Optimal Strategy Synthesis for Request-Response Games

Joint work with Florian Horn, Nico Wallmeier, and Wolfgang Thomas

Martin Zimmermann
Saarland University

September 26th, 2014

AVACS Meeting, Saarbrücken, Germany
You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 
Let’s Play

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 

Open req’s

1 2 3 4
Let’s Play

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 

Open req’s 1 2 3 4 

$Q_1, Q_2, Q_3, Q_4$ 

$P_1, P_2, P_3, P_4$ 

$Q_1, Q_2, Q_3$ 

$P_2, P_3, P_4$ 

$P_1$ 

$P_3$ 

$P_4$ 

$Q_1$ 

$Q_1, Q_2, Q_3, Q_4$
You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 

Let’s Play
Let’s Play

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 

Open req’s 2 3 4
Let’s Play

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 
Let’s Play

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 
Let’s Play

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 
Let’s Play

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 
You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 

Open req’s 3 4
You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 

Let’s Play
Let’s Play

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 

Open req’s 3 4
Let’s Play

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$.
Let’s Play

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 

Open req’s 

$1 \quad 2 \quad 4$
Let’s Play

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 

$Q_1, Q_2, Q_3$

$P_2, P_3, P_4$

$P_1$

$P_3$

$P_4$

$Q_1, Q_2$

$Q_1, Q_2, Q_3, Q_4$
Let’s Play

You move at \( h \) and have to answer every visit to \( Q_j \) by visit to \( P_j \).
Let’s Play

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 

Open req’s

$Q_1, Q_2, Q_3$

$P_2, P_3, P_4$

$Q_1, Q_2, Q_3, Q_4$
You move at \( h \) and have to answer every visit to \( Q_j \) by visit to \( P_j \).
Let’s Play

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 

Open req’s 4

$Q_1, Q_2, Q_3$
Let’s Play

You move at \( h \) and have to answer every visit to \( Q_j \) by visit to \( P_j \).
You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 
You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 

**Let’s Play**
Let’s Play

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 

Open req’s 4
Let’s Play

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 
Let’s Play

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 

Open req’s
Let’s Play

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$.  

Open req’s

$1 \ 2 \ 3$
Let’s Play

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 

Open req’s $1 \ 2 \ 3$
Let’s Play

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 
You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 
Let’s Play

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 
Let’s Play

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 

Open req’s

$Q_1, Q_2, Q_3$  

$P_1, P_2, P_3, P_4$  

$Q_1, Q_2$  

$Q_1, Q_2, Q_3, Q_4$
You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 

Let’s Play

$Q_1, Q_2, Q_3$

$P_1, P_2, P_3, P_4$

Open req’s

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 

Martin Zimmermann  
Saarland University  
Optimal Strategy Synthesis for Request-Response Games  
2/16
Let’s Play

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 
You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 
Let’s Play

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 
Let’s Play

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 

Open req’s 3
You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 
Let’s Play

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 

Let’s Play

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 
Let’s Play

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 

Open req’s

2

$Q_1, Q_2, Q_3$

$P_1, P_2, P_3, P_4$

$Q_1, Q_2$

$Q_1, Q_2, Q_3, Q_4$
You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 
Let’s Play

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 
You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 
You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 

Open req’s 1
You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 
Let’s Play

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 
Let’s Play

Open req’s

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 
Let’s Play

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 

Open req’s
Let’s Play

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 
Open req’s

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 
Let’s Play

You move at $h$ and have to answer every visit to $Q_j$ by visit to $P_j$. 

Open req’s

$Q_1, Q_2, Q_3$

$P_1, P_2, P_3, P_4$

$Q_1, Q_2$

$Q_1, Q_2, Q_3, Q_4$
Definitions

- Arena: $\mathcal{A} = (V, V_0, V_1, E)$ with finite, directed graph $(V, E)$, $V_0 \subseteq V$, and $V_1 = V \setminus V_0$ (positions of the players).
Definitions

- **Arena:** $\mathcal{A} = (V, V_0, V_1, E)$ with finite, directed graph $(V, E)$, $V_0 \subseteq V$, and $V_1 = V \setminus V_0$ (positions of the players).
- **Play:** infinite path through $\mathcal{A}$
Definitions

- **Arena**: \( \mathcal{A} = (V, V_0, V_1, E) \) with finite, directed graph \((V, E)\), \(V_0 \subseteq V\), and \(V_1 = V \setminus V_0\) (positions of the players).
- **Play**: infinite path through \( \mathcal{A} \)
- **Game**: \( G = (\mathcal{A}, \text{Win}) \) with set \( \text{Win} \subseteq V^\omega \) of winning plays for Player 0. Player 1 wins all other plays.
Definitions

- **Arena**: $A = (V, V_0, V_1, E)$ with finite, directed graph $(V, E)$, $V_0 \subseteq V$, and $V_1 = V \setminus V_0$ (positions of the players).
- **Play**: infinite path through $A$
- **Game**: $G = (A, \text{Win})$ with set $\text{Win} \subseteq V^\omega$ of winning plays for Player 0. Player 1 wins all other plays.
- **Strategy for Player 0**: $\sigma : V^* V_0 \rightarrow V$ s.t. $(v, \sigma(wv)) \in E$ for all $wv \in V^* V_0$. 

