Adversarial Formal Semantics of Attack Trees and Related Problems

Thomas Brihaye, Sophie Pinchinat, Alexandre Terefenko

Université de Mons
Université de Rennes

July 4, 2023
1 Introduction
   - The attack tree model
   - Related work
2 Semantics for attack tree
3 Results on decision problems
A thief wants to steal some document in a safe of a building without being seen.
The entrance in the building
An example of an attack tree

- Thief
- Door 1
- Door 2
- Camera
- Key
- Locked door
- Safe
- Guard
- Cross first door unseen
- Cross second door unseen
- Deactivate camera
- Find key
- Enter room 2
Related work

**Different syntax of attack trees:**

**Different semantics for attack trees:**

1 Introduction

2 Semantics for attack tree
   - Syntax
   - Path semantics
   - Strategy semantics

3 Results on decision problems
Syntax of an attack tree

Syntax of an attack tree

Syntax of an attack tree (no precondition)

**Definition**

An *attack tree* $\tau$ is:

- a Boolean formula $\phi$ over a set of proposition $\text{Prop}$,
- an expression $OP(\tau_1, \ldots, \tau_n)$ where $OP \in \{\text{OR}, \text{AND}, \text{SAND}\}$ and $\tau_1, \ldots, \tau_n$ are attack trees.

```
   φ₅
   /   \
 /     \   
 /       \
φ₁      φ₂     φ₃
     / \
    /   \   
   φ₄   φ₃
```
Let $S = (S, \rightarrow, \text{val})$ be a transition system with $\text{val} : S \rightarrow \text{PROP}$ a valuation function. We denote $\Pi(S)$ the set of all paths over $S$.

**Definition**

Paths($\tau$)$_S$ is inductively defined as follow:

- $\text{Paths}(\phi)_S = \{ s_0s_1...s_n \in \Pi(S) | s_n \models \phi \}$,
- For $\text{Paths}(\text{OR}(\tau_1, ..., \tau_n))_S$, we use the **union** of the semantics,
- For $\text{Paths}(\text{SAND}(\tau_1, ..., \tau_n))_S$, we use the **synchronised concatenation** of the semantics,
- For $\text{Paths}(\text{AND}(\tau_1, ..., \tau_n))_S$, we use the **merge** of the semantics.
Example: path semantics

- Thief
- Door 1
- Camera
- Locked door
- Door 2
- Guard
- Key
- Safe
- Safe
- D1 \land \neg \text{seen}
- D2 \land \neg \text{seen}
- C
- K
- R2

Semantics for attack tree
Path semantics
The entrance in the building

\begin{align*}
\{D1, \text{seen}\} & \quad (d_1, d_1) \quad (o, d_1) \quad (d_2, d_1) \quad \{D2\} \\
\{D1\} & \quad (d_1, m) \quad (o, m) \quad (d_2, m) \quad \{D2\} \\
\{D1\} & \quad (d_1, d_2) \quad (o, d_2) \quad (d_2, d_2) \quad \{D2, \text{seen}\}
\end{align*}
### Intuition for a strategy semantics

<table>
<thead>
<tr>
<th>Paths semantics</th>
<th>Strategy semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transition system</td>
<td>Game arena</td>
</tr>
<tr>
<td>Paths</td>
<td>Strategies</td>
</tr>
</tbody>
</table>

\[
Paths(\phi) = \{s_0...s_n \in \Pi(S) | s_n \models \phi \}
\]

\[
Strat(\phi) \text{ winning strategies for the reachability game defined by } \phi
\]

For an attack tree \( \tau \), \( Strat(\tau) \) denotes all winning attacking strategies.
Example: strategy semantics

Key:
- Thief
- Camera
- Guard
- Door 1
- Door 2
- Key
- Safe
- Locked door
- Safe

Semantics for attack tree

Strategy semantics

R2

D1 \land \neg seen

D2 \land \neg seen

C

K

Terefenko

Adversarial Formal Semantics of Attack Trees and Rel.
Problems with a compositional semantics

\[
\begin{align*}
(d_1, m) & \quad \rightarrow \quad (o, m) \\
(d_2, m) & \quad \rightarrow \quad (o, m) \\
(d_1, d_1) & \quad \rightarrow \quad (o, d_1) \\
(d_2, d_1) & \quad \rightarrow \quad (o, d_1) \\
(d_1, d_2) & \quad \rightarrow \quad (o, d_2) \\
(d_2, d_2) & \quad \rightarrow \quad (o, d_2)
\end{align*}
\]

\[
\begin{align*}
(d_1, m) & \quad \rightarrow \quad (d_1, m) \\
(d_2, m) & \quad \rightarrow \quad (d_2, m) \\
(d_1, d_1) & \quad \rightarrow \quad (d_1, d_1) \\
(d_2, d_1) & \quad \rightarrow \quad (d_2, d_1) \\
(d_1, d_2) & \quad \rightarrow \quad (d_1, d_2) \\
(d_2, d_2) & \quad \rightarrow \quad (d_2, d_2)
\end{align*}
\]

\[
\begin{align*}
\{D1, \text{seen}\} & \quad \rightarrow \quad \{D1, \text{seen}\} \\
\{D2\} & \quad \rightarrow \quad \{D2\} \\
\{D1\} & \quad \rightarrow \quad \{D1\} \\
\{D2\} & \quad \rightarrow \quad \{D2\} \\
\{D1\} & \quad \rightarrow \quad \{D1\} \\
\{D2, \text{seen}\} & \quad \rightarrow \quad \{D2, \text{seen}\}
\end{align*}
\]
Strategy semantics

