Optimally Resilient Strategies in Pushdown Safety Games

Joint work with Daniel Neider (MPI-SWS) and Patrick Totzke (Liverpool)
Artwork by Paulina Zimmermann

Martin Zimmermann

University of Liverpool

August 2020
MFCS 2020
Reactive Synthesis

Automatically generate **correct-by-construction** systems.
Automatically generate correct-by-construction systems.
Automatically generate *correct-by-construction* systems.
Reactive Synthesis

Automatically generate correct-by-construction systems.
Reactive Synthesis

Automatically generate **correct-by-construction** systems.

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>![Image 29]</td>
<td>![Image 30]</td>
<td>![Image 31]</td>
<td>![Image 32]</td>
</tr>
<tr>
<td>![Image 33]</td>
<td>![Image 34]</td>
<td>![Image 35]</td>
<td>![Image 36]</td>
</tr>
<tr>
<td>![Image 37]</td>
<td>![Image 38]</td>
<td>![Image 39]</td>
<td>![Image 40]</td>
</tr>
</tbody>
</table>


Automatically generate **correct-by-construction** systems.
**Reactive Synthesis**

Automatically generate **correct-by-construction** systems.

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="blue_square.png" alt="Image" /></td>
<td><img src="red_square.png" alt="Image" /></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><img src="red_square.png" alt="Image" /></td>
<td><img src="red_square.png" alt="Image" /></td>
<td><img src="red_square.png" alt="Image" /></td>
<td><img src="red_square.png" alt="Image" /></td>
<td></td>
</tr>
<tr>
<td><img src="red_square.png" alt="Image" /></td>
<td><img src="red_square.png" alt="Image" /></td>
<td><img src="red_square.png" alt="Image" /></td>
<td><img src="red_square.png" alt="Image" /></td>
<td></td>
</tr>
<tr>
<td></td>
<td><img src="red_square.png" alt="Image" /></td>
<td><img src="red_square.png" alt="Image" /></td>
<td><img src="red_square.png" alt="Image" /></td>
<td></td>
</tr>
</tbody>
</table>
Automatically generate *correct-by-construction* systems.
Automatically generate correct-by-construction systems.
Automatically generate **correct-by-construction** systems.
Automatically generate **correct-by-construction** systems.
Automatically generate **correct-by-construction** systems.
Automatically generate correct-by-construction systems.

- Model the interaction between a system and its environment by an infinite-duration zero-sum game on graph. The winning condition captures a specification of the system.
- A winning strategy for the system player corresponds to an implementation satisfying the system specification.
Automatically generate **correct-by-construction** systems.

- Model the interaction between a system and its environment by an infinite-duration zero-sum game on graph. The winning condition captures a specification of the system.
- A winning strategy for the system player corresponds to an implementation satisfying the system specification.
Reactive Synthesis

Automatically generate correct-by-construction systems.

- Model the interaction between a system and its environment by an infinite-duration zero-sum game on graph. The winning condition captures a specification of the system.
- A winning strategy for the system player corresponds to an implementation satisfying the system specification.
Automatically generate correct-by-construction systems.

- Model the interaction between a system and its environment by an infinite-duration zero-sum game on graph. The winning condition captures a specification of the system.
- A winning strategy for the system player corresponds to an implementation satisfying the system specification.
Reactive Synthesis

Automatically generate correct-by-construction systems.

- Model the interaction between a system and its environment by an infinite-duration zero-sum game on graph. The winning condition captures a specification of the system.
- A winning strategy for the system player corresponds to an implementation satisfying the system specification.
Automatically generate correct-by-construction systems.

- Model the interaction between a system and its environment by an infinite-duration zero-sum game on graph. The winning condition captures a specification of the system.
- A winning strategy for the system player corresponds to an implementation satisfying the system specification.
Reactive Synthesis

Automatically generate correct-by-construction systems.

- Model the interaction between a system and its environment by an infinite-duration zero-sum game on graph. The winning condition captures a specification of the system.
- A winning strategy for the system player corresponds to an implementation satisfying the system specification.
Automatically generate **correct-by-construction** systems.

- Model the interaction between a system and its environment by an infinite-duration zero-sum game on graph. The winning condition captures a specification of the system.
- A winning strategy for the system player corresponds to an implementation satisfying the system specification.
Resilience in Safety Games

**Dallal, Tabuada and Neider**: Add disturbances edges to model non-antagonistic external influences.

Question: How many disturbances make the system player lose?
**Resilience in Safety Games**

**Dallal, Tabuada and Neider:** Add disturbances edges to model non-antagonistic external influences.

![Diagram of a system with disturbances](image)

**Question:** How many disturbances make the system player lose?
Resilience in Safety Games

Dallal, Tabuada and Neider: Add disturbances edges to model non-antagonistic external influences.

Question: How many disturbances make the system player lose?
Resilience in Safety Games

Dallal, Tabuada and Neider: Add disturbances edges to model non-antagonistic external influences.

Question: How many disturbances make the system player lose?
Resilience in Safety Games

Dallal, Tabuada and Neider: Add disturbances edges to model non-antagonistic external influences.

Question: How many disturbances make the system player lose?
Resilience in Safety Games

**Dallal, Tabuada and Neider:** Add disturbances edges to model non-antagonistic external influences.

Question: How many disturbances make the system player lose?
Resilience in Safety Games

Dallal, Tabuada and Neider: Add disturbances edges to model non-antagonistic external influences.

