G and a *winning condition* φ specifying the set of *winning plays* for Player 0. All other plays are won by Player 1. A strategy σ is a *winning strategy for Player* i if every play consistent with σ is won by Player i. Player i *wins* $\mathcal{G}$ (and Player $1-i$ loses $\mathcal{G}$) if she has a winning strategy for $\mathcal{G}$. A game is *determined* if one of the Players has a winning strategy.

Game Reductions. A *memory structure* $\mathfrak{M} = (M, m_0, \text{update})$ *for* G consists of a set M of *memory states*, an *initial memory state* $m_0 \in M$, and an *update function* update : $M \times V \to M$. The update function can be extended to a function $\text{update}^* : V^* \to M$ by defining $\text{update}^*(s_0) = m_0$ and $\text{update}^*(ws) = \text{update}(\text{update}^*(w), s)$. A *next-move function for Player* i $\text{next} : V_i \times M \to V$ has to satisfy $(s, \text{next}(s, m)) \in E$ for all $s \in V_i$ and all $m \in M$. It induces a *strategy* σ *with memory* $\mathfrak{M}$ via $\sigma(ws) = \text{next}(s, \text{update}^*(ws))$. A strategy is called *finite-state* if it can be implemented with finite memory, and *positional* if it can be implemented with a single memory state.

An arena G and a memory structure $\mathfrak{M}$ for G induce the *expanded arena* $G \times \mathfrak{M} = (V \times M, V_0 \times M, V_1 \times M, E', (s_0, m_0), l_{G\times\mathfrak{M}})$ where $((s, m), (s', m')) \in E'$ iff $(s, s') \in E$ and $\text{update}(m, s') = m'$, and $l_{G\times\mathfrak{M}}(s, m) = l_G(s)$. Every play $\rho' = (\rho_0, m_0)(\rho_1, m_1)(\rho_2, m_2)\ldots$ in $G \times \mathfrak{M}$ has a unique *projected play* $\rho = \rho_0\rho_1\rho_2\ldots$ in G. Conversely, every play $\rho = \rho_0\rho_1\rho_2\ldots$ in G has a unique *expanded play* $\rho' = (\rho_0, m_0)(\rho_1, m_1)(\rho_2, m_2)\ldots$ in $G \times \mathfrak{M}$ defined by $m_{n+1} = \text{update}(m_n, \rho_{n+1})$. A game $\mathcal{G} = (G, \varphi)$ is *reducible* to $\mathcal{G}' = (G', \varphi')$ via $\mathfrak{M}$, written $\mathcal{G} \leq_{\mathfrak{M}} \mathcal{G}'$, if $G' = G \times \mathfrak{M}$ and every play in $\mathcal{G}'$ is won by the player who wins the projected play in $\mathcal{G}$.

Remark 1. If $\mathcal{G} \leq_{\mathfrak{M}} \mathcal{G}'$ and Player i has a positional winning strategy for $\mathcal{G}'$, then she also has a finite-state winning strategy with memory $\mathfrak{M}$ for $\mathcal{G}$.

Partially Ordered Sets. A *(labeled) partially ordered set* (*poset* for short) $\mathcal{P} = (D, \preceq, l_{\mathcal{P}})$ consists of a domain D, a reflexive, antisymmetric and transitive relation $\preceq$ over D, and a labeling function $l_{\mathcal{P}} : D \to P$. The set of non-empty upwards-closed subsets of $\mathcal{P}$ is denoted by $\text{Up}(\mathcal{P})$; its size can be bounded by $|D| \leq |\text{Up}(\mathcal{P})| \leq 2^{|D|} - 1$.

If D is finite, we define the *transitive reduction* $\preceq^{red}$ *of* $\preceq$ by $d \preceq^{red} d'$ iff $d \neq d'$ and there is no d'' such that $d'' \neq d$, $d'' \neq d'$, and $d \preceq d'' \preceq d'$. The reduction $\preceq^{red}$ contains all the essential information of $\preceq$, i.e., the reflexive and transitive closure of $\preceq^{red}$ is $\preceq$.

Let ρ be an infinite path in an arena G with labeling function l_G. An *embedding in time*, *embedding* for short, of $\mathcal{P}$ in ρ is a function $f : D \to \mathbb{N}$ such that $l_{\mathcal{P}}(d) \in l_G(\rho_{f(d)})$ and $d \preceq d'$ implies $f(d) \leq f(d')$. An embedding of $\mathcal{P}$ in a finite path w is defined analogously.

The *length of an embedding* f is $\max_{d\in D} f(d)$, and an embedding is *minimal*, if its length is minimal in the set of all embeddings, i.e., $\max_{d\in D} f(d) \leq \max_{d\in D} f'(d)$ for all embeddings f'.

Example 1. A diagram of $\mathcal{P} = (2^{\{a,b\}}, \subseteq)$ is given in the upper part of Figure 1. The solid edges represent the ordering relation. The lower part shows an embedding f of $\mathcal{P}$ (where $l(S) = S$ for every $S \subseteq \{a, b\}$). The labeling of ρ is given by the subsets below the positions. The length of f is 7, but it is not a minimal embedding.

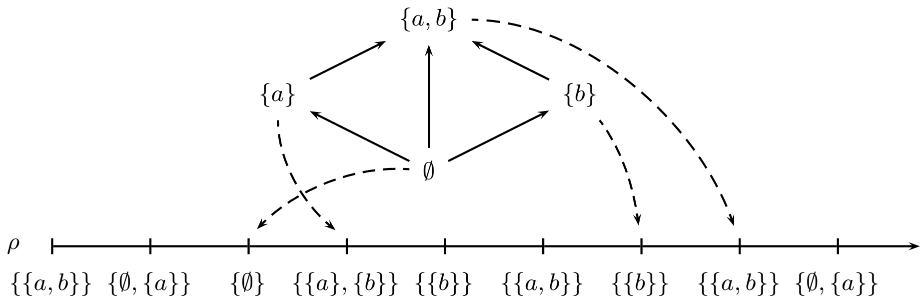

Fig. 1. A poset $\mathcal{P}$ and an embedding f of $\mathcal{P}$ in ρ

3 Poset Games

In this section, we introduce poset games and the framework for defining waiting times and time-optimal winning strategies. We close this section by stating some simple properties about poset games and the framework for time-optimal strategies.

A *poset game* $\mathcal{G} = (G, (q_j, \mathcal{P}_j)_{j\in[k]})$ consists of an arena G as above and a finite collection of *(request-poset) conditions* $(q_j, \mathcal{P}_j)$ where $q_j \in P$ is a *request (of condition j)* and $\mathcal{P}_j = (D_j, \preceq_j, l_j)$ is a finite labeled poset. Player 0 wins a play ρ iff $q_j \in l_G(\rho_n)$ implies that $\mathcal{P}_j$ can be embedded in $\rho_n\rho_{n+1}\rho_{n+2}\ldots$ for all $j \in [k]$ and all $n \in \mathbb{N}$.

For the remainder of this paper, let $(G, (q_j, \mathcal{P}_j)_{j\in[k]})$ be a poset game, where $\mathcal{P}_j = (D_j, \preceq_j, l_j)$. Furthermore, let $c_j := |\mathrm{Up}(\mathcal{P}_j)|$ and $c := \sum_{j=1}^{k} c_j$.

A vertex s such that $q_j \in l_G(s)$ is called *request of condition* j, too. A *response* of that request is a finite play w starting in s that allows an embedding of $\mathcal{P}_j$. If w does not allow an embedding, then the request is still *open* after w. This notion can be refined, since a request is responded by a chain of vertices: we say that $D \subseteq D_j$ is *open* after the finite play $\rho_0 \ldots \rho_n$, if there was a request of condition j at position $k \leq n$ such that the elements in $D_j \backslash D$ can be embedded in $\rho_k \ldots \rho_n$, but no superset of $D_j \backslash D$ can be embedded in this suffix. This means Player 0 was able to embed the elements of $D_j \backslash D$ (which form a downwards-closed subset by the requirements on an embedding) and the elements of D (which form an upwards-closed subset) have to be embedded yet.

Example 2. Let $D_j = \{d_1, d_2, d_3\}$, $\preceq_j$ be specified by $d_1 \preceq_j^{red} d_2 \preceq_j^{red} d_3$, and $l_j(d) = d$ for all $d \in D_j$. Finally, let $(q_j, (D_j, \preceq_j, l_j))$ be a condition and ρ as in Figure 2. The positions of ρ are denoted above and the sets below are the labels of each position. The vertices ρ_0, ρ_3, ρ_6, and ρ_7 are requests, D_j is open after the finite play $\rho_0\rho_1$, and there are three open requests after $\rho_0 \ldots \rho_6$: The set

D_j from the request at ρ_6, $\{d_2, d_3\}$ from the request at ρ_3, and $\{d_3\}$ from the request at ρ_0. All requests are responded completely after ρ_9.

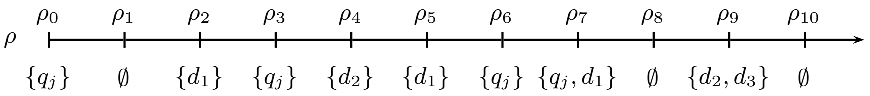

Fig. 2. A play with (open) requests

To define the waiting times we need to keep track of the open requests. For $j \in [k]$, $D \subseteq D_j$ and $s \in V$ let

$$\mathrm{New}_j(s) = \begin{cases} D_j & \text{if } q_j \in l_G(s) \\ \emptyset & \text{otherwise} \end{cases}$$

and

$$\mathrm{Emb}_j(D, s) = \{d \in D \mid \exists d' \in D : d' \preceq_j d \text{ and } l_j(d') \notin l_G(s)\} \ .$$

The set $\mathrm{Emb}_j(D, s)$ contains the elements of D that cannot be embedded into s since a smaller element's labeling is not compatible with the labeling of s. The *set of open requests of condition j after the finite play w* is defined inductively by $\mathrm{Open}_j(\varepsilon) = \emptyset$ and

$$\mathrm{Open}_j(ws) = \{(\mathrm{Emb}_j(D, s), t+1) \mid (D, t) \in \mathrm{Open}_j(w) \cup \{(\mathrm{New}_j(s), 0)\}\} \setminus \{\emptyset\} \times \mathbb{N}.$$

That is, Open_j deletes all those elements from the open requests D in $\mathrm{Open}_j(w)$ that can be embedded into s, adds a tick to the clock t of every request, checks for new requests, and deletes responded requests. If $(D, t+1) \in \mathrm{Open}_j(\rho_0 \dots \rho_n)$, then there was a request of condition j at position $n - t$, the elements of $D_j \setminus D$ can be embedded into $\rho_{n-t} \dots \rho_n$, and Player 0 has to embed all elements of D in the future to respond to this request.

Example 3. Consider the play ρ in Figure 2 with condition $(q_j, (D_j, \preceq_j, l_j))$ as in Example 2. We have $\mathrm{Open}_j(\rho_0 \dots \rho_3) = \{(\{d_2, d_3\}, 4), (D_j, 1)\}$, $\mathrm{Open}_j(\rho_0 \dots \rho_6) = \{(\{d_3\}, 7), (\{d_2, d_3\}, 4), (D_j, 1,)\}$ and $\mathrm{Open}_j(\rho_0 \dots \rho_9) = \emptyset$.

