The Complexity of Admissibility in $\omega$-Regular Games

R. Brenguier    J.-F. Raskin    M. Sassolas

Highlights 2013
September, 19-21
Controller synthesis

Mathieu Sassolas (UPEC)
Controller synthesis
Controller synthesis

controllers

strategies
Models of rationality

- Nash equilibria \(\Rightarrow\) no player has interest in deviating.
- Regret minimization \(\Rightarrow\) players prefer moves that would induce less regret had they known the other players strategy.
- Elimination of dominated strategies \(\Rightarrow\) players eliminate “bad” strategies

In all cases it is assumed everybody knows and uses the model of rationality.
Iterative elimination of dominated strategies

- What is a “bad” strategy? \( \sigma \) is strictly dominated by \( \sigma' \) if
  - for all profiles of the other players, if \( \sigma \) wins, so does \( \sigma' \).
  - for some profile of the other players, \( \sigma \) loses while \( \sigma' \) wins.
Iterative elimination of dominated strategies

- What is a “bad” strategy? $\sigma$ is strictly dominated by $\sigma'$ if
  - for all profiles of the other players, if $\sigma$ wins, so does $\sigma'$.
  - for some profile of the other players, $\sigma$ loses while $\sigma'$ wins.

Each player eliminates its dominated strategies.
Repeat until stabilized.
Iterative elimination of dominated strategies

- What is a “bad” strategy? $\sigma$ is strictly dominated by $\sigma'$ if
  - for all profiles of the other players, if $\sigma$ wins, so does $\sigma'$.
  - for some profile of the other players, $\sigma$ loses while $\sigma'$ wins.
- Each player eliminates its dominated strategies.
Iterative elimination of dominated strategies

- What is a “bad” strategy? \( \sigma \) is strictly dominated by \( \sigma' \) if
  - for all profiles of the other players, if \( \sigma \) wins, so does \( \sigma' \).
  - for some profile of the other players, \( \sigma \) loses while \( \sigma' \) wins.
- Each player eliminates its dominated strategies.
Iterative elimination of dominated strategies

- What is a “bad” strategy? $\sigma$ is strictly dominated by $\sigma'$ if
  - for all profiles of the other players, if $\sigma$ wins, so does $\sigma'$.
  - for some profile of the other players, $\sigma$ loses while $\sigma'$ wins.

- Each player eliminates its dominated strategies.
- Repeat until stabilized.
Our setting

- Turn based games on graphs.

Objective of player $i$: $WIN_i \subseteq V^\omega$. 
Our setting

- Turn based games on graphs.

- Objective of player $i$: $\text{WIN}_i \subseteq V^\omega$.

- Muller objectives:
  \[ \rho \in \text{WIN}_i \iff \text{Inf}(\rho) \in \mathcal{F}. \]
  \[ \leadsto \text{Generalizes Büchi and parity conditions.} \]

- Weak Muller objectives:
  \[ \rho \in \text{WIN}_i \iff \text{Occ}(\rho) \in \mathcal{F}. \]
  \[ \leadsto \text{Generalizes safety and reachability conditions.} \]
Admissibility

- **Dominance**: $\sigma'_i \succ_S^n \sigma_i$ if $\sigma'_i$ strictly dominates $\sigma_i$ w.r.t $S^n$.
- **Iterative admissibility**: $S^0_i = S_i$ and
  
  $$S^{n+1}_i := S^n_i \setminus \{\sigma_i \mid \exists \sigma'_i \in S^n_i, \sigma'_i \succ_S^n \sigma_i\}.$$ 

- Set of iteratively admissible strategies: $S^* = \bigcap_{n \in \mathbb{N}} S^n$
Admissibility

- **Dominance**: $\sigma'_i \succ_{S^n} \sigma_i$ if $\sigma'_i$ strictly dominates $\sigma_i$ w.r.t $S^n$.
- **Iterative admissibility**: $S_i^0 = S_i$ and 
  
  $S_i^{n+1} := S_i^n \setminus \{\sigma_i \mid \exists \sigma'_i \in S_i^n, \sigma'_i \succ_{S^n} \sigma_i\}$.
  
- **Set of iteratively admissible strategies**: $S^* = \bigcap_{n \in \mathbb{N}} S^n$

< Goal: compute $S^*$ or at least decide properties thereof.
Admissibility

- **Dominance:** $\sigma'_i \succ S^n \sigma_i$ if $\sigma'_i$ strictly dominates $\sigma_i$ w.r.t $S^n$.
- **Iterative admissibility:** $S^0_i = S_i$ and
  
  $$S_{i}^{n+1} := S_{i}^{n} \setminus \{\sigma_i \mid \exists \sigma'_i \in S_{i}^{n}, \sigma'_i \succ S^n \sigma_i\}.$$ 
- Set of iteratively admissible strategies: $S^* = \bigcap_{n \in \mathbb{N}} S^n$

Goal: compute $S^*$ or at least decide properties thereof.

Remark

$S^*$ is well defined and is reached after a finite number of iterations.

"Admissibility in Infinite Games" [Berwanger, STACS'07]
Admissibility

- **Dominance:** $\sigma'_i \succ^S \sigma_i$ if $\sigma'_i$ strictly dominates $\sigma_i$ w.r.t $S^n$.

- **Iterative admissibility:**
  
  $S^0_i = S_i$ and
  
  $S^{n+1}_i := S^n_i \setminus \{ \sigma_i \mid \exists \sigma'_i \in S^n_i, \sigma'_i \succ^S \sigma_i \}.$

- **Set of iteratively admissible strategies:** $S^* = \bigcap_{n \in \mathbb{N}} S^n$

**Goal:** compute $S^*$ or at least decide properties thereof.

**Remark**

$S^*$ is well defined and is reached after a finite number of iterations.