Definitions

- **Arena**: $\mathcal{A} = (V, V_0, V_1, E)$ with finite, directed graph $(V, E)$, $V_0 \subseteq V$, and $V_1 = V \setminus V_0$ (positions of the players).
- **Play**: infinite path through $\mathcal{A}$
- **Game**: $G = (\mathcal{A}, \text{Win})$ with set $\text{Win} \subseteq V^\omega$ of winning plays for Player 0. Player 1 wins all other plays.
- **Strategy for Player 0**: $\sigma: V^* V_0 \rightarrow V$ s.t. $(v, \sigma(wv)) \in E$ for all $wv \in V^* V_0$.
- **$\rho$ consistent with $\sigma$**: $\rho_{n+1} = \sigma(\rho_0 \cdots \rho_n)$ for all $n$ s.t. $\rho_n \in V_0$.

$$\text{Beh}(v, \sigma) = \{\rho \mid \rho \text{ starting in } v, \text{ consistent with } \sigma\}$$
Definitions

- **Arena**: $\mathcal{A} = (V, V_0, V_1, E)$ with finite, directed graph $(V, E)$, $V_0 \subseteq V$, and $V_1 = V \setminus V_0$ (positions of the players).
- **Play**: infinite path through $\mathcal{A}$
- **Game**: $G = (\mathcal{A}, \text{Win})$ with set $\text{Win} \subseteq V^\omega$ of winning plays for Player 0. Player 1 wins all other plays.
- **Strategy for Player 0**: $\sigma : V^* V_0 \rightarrow V$ s.t. $(v, \sigma(wv)) \in E$ for all $wv \in V^* V_0$.
- $\rho$ consistent with $\sigma$: $\rho_{n+1} = \sigma(\rho_0 \cdots \rho_n)$ for all $n$ s.t. $\rho_n \in V_0$.

$$\text{Beh}(v, \sigma) = \{ \rho \mid \rho \text{ starting in } v, \text{ consistent with } \sigma \}$$

- $\sigma$ winning from $v$ for Player 0: $\text{Beh}(v, \sigma) \subseteq \text{Win}$. 

Martin Zimmermann  
Saarland University  
Optimal Strategy Synthesis for Request-Response Games  
3/16
Definitions

- **Arena**: $\mathcal{A} = (V, V_0, V_1, E)$ with finite, directed graph $(V, E)$, $V_0 \subseteq V$, and $V_1 = V \setminus V_0$ (positions of the players).
- **Play**: infinite path through $\mathcal{A}$
- **Game**: $\mathcal{G} = (\mathcal{A}, \text{Win})$ with set $\text{Win} \subseteq V^\omega$ of winning plays for Player 0. Player 1 wins all other plays.
- **Strategy for Player 0**: $\sigma : V^* V_0 \to V$ s.t. $(v, \sigma(wv)) \in E$ for all $wv \in V^* V_0$.
- $\rho$ consistent with $\sigma$: $\rho_{n+1} = \sigma(\rho_0 \cdots \rho_n)$ for all $n$ s.t. $\rho_n \in V_0$.

\[
\text{Beh}(v, \sigma) = \{\rho \mid \rho \text{ starting in } v, \text{ consistent with } \sigma\}
\]

- $\sigma$ winning from $v$ for Player 0: $\text{Beh}(v, \sigma) \subseteq \text{Win}$.
- **Winning region of Player 0**:

\[
W_0 = \{v \mid \text{Player 0 has winning strategy from } v\} \]
Reductions and Finite-state Strategies

- Positional Strategies: move only depends on last vertex

\[ \sigma(wv) = \sigma(v) \]

- Finite-state strategies: implemented by DFA with output reading play prefix \( \rho_0 \cdots \rho_n \) and outputting \( \sigma(\rho_0 \cdots \rho_n) \).

![Diagram of a DFA with states S0, S1, S2 and transitions labeled with input/output pairs 0/0, 1/1.](image-url)
RR Games

Request-response game (RR game): \((\mathcal{A}, (Q_j, P_j)_{j \in [k]})\) with

- arena \(\mathcal{A} = (V, V_0, V_1, E)\),
- \(Q_j \subseteq V\): reQuests of condition \(j\), and
- \(P_j \subseteq V\): resPonses of condition \(j\).

Theorem (Wallmeier, Hütten, Thomas '03)

RR games can be reduced to Büchi games of size \(s^2 + 1\), where \(s = |V|\).

Corollary

Finite-state winning strategies of size \(k^2 + 1\) for both players.

Solvable in Exptime.
RR Games

Request-response game (RR game): \((\mathcal{A}, (Q_j, P_j)_{j \in [k]})\) with
- arena \(\mathcal{A} = (V, V_0, V_1, E)\),
- \(Q_j \subseteq V\): requests of condition \(j\), and
- \(P_j \subseteq V\): responses of condition \(j\).
- Player 0 wins if every request is answered by corresponding response: \(\bigwedge_{j \in [k]} \mathbb{G}(Q_j \to \mathbb{F}P_j)\)
RR Games

Request-response game (RR game): \((\mathcal{A}, (Q_j, P_j)_{j \in [k]})\) with

- arena \(\mathcal{A} = (V, V_0, V_1, E)\),
- \(Q_j \subseteq V\): reQuests of condition \(j\), and
- \(P_j \subseteq V\): resPoneses of condition \(j\).
- Player 0 wins if every request is answered by corresponding response: \(\bigwedge_{j \in [k]} G(Q_j \rightarrow FP_j)\)

