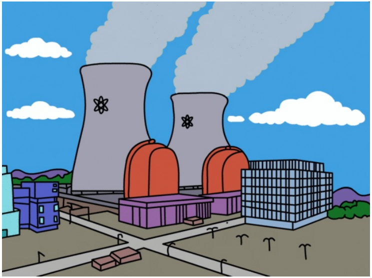
Uniform strategies

Sophie Pinchinat
IRISA/INRIA/Université de Rennes 1
joint work with Bastien Maubert and Laura Bozzelli

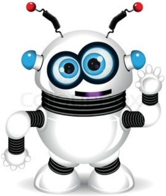
French Symposium on Games
Theory and Applications
Paris, 26 - 30 May 2015

Plan

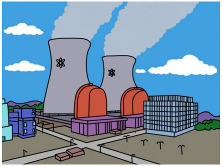
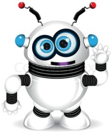
Nuclear power plant



Nuclear power plant



Controller

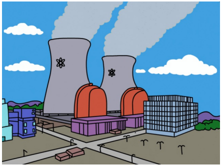
 ⟷ 

Formalization

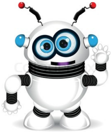
- Two-player game between the plant and the controller
- Purpose: compute a good strategy for the controller

What is a "good" strategy?

- Winning:
  - The plant must never explode: temporal winning conditions
  - Spy must ignore defects: epistemic winning conditions
- Consistent with observation/knowledge



⟷

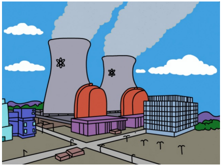


Formalization

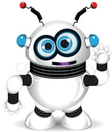
- Two-player game between the plant and the controller
- Purpose: compute a good strategy for the controller

What is a “good” strategy?

- Winning:
    - The plant must never explode: temporal winning conditions
    - Spy must ignore defects: epistemic winning conditions
- Consistent with observation/knowledge



⟷

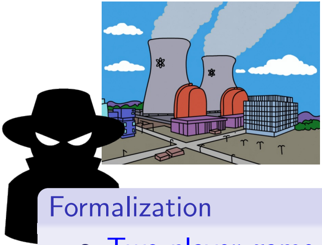


Formalization

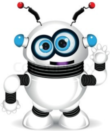
- Two-player game between the plant and the controller
- Purpose: compute a good strategy for the controller

What is a "good" strategy?

- Winning:
  - The plant must never explode: temporal winning conditions
  - Spy must ignore defects: epistemic winning conditions
- Consistent with observation/knowledge



⟷

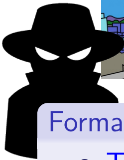


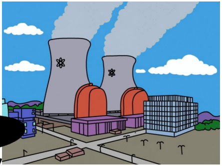
Formalization

- Two-player game between the plant and the controller
- Purpose: compute a good strategy for the controller

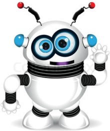
What is a "good" strategy?

- Winning:
  - The plant must never explode: temporal winning conditions
  - Spy must ignore defects: epistemic winning conditions
- Consistent with observation/knowledge





$\longleftrightarrow$

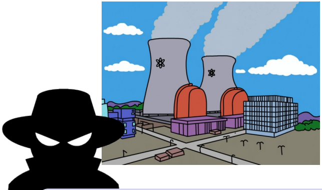


Formalization

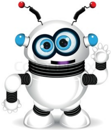
- Two-player game between the plant and the controller
- Purpose: compute a good strategy for the controller

What is a "good" strategy?

- Winning:
  - The plant must never explode: temporal winning conditions
  - Spy must ignore defects: epistemic winning conditions
- Consistent with observation/knowledge



⟷



Formalization

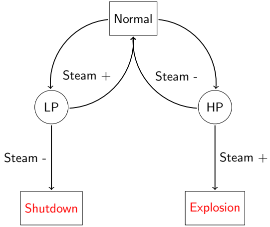
- Two-player game between the plant and the controller
- Purpose: compute a good strategy for the controller

What is a “good” strategy?

- Winning:
  - The plant must never explode: temporal winning conditions
  - Spy must ignore defects: epistemic winning conditions
- Consistent with observation/knowledge

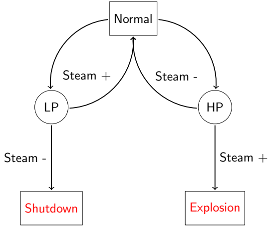
Two-player games played on graphs/arenas

Arena

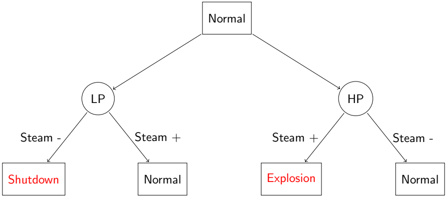


Two-player games played on graphs/arenas

Arena

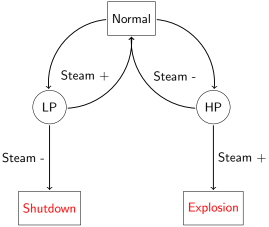


History tree (arena unfolding)

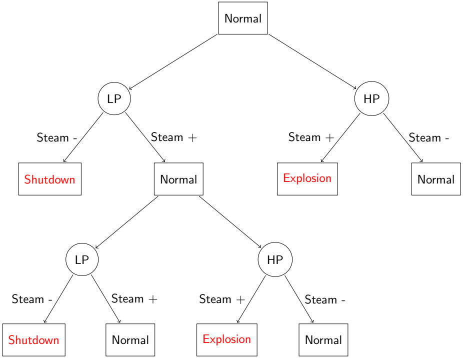


Two-player games played on graphs/arenas

Arena

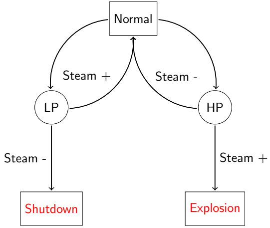


History tree (arena unfolding)

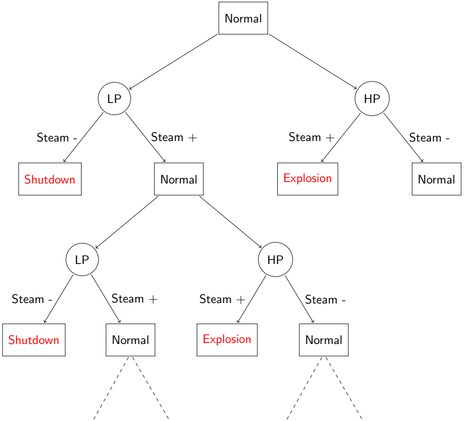


Two-player games played on graphs/arenas

Arena

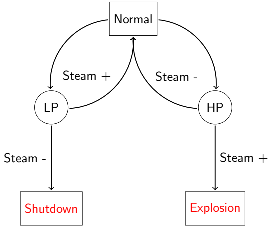


History tree (arena unfolding)