Note that $\mathrm{Open}_j(w)$ contains only upwards-closed subsets of $\mathcal{P}_j$. The *number of open requests $D \in \mathrm{Up}(\mathcal{P}_j)$ of condition j after w* is

$$s_{j,D}(w) = |\{t \mid (D, t) \in \mathrm{Open}_j(w)\}| \ .$$

A set $D \in \mathrm{Up}(\mathcal{P}_j)$ is *open indefinitely* in $\rho_0\rho_1\rho_2 \dots$, if there exists a position n such that $(D, t) \in \mathrm{Open}_j(\rho_0 \dots \rho_{n+t})$ for all $t > 1$.

Lemma 1. *Let $\rho = \rho_0\rho_1\rho_2 \dots$ be a play. For all $j \in [k]$:*

(i) If Player 0 wins ρ, then $(\mathrm{Open}_j(\rho_0 \dots \rho_n))_{n \in \mathbb{N}}$ induces a minimal embedding f_m of $\mathcal{P}_j$ in $\rho_m\rho_{m+1}\rho_{m+2} \dots$ for every position m such that $q_j \in l_G(\rho_m)$.

(ii) ρ is won by Player 0 iff there is no $D \in \mathrm{Up}(\mathcal{P}_j)$ that is open indefinitely.

Proof. We begin by defining the sequence $(D_{m,t})_{t\in\mathbb{N}}$ for every request of condition j at position m by $D_{m,0} = \mathrm{Emb}_j(D_j, \rho_m)$ and $D_{m,t+1} = \mathrm{Emb}_j(D_{m,t}, \rho_{m+t+1})$. We have $D_{m,t} \supseteq D_{m,t+1}$ and

$$\mathrm{Open}_j(\rho_0 \dots \rho_n) = \{(D_{m,t}, t+1) \mid m+t = n,\ q_j \in l_G(\rho_m),\ \text{and } D_{m,t} \neq \emptyset\}\ ,$$

which can be verified by an easy induction.

(i) Let $q_j \in l_G(\rho_m)$ and let f be a minimal embedding of $\mathcal{P}_j$ in the suffix $\rho_m\rho_{m+1}\rho_{m+2}\dots$, whose existence is guaranteed, since ρ is won by Player 0. We define the embedding f_m by $f_m(d) = \min\{t \mid d \notin D_{m,t}\}$. We show that f_m is well-defined and minimal by proving $d \notin D_{m,f(d)}$ by Noetherian induction (on $\preceq_j^{red}$) over d. Let $d \in D_j$. Towards a contradiction, assume $d \in D_{m,f(d)}$.

If $f(d) = 0$, then $d \in D_{m,0} = \mathrm{Emb}_j(D_j, \rho_m)$, i.e., there exists a $d' \in D_j$ such that $d' \preceq_j d$ and $l_j(d') \notin l_G(\rho_m)$. Since $d' \preceq_j d$ implies $f(d') = 0$, we conclude $l_j(d') \in l_G(\rho_{m+f(d')})$, which yields the desired contradiction.

If $f(d) = t+1$, then $d \in D_{m,t+1} = \mathrm{Emb}_j(D_{m,t}, \rho_{m+t+1})$, i.e., there exists a $d' \in D_{m,t}$ such that $d' \preceq_j d$ and $l_j(d') \notin l_G(\rho_{m+t+1})$. Since f is an embedding, $l_j(d) \in l_G(\rho_{m+f(d)})$, which rules out $d' = d$. Thus, we can assume $d' \preceq_j^{red} d$ without loss of generality, which allows us to apply the induction hypothesis to obtain $d' \notin D_{m,f(d')}$. If $f(d') < t+1$, then $d' \notin D_{m,t}$, since $D_{m,f(d')} \supseteq D_{m,t}$. On the other hand, $f(d') = t+1$ implies $l_j(d') \in l_G(\rho_{m+t+1})$. So, both cases yield the desired contradiction.

(ii) In (i) we have seen that $(D_{m,t})_{t\in\mathbb{N}}$ converges to the empty set for every m, if Player 0 wins ρ. Thus, there cannot be a $D \in \mathrm{Up}(\mathcal{P}_j)$ that is open indefinitely.

On the other hand, if there is no $D \in \mathrm{Up}(\mathcal{P}_j)$ that is open indefinitely, then $(D_{m,t})_{t\in\mathbb{N}}$ converges to the empty set for every m. We define an embedding f_m for every m such that $q \in l_G(\rho_m)$ by $f_m(d) = \min\{t \mid d \notin D_{m,t}\}$. It remains to show that f_m is an embedding: the element d leaves $(D_{m,t})_{t\in\mathbb{N}}$ at $f_m(d)$ iff the labeling requirement is fulfilled, by definition of Emb_j. Also,

$$\min\{t \mid d' \notin D_{m,t}\} = f_m(d') > f_m(d) = \min\{t \mid d \notin D_{m,t}\}$$

for $d' \preceq_j d$ contradicts the definition of Emb_j. □

Waiting times for poset games are defined employing the information given by the open requests in $\mathrm{Open}_j(w)$. Define the

- *totalized waiting time for $D \in \mathrm{Up}(\mathcal{P}_j)$ after w*: $t_{j,D}(w) = \sum_{(D,t)\in\mathrm{Open}_j(w)} t$,
- *penalty after w*: $p(w) = \sum_{j=1}^{k} \sum_{D\in\mathrm{Up}(\mathcal{P}_j)} t_{j,D}(w)$,
- *value of a play ρ*: $v(\rho) = \limsup_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} p(\rho_0 \dots \rho_i)$,
- *value of a strategy σ*: $v(\sigma) = \sup\{v(\rho(\sigma,\tau)) \mid \tau \text{ strategy for Player 1}\}$.

Hence, the influence of an open request on the value of a play grows quadratically in the waiting time, which penalizes longer waiting times more severely. A strategy σ for Player 0 is *optimal* if $v(\sigma) \leq v(\sigma')$ for all strategies σ' for Player 0.

A simple calculation proves the following fact about the evolution of the waiting times and the number of open requests.

Lemma 2. *Let $x, y \in V^*$ and $s \in V$ such that $t_{j,D}(x) \leq t_{j,D}(y)$ and $s_{j,D}(x) \leq s_{j,D}(y)$ for all $j \in [k]$ and all $D \in \mathrm{Up}(\mathcal{P}_j)$. Then, $t_{j,D}(xs) \leq t_{j,D}(ys)$ and $s_{j,D}(xs) \leq s_{j,D}(ys)$ for all $j \in [k]$ and all $D \in \mathrm{Up}(\mathcal{P}_j)$.*

Let σ be a strategy for Player 0 and $D \in \mathrm{Up}(\mathcal{P}_j)$ for some condition j. We say that σ *uniformly bounds the waiting time for D to b*, if for all finite plays w consistent with σ it holds that $t \leq b$ for all $(D, t) \in \mathrm{Open}_j(w)$. Analogously, σ *uniformly bounds the totalized waiting time for D to b*, if $t_{j,D}(w) \leq b$ for all finite plays w consistent with σ. If the (totalized) waiting time for all $D \in \mathrm{Up}(\mathcal{P}_j)$ is bounded, then the length of the embeddings that respond to a request of condition j is also bounded.

Remark 2. Let σ be a strategy for Player 0. If σ uniformly bounds the waiting time for D to b, then σ also uniformly bounds the totalized waiting time for D to $\frac{1}{2}b(b+1)$.

The following lemma is a simple consequence of Lemma 1 (ii).

Lemma 3. *Let ρ be a play and σ a strategy for Player 0.*

(i) If $v(\rho) < \infty$, then Player 0 wins ρ.
(ii) If $v(\sigma) < \infty$, then σ is a winning strategy for Player 0.

Note that the other directions of the statements are false: there are plays of infinite value that are won by Player 0.

4 Solving Poset Games

In this section, poset games are reduced to Büchi games. The memory stores the elements of the posets $\mathcal{P}_j$ that still have to be embedded. A cyclic counter ensures that all requests are responded by an embedding eventually.

Theorem 1. *Poset games are reducible to Büchi games and therefore determined with finite-state strategies.*

Proof. Let $h = \sum_{j=1}^{k} |D_j|$ and fix an enumeration $e : [h] \to \bigcup_{j=1}^{k} \{j\} \times D_j$. We assume $h > 1$ (without loss of generality) to obtain a nontrivial counter. The memory structure $\mathfrak{M} = (M, m_0, \mathrm{update})$ consists of $M = \prod_{j=1}^{k} \mathrm{Up}(\mathcal{P}_j) \times [h] \times \{0,1\}$, $m_0 = (\mathrm{Emb}_1(\mathrm{New}_1(s_0), s_0), \ldots, \mathrm{Emb}_k(\mathrm{New}_k(s_0), s_0), 1, 0)$, and we define $\mathrm{update}((O_1, \ldots, O_k, m, f), s) = (O'_1, \ldots, O'_k, m', f')$ with

- $O'_j = \begin{cases} \mathrm{Emb}_j(D_j, s) & \text{if } q_j \in l_G(s) \\ \mathrm{Emb}_j(O_j, s) & \text{if } q_j \notin l_G(s) \end{cases},$
- $m' = \begin{cases} (m \mod h) + 1 & \text{if } e(m) = (j, d) \text{ and } d \notin O'_j \text{ or } l_j(d) \in l_G(s) \\ m & \text{if } e(m) = (j, d) \text{ and } d \in O'_j \text{ and } l_j(d) \notin l_G(s) \end{cases},$
- $f' = \begin{cases} 1 & \text{if } m \neq m' \\ 0 & \text{otherwise} \end{cases}.$

Finally, let $F = V \times \prod_{j=1}^{k} \mathrm{Up}(\mathcal{P}_j) \times [h] \times \{1\}$ and let $\mathcal{G}' = (G \times \mathfrak{M}, F)$ be a Büchi game in the expanded arena. It remains to verify $\mathcal{G} \leq_{\mathfrak{M}} \mathcal{G}'$. Then, positional determinacy of Büchi games [3] and Remark 1 finish the proof.

Let $\rho = \rho_0\rho_1\rho_2 \dots$ be a play of $\mathcal{G}$ and $\rho' = \rho'_0\rho'_1\rho'_2 \dots$ the unique expanded play in $\mathcal{G}'$ where $\rho'_n = (\rho_n, (O^n_1, \dots, O^n_k, m^n, f^n))$.