**Theorem (Wallmeier, Hütten, Thomas ’03)**

*RR games can be reduced to Büchi games of size \(sk2^{k+1}\), where \(s = |V|\).*
Request-response game (RR game): \( (\mathcal{A}, (Q_j, P_j)_{j \in [k]}) \) with

- arena \( \mathcal{A} = (V, V_0, V_1, E) \),
- \( Q_j \subseteq V \): reRequests of condition \( j \), and
- \( P_j \subseteq V \): resPonses of condition \( j \).
- Player 0 wins if every request is answered by corresponding response: \( \bigwedge_{j \in [k]} G(Q_j \rightarrow FP_j) \)

**Theorem (Wallmeier, Hütten, Thomas ’03)**

RR games can be reduced to Büchi games of size \( sk2^{k+1} \), where \( s = |V| \).

**Corollary**

- Finite-state winning strategies of size \( k2^{k+1} \) for both players.
- Solvable in Exptime.
\[
\text{wt}_j(\varepsilon) = 0, \quad \text{and}
\]
\[
\text{wt}_j(wv) = \begin{cases} 
0 & \text{if } \text{wt}_j(w) = 0 \text{ and } v \notin Q_j \setminus P_j, \\
1 & \text{if } \text{wt}_j(w) > 0 \text{ and } v \in P_j. 
\end{cases}
\]
Waiting Times

- $\text{wt}_j(\varepsilon) = 0$, and

\[
\text{wt}_j(wv) = \begin{cases} 
0 & \text{if } \text{wt}_j(w) = 0 \text{ and } v \notin Q_j \setminus P_j, \\
1 & \text{if } \text{wt}_j(w) = 0 \text{ and } v \in Q_j \setminus P_j,
\end{cases}
\]
Waiting Times

- \( wt_j(\varepsilon) = 0, \) and

\[
wt_j(wv) = \begin{cases} 
0 & \text{if } wt_j(w) = 0 \text{ and } v \notin Q_j \setminus P_j, \\
1 & \text{if } wt_j(w) = 0 \text{ and } v \in Q_j \setminus P_j, \\
0 & \text{if } wt_j(w) > 0 \text{ and } v \in P_j,
\end{cases}
\]
Waiting Times

- $\text{wt}_j(\varepsilon) = 0$, and

\[
\text{wt}_j(wv) = \begin{cases} 
0 & \text{if } \text{wt}_j(w) = 0 \text{ and } v \notin Q_j \setminus P_j, \\
1 & \text{if } \text{wt}_j(w) = 0 \text{ and } v \in Q_j \setminus P_j, \\
0 & \text{if } \text{wt}_j(w) > 0 \text{ and } v \in P_j, \\
\text{wt}_j(w) + 1 & \text{if } \text{wt}_j(w) > 0 \text{ and } v \notin P_j.
\end{cases}
\]
Waiting Times

- \( wt_j(\varepsilon) = 0 \), and

\[
wt_j(wv) = \begin{cases} 
0 & \text{if } wt_j(w) = 0 \text{ and } v \notin Q_j \setminus P_j, \\
1 & \text{if } wt_j(w) = 0 \text{ and } v \in Q_j \setminus P_j, \\
0 & \text{if } wt_j(w) > 0 \text{ and } v \in P_j, \\
wt_j(w) + 1 & \text{if } wt_j(w) > 0 \text{ and } v \notin P_j.
\end{cases}
\]

- \( \overline{wt}(w) = (wt_1(w), \ldots, wt_k(w)) \in \mathbb{N}^k \)
Waiting Times

- $\text{wt}_j(\varepsilon) = 0$, and

$$
\text{wt}_j(wv) = \begin{cases} 
0 & \text{if } \text{wt}_j(w) = 0 \text{ and } v \notin Q_j \setminus P_j, \\
1 & \text{if } \text{wt}_j(w) = 0 \text{ and } v \in Q_j \setminus P_j, \\
0 & \text{if } \text{wt}_j(w) > 0 \text{ and } v \in P_j, \\
\text{wt}_j(w) + 1 & \text{if } \text{wt}_j(w) > 0 \text{ and } v \notin P_j.
\end{cases}
$$

- $\overline{\text{wt}}(w) = (\text{wt}_1(w), \ldots, \text{wt}_k(w)) \in \mathbb{N}^k$

- $\text{val}(\rho) = \limsup_{n \to \infty} \frac{1}{n} \sum_{\ell=0}^{n-1} \sum_{j \in [k]} \text{wt}_j(\rho_0 \cdots \rho_{\ell})$
Waiting Times

- \( \text{wt}_j(\varepsilon) = 0 \), and

\[
\text{wt}_j(wv) = \begin{cases} 
0 & \text{if } \text{wt}_j(w) = 0 \text{ and } v \notin Q_j \setminus P_j, \\
1 & \text{if } \text{wt}_j(w) = 0 \text{ and } v \in Q_j \setminus P_j, \\
0 & \text{if } \text{wt}_j(w) > 0 \text{ and } v \in P_j, \\
\text{wt}_j(w) + 1 & \text{if } \text{wt}_j(w) > 0 \text{ and } v \notin P_j.
\end{cases}
\]