“Admissibility in Infinite Games” [Berwanger, STACS’07]

**Decision problems on $S^*$**

- **The winning coalition problem:** Given $W, L \subseteq P$, does there exists $\sigma_P \in S^*$ such that all players of $W$ win the game, and all players of $L$ lose.

- **The model-checking under admissibility problem:** Given $\varphi$ an LTL formula, is it the case that for any profile $\sigma_P \in S^*$, $Out(\sigma_P) \models \varphi$?
Values

Introduced in [Berwanger, STACS’07]

- If there is a winning strategy

→ admissible strategies are the winning ones.

- It is impossible to win

→ all strategies are admissible.

Remark

A player should never decrease its own value.

The value depends on $S^n$.

→ How to compute those values?
Values

Introduced in [Berwanger, STACS'07]

- If there is a winning strategy
  \[\implies\] admissible strategies are the winning ones.
- It is impossible to win
  \[\implies\] all strategies are admissible.
- Otherwise: it is possible to win, but only with the help of others
  \[\implies\] What are the admissible strategies in this case?

```
Val_1 = 0
Val_2 = 1
Val_3 = -1
```

A player should never decrease its own value. The value depends on $S_n$. How to compute those values?
Values

Introduced in [Berwanger, STACS'07]

- If there is a winning strategy: value 1.
  \[\Rightarrow\] admissible strategies are the winning ones.

- It is impossible to win: value \(-1\).
  \[\Rightarrow\] all strategies are admissible.

- Otherwise: it is possible to win, but only with the help of others: value 0.
  \[\Rightarrow\] What are the admissible strategies in this case?

\[\begin{align*}
\text{Val}_1 &= 0 \\
\text{Val}_2 &= 1 \\
\text{Val}_3 &= -1
\end{align*}\]
Values

Introduced in [Berwanger, STACS'07]

- If there is a winning strategy: value 1.
- admissible strategies are the winning ones.
- It is impossible to win: value −1.
- all strategies are admissible.
- Otherwise: it is possible to win, but only with the help of others: value 0.

What are the admissible strategies in this case?

Remark

- A player should never decrease its own value.
- The value depends on $S^n$.

How to compute those values?
Safety objectives: a local notion of dominance

- Objective: avoid *Bad* states
- Existence of a winning strategy depends only on:
  - the current state
  - *Bad* states visited
  - unfold the graph to keep this information
  - size: \(|V| \times 2^{|P|}\).

- In unfolded safety games the rule to never decrease one’s own value is sufficient for admissibility.
- The structure of the unfolding avoid explosion in complexity.

**Theorem**

*The winning coalition problem is PSPACE-complete for safety.*
Prefix-independent objectives

In general: the local condition is not sufficient

In case the value is 0, need to allow other players to help. "Help!"-state for $i$: a state where $j \neq i$ has several choices with value $\geq 0$ for $i$, while not changing the value for $j$.

Admissible strategies should be winning if the other players played fairly in those states.

$A_n$ recognizes $\text{Out}(S_n)$ with circuit winning condition.

In turn, $A_n$ is used to compute the values at the next step.
Prefix-independent objectives

In general: the local condition is not sufficient

- In case the value is 0, need to allow other players to help.
- "Help!"-state for $i$: a state where $j \neq i$ has several choices with value $\geq 0$ for $i$, while not changing the value for $j$.

Admissible strategies should be winning if the other players played fairly in those states.
Prefix-independent objectives

In general: the local condition is not sufficient

- In case the value is 0, need to allow other players to help.
- “Help!”-state for \( i \): a state where \( j \neq i \) has several choices with value \( \geq 0 \) for \( i \), while not changing the value for \( j \).

Admissible strategies should be winning if the other players played fairly in those states.

Give rises to automaton \( A_n \) recognizing \( \text{Out}(S^n) \) with circuit winning condition.
Prefix-independent objectives

- In general: the local condition is not sufficient

- In case the value is 0, need to allow other players to help.

- “Help!”-state for \( i \): a state where \( j \neq i \) has several choices with value \( \geq 0 \) for \( i \), while not changing the value for \( j \).

- Admissible strategies should be winning if the other players played fairly in those states.

- Give rises to automaton \( A_n \) recognizing \( Out(S^n) \) with circuit winning condition.

- In turn, \( A_n \) is used to compute the values at the next step.
Complexity for Objectives defined by Circuits

Theorem (Winning coalition problem)

- The winning coalition problem PSPACE-complete for circuits.
- The winning coalition problem with Büchi objectives is in $P^{NP}$.
- The winning coalition problem for weak circuit is PSPACE-complete.

Theorem (Model-checking under admissibility problem)

The model-checking under admissibility problem is PSPACE-complete for games where the winning condition of each player is given by a circuit condition.
Summary

- Automata representing all outcomes of admissible strategies.
- Algorithms with **tight complexity bounds** to compute the set of **all outcomes** of iteratively admissible strategies.
- Application to model-checking of LTL assuming all players follow rationality.

Future work

- Extension to quantitative games.
- Implementation.

Thank you
Summary

- Automata representing all outcomes of admissible strategies.
- Algorithms with tight complexity bounds to compute the set of all outcomes of iteratively admissible strategies.
- Application to model-checking of LTL assuming all players follow rationality.

Future work

- Extension to quantitative games.
- Implementation.
Summary

- Automata representing all outcomes of admissible strategies.
- Algorithms with tight complexity bounds to compute the set of all outcomes of iteratively admissible strategies.
- Application to model-checking of LTL assuming all players follow rationality.

Future work

- Extension to quantitative games.
- Implementation.

Thank you