Let ρ be winning for Player 0 and assume towards a contradiction that ρ' is winning for Player 1. Then, the counter stops at some position n' with some value c and does not change anymore. This means $d \in O^n_j$ and $l_j(d) \notin l_G(\rho_n)$ for all $n \geq n'$, where $e(c) = (j, d)$. If condition j is requested infinitely often in ρ, then $\mathcal{P}_j$ is embedded infinitely often in ρ. Thus, there are infinitely many n such that $l_j(d) \in l_G(\rho_n)$, which yields the desired contradiction. On the other hand, if there is a final request at position n, then there is also an embedding f of $\mathcal{P}_j$ in $\rho_n\rho_{n+1}\rho_{n+2} \dots$. It holds $O^{n+t}_j = \{d \in D_j \mid f(d) > t\}$. Thus, the O_j-component is empty from some position onwards, which again is a contradiction.

Now, let ρ' be winning for Player 0. For every n such that $q_j \in l_G(\rho_n)$, we have to construct an embedding f_n of $\mathcal{P}_j$ in $\rho_n\rho_{n+1}\rho_{n+2} \dots$. Since ρ' is won by Player 0, there are infinitely many positions n' such that $d \notin O^{n'}_j$ or $l_j(d) \in l_G(\rho_{n'})$.

Let $q_j \in l_G(\rho_n)$. We define the sequence $(D_{n,t})_{t \in \mathbb{N}}$ by $D_{n,0} = \mathrm{Emb}_j(D_j, \rho_n)$ and $D_{n,t+1} = \mathrm{Emb}_j(D_{n,t}, \rho_{n+t+1})$. We have $D_{n,t} \subseteq O^{n+t}_j$ for all t, which can be verified by an easy induction. Also, $D_{n,t} \supseteq D_{n,t+1}$ for all t. Now, if $D_{n,t} = \emptyset$ for some t, then the sequence induces f_n by $f_n(d) = \min\{t \mid d \notin D_{n,t}\}$ as we have shown in the proof of Lemma 1 (ii). Hence, it remains to be shown that for every $d \in D_j$ there is a t such that $d \notin D_{n,t}$.

Towards a contradiction, assume there exists t such that $\emptyset \neq D_{n,t} = D_{n,t'}$ for all $t' \geq t$, and let d be minimal in $D_{n,t}$, i.e., there is no $d' \neq d$ such that $d' \preceq_j d$ and $d' \in D_{n,t}$. Thus, $d \in O^{n+t'}_j$ for all $t' \geq t$. Since ρ' is winning for Player 0, there is some $t' > t$ such that $l_j(d) \in l_G(\rho_{n+t'})$, which implies that d can be embedded in $\rho_{n+t'}$, i.e., we have $d \notin \mathrm{Emb}_j(D_{n,t'-1}, \rho_{n+t'}) = D_{n,t'}$, which yields the desired contradiction. $\square$

If e is defined such that the elements of each domain D_j are enumerated consecutively and such that $d \preceq_j d'$ implies $e^{-1}(j, d) \leq e^{-1}(j, d')$, then it takes at most $h + |D_j|$ visits to vertices in F after a request of condition j to complete an embedding of $\mathcal{P}_j$ in the projected play.

The size of $\mathfrak{M}$ can be bounded by $|M| \leq h \cdot 2^{h+1}$, which is asymptotically optimal. This can be shown by transforming the family of request-response games presented in Theorem 2 of [6] into poset games.

Theorem 1 implies an upper bound on the value of an optimal strategy.

Corollary 1. *Let $h = \sum_{j=1}^{k} |D_j|$. If Player 0 wins $\mathcal{G}$, then she also has a winning strategy σ with*

$$v(\sigma) \leq \sum_{j=1}^{k} \left(c_j \cdot \frac{|G| \cdot (h + |D_j|) \cdot (|G| \cdot (h + |D_j|) + 1)}{2} \right) =: b_{\mathcal{G}} \enspace .$$

Proof. Let σ' be the positional winning strategy for $\mathcal{G}'$ from Theorem 1 and let σ be the induced finite-state strategy for $\mathcal{G}$. There is no infix of length $|G|$ of a play ρ' played according to σ' that does not visit F at least once. If there is such an infix, then there is a loop in that infix in which no vertex of F is visited. Moving through that loop indefinitely is consistent with σ'. Thus, it is not a winning strategy for Player 0, contrary to our assumptions. Therefore, the counter m

changes its value after at most $|G|$ steps. We remarked that the counter can be constructed such that it takes at most $h+|D_j|$ visits to a state in F to complete an embedding after a request in the projected play. Hence, the length of every embedding of $\mathcal{P}_j$ in a play consistent with σ is bounded by $|G|\cdot(h+|D_j|)$, which gives

$$t_{j,D}(w) \leq \frac{|G|\cdot(h+|D_j|)\cdot(|G|\cdot(h+|D_j|)+1)}{2}$$

for every finite play w consistent with σ, by Remark 2. We obtain

$$\begin{aligned}
\frac{1}{n}\sum_{i=0}^{n-1} p(\rho_0\dots\rho_i) &= \frac{1}{n}\sum_{i=1}^{n-1}\sum_{j=1}^{k}\sum_{D\in\mathrm{Up}(\mathcal{P}_j)} t_{j,D}(\rho_0\dots\rho_i)\\
\leq &\frac{1}{n}\sum_{i=0}^{n-1}\sum_{j=1}^{k}\sum_{D\in\mathrm{Up}(\mathcal{P}_j)} \frac{|G|\cdot(h+|D_j|)\cdot(|G|\cdot(h+|D_j|)+1)}{2}\\
=&\sum_{j=1}^{k}\left(c_j\cdot\frac{|G|\cdot(h+|D_j|)\cdot(|G|\cdot(h+|D_j|)+1)}{2}\right) = b_{\mathcal{G}}
\end{aligned}$$

for every play $\rho = \rho_0\rho_1\rho_2\dots$ that is played according to σ. Hence,

$$v(\rho) = \lim_{n\to\infty}\frac{1}{n}\sum_{i=0}^{n-1} p(\rho_0\dots\rho_i) \leq b_{\mathcal{G}}$$

for every play ρ consistent with σ, which gives $v(\sigma)\leq b_{\mathcal{G}}$. □

5 Time-optimal Strategies for Poset Games

We are now able to state the main theorem of this paper, which will be proved in the remainder of this section.

Theorem 2. *If Player* 0 *wins a poset game* $\mathcal{G}$*, then she also has an optimal winning strategy which is finite-state and effectively computable. The value of an optimal strategy is effectively computable as well.*

5.1 Strategy Improvement for Poset Games

We begin by defining a *strategy improvement operator* $I_{j,D}$ for every $D\in\mathrm{Up}(\mathcal{P}_j)$. It deletes loops of plays, consistent with the given strategy, that are spent waiting for a position into which an element from D has been embedded. Consider the embedding f in Figure 3: loops in the interval between the request q and the position $f(d_1)$ and the intervals between $f(d_1)$ and $f(d_2)$ respectively between $f(d_2)$ and $f(d_3)$ can be deleted to shorten the waiting times.

Hence, the intervals in which D is an open request will be shorter if Player 0 plays according to the improved strategy. Doing this repeatedly will uniformly bound the waiting time $t_{j,D}$. However, the improved strategy has to ensure that no other responses get incomplete by deleting loops, i.e., the improved strategy is still winning for Player 0. Also, we do not want the value of the improved strategy

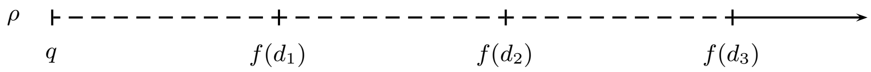

Fig. 3. An request q and the corresponding embedding. The intervals in which Player 0 can improve her strategy are dashed

to be greater than the value of the original strategy. We begin by defining the operator and then prove that it has the desired properties. Afterwards we show how to obtain uniform bounds on the waiting time by applying each $I_{j,D}$ infinitely often.

Let σ be a winning strategy (not necessarily finite-state) for Player 0 such that $v(\sigma) \leq b_{\mathcal{G}}$. The strategy $I_{j,D}(\sigma)$ is implemented with memory structure $\mathfrak{M} = (M, m_0, \mathrm{update})$ where M is a subset of the finite plays consistent with σ and defined implicitly. The initial memory state is $m_0 = s_0$ and $\mathrm{update}(w, s)$ is defined by a case distinction:

Player 0 only skips loops if the totalized waiting time for D is guaranteed to be higher than the value of the strategy, i.e., at least $b_{\mathcal{G}}$. Then, the value of the play does not increase from taking a shortcut. Thus, if $t_{j,D}(ws) \leq b_{\mathcal{G}}$, let $\mathrm{update}(w, s) = ws$. Hence, if the totalized waiting time is small, then she copies the original play according to σ.

Otherwise, if $t_{j,D}(ws) > b_{\mathcal{G}}$ consider the tree $\mathfrak{T}^{\sigma}_{ws}$ containing all finite continuations of ws that are consistent with σ restricted to those paths wsx such that $\mathrm{Open}_j(wsx') \cap (\{D\} \times \mathbb{N}) \neq \emptyset$ for all $x' \sqsubseteq x$. This tree contains all continuations of ws up to the point where the first element of the open request D can be embedded into. This tree is finite since σ is a winning strategy. The set of finite plays zs of $\mathfrak{T}^{\sigma}_{ws}$ such that $t_{j',D'}(zs) \geq t_{j',D'}(ws)$ and $s_{j',D'}(zs) \geq s_{j',D'}(ws)$ for all $j' \in [k]$ and all $D' \in \mathrm{Up}(\mathcal{P}_{j'})$ is non-empty as it contains ws. Let x be a play of maximal length in that set. Then, $\mathrm{update}(ws) = x$. So, if the totalized waiting time for D is sufficiently high, then Player 0 looks ahead whether ws is the start of a loop such that the totalized waiting times and the number of open requests for all $j' \in [k]$ and all $D' \in \mathrm{Up}(\mathcal{P}_{j'})$ are higher at the end of the loop than they were at the beginning. Then, she jumps ahead (by updating the memory to x) and continues to play as if she had finished the loop already.

The condition on $t_{j',D'}$ ensures that she does not miss a vertex that she has to visit in order to embed an element of the posets. This ensures that the improved strategy is still winning for Player 0. The condition on $s_{j',D'}$ guarantees that the value of the play does not increase from taking a shortcut by jumping ahead to a position where more requests will be open than before.

Finally, define $\mathrm{next}(s, ws) = \sigma(ws)$. Thus, Player 0's choice of the next move assumes that she has already finished the loops which were skipped by the memory update. The improved strategy $I_{j,D}(\sigma)$ is now given by $\mathfrak{M}$ and next.