- \( \overline{\text{wt}}(w) = (\text{wt}_1(w), \ldots, \text{wt}_k(w)) \in \mathbb{N}^k \)

- \( \text{val}(\rho) = \limsup_{n \to \infty} \frac{1}{n} \sum_{\ell=0}^{n-1} \sum_{j \in [k]} \text{wt}_j(\rho_0 \cdots \rho_\ell) \)

- \( \text{val}(\sigma, \nu) = \sup_{\rho \in \text{Beh}(\nu,\sigma)} \text{val}(\rho) \)
Waiting Times

\[ wt_j(\varepsilon) = 0, \text{ and} \]

\[ wt_j(wv) = \begin{cases} 
0 & \text{if } wt_j(w) = 0 \text{ and } v \notin Q_j \setminus P_j, \\
1 & \text{if } wt_j(w) = 0 \text{ and } v \in Q_j \setminus P_j, \\
0 & \text{if } wt_j(w) > 0 \text{ and } v \in P_j, \\
wt_j(w) + 1 & \text{if } wt_j(w) > 0 \text{ and } v \notin P_j. 
\end{cases} \]

\[ \overline{wt}(w) = (wt_1(w), \ldots, wt_k(w)) \in \mathbb{N}^k \]

\[ \text{val}(\rho) = \limsup_{n \to \infty} \frac{1}{n} \sum_{\ell=0}^{n-1} \sum_{j \in [k]} wt_j(\rho_0 \cdots \rho_\ell) \]

\[ \text{val}(\sigma, \nu) = \sup_{\rho \in \text{Beh}(\nu, \sigma)} \text{val}(\rho) \]

**Goal:**
Prove that optimal winning strategies exist and are computable.
Example

Winning strategy $\sigma$: answer $Q_1$ and $Q_2$ alternatingly

$\text{val}(\sigma, v) = 56$ for every $v$
Example

- Winning strategy $\sigma$: answer $Q_1$ and $Q_2$ alternatingly
- $\text{val}(\sigma, v) = \frac{56}{10}$ for every $v$
Lemma

Player 0 has a winning strategy $\sigma$ with $\text{val}(\sigma, v) \leq \sum_{j \in [k]} s k 2^{k+1}$ for every $v \in W_0(G)$. 
Lemma

Player 0 has a winning strategy $\sigma$ with $\text{val}(\sigma, v) \leq \sum_{j \in [k]} sk^{2^k+1}$ for every $v \in W_0(\mathcal{G})$.

Consequence: Upper bound on value of optimal strategies.
Lemma

Player 0 has a winning strategy $\sigma$ with \( \text{val}(\sigma, v) \leq \sum_{j \in [k]} s^k 2^{k+1} \)
for every $v \in W_0(\mathcal{G})$.

Consequence: Upper bound on value of optimal strategies.

Lower bounds:
Waiting Times: Upper Bounds

**Lemma**

Player 0 has a winning strategy $\sigma$ with $\text{val}(\sigma, v) \leq \sum_{j \in [k]} sk^{2^{k+1}}$ for every $v \in W_0(G)$.

Consequence: Upper bound on value of optimal strategies.

Lower bounds:
**Lemma**

*Player 0 has a winning strategy* $\sigma$ *with* $\text{val}(\sigma, v) \leq \sum_{j \in [k]} sk^{2^{k+1}}$ *for every* $v \in W_0(G)$.

**Consequence:** Upper bound on value of optimal strategies.

**Lower bounds:**
- It takes $2^3$ visits to $h$ to answer $Q_4$. 
Lemma

Player 0 has a winning strategy $\sigma$ with $\text{val}(\sigma, v) \leq \sum_{j \in [k]} sk^{2k+1}$ for every $v \in W_0(\mathcal{G})$.

Consequence: Upper bound on value of optimal strategies.

Lower bounds:
- It takes $2^3$ visits to $h$ to answer $Q_4$.
- Generalizable to $k$ pairs.
- Lower bound $2^{k-1}$
Main Theorem

Theorem

*Optimal strategies for RR games exist, are effectively computable, and finite-state.*
Main Theorem

Theorem

Optimal strategies for RR games exist, are effectively computable, and finite-state.

Proof strategy:

1. Strategies of small value can be turned into strategies with bounded waiting times without increasing the value.
**Main Theorem**

**Theorem**
*Optimal strategies for RR games exist, are effectively computable, and finite-state.*

**Proof strategy:**

1. Strategies of *small* value can be turned into strategies with bounded waiting times without increasing the value.
   - This applies to optimal strategies as well.
   - Makes the search space for optimal strategies finite.
   - Involves removing parts of plays with large waiting times.
Main Theorem

Theorem

Optimal strategies for RR games exist, are effectively computable, and finite-state.

Proof strategy:

1. Strategies of *small* value can be turned into strategies with bounded waiting times without increasing the value.
   - This applies to optimal strategies as well.
   - Makes the search space for optimal strategies finite.
   - Involves removing parts of plays with large waiting times.

2. Expand arena by keeping track of waiting time vectors up to bound from 1.). RR-values equal to mean-payoff condition.
Main Theorem

Theorem

Optimal strategies for RR games exist, are effectively computable, and finite-state.