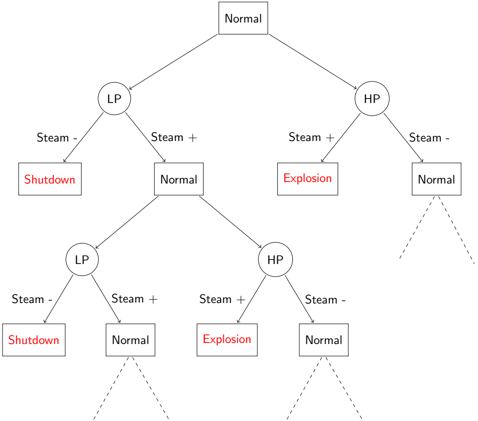


Two-player games played on graphs/arenas

Arena

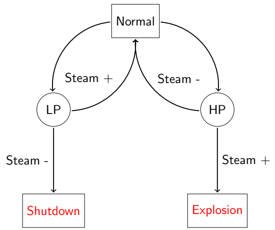


History tree (arena unfolding)



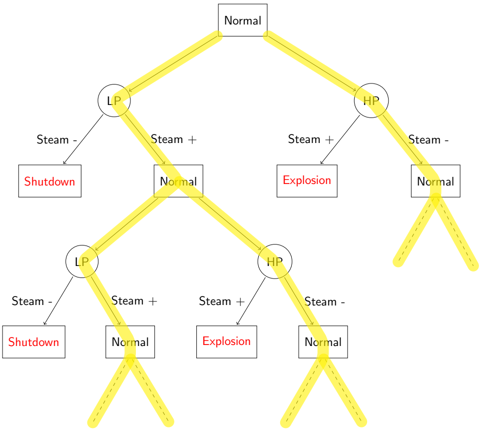
Two-player games played on graphs/arenas

Arena

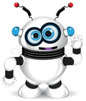


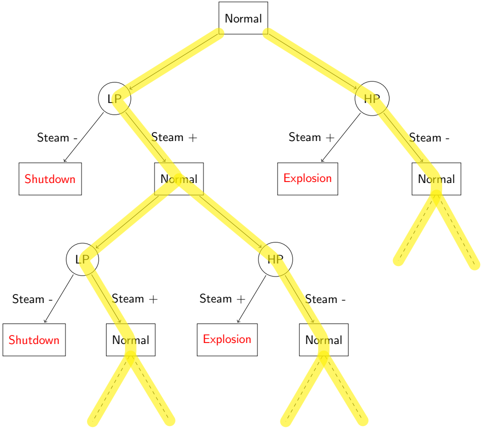
A strategy is a subtree of the history tree

History tree (arena unfolding)



Imperfect-information strategies

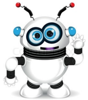


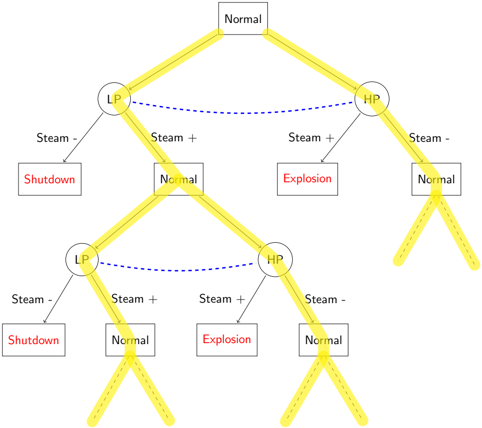


The controller confuses

Low and High Pressure (LP and HP)

Imperfect-information strategies

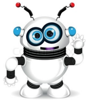


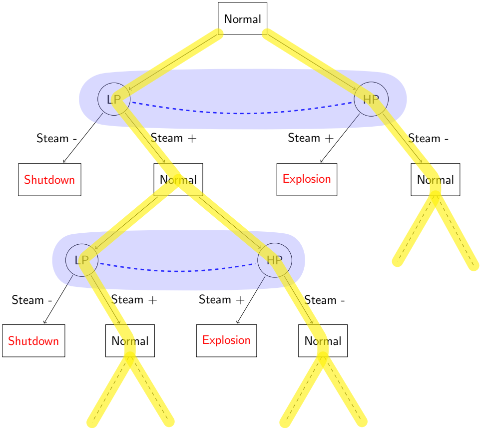


The controller confuses

Low and High Pressure (LP and HP)

Imperfect-information strategies

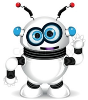


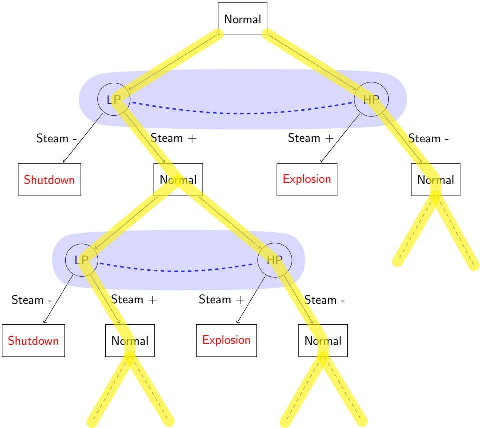


The controller confuses

Low and High Pressure (LP and HP)

Imperfect-information strategies





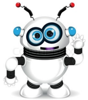
The controller confuses

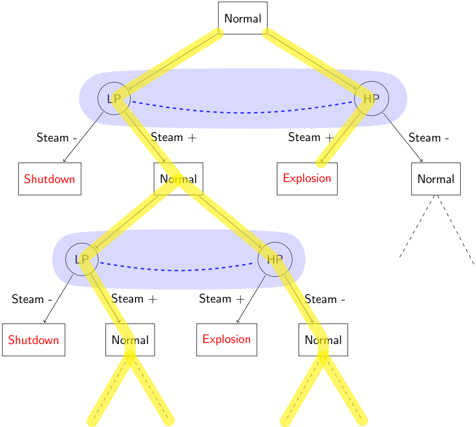
Low and High Pressure (LP and HP)

Meaningless control policies/strategies

Same decisions in indistinguishable histories.

Imperfect-information strategies





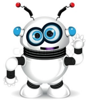
The controller confuses

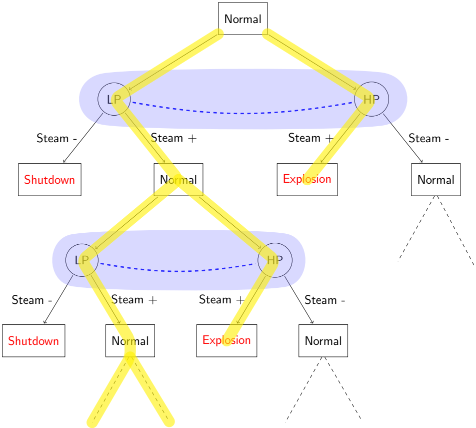
Low and High Pressure (LP and HP)

Meaningless control policies/strategies

Same decisions in indistinguishable histories.

Imperfect-information strategies





The controller confuses

Low and High Pressure (LP and HP)

Meaningless control policies/strategies

Same decisions in indistinguishable histories.

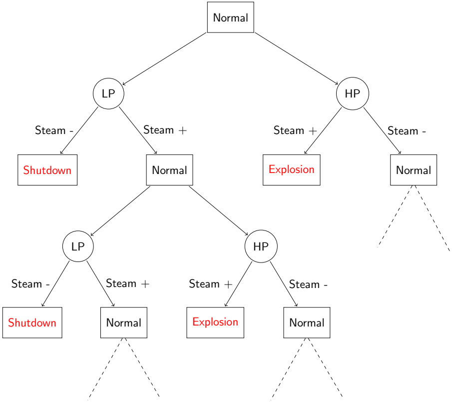
Epistemic winning conditions

Opacity

The spy must "ignore" /"never be certain" that there is a defect.

