Responsibility in verification

Corto Mascle

Joint work with Christel Baier, Florian Funke, Simon Jantsch, Stefan Kiefer, Karoliina Lehtinen

October 28th 2021

What is responsibility?

Consider a system with some **agents**, each performing **actions** resulting in **events**.

What is responsibility?

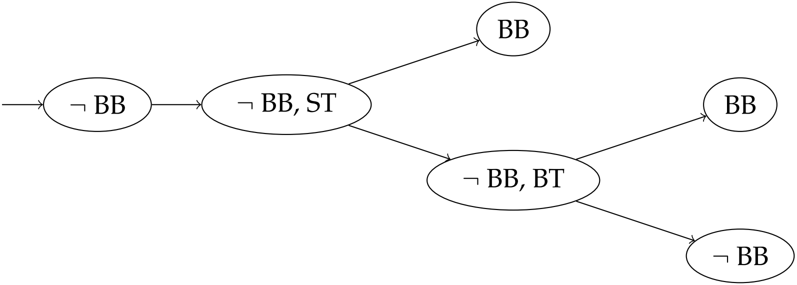
Consider a system with some **agents**, each performing **actions** resulting in **events**.
Chockler, Halpern and Kupferman proposed a notion of responsibility based on counterfactuality:
Agent A is responsible for event E if, had A acted differently, E would not have happened.

What is responsibility?

Consider a system with some **agents**, each performing **actions** resulting in **events**.
Chockler, Halpern and Kupferman proposed a notion of responsibility based on counterfactuality:
Agent A is responsible for event E if, had A acted differently, E would not have happened.
How do we distribute responsibility fairly?

Previous work

Their recurring example involves two characters, Suzy and Billy, throwing stones at a glass bottle. We use three variables, BB (the bottle is broken), ST (Suzy throws a stone) and BT (Billy throws a stone).



Previous work

A specification is given in CTL.

Previous work

A specification is given in CTL.

The responsibility is evaluated on states with respect to atomic propositions:

Previous work

A specification is given in CTL.

The responsibility is evaluated on states with respect to atomic propositions:

The degree of responsibility of state s with respect to a is $\frac{1}{m}$, where m is the minimal size of a set of states containing s such that flipping the value of a in those states makes the system fail the specification.

Voting

An intuitive situation in which we assign responsibility in a "human" way is after a vote.

Voting

An intuitive situation in which we assign responsibility in a "human" way is after a vote.

Responsibility is often understood either in a binary way (responsible or not), or in a weighted way (we distribute responsibility).

Shapley values

We fix a set of players $\{1, \ldots, N\}$, and we note $\mathcal{P}(N)$ the set of partitions of $\{1, \ldots, N\}$.

Definition

A *coalition* is a pair (T, P) with $P \in \mathcal{P}(N)$ and $T \in P$. The set of coalitions is denoted $\mathcal{C}(N)$.

Shapley values

We fix a set of players $\{1, \ldots, N\}$, and we note $\mathcal{P}(N)$ the set of partitions of $\{1, \ldots, N\}$.

Definition

A *coalition* is a pair (T, P) with $P \in \mathcal{P}(N)$ and $T \in P$. The set of coalitions is denoted $\mathcal{C}(N)$.

Definition

A *coalitional game* is a function $\nu : \mathcal{C}(N) \to \mathbb{R}$.

Shapley values

We fix a set of players $\{1, \ldots, N\}$, and we note $\mathcal{P}(N)$ the set of partitions of $\{1, \ldots, N\}$.

Definition

A *coalition* is a pair (T, P) with $P \in \mathcal{P}(N)$ and $T \in P$. The set of coalitions is denoted $\mathcal{C}(N)$.

Definition

A *coalitional game* is a function $\nu : \mathcal{C}(N) \to \mathbb{R}$.

We focus on games such that $\nu(T, P)$ only depends on T.

Cooperative games and the Shapley value

A function:

$$\nu : 2^A \to \mathbb{R}$$

defines a **cooperative game** (we assume $\nu(\varnothing) = 0$).

Cooperative games and the Shapley value

A function:

$$\nu : 2^A \to \mathbb{R}$$

defines a **cooperative game** (we assume $\nu(\varnothing) = 0$).

The number $\nu(C)$, with $C \subseteq A$, describes the **value of coalition** C.