Lemma 4. *Let σ be a winning strategy for Player 0 such that $v(\sigma) \leq b_{\mathcal{G}}$, $j \in [k]$, and $D \in \mathrm{Up}(\mathcal{P}_j)$.*

(i) $\mathrm{update}^*(w)$ *is consistent with σ for all w consistent with $I_{j,D}(\sigma)$.*

(ii) $t_{j',D'}(w) \leq t_{j',D'}(\text{update}^*(w))$ *and* $s_{j',D'}(w) \leq s_{j',D'}(\text{update}^*(w))$ *for all* w *consistent with* $I_{j,D}(\sigma)$*, for all* $j' \in [k]$ *and all* $D' \in \text{Up}(\mathcal{P}_{j'})$*.*

(iii) If σ *bounds the totalized waiting time for some* $D' \in \text{Up}(\mathcal{P}_{j'})$ *to* b*, then so does* $I_{j,D}(\sigma)$*.*

(iv) $v(I_{j,D}(\sigma)) \leq v(\sigma)$*.*

(v) $I_{j,D}(\sigma)$ *is a winning strategy for Player* 0*.*

Proof. (i) By induction over w: every play starts in s_0, so the induction base is trivial, as $\text{update}^*(s_0) = s_0$ holds. For the induction step, let $w = \rho_0 \ldots \rho_n$ be played according to $I_{j,D}(\sigma)$. Applying the induction hypothesis, we can assume that $\text{update}^*(\rho_0 \ldots \rho_{n-1})$ is consistent with σ. By definition of update^* we have

$$\text{update}^*(\rho_0 \ldots \rho_n) = \text{update}(\text{update}^*(\rho_0 \ldots \rho_{n-1}), \rho_n) \ .$$

Furthermore, the last vertex of $\text{update}^*(\rho_0 \ldots \rho_{n-1})$ is ρ_{n-1}. Finally, if $\rho_{n-1} \in V_0$, then

$$\begin{aligned} \rho_n &= I_{j,D}(\sigma)(\rho_0 \ldots \rho_{n-1}) \\ &= \text{next}(\rho_{n-1}, \text{update}^*(\rho_0 \ldots \rho_{n-1})) \\ &= \sigma(\text{update}^*(\rho_0 \ldots \rho_{n-1})). \end{aligned} \tag{1}$$

Analogously to the definition, we consider two cases: either, we have

$$\text{update}(\text{update}^*(\rho_0 \ldots \rho_{n-1}), \rho_n) = \text{update}^*(\rho_0 \ldots \rho_{n-1})\rho_n \ .$$

Then, by induction hypothesis and (1), $\text{update}^*(\rho_0 \ldots \rho_n)$ is consistent with σ_{j-1}. Otherwise, in the second case of the definition, we have

$$\text{update}(\text{update}^*(\rho_0 \ldots \rho_{n-1}), \rho_n) = \text{update}^*(\rho_0 \ldots \rho_{n-1})\rho_n z' \ ,$$

where z' is a path in $\mathfrak{T}^{\sigma}_{\text{update}^*(\rho_0 \ldots \rho_{n-1})\rho_n}$. Together with the induction hypothesis and (1), this shows that $\text{update}^*(\rho_0 \ldots \rho_n)$ is consistent with σ.

(ii) The induction base is clear as every play starts in s_0 and we have $s_0 = m_0 = \text{update}^*(s_0)$. By the induction hypothesis, we can assume

$$t_{j',D'}(w) \leq t_{j',D'}(\text{update}^*(w)) \text{ and } s_{j',D'}(w) \leq s_{j',D'}(\text{update}^*(w))$$

for all $j' \in [k]$ and all $D' \in \text{Up}(\mathcal{P}_{j'})$. Furthermore, we have

$$t_{j',D'}(ws) \leq t_{j',D'}(\text{update}^*(w)s) \text{ and } s_{j',D'}(ws) \leq s_{j',D'}(\text{update}^*(w)s)$$

by Remark 2. There are two possibilities for $\text{update}^*(ws)$. If $t_{j,D}(\text{update}^*(w)s) \leq b_{\mathcal{G}}$, then

$$\text{update}^*(ws) = \text{update}(\text{update}^*(w), s) = \text{update}^*(w)s \ .$$

Thus,

$$t_{j',D'}(ws) \leq t_{j',D'}(\text{update}^*(w)s) = t_{j',D'}(\text{update}^*(ws))$$

and

$$s_{j',D'}(ws) \leq s_{j',D'}(\text{update}^*(w)s) = s_{j',D'}(\text{update}^*(ws)) \ .$$

If $t_{j,D}(\text{update}^*(w)s) > b_{\mathcal{G}}$, then

$$\text{update}^*(ws) = \text{update}(\text{update}^*(w), s) = zs$$

where $t_{j',D'}(zs) \geq t_{j',D'}(\text{update}^*(w)s)$ and $s_{j',D'}(zs) \geq s_{j',D'}(\text{update}^*(w)s)$ by definition of $I_{j,D}$. Hence,

$$t_{j',D'}(ws) \leq t_{j',D'}(\text{update}^*(w)s) \leq t_{j',D'}(zs) = t_{j',D'}(\text{update}^*(ws))$$

and

$$s_{j',D'}(ws) \leq s_{j',D'}(\text{update}^*(w)s) \leq s_{j',D'}(zs) = s_{j',D'}(\text{update}^*(ws))$$

for all $j' \in [k]$ and all $D' \in \text{Up}(\mathcal{P}_{j'})$.

(iii) There is a bound b such that $t_{j',D'}(w') \leq b$ for all finite plays w' consistent with σ. Now, let w be a play consistent with $I_{j,D}(\sigma)$. Then, update(w) is a prefix of a play according to σ. Hence, $t_{j',D'}(w) \leq t_{j',D'}(\text{update}(w)) \leq b$. Thus, $I_{j,D}(\sigma)$ uniformly bounds the totalized waiting time for D' to b.

(iv) For a play $\rho = \rho_0\rho_1\rho_2\ldots$, let $\text{update}^*(\rho) = \lim_{n\to\infty} \text{update}^*(\rho_0 \ldots \rho_n)$. The limit is consistent with σ for every play ρ consistent with $I_{j,D}(\sigma)$. We show $v(\rho) \leq v(\text{update}^*(\rho))$ for all ρ consistent with $I_{j,D}(\sigma)$, which implies the claim. To this end, we define

$$S = \{w' \sqsubseteq \text{update}^*(\rho) \mid \neg\exists w \sqsubseteq \rho : \text{update}^*(w) = w'\} \ .$$

S contains exactly the vertices of the loops skipped by Player 0. Let $w' \in S$. Then, $t_{j,D}(w') > b_{\mathcal{G}}$ holds, as every improvement step only deletes loops of update$^*(\rho)$ if the totalized waiting time is higher than $b_{\mathcal{G}}$. Thus, $p(w') > b_{\mathcal{G}} \geq v(\sigma)$ and

$$\limsup_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} p(\text{update}^*(\rho_0 \ldots \rho_i)) \leq v(\text{update}^*(\rho)) \ , \tag{2}$$

since the average decreases if the summation omits the summands for the prefixes in S. Now, let $w \sqsubseteq \rho$: We have $t_{j',D'}(w) \leq t_{j',D'}(\text{update}^*(w))$ for all $j' \in [k]$ and all $D' \in \text{Up}(\mathcal{P}_{j'})$, and therefore $p(w) \leq p(\text{update}^*(w))$. Thus,

$$\frac{1}{n} \sum_{i=0}^{n-1} p(\rho_0 \ldots \rho_i) \leq \frac{1}{n} \sum_{i=0}^{n-1} p(\text{update}^*(\rho_0 \ldots \rho_i)) \ .$$

The latter term converges to a value less than or equal to $v(\text{update}^*(\rho))$, by (2). Thus, we conclude $v(\rho) \leq v(\text{update}^*(\rho))$.

(v) We have $v(I_{j,D}(\sigma)) \leq v(\sigma) \leq b_{\mathcal{G}}$. Now apply Lemma 3. □

In order to obtain small bounds on the waiting times, each improvement operator $I_{j,D}$ is now applied infinitely often to a given initial winning strategy. The limit of the strategies improved with $I_{j,D}$ uniformly bounds the totalized waiting time for D. Furthermore, all properties stated in Lemma 4 can be lifted to the limit strategy as well.

The order of improvement is given by enumerations $e_j : [c_j] \to \text{Up}(\mathcal{P}_j)$ such that $|D| > |D'|$ implies $e_j^{-1}(D) < e_j^{-1}(D')$. Thus, the subsets are enumerated in order of decreasing size.

Example 4. Consider the play ρ, depicted in Figure 4, of a game with a single condition $(q_j, (D_j, \preceq_j))$ where $D = \{d_1, d_2, d_3\}$ and $d_1 \preceq_j^{red} d_2 \preceq_j^{red} d_3$ (again, we ignore the labeling functions). The improvement scheme to be defined starts with the improvement with respect to D, then with respect to $\{d_2, d_3\}$ and finally with respect to $\{d_3\}$.

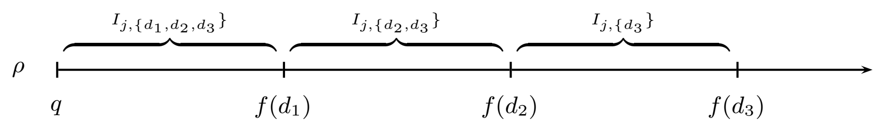

Fig. 4. The order of improvement with $I_{j,D}$ for $D \subseteq \{d_1, d_2, d_3\}$

Given a winning strategy σ_0 for Player 0 such that $v(\sigma_0) \leq b_{\mathcal{G}}$ (whose existence is guaranteed by Corollary 1), define

- $\sigma_{j,l,0} = \begin{cases} \sigma_{j-1} & \text{if } l = 1 \\ \sigma_{j,l-1} & \text{otherwise} \end{cases}$ for $j \in [k]$ and $l \in [c_j]$,
- $\sigma_{j,l,n+1} = I_{j,e_j(l)}(\sigma_{j,l,n})$ for $j \in [k]$, $l \in [c_j]$, and $n \in \mathbb{N}$,
- $\sigma_{j,l} = \lim_{n\to\infty} \sigma_{j,l,n}$ for $j \in [k]$ and $l \in [c_j]$, and
- $\sigma_j = \sigma_{j,c_j}$ for $j \in [k]$.

For notational convenience, let $\sigma_{j,0} = \sigma_{j-1}$ for $j \in [k]$. Also, let $\mathrm{update}_{j,l,n}$ be the update function and $\mathrm{next}_{j,l,n}$ be the next-move function used to define $\sigma_{j,l,n}$.

Before we discuss the properties of the strategies defined above, we need to introduce some additional notation that is used to bound the waiting times.

The strategy improvement operator $I_{j,D}$ skips a loop if the vertices at the beginning and at the end coincide and the values $s_{j',D'}$ and $t_{j',D'}$ at the end are greater than or equal to the values at the beginning. Hence, we say that two finite plays $y_1 \sqsubset y_2$ form a *Dickson pair* [1] if their last vertices coincide and $s_{j',D'}(y_1) \leq s_{j',D'}(y_2)$ and $t_{j',D'}(y_1) \leq t_{j',D'}(y_2)$ for all $j' \in [k]$ and all $D' \in \mathrm{Up}(\mathcal{P}_{j'})$. Dickson pairs are candidates for deletion by $I_{j,D}$.

The set D is in Open_j throughout a loop skipped by $I_{j,D}$. Accordingly, an infix $\rho_m \ldots \rho_{m+n}$ of a play ρ is called *non-Dickson save D* if $t_{j,D}$ increases strictly monotonic throughout the infix and if there are no $m \leq g < h \leq m + n$ such that $\rho_0 \ldots \rho_g$ and $\rho_0 \ldots \rho_h$ are a Dickson pair. The length of such an infix can be bounded inductively by a function b in the size n of G and in $c = \sum_{j=1}^{k} |\mathrm{Up}(\mathcal{P}_j)|$.