The spy observes/ears:

- number of events
- Explosion



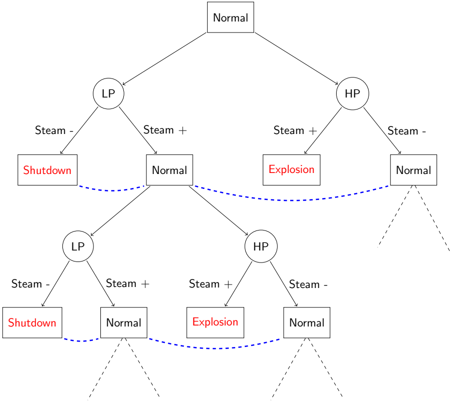
Epistemic winning conditions

Opacity

The spy must "ignore"/"never be certain" that there is a defect.

The spy observes/ears:

- number of events
- Explosion



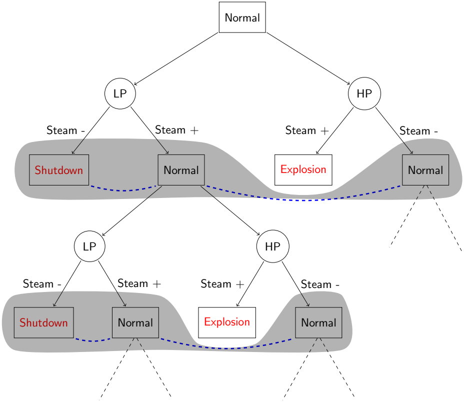
Epistemic winning conditions

Opacity

The spy must “ignore”/”never be certain” that there is a defect.

The spy observes/ears:

- number of events
- Explosion



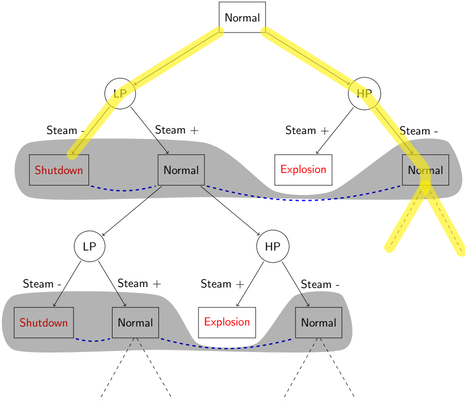
Epistemic winning conditions

Opacity

The spy must "ignore"/"never be certain" that there is a defect.

The spy observes/ears:

- number of events
- Explosion



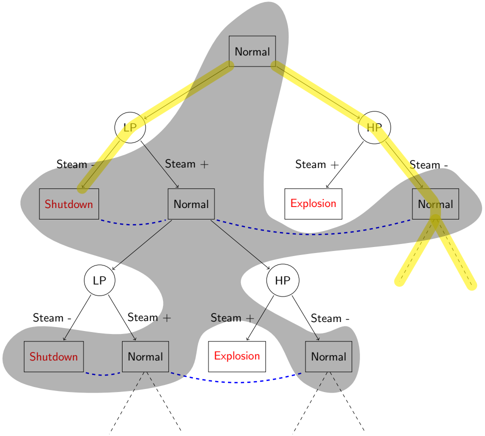
Epistemic winning conditions

Opacity

The spy must "ignore"/"never be certain" that there is a defect.

The spy observes/ears:

- ~~number of events~~
- Explosion



Epistemic winning conditions

Opacity

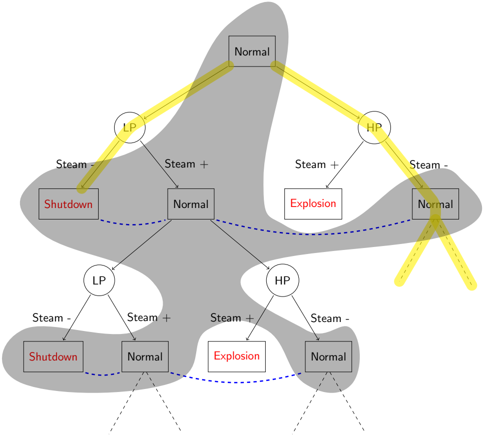
The spy must "ignore" /"never be certain" that there is a defect.

The spy observes/ears:

- ~~number of events~~
- Explosion

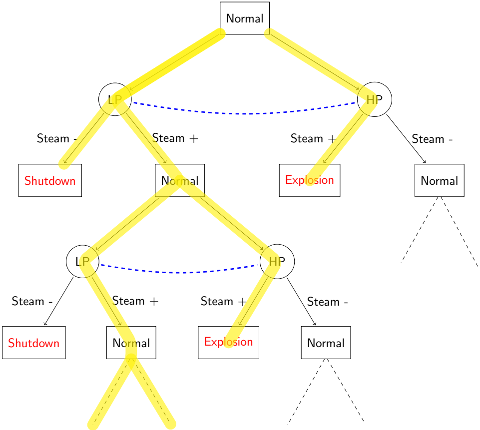
More examples:

- Anonimity
- Non-interference



A subtilty regarding relevant histories

-strategy σ is observation-based

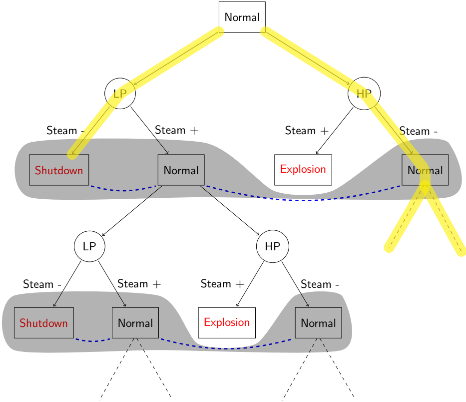


For every pair h, h' of -histories inside σ,

if h and h' are -undistinguishable, then $\sigma(h) = \sigma(h')$.
(i.e. the recommended moves coincide)

A subtlty regarding relevant histories

-strategy σ ensures 's ignorance about a defect



For every h inside σ,
there exists h' in the history tree s.t.

h and h' are -undistinguishable
and
h or h' has no defect.

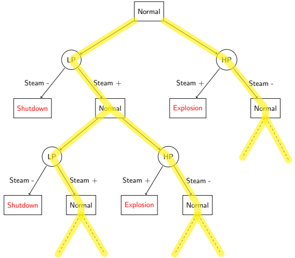
State of the Art

“Vertical” properties of strategies

Classic winning conditions

- Safety, Reachability
- all ω-regular properties (Parity LTL, CTL)

Well understood



Plethora of “transversal” properties

- Games with imperfect information [Berwanger et al. 2010]
- Games with epistemic winning conditions
  - Games with opacity condition [Bozzelli, Maubert, P. 2011]
  - Epistemic protocol synthesis [van der Meyden, Vardi 1998]
  - ATEL [van der Hoek, Wooldridge 2003] [Dima et al. 2008]
- Games for Dependence Logics [Grädel 2013]

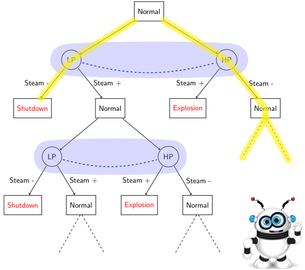
State of the Art

"Vertical" properties of strategies

Classic winning conditions

- Safety, Reachability
- all ω-regular properties (Parity LTL, CTL)