Cooperative games and the Shapley value

A function:

$$\nu : 2^A \to \mathbb{R}$$

defines a **cooperative game** (we assume $\nu(\varnothing) = 0$).

The number $\nu(C)$, with $C \subseteq A$, describes the **value of coalition** C.

Given an **ordering** π of A, let $\pi_{\geq a} = \{i \in A \mid \pi(i) \geq \pi(a)\}$.

Cooperative games and the Shapley value

A function:

$$\nu : 2^A \to \mathbb{R}$$

defines a **cooperative game** (we assume $\nu(\varnothing) = 0$).

The number $\nu(C)$, with $C \subseteq A$, describes the **value of coalition** C.

Given an **ordering** π of A, let $\pi_{\geq a} = \{i \in A \mid \pi(i) \geq \pi(a)\}$.

Cooperative games and the Shapley value

A function:

$$\nu : 2^A \to \mathbb{R}$$

defines a **cooperative game** (we assume $\nu(\varnothing) = 0$).

The number $\nu(C)$, with $C \subseteq A$, describes the **value of coalition** C.

Given an **ordering** π of A, let $\pi_{\geq a} = \{i \in A \mid \pi(i) \geq \pi(a)\}$.

The **Shapley value** is defined as:

$$\mathrm{Sh}(a) = \frac{1}{n!} \sum_{\pi \in \Pi_n} \nu(\pi_{\geq a}) - \nu(\pi_{\geq a} \setminus \{a\})$$

Shapley values

The Shapley value function is the only one satisfying the following conditions.

1. **Efficiency** $\sum_{i=0}^{N} Sh_i(\nu) = \nu(\{1,\ldots,N\},\{\{1,\ldots,N\}\})$
2. **Symmetry** Renaming players does not affect their rewards.
3. **Additivity** For all games μ, ν, and $C \in \mathbb{R}$, $Sh_i(C\mu + \nu) = CSh_i(\mu) + Sh_i(\nu)$, i.e., Sh_i is a linear function.
4. **Null-Player Axiom** If for all $(T,P) \in \mathcal{C}(N)$, $\nu(T \cup \{i\}, P_{T\leftarrow i}) = \nu(T,P)$ then $Sh_i(\nu) = 0$.

Simple games

A (monotone) **value** function:

$$\nu : 2^A \to \{0, 1\}$$

defines a **simple cooperative game**.

Simple games

A (monotone) **value** function:

$$\nu : 2^A \to \{0, 1\}$$

defines a **simple cooperative game**.

Given an **ordering** π of A, let $\pi_{\geq a} = \{i \in A \mid \pi(i) \geq \pi(a)\}$.

Simple games

A (monotone) **value** function:

$$\nu : 2^A \to \{0, 1\}$$

defines a **simple cooperative game**.

Given an **ordering** π of A, let $\pi_{\geq a} = \{i \in A \mid \pi(i) \geq \pi(a)\}$.

Agent $a \in A$ is **decisive** for ordering $\pi \in \Pi_n$ if:

$$\nu(\pi_{\geq a}) = 1 \quad \text{and} \quad \nu(\pi_{\geq a} \setminus \{a\}) = 0$$

Simple games

A (monotone) **value** function:

$$\nu : 2^A \to \{0, 1\}$$

defines a **simple cooperative game**.

Given an **ordering** π of A, let $\pi_{\geq a} = \{i \in A \mid \pi(i) \geq \pi(a)\}$.

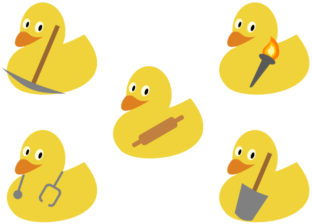
Agent $a \in A$ is **decisive** for ordering $\pi \in \Pi_n$ if:

$$\nu(\pi_{\geq a}) = 1 \quad \text{and} \quad \nu(\pi_{\geq a} \setminus \{a\}) = 0$$

$$\mathrm{Sh}(a) = \frac{1}{n!} |\{\pi \in \Pi_n \mid a \text{ is decisive for } \pi\}|$$

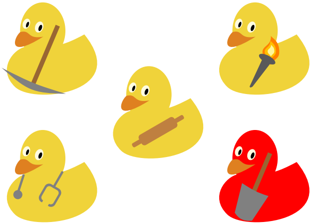
Process

- $\rightarrow$ We start with a system with agents $\{1, \ldots, n\}$.
- $\rightarrow$ Agents stop cooperating one by one in a random order.