Proof strategy:

1. Strategies of small value can be turned into strategies with bounded waiting times without increasing the value.
   - This applies to optimal strategies as well.
   - Makes the search space for optimal strategies finite.
   - Involves removing parts of plays with large waiting times.

2. Expand arena by keeping track of waiting time vectors up to bound from 1.). RR-values equal to mean-payoff condition.
   - Optimal strategy for mean-payoff yields optimal strategy for RR game.
Dickson’s Lemma

- Fix $k > 0$ and order $\mathbb{N}^k$ componentwise: $(3, 7) \leq (7, 11)$. 
Dickson’s Lemma

- Fix $k > 0$ and order $\mathbb{N}^k$ componentwise: $(3, 7) \leq (7, 11)$.
- A partial order $(D, \leq)$ is a well-quasi-order (WQO), if every infinite sequence $a_0a_1a_2\cdots \in D^\omega$ has two positions $m < n$ with $a_m \leq a_n$. $(m, n)$ is called dickson pair.
Dickson’s Lemma

- Fix $k > 0$ and order $\mathbb{N}^k$ componentwise: $(3, 7) \leq (7, 11)$.
- A partial order $(D, \leq)$ is a well-quasi-order (WQO), if every infinite sequence $a_0 a_1 a_2 \cdots \in D^\omega$ has two positions $m < n$ with $a_m \leq a_n$. $(m, n)$ is called a dickson pair.
  - $(\mathbb{N}, \leq)$ is a WQO.
  - $(\mathbb{Z}, \leq)$ is not a WQO.
  - $(2^\mathbb{N}, \subseteq)$ is not a WQO.
Dickson’s Lemma

- Fix $k > 0$ and order $\mathbb{N}^k$ componentwise: $(3, 7) \leq (7, 11)$.
- A partial order $(D, \leq)$ is a well-quasi-order (WQO), if every infinite sequence $a_0a_1a_2\cdots \in D^\omega$ has two positions $m < n$ with $a_m \leq a_n$. $(m, n)$ is called dickson pair.

**Lemma (Dickson ’13)**

$(\mathbb{N}^k, \leq)$ is a WQO.
Dickson’s Lemma

- Fix $k > 0$ and order $\mathbb{N}^k$ componentwise: $(3, 7) \leq (7, 11)$.

- A partial order $(D, \leq)$ is a well-quasi-order (WQO), if every infinite sequence $a_0a_1a_2\cdots \in D^\omega$ has two positions $m < n$ with $a_m \leq a_n$. $(m, n)$ is called dickson pair.

Lemma (Dickson '13)

$(\mathbb{N}^k, \leq)$ is a WQO.

However, Dickson’s Lemma does not give any bound on length of infixes without dickson pairs.
Dickson’s Lemma

- Fix $k > 0$ and order $\mathbb{N}^k$ componentwise: $(3, 7) \leq (7, 11)$.
- A partial order $(D, \leq)$ is a well-quasi-order (WQO), if every infinite sequence $a_0a_1a_2\cdots \in D^\omega$ has two positions $m < n$ with $a_m \leq a_n$. $(m, n)$ is called dickson pair.

**Lemma (Dickson ’13)**

$(\mathbb{N}^k, \leq)$ is a WQO.

However, Dickson’s Lemma does not give any bound on length of infixes without dickson pairs. Indeed, there is no such bound:

$$(n)\ (n - 1)\ (n - 2)\ \cdots\ (0)$$
Waiting times vectors are special:

- either increment, or
- reset to zero.
Waiting times vectors are special:
- either increment, or
- reset to zero.

**Lemma**

Let $G$ be an RR game with $s$ vertices and $k$ RR conditions. There is a function $b(s, k) \in \mathcal{O}(2^{2s \cdot k + 2})$ such that every play infix of length $b(s, k)$ has a dickson pair.
Waiting times vectors are special:

- either increment, or
- reset to zero.

**Lemma**

Let $G$ be an RR game with $s$ vertices and $k$ RR conditions. There is a function $b(s, k) \in \mathcal{O}(2^{s \cdot k + 2})$ such that every play infix of length $b(s, k)$ has a dickson pair.

**Lemma (Czerwiński, Gogac, Kopczyński ’14)**

Lower bound: $2^{2k/2}$.
We have $\sigma$ with $\text{val}(\sigma, v) \leq \sum_{j \in [k]} sk2^k =: b_G$ for all $v \in W_0(G)$. 
Bounding the Waiting Times

We have $\sigma$ with $\text{val}(\sigma, v) \leq \sum_{j \in [k]} s_k 2^k =: b_G$ for all $v \in W_0(G)$.

**Lemma**

Let $\sigma$ be s.t. $\text{val}(\sigma, v) \leq b_G$ for all $v \in W_0(G)$. There is $\sigma'$ with $\text{val}(\sigma', v) \leq \text{val}(\sigma, v)$ for all $v$ that uniformly bounds the waiting times for every condition $j$ by $b_G + b(s, k - 1)$. 
We have $\sigma$ with $\text{val}(\sigma, v) \leq \sum_{j \in [k]} s^j 2^k =: b_\mathcal{G}$ for all $v \in W_0(\mathcal{G})$.