If $c = 1$, then the single set is $D \in \bigcup_{j\in[k]} \mathrm{Up}(\mathcal{P}_j)$, whose values increase monotonically. Hence, there is a vertex repetition after at most $|G|$ steps. Therefore, $b(n, 1) = n$.

If $c + 1 > 1$, then $t_{j',D'}$ (and thereby also $s_{j',D'}$) has to be reset to 0 after at most $b(n, c)$ steps for every $D' \in \bigcup_{j\in[k]} \mathrm{Up}(\mathcal{P}_j) \setminus \{D\}$. If not, then the initial prefix of length $b(n, c)$ contains a Dickson pair by induction hypothesis. For the same reason, for every $c' \in [c]$ there are c' sets D' such that $t_{j',D'}$ (and also $s_{j',D'}$) was reset to 0 in the last $b(n, c')$ steps. If not, then this infix would again contain a

Dickson pair by induction hypothesis. Accounting for all possible combinations, we obtain

$$b(n, c+1) = b(n, c) + nc! \prod_{j=1}^{c} \frac{1}{2}(b(n, j))^2(b(n, j) + 1) \ ,$$

as we have $t_{j',D'}(xy) \leq \frac{1}{2}|y|(|y|+1)$ and $s_{j',D'}(xy) \leq |y|$ if $t_{j',D'}(x) = 0$.

Note that the same idea can be applied to request-response games, which lowers the bounds given in [4, 5].

Now, we are able to lift the properties of the strategy improvement operator to the limit of the improved strategies and to bound the waiting times.

Lemma 5. *Let $j \in [k]$, $l \in [c_j]$, and $e_j(l) = D$. Then:*

(i) $\lim_{n\to\infty} \sigma_{j,l,n}$ *exists.*

(ii) If $\sigma_{j,l-1}$ uniformly bounds the totalized waiting time for some $D' \in \mathrm{Up}(\mathcal{P}_{j'})$, then so does $\sigma_{j,l}$.

(iii) $v(\sigma_{j,l}) \leq v(\sigma_{j,l-1})$, *and therefore* $v(\sigma_j) \leq v(\sigma_{j-1})$.

(iv) $\sigma_{j,l}$ *uniformly bounds the waiting time for D to*

$$b_{j,D} := b_{\mathcal{G}} + (|D_j \backslash D| + 1) \cdot b\,(|G|, c)$$

Proof. (i) We begin by proving that $(\mathrm{update}^*_{j,l,n})_{n\in\mathbb{N}}$ converges to the identity function, proceeding by induction over w: we have $\mathrm{update}^*_{j,l,n}(s_0) = s_0$ for all n. Now, assume $\mathrm{update}^*_{j,l,n}(w) = w$ for all $n \geq n_w$. If $t_{j,D}(ws) \leq b_{\mathcal{G}}$, then $\mathrm{update}_{j,l,n}(w, s) = ws$ and therefore also $\mathrm{update}^*_{j,l,n}(ws) = ws$ for all $n \geq n_w$.

Thus, let $t_j(ws) > b_{\mathcal{G}}$ and let $\mathfrak{T}_n$ be $\mathfrak{T}_{ws}^{\sigma_{j,l,n-1}}$ restricted to the maximal paths that continuously contain D in their set of open requests. By definition, $\mathrm{update}^*_{j,l,n}(ws)$ is a vertex of $\mathfrak{T}_n$. Since every path in $\mathfrak{T}_n$ is a path in $\mathfrak{T}_{n-1}$ from which some loops might be deleted, the size of the $\mathfrak{T}_n$ is decreasing. Finally, if $\mathfrak{T}_n = \mathfrak{T}_{n+1}$, then $\mathfrak{T}_{n'} = \mathfrak{T}_n$ for all $n' \geq n$. Thus, there is an index $n_{ws} \geq n_w$ such that the $\mathfrak{T}_n$ are equal for all $n \geq n_{ws}$. From that index on we have $\mathrm{update}_{j,l,n}(w, s) = ws$ and thus $\mathrm{update}^*_{j,l,n}(ws) = ws$.

Hence,

$$\begin{aligned} &\sigma_{j,l,n}(\rho_0 \ldots \rho_i) = \mathrm{next}_{j,l,n}(\rho_i, \mathrm{update}^*_{j,l,n}(\rho_0 \ldots \rho_i)) \\ =&\mathrm{next}_{j,l,n}(\rho_i, \rho_0 \ldots \rho_i) = \sigma_{j,l,n-1}(\rho_0 \ldots \rho_i) \end{aligned}$$

for all finite plays $\rho_0 \ldots \rho_i$ and all sufficiently large n. Thus, $(\sigma_{j,l,n})_{n\in\mathbb{N}}$ converges.

For the next two claims we need to introduce some additional notation. We have to determine the original play according to $\sigma_{j,l-1}$ for every play according to $\sigma_{j,l}$. To this end, we apply the functions $\mathrm{update}_{j,l,n}$ successively in reverse order, thereby reconstructing the original play. Then, we can copy the proofs of Lemma 4.

For $1 \leq m \leq n$ define $\mathrm{update}^*_{j,l,[m,n]}$ by

$$\mathrm{update}^*_{j,l,[m,m]}(w) = \mathrm{update}^*_{j,l,m}(w)$$

and

$$\text{update}^*_{j,l,[m,n+1]}(w) = \text{update}^*_{j,l,[m,n]}(\text{update}^*_{j,l,n+1}(w)).$$

Applying Lemma 4 (i) inductively, we can show that $\text{update}^*_{j,l,[m,n]}(w)$ is a finite play consistent with $\sigma_{j,l,m-1}$ for every play w consistent with $\sigma_{j,l,n}$. Analogously, applying Lemma (ii) inductively, we get $t_{j,D}(w) \leq t_{j,D}(\text{update}^*_{j,l,[m,n]}(w))$ for all w consistent with $\sigma_{j,l,n}$, for all $j \in [k]$, and all $D \in \text{Up}(\mathcal{P}_j)$.

Example 5. This construction is illustrated in Figure 5, where a request $D = e_j(l)$ of condition j is open in the dashed intervals. Assume x is consistent with $\sigma_{j,l,3}$. Then, we have $x_2 = \text{update}^*_{j,l,3}(x) = \text{update}^*_{j,l,[3,3]}(x)$, which is consistent with $\sigma_{j,l,2}$. Applying $\text{update}^*_{j,l,2}$, we obtain $x_1 = \text{update}^*_{j,l,2}(x_2) = \text{update}^*_{j,l,[2,3]}(x)$, which is consistent with $\sigma_{j,l,1}$. Finally, we apply $\text{update}^*_{j,l,1}$, and obtain $x_0 = \text{update}^*_{j,l,1}(x_1) = \text{update}^*_{j,l,[1,3]}(x)$, which is consistent with $\sigma_{j,l,0} = \sigma_{j,l-1}$.

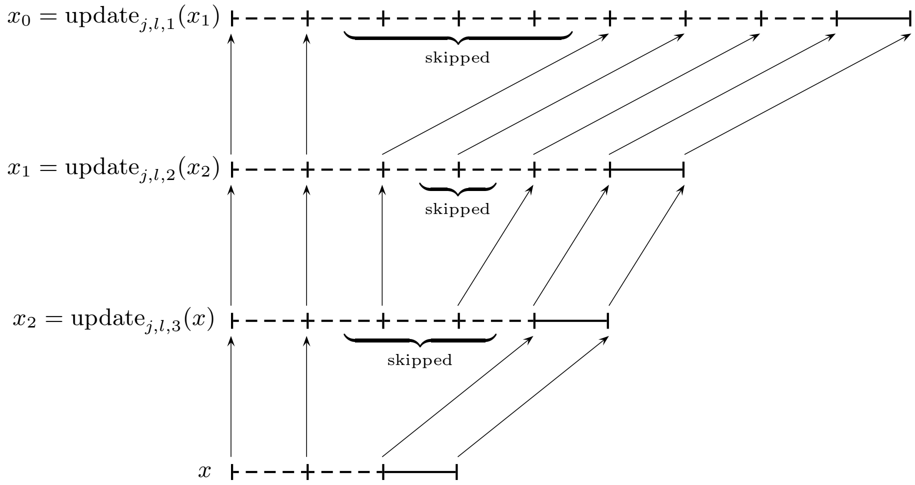

Fig. 5. Reconstruction of a play with $\text{update}^*_{j,l,[m,n]}$

We have shown in the proof of Lemma 5 (i) that for every finite play x according to $\sigma_{j,l}$ there is an n_x such that $\text{update}^*_{j,l,n}(x) = x$ for all $n \geq n_x$. We define $\text{update}^*_{j,l,\omega}$ by

$$\text{update}_{j,l,\omega}(x) = \text{update}^*_{j,l,[1,n_x]}(x) \ .$$

By the remarks above, we know that $\text{update}^*_{j,l,\omega}(x)$ is consistent with $\sigma_{j,l,0} = \sigma_{j,l-1}$ and $t_{j,D}(x) \leq t_{j,D}(\text{update}^*_{j,l,\omega}(x))$ for every play x consistent with $\sigma_{j,l}$, for all $j \in [k]$, and all $D \in \text{Up}(\mathcal{P}_j)$.

Example 6. Going back to Figure 5, assume $n_x \leq 3$, i.e., x is consistent with $\sigma_{j,l,n}$ for all $n \geq 3$ and especially $\sigma_{j,l,}$. Then, we have $\text{update}^*_{j,l,\omega}(x) = x_0$, which is a play consistent with σ_{j-1}.

(ii) We have $t_{j',D'}(x) \leq b$ for all finite plays x consistent with $\sigma_{j,l-1}$. Now, let x be a play consistent with $\sigma_{j,l}$. Then, $\mathrm{update}^*_{j,l,\omega}(x)$ is a prefix of a play according to $\sigma_{j,l-1}$. Hence, $t_{j',D'}(x) \leq t_{j',D'}(\mathrm{update}^*_{j,l,\omega}(x)) \leq b$. Thus, $\sigma_{j,l}$ uniformly bounds the totalized waiting time for D' to b.