Question: How many disturbances make the system player lose?
Resilience in Safety Games

Dallal, Tabuada and Neider: Add disturbances edges to model non-antagonistic external influences.

Theorem (Dallal, Neider & Tabuada, 2016)

- A safety game with $n$ vertices has resilience values in $\{0, \cdots, n - 1\} \cup \{\omega + 1\}$.
- The resilience values and an optimally resilient strategy can be computed in polynomial time.
Systems with Infinite State Space

- Pushdown graphs are configuration graphs of pushdown automata.
- One-counter automata are pushdown automata with a single stack symbol (that can still test the stack for emptiness).
Systems with Infinite State Space

- Pushdown graphs are configuration graphs of pushdown automata.
- One-counter automata are pushdown automata with a single stack symbol (that can still test the stack for emptiness).
Theorem

- A pushdown safety game has resilience values in \( \{0, 1, 2, \cdots \} \cup \{\omega + 1\} \).
- An optimally resilient strategy always exists.

Lemma

The following problem is in \( \text{ExpTime} \): “Given a pushdown safety game \( \mathcal{G} \) with initial vertex \( v_I \), is \( r(v_I) = \omega + 1? \)”.

Note

\( \text{PSPACE} \) for one-counter safety games.
Building Blocks for an Algorithm

Theorem

- A pushdown safety game has resilience values in \( \{0, 1, 2, \cdots \} \cup \{ \omega + 1 \} \).
- An optimally resilient strategy always exists.

Lemma

The following problem is in \( 2\text{ExpTime} \): “Given a pushdown safety game \( G \) with initial vertex \( v_I \) and \( k \in \omega \) (encoded in binary), is \( r(v_I) = k \)?”.

Note

\( \text{ExpSpace} \) for one-counter safety games.
A Naive Algorithm

1: if \( r(v_I) = \omega + 1 \) then
2: \( \text{return } \omega + 1 \)
3: \( k = 0 \)
4: while true do
5: \( \text{if } r(v_I) = k \text{ then} \)
6: \( \text{return } k \)
7: else
8: \( k = k + 1 \)

- The algorithm terminates, as the only possible resilience values are \( \omega + 1 \) or some \( k \in \omega \).
- To obtain an upper bound on the running time, we need an upper bound on the resilience value of the initial vertex.
Upper Bounds on Resilience Values

Note that resilience values can be unbounded. Nevertheless, we can bound the resilience value of the initial vertex.

For a pushdown automaton $\mathcal{P}$ with $n$ states and $s$ stack symbols, define

$$b(\mathcal{P}) = n \cdot h(\mathcal{P}) \cdot s^{h(\mathcal{P})}$$

with

$$h(\mathcal{P}) = n \cdot s \cdot 2^{n+1} + 1$$

**Lemma**

Let $\mathcal{G}$ be a pushdown safety game with initial vertex $v_I$. If $r(v_I) \neq \omega + 1$, then $r(v_I) < b(\mathcal{P})$, where $\mathcal{P}$ is the automaton underlying $\mathcal{G}$. 
An Improved Algorithm

1: if \( r(v_I) = \omega + 1 \) then
2:     return \( \omega + 1 \)
3: for \( k = 0 \) to \( b(P) \) do
4:     if \( r(v_I) = k \) then
5:         return \( k \)
An Improved Algorithm

1: if \( r(v_I) = \omega + 1 \) then
2: return \( \omega + 1 \)
3: for \( k = 0 \) to \( b(\mathcal{P}) \) do
4: if \( r(v_I) = k \) then
5: return \( k \)

Theorem
The following problem can be solved in triply-exponential time: “Given a pushdown safety game \( \mathcal{G} \) with initial vertex \( v_I \), determine the resilience value of \( v_I \).” If yes, an \( r(v_I) \)-resilient strategy from \( v_I \) can be computed in triply-exponential time.
An Improved Algorithm

1: if \( r(v_I) = \omega + 1 \) then
2: \hspace{1em} return \( \omega + 1 \)
3: for \( k = 0 \) to \( b(P) \) do
4: \hspace{1em} if \( r(v_I) = k \) then
5: \hspace{2em} return \( k \)

Theorem
The following problem can be solved in polynomial space: “Given a one-counter safety game \( \mathcal{G} \) with initial vertex \( v_I \), determine the resilience value of \( v_I \).”

Note
No strategy computed.
Conclusion

Also in the paper/arXiv version:

1. An outlook on resilient strategies in pushdown reachability games (new resilience values appear).
2. A new result on optimal strategies in one-counter reachability games (without disturbance edges).
3. Lower bounds on computational complexity and on the resilience value of the initial vertex.
Conclusion

Also in the paper/arXiv version:

1. An outlook on resilient strategies in pushdown reachability games (new resilience values appear).
2. A new result on optimal strategies in one-counter reachability games (without disturbance edges).
3. Lower bounds on computational complexity and on the resilience value of the initial vertex.

Open problems:

1. Extension to more expressive winning conditions.
3. Computing optimally resilient strategies for one-counter safety games in polynomial space.
The Last Slide

Thank you for watching.

Daniel Neider: neider@mpi-sws.org
Patrick Totzke: totzke@liverpool.ac.uk
Martin Zimmermann: martin.zimmermann@liverpool.ac.uk