**Lemma**

Let $\sigma$ be s.t. $\text{val}(\sigma, v) \leq b_\mathcal{G}$ for all $v \in W_0(\mathcal{G})$. There is $\sigma'$ with $\text{val}(\sigma', v) \leq \text{val}(\sigma, v)$ for all $v$ that uniformly bounds the waiting times for every condition $j$ by $b_\mathcal{G} + b(s, k - 1)$. 

![Diagram](image-url)
Bounding the Waiting Times

We have $\sigma$ with $\text{val}(\sigma, v) \leq \sum_{j \in [k]} s k 2^k =: b_G$ for all $v \in W_0(G)$.

**Lemma**

Let $\sigma$ be s.t. $\text{val}(\sigma, v) \leq b_G$ for all $v \in W_0(G)$. There is $\sigma'$ with $\text{val}(\sigma', v) \leq \text{val}(\sigma, v)$ for all $v$ that uniformly bounds the waiting times for every condition $j$ by $b_G + b(s, k - 1)$.

($\bullet, \bullet$): dickson pair
We have $\sigma$ with $\text{val}(\sigma, v) \leq \sum_{j \in [k]} sk2^k =: b_G$ for all $v \in W_0(G)$.

**Lemma**

Let $\sigma$ be s.t. $\text{val}(\sigma, v) \leq b_G$ for all $v \in W_0(G)$. There is $\sigma'$ with $\text{val}(\sigma', v) \leq \text{val}(\sigma, v)$ for all $v$ that uniformly bounds the waiting times for every condition $j$ by $b_G + b(s, k - 1)$.

$(\bullet, \bullet)$: dickson pair

$P_j$
Bounding the Waiting Times

We have $\sigma$ with $\text{val}(\sigma, v) \leq \sum_{j \in [k]} sk2^k =: b_G$ for all $v \in W_0(G)$.

**Lemma**

Let $\sigma$ be s.t. $\text{val}(\sigma, v) \leq b_G$ for all $v \in W_0(G)$. There is $\sigma'$ with $\text{val}(\sigma', v) \leq \text{val}(\sigma, v)$ for all $v$ that uniformly bounds the waiting times for every condition $j$ by $b_G + b(s, k - 1)$. 

![Diagram showing waiting times]

$\text{wt}_j > b_G$
Mean-Payoff Games

Mean-payoff game: $G = (\mathcal{A}, w)$ with $w : E \rightarrow \{-W, \ldots, W\}$.

- Given $\rho = \rho_0 \rho_1 \rho_2 \cdots$ define value for
  - Player 0: $\nu_0(\rho) = \limsup_{n \to \infty} \frac{1}{n} \sum_{\ell=1}^{n} w(\rho_{\ell-1}, \rho_\ell)$

Theorem (Ehrenfeucht, Mycielski '79)
For every mean-payoff game there exist positional strategies $\sigma_{\text{opt}}$ for Player 0 and $\tau_{\text{opt}}$ for Player 1 and values $\nu(\nu)$ such that

1. every play $\rho \in \text{Beh}(\nu, \sigma_{\text{opt}})$ satisfies $\nu_0(\rho) \leq \nu(\nu)$, and
2. every play $\rho \in \text{Beh}(\nu, \tau_{\text{opt}})$ satisfies $\nu_1(\rho) \geq \nu(\nu)$.

Strategies and values are computable in pseudo-polynomial time.
Mean-Payoff Games

Mean-payoff game: $G = (A, w)$ with $w: E \rightarrow \{-W, \ldots, W\}$.

- Given $\rho = \rho_0 \rho_1 \rho_2 \cdots$ define value for
  - Player 0: $\nu_0(\rho) = \lim \sup_{n \to \infty} \frac{1}{n} \sum_{\ell=1}^{n} w(\rho_{\ell-1}, \rho_{\ell})$
  - Player 1: $\nu_1(\rho) = \lim \inf_{n \to \infty} \frac{1}{n} \sum_{\ell=1}^{n} w(\rho_{\ell-1}, \rho_{\ell})$

Theorem (Ehrenfeucht, Mycielski '79)

For every mean-payoff game there exist positional strategies $\sigma_{\text{opt}}$ for Player 0 and $\tau_{\text{opt}}$ for Player 1 and values $\nu(\nu)$ such that

1. every play $\rho \in \text{Beh}(v, \sigma_{\text{opt}})$ satisfies $\nu_0(\rho) \leq \nu(v)$, and
2. every play $\rho \in \text{Beh}(v, \tau_{\text{opt}})$ satisfies $\nu_1(\rho) \geq \nu(v)$.

Strategies and values are computable in pseudo-polynomial time.
Mean-Payoff Games

Mean-payoff game: $\mathcal{G} = (\mathcal{A}, w)$ with $w: E \to \{-W, \ldots, W\}$.

- Given $\rho = \rho_0\rho_1\rho_2 \cdots$ define value for
  - Player 0: $\nu_0(\rho) = \limsup_{n \to \infty} \frac{1}{n} \sum_{\ell=1}^{n} w(\rho_{\ell-1}, \rho_{\ell})$
  - Player 1: $\nu_1(\rho) = \liminf_{n \to \infty} \frac{1}{n} \sum_{\ell=1}^{n} w(\rho_{\ell-1}, \rho_{\ell})$

- $-W \leq \nu_1(\rho) \leq \nu_0(\rho) \leq W$
Mean-Payoff Games

Mean-payoff game: $G = (A, w)$ with $w : E \rightarrow \{-W, \ldots, W\}$.