(iii) Given $\rho = \rho_0\rho_1\rho_2\ldots$, let $\mathrm{update}^*_{j,l,\omega}(\rho) = \lim_{n\to\infty} \mathrm{update}^*_{j,l,\omega}(\rho_0 \ldots \rho_n)$. The limit $\mathrm{update}^*_{j,l,\omega}(\rho)$ is a play consistent with $\sigma_{j,l-1}$ for every play ρ consistent with $\sigma_{j,l}$. We show $v(\rho) \leq v(\mathrm{update}^*_{j,l,\omega}(\rho))$ for all ρ consistent with $\sigma_{j,l}$, which implies the claim. To this end, we define

$$S = \{x' \sqsubseteq \mathrm{update}^*_{j,l,\omega}(\rho) \mid \neg\exists x \sqsubseteq \rho : \mathrm{update}^*_{j,l,\omega}(x) = x'\} \enspace .$$

S contains exactly the vertices of the loops skipped by Player 0 throughout the improvement steps. Let $x' \in S$. Then, $t_{j,D}(x') > b_{\mathcal{G}}$ holds, as every improvement step only deletes loops of $\mathrm{update}^*_{j,l,\omega}(\rho)$ if the totalized waiting time is greater than $b_{\mathcal{G}}$. Thus, $p(x') > b_{\mathcal{G}} \geq v(\sigma)$ and

$$\limsup_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} p(\mathrm{update}^*_{j,l,\omega}(\rho_0 \ldots \rho_i)) \leq v(\mathrm{update}^*_{j,l,\omega}(\rho)) \enspace , \tag{3}$$

since the average decreases if the summation omits the summands for the prefixes in S. Now, let $x \sqsubseteq \rho$: We have $t_{j',D'}(x) \leq t_{j',D'}(\mathrm{update}^*_{j,l,\omega}(x))$, for all $j' \in [k]$, and all $D' \in \mathrm{Up}(\mathcal{P}_{j'})$, and therefore $p(x) \leq p(\mathrm{update}^*_{j,l,\omega}(x))$. Thus,

$$\frac{1}{n} \sum_{i=0}^{n-1} p(\rho_0 \ldots \rho_i) \leq \frac{1}{n} \sum_{i=0}^{n-1} p(\mathrm{update}^*_{j,l,\omega}(\rho_0 \ldots \rho_i)) \enspace .$$

The latter term converges to a value less than or equal to $v(\mathrm{update}^*_{j,l,\omega}(\rho))$, by (3). Thus, we conclude $v(\rho) \leq v(\mathrm{update}^*_{j,l,\omega}(\rho))$.

(iv) The last claim is proven by induction over l. There are at most $|D_j \backslash D|$ intervals in between the elements of D_j that were embedded already (and the earliest request that was not yet responded). By induction hypothesis, we can assume that the claim holds for all those intervals, i.e., for all $D' \in \mathrm{Up}(\mathcal{P}_j)$ such that $|D'| > |D|$, since $e_j^{-1}(D') < e_j^{-1}(D)$. Now, assume there is a play w consistent with $\sigma_{j,l}$ such that $(D, t) \in \mathrm{Open}_j(w)$ for some $t > b_{j,D}$ and let $w = xy$ such that $|y| = b(|G|, c)$. From the induction hypothesis and the fact that the waiting times for all D' that could precede D are already bounded, we can conclude that

$$(D, b_{\mathcal{G}} + (|D\backslash D_j|) \cdot b(|G|, c) + 1 + |y'|) \in \mathrm{Open}_j(xy')$$

for all $y' \sqsubseteq y$. Also, we have $s_{j,D}(y') \leq s_{j,D}(y'')$ for all $y' \sqsubseteq y'' \sqsubseteq y$, since no $d \in D$ can be embedded in a vertex of y. Thus, if the waiting time is longer than $b_{j,D}$, then the last interval is longer than $b(|G|, c)$. The situation is depicted in Figure 6.

Finally, there exists n such that $\mathrm{update}^*_{j,l,n}(xy') = xy'$ for all $y' \sqsubseteq y$. Thus, y is non-Dickson save D, which contradicts the definition of the function b. □

The results of Lemma 5 can be combined by an induction over j to show that the waiting times can be bounded without increasing the value of a strategy.

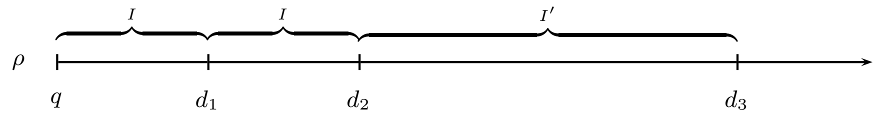

Fig. 6. The inductive step for Lemma 5 (iv): the intervals I are *short* by induction hypothesis. Thus, I' is *long* and contains a Dickson pair

Lemma 6. *For every winning strategy σ_0 for Player 0 with $v(\sigma_0) \leq b_{\mathcal{G}}$, there is a winning strategy σ_k for Player 0 that uniformly bounds $s_{j,D}$ to $b_{j,D}$ and $t_{j,D}$ to $tb_{j,D} := \frac{1}{2}(b_{j,D}(b_{j,D}+1))$ for all $j \in [k]$ and all $D \in \mathrm{Up}(\mathcal{P}_j)$. Furthermore, $v(\sigma_k) \leq v(\sigma_0)$.*

5.2 Reducing Poset Games to Mean-Payoff Games

In this subsection, we reduce the poset game to a mean-payoff game [2], which we will introduce in the following. An optimal strategy for the mean-payoff game will induce an optimal strategy for the poset game. Lemma 6 is crucial to obtain a finite arena.

A *mean-payoff game* $\mathcal{G} = (G, d, l)$ consists of an arena $G = (V, V_0, V_1, E, s_0)$, $d \in \mathbb{N}$ and a labeling function $l : E \to \{-d, \dots, d\}$ (note that l labels the edges in this case). Let ρ be a play in G. The *gain* $v_0(\rho)$ for Player 0 is defined as $v_0(\rho) = \liminf_{n\to\infty} \frac{1}{n}\sum_{i=0}^{n-1} l(\rho_i, \rho_{i+1})$ and the *loss* $v_1(\rho)$ for Player 1 is $v_1(\rho) = \limsup_{n\to\infty} \frac{1}{n}\sum_{i=0}^{n-1} l(\rho_i, \rho_{i+1})$. Player 0's goal is to maximize $v_0(\rho)$ whereas Player 1 aims to minimize $v_1(\rho)$. A strategy σ for Player 0 guarantees a gain of v if $v_0(\rho) \geq v$ for every play ρ consistent with σ. Analogously, τ for Player 1 guarantees a loss of v if $v_1(\rho) \leq v$ for every play ρ consistent with τ.

Theorem 3 ([2, 9]). *Let $\mathcal{G}$ be a mean-payoff game. There exists a value $\nu_{\mathcal{G}}$ and positional strategies σ and τ that guarantee $\nu_{\mathcal{G}}$ for Player 0 and Player 1, respectively. These strategies are optimal, i.e., there is no strategy for Player i that guarantees a better value for her. Furthermore, σ, τ and $\nu_{\mathcal{G}}$ are computable in pseudo-polynomial time.*

Now, we explain the reduction. The memory keeps track of the totalized waiting time $t_{j,D}(w)$ for every $j \in [k]$ and every $D \in \mathrm{Up}(\mathcal{P}_j)$. To be able to compute $t_{j,D}(ws)$ from $t_{j,D}(w)$ in every update of the memory state, $s_{j,D}(w)$ has to be stored as well. Due to Lemma 6, we can bound $t_{j,D}(w)$ by $tb_{j,D}$ and $s_{j,D}(w)$ by $b_{j,D}$. If these bounds are exceeded, then the memory is updated to a sink state $m_\uparrow$. Hence, we obtain a finite memory structure $\mathfrak{M}$. An edge is labeled by the sum of the totalized waiting times at the source of the edge. The value d is defined appropriately and is also the weight of all edges leading to a vertex with memory state $m_\uparrow$. As it is Player 1's goal to minimize the limit superior of the average edge labels, we have to exchange the positions of the players.

Let T_j and S_j be the set of functions $t, s : \mathrm{Up}(\mathcal{P}_j) \to \mathbb{N}$ such that $t(D) \leq tb_{j,D}$ respectively $s(D) \leq b_{j,D}$ for all $D \in \mathrm{Up}(\mathcal{P}_j)$. Every T_j and S_j is obviously finite. Define the memory structure $\mathfrak{M} = (M, m_0, \mathrm{update})$ where

– $M = \prod_{j=1}^{k}(T_j \times S_j) \cup \{m_\uparrow\}$ with sink state $m_\uparrow$,

– $m_0 = (t_1, s_1, \dots, t_k, s_k)$ where $t_j(D) = s_j(D) = |\mathrm{Emb}_j(\mathrm{New}_j(s_0), s_0) \cap \{D\}|$ for all $D \in \mathrm{Up}(\mathcal{P}_j)$, and
– The update function is given by $\mathrm{update}(m_\uparrow, s) = m_\uparrow$ for all vertices s; otherwise, if $m = (t_1, s_1, \dots, t_k, s_k)$ define t'_j and s'_j by

$$t'_j(D) = |\mathrm{Emb}_j(\mathrm{New}_j(s), s) \cap \{D\}| + \sum_{\substack{D' \in \mathrm{Up}(\mathcal{P}_j):\\ \mathrm{Emb}_j(D', s) = D}} (t_j(D') + s_j(D'))$$

and

$$s'_j(D) = |\mathrm{Emb}_j(\mathrm{New}_j(s), s) \cap \{D\}| + \sum_{\substack{D' \in \mathrm{Up}(\mathcal{P}_j):\\ \mathrm{Emb}_j(D', s) = D}} s_j(D')$$

for all $D \in \mathrm{Up}(\mathcal{P}_j)$. If $t'_j \notin T_j$ or $s'_j \notin S_j$ for some j, then $\mathrm{update}(m, s) = m_\uparrow$, otherwise $\mathrm{update}(m, s) = (t'_1, s'_1, \dots, t'_k, s'_k)$.

Since it is Player 1's goal to minimize the limit superior of the average edge weights in a mean-payoff game, we have to swap the player's positions, i.e., $V'_0 = V_1 \times M$ and $V'_1 = V_0 \times M$. Let $s'_0 = (s_0, m_0)$ and $((s, m), (s', m')) \in E'$ iff $(s, s') \in E$ and $\mathrm{update}(m, s) = m'$. Furthermore, let $d = \sum_{j=1}^{k} tb_{j,D} + 1$. To complete the definition of the mean-payoff game define $l((s, m), (s', m_\uparrow)) = d$ for all $(s, m) \in V \times M$ and

$$l((s, (t_1, s_1, \dots, t_k, s_k)), (s', (t'_1, s'_1, \dots, t'_k, s'_k))) = \sum_{j=1}^{k} \sum_{D \in \mathrm{Up}(\mathcal{P}_j)} t_j(D) \enspace .$$

Now, let $\mathcal{G}' = (G', d, l)$ with arena $G' = (V \times M, V'_0, V'_1, E', s'_0)$.

The following remark shows that the construction is correct: the values of the plays in the two arenas coincide. Its proof is a simple, but tedious induction showing that $\mathrm{update}^*(w)$ stores the waiting times $t_{j,D}(w)$ and the number of open requests $s_{j,D}(w)$.

Remark 3. (i) Let ρ be a play in G such that the totalized waiting times for all $j \in [k]$ and all $D \in \mathrm{Up}(\mathcal{P}_j)$ are uniformly bounded by $tb_{j,D}$, and let ρ' be the expanded play in G'. Then, $v(\rho) = v_1(\rho')$.
(ii) Let ρ' be a play in G' that does not visit a vertex with memory state $m_\uparrow$, and let ρ be the projected play in G. Then, $v_1(\rho') = v(\rho)$.