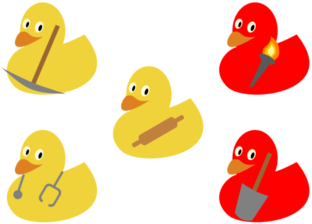
Process

- $\rightarrow$ We start with a system with agents $\{1, \ldots, n\}$.
- $\rightarrow$ Agents stop cooperating one by one in a random order.



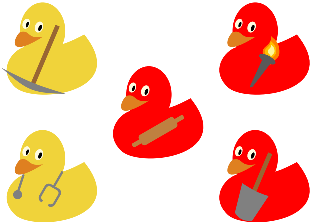
Process

- $\rightarrow$ We start with a system with agents $\{1, \ldots, n\}$.
- $\rightarrow$ Agents stop cooperating one by one in a random order.



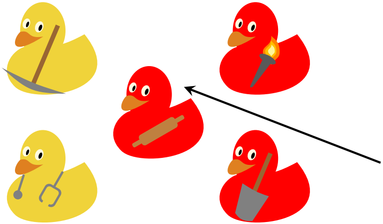
Process

- We start with a system with agents $\{1, \ldots, n\}$.
- Agents stop cooperating one by one in a random order.
- At some point the system stops working



Process

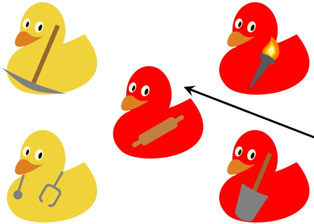
- $\rightarrow$ We start with a system with agents $\{1, \ldots, n\}$.
- $\rightarrow$ Agents stop cooperating one by one in a random order.
- $\rightarrow$ At some point the system stops working



The agent who just left the team is called decisive.

Process

- $\rightarrow$ We start with a system with agents $\{1, \ldots, n\}$.
- $\rightarrow$ Agents stop cooperating one by one in a random order.
- $\rightarrow$ At some point the system stops working



The agent who just left the team is called decisive.

The importance of an agent is its probability to be decisive.

LTL model checking and importance of states

LTL is a logics designed to express properties of infinite words.

$$\varphi ::= a \mid \varphi \wedge \varphi \mid \neg\varphi \mid X\varphi \mid \varphi U \varphi \mid G\varphi \mid F\varphi$$

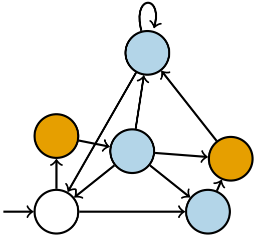
For instance, GFa expresses that at all positions of the word there is a further position at which a is true.

LTL model checking and importance of states

Given a Kripke structure K and an LTL formula φ.

LTL model checking and importance of states

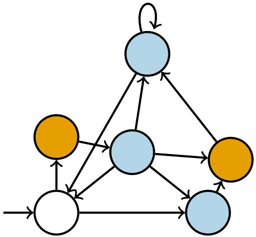
Given a Kripke structure K and an LTL formula φ.



$$\varphi = GF \bullet$$

"infinitely often" •

Given a Kripke structure K and an LTL formula φ.



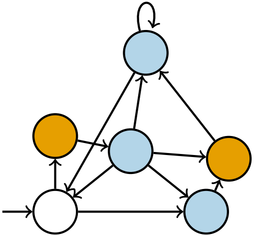
Which states of K are important for φ?

$$\varphi = GF$$

"infinitely often"

LTL model checking and importance of states

Given a Kripke structure K and an LTL formula φ.



Which states of K are important for φ?

A state of K is (more) important for φ if **its nondeterminism matters (more)**.

$$\varphi = GF$$

“infinitely often”

Formalizing importance for LTL

Given a **Kripke structure** K with states S, an LTL formula φ

Formalizing importance for LTL

Given a **Kripke structure** K with states S, an LTL formula φ and a set of states $C \subseteq S$, define:

Formalizing importance for LTL

Given a **Kripke structure** K with states S, an LTL formula φ and a set of states $C \subseteq S$, define:

> the **LTL-game** $\mathcal{G}(C)$ over K under partition $C, S \setminus C$ and **winning objective** φ.