Well understood



Plethora of "transversal" properties

- Games with imperfect information [Berwanger et al. 2010]
- Games with epistemic winning conditions
  - Games with opacity condition [Bozzelli, Maubert, P. 2011]
  - Epistemic protocol synthesis [van der Meyden, Vardi 1998]
  - ATEL [van der Hoek, Wooldridge 2003] [Dima et al. 2008]
- Games for Dependence Logics [Grädel 2013]

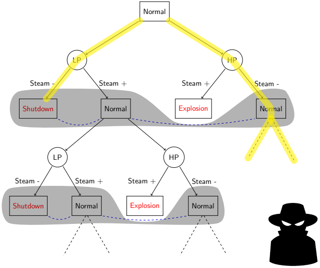
State of the Art

"Vertical" properties of strategies

Classic winning conditions

- Safety, Reachability
- all ω-regular properties (Parity LTL, CTL)

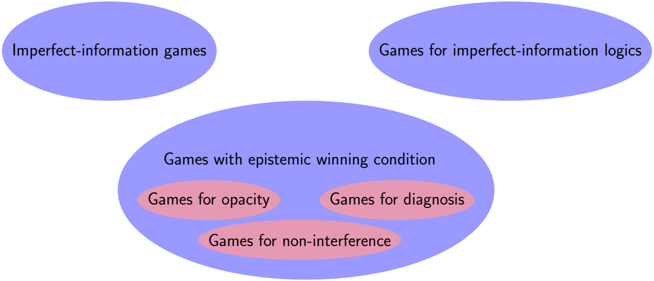
Well understood



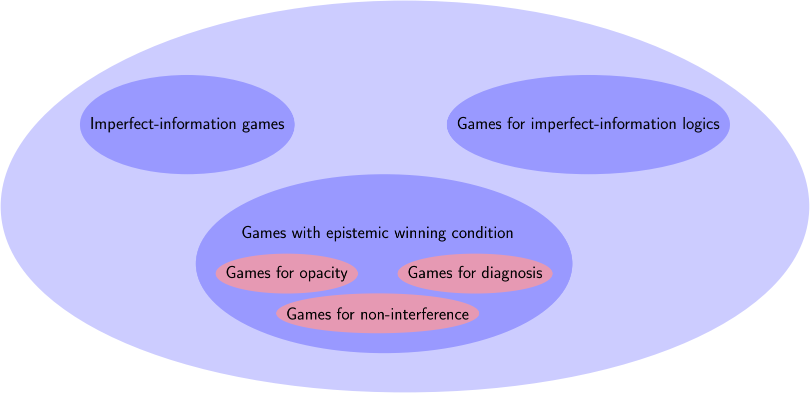
Plethora of "transversal" properties

- Games with imperfect information [Berwanger et al. 2010]
- Games with epistemic winning conditions
  - Games with opacity condition [Bozzelli, Maubert, P. 2011]
  - Epistemic protocol synthesis [van der Meyden, Vardi 1998]
  - ATEL [van der Hoek, Wooldridge 2003] [Dima et al. 2008]
- Games for Dependence Logics [Grädel 2013]

Uniform Strategies for "Transversal" properties



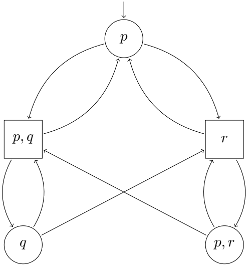
Uniform Strategies for “Transversal” properties

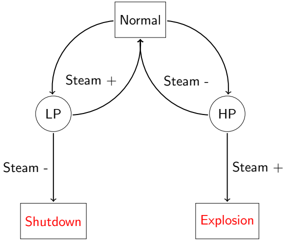


A General Notion of Uniform Strategies.
Maubert and P.
International Game Theory Review, 16(1), 2014.

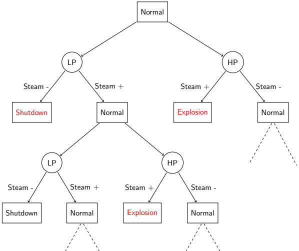
The theory of Uniform Strategies

- Arenas

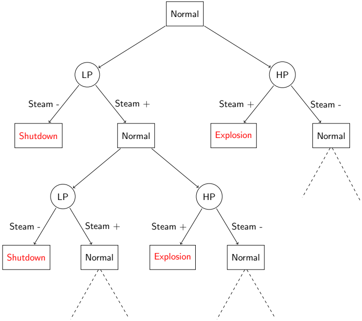




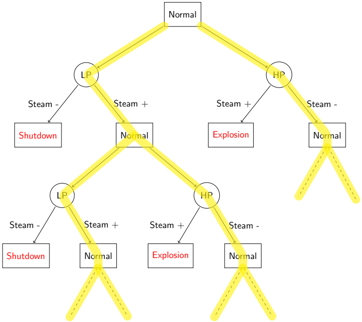
- History trees (unfolding of the arena)



A language $\mathcal{L}_{\circlearrowleft}$ to mix vertical and horizontal properties

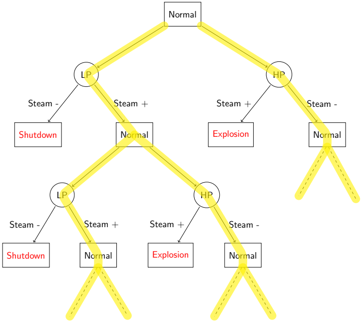


A language $\mathcal{L}_{\circlearrowleft}$ to mix vertical and horizontal properties



- $\mathcal{H} \subseteq V^*$: the set of histories (unfolding of the arena)
- $\circlearrowleft \subseteq \mathcal{H} \times \mathcal{H}$
- $\sigma \subseteq \mathcal{H}$: a strategy

A language $\mathcal{L}_{\curvearrowright}$ to mix vertical and horizontal properties



- $\mathcal{H} \subseteq V^*$: the set of histories (unfolding of the arena)
- $\curvearrowright \ \subseteq \mathcal{H} \times \mathcal{H}$
- $\sigma \subseteq \mathcal{H}$: a strategy

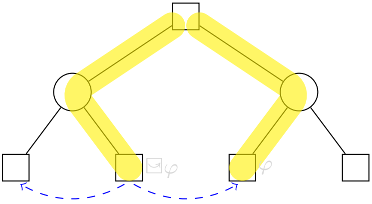
Semantics of $\varphi \in \mathcal{L}_{\curvearrowright}$: $(\mathcal{H}, \curvearrowright), \sigma, h \models \varphi$

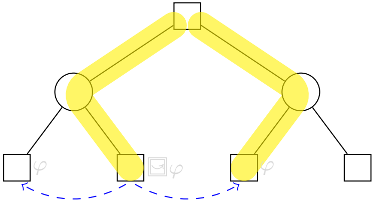
- $\mathbf{A}\psi, \mathbf{G}\psi, \ldots$: classic temporal logic semantics
- $(\mathcal{H}, \curvearrowright), \sigma, h \models \Box\varphi$
  if, for all $h' \in \sigma$, $h \curvearrowright h'$ implies $(\mathcal{H}, \curvearrowright), \sigma, h' \models \varphi$
- $(\mathcal{H}, \curvearrowright), \sigma, h \models \boxed{\Box}\varphi$
  if, for all $h' \in \mathcal{H}$, $h \curvearrowright h'$ implies $(\mathcal{H}, \curvearrowright), \sigma, h' \models \varphi$