Definition

The strategy semantics of an attack tree $\tau$ is the set of all trees $\sigma$ respecting the two following conditions:

- $\sigma$ denotes a strategy
- every branch of $\sigma$ is a path in $\text{Paths}(\tau)$
1 Introduction

2 Semantics for attack tree

3 Results on decision problems
Considered decision problems

- **Membership problem**
  \[ x \in \llbracket \tau \rrbracket \ ? \]

- **Non-Emptiness problem**
  \[ \llbracket \tau \rrbracket \neq \emptyset \ ? \]
### Results summary

<table>
<thead>
<tr>
<th></th>
<th>Paths semantics</th>
<th>Strategy semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Membership Problem</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-Emptiness Problem</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- NP-complete if preconditions for leaves.\(^1\)

---

### Results summary

<table>
<thead>
<tr>
<th>Membership Problem</th>
<th>Paths semantics</th>
<th>Strategy semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>Non-Emptiness Problem</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- NP-complete if preconditions for leaves.\(^1\)
- Without preconditions: a backward induction over the input path can solve the problem in polynomial time.

---

## Results summary

<table>
<thead>
<tr>
<th></th>
<th>Paths semantics</th>
<th>Strategy semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Membership Problem</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>Non-Emptiness Problem</td>
<td>NP-complete$^2$</td>
<td></td>
</tr>
</tbody>
</table>

Results on decision problems

<table>
<thead>
<tr>
<th>Paths semantics</th>
<th>Strategy semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Membership Problem</td>
<td>( P )</td>
</tr>
<tr>
<td>Non-Emptiness Problem</td>
<td>NP-complete</td>
</tr>
</tbody>
</table>

Hardness: \( \exists x_1 \exists x_2 \exists x_3, x_1 \land (x_2 \lor x_3) \land (\neg x_2 \lor x_3) \)

\[
\begin{align*}
\{ P_1 \} & \quad \{ P_2 \} & \quad \{ P_2, P_3 \} \\
\{\text{start}\} & \quad p_1 & \quad p_2 & \quad p_3 \\
\neg p_1 & \quad \neg p_2 & \quad \neg p_3 \\
\emptyset & \quad \{ P_3 \} & \quad \emptyset \\
\end{align*}
\]
### Results summary

<table>
<thead>
<tr>
<th></th>
<th>Paths semantics</th>
<th>Strategy semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Membership Problem</td>
<td>P</td>
<td>(\text{PSPACE})-complete</td>
</tr>
<tr>
<td>Non-Emptiness Problem</td>
<td>NP-complete</td>
<td>(\text{PSPACE})-complete</td>
</tr>
</tbody>
</table>

**Membership:**

- Semantics non-empty \(\iff\) existence of not too long strategies
- construct an alternating Turing machine:
  - Guess a play \(\pi\) (AP)
  - check whether \(\pi \in \text{Paths}(\tau)\)
Results summary

<table>
<thead>
<tr>
<th></th>
<th>Paths semantics</th>
<th>Strategy semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Membership Problem</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>Non-Emptiness Problem</td>
<td>NP-complete</td>
<td>PSPACE-complete</td>
</tr>
</tbody>
</table>

**Hardness:** \( \exists x_1 \forall x_2 \exists x_3, x_1 \land (x_2 \lor x_3) \land (\neg x_2 \lor x_3) \)
### Results summary

<table>
<thead>
<tr>
<th></th>
<th>Paths semantics</th>
<th>Strategy semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Membership Problem</td>
<td>$P$</td>
<td>$\text{coNP}$-complete</td>
</tr>
<tr>
<td>Non-Emptiness Problem</td>
<td>$\text{NP}$-complete</td>
<td>$\text{PSPACE}$-complete</td>
</tr>
</tbody>
</table>
Results on decision problems

Results summary

<table>
<thead>
<tr>
<th></th>
<th>Paths semantics</th>
<th>Strategy semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Membership Problem</td>
<td>$P$</td>
<td>$\text{coNP}$-complete</td>
</tr>
<tr>
<td>Non-Emptiness Problem</td>
<td>$\text{NP}$-complete</td>
<td>$\text{PSPACE}$-complete</td>
</tr>
</tbody>
</table>

**Hardness:** $\forall x_1 \forall x_2 \forall x_3, x_1 \land (x_2 \lor x_3) \land (\neg x_2 \lor x_3)$
## Results summary

<table>
<thead>
<tr>
<th></th>
<th>Paths semantics</th>
<th>Strategy semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Membership Problem</td>
<td>$P$</td>
<td>$\text{coNP}$-complete</td>
</tr>
<tr>
<td>Non-Emptiness Problem</td>
<td>$\text{NP}$-complete</td>
<td>$\text{PSPACE}$-complete</td>
</tr>
</tbody>
</table>

### In conclusion


Ross Horne, Sjouke Mauw, and Alwen Tiu, Semantics for specialising attack trees based on linear logic, Fundamenta Informaticae 153 (2017), no. 1-2, 57-86.