Formalizing importance for LTL

Given a **Kripke structure** K with states S, an LTL formula φ and a set of states $C \subseteq S$, define:

> the **LTL-game** $\mathcal{G}(C)$ over K under partition $C, S \setminus C$ and **winning objective** φ.
>
> The system works if the player owning C wins the game.

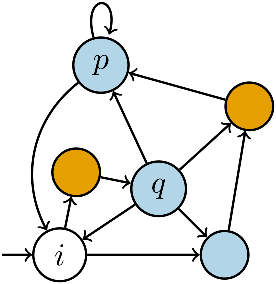
Formalizing importance for LTL

Given a **Kripke structure** K with states S, an LTL formula φ and a set of states $C \subseteq S$, define:

> the **LTL-game** $\mathcal{G}(C)$ over K under partition $C, S \setminus C$ and **winning objective** φ.
>
> The system works if the player owning C wins the game.

$$\mathcal{I}(q) = \frac{1}{|S|!} |\{\pi \in \Pi_S \mid q \text{ is decisive for } \pi\}|$$

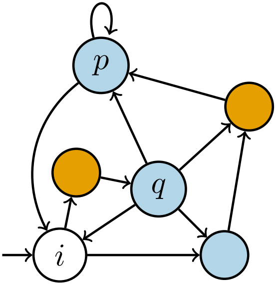
Example



$$\varphi = GF$$

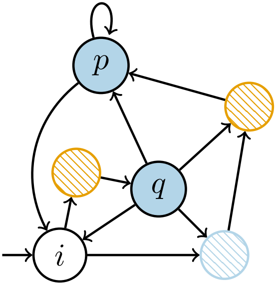
Example

Deterministic states have importance 0.



$$\varphi = GF$$

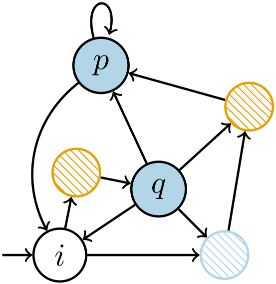
Example



$$\varphi = GF \bullet$$

Deterministic states have importance 0.

Example

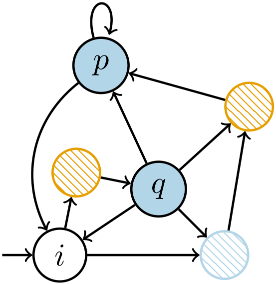


$$\varphi = GF\,\bullet$$

Deterministic states have importance 0.

Controlling p or $\{q, i\}$ wins.

Example



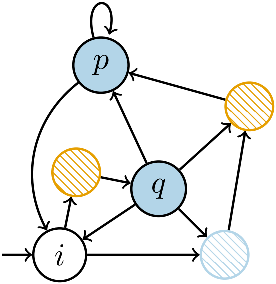
$$\varphi = GF\,\bullet$$

Deterministic states have importance 0.

Controlling p or $\{q, i\}$ wins.

$$\mathcal{I}(p) = \frac{1}{|S|!} |\{\pi \in \Pi_S \mid p \textbf{ is decisive for } \pi\}|$$

Example



$$\varphi = GF\,\bullet$$

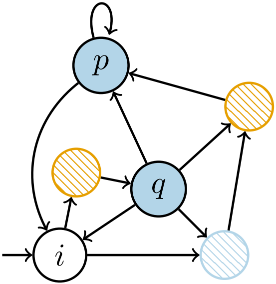
Deterministic states have importance 0.

Controlling p or $\{q, i\}$ wins.

$$\mathcal{I}(p) = \frac{1}{|S|!} |\{\pi \in \Pi_S \mid p \textbf{ is decisive for } \pi\}|$$

$$= \frac{1}{|S|!} |\{\pi \in \Pi_S \mid p \textbf{ is after } q \textbf{ or } i \textbf{ in } \pi\}|$$

Example



$$\varphi = GF\,●$$

Deterministic states have importance 0.