A language $\mathcal{L}_{\rightsquigarrow}$ to mix vertical and horizontal properties

Semantics of $\varphi \in \mathcal{L}_{\rightsquigarrow}$: $(\mathcal{H}, \rightsquigarrow), \sigma, h \models \varphi$

- $\mathbf{A}\psi, \mathbf{G}\psi, \ldots$: classic temporal logic semantics
- $(\mathcal{H}, \rightsquigarrow), \sigma, h \models \Box_{\rightsquigarrow}\varphi$
  if, for all $h' \in \sigma$, $h \rightsquigarrow h'$ implies $(\mathcal{H}, \rightsquigarrow), \sigma, h' \models \varphi$
- $(\mathcal{H}, \rightsquigarrow), \sigma, h \models \boxed{\Box_{\rightsquigarrow}}\varphi$
  if, for all $h' \in \mathcal{H}$, $h \rightsquigarrow h'$ implies $(\mathcal{H}, \rightsquigarrow), \sigma, h' \models \varphi$





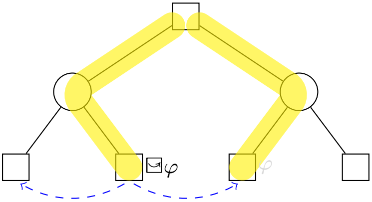
$\Box_{\rightsquigarrow}$: Strict quantifier

$\boxed{\Box_{\rightsquigarrow}}$: Full quantifier

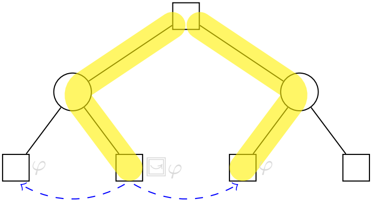
A language $\mathcal{L}_{\rightarrowtail}$ to mix vertical and horizontal properties

Semantics of $\varphi \in \mathcal{L}_{\rightarrowtail}$: $(\mathcal{H}, \rightarrowtail), \sigma, h \models \varphi$

- $\mathbf{A}\psi, \mathbf{G}\psi, \ldots$: classic temporal logic semantics
- $(\mathcal{H}, \rightarrowtail), \sigma, h \models \Box\varphi$
  if, for all $h' \in \sigma$, $h \rightarrowtail h'$ implies $(\mathcal{H}, \rightarrowtail), \sigma, h' \models \varphi$
- $(\mathcal{H}, \rightarrowtail), \sigma, h \models \boxed{\Box}\varphi$
  if, for all $h' \in \mathcal{H}$, $h \rightarrowtail h'$ implies $(\mathcal{H}, \rightarrowtail), \sigma, h' \models \varphi$



$\Box$: Strict quantifier

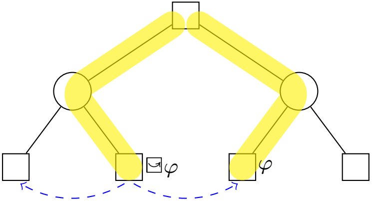


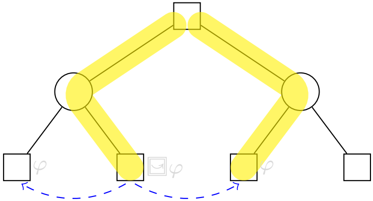
$\boxed{\Box}$: Full quantifier

A language $\mathcal{L}_{\curvearrowright}$ to mix vertical and horizontal properties

Semantics of $\varphi \in \mathcal{L}_{\curvearrowright}$: $(\mathcal{H}, \curvearrowright), \sigma, h \models \varphi$

- $\mathbf{A}\psi, \mathbf{G}\psi, \ldots$: classic temporal logic semantics
- $(\mathcal{H}, \curvearrowright), \sigma, h \models \Box_{\curvearrowright}\varphi$
  if, for all $h' \in \sigma$, $h \curvearrowright h'$ implies $(\mathcal{H}, \curvearrowright), \sigma, h' \models \varphi$
- $(\mathcal{H}, \curvearrowright), \sigma, h \models \boxed{\Box_{\curvearrowright}}\varphi$
  if, for all $h' \in \mathcal{H}$, $h \curvearrowright h'$ implies $(\mathcal{H}, \curvearrowright), \sigma, h' \models \varphi$





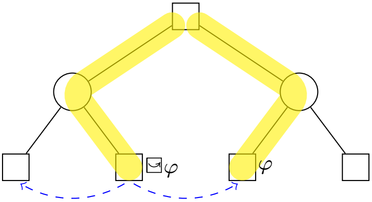
$\Box_{\curvearrowright}$: Strict quantifier

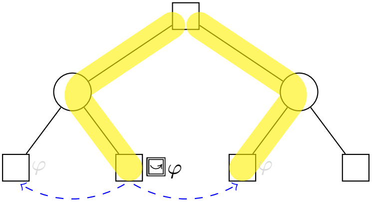
$\boxed{\Box_{\curvearrowright}}$: Full quantifier

A language $\mathcal{L}_{\curvearrowright}$ to mix vertical and horizontal properties

Semantics of $\varphi \in \mathcal{L}_{\curvearrowright}$: $(\mathcal{H}, \curvearrowright), \sigma, h \models \varphi$

- $\mathbf{A}\psi, \mathbf{G}\psi, \ldots$: classic temporal logic semantics
- $(\mathcal{H}, \curvearrowright), \sigma, h \models \Box_{\curvearrowright}\varphi$
  if, for all $h' \in \sigma$, $h \curvearrowright h'$ implies $(\mathcal{H}, \curvearrowright), \sigma, h' \models \varphi$
- $(\mathcal{H}, \curvearrowright), \sigma, h \models \boxed{\Box_{\curvearrowright}}\varphi$
  if, for all $h' \in \mathcal{H}$, $h \curvearrowright h'$ implies $(\mathcal{H}, \curvearrowright), \sigma, h' \models \varphi$





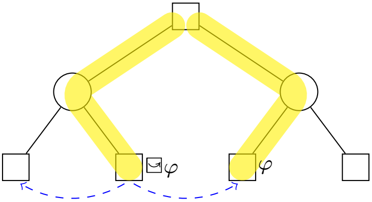
$\Box_{\curvearrowright}$: Strict quantifier

$\boxed{\Box_{\curvearrowright}}$: Full quantifier

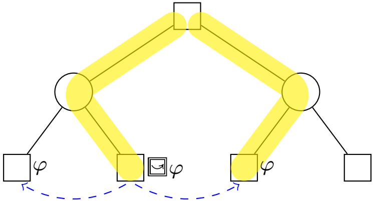
A language $\mathcal{L}_{\circlearrowleft}$ to mix vertical and horizontal properties

Semantics of $\varphi \in \mathcal{L}_{\circlearrowleft}$: $(\mathcal{H}, \circlearrowleft), \sigma, h \models \varphi$