- Given $\rho = \rho_0 \rho_1 \rho_2 \cdots$ define value for
  - Player 0: $\nu_0(\rho) = \limsup_{n \to \infty} \frac{1}{n} \sum_{\ell=1}^{n} w(\rho_{\ell-1}, \rho_{\ell})$
  - Player 1: $\nu_1(\rho) = \liminf_{n \to \infty} \frac{1}{n} \sum_{\ell=1}^{n} w(\rho_{\ell-1}, \rho_{\ell})$
- $-W \leq \nu_1(\rho) \leq \nu_0(\rho) \leq W$

**Theorem (Ehrenfeucht, Mycielski ’79)**

*For every mean-payoff game there exist positional strategies $\sigma_{\text{opt}}$ for Player 0 and $\tau_{\text{opt}}$ for Player 1 and values $\nu(\nu)$ such that*

1. *every play $\rho \in \text{Beh}(\nu, \sigma_{\text{opt}})$ satisfies $\nu_0(\rho) \leq \nu(\nu)$, and*
2. *every play $\rho \in \text{Beh}(\nu, \tau_{\text{opt}})$ satisfies $\nu_1(\rho) \geq \nu(\nu)$.*

*Strategies and values are computable in pseudo-polynomial time.*
Mean-Payoff Games

Mean-payoff game: \( G = (A, w) \) with \( w : E \to \{-W, \ldots, W\} \).

- Given \( \rho = \rho_0\rho_1\rho_2 \cdots \) define value for
  - Player 0: \( \nu_0(\rho) = \lim \sup_{n \to \infty} \frac{1}{n} \sum_{\ell=1}^{n} w(\rho_{\ell-1}, \rho_{\ell}) \)
  - Player 1: \( \nu_1(\rho) = \lim \inf_{n \to \infty} \frac{1}{n} \sum_{\ell=1}^{n} w(\rho_{\ell-1}, \rho_{\ell}) \)
- \(-W \leq \nu_1(\rho) \leq \nu_0(\rho) \leq W\)

**Theorem (Ehrenfeucht, Mycielski ’79)**

For every mean-payoff game there exist positional strategies \( \sigma_{\text{opt}} \) for Player 0 and \( \tau_{\text{opt}} \) for Player 1 and values \( \nu(\nu) \) such that

1. every play \( \rho \in \text{Beh}(\nu, \sigma_{\text{opt}}) \) satisfies \( \nu_0(\rho) \leq \nu(\nu) \), and
2. every play \( \rho \in \text{Beh}(\nu, \tau_{\text{opt}}) \) satisfies \( \nu_1(\rho) \geq \nu(\nu) \).

Strategies and values are computable in pseudo-polynomial time.

- \( \rho \in \text{Beh}(\nu, \sigma_{\text{opt}}) \cap \text{Beh}(\nu, \tau_{\text{opt}}) \) satisfies \( \nu_0(\rho) = \nu_1(\rho) = \nu(\nu) \)
Let \( t_{\text{max}_j} = \text{val}_G + b(s, k - 1) \).
Let \( t_{\max_j} = \text{val}_G + b(s, k - 1) \).
Let \( \mathcal{A} \) be DFA that keeps track of waiting vectors as long as each coordinate \( j \) is bounded by \( t_{\max_j} \) (sink state \( \perp \)).
From RR Games to Mean-Payoff Games

- Let $t_{\text{max}} = \text{val}_G + b(s, k - 1)$.
- Let $\mathcal{A}$ be DFA that keeps track of waiting vectors as long as each coordinate $j$ is bounded by $t_{\text{max}}$ (sink state $\perp$).
- Take cartesian product of $\mathcal{A}$ and $\mathcal{A}$.
Let $t_{\text{max}_j} = \text{val}_G + b(s, k - 1)$.

Let $\mathcal{A}$ be DFA that keeps track of waiting vectors as long as each coordinate $j$ is bounded by $t_{\text{max}_j}$ (sink state $\bot$).

Take cartesian product of $\mathcal{A}$ and $\mathcal{A}$.

Define $w$ by $w((v, \bot), (v', \bot)) = 1 + \sum_{j \in [k]} t_{\text{max}_j}$ and

$$w((v, (t_1, \ldots, t_k)), (v', (t'_1, \ldots, t'_k))) = \sum_{j \in [k]} t_j$$
Let \( t_{\text{max}} = \text{val}_G + b(s, k - 1) \).

Let \( \mathcal{A} \) be DFA that keeps track of waiting vectors as long as each coordinate \( j \) is bounded by \( t_{\text{max}} \) (sink state \( \bot \)).

Take cartesian product of \( \mathcal{A} \) and \( \mathcal{A} \).