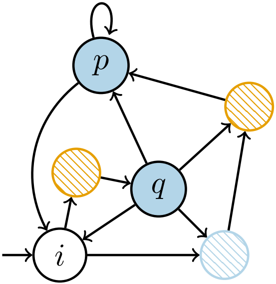
Controlling p or $\{q, i\}$ wins.

$$\mathcal{I}(p) = \frac{1}{|S|!}|\{\pi \in \Pi_S \mid p \textbf{ is decisive for } \pi\}|$$

$$= \frac{1}{|S|!}|\{\pi \in \Pi_S \mid p \textbf{ is after } q \textbf{ or } i \textbf{ in } \pi\}|$$

$$= \frac{2}{3}$$

Example



$$\varphi = GF \bullet$$

Deterministic states have importance 0.

Controlling p or $\{q, i\}$ wins.

$$\mathcal{I}(p) = \frac{1}{|S|!}|\{\pi \in \Pi_S \mid p \textbf{ is decisive for } \pi\}|$$

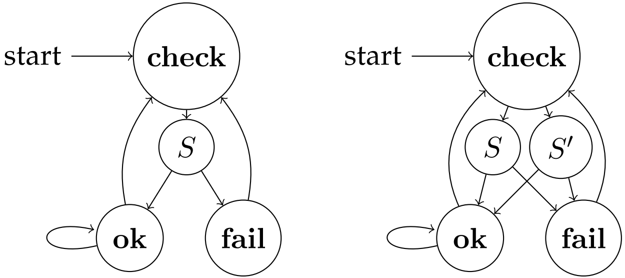
$$= \frac{1}{|S|!}|\{\pi \in \Pi_S \mid p \textbf{ is after } q \textbf{ or } i \textbf{ in } \pi\}|$$

$$= \frac{2}{3}$$

$$\mathcal{I}(i) = \mathcal{I}(q) = 1/6$$

Comparison

A server is tested by sending requests. If the server fails to answer, it is tested again, otherwise the system may wait before testing again.



$\varphi = GF\,\textbf{check} \wedge FG\,\neg\textbf{fail}$

We obtain an importance of $\frac{1}{2}$ for S and ok in the first example, $\frac{1}{2}$ for ok and $\frac{1}{6}$ for $check$, S and S' in the other.

Complexity of determining importance

Value problem. Given $C \subseteq S$:

Is C enough to make the system work?

$\to$ this requires **solving the game**.

Complexity of determining importance

Value problem. Given $C \subseteq S$:

Is C enough to make the system work?

$\rightarrow$ this requires **solving the game**.

Usefulness problem. Given $q \in S$:

Decide $\mathcal{I}(q) > 0$?

Complexity of determining importance

Value problem. Given $C \subseteq S$:

Is C enough to make the system work?

$\rightarrow$ this requires **solving the game**.

Usefulness problem. Given $q \in S$:

Decide $\mathcal{I}(q) > 0$?

Importance threshold problem. Given $q \in S$, $\gamma \in \mathbb{Q}$:

Decide $\mathcal{I}(q) > \gamma$?

Complexity of determining importance

Value problem. Given $C \subseteq S$:

Is C enough to make the system work?

$\rightarrow$ this requires **solving the game**.

Usefulness problem. Given $q \in S$:

Decide $\mathcal{I}(q) > 0$?

Importance threshold problem. Given $q \in S$, $\gamma \in \mathbb{Q}$:

Decide $\mathcal{I}(q) > \gamma$?

Importance computation problem. Given $q \in S$:

compute $|S|! \cdot \mathcal{I}(q)$

Complexity of determining importance

TABLE I
A SUMMARY OF THE RESULTS ON THE COMPLEXITY OF THE VALUE, USEFULNESS AND IMPORTANCE PROBLEMS FOR VARIOUS TYPES OF SPECIFICATIONS.

	Büchi	**Rabin**	**Streett**	**Parity**	**Explicit Muller**
Value	P	NP	coNP	$\in$ NP $\cap$ coNP	P
Usefulness	NP	Σ_2^P	Σ_2^P	NP	NP
Importance	#P	$\#P^{NP}$	$\#P^{NP}$	#P	#P

	Emerson-Lei	**LTL**	**2-turn CTL**	**Concurrent CTL**
Value	PSpace	2ExpTime	Σ_2^P	$\in$ ExpTime
Usefulness	PSpace	2ExpTime	Σ_3^P	$\in$ ExpTime
Importance	PSpace	2ExpTime	$\#P^{\Sigma_2^P}$	$\in$ ExpTime

Mitigating the complexity

- We do not consider single states but sets of states corresponding to parts of the system (less syntax-sensitive).