- $\mathbf{A}\psi, \mathbf{G}\psi, \ldots$: classic temporal logic semantics
- $(\mathcal{H}, \circlearrowleft), \sigma, h \models \Box\varphi$
  if, for all $h' \in \sigma$, $h \circlearrowleft h'$ implies $(\mathcal{H}, \circlearrowleft), \sigma, h' \models \varphi$
- $(\mathcal{H}, \circlearrowleft), \sigma, h \models \boxed{\Box}\varphi$
  if, for all $h' \in \mathcal{H}$, $h \circlearrowleft h'$ implies $(\mathcal{H}, \circlearrowleft), \sigma, h' \models \varphi$



$\Box$: Strict quantifier

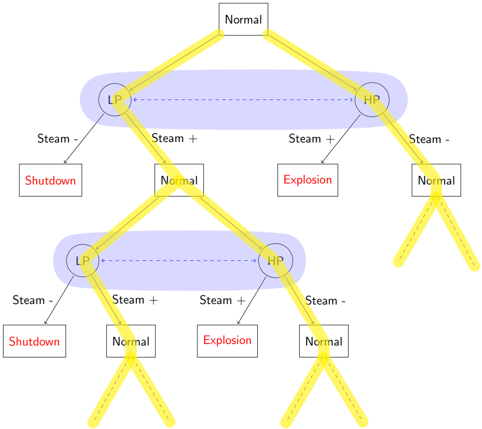


$\boxed{\Box}$: Full quantifier

Imperfect-information strategies

The controller confuses

HP and LP
(High and Low
Pressure).

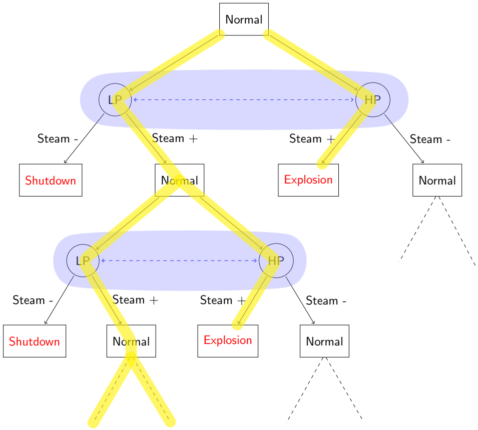


$$\mathbf{AG}(\bigcirc \rightarrow (\boxminus\mathbf{X}\ \text{Steam-} \vee \boxminus\mathbf{X}\ \text{Steam+}))$$

Imperfect-information strategies

The controller confuses

HP and LP
(High and Low
Pressure).



$$\mathbf{AG}(\bigcirc \rightarrow (\Box\mathbf{X}\ \text{Steam-} \vee \Box\mathbf{X}\ \text{Steam+}))$$

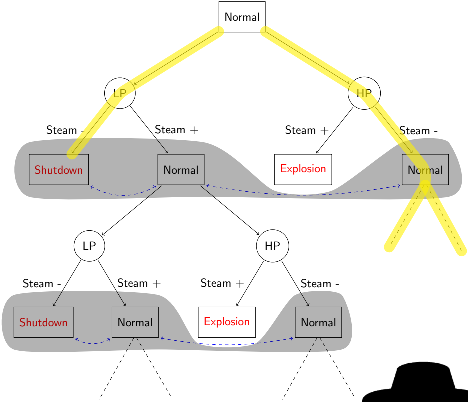
Epistemic winning conditions

Example: opacity

The spy must never be sure that there is a problem.

The spy observes:

- number of events
- exploded or not



$\mathbf{AG}\diamondsuit$ (Shutdown $\vee$ Explosion)

Usages of strictly and fully quantifiers

Strict uniformity

- Games with imperfect information [Berwanger et al. 2010]
- Games for Dependence Logics [Grädel 2013]
- Games with epistemic winning condition
  $\rightarrow$ when the agent knows the strategy
  e.g. : [van der Meyden, Vardi 1998]

Full uniformity

- Games with epistemic winning condition
  $\rightarrow$ when the agent ignores the strategy
  e.g. : ATEL, [Puchala 2010], [Bozzelli, Maubert, P. 2011]
- Dynamic Epistemic Logic (epistemic planning)

Plan

The uniform strategy problem

The uniform strategy problem

Input:

- Finite arena $\mathcal{G}$
- Relation $\looparrowright$
- $\varphi \in \mathcal{L}_{\looparrowright}$

Output:

- Is there a strategy for Player $\bigcirc$ satisfying φ?

The uniform strategy problem

The uniform strategy problem

Input:

- Finite arena $\mathcal{G}$
- Relation $\hookrightarrow$
- $\varphi \in \mathcal{L}_{\hookrightarrow}$

Output:

- Is there a strategy for Player $\bigcirc$ satisfying φ?

The uniform strategy problem

The uniform strategy problem

Input:

- Finite arena $\mathcal{G}$
- Rational relation $\leadsto$
- $\varphi \in \mathcal{L}_{\leadsto}$

Output:

- Is there a strategy for Player $\bigcirc$ satisfying φ?

The uniform strategy problem

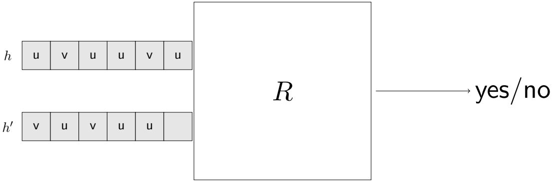
The uniform strategy problem

Input:

- Finite arena $\mathcal{G}$
- Rational relation $\looparrowright$ (recognized by transducer R)
- $\varphi \in \mathcal{L}_{\looparrowright}$

Output:

- Is there a strategy for Player $\bigcirc$ satisfying φ?



Asynchronous/Synchronous/Recognizable (bounded amount of information)

The uniform strategy problem

The uniform strategy problem

Input:

- Finite arena $\mathcal{G}$
- Rational relation $\rightsquigarrow$ (recognized by transducer R)
- $\varphi \in \mathcal{L}_{\rightsquigarrow}$

Output:

- Is there a strategy for Player $\bigcirc$ satisfying φ?

Questions:

- When is it decidable?
- When it is, what is the complexity?
- Can we automatically synthesize such strategies?

The uniform strategy problem

The uniform strategy problem

Input:

- Finite arena $\mathcal{G}$
- Rational relation $\looparrowright$ (recognized by transducer R)
- $\varphi \in \mathcal{L}_{\looparrowright}$

Output:

- Is there a strategy for Player $\bigcirc$ satisfying φ?

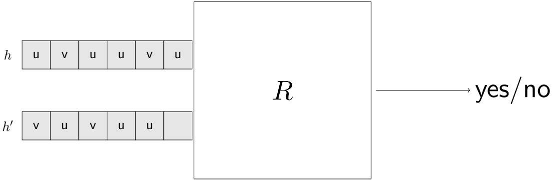
Proposition

The uniform strategy problem is 2-EXPTIME-hard.

Proof.

It contains the problem of solving games with temporal logic-definable winning conditions, which is 2-EXPTIME-complete [Kupferman, Vardi 1997]. □

Subclasses of rational relations



Recognizable $\subsetneq$ Regular $\subsetneq$ Rational

Regular relations

Recognized by a synchronous transducer. In particular, $|h| = |h'|$.

Recognizable relations

$\{h\#h' \mid h \smile h'\}$ is recognized by a finite-state word automaton

Strictly-uniform strategy problem (SUS)

Language $\mathcal{SL}_{\looparrowright}$

No use of transversal quantifiers $\boxdot$ and $\Diamond$.