Define \( w \) by \( w((v, \bot), (v', \bot)) = 1 + \sum_{j \in [k]} t_{\text{max}_j} \) and

\[
\begin{align*}
w((v, (t_1, \ldots, t_k)), (v', (t'_1, \ldots, t'_k))) &= \sum_{j \in [k]} t_j \\
\end{align*}
\]

Obtain mean-payoff game \( G' = (\mathcal{A} \times \mathcal{A}, w) \).
From RR Games to Mean-Payoff Games

- Let \( t_{\text{max}_j} = \text{val}_G + b(s, k - 1) \).
- Let \( \mathcal{A} \) be DFA that keeps track of waiting vectors as long as each coordinate \( j \) is bounded by \( t_{\text{max}_j} \) (sink state \( \perp \)).
- Take cartesian product of \( \mathcal{A} \) and \( \mathcal{A} \).
- Define \( w \) by
  \[
  w((v, \perp), (v', \perp)) = 1 + \sum_{j \in [k]} t_{\text{max}_j}
  \]
  and
  \[
  w((v, (t_1, \ldots, t_k)), (v', (t'_1, \ldots, t'_k))) = \sum_{j \in [k]} t_j
  \]
- Obtain mean-payoff game \( G' = (\mathcal{A} \times \mathcal{A}, w) \).

**Lemma**

Let \( \rho = \rho_0\rho_1\rho_2 \cdots \) be a play in \( G \), \( \rho' \) the corresponding one in \( G' \).

1. \( \rho' \) does not reach \( \perp \): \( \text{val}(\rho) = \nu_0(\rho') < 1 + \sum_{j \in [k]} t_{\text{max}_j} \).
Let \( t_{\text{max}_j} = \text{val}_G + b(s, k - 1) \).

Let \( \mathcal{A} \) be DFA that keeps track of waiting vectors as long as each coordinate \( j \) is bounded by \( t_{\text{max}_j} \) (sink state \( \perp \)).

Take cartesian product of \( \mathcal{A} \) and \( \mathcal{A} \).

Define \( w \) by
\[
w((v, \perp), (v', \perp)) = 1 + \sum_{j \in [k]} t_{\text{max}_j}
\]
\[
w((v, (t_1, \ldots, t_k)), (v', (t'_1, \ldots, t'_k))) = \sum_{j \in [k]} t_j
\]

Obtain mean-payoff game \( G' = (\mathcal{A} \times \mathcal{A}, w) \).

**Lemma**

Let \( \rho = \rho_0\rho_1\rho_2 \cdots \) be a play in \( G \), \( \rho' \) the corresponding one in \( G' \).

1. \( \rho' \) does not reach \( \perp \): \( \text{val}(\rho) = \nu_0(\rho') < 1 + \sum_{j \in [k]} t_{\text{max}_j} \).
2. \( \rho' \) reaches \( \perp \): \( \nu_0(\rho') = 1 + \sum_{j \in [k]} t_{\text{max}_j} \).
Proof of Main Theorem

RR game $\mathcal{G}$, mean-payoff game $\mathcal{G}'$.

- $\sigma$ uniformly bounds the waiting times in $\mathcal{G}$ by $t_{\max_j}$.
- Turn into $\sigma'$ for $\mathcal{G}'$ which never reaches $\bot$, bounds $\nu(\nu)$ strictly below $1 + \sum_{j \in [k]} t_{\max_j}$.

Claim: $\sigma_{\text{opt}}$ is optimal.

Assume $\hat{\sigma}_{\text{opt}}$ is strictly better.

Turn into $\hat{\sigma}'_{\text{opt}}$ for $\mathcal{G}'$, which is strictly better than $\sigma'_{\text{opt}}$.

Contradiction.
Proof of Main Theorem

RR game $\mathcal{G}$, mean-payoff game $\mathcal{G}'$.

- $\sigma$ uniformly bounds the waiting times in $\mathcal{G}$ by $t_{\max_j}$.
- Turn into $\sigma'$ for $\mathcal{G}'$ which never reaches $\bot$, bounds $\nu(v)$ strictly below $1 + \sum_{j \in [k]} t_{\max_j}$.

- $\sigma'_{\text{opt}}$ is optimal strategy for $\mathcal{G}'$ (never reaches $\bot$).
- Turn into $\sigma_{\text{opt}}$ for $\mathcal{G}$ with bounded waiting times (as $\sigma'_{\text{opt}}$ never reaches $\bot$).

Claim: $\sigma_{\text{opt}}$ is optimal.
Proof of Main Theorem

RR game $\mathcal{G}$, mean-payoff game $\mathcal{G}'$.

- $\sigma$ uniformly bounds the waiting times in $\mathcal{G}$ by $t_{\max j}$.
- Turn into $\sigma'$ for $\mathcal{G}'$ which never reaches $\bot$, bounds $\nu(v)$ strictly below $1 + \sum_{j \in [k]} t_{\max j}$.

- $\sigma'_{\text{opt}}$ is optimal strategy for $\mathcal{G}'$ (never reaches $\bot$).
- Turn into $\sigma_{\text{opt}}$ for $\mathcal{G}$ with bounded waiting times (as $\sigma'_{\text{opt}}$ never reaches $\bot$).

Claim: $\sigma_{\text{opt}}$ is optimal.

- assume $\hat{\sigma}_{\text{opt}}$ is strictly better.
- Turn into $\hat{\sigma}'_{\text{opt}}$ for $\mathcal{G}'$, which is strictly better than $\sigma'_{\text{opt}}$.

Contradiction.
Conclusion

Optimal strategies for RR games exist and can be effectively computed.

- But they are larger than arbitrary strategies.
- Is this avoidable or is there a price to pay for optimality?
- What about heuristics, approximation algorithms?

Same questions can be asked for other winning conditions and other combinations of quality measures.