Mitigating the complexity

- We do not consider single states but sets of states corresponding to parts of the system (less syntax-sensitive).
- Probabilistic approximations are enough for our purpose: Just draw orders at random and do a dichotomic search for the critical state.

An importance value for CTL

CTL combines **path-** and **state-formulas**. (e.g. EGa)

An importance value for CTL

CTL combines **path-** and **state-formulas**. (e.g. EGa)

Semantics: **infinite trees**, corresponding to the **unfolding of the system**.

An importance value for CTL

CTL combines **path-** and **state-formulas**. (e.g. EGa)

Semantics: **infinite trees**, corresponding to the **unfolding of the system**.

$\rightarrow$ The nondeterminism is captured by this infinite tree.

An importance value for CTL

CTL combines **path-** and **state-formulas**. (e.g. EGa)

Semantics: **infinite trees**, corresponding to the **unfolding of the system**.

$\rightarrow$ The nondeterminism is captured by this infinite tree.

Let's consider **modal transition systems.**

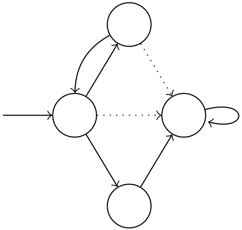
An importance value for CTL

CTL combines **path-** and **state-formulas**. (e.g. EGa)

Semantics: **infinite trees**, corresponding to the **unfolding of the system**.

$\rightarrow$ The nondeterminism is captured by this infinite tree.

Let's consider **modal transition systems.**



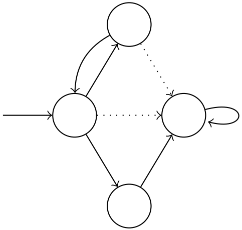
An importance value for CTL

CTL combines **path-** and **state-formulas**. (e.g. EGa)

Semantics: **infinite trees**, corresponding to the **unfolding of the system**.

$\rightarrow$ The nondeterminism is captured by this infinite tree.

Let's consider **modal transition systems.**



It is not clear how to design a **turn-based game**.

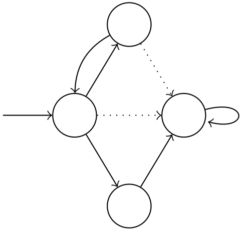
An importance value for CTL

CTL combines **path-** and **state-formulas**. (e.g. EGa)

Semantics: **infinite trees**, corresponding to the **unfolding of the system**.

$\rightarrow$ The nondeterminism is captured by this infinite tree.

Let's consider **modal transition systems.**



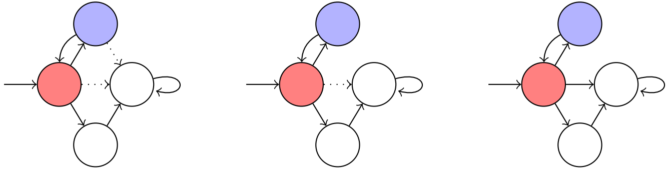
It is not clear how to design a **turn-based game**.

We considered **one-shot** games instead.

CTL complexity

We consider two interpretations for CTL:

1) One player chooses all its transitions, then the other one does.

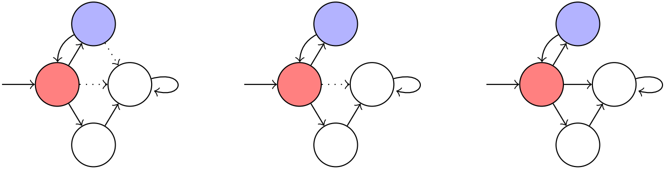


2) Both players choose their transitions concurrently.

CTL complexity

We consider two interpretations for CTL:

1) One player chooses all its transitions, then the other one does.



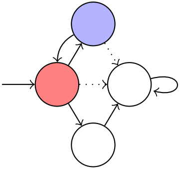
- The game is asymmetric, one of the players has an advantage.

2) Both players choose their transitions concurrently.

CTL complexity

We consider two interpretations for CTL:

1) One player chooses all its transitions, then the other one does.



- The game is asymmetric, one of the players has an advantage.