Typically, observation-based strategies

$$\mathbf{AG}(\bigcirc \rightarrow (\boxdot\mathbf{X}\ \text{Steam-} \vee \boxdot\mathbf{X}\ \text{Steam+}))$$

SUS

Input:

- Finite arena $\mathcal{G}$
- A transducer R (denoting the rational binary relation $\looparrowright$)
- $\varphi \in \mathcal{SL}_{\looparrowright}$

Output:

- Is there a strategy for Player $\bigcirc$ satisfying φ?

Results on SUS

Theorem [Maubert, P. 2013]

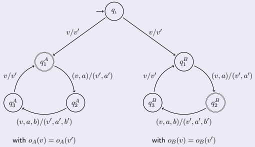
SUS is undecidable for regular equivalence relations, hence for rational relations.

Proof sketch

Reduction of the distributed strategy problem in three-player games with imperfect information.

+ Simulate several players with one:

  A nondeterministic transducer guesses the relevant relation for each play.



Results on SUS

Theorem [Maubert, P. 2013]

SUS is undecidable for regular equivalence relations, hence for rational relations.

Theorem [Maubert, P. 2013]

SUS with recognizable relations is 2-EXPTIME-complete.

Results on SUS

Theorem [Maubert, P. 2013]

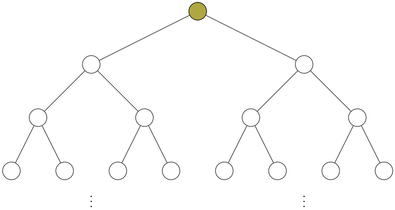
SUS is undecidable for regular equivalence relations, hence for rational relations.

Theorem [Maubert, P. 2013]

SUS with recognizable relations is 2-EXPTIME-complete.

Main tools (jumping) tree automata:

Alternating tree automata: ↓



Results on SUS

Theorem [Maubert, P. 2013]

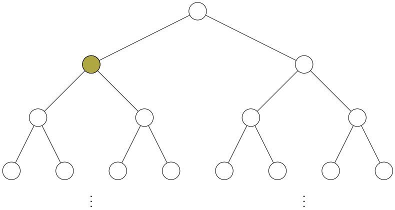
SUS is undecidable for regular equivalence relations, hence for rational relations.

Theorem [Maubert, P. 2013]

SUS with recognizable relations is 2-EXPTIME-complete.

Main tools (jumping) tree automata:

Alternating tree automata: ↓



Results on SUS

Theorem [Maubert, P. 2013]

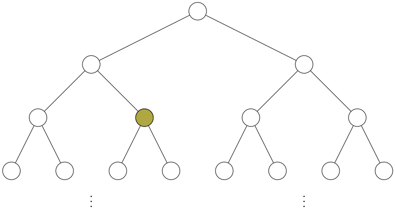
SUS is undecidable for regular equivalence relations, hence for rational relations.

Theorem [Maubert, P. 2013]

SUS with recognizable relations is 2-EXPTIME-complete.

Main tools (jumping) tree automata:

Alternating tree automata: ↓



Results on SUS

Theorem [Maubert, P. 2013]

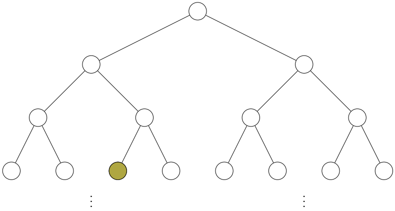
SUS is undecidable for regular equivalence relations, hence for rational relations.

Theorem [Maubert, P. 2013]

SUS with recognizable relations is 2-EXPTIME-complete.

Main tools (jumping) tree automata:

Alternating tree automata: ↓



Results on SUS

Theorem [Maubert, P. 2013]

SUS is undecidable for regular equivalence relations, hence for rational relations.

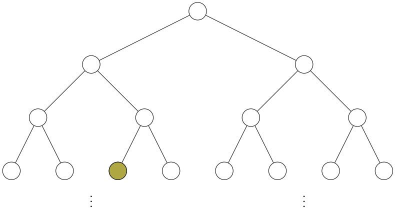
Theorem [Maubert, P. 2013]

SUS with recognizable relations is 2-EXPTIME-complete.

Main tools (jumping) tree automata:

Alternating tree automata: $\downarrow$

Two-way tree automata: $\downarrow + \uparrow$



Results on SUS

Theorem [Maubert, P. 2013]

SUS is undecidable for regular equivalence relations, hence for rational relations.

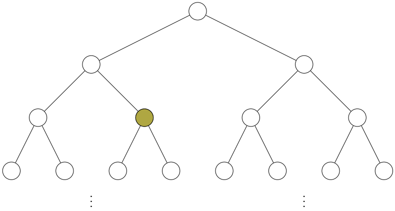
Theorem [Maubert, P. 2013]

SUS with recognizable relations is 2-EXPTIME-complete.

Main tools (jumping) tree automata:

Alternating tree automata: $\downarrow$

Two-way tree automata: $\downarrow + \uparrow$



Results on SUS

Theorem [Maubert, P. 2013]

SUS is undecidable for regular equivalence relations, hence for rational relations.

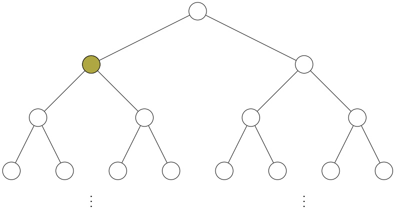
Theorem [Maubert, P. 2013]

SUS with recognizable relations is 2-EXPTIME-complete.

Main tools (jumping) tree automata:

Alternating tree automata: $\downarrow$

Two-way tree automata: $\downarrow + \uparrow$



Results on SUS

Theorem [Maubert, P. 2013]

SUS is undecidable for regular equivalence relations, hence for rational relations.

Theorem [Maubert, P. 2013]

SUS with recognizable relations is 2-EXPTIME-complete.

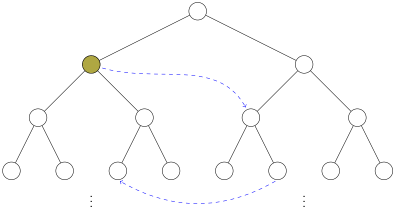
Main tools (jumping) tree automata:

Alternating tree automata: $\downarrow$

Two-way tree automata: $\downarrow + \uparrow$

Binary relation $\hookrightarrow$ over Σ^*

Jumping tree automata: $\downarrow + \hookrightarrow$



Results on SUS

Theorem [Maubert, P. 2013]

SUS is undecidable for regular equivalence relations, hence for rational relations.

Theorem [Maubert, P. 2013]

SUS with recognizable relations is 2-EXPTIME-complete.

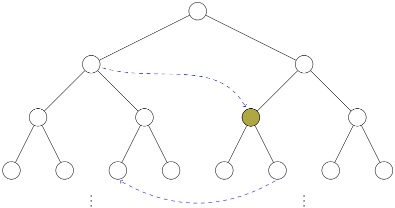
Main tools (jumping) tree automata:

Alternating tree automata: $\downarrow$

Two-way tree automata: $\downarrow + \uparrow$

Binary relation $\hookrightarrow$ over Σ^*

Jumping tree automata: $\downarrow + \hookrightarrow$



Results on SUS

Theorem [Maubert, P. 2013]

SUS is undecidable for regular equivalence relations, hence for rational relations.