Now, we are able to prove Theorem 2.

Proof. Let Player 0 win $\mathcal{G}$. We begin by relating strategies and values for Player 0 for $\mathcal{G}$ and Player 1 in $\mathcal{G}'$.

Let σ be a strategy for Player 0 for $\mathcal{G}$ that uniformly bounds the totalized waiting times for all conditions $D \in \mathrm{Up}(\mathcal{P}_j)$ to $tb_{j,D}$. We define the strategy τ' for Player 1 in $\mathcal{G}'$ by $\tau'((\rho_0, m_0) \dots (\rho_n, m_n)) = \sigma(\rho_0 \dots \rho_n)$. We claim τ' guarantees $v(\sigma)$ for Player 1 in $\mathcal{G}'$. Assume it does not. Then, Player 0 has a strategy σ' for $\mathcal{G}'$ such that $v_1(\rho(\sigma', \tau')) > v(\sigma)$. The projected play ρ of $\rho(\sigma', \tau')$ is consistent with σ by construction of τ'. Thus, $v(\sigma) \geq v(\rho) = v_1(\rho(\sigma', \tau')) > v(\sigma)$ by Remark 3 (i), which yields the desired contradiction.

Conversely, let τ' be a strategy for Player 1 in $\mathcal{G}'$ that guarantees a loss $d' < d$. Thus, no play consistent with τ' visits a vertex with memory state $m_{\uparrow}$. Let σ be the strategy for Player 0 for $\mathcal{G}$ induced by τ' via $\mathcal{G} \leq_{\mathfrak{M}} \mathcal{G}'$. We claim $v(\sigma) \leq d'$. Assume Player 1 has a strategy τ for $\mathcal{G}$ such that $v(\rho(\sigma, \tau)) > d'$. The expanded play ρ' of $\rho(\sigma, \tau)$ is consistent with τ'. Thus, $d' \geq v_1(\rho') = v(\rho(\sigma, \tau)) > d'$ by Remark 3 (ii), which again amounts to a contradiction.

Now, we can begin with the actual proof: since Player 0 wins $\mathcal{G}$, Corollary 1 and Lemma 6 guarantee that she also has a winning strategy σ that uniformly bounds the totalized waiting times for all $D \in \mathrm{Up}(\mathcal{P}_j)$ to $tb_{j,D}$. Let τ' be the induced strategy for Player 1 in $\mathcal{G}'$. Every play consistent with τ' does not reach a vertex with memory state $m_{\uparrow}$. Thus, this strategy guarantees a loss less than d. Hence, $\nu_{\mathcal{G}'} < d$.

Let τ_{opt} be a positional strategy guaranteeing $\nu_{\mathcal{G}'}$ for Player 1 in $\mathcal{G}'$. We show that the strategy σ_{opt} induced by $\mathcal{G} \leq_{\mathfrak{M}} \mathcal{G}'$ and τ_{opt} is an optimal winning strategy for Player 0 for $\mathcal{G}$. This suffices since σ_{opt} is finite-state with memory $\mathfrak{M}$ and effectively computable by Theorem 3.

By the remarks above, we have $v(\sigma_{opt}) = \nu_{\mathcal{G}'}$. To conclude the proof we assume that σ_{opt} is not optimal, i.e., Player 0 has a strategy σ for $\mathcal{G}$ such that $v(\sigma) < v(\sigma_{opt})$. By Lemma 6, we can assume without loss of generality that σ uniformly bounds the totalized waiting times for all $D \in \mathrm{Up}(\mathcal{P}_j)$ to $tb_{j,D}$. Then, the strategy τ' for Player 1 in $\mathcal{G}'$ induced by σ guarantees $v(\sigma) < v(\sigma_{opt}) = \nu_{\mathcal{G}'}$. This amounts to a contradiction, since τ_{opt} is optimal for $\mathcal{G}'$.

Furthermore, the values of both optimal strategies coincide. □

6 Conclusion

We have introduced a novel winning condition for infinite two-player games that extends the request-response condition while retaining a natural definition of waiting times. These games are well-suited to add aspects of planning to the synthesis of finite-state controllers for reactive systems. We proved that optimal strategies (with respect to long-term average accumulated waiting times) exist and are effectively computable. The memory size of the optimal strategy computed here is super-exponential. However, this holds already for request-response games. Thus, the increased expressiveness of the poset condition does not add too much additional complexity.

In future research, the memory size should be analyzed: determining the computational complexity of finding optimal strategies and proving tight upper and lower bounds on the memory size of an optimal strategy. The size of the mean-payoff game (and thus the memory) can be reduced by finding better bounds on the length of non-Dickson infixes. Also, one should investigate, whether the (costly, in terms of time and space) reduction to mean-payoff games is necessary: can an optimal strategy be computed without a reduction?

Another direction of further research is to consider *discounted* waiting times and to establish a reduction to *discounted payoff games* [9]. Furthermore, the reduction to Büchi games induces a uniform upper bound on the waiting times in poset games, but the (efficient) computation of optimal bounds should be addressed as well.

Acknowledgments. This work presents results of the author's diploma thesis [7] prepared under the supervision of Wolfgang Thomas. I want to thank him for his advice and suggestions. Also, I want to thank the anonymous referees of [8] for their helpful remarks.

References

1. Leonard E. Dickson. Finiteness of the odd perfect and primitive abundant numbers with n distinct prime factors. *Amer. J. Math.*, 35(4):413–422, 1913.
2. Andrzej Ehrenfeucht and Jan Mycielski. Positional strategies for mean payoff games. *International Journal of Game Theory*, 8(2):109–113, 1979.
3. Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata, Logics, and Infinite Games*, volume 2500 of *LNCS*. Springer, 2002.
4. Florian Horn, Wolfgang Thomas, and Nico Wallmeier. Optimal strategy synthesis in request-response games. In Sung Deok Cha, Jin-Young Choi, Moonzoo Kim, Insup Lee, and Mahesh Viswanathan, editors, *ATVA*, volume 5311 of *LNCS*, pages 361–373. Springer, 2008.
5. Nico Wallmeier. *Strategien in unendlichen Spielen mit Liveness-Gewinnbedingungen: Syntheseverfahren, Optimierung und Implementierung.* PhD thesis, RWTH Aachen University, 2008.
6. Nico Wallmeier, Patrick Hütten, and Wolfgang Thomas. Symbolic synthesis of finite-state controllers for request-response specifications. In Oscar H. Ibarra and Zhe Dang, editors, *CIAA*, volume 2759 of *Lecture Notes in Computer Science*, pages 11–22. Springer, 2003.
7. Martin Zimmermann. *Time-optimal Winning Strategies in Infinite Games.* Diploma Thesis, RWTH Aachen University, 2009. `automata.rwth-aachen.de/~zimmermann`.
8. Martin Zimmermann. Time-optimal winning strategies for poset games. In Sebastian Maneth, editor, *CIAA*, volume 5642 of *Lecture Notes in Computer Science*, pages 217–226. Springer, 2009.
9. Uri Zwick and Mike Paterson. The complexity of mean payoff games on graphs. *Theoretical Computer Science*, 158(1&2):343–359, 1996.

Aachener Informatik-Berichte

This list contains all technical reports published during the past five years. A complete list of reports dating back to 1987 is available from `http://aib.informatik.rwth-aachen.de/`. To obtain copies consult the above URL or send your request to: Informatik-Bibliothek, RWTH Aachen, Ahornstr. 55, 52056 Aachen, Email: `biblio@informatik.rwth-aachen.de`

2004-01 * Fachgruppe Informatik: Jahresbericht 2003
2004-02 Benedikt Bollig, Martin Leucker: Message-Passing Automata are expressively equivalent to EMSO logic
2004-03 Delia Kesner, Femke van Raamsdonk, Joe Wells (eds.): HOR 2004 – 2nd International Workshop on Higher-Order Rewriting
2004-04 Slim Abdennadher, Christophe Ringeissen (eds.): RULE 04 – Fifth International Workshop on Rule-Based Programming
2004-05 Herbert Kuchen (ed.): WFLP 04 – 13th International Workshop on Functional and (Constraint) Logic Programming
2004-06 Sergio Antoy, Yoshihito Toyama (eds.): WRS 04 – 4th International Workshop on Reduction Strategies in Rewriting and Programming
2004-07 Michael Codish, Aart Middeldorp (eds.): WST 04 – 7th International Workshop on Termination
2004-08 Klaus Indermark, Thomas Noll: Algebraic Correctness Proofs for Compiling Recursive Function Definitions with Strictness Information
2004-09 Joachim Kneis, Daniel Mölle, Stefan Richter, Peter Rossmanith: Parameterized Power Domination Complexity
2004-10 Zinaida Benenson, Felix C. Gärtner, Dogan Kesdogan: Secure Multi-Party Computation with Security Modules
2005-01 * Fachgruppe Informatik: Jahresbericht 2004
2005-02 Maximillian Dornseif, Felix C. Gärtner, Thorsten Holz, Martin Mink: An Offensive Approach to Teaching Information Security: "Aachen Summer School Applied IT Security"
2005-03 Jürgen Giesl, René Thiemann, Peter Schneider-Kamp: Proving and Disproving Termination of Higher-Order Functions
2005-04 Daniel Mölle, Stefan Richter, Peter Rossmanith: A Faster Algorithm for the Steiner Tree Problem
2005-05 Fabien Pouget, Thorsten Holz: A Pointillist Approach for Comparing Honeypots
2005-06 Simon Fischer, Berthold Vöcking: Adaptive Routing with Stale Information
2005-07 Felix C. Freiling, Thorsten Holz, Georg Wicherski: Botnet Tracking: Exploring a Root-Cause Methodology to Prevent Distributed Denial-of-Service Attacks
2005-08 Joachim Kneis, Peter Rossmanith: A New Satisfiability Algorithm With Applications To Max-Cut
2005-09 Klaus Kursawe, Felix C. Freiling: Byzantine Fault Tolerance on General Hybrid Adversary Structures
2005-10 Benedikt Bollig: Automata and Logics for Message Sequence Charts
2005-11 Simon Fischer, Berthold Vöcking: A Counterexample to the Fully Mixed Nash Equilibrium Conjecture

2005-12 Neeraj Mittal, Felix Freiling, S. Venkatesan, Lucia Draque Penso: Efficient Reductions for Wait-Free Termination Detection in Faulty Distributed Systems

2005-13 Carole Delporte-Gallet, Hugues Fauconnier, Felix C. Freiling: Revisiting Failure Detection and Consensus in Omission Failure Environments

2005-14 Felix C. Freiling, Sukumar Ghosh: Code Stabilization

2005-15 Uwe Naumann: The Complexity of Derivative Computation

2005-16 Uwe Naumann: Syntax-Directed Derivative Code (Part I: Tangent-Linear Code)

2005-17 Uwe Naumann: Syntax-directed Derivative Code (Part II: Intraprocedural Adjoint Code)

2005-18 Thomas von der Maßen, Klaus Müller, John MacGregor, Eva Geisberger, Jörg Dörr, Frank Houdek, Harbhajan Singh, Holger Wußmann, Hans-Veit Bacher, Barbara Paech: Einsatz von Features im Software-Entwicklungsprozess - Abschlußbericht des GI-Arbeitskreises “Features”

2005-19 Uwe Naumann, Andre Vehreschild: Tangent-Linear Code by Augmented LL-Parsers

2005-20 Felix C. Freiling, Martin Mink: Bericht über den Workshop zur Ausbildung im Bereich IT-Sicherheit Hochschulausbildung, berufliche Weiterbildung, Zertifizierung von Ausbildungsangeboten am 11. und 12. August 2005 in Köln organisiert von RWTH Aachen in Kooperation mit BITKOM, BSI, DLR und Gesellschaft fuer Informatik (GI) e.V.