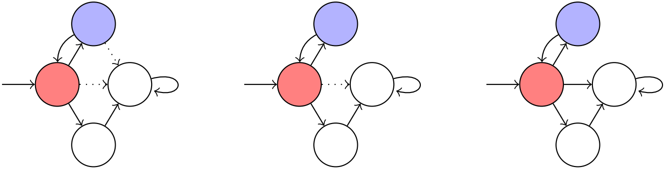
2) Both players choose their transitions concurrently.

- Players may use randomized strategies to choose their transitions.

CTL complexity

We consider two interpretations for CTL:

1) One player chooses all its transitions, then the other one does.



- The game is asymmetric, one of the players has an advantage.

2) Both players choose their transitions concurrently.

- Players may use randomized strategies to choose their transitions.
- Computing the value of a set of states comes down to solving a linear optimization problem with exponential input.

Tree games

Definition

A *tree arena* is a pair (V, E) with V a finite set and $E \subseteq V \times 2^V$.

Tree games

Definition

A *tree arena* is a pair (V, E) with V a finite set and $E \subseteq V \times 2^V$.

Definition

A *tree game* is played over a tree arena (V, E), with an initial vertex $init$, a partition $V = V_{Sat} \sqcup V_{Unsat}$ and an objective (a tree language) Ω.

Tree games

Definition

A *tree arena* is a pair (V, E) with V a finite set and $E \subseteq V \times 2^V$.

Definition

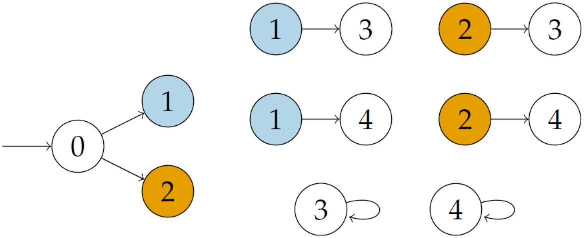
A *tree game* is played over a tree arena (V, E), with an initial vertex $init$, a partition $V = V_{Sat} \sqcup V_{Unsat}$ and an objective (a tree language) Ω.

Definition

A *linear strategy* is a function $\sigma : E^* \to E$.

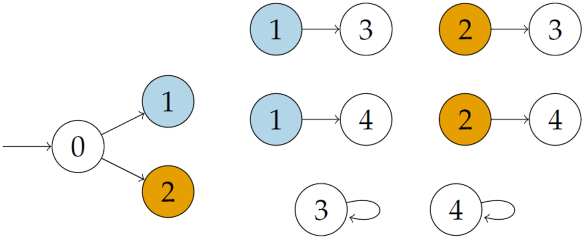
Sat (resp. Unsat) wins if they have a strategy to guarantee that the resulting tree (un)satisfies the objective. The game is sometimes undetermined.

Example of undeterminacy



In this example we consider the CTL formula $EF3 \wedge EF4$.

Example of undeterminacy



In this example we consider the CTL formula $EF3 \wedge EF4$.

Question

Which tree objectives allow tree games to be determined?

Other examples

$\rightarrow EFa$ is a determined language.

Other examples

$\rightarrow$ EFa is a determined language.

$\rightarrow$ $AG(a \vee EFb)$ as well.

Other examples

$\rightarrow$ EFa is a determined language.

$\rightarrow$ $AG(a \vee EFb)$ as well.

$\rightarrow$ $AG(EFa \wedge EFb)$ is not.

Game automata

Definition

A *game automaton* is an alternating tree automaton in which a pair (q, i) appears at most once in each transition.

Proposition

Languages expressed by game automata yield determined tree games.

However there exist languages non expressible by game automata which yield determined tree games.
It is the case with the language of trees having countably many branches fully labelled with a.

Continuation

$\rightarrow$ Fragment of CTL for which tree games are determined.

Continuation

- → Fragment of CTL for which tree games are determined.
- → Probabilistic approximation methods for Shapley values in this framework.

Continuation

- $\rightarrow$ Fragment of CTL for which tree games are determined.
- $\rightarrow$ Probabilistic approximation methods for Shapley values in this framework.
- $\rightarrow$ Extension to probabilistic systems.

Continuation

- → Fragment of CTL for which tree games are determined.
- → Probabilistic approximation methods for Shapley values in this framework.
- → Extension to probabilistic systems.
- → Control point of view (make the adversary all-knowing)

Thank you for your attention!