Theorem [Maubert, P. 2013]

SUS with recognizable relations is 2-EXPTIME-complete.

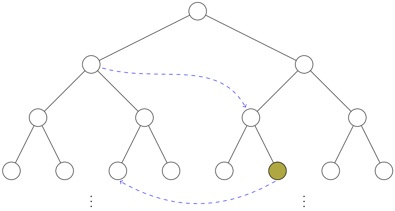
Main tools (jumping) tree automata:

Alternating tree automata: $\downarrow$

Two-way tree automata: $\downarrow + \uparrow$

Binary relation $\hookrightarrow$ over Σ^*

Jumping tree automata: $\downarrow + \hookrightarrow$



Results on SUS

Theorem [Maubert, P. 2013]

SUS is undecidable for regular equivalence relations, hence for rational relations.

Theorem [Maubert, P. 2013]

SUS with recognizable relations is 2-EXPTIME-complete.

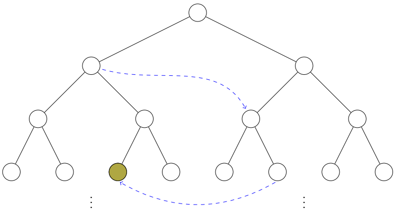
Main tools (jumping) tree automata:

Alternating tree automata: $\downarrow$

Two-way tree automata: $\downarrow + \uparrow$

Binary relation $\hookrightarrow$ over Σ^*

Jumping tree automata: $\downarrow + \hookrightarrow$



Results on SUS

Theorem [Maubert, P. 2013]

SUS is undecidable for regular equivalence relations, hence for rational relations.

Theorem [Maubert, P. 2013]

SUS with recognizable relations is 2-EXPTIME-complete.

Proposition

For every relation $\rightsquigarrow$ and every $\varphi \in \mathcal{SL}_{\rightsquigarrow}$, there is a jumping tree automaton that accepts strategies σ that fulfill φ and whose size is in $O(\exp(|\varphi|))$.

Theorem: simulation

For every jumping tree automaton with a recognizable relation, there is an equivalent nondeterministic tree automaton of exponential size.

Results on SUS

Theorem [Maubert, P. 2013]

SUS is undecidable for regular equivalence relations, hence for rational relations.

Theorem [Maubert, P. 2013]

SUS with recognizable relations is 2-EXPTIME-complete.

Proposition

For every relation $\backsim$ and every $\varphi \in \mathcal{SL}_{\backsim}$, there is a jumping tree automaton that accepts strategies σ that fulfill φ and whose size is in $O(\exp(|\varphi|))$.

Theorem: simulation

For every jumping tree automaton with a recognizable relation, there is an equivalent nondeterministic tree automaton of exponential size.

Fully-uniform strategy problem (FUS)

Language $\mathcal{FL}_{\curvearrowright}$

No use of transversal quantifiers $\boxminus$ and $\diamondminus$.

Typically, epistemic winning conditions

$$\mathbf{AG}\diamondminus\,(\text{Shutdown} \vee \text{Explosion})$$

FUS

Input:

- Finite arena $\mathcal{G}$
- A transducer R (denoting the rational binary relation $\curvearrowright$)
- $\varphi \in \mathcal{FL}_{\curvearrowright}$

Output:

- Is there a strategy for Player $\bigcirc$ satisfying φ?

Results on FUS

The $\square$-depth

$d(\varphi)$ is the maximum nesting of $\square$ quantifiers in φ.

Results on FUS

The $\square$-depth

$d(\varphi)$ is the maximum nesting of $\square$ quantifiers in φ.

Theorem [Bozzelli, Maubert, P. 2013]

For formulas of $\square$-depth bounded by d, FUS is

- 2-EXPTIME-complete if $d \leq 2$,
- d-EXPTIME-complete if $d > 2$.

Results on FUS

The □-depth

$d(\varphi)$ is the maximum nesting of □ quantifiers in φ.

Theorem [Bozzelli, Maubert, P. 2013]

For formulas of □-depth bounded by d, FUS is

- 2-EXPTIME-complete if $d \leq 2$,
- d-EXPTIME-complete if $d > 2$.

Main ingredients

1. Information set automata $\mathcal{A}^R$ that keeps track of what happens in the transducer R,
2. Synchronize $\mathcal{A}^R$ with the arena $\mathcal{G}$.
3. Iterate □-quantifier elimination
4. Solve the final game (temporal logic objective)

Results on FUS

The $\boxed{\smile}$-depth

$d(\varphi)$ is the maximum nesting of $\boxed{\smile}$ quantifiers in φ.

Theorem [Bozzelli, Maubert, P. 2013]

For formulas of $\boxed{\smile}$-depth bounded by d, FUS is

- 2-EXPTIME-complete if $d \leq 2$,
- d-EXPTIME-complete if $d > 2$.

Corollary

FUS is non-elementary.

Results on FUS

Theorem [Bozzelli, Maubert, P. 2013]

For formulas of $\boxdot$-depth bounded by d, FUS is

- 2-EXPTIME-complete if $d \leq 2$,
- d-EXPTIME-complete if $d > 2$.

Corollary

FUS is non-elementary.

No Miracles property

$h \smile h'$ implies $(h \cdot \ell) \smile (h' \cdot \ell)$, for all ℓ.

Results on FUS

Theorem [Bozzelli, Maubert, P. 2013]

For formulas of $\boxed{\leadsto}$-depth bounded by d, FUS is

- 2-EXPTIME-complete if $d \leq 2$,
- d-EXPTIME-complete if $d > 2$.

Corollary

FUS is non-elementary.

No Miracles property

$h \leadsto h'$ implies $(h \cdot \ell) \leadsto (h' \cdot \ell)$, for all ℓ.

Remark

Classic relations used in Epistemic Temporal Logic fall into this class.

Results on FUS

Theorem [Bozzelli, Maubert, P. 2013]

For formulas of $\boxtimes$-depth bounded by d, FUS is

- 2-EXPTIME-complete if $d \leq 2$,
- d-EXPTIME-complete if $d > 2$.

Corollary

FUS is non-elementary.

No Miracles property

$h \smile h'$ implies $(h \cdot \ell) \smile (h' \cdot \ell)$, for all ℓ.

Theorem

FUS with transitive, Euclidean, No Miracles relations is 2-EXPTIME-complete.

Concluding remarks

- (Finite memory) Strategies can be synthezised tree automata + associated non-emptiness parity games techniques

Concluding remarks

- (Finite memory) Strategies can be synthezised
  tree automata + associated non-emptiness parity games techniques
- Mixing $\Box$ and $\boxed{\Box}$: The uniform strategy problem is decidable (non-elementary) if
  - $\Box_i$'s go with recognizable relations
  - No $\Box_i$'s in the scope of $\boxed{\Box}_j$'s

Concluding remarks

- (Finite memory) Strategies can be synthezised
  tree automata + associated non-emptiness parity games techniques

- Mixing $\unlhd$ and $\boxed{\unlhd}$: The uniform strategy problem is decidable (non-elementary) if
  - $\unlhd_i$'s go with recognizable relations
  - No $\unlhd_i$'s in the scope of $\boxed{\unlhd}_j$'s

- Results extend to n relations $\unlhd_i$'s and $\boxed{\unlhd}_j$'s (still two players)