2005-21 Thomas Noll, Stefan Rieger: Optimization of Straight-Line Code Revisited

2005-22 Felix Freiling, Maurice Herlihy, Lucia Draque Penso: Optimal Randomized Fair Exchange with Secret Shared Coins

2005-23 Heiner Ackermann, Alantha Newman, Heiko Röglin, Berthold Vöcking: Decision Making Based on Approximate and Smoothed Pareto Curves

2005-24 Alexander Becher, Zinaida Benenson, Maximillian Dornseif: Tampering with Motes: Real-World Physical Attacks on Wireless Sensor Networks

2006-01 * Fachgruppe Informatik: Jahresbericht 2005

2006-02 Michael Weber: Parallel Algorithms for Verification of Large Systems

2006-03 Michael Maier, Uwe Naumann: Intraprocedural Adjoint Code Generated by the Differentiation-Enabled NAGWare Fortran Compiler

2006-04 Ebadollah Varnik, Uwe Naumann, Andrew Lyons: Toward Low Static Memory Jacobian Accumulation

2006-05 Uwe Naumann, Jean Utke, Patrick Heimbach, Chris Hill, Derya Ozyurt, Carl Wunsch, Mike Fagan, Nathan Tallent, Michelle Strout: Adjoint Code by Source Transformation with OpenAD/F

2006-06 Joachim Kneis, Daniel Mölle, Stefan Richter, Peter Rossmanith: Divide-and-Color

2006-07 Thomas Colcombet, Christof Löding: Transforming structures by set interpretations

2006-08 Uwe Naumann, Yuxiao Hu: Optimal Vertex Elimination in Single-Expression-Use Graphs

2006-09 Tingting Han, Joost-Pieter Katoen: Counterexamples in Probabilistic Model Checking

2006-10 Mesut Günes, Alexander Zimmermann, Martin Wenig, Jan Ritzerfeld, Ulrich Meis: From Simulations to Testbeds - Architecture of the Hybrid MCG-Mesh Testbed

2006-11 Bastian Schlich, Michael Rohrbach, Michael Weber, Stefan Kowalewski: Model Checking Software for Microcontrollers

2006-12 Benedikt Bollig, Joost-Pieter Katoen, Carsten Kern, Martin Leucker: Replaying Play in and Play out: Synthesis of Design Models from Scenarios by Learning

2006-13 Wong Karianto, Christof Löding: Unranked Tree Automata with Sibling Equalities and Disequalities

2006-14 Danilo Beuche, Andreas Birk, Heinrich Dreier, Andreas Fleischmann, Heidi Galle, Gerald Heller, Dirk Janzen, Isabel John, Ramin Tavakoli Kolagari, Thomas von der Maßen, Andreas Wolfram: Report of the GI Work Group "Requirements Management Tools for Product Line Engineering"

2006-15 Sebastian Ullrich, Jakob T. Valvoda, Torsten Kuhlen: Utilizing optical sensors from mice for new input devices

2006-16 Rafael Ballagas, Jan Borchers: Selexels: a Conceptual Framework for Pointing Devices with Low Expressiveness

2006-17 Eric Lee, Henning Kiel, Jan Borchers: Scrolling Through Time: Improving Interfaces for Searching and Navigating Continuous Audio Timelines

2007-01 * Fachgruppe Informatik: Jahresbericht 2006

2007-02 Carsten Fuhs, Jürgen Giesl, Aart Middeldorp, Peter Schneider-Kamp, René Thiemann, and Harald Zankl: SAT Solving for Termination Analysis with Polynomial Interpretations

2007-03 Jürgen Giesl, René Thiemann, Stephan Swiderski, and Peter Schneider-Kamp: Proving Termination by Bounded Increase

2007-04 Jan Buchholz, Eric Lee, Jonathan Klein, and Jan Borchers: coJIVE: A System to Support Collaborative Jazz Improvisation

2007-05 Uwe Naumann: On Optimal DAG Reversal

2007-06 Joost-Pieter Katoen, Thomas Noll, and Stefan Rieger: Verifying Concurrent List-Manipulating Programs by LTL Model Checking

2007-07 Alexander Nyßen, Horst Lichter: MeDUSA - MethoD for UML2-based Design of Embedded Software Applications

2007-08 Falk Salewski and Stefan Kowalewski: Achieving Highly Reliable Embedded Software: An empirical evaluation of different approaches

2007-09 Tina Kraußer, Heiko Mantel, and Henning Sudbrock: A Probabilistic Justification of the Combining Calculus under the Uniform Scheduler Assumption

2007-10 Martin Neuhäußer, Joost-Pieter Katoen: Bisimulation and Logical Preservation for Continuous-Time Markov Decision Processes

2007-11 Klaus Wehrle (editor): 6. Fachgespräch Sensornetzwerke

2007-12 Uwe Naumann: An L-Attributed Grammar for Adjoint Code

2007-13 Uwe Naumann, Michael Maier, Jan Riehme, and Bruce Christianson: Second-Order Adjoints by Source Code Manipulation of Numerical Programs

2007-14 Jean Utke, Uwe Naumann, Mike Fagan, Nathan Tallent, Michelle Strout, Patrick Heimbach, Chris Hill, and Carl Wunsch: OpenAD/F: A Modular, Open-Source Tool for Automatic Differentiation of Fortran Codes

2007-15 Volker Stolz: Temporal assertions for sequential and concurrent programs

2007-16 Sadeq Ali Makram, Mesut Güneç, Martin Wenig, Alexander Zimmermann: Adaptive Channel Assignment to Support QoS and Load Balancing for Wireless Mesh Networks

2007-17 René Thiemann: The DP Framework for Proving Termination of Term Rewriting

2007-18 Uwe Naumann: Call Tree Reversal is NP-Complete

2007-19 Jan Riehme, Andrea Walther, Jörg Stiller, Uwe Naumann: Adjoints for Time-Dependent Optimal Control

2007-20 Joost-Pieter Katoen, Daniel Klink, Martin Leucker, and Verena Wolf: Three-Valued Abstraction for Probabilistic Systems

2007-21 Tingting Han, Joost-Pieter Katoen, and Alexandru Mereacre: Compositional Modeling and Minimization of Time-Inhomogeneous Markov Chains

2007-22 Heiner Ackermann, Paul W. Goldberg, Vahab S. Mirrokni, Heiko Röglin, and Berthold Vöcking: Uncoordinated Two-Sided Markets

2008-01 * Fachgruppe Informatik: Jahresbericht 2007

2008-02 Henrik Bohnenkamp, Marielle Stoelinga: Quantitative Testing

2008-03 Carsten Fuhs, Jürgen Giesl, Aart Middeldorp, Peter Schneider-Kamp, René Thiemann, Harald Zankl: Maximal Termination

2008-04 Uwe Naumann, Jan Riehme: Sensitivity Analysis in Sisyphe with the AD-Enabled NAGWare Fortran Compiler

2008-05 Frank G. Radmacher: An Automata Theoretic Approach to the Theory of Rational Tree Relations

2008-06 Uwe Naumann, Laurent Hascoet, Chris Hill, Paul Hovland, Jan Riehme, Jean Utke: A Framework for Proving Correctness of Adjoint Message Passing Programs

2008-07 Alexander Nyßen, Horst Lichter: The MeDUSA Reference Manual, Second Edition

2008-08 George B. Mertzios, Stavros D. Nikolopoulos: The λ-cluster Problem on Parameterized Interval Graphs

2008-09 George B. Mertzios, Walter Unger: An optimal algorithm for the k-fixed-endpoint path cover on proper interval graphs

2008-10 George B. Mertzios, Walter Unger: Preemptive Scheduling of Equal-Length Jobs in Polynomial Time

2008-11 George B. Mertzios: Fast Convergence of Routing Games with Splittable Flows

2008-12 Joost-Pieter Katoen, Daniel Klink, Martin Leucker, Verena Wolf: Abstraction for stochastic systems by Erlang's method of stages

2008-13 Beatriz Alarcón, Fabian Emmes, Carsten Fuhs, Jürgen Giesl, Raúl Gutiérrez, Salvador Lucas, Peter Schneider-Kamp, René Thiemann: Improving Context-Sensitive Dependency Pairs

2008-14 Bastian Schlich: Model Checking of Software for Microcontrollers

2008-15 Joachim Kneis, Alexander Langer, Peter Rossmanith: A New Algorithm for Finding Trees with Many Leaves

2008-16 Hendrik vom Lehn, Elias Weingärtner and Klaus Wehrle: Comparing recent network simulators: A performance evaluation study

2008-17 Peter Schneider-Kamp: Static Termination Analysis for Prolog using Term Rewriting and SAT Solving

2008-18 Falk Salewski: Empirical Evaluations of Safety-Critical Embedded Systems

2009-03 Alexander Nyßen: Model-Based Construction of Embedded Real-Time Software - A Methodology for Small Devices

2009-04 Daniel Klünder: Entwurf eingebetteter Software mit abstrakten Zustandsmaschinen und Business Object Notation

2009-05 George B. Mertzios, Ignasi Sau, Shmuel Zaks: A New Intersection Model and Improved Algorithms for Tolerance Graphs

2009-06 George B. Mertzios, Ignasi Sau, Shmuel Zaks: The Recognition of Tolerance and Bounded Tolerance Graphs is NP-complete

2009-07 Joachim Kneis, Alexander Langer, Peter Rossmanith: Derandomizing Non-uniform Color-Coding I

2009-08 Joachim Kneis, Alexander Langer: Satellites and Mirrors for Solving Independent Set on Sparse Graphs

2009-09 Michael Nett: Implementation of an Automated Proof for an Algorithm Solving the Maximum Independent Set Problem

2009-10 Felix Reidl, Fernando Sánchez Villaamil: Automatic Verification of the Correctness of the Upper Bound of a Maximum Independent Set Algorithm

2009-11 Kyriaki Ioannidou, George B. Mertzios, Stavros D. Nikolopoulos: The Longest Path Problem is Polynomial on Interval Graphs

2009-12 Martin Neuhäußer, Lijun Zhang: Time-Bounded Reachability in Continuous-Time Markov Decision Processes

* These reports are only available as a printed version.

Please contact biblio@informatik.rwth-aachen.de to obtain copies.